Intelligent, Item-Based Stereotype Recommender System

NOURAH ABDULMOHSEN AL-ROSSAIS

Ph.D. Thesis

THE UNIVERSITY of York

Computer Science

Feb 2021

To my beloved father, my role model and inspiration who has sadly passed away during my study ... You will always motivate me to achieve more in life ...

Abstract

Recommender systems (RS) have become key components driving the success of e-commerce, and other platforms where revenue and customer satisfaction is dependent on the user's ability to discover desirable items in large catalogues. As the number of users and items on a platform grows, the computational complexity, the vastness of the data, and the sparsity problem constitute important challenges for any recommendation algorithm. In addition, the most widely studied filtering-based RS, while effective in providing suggestions for established users and items, are known for their poor performance for the new user and new item (cold start) problems.

Stereotypical modelling of users and items is a promising approach to solving these problems. A stereotype represents an aggregation of the characteristics of the items or users which can be used to create general user or item classes. This work propose a set of methodologies for the automatic generation of stereotypes during the cold-starts. The novelty of the proposed approach rests on the findings that stereotypes built independently of the user-to-item ratings improve both recommendation metrics and computational performance during cold-start phases. The resulting RS can be used with any machine learning algorithm as a solver, and the improved performance gains due to rate-agnostic stereotypes are orthogonal to the gains obtained using more sophisticated solvers.

Recommender Systems using the primitive metadata features (baseline systems) as well as factorisation-based systems are used as benchmarks for state-of-the-art methodologies to assess the results of the proposed approach under a wide range of recommendation quality metrics. The results demonstrate how such generic groupings of the metadata features, when performed in a manner that is unaware and independent of the user's community preferences, may greatly reduce the dimension of the recommendation model, and provide a framework that improves the quality of recommendations in the cold start.

Contents

List of Tables . . . 10
List of Figures . . . 14
Acknowledgements . . . 15
Declaration . . . 16
Publications . . . 17

1 Introduction **18**
1.1 Background and Rationals . . . 18
1.2 Problem Statement . . . 20
1.3 Research Questions and Hypothesis . . . 22
1.4 Structure of the Thesis . . . 22

2 Literature Review **24**
2.1 Recommender System in e-commerce . . . 24
2.2 Popular Recommender Systems . . . 26
2.3 Existing Recommender System Approaches . . . 27
2.3.1 Collaborative Filtering (CF) . . . 27
2.3.2 Content-Based Filtering (CBF) . . . 29
2.3.3 Demographic . . . 30
2.3.4 Hybrid . . . 30
2.3.5 Pairwise Preference Learning . . . 31
2.4 Cold-Start Problem for Recommender Systems . . . 31
2.5 Stereotype-Based Modeling . . . 33
2.5.1 Stereotype - Definition and Evolution . . . 33
2.5.2 Stereotypes in Recommender Systems . . . 37
2.5.3 Approaches for Stereotypical Groups . . . 39
2.6 Evaluation of Recommender Systems . . . 41
2.7 Summary . . . 42

3 Problem Analysis **44**
3.1 Research Methodology . . . 45
3.2 MovieLens 1 Million Dataset . . . 46
3.3 Clustering Algorithms . . . 48

3.4 Experimental Evaluation . . . 49
3.5 Summary . . . 68

4 Automatic Construction of Item-Based Stereotypes 70
4.1 Stereotypes for Complex Categorical Features . . . 70
4.2 Stereotypes for Numerical Features . . . 74
4.3 Stereotype Creation Experiment . . . 80
4.3.1 Results with Complex Categorical Features . . . 80
4.3.2 Results with Numerical Features . . . 96
4.4 Summary . . . 102

5 Preliminary Evaluation of Stereotypes 103
5.1 Homogeneity of Training and Test Datasets . . . 104
5.2 Stereotypes Evaluation . . . 106
5.2.1 A Hard Test . . . 106
5.2.2 A Soft Test . . . 114
5.2.3 Predictive Power of Stereotypes . . . 122
5.3 Summary . . . 124

6 Stereotype-Based Recommendation Performance 125
6.1 Experimental Evaluation . . . 126
6.1.1 Cold Start Assessment of Item Consumption . . . 127
6.1.2 Cold Start Assessment of Item Rating . . . 133
6.1.3 Cold Start Assessment of Recommendations Driven by Stereotypes versus SVD-Based RS (with metadata) . . . 139
6.2 Summary . . . 149

7 Validation of the Stereotype-Driven Methodology 151
7.1 Amazon Dataset . . . 152
7.2 Constructing Item-Based Stereotypes . . . 153
7.2.1 Stereotypes for Complex Categorical Features . . . 154
7.2.2 Stereotypes for the Numerical Features . . . 159
7.3 Evaluation of Stereotypes . . . 163
7.3.1 Homogeneity of Training and Test Datasets . . . 163
7.3.2 A Hard Test . . . 166
7.3.3 A Soft Test . . . 171
7.3.4 Predictive Power of Stereotypes . . . 174
7.4 Recommendation Performance . . . 176
7.4.1 Cold Start Assessment of Item Rating . . . 177
7.4.2 Cold Start Assessment of Recommendations Driven by Stereotypes versus SVD-Based RS (with metadata) . . . 184
7.5 Summary . . . 191

8 Conclusion and Future Work 193
8.1 Conclusion . . . 193
8.2 Contribution to knowledge . . . 196
8.3 Limitations and Future Work . . . 199

References 201

List of Tables

3.1 Combined ML 1 Million/IMDb movie and user features 47
3.2 Movie features transformed to a numerical feature vector for Analysis 1 and Analysis 1.B 49
3.3 Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to k-means 59
3.4 Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to GMM 59
3.5 Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for DBSCAN 59
3.6 Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to k-means for Analysis 1.B 64
3.7 Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to gaussian mixture model (GMM) for Analysis 1.B 64
3.8 Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for DBSCAN for Analysis 1B 64
3.9 Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to k-means for Analysis 1.C 67
3.10 Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to GMM for Analysis 1.C 68
3.11 Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for DBSCAN for Analysis 1.C 68

4.1 Stereotypes automatically generated using algorithm 1 for the feature: genre and keywords 92
4.2 k-modes resulting centroids composition for 5 clusters and the genre feature, with two alternative methodologies for the initialisation of the position of the centroids 94
4.3 k-modes resulting centroids composition for 10 clusters and the genre feature, with two alternative methodologies for the initialisation of the position of the centroids 94

4.4 Centroid composition identified by k-modes for the feature keywords with Huang initialisation methodology and k= 20 . . . 95
4.5 Filtered modes discovered and ranked via the persistence algorithm for feature: log(budget+1) 97
4.6 Filtered modes discovered and ranked via the persistence algorithm for feature: log(revenue+1) 97
4.7 Filtered modes discovered and ranked via the persistence algorithm for feature: director popularity . . . 97
4.8 Filter modes discovered and ranked via the persistence algorithm for feature: country distance . . . 97
4.9 Filtered modes discovered and ranked via the persistence algorithm for feature: cast popularity . . . 97
4.10 Filter modes discovered and ranked via the persistence algorithm for feature: language popularity . . . 97
4.11 Filter modes discovered and ranked via the persistence algorithm for feature: cast gender bias . . . 98
4.12 Filtered modes discovered and ranked via the persistence algorithm for feature: popularity of movie . . . 98
4.13 Filtered modes discovered and ranked via the persistence algorithm for feature: production company popularity . . . 98
4.14 Filtered modes discovered and ranked via the persistence algorithm for feature: release time of the year . . . 98
4.15 Filtered modes discovered and ranked via the persistence algorithm for feature: vote average 98
4.16 Filtered modes discovered and ranked via the persistence algorithm for feature: release year 98
4.17 Filtered modes discovered and ranked via the persistence algorithm for feature: runtime . 99
4.18 Filtered modes discovered and ranked via the persistence algorithm for feature: log(vote count) . . . 99
4.19 Classification of numerical features between Type I (stereotyping can be done via the modes) and Types II (stereotyping can be done via percentile intervals) . . . 99
4.20 Stereotypes for feature: log(budget+1) . . . 100
4.21 Stereotypes for feature: log(revenue+1) . . . 100
4.22 Stereotypes for feature: director popularity . . . 100
4.23 Stereotypes for feature: country distance . . . 100
4.24 Stereotypes for feature: cast popularity . . . 100
4.25 Stereotypes for feature: language popularity . . . 100
4.26 Stereotypes for feature: cast gender bias . . . 100
4.27 Stereotypes for feature: popularity of movie . . . 101
4.28 Stereotypes for feature: production company popularity . . . 101
4.29 Stereotypes for feature: release time of the year . . . 101
4.30 Stereotypes for feature: vote average . . . 101
4.31 Stereotypes for feature: release year . . . 101
4.32 Stereotypes for feature: runtime . . . 101
4.33 Stereotypes for feature: log(vote count) . . . 102

5.1 Stereotypes automatically generated using algorithm 1 for the feature genre over the test dataset only . . . 109
5.2 Hard Test comparison statistics for the stereotypes of the feature genre generated over the training vs the test datasets . . . 109

5.3 Stereotypes for the keywords feature using the training dataset and the list of enlarged keywords are shown on the left. The keywords highlighted are those that meet the filter condition in the test dataset. The right column show the stereotypes for the keywords using test dataset . . . 110
5.4 Comparison statistics for the stereotypes of the feature keywords generated over the test vs the training datasets . . . 110
5.5 Hard test metrics for the evaluation of the numerical stereotypes. The table reports for all features and all stereotypes the dissimilarity in probability mass and centre, and the accuracy from relationships 5.4. The last column gives standard metric of the accuracy . . 113
5.6 Stereotypes automatically generated using algorithm 1 for the feature genre over the Full dataset . . . 116
5.7 Stereotypes automatically generated using algorithm 1 for the feature keywords over the Full dataset and the keywords enlarged set . . . 117
5.8 Statistics describing the mismatch ratios for the stereotypical representations of complex categorical features for the items in test dataset using Full vs the Training stereotypes . . . 118
5.9 An example of confusion matrix. Highlighted the areas of True Positives, True Negatives, False Positives, False Negatives for the prediction of s1 . . . 119
5.10 Confusion matrix for the feature: cast gender bias . . . 119
5.11 Confusion matrix for the feature: cast popularity . . . 120
5.12 Confusion matrix for the feature: country distance . . . 120
5.13 Confusion matrix for the feature: director popularity . . . 120
5.14 Confusion matrix for the feature: log budget . . . 120
5.15 Confusion matrix for the feature: log revenue . . . 120
5.16 Confusion matrix for the feature: log vote count . . . 120
5.17 Confusion matrix for the feature: popularity . . . 120
5.18 Confusion matrix for the feature: production company popularity . . . 121
5.19 Confusion matrix for the feature: release time of the year . . . 121
5.20 Confusion matrix for the feature: release year . . . 121
5.21 Confusion matrix for the feature: runtime . . . 121
5.22 Confusion matrix for the feature: vote average . . . 121
5.23 Numerical features stereotypes evaluation, soft test. F1-score and accuracy metrics for the classification problem of the test items using the stereotypes generated on full items . . . 122
5.24 Summary for all features of the search for explanatory power of stereotypes via the Agresti-Coull test. Confidence level of 99% . . . 124

6.1 Classification-prediction metrics derived from the confusion matrices, including the area under the curve (AUC) for both the receiver operating characteristic (ROC) and the precision-recall curve (PRC) for the new-user and new-item experiments in the ML/IMDb. (T.P. refers to true positive, and F.P. refers to false positive) . . . 128
6.2 New User: Top-N Recommendations - performance metrics of stereotype models $NNSR_c$, $NNSR_n$, $NNSR_{com}$ vs baseline model $NNSR_b$, plus performance increase and p-value of the test on the significance of the increased performance due to stereotypes . . . 129
6.3 New Item: Top-N Recommendations - performance metrics of stereotype models $NNSR_c$, $NNSR_n$, $NNSR_{com}$ vs baseline model $NNSR_b$, plus performance increase and p-value of the test on the significance of the increased performance due to stereotypes . . . 129
6.4 Handling imbalanced dataset for new user problem . . . 133
6.5 Performance metrics: new user problem . . . 137
6.6 Performance metrics: new item problem . . . 137
6.7 RMSE obtained by [1] . . . 138

6.8 Performance metrics XGBoost regression for the mixed case . . . 139
6.9 New user and new item cold start comparisons between the recommendation models: stereotypes and SVD with and without metadata . . . 142
6.10 Hit rate for top-N recommendation list . . . 142
6.11 Mean reciprocal rank (MRR) and mean average precision (MAP) . . . 143
6.12 Example evaluation of DCG@10 following equation 6.6 . . . 145
6.13 Example evaluation of IDCG@10 following equation 6.8 . . . 145
6.14 Comparison nDCG for model with stereotype and SVD with metadata . . . 146
6.15 Comparison half-life utility for model with stereotypes and SVD with metadata using a decay factor α equal to 3 . . . 147

7.1 Proportion of products with features available for Amazon dataset . . . 153
7.2 Statistics of Amazon dataset . . . 153
7.3 Stereotypes automatically generated using algorithm 1 for the feature: Categories. Amazon product group: 'Sports & Outdoors' . . . 160
7.4 Stereotypes automatically generated using algorithm 1 for the feature: Categories. Amazon product group: 'Clothing, Shoes and Jewellery' . . . 161
7.5 Stereotypes for feature: Price. Amazon product group: 'Sports & Outdoor' . . . 162
7.6 Stereotypes for feature: Brand Popularity. Amazon product group: 'Sports & Outdoor' . . . 162
7.7 Stereotypes for feature: log(sales rank). Amazon product group: 'Sports & Outdoor' . . . 162
7.8 Stereotypes for feature: Price. Amazon product group: 'Clothing, Shoes and Jewellery' . . . 162
7.9 Stereotypes for feature: log(brand popularity). Amazon product group: 'Clothing, Shoes and Jewellery' . . . 163
7.10 Stereotypes for feature: log(sales rank). Amazon product group: 'Clothing, Shoes and Jewellery' . . . 163
7.11 Stereotypes automatically generated using Algorithm 1 for the feature 'categories' in the test data. Amazon product group: 'Sports & Outdoors' . . . 168
7.12 Stereotypes automatically generated using Algorithm 1 for the feature: 'categories' in the test data. Amazon product group: 'Clothing, Shoes and Jewellery' . . . 169
7.13 Hard test comparison statistics for the stereotypes of the feature 'categories' generated over the training vs the test datasets . . . 169
7.14 Hard test metrics for the evaluation of the numerical stereotypes. The table reports for all numerical features and all stereotypes the dissimilarity in probability mass and centre, and the accuracy from equation 5.4. The last column gives standard metric of the accuracy. Amazon product group: 'Sports & Outdoors' . . . 170
7.15 Hard test metrics for the evaluation of the numerical stereotypes. The table reports for all numerical features and all stereotypes the dissimilarity in probability mass and centre, and the accuracy from equation 5.4. The last column gives standard metric of the accuracy. Amazon product group: 'Clothing, Shoes and Jewelry' . . . 170
7.16 Statistics describing the mismatch ratios for the stereotypical representations of 'categories' feature for the items in test dataset using full vs the training stereotypes . . . 172
7.17 Confusion matrix for the feature 'price'. Amazon product group: Sports & Outdoors . . . 172
7.18 Confusion matrix for the feature 'brand popularity'. Amazon Product Group: Sports & Outdoors . . . 172
7.19 Confusion matrix for the feature 'sales rank'. Amazon product group: Sports & Outdoors 173
7.20 Confusion matrix for the feature 'price'. Amazon product group: Clothing, Shoes and Jewellery . . . 173
7.21 Confusion matrix for the feature 'brand popularity'. Amazon product group: Clothing, Shoes and Jewellery . . . 173

7.22 Confusion matrix for the feature 'sales rank'. Amazon product group: Clothing, Shoes and Jewellery 173
7.23 Numerical features stereotypes evaluation, soft test. F1-score and accuracy metrics for the classification problem of the test items using the stereotypes generated on training items . 173
7.24 Summary for all features of the search for explanatory power of stereotypes via the Agresti-Coull test. confidence level of 95%. Amazon product group: Sports & Outdoors 175
7.25 Summary for all features of the search for explanatory power of stereotypes via the Agresti-Coull test. Confidence level of 95%. Amazon product group: Clothing, Shoes and Jewellery 175
7.26 Performance metrics for new user problem - Amazon dataset 181
7.27 Performance metrics for new item problem - Amazon dataset 181
7.28 New user and new item cold start comparisons between the recommendation models: stereotypes and SVD with and without metadata 185
7.29 Hit rate for Top-N recommendation list - Amazon dataset 186
7.30 Mean reciprocal rank (MRR) and mean average precision (MAP) - Amazon dataset . . . 188
7.31 Comparison nDCG for model with stereotypes and SVD with metadata - Amazon dataset . 188
7.32 Comparison half-life utility for model with stereotypes and SVD with metadata using a decay factor α equal to 3 - Amazon dataset 189

List of Figures

2.1 Sample triggers by Rich [2] . . . 35
2.2 Stereotype hierarchy as developed in GRUNDY by Rich [2] . . . 35

3.1 Description of the overall research methodology . . . 45
3.2 Problem analysis process . . . 47
3.3 Elbow method considerations for k-means using 'fit' a standardised Eucliedean measure . . . 52
3.4 Elbow method considerations for k-means using 'fit' a sklearn score measure . . . 53
3.5 Elbow method considerations for GMM using a standardized Eucliedean measure 'fit' . . . 54
3.6 Elbow method considerations for GMM using a sklearn score measure 'fit' . . . 54
3.7 Approximation of the first derivative of score for the GMM elbow procedure. Two plateaus can be identified suggesting a first elbow at around k= 5 and a second elbow at around k= 20 . . . 55
3.8 Approximation of the first derivative of the distortion for the GMM elbow procedure. Two plateaus can be identified suggesting a first elbow at around k= 5 and a second elbow at around k= 20 . . . 55
3.9 Silhouette score as a function of the two parameters of the DBSCAN algorithm. Elbow method considerations suggest an epsilon of about 20 . . . 56
3.10 Ranking feature separation in feature space using $\delta_j^{n,a}$ . . . 58
3.11 Elbow method considerations for k-means, Analysis 1 vs Analysis 1.B . . . 62
3.12 Elbow method considerations for GMM, Analysis 1 vs Analysis1.B . . . 62
3.13 Elbow method considerations for the silhouette score over the parameter space for DBSCAN, Analysis 1 vs Analysis 1.B . . . 62
3.14 Elbow method considerations for k-means, Analyses 1 vs 1.B vs 1.C . . . 66
3.15 Elbow method considerations for GMM, Analyses 1 vs 1.B vs 1.C . . . 66
3.16 Silhouette score as a function of the two parameters of the DBSCAN algorithm. Elbow method considerations suggest an epsilon of about 20 across all analyses . . . 67

4.1 Probability density approximation via histograms and KDE for the features: log(budget+1) and log(revenue+1) . . . 76
4.2 Probability density approximation via histograms and KDE for the features: cast popularity and cast gender bias . . . 76

4.3 Probability density approximation via histograms and KDE for the features: country distance and director popularity . . . 77
4.4 Probability density approximation via histograms and KDE for the features: language popularity and popularity of the movie . . . 77
4.5 Probability density approximation via histograms and KDE for the features: production company popularity and release time of the year . . . 77
4.6 Probability density approximation via histograms and KDE for the features: release year and runtime . . . 78
4.7 Probability density approximation via histograms and KDE for the features: vote average and vote count . . . 78
4.8 A fictitious probability density approximation illustrating the idea behind the ranking of modes . . . 79
4.9 The probability proportion that can be associated to each structure (purple for A, green for B, yellow for C and grey for D) . . . 79
4.10 Correlation matrix for the genre feature, each category of the genre feature is ordered 'as seen' in the training dataset . . . 81
4.11 Correlation matrix for the genre feature, each category of the genre feature is ordered as suggested by the permutation search for similar 'groups' . . . 82
4.12 Correlation matrix for the keywords feature, each keyword is ordered as suggested by the permutation search for similar 'groups' . . . 82
4.13 Genre grouping result from metric 4.4 and linkage 4.6 . . . 85
4.14 Genre grouping result from metric 4.4 and linkage 4.7 . . . 86
4.15 Genre grouping result from metric 4.4 and linkage 4.8 . . . 87
4.16 Keywords grouping result from metric 4.4 and linkage 4.7 . . . 89
4.17 Keywords grouping result from metric 4.4 and linkage 4.8 . . . 90
4.18 Genre feature hierarchical cluster of the correlation matrix, assembly iterations. Average cluster size over the clusters formed at each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward . . . 91
4.19 Genre feature hierarchical cluster of the correlation matrix, assembly iterations. Total number of clusters present after each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward . . . 91
4.20 Genre feature hierarchical cluster of the correlation matrix, assembly iterations. Dendrogram iteration ratio using dissimilarity metric 4.4, linkage criterium Ward. The red circle indicates the local minimum which is most to the right . . . 91
4.21 Inverse elbow methodology applied to the k-modes clustering of the genre feature with two alternative methodologies for the initialisation of the position of the artefact centroids 93
4.22 Inverse elbow methodology applied to the k-modes clustering of the keywords feature with two alternative methodologies for the initialisation of the position of the artefact centroids 95

5.1 Histogram distribution of the movie's production year . . . 104
5.2 Histogram distribution of the movie's genres percentage occurrence . . . 105
5.3 Histogram distribution of the movie's popularity feature . . . 105
5.4 Hierarchical dendrogram resulting from the abs metric and the Ward linkage for the genre feature over the test dataset . . . 108
5.5 Genre feature hierarchical cluster of the correlation matrix of the test dataset, assembly iterations. Dendrogram iteration ratio using dissimilarity metric 4.4, linkage criterium Ward. The red circle indicates the local minimum which is most to the right . . . 108
5.6 Fictitious comparison of numerical stereotypes set up. Stereotypes mass, centre of mass and numbers are identified by the dots . . . 112

5.7 Mismatch ratio for the genre stereotyping representations of 1149 movies in the test dataset using Full versus Training stereotypes 116
5.8 Mismatch ratio for the keywords stereotyping representations of 1149 movies in the test dataset using Full versus Training stereotypes 117
5.9 Example of the Agresti-Coull test for an imaginary user in the example described in the text 122

6.1 Comparison between methods to treat imbalance on base model and model with stereotypes 133
6.2 Distribution of rating error - new user problem 138
6.3 Performance comparison between different algorithm implementation. Source [1] 138
6.4 Example of adding item factors to Equation 6.5 141
6.5 Half-life utility (R) new user and (L) new item cases as a function of the α decay factor (x-axis) 146
6.6 Genre diversity (number of distinct genres recommended for the model with stereotype) . 148
6.7 Comparison genre diversity for the models: stereotype and SVD with metadata 148

7.1 Correlation matrix for the complex categorical feature 'categories' in Amazon product group: 'Sport and Outdoors' 155
7.2 Correlation matrix for the complex categorical feature 'categories' in Amazon product group: 'Clothing, Shoes and Jewellery' 156
7.3 Dendrogram for the hierarchical clustering of 'categories' feature in Amazon product group: 'Sport and Outdoors' 156
7.4 Dendrogram for the hierarchical clustering of 'categories' feature in Amazon product group: 'Clothing, Shoes and Jewellery' 157
7.5 Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Average cluster size over the clusters formed at each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward. Amazon product group: 'Sports & Outdoors' . 157
7.6 Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Average cluster size over the clusters formed at each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward. Amazon product group: 'Clothing, Shoes & Jewellery' 157
7.7 Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Total number of clusters present after each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward. Amazon product group: 'Sports & Outdoors' 158
7.8 Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Total number of clusters present after each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward. Amazon product group: 'Clothing, Shoes and Jewellery' . . 158
7.9 Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Dendrogram iteration ratio using dissimilarity metric 4.4, linkage criterium Ward. The red circle indicates the local minimum which is most to the right. Amazon product group: 'Sports & Outdoors' 159
7.10 Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Dendrogram iteration ratio using dissimilarity metric 4.4, linkage criterium Ward. The red circle indicates the local minimum which is most to the right. Amazon product group: 'Clothing, Shoes & Jewelry' 159
7.11 Probability density approximation via histograms and KDE for the features: Price, Brand Popularity and log (sales rank). Amazon product group: 'Sports & Outdoors' 161
7.12 Probability density approximation via histograms and KDE for the features: Price, Brand Popularity and log (sales rank). Amazon product group: 'Clothing, Shoes and Jewellery' . 162

7.13 Histogram distribution of the price for the training set (in blue) with 371k products and the test set (in red) with 159k products. Amazon product group: 'Sports & Outdoors' . . . 164
7.14 Histogram distribution of the price for the training set (in blue) with about 1,052k products and the test set (in red) with about 450k products. Amazon product group: 'Clothing, Shoes and Jewellery' . . . 165
7.15 Histogram distribution of the 'categories' frequency in the training set (in blue) and the test set (in red). Note that only the most frequent categories are being displayed to facilitate comparison. Amazon product group: 'Sports & Outdoors' . . . 165
7.16 Histogram distribution of the 'categories' frequency in the training set (in blue) and the test set (in red). Note that only the most frequent categories are being displayed to facilitate comparison. Amazon product group: 'Clothing, Shoes and Jewelry' . . . 165
7.17 Dendrogram for the hierarchical clustering of the 'categories' feature in test dataset. Amazon product group: 'Sports & Outdoors' . . . 166
7.18 Dendrogram for the hierarchical clustering of the 'categories' feature in test dataset. Amazon product group: 'Clothing, Shoes and Jewellery' . . . 166
7.19 Dendrogram iteration ratio using dissimilarity metric 4.4, Linkage criterium Ward for the test dataset. The red circle indicates the local minimum which is most to the right. Amazon product group 'Sports & Outdoors' . . . 167
7.20 Dendrogram iteration ratio using dissimilarity metric 4.4, linkage criterium Ward for the test dataset. The red circle indicates the local minimum which is most to the right. Amazon product group 'Clothing, Shoes and Jewelry' . . . 167
7.21 Composition of rating - Amazon dataset . . . 178
7.22 Distribution of rating error, new item case, XGBoost . . . 183
7.23 Distribution of number of users by number of items reviewed for the Sports & Outdoors product group . . . 184
7.24 Product category diversity (number of distinct product categories recommended in the top-N list) for the model with stereotypes . . . 190
7.25 Comparison product category diversity for the model with stereotypes and the SVD with metadata . . . 190

Acknowledgements

I praise and thank **Allah** SWT for His greatness and for giving me the strength during my PhD journey.

Doctor Daniel Kudenko, thank you very much for being a great supervisor. Your smile has always affected me positively. It was a great pleasure to have had the opportunity to work with you. Even though you were no longer a faculty member of the University of York you were still committed to supervise and collaborate with me and I am really grateful to you for that.

I thank Dr. Tommy Yuan for taking over my supervision, for being supportive and patient, and always willing to help and cooperate. I really appreciate it.

To Dr. Nick Pears, thank you for your valuable feedback and support.

I thank my internal examiner, Dr. Dimitar Kazakov, for the meetings and conversations that were extremely vital to improve the quality of the thesis. Also, I thank my external examiner Professor Vania Dimitrova for her very helpful comments and suggestions.

I would like to thank all the Computer Science Department research administrators and IT staff who have been very helpful and always responsive to all my requests. I had a good time in York, during which I learned a lot, and I am appreciative for that.

I am really grateful to King Saud University (KSU) for the valuable scholarship which funded my studies in York.

Lastly, I would like to thank my beloved children Abdulaziz, Alhanouf and Alshihana. You all gave me strength and courage that enabled me to overcome all the difficulties we faced while abroad and for that I am eternally grateful.

Declaration

This thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree other than Doctor of Philosophy of the University of York. This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by explicit references.

I hereby give consent for my thesis, if accepted, to be made available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .....(candidate)

Date .....

Publications

1. N. A. Alrossais and D. Kudenko, “iSynchronizer: A Tool for Extracting, Integration and Analysis of MovieLens and IMDb Datasets,” in *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization - UMAP 18*, Singapore, 2018.

2. N. A. Alrossais, “Integrating Item Based Stereotypes in Recommender Systems,” in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization - UMAP 18*, Singapore, 2018.

3. N. A. Alrossais and D. Kudenko, “Evaluating Stereotype and Non-Stereotype Recommender Systems,” in Knowledge-aware and Conversational Recommender Systems Workshop KaRS at *12th ACM Conference on Recommender Systems RecSys 2018*, Vancouver, Canada, 2018, pp. 23-28.

4. N. A. Alrossais and D. Kudenko, “Generating Stereotypes Automatically For Complex Categorical Features,” in Proceedings of the Second Workshop on Knowledge-aware and Conversational Recommender Systems, co-located with *28th ACM International Conference on Information and Knowledge Management, KaRS@CIKM 2019*, Beijing, China, November 7, 2019, pp. 8-14.

5. N. A. Alrossais, D. Kudenko and T. Yuan, “Improving Cold Start Recommendations Using Item-Based Stereotypes,” in *User Modeling and User-Adapted Interaction Journal*, 2020. Accepted subject to revision, under review.

CHAPTER 1

Introduction

1.1 Background and Rationals

Recommender Systems (RS) have been developed to enhance the user experience, sales revenue and in turn the profitability for an organization. As stated by Aggarwal et al. [3], growing product sales is the fundamental purpose of a recommender system. In other words, recommender systems are employed by businesses to enhance their profit. By recommending thoughtfully chosen items to users, recommender systems draw appropriate items to the consideration of users, thus enhancing the probability of further sales. It is potentially the most critical role for a commercial recommender system, i.e., to sell a supplementary collection of items opposed to those normally sold devoid of any recommendation. This objective is accomplished by proposing items carefully selected to satisfy the user's needs and requirements [4]. According to [5], recommender systems have been reported to increase sales on online market places by over 35% for Amazon.com and more than 60% for Netflix.

Despite the benefits of RSs, Linden et al. [6] in their paper related to Amazon.com recommender system stated that e-commerce recommendation algorithms frequently function in a challenging environment, such as:

- Several applications require the results set to be returned in real-time, in no more than half a second, while still providing high-quality recommendations.
- Customer data is volatile which means that every interaction contributes valuable customer data, and the algorithm must respond instantly to new information.

- RSs usually have insufficient data of new customers (new user problem), which is created by a few purchases or product ratings while the older customers can have an overabundance of information based on thousands of purchases and ratings.

- In addition to the new user problem, also the new item problem or early-rater problem occurring when an item has not been rated yet by any of the members, of the user community, is considered a challenging problem to recommender algorithms.

The growing importance of RSs has motivated the research community to invent diverse techniques for the development of recommender systems to solve the above mentioned problems. A promising approach for improving recommendations and solve the new user and the new item problems is stereotype-based modeling. Rich [2] was the first to propose the utilisation of stereotypes in user modeling and recommender systems. A stereotype depicts a collection of attributes that are relevant for a collection of users [7], for example based on demographic information. User-based stereotyping was engaged by Rich [2] as a method for addressing the new user problem, with the objective that recommendations could be presented to a new user without the requirement to gather a set of ratings from the users for the purpose of user model training.

In the stereotype approach, users are grouped instead of each being treated individually. This approach has the advantage of larger sets of training data available for a group of users in comparison to a single user. In e-commerce, for example, for a single user without a history of transactions, construction of any model is not possible. However, if it can determined that the user shares some meaningful features with one of the predefined groups - then one can use a model based on the existing preferences of the group to generate recommendations. Thus, stereotypes can be built on the idea that users with similar features may also share similar broad-level preferences and that items with similar features may be preferred by certain types of users. Therefore, a stereotype can be viewed as an aggregation of the characteristics of the items or users that allow one to group items and users in general classes.

Historically, in the pioneering works on stereotyping, the majority of the stereotype classes were built manually by the operator with knowledge of the problem and data [2, 8, 9, 10, 11, 12]. This approach has obvious limitations: firstly, the operator building classes manually may miss or disregard important relationships (features) that effectively classify and define a stereotype, just because the operator intuition did not consider such possible dependencies between features; secondly, the ad-hoc construction of stereotypes is inefficient and unable to cope with a general dynamic evolution of the features, for example relationships that were not important in the past become important after the stereotypes have already been deployed; thirdly, inconsistencies among stereotype classes often arise especially as their representation evolve as discussed in [13], such inconsistencies are larger if the stereotype building problem is not addressed in a systematic manner especially in online platforms.

Online platforms are often judged based on user experience, and recommendation quality is

one of the keys for improving or degrading the experience. Cold start is when a new user first approaches the platform or a novel item is launched. Improving the recommendations during that phase is key to customer (i.e. user and item provider) retention. It would be unfeasible, and likely error-prone, to rely on expert human knowledge to correctly classify a new user or a new item to the platform.

While the majority of the literature focuses on user-based stereotypes to solve the new user problem, it is important to note that this work focuses on item-based stereotypes to solve the new user and the new item problems. Cold start phase is defined as the situation in which the RS needs to cope with a new user first approaching the platform or a novel item being launched. This project investigates the possibility of obtaining a viable recommender system that uses stereotypes generated directly via item's metadata similarities, *without* using ratings and preferences information.

This work examines how stereotypes can be built *automatically*, and most importantly in a way that is *independent of user's preferences*. The work also aims to demonstrate how stereotypes can effectively improve recommendations during the cold start phase.

To the author's knowledge there are no works in the literature that attempt to group the items and/or users based on their primitive metadata, delineating similarities that are independent of the preferences expressed by users toward items. The body of the existing research has focused dis proportionally on consumption/preference driven grouping of the users/items. Only after such common preferences were identified, machine learning methods are used to link groups of preferences directly to the metadata features. We see a gap in such a procedure: creating clusters of items and/or users that are independent of past user-to-item preferences should in principle allow a machine learning driven algorithm to discover patterns that are not directly observable in the preferences when regressed back on the metadata. The difference between the two approaches may seem small, but given the non-linearity of these problems, it will be demonstrated how it leads to important patterns not discoverable in the case of explaining the preference groups with the metadata features.

1.2 Problem Statement

The new user and new item problems remain important challenges for any recommender system. With the rapid growth of the number of users and items, the majority of the recommender systems using collaborative filtering techniques suffer from problems like data sparsity and scalability. In the literature related to the application of clustering methods to solve the cold start problem in RS, the clustering methods are applied directly to the ratings (for example clustering the rating matrix, as in [14]), or indirectly via the preferences of users (for example clustering groups of users based on tastes, as in [15] as a way to address sparsity and as a way to generalise preferences).

Modern datasets have a range of features that often fall outside the established clustering

methods for generating stereotypes. The vastity and complexity of the datasets also calls for fully automated procedures, rather than the manually/operator assisted stereotype creation found in the preliminary literature on stereotypes. A feature category that is often found in modern datasets, and that will be formally introduced in Chapter 3, is that of complex categorical features; these are multi-label categorical features, where the number of labels is not strictly defined. For this type of features their labels often has a lexical meaning, that describe one or more characteristics the item (user). These can be for example keywords attached to an item. We recognise that such features play a crucial role in user and item pairing, and they are not handled by the existing stereotype construction methodologies.

The novelty of this research is to show that there are other dimensions in the problem of obtaining recommendation improvements in cold starts, and these dimensions *do not* rely on the study of the rating matrix. The work analyses a different aspect, namely the possibility of obtaining a viable RS that operates on stereotypes of users and items, that are generated directly via their features similarities, other than ratings and preferences information. In other words, we first propose and test systematic ways to produce stereotypes that are rating and preferences 'agnostic'; ultimately, they allow us to reduce the dimensionality of the problem. The stereotypes provide sufficient flexibility to be used later on in the development of a recommender system to describe preferences traits in a population of users, e.g. males in their 20s likes action and comedy movies.

Ratings and preferences are introduced after the stereotypes have been built to train a recommender system that is operated in a reduced space of user and item classes. This approach in addition to reducing the sparsity of the rating matrix- *reduction that is driven by a process that is independent of the rating matrix itself* - generates a framework that is able to handle seamlessly the treatment of new user and new item without the need of any ratings, as long as the basic item and user descriptions are available to allow the system to associate the user and item with a stereotype combination.

Our approach leads to an hybrid recommender system using a combination between content-based recommendation, i.e. it suggests similar items based on item's metadata - stereotyped, and demographic collaborative filtering, i.e. it relies on statistical preferences given by users that are similar to the user considered from the point of view of user's information. As the number of ratings expressed by the new user, or the number of ratings about the new item grows, the stereotypical system can transition to a classical RS. Due to the fact that a stereotypical approach can only generate very general type of recommendations, it should only be employed during cold start phases, where none or few ratings have been expressed by the new user or about the new item (see for example [16]). An alternative use of stereotypes can be to provide an element of novelty in recommendations once overspecialisation is detected.

1.3 Research Questions and Hypothesis

In the context of providing the users of a platform with recommendations about interesting items, this work will demonstrate how the use of rating and preference independent item-based stereotypes lead to a substantial dimensionality reduction that has the dual benefit of improving computational efficiency and also increasing recommendation quality during cold-start phases.

The project seeks to answer the following research questions, with the designated research hypotheses.

1. Can item-based stereotypes, not based on rating, be constructed automatically?
    - H1. Item-based stereotypes, not based on rating, can be constructed automatically.
2. Can automatically constructed item-based stereotypes improve recommendations during the cold-start phases?
    - H2. Automatic item-based stereotypes have positive result in improving recommendation over no stereotypes during the cold-start phases.
3. How do stereotype-based item models compare to other state-of-the-art recommender systems models in terms of predictive performance during the cold-start phases?
    - H3. Stereotype-based item models will result in a better predictive performance when compared to other state-of-the-art recommender system models during the cold-start phases.
4. Apart from recommendation accuracy, what are the other benefits of item-based stereotypes?
    - H4. Item-based stereotypes add other benefits to recommender systems apart from accuracy like serendipity, system performance measured in run time and hit rate.

A number of experiments will be conducted in the subsequent chapters to test the hypotheses. Hypothesis H1 will be tested in Chapters 4 and 5. Hypothesis H2 will be tested in Chapters 6 and 7 while the same chapters will accept hypothesis H3 and H4.

1.4 Structure of the Thesis

The rest of the thesis is organised as follows:

- The technical introduction to recommender systems and related issues are discussed in Chapter 2. Chapter 2 also contains a review of the literature's in stereotyping and its root in solving cold-start problems.

- Chapter 3 starts by looking into feature identification and selection in order to propose a novel approach for automatic stereotype construction. The chapter then documents clustering-based algorithms for numerical features applied to a simplified dataset.
- Chapter 4 documents the investigation of hierarchical clustering applied to the correlation matrix for complex categorical features as a method to discover stereotypes automatically. The same chapter proposes a method for the automatic construction of stereotypes for numerical features.
- Chapter 5 provides a preliminary evaluation of stereotypes using three different tests to ensure that the generated stereotypes are stable prior to testing them in recommendation context.
- Cold start into recommendation performance are addressed in Chapter 6. The comparison with the state-of-the-art factorisation technique (SVD) against the proposed approach to benchmark the findings is discussed in the same chapter.
- The generalisation of the methodology on a second dataset from the retail sector is tackled in Chapter 7.
- The thesis is concluded in Chapter 8. This chapter also summarises the results of the project and discusses future work.

CHAPTER 2

Literature Review

This chapter describes the key concepts from the existing literature that considered crucial for the current research. It starts with a general overview of the recommender systems field including RS models and their evaluation and it progresses with the stereotypes-based modeling.

2.1 Recommender System in e-commerce

In the present times, personalisation of product information has become a crucial aspect that influences a customer's product selection and satisfaction. Personalised service demands organisations to comprehend customers and offer goods or services that satisfy their needs. Successful organisations are those which provide the right products to the right customers at the right time and for the right price. Since e-commerce websites started to grow, a compelling need arose for rendering recommendations obtained from filtering the complete array of choices that were available. Users found it challenging to reach the most suitable options from the enormous range of items that such websites were providing. The stupendous growth and diversity of information present on the internet and the quick introduction of novel e-business services such as purchasing products, comparison of products and auction regularly confounded users, causing them to make bad decisions [17]. With the expanding of the new level of customisation, organisations raise the volume of information that users are required to process before choosing the items that meet their requirements. To resolve this information overload problem, one of the solutions is the employment of recommender systems [18].

In the recent years, RSs have proved to be important means of coping with the problem of information overload. A recommender system responds to this phenomenon by steering a user

towards items which are relevant to the current task of the user. Based on a user's request, which can be expressed, grounded on the recommendation approach, context and need of the user, RS produce recommendations utilising different types of data and knowledge related to the users. RSs save past transactions in customised databases to allow the user to access the recommendations later.

The user may accept or reject them and give either implicit or explicit feedback instantly or at a subsequent stage. All such actions and user's feedbacks may be saved in the recommender database and could be employed for making new recommendations in subsequent user-system interactions [17]. Hence, recommender systems automate personalisation on the e-commerce sites by fostering personalisation for each individual customer. As Jeff Bezos, the CEO of e-commerce giant Amazon.com, said "If I have 2 million customers on the Web, I should have 2 million stores on the Web" [19, p. 158].

RSs can be defined as the software tools and techniques which provide recommendations for items to be of value to a user [20, 21, 22]. Recommender systems are employed by e-commerce websites to recommend products to their customers [19]. Almost every major company has applied them in one form or the other, for example Amazon uses it to recommend products to its customers, Youtube uses it to determine which video to play next to the autoplay and Facebook uses it to recommend posts that you like and people to follow.

The products may be recommended on the basis of the best overall sellers on a site, the customer's demographics characteristics, or an evaluation of the customer's past purchase behavior as a forecast for anticipated purchase behavior. In a broader perspective, such techniques are a component of personalisation on a website, as they support the website to adjust itself to an individual customer.

E-commerce recommendation algorithms frequently function in a challenging environment to enhance their profit. In order to accomplish this large business-centric objective of raising revenue, the general operational and technical objectives of recommender systems must be achieved as follow:

- Relevance - One of the most apparent operational objectives of a recommender system is to recommend items that are relevant to the customer's needs. Users have a higher probability to consume items that are appealing to them. Even though relevance is the fundamental operational objective of a recommender system, it may not be sufficient to use it in isolation [3].

- Novelty - Recommender systems are more effective when the suggested item is something that has not been seen by the user previously [23].

- Serendipity - It indicates the ability of a RS to surprise the user, the items recommended are slightly unexpected, and hence there is a reasonable component of lucky discovery, in contrast to evident recommendations [18]. Serendipity is distinctive from novelty in that

the recommendations are unexpected for the user, rather than merely something which the users were not aware about in the past [24].

- Enhancing recommendation diversity - Recommender systems generally recommend a top-k item list. When the recommended list comprises items of distinct types, there is a higher probability that the user may approve at a minimum one such item. Item diversity has the advantage of guaranteeing that the user is not wearied by recurring recommendation of comparable items [3, 25].

RSs are information processing systems which dynamically collect different types of data to generate their recommendations. The collected data is basically about the items to recommend and the users who will obtain such recommendations. Yet, as the available sources of knowledge and data for recommender systems can be quite assorted, their utilisation relies on the recommendation techniques [17]. RSs often need to access the following content: a) information describing the users, b) information describing the content (i.e. items), c) implicit information about users selecting certain items and d) explicit information about users rating and reviewing certain items.

2.2 Popular Recommender Systems

To provide an understanding of the state-of-the-art RSs, this section discusses a few popular RSs as reported by Aggarwal et al. [3].

GroupLens Recommender System. It was a ground-breaking recommender system that was developed as a research prototype for recommending Usenet news in 1994. The recommender system gathered ratings from Usenet readers and employed them to forecast if the readers would like an article prior to reading it. Few of the initial automated algorithms for collaborative filtering (CF) were created in the GroupLens setting [26].

Amazon.com Recommender System. It is one of the earliest recommender systems developed in the commercial context. Amazon.com presented recommendations based on explicit user ratings, purchase information and customer's browsing behavior. Amazon's ratings are defined on a 5-point scale, where 1-star is the lowest rating and 5-star is the highest rating. The customer-specific data concerning purchase and browsing can be simply obtained when customers are logged in with a mechanism of account authentication offered by Amazon. Linden et al. [6] reported that Amazon's use of item-to-item CF technique had a huge successful impact on the business through increased click-through and conversion rates.

Netflix Movie Recommender System. Established as a mail-order rental company for digital video disc (DVD) of movies and television shows, Netflix ultimately extended to streaming delivery. Netflix offers users the facility to give ratings to the movies as well as television programs on a 5-point scale. Moreover, the user actions concerning viewing different items are also saved by Netflix. Such ratings and actions are later employed by the company to generate recom-

mendations. Moreover, Netflix presents explanations for the recommended items. It explicitly provides instances of recommendations relying on particular items that were viewed by the user. Such an approach can help develop customer loyalty as well as retention. Netflix reports that at least 75% of its downloads come from their RS [27]. Netflix increased the awareness of research community toward incorporating machine learning techniques into commercial RSs when they launched prize competition in 2006 [28]. The competition attracted more than 5000 teams for the one million dollar award to the first algorithm that outperforms Netflix's in-house RS by 10% of accuracy.

Google News Personalisation System. It is capable of recommending news to users based on their browsing history. The clicks are linked to particular users via their Gmail accounts. In such a context, news articles are used as items. The action of clicks by a user on a news article can be seen as a positive rating for such an article. These ratings can be seen as unary where a method is present for a user to convey their liking for an item, but no method is available for users to display their dislike. Collaborative recommendation algorithms are implemented to the gathered ratings so that inferences can be made about the personalised articles for specific users.

Facebook Friend Recommendations. Facebook is an instance of a social networking website. Their recommendations have slightly varying goals than a product recommendation. Facebook friend recommendations lead to a growth in the number of social connections which enhances the experience of a user and promotes the growth of the social network. The recommendation of potential friends (or links) permits better connectivity and growth of the network on which the advertisement revenue depends.

2.3 Existing Recommender System Approaches

According to [7], many recommendation strategies have been created and utilised in diverse information domains. Many hybrid approaches have been established to take the advantages of the various approaches. The five key recommendation strategies in general use are collaborative, content-based, demographic, hybrid and pairwise preference learning. These are discussed in turn below.

2.3.1 Collaborative Filtering (CF)

One of the more successful underlying ideas driving the range of techniques behind user-based collaborative filtering (CF) is to establish preference patterns exhibited by users, for example via a domain of similar users (i.e. user neighbours), where similarity among users is defined via known past preferences and a range of possible metrics. Item-based CF methods [29, 6, 30] were proposed later on, which compute predicted ratings as a function of the ratings of the same user on similar items (i.e. item neighbours).

CF approach is possibly the most widely utilised and established, as verified by its deploy-

ment on e-commerce websites like Amazon.com [6]. The term was first coined by Goldberg et al. [31] and a large body of research was conducted in the late 90s and early 2000s to investigate such systems, see for instance [31, 32, 33, 34, 35, 36, 37, 38, 39] and references therein.

From those years on, thanks to the large range of algorithms developed in machine learning (ML), a wealth of research focused on the application of classification and grouping methodologies to CF, see for example approaches based on bayesian networks [36], latent semantic analysis [40], maximum entropy [41], support vector machines [42], singular value decomposition (SVD) [43, 44], SVD with neural net classification [38] and inductive rule learning [39]. These techniques are prone to become computationally expensive when the number of users increases, as the number of similarity pairs between users grow too large, and also the matrix of all possible user-to-item pairs becomes extremely large and sparse. Goldberg et al. [37] provides a framework to reduce the dimensionality by performing the principal component analysis (PCA) and clustering of the rating matrix.

However, the major shortcomings of CF consist in the fact that the users for which a recommendation needs to be performed must have a past sufficiently large stencil of implicit or explicit rating preferences recorded. The similarity metrics, like correlation or cosine distances, appear to be too noisy when too little past preferences are known about a user or an item. This is the case in the 'new user' and 'new item' problems, which can arise when a new user first approaches the platform or a novel item is launched and there is no implicit or explicit preference history expressed by the new user (or expressed toward the new item). The CF approach also encounters the latency problem, which is the time between the release of a new item and its first appearance within a recommendation list [45, 46]. This problem is especially apparent in the e-commerce domain where item catalogue is constantly updated. The reduction of item latency directly affects the revenue of e-commerce systems.

While user-based methods were initially considerably popular, they have also been proven not easily scalable and sometimes inaccurate. Item-based CF methods [29, 6, 30] are less affected by these drawbacks. Another advantage of item-based methods is that they can be easily used to justify a recommendation. Hence, the list of neighbour items used in the prediction, as well as their similarity weights, can be provided to the user as a clarification of the recommendation. In user-based methods, however, the active user does not recognise the other users who serve as neighbours in the recommendation [17]. Justifiability is continually highlighted in major recommender system conferences and journals [47, 48]. It claims to increase the possibility of an item to be clicked as well as increasing user loyalty and trust which are important features in a competitive e-commerce domain. The rising interest to have explainable and transparent RS shed the light into using item properties and metadata as well as user-generated content like textual reviews and tags.

Nonetheless, the item-based CF technique has been shown to be more effective in terms of the prediction accuracy than the user-based CF technique [30, 3]. Despite its efficiency, the

item-based collaborative filtering does not perform well and may generate inaccurate recommendations to users due to two key impediments namely: the sparsity and the new item problem [30, 3]. To resolve these problems, contemporary recommender systems have focused on the integration of supplementary information, hence, allowing RS to exploit the additional information to compensate the insufficient users' ratings to produce more accurate recommendations. Illustrations of such additional information are the semantic relationships that exist among users or items [49, 50, 51, 52, 53]; and the multi-criteria ratings which can imply more complex users' preferences [3, 54, 55, 56].

The approach developed by Sun et al. [57] for presenting recommendations based on group-to-item associations has been confirmed to give better results than traditional item-to-item associations. This approach seems quite effective and highly relevant for our area of research which deals with creating item groups based on stereotypes for recommendations. The idea of implicit relation between items is discussed for the first time in this study. Sun et al. [57] introduced a novel matrix factorisation (MF) model by exploiting association rule-based implicit item relationships (IIR). It is produced solely based on user-to-item rating information and it does not use additional user or item side information. The authors employed an adapted associate rule technique to reveal the implicit item relationships as item-to-item and group-to-item associations, which are then utilised to regularise the creation of low-rank user and item feature matrices in the suggested IIR model. Moreover, they design four distinct strategies to pick the most reliable item associations to train the model. MF techniques have been extensively applied in recommender systems. Sun et al. [57] defined the implicit item relationships as the item associations between a target item and another item or a set of other items. The association rule mining is to search the associated item pairs that often co-occur in transaction events and it usually produces high reliable result.

2.3.2 Content-Based Filtering (CBF)

An alternative approach, that builds on the poor scalability of user-based CF, is that of content-based filtering (CBF), given its roots in the field of document classification and retrieval. While user-based CF methods rely on the opinion of similarly-minded users to predict a rating, CBF approaches look at ratings given to similar items where the concepts of similarity are applied between items features [6, 29, 58]. The resulting recommender systems have proved more efficient and suitable to offline precomputation [17]. The content analyser of a CBF system is the part that extracts features from the items descriptions and properties and stores them in a structured item representation [59]. Content-based filtering approaches consider the past preferences of an individual user to discover preference models on a feature-based representation of the content.

The content-based systems have their limitations such as the content-based recommenda-

tion techniques comprise of a natural limit on the type and number of features that are linked, whether manually or automatically, with the items that they recommend. Domain knowledge is generally essential. Moreover, content-based recommenders have no internal method for discovering unanticipated items. This shortcoming is termed as the serendipity problem to emphasize on the propensity of the content-based systems to generate recommendations with a limited novelty (i.e. overspecialisation). Many recent research [60, 61, 27] recognise the importance of novelty in modern recommendation setting. Lastly, ample ratings have to be gathered before a content-based recommender system can actually recognise user tastes and present precise recommendations which are not possible when there are few ratings [17]. A wealth of research has been dedicated to the application of semantic techniques to extract features from descriptions, for example see [62, 63, 64] and references therein among others.

2.3.3 Demographic

Demographic recommender systems suggest items based on the demographic profile of the user [17]. The underlying assumption is that distinctive recommendations should be made for separate demographic niches. Several websites implement uncomplicated and efficient personalisation solutions grounded on demographics. For instance, users are forwarded to specific websites depending on their language or country. Or recommendations may be personalised based on the user's age. According to [7], the demographic recommender systems utilise additional user information such as the gender, age or occupation apart from their ratings. Demographic recommendation depends on the assumption that users having common demographic attributes have similar needs or interests. Examples of the use of demographic information in RS can be seen from [65, 66].

2.3.4 Hybrid

To achieve improved results, certain recommender systems blend techniques for content-based approaches and collaborative filtering approaches [67, 46]. Broadly, all the knowledge available from diverse sources of data must be utilised and the algorithmic power of different recommender systems needs to be applied to generate inferences that are robust. Hybrid recommender systems have been developed to investigate such routes [3]. By the application of hybrid approaches, few restrictions and issues of standalone recommender systems such as the cold-start problem can be avoided [68, 43, 69]. The mixture of approaches can be implemented in several ways: 1) the independent implementation of algorithms and recombination of the results, 2) utilising certain rules of content-based filtering in the collaborative filtering approach and 3) utilising certain rules of collaborative-filtering approach in content-based approach and building a integrated recommender system which combines both discussed approaches [67]. More approaches to build hybrid systems are proposed by Burke [70]. The work presented in [1, 43, 71, 68] demonstrate successful examples of hybrid recommender systems.

2.3.5 Pairwise Preference Learning

A novel area of research consists in utilising ontologies for potential improvement of RSs are discussed in [72, 73]. For example [74] proposes to use inductive logic programming (ILP) for preference learning (PL) and description logic (DL) learning and apply them in the context of recommendations. DL represents a problem in concepts, objects and roles. When ILP is used to learn general concepts, the system needs to be presented with examples on which such ontologies can be discovered. The difficulty of adopting such novel strategy in the field of RS is due to inconsistency in the logics that can be learned using ratings, for the very nature of ratings: users are often inconsistent in providing ratings. For this reason, ontologies driven RS have only been applicable to a learning strategy driven by pairwise comparisons, see for example [75], [76] and [74]. While such approach may yield in the near future positive developments in standard non cold start recommendations, it is perhaps too early to find application in cold start scenarios, also given the difficulties in learning and obtaining consistent binary pairwise preferences to use in the learning phase.

2.4 Cold-Start Problem for Recommender Systems

While the majority of the literature focuses on user CF and CBF as well as hybrid approaches of the two, the review work by Jannach et al. [77] has shown that less than 5% of the existing research addresses the new user and new item problems (see Schein et al. [78] for one example).

Elahi et al. [79] provide a comprehensive review of the recent developments in addressing the cold-start problems. Historically, cold-start phases have been addressed by implementing hybrid recommendation techniques, combining collaborative and content-based filtering [80, 68, 22, 81, 82, 43, 83]. Deshpande and Karypis [29] argue that the new user and new item problems can be related to the sparsity of the rating matrices. The work in [77, 84] proposed a graph technique to address the sparsity of the rating matrix to handle the cold-start problems.

Recently, the cold-start problem has been an active research subject, with several works addressing techniques tailored to handle either the new user or the new item problem [85, 43, 86, 87]. For example, Fernandez-Tobias et al. [69] propose to solve the new user problem via latent rating patterns discovered using item metadata in a factorisation-based method. Deldjoo et al. [88] focus instead on the new item problem, using innovative audio and video metadata in the movie recommendation domain. The emerging pattern is characterised by heavier use of the available metadata context of both items and users, coupled often with factorisation-based methods. The general findings suggest that during extreme cold starts it is difficult for any of the researched systems to significantly improve over basic baseline models. Moving away from pure cold start, the researched models improve over the baseline once the first few ratings have been collected.

Other works suggest extracting information about the new user from social media - for ex-

ample, see [89], [90] and [91] - or linking across domains - for example, see [92], [69] and [87] - by using the knowledge of ratings and tags assigned by the users to items in an auxiliary domain (e.g. movie ratings) to model preferences in a target domain (e.g. book purchases). Such techniques reveal user social data and might cause privacy concerns.

Fernandez-Tobias et al. [93] proposed three strategies of user personality information and applied them to CF to solve the new user problem, while [94] and [95] incorporated feature-based preferences between items to alleviate the cold-start problem.

Special approaches for handling the new user and new item problems consist of requiring a first compulsory training period of the RS on every new user and new item before performing recommendations, see [96, 6]. Such works demonstrate the inherent difficulties of handling extreme cold starts; they usually improve recommendations over simpler baseline models once the users and items become 'known' via a series of directly expressed preferences or, as Nasery et al. [94] suggests, as indirect preferences expressed to features.

A range of techniques that increase efficiency by reducing the cardinality and sparsity of the rating and consumption matrix include those built upon the idea of factorisation of the user-to-item rating matrix [97, 43, 98, 99]. These techniques, among which the singular value decomposition (SVD) is probably the most popular thanks to the success obtained in the Netflix grand prize [100, 101], aim to reduce the dimensionality of the rating matrix by projecting the ratings over a latent factor space. This process enables researchers to determine how users rate items. More details on SVD are discussed in Chapter 6. Most of the studies referenced in this work, when reaching the prediction stage, rely on factorisation techniques to reduce the dimensionality of the user-to-item matrix or to provide a latent space where clustering methods are applied, for example, see [98]

In addition to the above methods, classes such as stereotypes can also be used as a tool to generate recommendations for users in the cold-start scenario. A range of studies [10, 102, 9] followed the ideas of user-based stereotyping presented by Rich [2]. Rich [2] was the first to recognise and propose that users could be 'stereotyped' in the absence of past preferences for a user, assuming that his/her preferences would be in line to those of the stereotypical group to which the user belonged. For example, it is easy to argue that a female user in her 20s located in the US will exhibit on average different preference traits than those of an average man in his 60s located in India. The issue is whether common preference traits based on feature characteristics like gender, age, location and education are strong enough to provide recommendations.

Up until the late 90s, the construction of stereotypes had been almost exclusively manual and driven by expert knowledge. The work of [103] provides one of the first attempts to 'learn' the user and item classes via supervised learning techniques. Grouping of features, or clustering, was soon introduced as a way to address the sparsity of rating matrices, especially in the context of classifiers and probabilistic-based systems [104, 105, 106]. A wealth of research has focused on the application of classification and grouping methodologies to CF and CBF for clustering - see

[107] - and for forests of trees - see [108]. However, this research does not address the cold-start phases.

Adomavicius and Tuzhilin [80] and Braunhofer et al. [97] attempted to apply grouping methodologies to the cold-start phase and, in particular, to the new user case. In the extreme cold-start scenario the system may recommend popular items or items with the highest average ratings, as discussed by [93].

As reviewed in the present work, extreme cold start is a relatively poorly researched area of RS compared to specialised recommendations for well-known users and items. Only in recent years this problem has received a wider attention, despite the importance that high quality recommendations during cold start have in attracting and retaining new users and new item (content) providers. A RS operating outside the cold start realm can rely on several well researched techniques applicable directly to the user-to-items preference matrix in order to produce meaningful collaborative filtering like recommendations. In the cold start situations, it is now fully acknowledged that user's and item's metadata provide invaluable extra information to improve recommendations to a new user or of a new unrated item. The main gap that we see in the literature is the absence of a comprehensive treatment of users and items metadata, that could be seen as "independent" of the recommendation system technique used. For example, the singular value decomposition methods (SVD) have been expanded and generalised (Factorisation Machines) to embed user and item metadata. However, the metadata treatment is a constitutional part of the solver and could not be generalised to other approaches. This work addresses the use of metadata, via stereotypes, creating a unified approach for multiple types of user and item metadata features as discussed in the following section(s). We aim at providing a construct for the treatment of metadata that can be reused in different types of RS . The next section discusses how stereotypes are defined in the literature and how this works expands on their definition to use them for extreme cold start recommendations.

2.5 Stereotype-Based Modeling

2.5.1 Stereotype - Definition and Evolution

Rich was the first to propose the utilisation of stereotypes in user modeling and recommender systems in her Ph.D. thesis published in 1979 [109]. Stereotyping was adopted by Rich as a method to resolve the new user problem. The aim is to make recommendations to new user without the need of gathering a set of user ratings for the purpose of user model training. Rich [2] mentioned that an additional benefit of stereotyping is its space-efficiency as the characteristics that are applicable to several users are required to be stored only once, but they can be employed by all members belonging to a stereotype when necessary.

Rich addressed the problems when systems are supposed to assume their users as individuals with distinctive goals and personalities. She outlined the challenges and later introduced stereo-

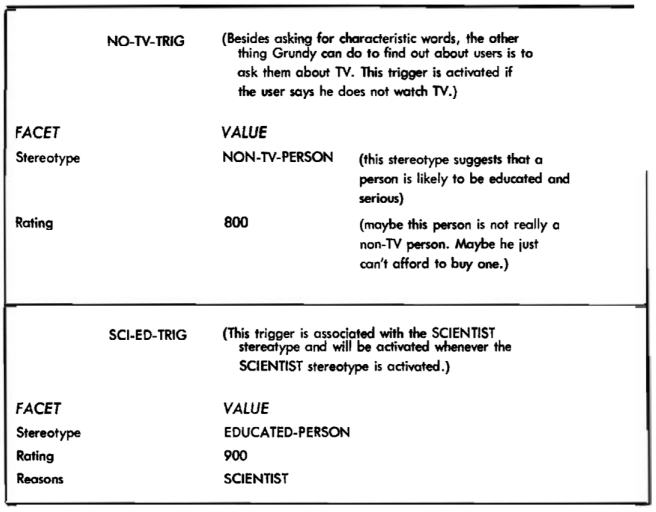
types as a valuable method for developing models of distinct users with limited user information. In order to create user models quickly, a vast amount of vague knowledge must be included in the user models [2].

In user modeling, the idea of a stereotype robustly associates to its definition in English - a body of default information about a set of people. In the study by Rich [109, p. 44], stereotypes depict "the collection of frequently occuring characteristics of users". It is crucial to note that, a stereotype may or may not be a precise representation of the user group or any specific group member, but it may simply be an estimation of certain characteristics of the group [7]. It is not essentially a set of specifics which are valid for all members of the group. The fundamental motivation for applying stereotyping is to present personalisation with inadequate information from new user via the recognition of the assigned stereotypes.

To exhibit and appraise stereotyping, Rich [2] created GRUNDY to suggest novels to users on the basis of socio-demographic characteristics of users. The enhanced utilisation of machines which are controlled by software to mediate between the interactions amid users and systems has rendered it likely for such interactions to have more flexibility and are more personalised with the suitable customised user models [110]. Therefore, numerous system developers have recognised the individuality of users and user groups and have developed distinct user or group models with the aim to account for dissimilarities [7].

A stereotype has two indispensable parts: a) the designation of the stereotype in terms of the characteristics similar to members or the features of typical users, b) the pre-conditions are the triggers that must hold to allocate the stereotype to a user [2, 7]. The first constituent is called the body or traits of the stereotype [109]. The body comprises of the characteristics that are shared by the members of the stereotype. In regard to the RS, such characteristics are generally the attribute items of shared interest for the stereotype members. The body, hence, can be utilised to categorise items for recommendation to stereotype members. The representation of the body relies on the application [7]. The second constituent of a stereotype is generally termed as the trigger and it signifies the conditions or events that are required to be held true when allocating the stereotype to a user. Such conditions are generally propositions on demographic attributes of the user. In the case of the activation of a stereotype by a user, the body can be utilised to forecast his characteristics. Moreover, the trigger can be explicitly depicted in diverse ways similar to the body [7].

As stated by Rich [2], for a system to effectively utilise stereotypes, it should possess two types of information. Firstly, it should have the information on stereotypes themselves e.g. the collections of traits, and secondly, it must have information related to a collection of triggers which are those events whose incidence signals the suitability of specific stereotypes. For example, if a user instantaneously utilises a sophisticated construct in a system, that is required to act as a trigger for the 'expert-user' stereotype, which should then be activated. During the activation of a stereotype, the predictions that the system makes about diverse features must be



NO-TV-TRIG (Besides asking for characteristic words, the other thing Grundy can do to find out about users is to ask them about TV. This trigger is activated if the user says he does not watch TV.)

FACET	VALUE	
Stereotype	NON-TV-PERSON	(this stereotype suggests that a person is likely to be educated and serious)
Rating	800	(maybe this person is not really a non-TV person. Maybe he just can't afford to buy one.)

SCI-ED-TRIG (This trigger is associated with the SCIENTIST stereotype and will be activated whenever the SCIENTIST stereotype is activated.)

FACET	VALUE
Stereotype	EDUCATED-PERSON
Rating	900
Reasons	SCIENTIST

Figure 2.1: Sample triggers by Rich [2]

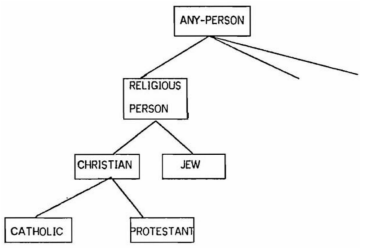


Figure 2.2: Stereotype hierarchy as developed in GRUNDY by Rich [2]

incorporated into the user model. An example of triggers used in The GRUNDY system by Rich [2] is depicted in Figure 2.1.

Stereotypes, when coupled with the capability of documenting explicit statements by the user and deducting direct inferences related to a user from his behavior, have the possibility to present a mechanism for developing computer systems that can respond in a per user way [2]. A number of stereotype-based systems [109, 111] include one more component to describe hierarchical relations among stereotypes. Definite stereotypes at the leaves of a hierarchy may inherit characteristics from the more common stereotypes above. This infers that additional space-efficiency can be achieved. Figure 2.2 depicts a sample of stereotype hierarchy as developed in GRUNDY.

Comparison with communities. As pointed by Paliouras et al. [103], an additional user modeling for grouping users based on their characteristics is to group them as per their shared interests in a community. This has correlation to the concept of a neighbourhood in CF recommender systems although an explicit model of the neighbourhood users' common interests is developed for a community [7]. Given the divergent descriptions of stereotypes and communities, they tend to

be derived in dissimilar manners. Generally, stereotypes are developed by hand [103], but once established they can provide a fast method for modeling and serving new user [7].

Due to this reason, they are highly dependent on the developer and might not be appropriate for all of the user groups as described by [102, p. 3]: "this is part of the statistical nature of stereotypes: for some users they are inaccurate, but their power is that they are useful for many users". In contrast, communities are well modeled with unsupervised learning and clustering [103]. Hence they are data-driven and tend to more precisely exhibit the users' common interests. Similar to stereotypes, communities can be optimise storage characteristics. Membership of a user to a community is computed by the interests expressed by user and their alikeness to the interests of members of the community. Stereotype membership is defined by the expressed user characteristics. In the case of CF recommendation approaches, a community or set of communities cannot be allocated to a new user without getting relevance feedback from the user, therefore community modeling is incapable to solve the new user problem [7].

As explained by Rich [2], as it will be rarely plausible to know accurately before a system is being utilised, the kinds of users it will have, what they will be attempting to achieve, or how they will deal with the system, it is essential that it is possible to modify old stereotypes and generate new ones after the system has been operating for a while. Although it is important to have the before-mentioned facility for the changes in the system, it is prudent to start out with stereotypes that are as reasonable as viable, so that the opening reception of the system will be positive. The latter can be promoted via protocols of users communicating with a simulated system. Such protocols can also provide insights on the other major problem, e.g. the building of the triggers [2].

Adapting Stereotypes. Apart from learning models of individual users, it is essential for a user modeling system to have the capability to transform its stereotypes database. This significance results from the lack of accurate data on which to base the initial development of the stereotypes. Moreover, computers have a notable lead over humans concerning the application of stereotypes as they are not devoted to them emotionally and hence can change them as necessary by experience. This characteristic is undoubtedly beneficial [2].

The majority of the events that happen when a stereotype is activated can provide insight into the accuracy of the stereotype and its triggers. If the user acts in a manner forecasted by the stereotype, it provides evidence to the suitability of that prediction as well as to the aptness of the triggers that prompted the activation of the stereotype. In another scenario, if the user exhibits a behavior that contradicts with a forecast of an active stereotype, then it can be inferred that either the prediction is unsuitable, or the triggers that made the stereotype to be activated are inapt (or probably both). It should be noted that this does not necessarily infer that the stereotype or the triggers are incorrect. They may accurately forecast most of the times while the current user is simply an exception. A single divergence will not cause the system to question its triggers and stereotypes, only as multiple divergences are being identified then the system will begin

un-reinforcing triggers and the stereotype by decreasing their ratings. If, in fact, some of the inferences are accurate, they will be reinforced in different scenarios and hence will be stored by the system. Incorrect inferences will ultimately be eliminated. It is due to this deceptive un-reinforcement that may occur that it is predominantly essential that reinforcement happens whenever a prediction is borne out by experience or else, it might be prudent not to do anything when the system is on the correct path and to alter the data base only in a situation when it is inferring inaccurate conclusions [2].

Creating New Stereotypes. Rich [2] indicated that the following step in the learning process after the alteration of the present stereotypes is the building of new stereotypes. This could be performed after a system has sufficient models of individual users from which patterns can be constructed. The development of new stereotypes can be done by utilising direct pattern categorisation techniques similar to [112]. These new stereotypes would have the benefits of the automated models, particularly the lack of susceptibility to the biases of the system developer.

To summarise the gaps that we find in the literature and motivate the presented research, the stereotypes often involve features as well as activation conditions (for example a user selecting a type of item or performing an action that actives or deactivates or changes the user stereotype). In our extreme cold start approach the stereotypes are all built based on metadata similarities, without having any activating condition which would preclude or overfit a behaviour during the very first few recommendations. An user (item) belongs to a given stereotype if its metadata features are indicative of a vicinity to the typical users (items) in the same stereotype.

2.5.2 Stereotypes in Recommender Systems

Germane is an instance of a stereotype-based user modelling system in which stereotypes are built automatically by utilising user's relevance feedback, and it is the first instance of a stereotype-based text recommender. Moreover, stereotype weights in Germane are based on stereotype performance in training rather than the degree to which a user is perceived to fit for a stereotype [7].

Germane has been created and implemented to team-role scenario instead of the socio-demographic context which is generally utilised for the research on stereotype-based user modelling. In Germane, automatic feature selection is carried out and in the similar manner for single-component user models as well as stereotypes. In Germane, every stereotype is a text classifier which is required to symbolise the shared interests of a group of users. To achieve this, the same text classification methods can be deployed to develop stereotypes. In Germane, stereotypes once trained, are stored in text files in distinct directories in a similar manner as single-component user models. Utilising a similar approach to develop stereotypes and single-component user models will mean that any dissimilarity among their levels of performance for the same user cannot be attributed to divergence in their derivation [113].

For the purpose of training of a stereotype, the training set is derived from the ratings from each of the group members. It is implicitly assumed that prior to the training, the group membership of each user has been identified. A stereotype can then be developed from a training set comprising of the binary feedback on several text documents from each of the group members. The stereotypes for a specific user are developed independently from each other [7].

Orwant [114] presented DOPPELGANGER which is a generalised tool for gathering, processing, and presenting information about users. In DOPPELGANGER, information flows in a bottom-up fashion where the available sensors decide what inferences could be made, and hence how well users would be modeled and which applications are feasible. Krulwich [66] introduced Lifestyle Finder, in which large scale data are employed to generalise user-specified data along the patterns common to the population, covering areas not described in the user's original data. The approach proposed by Krulwich [66] is very efficient profiles users by utilising a small amount of information. These advantages of the approach come at the expense of a trivial reduction in accuracy but the approach is still accurate enough to be effective.

The personal program guide (PPG) by Ardissono et al. [111] is based on a multi-agent architecture that promotes the integration of different user modeling techniques for recommending programs to viewers. PPG recognises the TV viewer's preferences and suggests the programs to watch. UM-TOOL by Brajnik et al. [115] provides a dynamic way to build stereotypes where such models are created and updated depending on the user activity. UM-TOOL supports a unique approach to user modeling, which is based on both the application of stereotypes and a dynamic reclassification scheme. Shapira et al. [116] evaluated the approaches to building stereotypes and proposed a new model for information filtering systems to resolve the issues related to those approaches. Shapira et al. [116] suggested a compromise method, the essence of defining a group of values for attributes that collectively form a trigger for a stereotype.

Lamche et al. [11] conducted an evaluation of the effectiveness of a user-based stereotypes recommender system for the mobile fashion domain. Results were aligned with previous studies in which the user model based on stereotypes generates better results than non stereotype-based user model. However, the stereotypes were identified by the author before hand. Kamitsios et al. [117] presented a stereotype-based user model in an educational game to offer personalisation according to a player's skill. Likewise, the stereotypes (i.e. mode of the game) were identified by the author. The effectiveness of stereotype-based RS in digital library 'Sowiport' were measured by Beel et al. [12]. The results were not encouraging as the authors assumed one class of stereotypes only (i.e. students and researchers). Thus, all Sowiport visitors were receiving similar recommendations related to specific topics. The work by [118] provides a first attempt at evaluating stereotype-based and non-stereotype-based RS. Nevertheless, in such work, stereotypes were still built using expert knowledge.

In summary, several claims have been made in the literature related to the effectiveness of stereotyping in user modeling, but only a few studies have validated them empirically. Building

a methodology for the automated generation of stereotypes from item's descriptive metadata (independent of user's rating) has not been carried out to date.

2.5.3 Approaches for Stereotypical Groups

In most of the pioneering works on stereotyping, the majority of the classes were built manually by the operator with the domain knowledge. This approach is called the behavioural approach as it is related to behavioral sciences [116]. This approach has obvious limitations, like the operator building classes manually may miss or disregard important relationships (features) that effectively classify and define a stereotype. It is therefore paramount to create a stereotype building procedure that assembles the classes in a systematic manner. To the best of our knowledge there has been no application of the concept of stereotypes to item modeling and efforts to automatically create stereotypes have been limited.

Later research employed the mathematical approach to determine stereotypic groups using some form of mathematical way to generate groups, such as clustering or graph theory [116, 105, 104]. In this research, we will follow the mathematical approach of machine learning. A machine learning based study should address the design of the optimal data representation to tackle the problem at hand, and the same applies to stereotyping. The problem of classification of users or items in stereotypes is itself a statistical challenge because stereotypes need to be defined. Such a problem in machine learning is labelled as an unsupervised learning problem. Clustering-based algorithms applied to a dataset describing items can provide either a direct representation of stereotypes or provide valuable insights into what features are most distinctively driving class separations.

Data clustering (or just clustering) is a way to create groups of objects in such a way that objects in one cluster are similar and objects in different clusters are quite different. Data clustering is confused with classification, where objects are allocated to predefined classes. In data clustering, the classes are not predefined [119]. One of the oldest unsolved problems associated with clustering is how to choose the number of clusters [120]. Data clustering is the process of defining natural groupings within multidimensional data based on similarity measure like Euclidean distance [121, 122].

Recent work on clustering for RS indicates its popularity as a method for enhancing recommendation quality [58]. It is important to remind that the majority of the clustering - similarity- and dimensionality-reduction approaches developed for filtering-based systems or to solve cold-start problems all operate on the users-to-items preference (or rating) matrix [91, 123, 85, 86, 124, 125, 107, 126, 46, 71, 127].

The present work approaches the problem differently by investigating the possibility of obtaining a viable RS that uses stereotypes generated directly via the feature's metadata similarities instead of ratings and preferences. Ratings and preferences-agnostic stereotypes lead to significant dimensionality reduction when the RS is trained. However, they are sufficiently flexible for

capturing general preference traits in a population of users.

One quite similar finding to our work is in Li et al's work [128] which recognised that making recommendation based on user is not appropriate in particular areas. Instead, they applied machine learning to construct clusters for items by implementing a distance matrix. However, the work does not address the cold-start problem.

A cluster is usually identified by a cluster center (or centroid) [129]. Data clustering is a difficult problem in unsupervised learning. With the absence of 'true' labels makes the evaluation of the result a difficult and some what subjective task. Moreover, the clusters in data may have different shapes and sizes [122]. Data-clustering algorithms are very much related to data types. Understanding scale, normalisation, and proximity is therefore necessary in interpreting the results of clustering algorithms. Data type refers to the degree of quantisation in the data [130, 131]. A single attribute can be binary, discrete, or continuous. A binary attribute has exactly two values, such as zero or one. A discrete attribute has a finite range of possible values. While continuous attributes come from an infinite set.

A body of research exists for the application of clustering concepts to numerical and categorical data [132, 133, 134, 135]. The problem of clustering data with mixed numerical and categorical features is an active subject of research. Huang [133] first introduced the so-called k-prototypes clustering algorithm. k-prototypes integrates the k-means and k-modes algorithms through the definition of a combined dissimilarity measure. Improvements on the dissimilarity measures achieving better clustering performance have later been proposed, see for instance [136]. Another popular family of clustering methods, the density-based approaches like DBSCAN first proposed by Ester et al. [137], have also been generalised recently to cope with mixed categorical and numerical features [138].

Most clustering algorithms are based on two common clustering techniques known as hierarchical and partition clustering [139, 140]. A partitional algorithm divides a data set into multiple non-overlapping subsets, whereas a hierarchical algorithm divides a data set into a sequence of nested partitions [119]. Algorithms in hierarchical clustering category generate a cluster tree (or dendrogram) by using heuristic splitting or merging techniques [141]. A cluster tree is defined as "a tree showing a sequence of clustering with each clustering being a partition of the dataset" [142, p. 586]. Several hierarchical algorithms have been proposed in the literature which differs in the way that the two most similar clusters are calculated [143, 131]. More details on specific clustering algorithms are discussed in Chapters 3 and 4.

In this work we demonstrate how clustering procedures, when used in their standard formulation, do not produce well behaved groups that would serve the scope of stereotypes. We attribute this to at least two facts: most modern datasets have high dimensionality in their feature space; modern dataset have complex categorical features which further increase (sometimes disproportionally) the dimension of the feature space. When clustering models attempt to partition such high dimensional spaces they are likely to fail, or to discover structures that are more driven by

slight standardisation issues along each of the coordinates. Our work proposes a different way to utilise clustering models to obtain stereotypes, tackling the problem with an automatic and systematic per feature clustering/grouping. We demonstrate that the resulting groups (stereotypes) are indeed able to describe per feature strong relationships across groups of users (items) that can be effectively and efficiently used during cold start recommendations.

2.6 Evaluation of Recommender Systems

The literature available on RS evaluation presents a large assortment of evaluation metrics, and the most appropriate evaluation metric can be chosen among them. Gunawardana and Shani [144] suggested several evaluation metrics for recommender systems: a) predictive accuracy metrics, b) classification accuracy metrics, c) rank accuracy metrics and d) non-accuracy metrics for the comparison of recommender system algorithms. Based on [145], predictive accuracy or rating prediction metrics are most suitable for situations where an exact prediction of the ratings for every item is of high concern. The most significant representatives of this class are: a) mean absolute error (MAE), b) mean squared error (MSE), c) root mean squared error (RMSE) and d) normalised mean absolute error (NMAE). Recently, rating prediction has been largely abandoned by recommendation researchers and practitioners as it is considered a bad proxy for actual user preferences [60, 61, 27].

Classification accuracy metrics measure the number of correct classifications of relevant or irrelevant items that are produced by the recommender system, and are hence useful for user tasks like finding good items. This type of metric is especially suitable for applications in e-commerce that attempt to convince users of making certain decisions like purchasing products or services. The general basic information retrieval (IR) metrics such as recall and precision are estimated from the number of items that may be either relevant or irrelevant, and either included in the recommendation set of a user or not. Cleverdon and Kean [146] have observed the inversely relation between precision and recall. Therefore, several approaches have been taken to combine both metrics. One commonly used approach is the F1 metric.

Receiver Operating Characteristics (ROC) and Precision Recall (PRC) plots are free-threshold measures as opposed to single-threshold measures such as precision and recall. ROC and PRC are plots of the true positive rate versus the false positive rate for the predictions of a model for multiple thresholds between 0.0 and 1.0.

A rank accuracy or ranking prediction metric estimates the capacity of a recommender system to calculate the correct order of items that best represents the user's preference, which is named the measurement of rank correlation in statistics. The most commonly used metrics include: hit rate (HR), mean reciprocal rank (MRR), normalised discounted cumulative gain (nDCG), mean average precision (MAP) and half-life utility metric (HLU).

Non-accuracy measures (quality measures) like novelty and diversity have moved into the focus of researchers in recent years. We will explore the meaning of serendipity and its applic-

ability to our model in Chapters 6 and 7.

A proper evaluation design is crucial in order to obtain an understanding of the effectiveness of various recommendation algorithms. The evaluation of RSs is often multifaceted, and a single criterion cannot capture many of the designer goals. Incorrect design of the experimental evaluation may result in either overestimation or underestimation of the accuracy of a particular algorithm [3].

Intuitively, RSs can be evaluated using either online methods or offline methods [74]. In an online system, the user reactions are assessed against the recommendations. User participation is therefore essential for online systems. However, because online evaluations require active user participation, it is often not viable to use them in benchmarking and research. Usually there are significant challenges in obtaining access to user conversion data from large-scale user participation systems. On the other hand, one also needs to use datasets of various types, and from multiple domains. Multiple dataset testing is particularly important to ensure that the recommended system is more generalised so that one can be assured that the algorithm works in a variety of settings. In such cases, offline evaluations with historical datasets are used. Offline methods are by far the most popular methods for evaluating RS from a research perspective [3].

Historical data, such as ratings, are used for offline testing. Well known examples of a historical dataset are the Netflix Prize dataset [101] and MovieLens and IMDb datasets [147, 148]. The Netflix[1] dataset was originally released in the context of an online contest, and has since been used as a standardised benchmark for testing many algorithms.

The main advantage of using historical datasets is that multiple datasets from different domains (e.g. news, music, movies) can be used to test the generalisability of the recommender system [3]. To demonstrate and assess the proposed recommendation methodology, we performed the cold-start experiments using two datasets: the integrated set of MovieLens with added metadata from the IMDb database and the publicly available dataset of reviews for item purchases from Amazon.com.

The main limitation of offline tests is that they do not measure the actual propensity of the user to respond to the RS in the future. Moreover, metrics like accuracy do not capture significant characteristics of recommendations, like novelty and serendipity. Such characteristics have important long-term impacts on the conversion rate of the recommendations.

Nevertheless, given these drawbacks, offline approaches tend to be the most commonly accepted techniques for RS evaluation [3] and hence they will be used in the current research.

2.7 Summary

The review of the relevant literature on RSs discussed in this chapter highlighted the three outstanding problems: new user, new item problems and computational efficiency. Rich proposed the idea of user-based stereotyping as a method to solve cold-start problems. However, her ap-

[1] http://www.netflixprize.com/rules

proach was based on expert-driven creation, given the early age for computational algorithm. When computational algorithms gained the ability to discover patterns most of the works were dedicated to the mining of patterns in the rating matrix. That was an obvious choice, but also contradicted the spirit of the original idea. The application of the idea that mining of data can be helpful during cold-start phase is an even lesser researched area, the previous work highlighted the need to solve such problems and in particular the new user problem. However, they revert back to clustering or standard factorisation methods or other transformation which uses the rating matrix.

This project aims to obtain recommendation improvements in cold starts from a different angle, namely feature transformations that are rating agnostic, compared to the widely researched approach up to now, transformations that operate on ratings. A wide range of fully reproducible experiments will be illustrated in the coming chapters.

CHAPTER 3

Problem Analysis

We now live in a world with diverse data sources. Modern data has become more complex and existing stereotype-based approach can not handle it. In order to answer our research questions, we are looking for a dataset that is complex and rich in item metadata such as the combined dataset of MovieLens and IMDb [148]. Such a dataset has a range of features, from simple numerical to categorical and complex categorical. A categorical variable is defined as complex when 1) it cannot be easily translated into a numerical variable via encoding, 2) when the semantics of the categories play an important role in the optimal determination of stereotypes and 3) when it is multi-choice (e.g. there is no predefined minimum or maximum number of labels that describe the item or user). These variables can be viewed as multiple-choice answers on a questionnaire, with the underlying idea being 'pick all that apply'. In the movie domain, typical examples of complex categorical features include the 'genre' and 'keywords' used for labelling movies. For instance, for one item the genre may be categorised as 'drama', whereas for another item might be 'drama' in addition to 'romance' and 'historic'. More details on research methodology and dataset is provided in Sections 3.1 and 3.2.

The problem of generating automatic stereotypes could be approached via successive analysis where the order of increasing complexity will enable to discern effects in the stereotype generation. Our methodology to answer our research questions is first aim to reduce the complexity of the problem by eliminating the complex categorical and categorical features by transforming them to numerical ones, reducing part of the dimensionality of the problem but still retaining as much information content as possible. More details are provided in Sections 3.3 and 3.4.

.

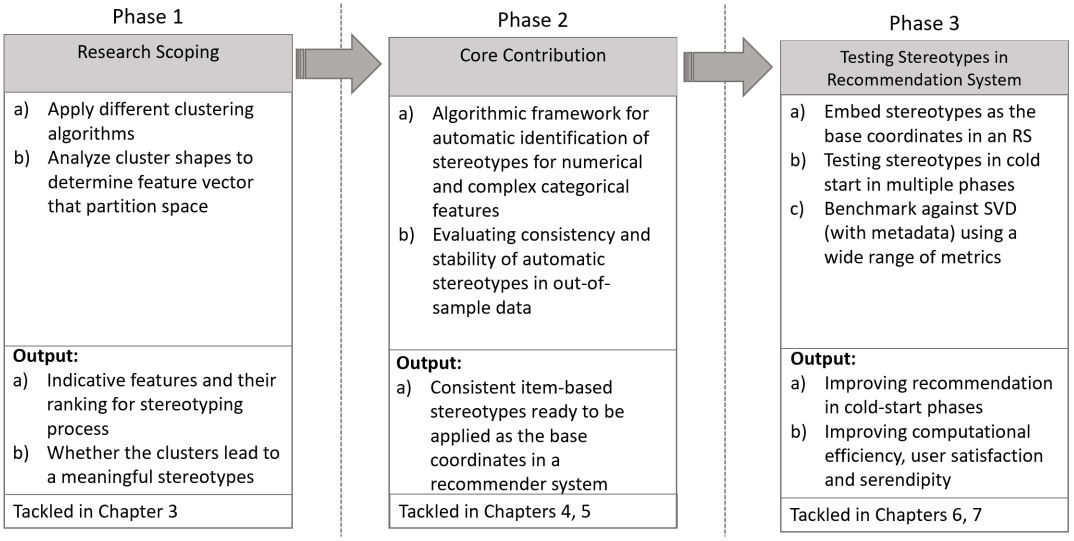


Figure 3.1: Description of the overall research methodology

3.1 Research Methodology

Figure 3.1 outlines the overall research methodology of this thesis. It illustrates three main phases that depict the overall research and answer the research questions.

Phase 1: Research Scoping (this Chapter). In the majority of the studies, both the ones where the learning of the stereotypes is addressed, as well as in those using the stereotypes in a classification/recommender system, the underlying question revolves around feature identification: which features, among the ones available, can be identified as the most indicative of users or items classes, and what values of these features have the highest predictive power for the recommendation? Such sets of features and values constitute the foundations of stereotypes.

For a problem such as item stereotyping, we first investigate whether a mixture of feature engineering and clustering techniques can lead to the automatic formation of clusters that can be used directly as stereotypes. The methodology adopted aims to identify which item features may have the highest importance for the creation of stereotypes. In this chapter, three different clustering algorithms are examined. The main idea is that a repeated identification of the same set of key features among different clustering algorithms would support the concept of item stereotyping in the relevant cluster. Figure 3.2 summarises the analysis process.

The output of this phase will indicates wither the mixture of feature engineering and clustering techniques can lead to a practical and automatic formation of clusters that can be used directly as stereotypes. If this is not the case, then we will handle each feature type separately in Phase 2.

Phase 2: Core Contribution (Chapters 4 and 5). The work in this phase addresses RQ1 outlined in Chapter 1. Our research objective is to define associations between metadata features for both users and items. Such associations prove helpful to an RS in categorising both new users and new items to generate recommendations when few reviews are available.

In Chapter 4 we proposed a methodology for the automatic construction of stereotypes. Firstly, we investigate the effect of complex categorical variables and how these can be handled in a clustering-based stereotyping framework. Secondly, we construct stereotypes automatically for remaining categorical and numerical features. We explain the stereotype creation experiments and showing key results using MovieLens 1M and IMDb dataset.

Chapter 5 proposed a comprehensive statistical tests to evaluate stereotypes accurately. For each stereotype created in Chapter 4, we performed three statistical tests to evaluate the stability, accuracy and predictive content of the stereotypes.

Phase 3: Testing Stereotypes in Recommender System (Chapters 6 and 7). The work in this phase addresses RQ2, RQ3 and RQ4 outlined in Chapter 1. We first proceeds to embed the stereotypes as the base coordinates in an recommender system. In Chapter 6, we predict and recommend which items a user is likely to consume under cold-start scenarios. Then, we focused on the rating predictions using different machine-learning approaches with increasing complexity. Finally, we benchmark our approach against matrix-factorisation techniques. An in-depth analysis of the recommendation quality of the two approaches (stereotypes and factorisation methods) is conducted; such analysis is not limited to recommendation accuracy, but it attempts to investigate other aspects and desirable properties of recommendations, such as utility and novelty.

Chapter 7 validate the stereotype-based approach for cold start recommendation using another dataset. In particular, the chapter validate our answers to research questions: RQ1, RQ2, RQ3 and RQ4 outlined in Chapter 1.

3.2 MovieLens 1 Million Dataset

For RSs research and development, resources such as the MovieLens and the internet movie database (IMDb) datasets provide a rich source of data for the testing and evaluation of recommender systems and have been utilised in many studies such as [43, 149, 126, 150, 60, 58, 151, 152, 61, 85, 104, 153, 154, 155, 156]. The MovieLens and IMDb databases of movies are selected as an example of a challenging item stereotyping problem.

For the stereotype-based approach, three types of information are required for the training and testing user and item metadata:

1. A set of information about users

2. A set of information about items

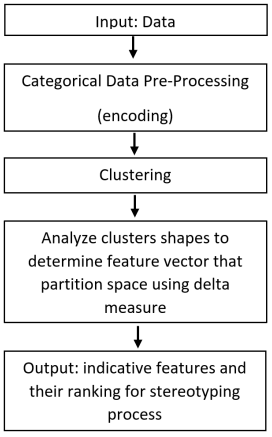


Figure 3.2: Problem analysis process

Item (movie) feature		
Boolean Features	Numerical Features	Categorical Features
Adult	Budget	Original Language
	Popularity	List of Cast ids
	Revenue	List of Cast genders
	Run Time	Movie Genres list
	Release Year	Release Month
	Vote Average	List of Crew ids and Crew roles
	Vote Count	Production Company(s)
		Production Country(s)
		Keywords
User feature		
		Gender
		Age
		Occupation
		Zip Code

Table 3.1: Combined ML 1 Million/IMDb movie and user features

3. Relevant feedback from users i.e. user's rating

The MovieLens dataset is highly popular among the research community with more than 140,000 downloads in 2014 and more than 7,500 references in Google Scholar [147].

Demographic features (e.g. age and gender) of users were extracted from ML 1M dataset and supplementary item features were extracted from IMDb. The combined dataset ML/IMDb contains 6,040 users, 3,827 movies, 1,000,209 ratings, 35,052 cast and 28,541 crew information along with other movie data and user generated features like keywords. Tables 3.1 describes the available users and items (movies) features in the dataset under investigation. This extensive dataset will be used to generate stereotypes in this chapter and Chapters: 4, 5 and to test RS performance in cold start in Chapter 6.

3.3 Clustering Algorithms

The machine learning approach starts with the design of appropriate data representations. Better performance is often obtained by the use of features extracted from the original data. Building a feature representation is an opportunity to leverage domain knowledge into the data that can be very application specific. Thus, a better understanding of the underlying features is useful for building good stereotypes. In spite of that, there are a number of generic feature construction methods such as clustering.

Clustering-based algorithms applied to item metadata provide a direct representation of stereotypes along with valuable insights into which features drive class separations. The main challenge in the application of a clustering algorithm resides in the standardisation of the data. In the most common scenarios, mixed numerical and categorical features are present. Additional complexity may arise from categorical features that are not simply labels but may require applications of semantic similarity metrics into the language they are expressed.

Standard clustering algorithms, like the well-known k-means and its variations, discover structures in the data by applying Euclidean distances, and minimising the total variance of the distances between the cluster's centroids and the individual data points [119]. For categorical features, the concept of distance and order may be difficult to define and, when not meaningless, may introduce unexpected false relationships. For example, suppose that a categorical feature in a dataset is the 'means of transport' and the possible entries are: car, boat, train and helicopter. If one tries to simply introduce a numerical representation for them, for example car= 1, boat= 2, train= 3, helicopter= 4, and apply a standard clustering algorithm on such a new numerical feature, an unwanted and unreal ordering effect will be introduced, namely that cars would be 'nearer' to boats than to trains for example. The standard way to translate such categorical feature to a numerical one is 1 to n encoding, whereas in the example discussed each one of the means of transportation can be encoded as a 1 at that 'coordinate' position amongst zeroes at all other positions indicating non-car transportation means.

A body of research exists for the application of clustering concepts to categorical data. In [133], the k-mode algorithm was introduced to deal with categorical data. In the k-modes algorithm, the centroid of clusters is no longer identified by means but with modes. A matching dissimilarity measure is introduced to deal with categorical objects, and the clustering cost function is minimised by using a frequency-based method to update modes. Several marginal improvements have been introduced to k-modes, for example [134, 135], where the improvements are all directed toward the formation of the dissimilarity measure used. In [157] similarity and efficiency of k-mode is investigated in comparison to the k-median approach.

The problem of clustering data with mixed numerical and categorical features is an active subject of research. Huang [133] first introduced the so-called k-prototypes clustering algorithm. k-prototypes integrates the k-means and k-modes algorithms through the definition of a com-

Numerical Movie (item) Feature Vector Components and Dimensions				
Analysis 1 Numerical Features	**Dimension**	**Analysis 1.B Numerical Features**	**Dimension**	**Categorical Feature Represented**
Budget	1			
Popularity	1			
Revenue	1			
Run Time	1			
Release Year	1			
Vote Average	1			
Vote Count	1			
Is Adult	1			
Cast Popularity	1	Cast Popularity	1	List of Cast Ids
Cast Gender Bias	1	Cast Gender Bias	1	List of Cast Genders
Director Popularity	1	Director Popularity	1	List of Crew Ids & Roles
Encoded Original Language	M (sparse)	Language Popularity	1	Original Language
Encoded Production Comp.	P (sparse)	Encoded Comp. Popularity	1	Production Comp.
Encoded Movie Genre	G (sparse)	Encoded Movie Genre	G (sparse)	Movie Genres List
Encoded Keywords	K (sparse)	Encoded Keywords	K (sparse)	Keywords
Release Time of Year	1	Release Time of Year	1	Release month
Production Coordinates	3	Production Normalized Distance	1	Production Country

Table 3.2: Movie features transformed to a numerical feature vector for Analysis 1 and Analysis 1.B

bined dissimilarity measure. Improvements on the dissimilarity measures for better clustering performance have later been proposed, for instance by Ji et al. [136].

Model-based clustering algorithms and in particular the gaussian mixture models (GMMs) are a major approach to clustering analysis [119]. GMMs assume the data are coming from a mixture of probability distribution, each represents a different cluster. GMMs provide an improvements over k-means as it can cluster stratified area which are difficult to be described well in k-means.

Another family of clustering methods are the density based approaches, like DBSCAN first proposed by Ester et al. [137]. These methods have recently been generalised to cope with mixed categorical and numerical features [138].

3.4 Experimental Evaluation

Table 3.2 describes the available items (movies) features in the combined ML 1M and IMDb datasets. The item features embed the typical complexity that can be encountered in the stereotyping process, with mixed numerical and categorical features. The unshaded rows represent numerical features and thus do not require any pre-processing. The last column of the table shows the original categorical features that have undergone a transformation.

In this and the following sections, a clustering-based technique for the automatic individuation of item-based stereotypes will be discussed. The main challenges posed by the application of clustering techniques to the movies dataset is constituted by the abundance of features and by

the complexity of the categorical features. For example, the feature describing the cast goes as far as having, for each cast role, the actor's name, the gender and the role name. In a similar fashion for each crew member the data contains the name (Stanley Kubrick, Oliver Stones) and role (director, screenwriter).

Thus it is important to perform a transformation to simplify the data. The simplification process is achieved by retaining all of the *numerical* features, as well as all of the categorical ones that can be translated, via feature engineering, to a *numerical* representation which retains the categorical feature's meaning. The allure of this simplified analysis resides in the ability to use a larger range of clustering methodologies - those developed for problems with *numerical* features only.

We apply three state-of-the-art clustering algorithms: k-means, GMMs and DBSCAN to the simplified item data. The results of the clustering algorithms are then investigated further in order to confirm the appropriateness of the encoding techniques that have been applied to categorical features and analyse the 'shape' of the resulting clusters. The aim is to identify which of the item features may have the highest descriptive content for the creation of stereotypes. Figure 3.2 summarises the analysis process.

Analysis 1

In this simplified analysis we aim at eliminating the categorical features by transforming them to numerical ones, thus reducing part of the dimensionality of the problem but still retaining as much information content as possible. The problem of dealing with the lists of cast ids and crew ids can be simplified by considering what a cast id and what a crew id do for a movie. At a very general level (the stereotype level) the cast and crew ids can be used to define popularity like features. For each cast c, let n_c be the count of the number of movies in which cast c is present. The popularity of cast c can be defined as $p_c = log(n_c)$. A Movie's cast popularity (p_m) can be defined as the average over the top N casts (sorted by popularity) of the popularity of each cast, N is chosen to be less or equal to 5 (a movie's cast popularity is dictated by its most popular 5 actors). Crew popularity could be defined in a similar fashion; however, it is possible to further approximate the popularity of the crew with that of the director-arguably the most representative crew member: $p_d = log(n_d)$.

For the remaining categorical features the approach taken is as follow:

- Encode cast gender list to gender bias: for each cast member the gender is available; an average gender bias for each movie can be defined by taking the average across genders of the top N casts (sorted as defined above), after having translated the entries to +1 and -1 for female and male respectively.

- Encode original language and Encode production company(s): the categorical features are transformed via a 1 to n encoding by differentiating across the most popular languages and

production companies, and classify the rest as 'others'.

- Encode movies genre list: the list of genres is again encoded as a 1 to n vector. Each movie might have multiple genres so this is an example of a feature where a non pre-specified number of labels may apply simultaneously to a given item.
- Encode keywords: a 1 to n encoding that retains only the keywords with the largest number of occurrences across all training items is adopted. This is another example of a feature where a non pre-specified number of labels, or sub-categories may apply simultaneously for a given item.
- Release month to release time of year: to capture the yearly periodicity of the months, and the well-known effect of increased movie release activity around the Nov to Jan months, a periodic function is used. Let the release month be represented by a number, 1 for Jan, 2 for Feb, and so on till 12 for Dec. A release time of the year for month i can be defined as: $rt_i = \cos(2\pi i/12)$.
- Production countries: a 1 to n encoding could be used as introduced for other features; however, it can be argued that such an encoding would obfuscate 'cultural vicinity' effects which may be embedded in the production country feature. For example, a Canadian movie may be 'closer' to the cultural mindset of a U.S. movie than an Indian movie, and can be said between European movies and an Australian movies. To capture both the categorical and the proximity effect described, each production country code is encoded to the latitude and longitude coordinates of their capital city. From the latitude and longitude values, using Earth's radius the 3-dimensional spatial coordinates are computed, and used in the feature vector - with the implicit assumption that the Euclidean distance between capital cities is a proxy for the real distance.

The simplified feature vector is as presented in the first two columns of Table 3.2. In the table, M represents the selected number of the most representative languages plus one, P is the number of most representative production companies plus one, G is the total number of movie genres, K represents the selected number of the most representative keywords plus one.

By retaining languages with more than 20 movies then M= 7, by retaining the production companies that have produced at least 20 movies over the data under exam then P= 27, by retaining all movie genres leads to G= 24. Finally, by retaining keywords with a presence in more than 20 movies then K= 84. Therefore, the numerical feature vector has size 157.

Experiment Applied to Analysis 1

k-means

k-means is one of the most popular clustering algorithm, even though technically it is not a clustering but a partition algorithm. The popularity of k-means derives mostly from its simplicity

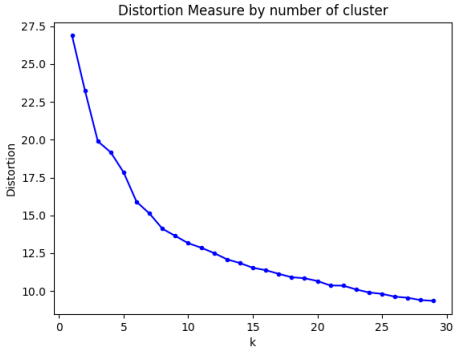


Figure 3.3: Elbow method considerations for k-means using 'fit' a standardised Eucliedean measure

and its computational efficiency. The fact that k-means is not technically a clustering but a partitioning algorithm implies that the algorithm does not directly discover the number of clusters in the data. The standard solution to the problem of using k-means for cluster discovery in a dataset, see for example [119], consists of performing the clustering analysis for an increasing number of clusters and observing metrics that describe the 'silhouette' of the clusters.

It is assumed that the number of clusters in the data can be identified by looking for a well defined kink (or elbow) that changes the slope of the silhouette metric. As the number of cluster increases, the metric describing the silhouette of the clusters (representative of the amount of variance of the points from their respective centroids) will improve. However, the marginal improvement in the silhouette metric may display a change of slope around a certain range of clustering points. That range is indicative of a 'regime change', whereas the marginal improvement of adding additional clusters after the kink will be limited compared to the improvement in the overall silhouette before the kink, and hence indicative of what the true number of clusters in the data may be.

The analysis described is applied to the dataset with numerical features (as described in Table 3.2). Two metrics for the silhouette of the clusters are used: the standard score metrics available in the sklearn[1] metrics, and a standardized Euclidean metric defined as:

$$\sqrt{\sum (u_i - v_i)^2 / V[u_i]} \tag{3.1}$$

For each vector u in the feature space from its respective centroid v, with $V[u_i]$ being the variance of the coordinate i for all $u(s)$ in that centroid. The resulting curves of the two approaches are depicted in Figures 3.3 and 3.4. It is possible to see that both methods suggest a slope change in the chosen silhouette method in the region of 8 to 10 clusters.

Two important drawbacks of using k-means are:

- By the way it is constructed k-means will assume that the clusters are of a globular shape,

[1] https://scikit-learn.org/

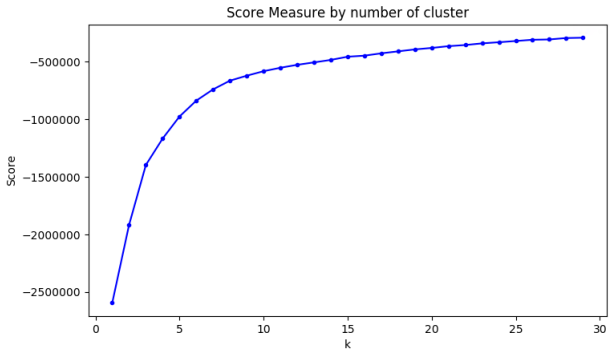


Figure 3.4: Elbow method considerations for k-means using 'fit' a sklearn score measure

other-shapes are difficult to accommodate within the k-means framework, yet clusters that have stratified area will be difficult to be described well in k-means.

- Every point must belong to a cluster, where it is natural to expect that in real case situations a point might not belong to any cluster, or the degree of belonging to a cluster is 'less' as the points are further away from the centroid.

Expectation-Maximization: Gaussian Mixture Models (GMMs)

GMMs provide an improvement over k-means on both the drawbacks of the assumed globular shape and the degree of belonging highlighted above. For each cluster, GMM assumes a N-dimensional gaussian distribution, and assembles the total distribution as the superposition of as many gaussians as clusters. The shape of the gaussian along any axis or particular combination of axis is dictated by the respective variances, therefore in a GMM model a cluster's shape may deviate from a globular shape to a very flat, almost compressed shape. Whilst k-means is a hard partitioning method, a point either belongs to a cluster or another, in GMM there is a degree of fuzziness. A point has a probability of belonging to each of the clusters. Hence, GMM is a soft partitioning method.

Gaussian distributions provide 'flexibility' to the clusters, which are determined via a two steps iterative process:

- An expectation step: for each point find the probability of belonging to each cluster.
- A maximization step: for each cluster, update location and shape based on the probabilities of the data points that have been determined in the previous step.

Similar to k-means, GMMs are technically not clustering, but partitioning algorithms. Therefore, a similar 'elbow' methodology needs to be put in place to discover the most likely number of clusters in the data. Figures 3.5 and 3.6 show an analysis for the increasing clusters numbers

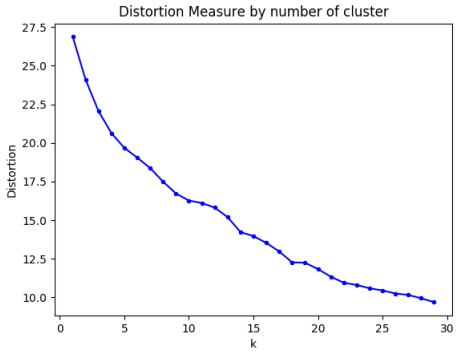


Figure 3.5: Elbow method considerations for GMM using a standardized Eucliedean measure 'fit'

Figure 3.6: Elbow method considerations for GMM using a sklearn score measure 'fit'

generated by the GMM partitioning of the feature space. Whilst from a first analysis of the score method after smoothing out the noise, one could put forward the hypothesis of a natural number of clusters in the data in the region of 5 to 7. The distortion measure exhibits a very uniform (in first derivative) decay of distortion and only a detailed analysis of the derivative displays two potential regions with slight regime shifts, for example at around k= 5, the other less pronounced at k= 20.

Such regions can be identified as 'local plateau' in the approximation of the first derivative of the distortion or score. A local plateau indicates that there is no obvious improvement adding more clusters. Detailed analysis of the score derivative also shows a similar pattern, as it can be seen in Figures 3.8 and 3.7.

Whilst k-means suggests 8 to 10 clusters in the data, the GMMs indicates that there are two 'tiers' of natural clusters in the data. A tier 1 group with about 5 clusters, and tier 2 group with 20 clusters. A more in depth understanding of these features requires the investigation of the clusters' structures as follows.

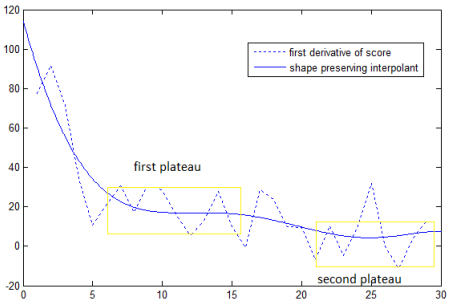


Figure 3.7: Approximation of the first derivative of score for the GMM elbow procedure. Two plateaus can be identified suggesting a first elbow at around k= 5 and a second elbow at around k= 20

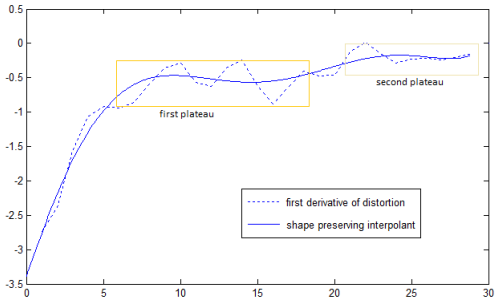


Figure 3.8: Approximation of the first derivative of the distortion for the GMM elbow procedure. Two plateaus can be identified suggesting a first elbow at around k= 5 and a second elbow at around k= 20

Density Based Spatial Clustering Analysis with Noise (DBSCAN)

DBSCAN was originally proposed in [137] and it is a density based clustering algorithm. It does not require that every point is assigned to a cluster, hence does not partition the entire data. Instead, it extracts the 'dense' clusters and leaves sparse background classified as 'noise'. The main advantages of DBSCAN over the k-means and GMM algorithms are:

- The algorithm does not require to specify the number of clusters in the data at prior, as opposed to k-means.
- The algorithm can find arbitrarily shaped clusters.

DBSCAN operationally transforms the space according to the density of the data, points in dense regions are left untouched, points in sparse regions are moved further away. It then iterates multiples over the transformed space determining distances and cutting points according to a distance parameter (called epsilon - ϵ - in most implementations) to get the clusters core points. Finally, it assigns each non-core point to a nearby cluster if the distance parameter satisfies the condition for that core point, otherwise assigns it to the noise.

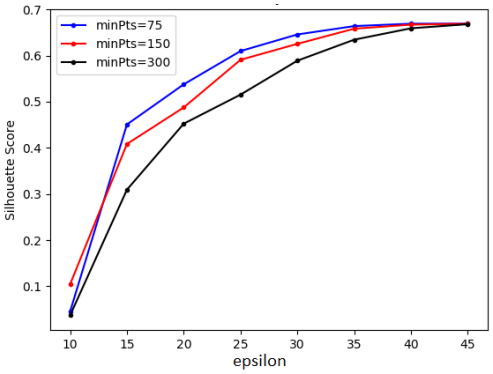


Figure 3.9: Silhouette score as a function of the two parameters of the DBSCAN algorithm. Elbow method considerations suggest an epsilon of about 20

The major drawbacks of DBSCAN resides in the difficult choice of two parameters: epsilon and minPts. Epsilon represents a distance parameter, whilst minPts represents the minimum number of points required to form a dense region. As a rule of thumb, minPts should be chosen to be between d and $2d$, where d is the dimensionality of the problem. As for epsilon (ϵ) if it is chosen much too small, a large part of the data will not be clustered; whereas for a too high value of ϵ, clusters could merge. A procedure similar to the elbow method previously described can be used to identify epsilon. For DBSCAN, the distortion measure cannot be readily defined, and the sklearn implementation does not support a scoring method. For this reason a 'silhouette' index score is used. This is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The silhouette score for a sample is

$$(b - a)/max(a, b) \tag{3.2}$$

In Figure 3.9, the silhouette is displayed for several combinations of minPts and epsilon. A value of 20 for epsilon seems appropriate. With an epsilon of 20, only two natural clusters are discovered inside the data by the DBSCAN algorithm for each of the minPts tested. This result is somewhat of dubious nature and further investigations would be required to analyse the clusters and potentially the data normalisation adopted.

It is worth noting the stability of DBSCAN, as reported in [137] that the method does not produce deterministic results as the order of the sample presented to the algorithm is changed. A sensitivity analysis that randomly change the order of the movie items presented to the model would be required to generalise the result obtained.

Analysis 1 Experimental Result

The initial clustering methods applied to the stereotyping of movie items, when the feature

vector is transformed to be a numerical one, has shown some contradictory findings. Whilst partitioning algorithms (k-means and GMMs) suggest a natural range of clusters in the data of the order of 5 to 10, DBSCAN only discovers 2. The latter result is suspicious and may be attributed to a sub-optimal normalisation of the feature coordinates which creates false density structures in some of the features coordinates or sub-spaces (i.e. sets of coordinates).

The results of clustering can be used to build stereotypes. Different methodologies to link the clustering results to stereotypes are subjects of ongoing investigations, however in the current context the simplest and most intuitive way to generate a stereotype class from a clustering result is to assign the cluster centroid the role of stereotype for all items that fall in its vicinity in the feature space. Investigation of the stereotypes (clusters) structures should reveal further insights into the directions along which the algorithms operate and shed light into potential normalisation or encoding problems.

The results of the clustering algorithms lead to further investigations into the analysis of the clusters 'shape'. It is possible to gain a sense of the clustering structures by synthesizing and studying an estimator of the 'rank' of the feature separation between each coordinate of the cluster centers. To do so, for a given clustering algorithm (a) and a given number of clusters (n), a normalised distance between the cluster centroid i and the cluster centroid o along coordinate x_j can be defined as:

$$d_j^{i,o} = \frac{abs(x_j^i - x_j^o)}{abs(x_j^o)} \tag{3.3}$$

And with that, a delta for each coordinate x_j and algorithm a, for a number of clusters n can be introduced as:

$$\delta_j^{n,a} = max_i(d_j^{i,o}) \qquad for \quad i = 1, \ldots, n-1 \tag{3.4}$$

The delta measure can serve as a proxy to estimate the degree of importance of each direction (coordinate) in determining the feature space separation. Based on the amount of 'space' between centroids, it provides a way to estimate the relative ranking of each direction and the 'shape' of the clusters. Figure 3.10 shows a simple example on how the delta works in ranking the directions of feature separation in a two dimensional space. It can be seen that direction X is much more important than direction Y in separating the three clusters.

The analysis of the directions ranking according to the degree of separation defined by $\delta_j^{n,a}$, as the number of clusters increases, is particularly relevant for the k-means and the GMMs models. It provides an intuition for what directions of the feature vector space are most effective and stable in separating the space. The analysis of the delta measure also enables a critical examination of the numerical feature directions, thus providing an idea of whether the normalisation and the encoding performed on the feature is suitable for clustering or may skew the results. In the case of DBSCAN, where the number of clusters is an output of the algorithm, the algorithm does not

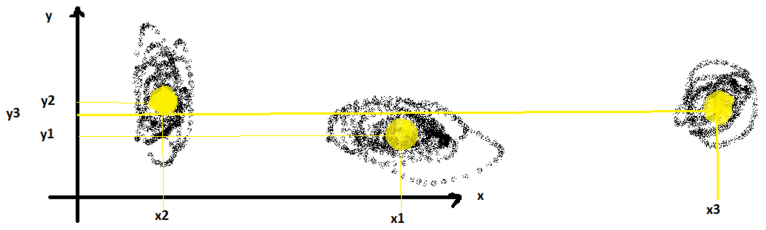


Figure 3.10: Ranking feature separation in feature space using $\delta_j^{n,a}$

define centroids, the measure defined in 3.4 is not directly applicable. To generalise the measures in 3.3 and 3.4 for DBSCAN, it is necessary to define the centroids of the clusters, and this can be done for cluster i by taking the expected value $\mathbb{E}[.]$ along each coordinate j for all ϵ points belonging to the cluster:

$$x_j^i = \mathbb{E}[\epsilon_j^i] \tag{3.5}$$

With such a definition for the cluster's centroid it is possible to introduce the delta measure in the same way as the measures in 3.3 and 3.4 for DBSCAN.

Tables 3.3, 3.4 and 3.5 show the top 7 ranked directions (or subspaces when a feature is composed by multiple coordinates) according to metric 3.4 for the three algorithms examined in Section 3.4. There is a clear and distinct feature separation, and that different algorithms tend to discover similar predominant features (coordinates) along which space separation occurs. For instance, it is noticeable how features like popularity, cast and director's popularity appear to be less effective as clustering coordinates than features like production country, keywords, genre and production company. Also, it is interesting to note how the ranking of features depends on the number of clusters. For a given number of clusters there is a definitive similarity across algorithms, with the partitioning algorithms (k-means and GMMs) agreeing on both the top four most important directions and their relative rankings.

A striking finding from the analysis is that all the directions scoring high in terms of feature space separation are the ones that have been encoded from categorical values to a subspace of a certain size in the numerical feature vector. For example, production company is encoded as a numerical feature of 27 dimensions as shown in Table 3.2. It is therefore important to evaluate whether the encoding is the potential root cause of such an observation. In the next section, changes are made to the way the features are transformed (encoded) and this leads to an alternative analysis.

Analysis 1.B

The initial findings of Section 3.4 suggest a further revision of the features introduced in Table 3.2, this is needed in order to shed light on the questions below:

- Why does the feature 'production company' has such a high importance in the clustering

Clus / Dir	2	3	5	10	15	20	25
Direction 1	Production Country	Production Company	Keywords	Genre	Production Country	Production Country	Production Country
Direction 2	Genre	Keywords	Genre	Keywords	Keywords	Keywords	Keywords
Direction 3	Keywords	Production Country	Production Country	Production Country	Genre	Genre	Genre
Direction 4	Production Company	Genre	Production Company	Language	Production Company	Production Company	Production Company
Direction 5	Release Month	Language	Language	Budget	Popularity of Cast	Popularity of Cast	Popularity of Cast
Direction 6	Popularity of Cast	Budget	Budget	Production Company	Release Month	Vote	Language
Direction 7	Budget	Vote	Popularity of Cast	Popularity of Director	Popularity of Director	Language	Vote

Table 3.3: Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to k-means

Clus / Dir	2	3	5	10	15	20	25
Direction 1	Production Company	Keywords	Keywords	Production Country	Production Country	Production Country	Production Country
Direction 2	Release Month	Genre	Genre	Keywords	Keywords	Keywords	Keywords
Direction 3	Genre	Production Company	Production Country	Genre	Genre	Genre	Genre
Direction 4	Keywords	Production Country	Production Company	Production Company	Production Company	Production Company	Production Company
Direction 5	Production Country	Language	Language	Budget	Release Month	Release Month	Release Month
Direction 6	Popularity of Cast	Budget	Popularity of Cast	Language	Language	Language	Language
Direction 7	Budget	Release Month	Release Month	Release Month	Budget	Budget	Budget

Table 3.4: Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to GMM

Clusters / Directions	2
Direction 1	Production Company
Direction 2	Keywords
Direction 3	Release Month
Direction 4	Production Country
Direction 5	Genre
Direction 6	Language
Direction 7	Budget

Table 3.5: Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for DBSCAN

process? Intuition would suggest that such a feature should be of the same order of importance as the feature cast Ids and director Id. However, 'production company' consistently ranks higher than the cast and director features. It is natural to question whether this effect is real or whether it is introduced by the different encoding. Production company was encoded 1 to n, whilst cast and director features were represented via the popularity scalar value.

- As the clustering number increases, the feature 'production country' becomes the prevailing separation feature. Is this due to the scale of this metric? Could the same information be presented in a different way to the clustering algorithms?
- Are keywords and genre truly important or their rank is a side effect of the large dimension of their subspace?

To gain further insights into the problems above, a revision of some of the features is proposed below:

- Language to language popularity: $p_L = log(n_L)$, where n_L is the count of movies in the training set with language L.
- Production company to production company popularity: in a similar fashion as what was done in Table 3.2 for cast and director popularity, the production company popularity can be introduced as $p_{pc} = log(n_{pc})$, where n_{pc} is the count of movies in the training set produced by the company.
- Production coordinates to normalised distances: in order to retain the basic idea that the closer the locations of production for a movie the more culturally similar the movies are likely to be, and at the same time simplify and scale this feature, it might be necessary to introduce a normalised distance. By fixing an origin on the surface of the earth, for example the coordinates of the Greenwich meridian in London, it is possible to compute the distance from such origin to each movie production country (which had been transformed to a location as shown in Table 3.2). The distance is computed via the geodesic (shortest distance on the curved surface). In order to differentiate between two points at a similar distance but on the opposite sides of the origin the distance is then complemented with a sign. Positive if the geodesic is moving east of the origin, negative if the geodesic is moving west of the origin (sign of longitude). One further problem arises, the earth can be approximated via a sphere and there could be points that have a large positive distance compared to the diameter of the earth and points that have a large negative distance and yet they are actually close in geographical sense. To account for such characteristic, a periodic function is introduced. Let $\omega_{0,P}$ be the signed geodesic distance between point 0 (the origin) and point P, and R be the earth radius, then the normalised distance can be defined as:

$$\Delta_{0,p} = cos(\frac{\pi\omega_{0,P}}{2R}) \tag{3.6}$$

Which also conveniently falls in the -1 to +1 range as to conform with the typical scales of the other numerical features.

With the proposed changes discussed above, the new analysis can be run with the new feature vector space of third and fourth columns from the right in Table 3.2. As before by retaining all movie genres then G= 24 and by retaining keywords with a presence in more than 20 movies then K= 84. Therefore, the new numerical feature vector has size 123.

Experiment Applied to Analysis 1.B

In the new feature vector in Table 3.2 with the new feature space of size 123 (compared to the size of 157 of the analysis 1), we repeat the elbow procedures for the identification of the most likely number of clusters for both k-means and GMM clustering models. The results are displayed in Figures 3.11 and 3.12. The measures introduced previously, namely the score and the distortion, are displayed together with those for the same algorithm applied to the wider feature vector from analysis 1.

The first important finding is that for the k-means algorithm, both the score and the distortion measures exhibit only minimal variations, despite a 20% reduction of the feature vector space between the two analyses. Analysis 1.B elbow procedure confirms the existence of a natural number of clusters in the data between 5 and 10, with 8 being the most likely value. Secondly, whilst in analysis 1 using GMM was only weakly conclusive about the kink (elbow) location, i.e. an elbow in the score was less pronounced and there were two likely regimes identified. In analysis 1.B, the score displays a much more defined elbow and the elbow location is consistent with a 5 to 10 natural clusters, confirming the k-means finding in a clearer manner. The score of GMM, which represents the log-likelihood of the sample data, slightly decreases which may be consistent with the dimension reduction.

To understand how the cluster's shape is influenced under the new feature vector, the same analysis is repeated for the newer clustering results. This derives the preferential directions as well as the ranking of directions according to the separation delta measure.

Tables 3.6 and 3.7 show the resulting features, ranked by maximum separation in the feature space, as the clustering models proceed from fewer to more clusters. The first important finding of this investigation is that there seems to be a more defined 'cut off' in separation measure, whereas a handful of features contribute to most of the separations, and the others contribute to at least one less separation. In order to reinforce the fact that there are dominant directions identified by the clustering algorithm, the non-dominant features are left blank.

The second finding is that the emerging features ranking is relatively stable across cluster's dimension as well as algorithm (GMM vs k-means). Keywords and genre appear to be the dom-

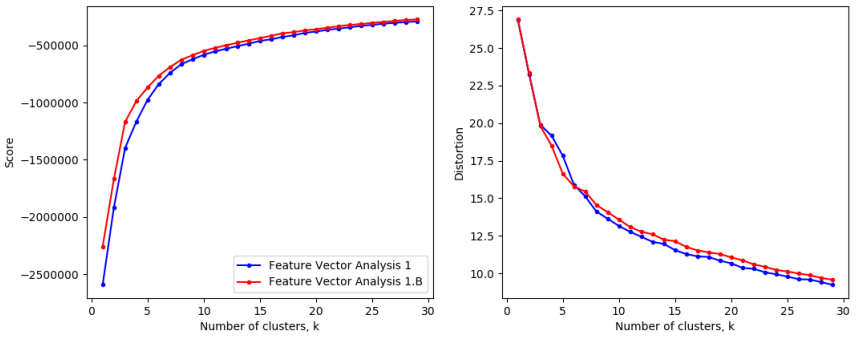


Figure 3.11: Elbow method considerations for k-means, Analysis 1 vs Analysis 1.B

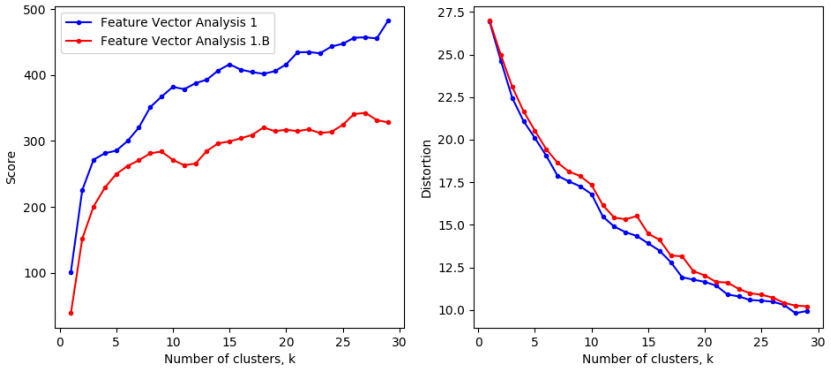


Figure 3.12: Elbow method considerations for GMM, Analysis 1 vs Analysis1.B

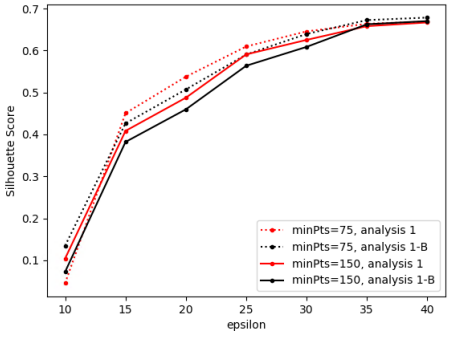


Figure 3.13: Elbow method considerations for the silhouette score over the parameter space for DBSCAN, Analysis 1 vs Analysis 1.B

inant features, and in such an order, pretty much across algorithms and cluster sizes, with only the exclusion of small clustering dimensions where genre and even budget can take a leading role. As the clustering dimension increases, country distance (the new feature introduced in analysis 1.B to represent the production country), still arises as one of the most important directions, and this confirms the finding of analysis 1. Even more importantly this show that even a complex mathematical manipulation, such as the one that reduces that feature space from dimension 3 to dimension 1, only minimally influence the results. Ultimately the importance of the feature budget is confirmed, and as clustering dimension increases, past the elbow, the new feature revenue is discovered as an important separating feature.

Having observed the results given by partitioning algorithms like k-means and GMMs, it is interesting to look also at the results of a density based approach like DBSCAN. As before, the DBSCAN procedure differs from that of the other partitioning algorithms as the number of cluster structures is actually an output of the model, once the parameters epsilon and minimum number of points are selected as discussed earlier. To select these parameters, an elbow like methodology was suggested over a silhouette target function as displayed in Figure 3.13. DBSCAN applied to the feature vector of analysis 1.B actually reveals a silhouette score that is slightly lower than that of analysis 1, but with an asymptote at high epsilon that is slightly higher. The two parameters of the model are chosen to be almost identical to that of analysis 1, and still lead to two cluster structures.

Table 3.8 shows the dominant feature directions according to the separation measure previously introduced and adapted for DBSCAN. The main finding is that for analysis 1.B the role of keywords is not confirmed by DBSCAN. Instead budget is ranked as the most significant feature to create separation, followed by genre, country distance and keywords.

Analysis 1.B Experimental Result

Analyses 1 and 1.B led to the following findings:

1. If a categorical feature is truly important for a given problem, and if that feature can be transformed to a numerical feature by retaining the meaning of what it represents, then the transformation is not likely to influence the feature's importance in an automated stereotyping process. This is the case for the production country, which is designed to reduce its dimension but retain the meaning of the feature. This does not affect the fact that it is a top ranking feature as per separation and hence its importance in a stereotyping process.

2. If a categorical feature has a small importance overall in defining separations among groups of items, it would be disregarded or poorly weighted in a stereotyping process. Encoding the feature simply as 1 to n could potentially introduce false structures, especially when the encoded dimensions are large. The example where the above applies in the current

Clus / Dir	2	3	5	10	15	20	25
Direction 1	Genre	Keywords	Keywords	Keywords	Keywords	Keywords	Keywords
Direction 2	Keywords	Budget	Genre	Genre	Genre	Genre	Genre
Direction 3	Budget	Genre	Budget	Budget	Budget	Budget	Revenue
Direction 4	Country Distance	Country Distance	Country Distance	Country Distance	Country Distance	Country Distance	Budget
Direction 5	Release Month	Release Month	Release Month			Revenue	Country Distance
Direction 6							
Direction 7							

Table 3.6: Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to k-means for Analysis 1.B

Clus / Dir	2	3	5	10	15	20	25
Direction 1	Genre	Genre	Keywords	Keywords	Keywords	Keywords	Keywords
Direction 2	Keywords	Keywords	Genre	Genre	Genre	Genre	Genre
Direction 3	Release Month	Country Distance	Country Distance	Country Distance	Country Distance	Budget	Budget
Direction 4	Budget	Prod. Comp.Pop.		Budget	Release Month	Revenue	Revenue
Direction 5		Budget		Release Month	Prod. Comp.Pop.	Country Distance	Country Distance
Direction 6		Release Month			Revenue	Prod. Comp.Pop.	Release Month
Direction 7					Budget	Release Month	Prod. Comp.Pop.

Table 3.7: Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to gaussian mixture model (GMM) for Analysis 1.B

Clusters / Directions	2
Direction 1	Budget
Direction 2	Genre
Direction 3	Country Distance
Direction 4	Keywords
Direction 5	Popularity
Direction 6	
Direction 7	

Table 3.8: Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for DBSCAN for Analysis 1B

problem is that of the production company feature. When such a feature is encoded as a 1 to n, it scores consistently in the top positions of the ranking features. However when it is transformed to production company popularity its rank is changed to that of a secondary feature.

3. If a categorical feature with a large number of categories can be synthesised via a mathematical formula, tailored to both retaining the information content and reduce its dimensionality, then this can facilitate the application to stereotyping of numerical clustering methods. However, if a categorical feature with a large number of categories cannot be reduced via a mathematical model to a simpler numerical feature, then one should not employ the naive 1 to n numerical encoding in the context of stereotyping.

The findings above imply that the conducted analyses 1 and 1.B lead to several important conclusions, some of which are not fully expected (for example the importance of production country, budget, release month). However the same analysis is affected by potentially spurious structures introduced by the 1 to n encoding of the two features: genre and keywords. Further reducing such feature spaces to smaller numerical coordinates is a challenging task. It would require in depth research on the potential applications of lexicographic and semantic distances, and grouping algorithms that would take into account the underlying meaning of the words in each category. For example, measuring how 'close' two genres are, and how 'close' keywords are, as well as how 'close' could categories of different features be amongst each other (e.g. how a keyword may be more likely related to a set of genres rather than another) constitute critical problems for such a research. Tackling these problems would be very beneficial for any robust and successful stereotyping model to be applied to real life problems.

Before formally introducing and researching potential solutions to such problems, a further simplified analysis can be carried out to confirm or disprove the current findings regarding the stereotyping problem. In the new analysis, which can be called analysis 1.C, the subspaces of genre and keywords are removed from the feature vector. The analysis of only the numerical feature vector, deprived of any 1 to n encoding, will demonstrate whether the presence of 1 to n encoded features having large n compared to the feature space is not only influencing the clustering along the encoded directions but also along the other directions. In simpler words, analysis 1.C is carried out to validate the findings about the numerical (non-encoded) features.

Analysis 1.C

Repeating the elbow procedures for the identification of the most likely number of clusters (stereotypes) in the feature vector of Table 3.2, where both genre and keywords 1 to n encoded features have been removed, using k-means and GMM clustering models, leads to the results of Figures 3.14 and 3.15. In particular, k-means clustering does not improve neither in score, nor in distortion over analysis 1.B, and the elbow suggests the very same number of clusters as that of

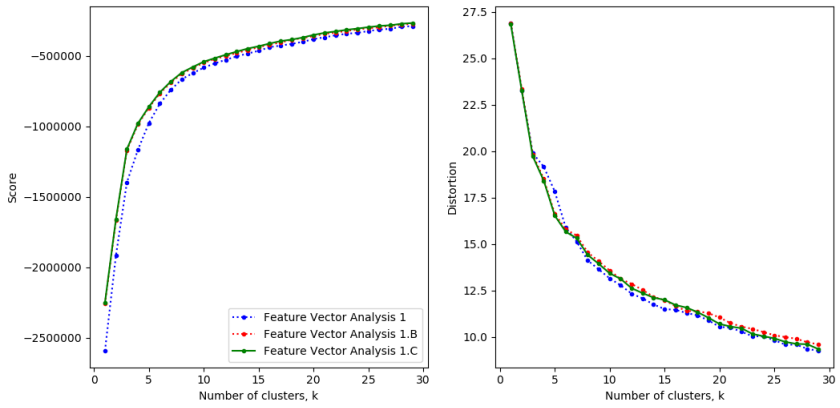


Figure 3.14: Elbow method considerations for k-means, Analyses 1 vs 1.B vs 1.C

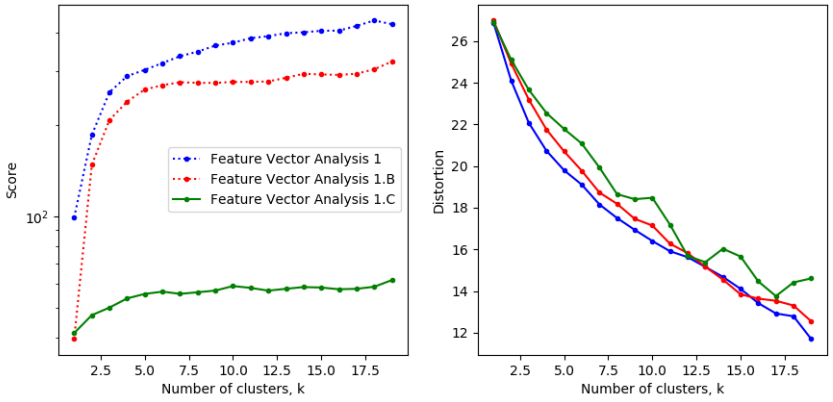


Figure 3.15: Elbow method considerations for GMM, Analyses 1 vs 1.B vs 1.C

analysis 1.B.

For the GMM however, the situation differs substantially. The score of analysis 1.C is much lower (in GMM the score represents the per sample average log-likelihood of the data), therefore suggesting that the data may not be well represented via joint normal distributions. Perhaps having removed the large number of encoded dimensions is what highlights the non-normality of the data. Somehow this fact should have been expected as several directions are introduced as the logarithm of a positive defined quantity. The distortion is also slightly worse for analysis 1.B than analysis 1 but it is at a comparable level. However, rather than the measures of cluster 'fit', the main objective of analysis 1.C is to look at the main directions identified by the clustering procedures in order to prove or disprove the conclusions that analyses 1 and 1.B suggested. Tables 3.9 and 3.10 show the features ranking for analysis 1.C, and the same exact features, with almost identical ranking as analysis 1.B, are identified by the separation measure as the

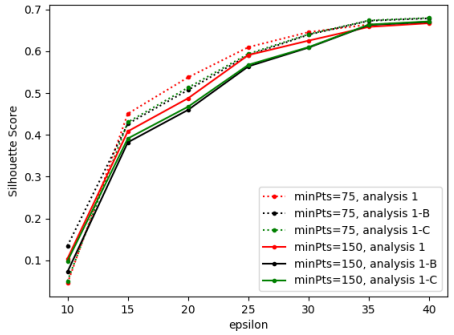


Figure 3.16: Silhouette score as a function of the two parameters of the DBSCAN algorithm. Elbow method considerations suggest an epsilon of about 20 across all analyses

Clus / Dir	2	3	5	10	15	20	25
Direction 1	Country Distance	Budget	Budget	Budget	Budget	Budget	Revenue
Direction 2	Budget	Country Distance	Country Distance	Country Distance	Revenue	Country Distance	Budget
Direction 3	Release Month	Release Month	Revenue	Revenue	Country Distance	Revenue	Country Distance
Direction 4	Revenue	Revenue	Release Month	Release Month	Release Month	Release Month	Release Month
Direction 5			Popularity	Popularity	Popularity	Popularity	Popularity
Direction 6							
Direction 7							

Table 3.9: Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to k-means for Analysis 1.C

Clus / Dir	2	3	5	10	15	20	25
Direction 1	Country Distance	Country Distance	Country Distance	Country Distance	Revenue	Budget	Revenue
Direction 2	Budget	Budget	Budget	Budget	Budget	Revenue	Revenue
Direction 3	Prod. Comp.Pop.	Release Month	Revenue	Revenue	Country Distance	Country Distance	Country Distance
Direction 4	Release Month	Prod. Comp.Pop.	Release Month	Release Month	Release Month	Release Month	Release Month
Direction 5		Revenue	Prod. Com.Pop.	Prod. Comp.Pop.	Prod. Comp.Pop.	Popularity	Prod. Comp.Pop.
Direction 6					Popularity of Cast	Prod. Comp.Pop.	
Direction 7							

Table 3.10: Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for an increasing number of clusters (n) and algorithm (a) equals to GMM for Analysis 1.C

Clusters / Directions	2
Direction 1	Country Distance
Direction 2	Budget
Direction 3	Revenue
Direction 4	Release Month
Direction 5	Popularity
Direction 6	Popularity of Cast
Direction 7	Popularity of Director

Table 3.11: Top 7 ranked feature directions (or subspace of directions) according to maximum separation as measured by $\delta_j^{n,a}$ for DBSCAN for Analysis 1.C

fundamental features in the clustering analysis 1.C by both k-means and GMM. These findings validate the conclusions reached during analyses 1 and 1.B.

Conducting analysis 1.C via DBSCAN leads to the elbow procedure in parameter space of Figure 3.16. The reduced feature vector of analysis 1.C still leads to the same parameter selection as analyses 1 and 1.B. Yet again, DBSCAN identifies two clusters and for these the main separating directions are reported in Table 3.11, and once more the usual top four directions of feature space separation identified are consistent with the previous findings. With lower degree of importance, DBSCAN applied to analysis 1.C also discovers the popularities of the movie, of the cast and of the director as further separation features. Analysis 1.C therefore overall confirms that the conclusions are valid for this problem.

3.5 Summary

An initial analysis of clustering methods applied to the problem of the identification of stereotypes for movie items using a series of specially assembled numerical feature vectors was conducted. The analysis was then further simplified to demonstrate the effect on the stereotyping

process of 1 to n encoding versus ad-hoc mathematical transformation of categorical features.

Results show that some features are more indicative than others in separating items, and their presence supports the idea of using cluster separation as a way to obtain stereotypes. Clustering of the item metadata appears to be a promising approach to generate stereotypes because it allows one to determine the importance of features automatically but, at the same time, the investigation demonstrates that clustering *cannot* be directly applied to stereotype creation - at least not for problems where both numerical and complex categorical features are present at the same time. This is because in some cases the separation of the cluster centroids may be dominated by a single feature (for example genre or keyword), as a result of the high dimensionality of the feature itself and not because of the feature true importance.

In the next chapter, we will investigate the effect of complex multi-valued categorical variables like genre and keywords, and how these can be handled in a clustering-based stereotyping framework. Also, we will look at how to automate remaining features - numerical and non complex categorical features.

CHAPTER 4

Automatic Construction of Item-Based Stereotypes

The research conducted in Chapter 3 into the direct application of clustering algorithms for the creation of stereotypes indicates an inherent difficulty in obtaining structures (stereotypes) due to the vastity of the dimension and the diversity and complexity of some of the features, with the resulting stereotypes dominated by the high dimensional features. Such a research lead to a preliminary conclusion that, when in the presence of high dimensionality and distinct complex categorical features, the straightforward application of a clustering approach might not lead to the desired, easily interpretable stereotyping framework.

The main contribution of Chapter 4 is to propose an algorithmic framework for the automatic identification of stereotypes for both numerical and complex categorical features. This chapter address the following research question:
Can item-based stereotypes, not based on rating, be constructed automatically?

The novel approach to automatic stereotype construction for complex categorical and numerical features is proposed in Sections 4.1 and 4.2 respectively, while Section 4.3 describes the experiment for automatic stereotype construction.

4.1 Stereotypes for Complex Categorical Features

Using clustering as a technique for the automatic individuation of stereotypes, lead to a gap/disconnect between categorical features that can be transformed to simple numerical features, still retaining their information content, and categorical features that are not easily transformable. In particular, for the problem at hand, the two features (i.e. movie genre, movie keywords) that were identified as being highly important in the previous context, cannot be easily translated into a numerical

representation without recurring to complex semantic distances or representations like those deriving from natural language processing research, see for instance the famous research that lead to the word2vec algorithm [158].

Complex categorical variables make the application of stereotypes to RS more challenging, but at the same time they are more representative of the real-world due to the fact that the most important categorical variables are multi-entry categorical in a non-strict sense. By that we mean that the entry of an item for a given categorical features, genre for example, is not simply one category, e.g. 'drama', but it can take several entries without any preassigned number, e.g. for one item the genre may be categorised as 'drama', for another item it may be 'drama' plus 'romance' plus 'historic'. These features can be viewed as multiple choices categorical variables, in such a way to resemble multiple choice answers to a questionnaire and the idea is 'pick all that applies'.

To the best of the author's knowledge, all the available categorical clustering algorithms like k-modes and its possible variations have been developed for single choice categorical variables, hence they cannot be applied directly to multi-entry categorical features. However, a transformation can be employed to transform multiple-entry categorical variables to a series of variables that can be handled by k-modes. For each possible value of the category, a new variable is introduced to represent a true/false encoding. This is not dissimilar from the concept of 1 to n encoding, but in this context it is used to define a multi-valued categorical representation, but not a numerical coordinate instead.

Multi-entry categorical features do not just bring an extra level of complexity, they also provide an understanding of the intrinsic relationships between categories which come directly from the process that generated the data.

For a complex categorical feature there exist a number of entries where multiple labels are assigned to the same item. By investigating a large enough set of items with multiple entries one can extract what type of relationships exist, if any, between the categories. This can be done by investigating the correlation matrix of the encoded multi-entry feature. The first step consists of converting the categorical feature in a multi-1 to n encoding, and then computing the correlation matrix between categories.

The correlation matrix can be defined in a standard way; given a multi-1 to n encoded observation for the multi-entry categorical feature, X_i for $i = 1, \ldots, N$ possible categories, the covariance matrix is defined as:

$$CV_{i,j} = \mathbb{E}\left[(X_i - \mu_i)(X_j - \mu_j)\right] \qquad i, j = 1, \ldots, N \tag{4.1}$$

$$\mu_k = \mathbb{E}\left[X_k\right]$$

Where $\mathbb{E}[\ \ ]$ is the expected value operator. From the covariance matrix, the correlation matrix R is obtained normalising the covariance matrix by the product of the standard deviations

along directions i, j:

$$R_{i,j} = \frac{CV_{i,j}}{\sigma_i \sigma_j} \tag{4.2}$$

$$\sigma_k^2 = \mathbb{E}\left[(X_k - \mu_k)^2\right]$$

The values in the correlation matrix 4.2 would already suggest which categories are better coupled with each other.

- A positive correlation indicates similarity, i.e. the two categories are likely to be found together as a multi-entry item description.
- A near zero correlation indicates that there is no significant relationship between the two categories
- A negative correlation indicates that the two categories are 'antithetic' meaning that the two categories are unlikely to be found together in a multi-entry item description.

To further enhance the grouping between categories, one can group entries of the correlation matrix that are most related with each other. Several algorithms have been proposed in the literature, see for example [159] and [160] references therein. Most of these revolve around the application of hierarchical clustering using the correlation matrix entries to define a penalty distance function. The penalty function can be introduced in several different ways, see [161] for a range of dissimilarity metric examples. In this context we will work with a simple linear metric (also referred to as penalty P) defined as:

$$P_{i,j} = 1 - |R_{i,j}| \tag{4.3}$$

Which constitutes a simple linear penalty: low correlations around 0 are penalised more than high positive (near +1) or negative correlations (near -1).

Following [160], in order to apply formal clustering to a correlation matrix, it is necessary to introduce both a 'metric' that defines distances between pair of observations and a 'linkage' criterion whose role is to define the similarity/dissimilarity across groups (clusters) of observations based on the metric of distances between single observations. In order to define a metric/distance that starts from the concept of correlation and that respects the properties:
a) positive defined, b) elements that are nearer to each other have lower distance than elements that are further apart from each other; the concept of correlation needs to be somewhat inverted.

Correlation measures stronger positive (negative) relationships with values that are further away from 0 and toward +1 (-1). This needs to be inverted in the sense that the closer the correlation to $|1|$, between two observations/entries, the smaller the distance, with the limiting case of correlation going toward +1 (-1) and distance approaching 0. Such an inverted correlation

metric, can be obtained in several different ways, and it is often called dissimilarity measure (to emphasize the inverse role respect to correlation which measures similarity), see [161] for examples. Two such ways in which the dissimilarity can be obtained are as follow:

$$D_{i,j}^{(a)} = 1 - |R_{i,j}| \tag{4.4}$$

$$D_{i,j}^{(a)} = \sqrt{1 - R_{i,j}^2} \tag{4.5}$$

The dissimilarity measures 4.4 and 4.5 need to be complemented with a linkage criterion which determines the distance between groups of observations based on the dissimilarity measures between single observations. In the hierarchical clustering literature, there are many alternative linkages proposed, see [162] for a general review. In this research the most widespread and general linkages will be employed. For a given set of observations **A**, and a second set of observations **B**, the three linkage criterions adopted for each dissimilarity measure D are:

- The single-linkage:

$$L(A, B) = min\{D_{i,j} \quad with \quad i \in A, j \in B\} \tag{4.6}$$

- The complete-linkage:

$$L(A, B) = max\{D_{i,j} \quad with \quad i \in A, j \in B\} \tag{4.7}$$

- The Ward-linkage: this is better explained by analysing what happens when group **A** is aggregated to a new group **C**, and their aggregate distance compute via the linkage from **B**:

$$\begin{aligned} L(A + C, B) = \alpha_1 D_{i,j} + \alpha_2 D_{k,j} + \alpha_3 D_{k,i} \quad with \quad i \in A, j \in B, k \in C \\ \alpha_1 = (n_A + n_B)/(n_A + n_B + n_C) \\ \alpha_2 = (n_C + n_B)/(n_A + n_B + n_C) \\ \alpha_3 = -n_B/(n_A + n_B + n_C) \end{aligned} \tag{4.8}$$

with n_A, n_B, n_C are the sizes of the sets **A**, **B**, **C** respectively. Linkage 4.8 defines a linkage criterion by recursion, which when applied as a clustering determinant minimises the squares of the intracluster dissimilarities.

Our suggested method for creating stereotypes automatically for complex categorical features rests on the systematic truncation of the dendrogram of the hierarchical clustering proced-

ure. More information about traditional hierarchical clustering methods can be found in [163]. A complication is found in the choice of the penalty function to adopt, a quadratic penalty as defined in formula 4.5 tends to compress excessively toward 1.0 entries that have low correlations (in general below 0.4 - 0.5 in absolute value) and the resulting dendrograms appear to be too compressed when dealing with correlation matrices that have average/low correlations in magnitude. The linear penalty function defined in formula 4.4 is more suited for exploring situations where the correlations tend to be low on average, while a quadratic penalty formula such as the on in 4.5 is suited for situations where the average correlation is high (above 0.4 - 0.5 in absolute value).

A dendrogram truncation criteria can be implemented by examining how the linkage merge iterations are shaping the clusters discovered from the bottom up (i.e. from the stronger links toward weaker links). As the iterations progress the number of clusters formed initially grows, then from a critical iteration onward, the structures discovered begin to merge back toward a single cluster. This dynamic can be summarised by monitoring the average cluster size and the number of clusters formed up to a given iteration. The cut off procedure can therefore be implemented via a dual criterion:

- By looking for the last maximum, or last local plateau in the number of clusters as a function of the iteration.
- By applying a reverse elbow procedure to the average cluster size.

The two criteria can also be coupled by taking the ratio, at any iteration, of the average cluster size divided by the number of clusters formed. For simplicity such quantity will be referred to as *dendrogram iteration ratio*. The cutoff procedure then reduces to finding the highest iteration exhibiting a local minimum in the iteration ratio. The only situation in which this idea would fail is in the case of a monotonically increasing dendrogram iteration ratio, that is found when there are no real underlying groups in the data, e.g. the data is a collection of items that do not belong together, and the data is just grouped into an ever growing single cluster that will end up comprising the entire dataset. In this special case the conclusion should be: the feature cannot be split into stereotypes. The complete procedure to create stereotypes for complex categorical features is illustrated in algorithm 1.

4.2 Stereotypes for Numerical Features

In Chapter 3 it was demonstrated how the entire feature set for the movie dataset under examination could be reduced, via features engineering techniques, to numerical features, thus eliminating the need to stereotype categorical features in the current problem.

In order to propose an algorithm for grouping (stereotyping) numerical features, a preliminary examination of the probability distribution of each feature needs to be undertaken using two

Algorithm 1 Algorithm to assemble stereotypes for complex categorical features

Compute the correlation matrix
Compute the average non-diagonal correlation value V

if V is low on average (for example below 0.4) **then**
  use linear dissimilarity
else
  use quadratic dissimilarity
end if

Hierarchically cluster the correlation matrix

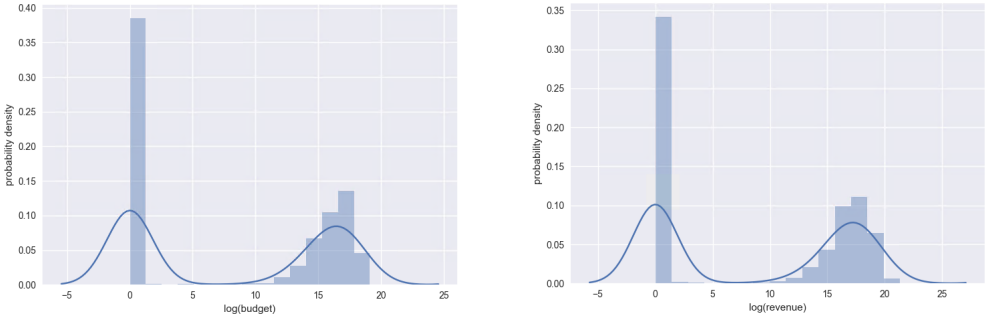
Assemble the dendogram
Compute the dendogram iteration ratio R
Find the highest iteration at which R displays a local minimum

if at least one local minimum **then**
  Assemble the stereotypes by cutting the dendogram between such iteration and the successive one
else
  No stereotype
end if

approximate representations of the individual feature's probability distributions: a standard histogram representation, and a kernel density estimation (KDE) [164] performed using a gaussian kernel.

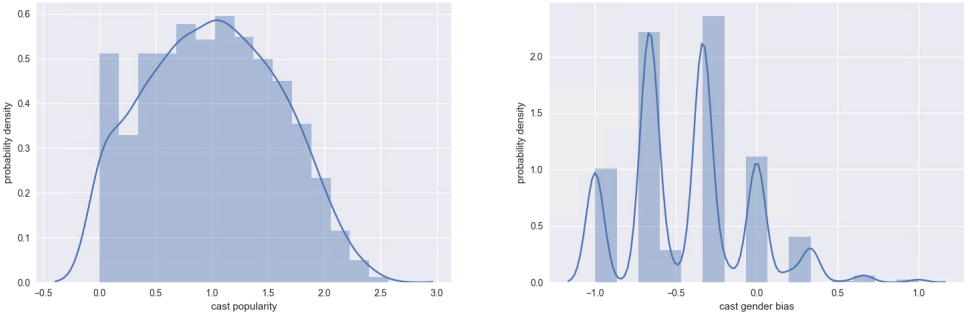
There are several observations that can be drawn by looking at the approximate probability distributions of the 14 features in Figures 4.1 to 4.7. The most important observation is that there is a clear distinction between features whose sample distributions are inherently multimodal, potentially discrete or mixed distributions (e.g. log (budget), log (revenue), cast gender bias, country distance, director popularity, language popularity and release time of year) and the features whose histograms are representative of a continuous potentially mono-modal distribution (e.g. cast popularity, popularity of movie, production company popularity, runtime, release year, vote average and log of vote count).

Among the mixed distributions typical examples are log (budget) and log (revenue), for which unknown budget or revenues are given a discrete value of 0, and then the known budgets/revenues are continuously distributed. There is a small set of features whose histogram or KDE representation is worth examining closer; firstly, language popularity: because of the skewed nature toward 'EN', the histogram and KDE try to represent a distribution that tends toward a delta probability distribution, and the effect of approximating a delta function (or any discontinuities), via gaussian kernels results in the many wiggles in the graph, see [165]. For the very same reason KDE plots are not as meaningful in the case of inherently discrete distribution or distributions exhibiting discontinuities (for example release month), but are very relevant in the case of continuous ones (for example vote average). Secondly, popularity features like for instance director popularity are



(a) log(budget+1) (b) log(revenue+1)

Figure 4.1: Probability density approximation via histograms and KDE for the features: log(budget+1) and log(revenue+1)

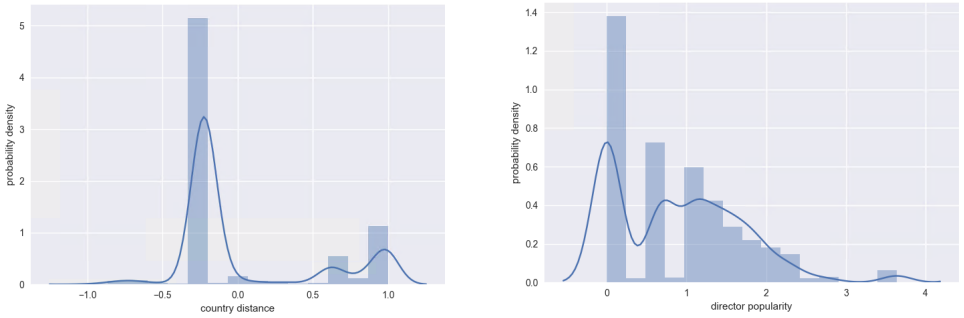


(a) cast popularity (b) cast gender bias

Figure 4.2: Probability density approximation via histograms and KDE for the features: cast popularity and cast gender bias

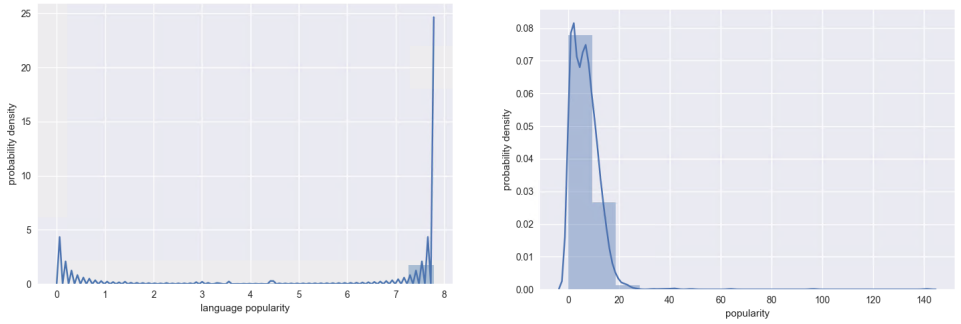
by construction discrete - as the definition of director popularity involves the log of number n of movies directed - however at high $\log(n)$ those could also be approximated as continuous distributions, so one could end up with a mixed distribution: discrete at low popularity and continuous at high popularity.

The multimodal identity of several features suggests a very natural way to create numerical stereotypes for such features: select the most relevant modes and intervals around them. The attractiveness of this definition consists in the fact that it seems operatively simple, and it is indeed the case for features like log(budget+1) or log(revenue+1) where two distinctive groups that are easy to identify would be created, the high budget/revenue items and the ones with unknown budget/revenue. However, the simplicity of the definition is not enough to determine stereotypes in those cases where there are several modes (distinctive peaks in the distribution), and where it is in principle not clear how many of such modes should be relevant (see for example features like production company popularity or cast gender bias).



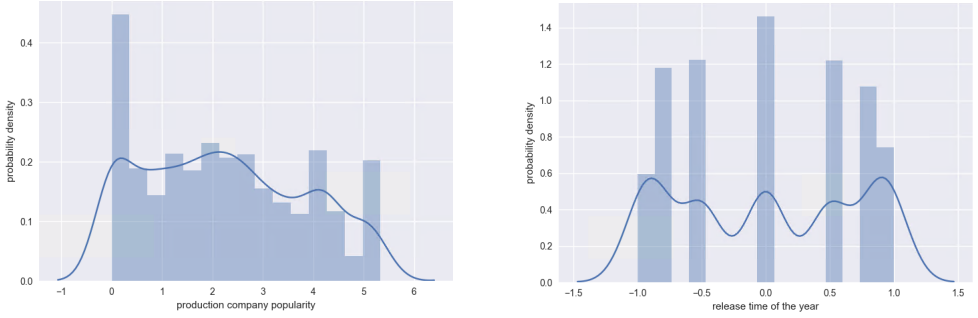
(a) country distance (b) director popularity

Figure 4.3: Probability density approximation via histograms and KDE for the features: country distance and director popularity



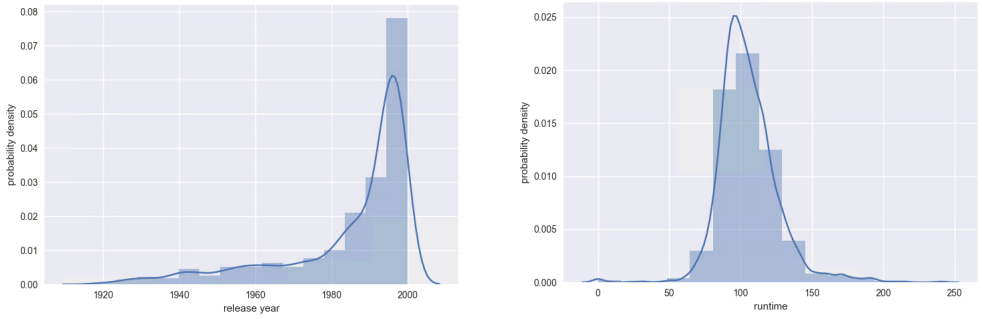
(a) language popularity (b) popularity of the movie

Figure 4.4: Probability density approximation via histograms and KDE for the features: language popularity and popularity of the movie



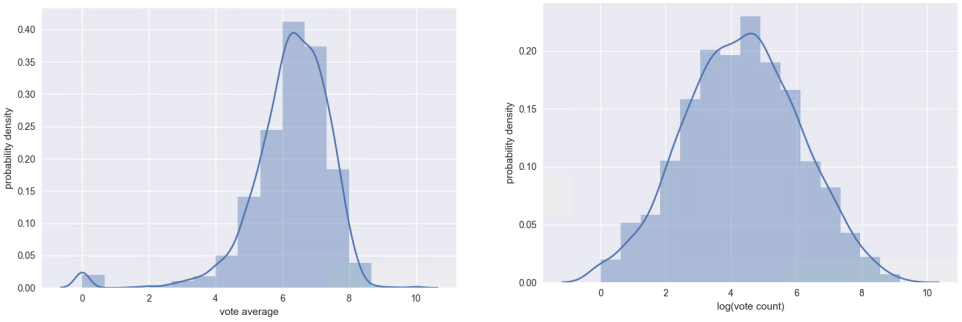
(a) production company popularity (b) release time of the year

Figure 4.5: Probability density approximation via histograms and KDE for the features: production company popularity and release time of the year



(a) release year (b) runtime

Figure 4.6: Probability density approximation via histograms and KDE for the features: release year and runtime



(a) vote average (b) vote count

Figure 4.7: Probability density approximation via histograms and KDE for the features: vote average and vote count

An algorithm is required to automatically 'detect' the peaks in a histogram plot and classify such peaks according to their significance. This problem is not as simple as the problem of discovering local maximums in a function and ranking them, and the reason for this is better explained via an example. Figure 4.8 shows a fictitious probability distribution with 4 local maximums (local modes); if the local maxima of the distribution were identified and ranked simply via their probability density value, i.e. ranking them as (A, B, C, D), the ranks would not be representative of the 'structures' in the data. This is because the shape of the curve around maximum B, even though B constitutes the second highest peak, suggests that it is potentially associated with a noise effect in the left branch of the A peak, or alternatively perhaps by a smaller effect around the large peak A. In either case the structure B would not be as significant when ranking the structures of the distribution, and it would be less dominant than C and D, as the shape of the distribution around C and D seems instead genuinely associated with structures relevant for stereotyping. This example shows that the relative maxima together with their significance (or dominance) must be identified in a stable, noise resilient fashion.

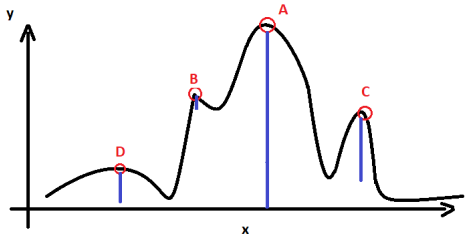


Figure 4.8: A fictitious probability density approximation illustrating the idea behind the ranking of modes

A formal solution to this problem was provided in the mathematical branch of computational topology and particularly in the field of persistent homology [166]. The concept of significance (i.e. persistence) can be used to such a scope. Persistence is better explained with a classic topology example: the function is analogous to a submerged mountain, with an initial water level above the global maximum A. As the level drops, whenever it reaches a local maximum, a new island is born, and whenever it reaches a local minimum, two islands merge (the lower island merges into the higher). The lifespan of an island is correlated to its significance, also called persistence. In Figure 4.8, the persistence of each local maxima is shown via the vertical blue bars, which allows the desired ranking of the local maxima: (A, C, D and B).

The importance of modes, as well as the absence of multiple characteristic modes, can be assessed when the distribution of a feature is computed via a numerical discretisation via binned histograms, as follow: if B bins are used to approximate the distribution, then if the distribution was a uniform pdf each bin would contain 1/B probability density, therefore a meaningful deviation between two neighbouring bins in terms of probability can be defined as 2/B. This constitutes a first order approximation of a minimum 'significance' of the modes.

In addition to the significance of the barcode associated to each mode identified, one additional criterion can be used to define what proportion of the sample is associated with the mode by approximating numerically the area under the curve connected to the specific barcode as depicted in the fictitious example of Figure 4.9. Such an approximate area is representative of the coverage of the mode, and it can be computed via standard Riemann's approximation [167].

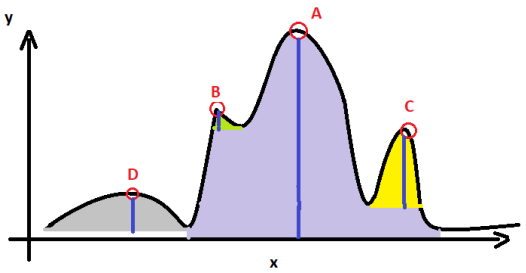


Figure 4.9: The probability proportion that can be associated to each structure (purple for A, green for B, yellow for C and grey for D)

In our study we distinguish numerical features into two categories, **Type I** features are such that the sample distribution has a number of modes with significant barcode greater or equal to two, and the estimated proportion of the population sample that can be attributed to such modes is relevant, i.e. greater or equal to 60%. Features that do not respect such conditions are called **Type II** features and the stereotypes are built using percentile driven intervals (for example quartiles).

4.3 Stereotype Creation Experiment

The integrated ML/IMDb dataset, contains approximately one million movie ratings provided by approximately 6000 users for 4000 movies. A special split procedure is implemented: the data that is left out for evaluation purposes should be able to provide a test environment for both new user and new item. In our experimental set up we select about 2000 users and about 1000 movies randomly, all these users and items (as well as their respective ratings) are left out for testing. The item's metadata is composed by simple numerical features like the movie's budget, the movie's revenue, the runtime as well as simple categorical features like the movie's production company, the production country and the movie's language, and finally some complex categorical features like the movie's genre and the movie's associated keywords.

4.3.1 Results with Complex Categorical Features

To gain an initial understanding and qualitatively validate the grouping in the two large and significant complex categorical features: 'genre' and 'keywords' in the dataset, a greedy grid search algorithm was developed in python to rank possible permutations of columns and rows in the correlation matrix according to the row and column values of function 4.3. This guarantees that the search will find a set of possible permutations that create localised 'zones' in the correlation matrix of minimal penalty, thus highlighting the implied relationships between the categories (as read in the rows and columns). It is important to note that the result of the search for permutations does not lead to a unique result, and the optimisation may lead to several local minima in the penalty where the search converges to.

Figure 4.10 shows the correlation matrix computed over the training set for the entries of feature 'genre'. In such a figure, the order of the categories is 'as encountered', meaning that each category (for example 'Drama' or 'Western') is set at the last position in the queue when it is first met whilst scanning the data. It is clear that the correlation matrix is able to convey important information, already in its unsorted form, by simply looking for large positive and negative associations. For example, one can see how 'Sci-Fi' and 'Science Fiction' are highly correlated, and so are 'Children's' and 'Family'. At the same time the most negatively correlated are for example 'Drama' and 'Science Fiction', or 'Thriller' and 'Comedy'.

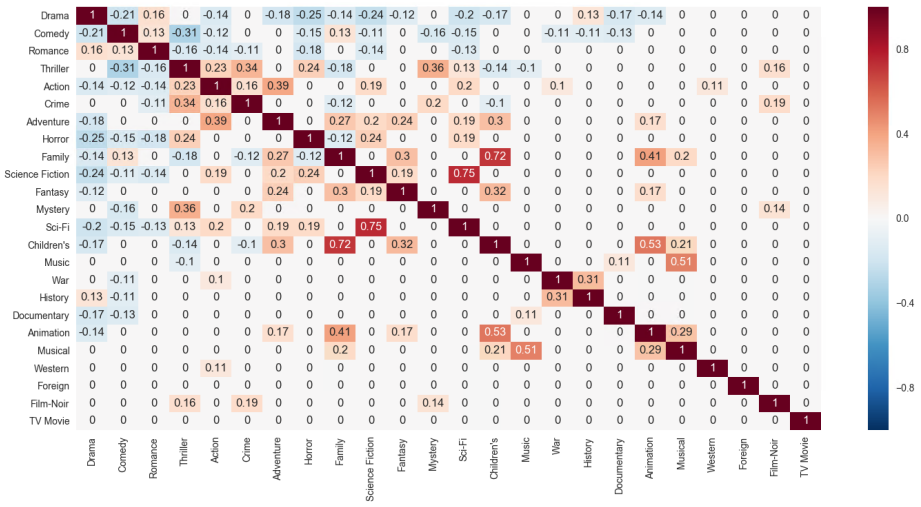


Figure 4.10: Correlation matrix for the genre feature, each category of the genre feature is ordered 'as seen' in the training dataset

However, the shape of Figure 4.10 is sub-optimal for identifying groups of relationships, which are instead easily spotted when the correlation matrix is permuted according to the algorithm discussed previously. When the greedy grid search is applied to row/columns permutations, Figure 4.11 is obtained (this grouping is just for display in the figure and does not affect what follows). This makes it easier to identify 'groups' that can be considered as clusters for that feature only. In the case at hand for the feature 'genre' we can see that the first group is constituted by ('Film-Noir', 'Thriller', 'Crime' and 'Mystery'). A second group is constituted by ('Children's', 'Animation', 'Family', 'Fantasy') and a third group by ('Sci-Fi', 'Science-Fiction', 'Action' and 'Horror') with 'Adventure' linked closely to both the second group and to the third.

Other groups can be identified, like ('Music', 'Musical' and 'Documentary') or ('Drama', 'War','History'), as well as individual categories whose link with others is too small and should be regarded as singletons (see for example 'Westerns', or 'Foreign'). Some singleton like 'Comedy' and 'Romance' do show a small correlation between each other, but this link is, at least on the training set, too weak to trigger grouping.

A similar correlation analysis was performed for the feature 'keywords' by restricting the attention to keywords that appear in the training data a sufficient number of times, in order to avoid fitting any pattern to keywords that are rarely used. In this context, only keywords that were seen more than twenty times each over the items of the training data were retained. Calculating the correlation matrix as defined in formula 4.2 and then applying the row and columns greedy search for permutations that minimise the penalty function 4.3 leads to a grouped depicted in Figure 4.12.

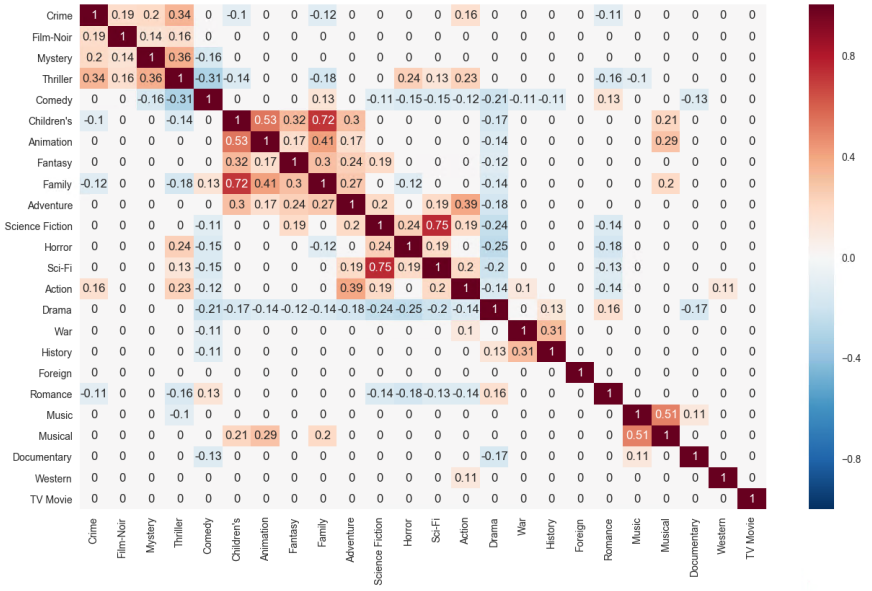


Figure 4.11: Correlation matrix for the genre feature, each category of the genre feature is ordered as suggested by the permutation search for similar 'groups'

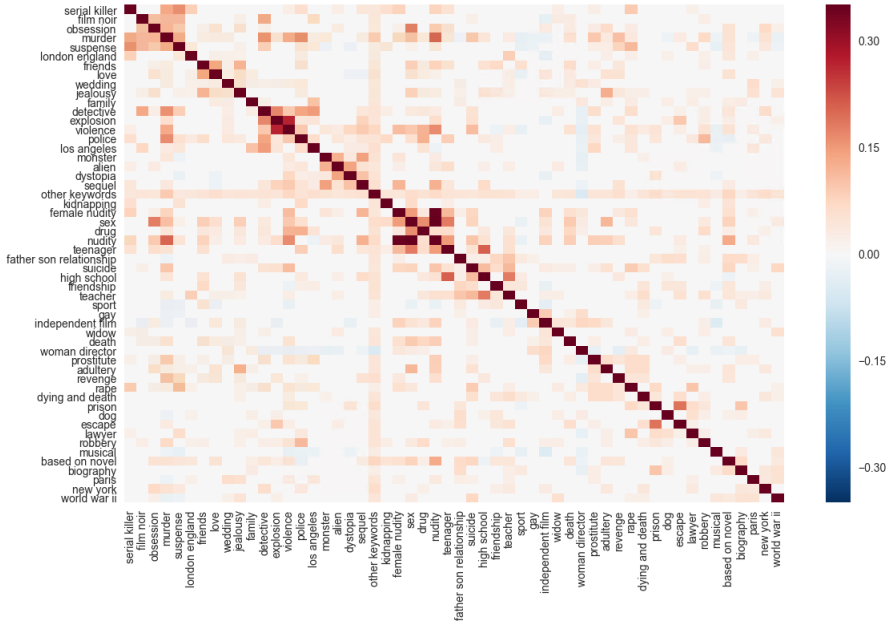


Figure 4.12: Correlation matrix for the keywords feature, each keyword is ordered as suggested by the permutation search for similar 'groups'

One can see how keyword correlations are in general much weaker than for genre, as a result of the vast 'scattering' over hundreds of keywords; by observing Figure 4.12, it is easy to see how very few cases of negative correlation exists between keywords, possibly related again to the vastity of keywords available. Ultimately, thanks to the correlation grouping, it is relatively easy to see how several well-defined groups appears. For example, just to list a few there is a first group in the top left corner that would relate to crime movies ('serial killer', 'film noir', 'obsession', 'murder', 'suspense'), a group that relates to relationships ('friends', 'love', 'wedding', 'jealousy', 'family') and a group that relate to police stories ('detective', 'explosion', 'violence', 'police').

The averages in sample correlation in both genre and keywords are low in absolute value, therefore our hierarchical clustering procedure has been carried out using a linear penalty 4.4 as suggested in Section 4.1. When the metric for dissimilarity 4.4 is applied in a hierarchical clustering algorithm via the three alternative linkages criteria 4.6, 4.7 and 4.8 to the correlation matrix for the feature 'genre' the resulting clustered correlation matrices and dendrograms are shown in Figures 4.13 to 4.15.

Before reviewing the results in the figures, it is necessary to introduce the concepts underlying the dendrogram plots. In a dendrogram the existing hierarchical connections between objects are displayed, the key to interpreting a dendrogram is to look for objects that are connected via branches that splits at a low linkage distance (or dissimilarity) which is read on the y axis. On the x axis one has the labels, or objects. Horizontal lines indicate merges, vertical lines indicate the objects participating in the merge as well as the height, y coordinate, at which the merge happened. The smaller (lower) the y of the split the more related two objects or group of objects are. By following the vertical branches of the dendrogram toward higher linkage distance (or dissimilarity) merges one finds the elements constituting different layers of clusters or groups, where the intensity of the relationship, in other words the strength or connection existing among the objects of the group, is inversely proportional to the height of the common branch from which they arise.

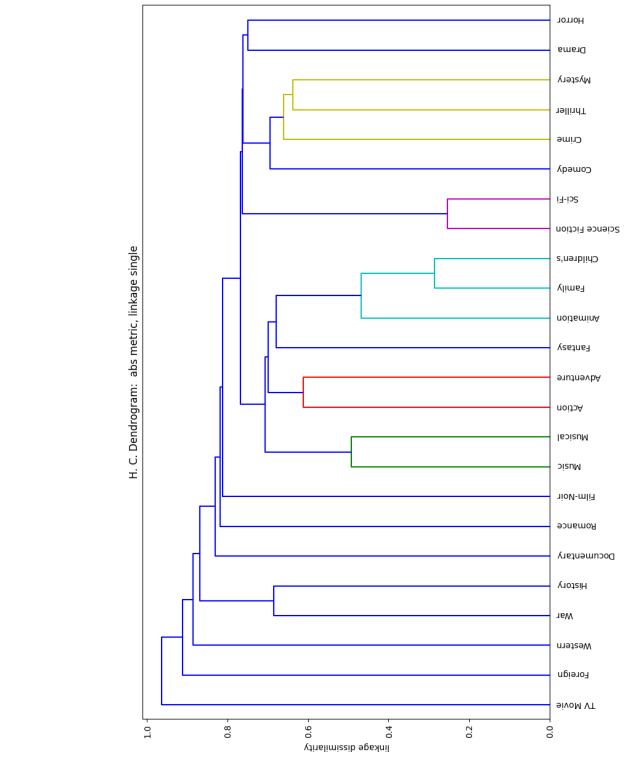
Figure 4.14 can be used as an example to illustrate the information conveyed by a dendrogram - this figure represents the hierarchical linkages for the genre feature using dissimilarity 4.4 and complete linkage 4.7. By beginning at the lowest linkage dissimilarity ($y = 0$) and looking for the lowest merges, one finds that 'Sci-Fi' and 'Science Fiction' merge at a dissimilarity linkage of around 0.2 and 'Children's' and 'Family' merge at a dissimilarity linkage of around 0.25. These are the two stronger merges; other strong mergers can be seen at slightly higher dissimilarities between 'Music' and 'Musical' and also between 'Action' and 'Adventure'. A particularly interesting strong merge is the one that connects the 'Children's' and 'Family' branch met above with the 'Animation' branch. This illustrates how clusters are formed, where one is able to see

more labels following a certain split, as well as how strong the relationship is between the labels below the split. In this case the cluster contains - at the level of the third merge - 'Children's', 'Family', 'Animation' with the link between the first two stronger than their links to the third. Following the same branch up one finds that also 'Fantasy' could be included within the same cluster, if one is happy to consider grouping at higher dissimilarity. At even higher dissimilarity the branch would merge with the 'History' and 'War' branch - again if one is willing to merge at a much higher dissimilarity.

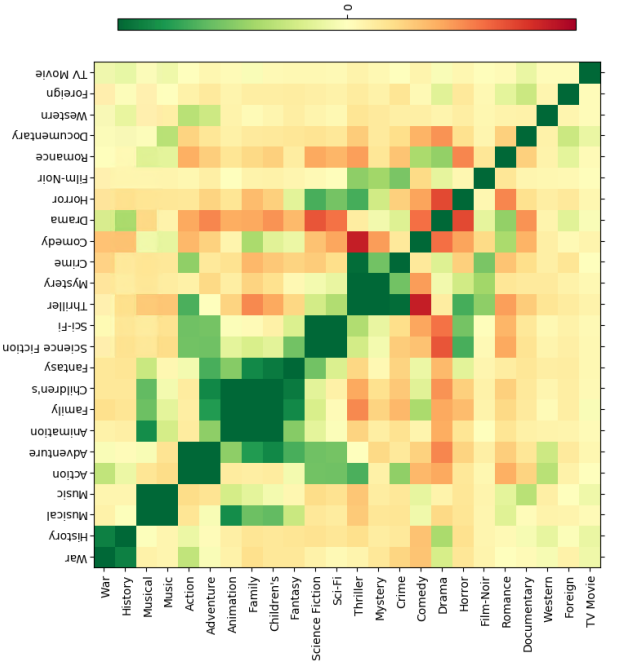
Finally, the last observation regarding how to read the dendrogram concerns the colour of the branches. The colour is just a visual aid in finding branches that merge at low dissimilarity, branches that merge at high dissimilarity are left blue to highlight weaker links.

The logical grouping identified by single linkage 4.6 and dissimilarity 4.4 - Figure 4.13 - seems to be less congruent with a 'human' subjective assignment. As discussed before, a metric could be used to evaluate the hierarchical links discovered, see for example the Fowkles and Mallows metric [168], however such a metric requires the availability of the true labels for the clusters. In addition to the fact that the true labels are not available, even if they were available for complex categorical features like genre and keywords, the true labels would reflect the 'expert opinion' of the operators defining the labels - hence they would be subjective rather than objective. For this very reason, evaluation of the clustered correlation matrices and dendrograms is performed via a subjective judgment.

The groupings provided by the complete linkage 4.7 and Ward linkage 4.8 as shown in Figures 4.14, and 4.15, respectively, are overall very similar. However, the Ward linkage seems to be able to better represent the hierarchical links between different levels of the groupings, at the highest dissimilarity the first split consists of movies that are family friendly versus movies that cater more to an adult audience. Both within the family friendly and within the adult audience subgroups, the genres are further split into sub-groups that can be endorsed by the author's subjective view (see for example the subgroups: 'Fantasy', 'Children's', 'Family', 'Animation' under the family oriented category, and the 'Film-Noir', 'Crime', 'Thriller', 'Mystery' subgroup in the adult audience category).

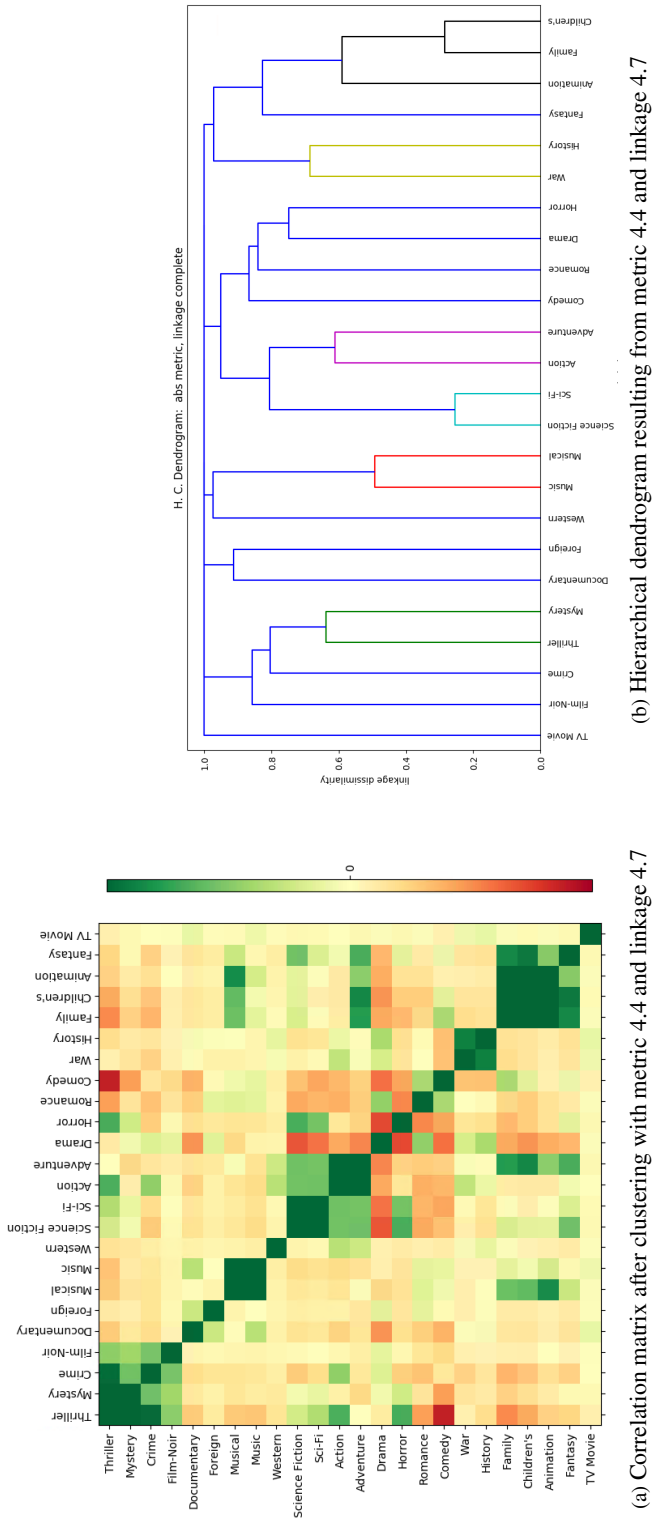


(b) Hierarchical dendrogram resulting from metric 4.4 and linkage 4.6



(a) Correlation matrix after clustering with metric 4.4 and linkage 4.6

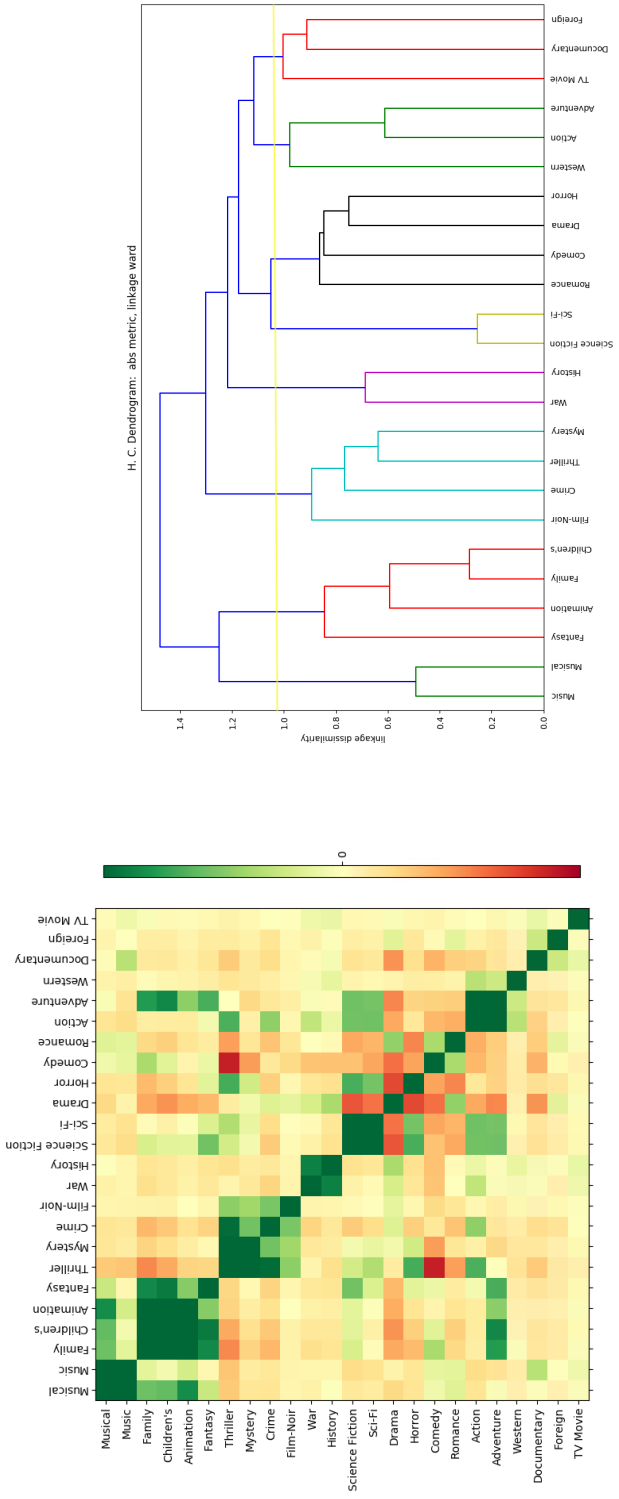
Figure 4.13: Genre grouping result from metric 4.4 and linkage 4.6



(a) Correlation matrix after clustering with metric 4.4 and linkage 4.7

(b) Hierarchical dendrogram resulting from metric 4.4 and linkage 4.7

Figure 4.14: Genre grouping result from metric 4.4 and linkage 4.7



(a) Correlation matrix after clustering with metric 4.4 and linkage 4.8

(b) Hierarchical dendrogram resulting from metric 4.4 and linkage 4.8

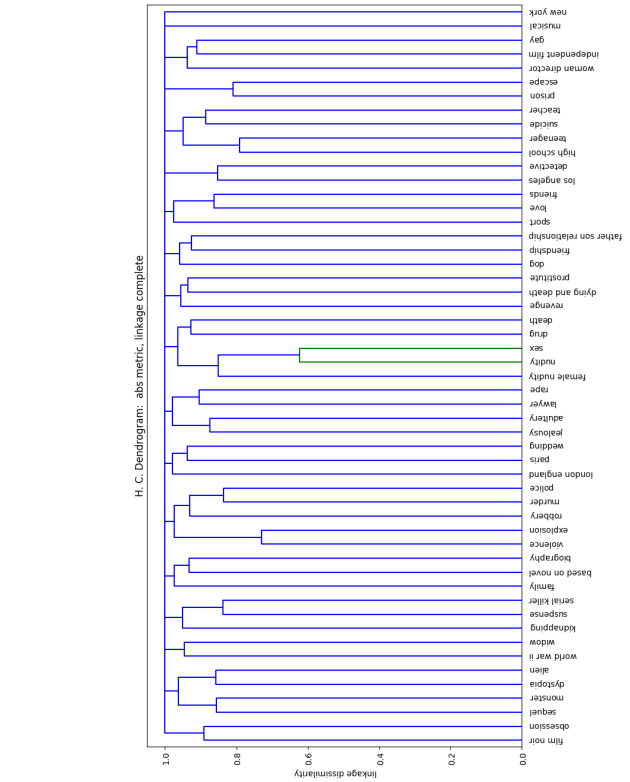
Figure 4.15: Genre grouping result from metric 4.4 and linkage 4.8

The following stage is to apply the same hierarchical clustering and dendrogram study of the linkage dissimilarity splits methodology to the other complex categorical feature of the sample data under investigation, the keywords feature.

Given that it was observed that the complete linkage, equation 4.7, and the Ward linkage, equation 4.8, generate better distinctions and grouping criteria the attention is focused on such methodologies for the analysis of the keyword feature. Figure 4.16 displays the clustered correlation matrix and the hierarchical dendrogram of the most frequent keywords using dissimilarity metric 4.4 and the complete linkage criterium 4.7, while Figure 4.17 displays the clustered correlation matrix and dendrogram for the most frequent keywords using the same dissimilarity but the Ward linkage criterium 4.8. The set of most frequent keywords was defined based on the number of counts of each keyword as in Section 4.3.1.

As it was found for the feature genre, the two linkages form very similar groups across keywords, and such groups are plausible when considered from a human lexical based prospective (see for example the group: 'sequel', 'monster', 'dystopia', 'alien' or the group 'violence', 'explosion', 'robbery', 'murder', 'police'). The Ward linkage appears again to be more suited in defining a vertical separation of the logical links in the dendrogram. This characteristic is extremely important when considering possible different levels for cutting automatically a dendrogram as a way to create logical groups within the categories of a complex categorical feature.

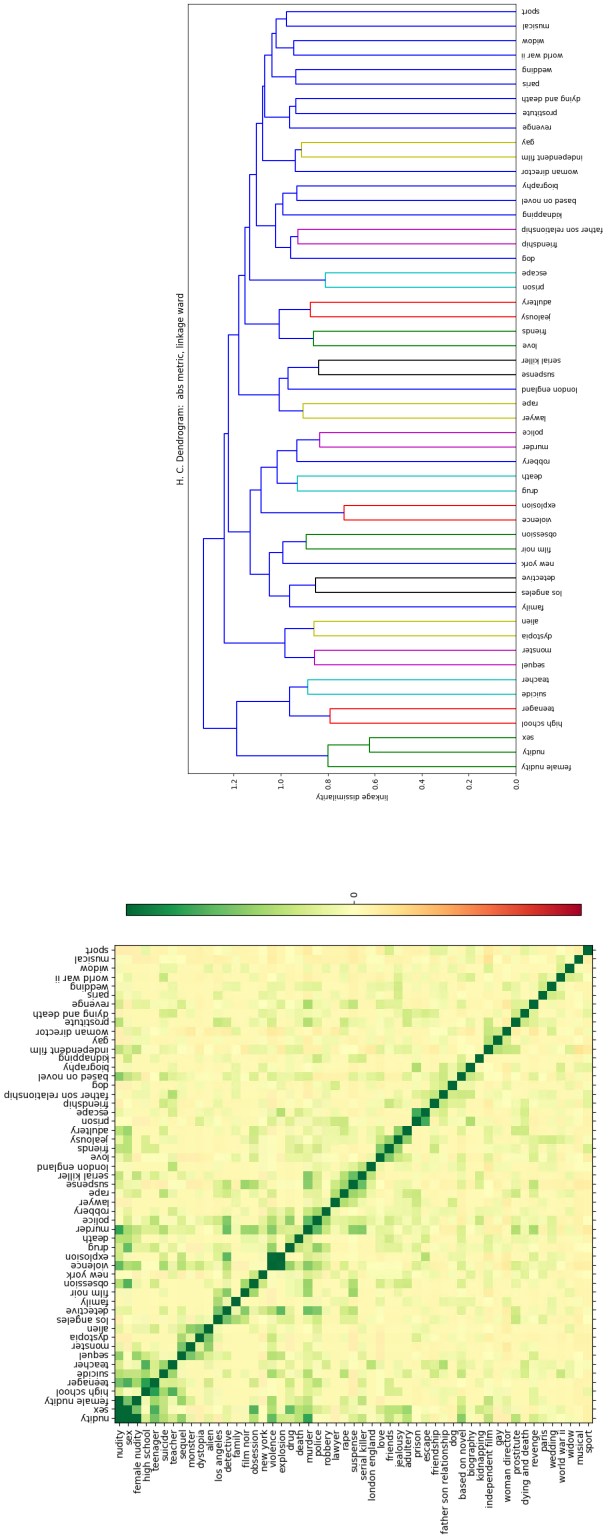
So far, it was demonstrated that performing the hierarchical clustering of the correlation matrix using the Ward linkage and the linear penalty as suggested in Section 4.1 leads to a sound way of obtaining grouping (stereotypes). Following the algorithm 1 to decide at which height of the dissimilarity linkage, one should truncate the dendrogram, leads to the average cluster size over all clusters formed up to a given iteration, as shown for the feature genre in Figure 4.18, and the number of clusters formed up to a given iteration in Figure 4.19. Figure 4.20 shows the resulting dendogram iteration ratio, with the highlighted iteration number where the algorithm discussed in Section 4.1 is used to cut the dendogram. The application of the algorithm for genre and keywords leads to the stereotypes listed in Table 4.1.



(a) Correlation matrix after clustering with metric 4.4 and linkage 4.7

(b) Hierarchical dendrogram resulting from metric 4.4 and linkage 4.7

Figure 4.16: Keywords grouping result from metric 4.4 and linkage 4.7



(a) Correlation matrix after clustering with metric 4.4 and linkage 4.8

(b) Hierarchical dendrogram resulting from metric 4.4 and linkage 4.8

Figure 4.17: Keywords grouping result from metric 4.4 and linkage 4.8

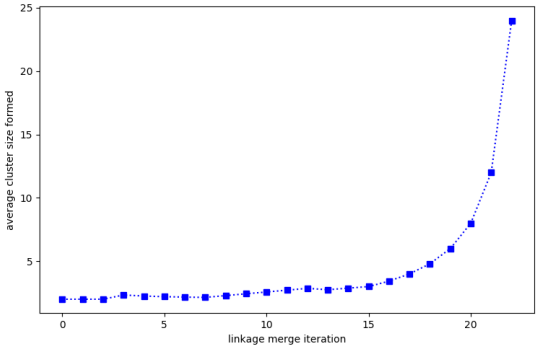


Figure 4.18: Genre feature hierarchical cluster of the correlation matrix, assembly iterations. Average cluster size over the clusters formed at each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward

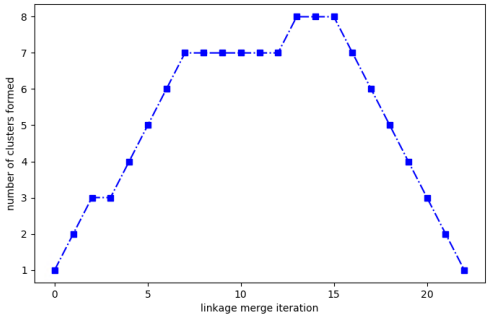


Figure 4.19: Genre feature hierarchical cluster of the correlation matrix, assembly iterations. Total number of clusters present after each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward

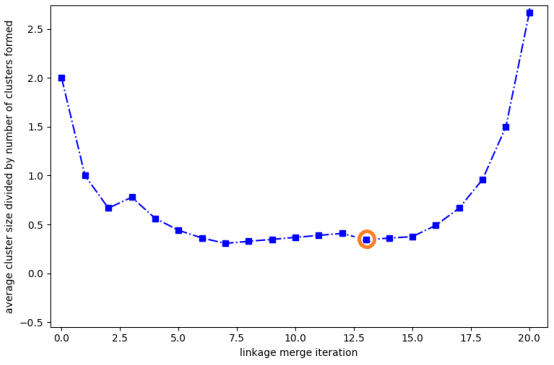


Figure 4.20: Genre feature hierarchical cluster of the correlation matrix, assembly iterations. Dendrogram iteration ratio using dissimilarity metric 4.4, linkage criterium Ward. The red circle indicates the local minimum which is most to the right

	Stereotypes - Genre (Order not relevant)	Stereotypes - Keywords (Order not relevant)
1	[‘Music’,‘Musical’]	[‘Violence’,‘Explosion’, ‘Shootout’, ‘Flashback’]
2	[‘Fantasy’,‘Animation’,‘Family’,‘Children’s’]	[‘Nudity’,‘Sex’,‘Female nudity’]
3	[‘Action’,‘Adventure’,‘Western’]	[‘Drug’,‘Death’,‘Murder’,‘Police’,‘Robbery’]
4	[‘War’,‘History’]	[‘Lawyer’,‘Rape’]
5	[‘TV Movie’, ‘Documentary’,‘Foreign’]	[‘Film noir’, ‘Obession’,‘Suspense’]
6	[‘Film Noir’,‘Crime’,‘Thriller’,‘Mystery’]	[‘Blood’,‘Murder’,‘Prostitute’,‘Revenge’]
7	[‘Romance’,‘Comedy’,‘Drama’,‘Horror’]	[‘High School’,‘Teenager’,‘Party’]
8	[‘Science Fiction’,‘Sci-Fi’]	[‘Independent film’,‘Gay’,‘Woman director’]
9		[‘Hostage’,‘Kidnapping’]
10		[‘Love’,‘Friends’,‘Sports’]
11		[‘Sequel’,‘Monster’,‘Dystopia’,‘Alien’]
12		[‘Cat’,‘Father son relationship’,‘Dog’]
13		[‘Los angeles’,‘Detective’,‘Corruption’, ‘New York’]
14		[‘World war ii’,‘Biography’, ‘Journalist’]
15		[‘Prison’,‘Escape’]
16		[‘After credits stinger’,‘During credits stinger’]
17		[‘Money’,‘London England’, ‘Paris’, ‘Wedding’]
18		[‘Musical’,‘Based on play or musical’]
19		[‘Jealousy’,‘Adultery’, ‘Divorce’]
20		[‘Serial killer’,‘Slasher’]
21		[‘Widow’,‘Small town’]
22		[‘Drug’,‘Police’,‘Robbery’]
23		[‘Suicide’,‘Teacher’,‘Dying and death’]
24		[‘Death’,‘Investigation’]
25		[‘Daughter’,‘Mother and daughter relationship’, ‘Family’]

Table 4.1: Stereotypes automatically generated using algorithm 1 for the feature: genre and keywords

Comparison with other Clustering Algorithms

In this section, the clustering results (Table 4.1) of the automatic stereotype algorithm 1 are compared with the results of the categorical clustering algorithm k-modes [133]. k-modes is a clustering algorithm that is widely used in the literature. In order to create meaningful and useful stereotypes in the context of a recommender system, we are interested in an algorithm which is capable of grouping *all* the labels of the categorical feature under examination in stereotypes, without at priory excluding any labels.

The k-modes clustering algorithm can be initialised in different ways, for example following Zhexue Huang [133] the artefacts (e.g. the localisation of the centroids) are placed in a random manner in feature space, or following Cao et al. [169], who suggested the artefacts to be placed in feature space based on initial density/frequency estimations. Once the method is initialised, the k-modes clustering implementation used tries to minimise a cost function, defined as the sum of all distances for each point of the cluster artefact it belongs to. Distance for categorical variables is defined based on a simple delta function dissimilarity measure as described in [133].

Let X, Y be two categorical objects described by m categorical attributes. The dissimilarity measure between X, Y can be defined by the total mismatches of the corresponding attributes

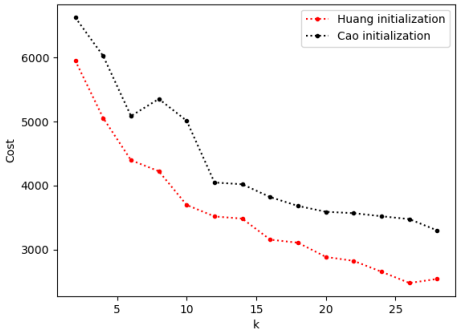


Figure 4.21: Inverse elbow methodology applied to the k-modes clustering of the genre feature with two alternative methodologies for the initialisation of the position of the artefact centroids

categories of the two objects. Formally,

$$d_1(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j) \tag{4.9}$$

where

$$\delta(x_j, y_j) = \begin{cases} 0 & (x_j = y_j) \\ 1 & (x_j \neq y_j) \end{cases} \tag{4.10}$$

In a similar fashion as k-means, k-modes is a partitioning algorithm, meaning it does not discover how many structures are present in the data, but rather it partitions the data in a number of pre-specified clusters. An inverse elbow methodology, based on a cost function to be minimised rather than a score or silhouette function that needs to be maximised, can be applied to the cost function of k-modes following closely what was discussed previously in Section 3.4 for k-means. Figure 4.21 displays the inverse elbow graph for the genre feature partitioned via k-modes with both Huang and Cao initialisations. Both cost functions decay with a lower rate of decay as the number of clusters, k, increases. However, it is not straightforward to identify a single well-defined kink in the decay graphs, and for this reason the results of the k-modes clustering for the genre feature is closely inspected by looking at the centroid characteristics at (k= 5) and (k= 10).

Tables 4.2 and 4.3 display, for (k= 5) and (k= 10) respectively, the centroid compositions for the genre feature and both initialisation methods (Huang and Cao). It is important to note that the ordering of the cluster centroids in the table is not relevant, i.e. centroid n.1 is not more nor less important than centroid n.4. The most interesting finding by observing Tables 4.2 and 4.3 is that k-modes seems to easily allow for centroids which substantially overlap on several sub-categories, for example for both (k= 5) and (k= 10) categories like ‘Drama’, ‘Comedy’, ‘Thriller’, ‘Action’, ‘Family’ and ‘Adventure’ appear several times in more than one centroid.

	Centroid Composition (Huang) (Order not relevant)	Centroid Composition (Cao) (Order not relevant)
1	[‘Drama’,‘Comedy’,‘Romance’]	[‘Family’,‘Children’s’,‘Animation’]
2	[‘Drama’]	[‘Drama’]
3	[‘Adventure’,‘Family’,‘Children’s’,‘Animation’]	[‘Comedy’,‘Adventure’,‘Family’,‘Fantasy’,‘Children’s’]
4	[‘Comedy’]	[‘Comedy’]
5	[‘Thriller’,‘Action’]	[‘Thriller’,‘Action’,‘Adventure’,‘Science Fiction’,‘Sci-Fi’]

Table 4.2: k-modes resulting centroids composition for 5 clusters and the genre feature, with two alternative methodologies for the initialisation of the position of the centroids

	Centroid Composition (Huang) (Order not relevant)	Centroid Composition (Cao) (Order not relevant)
1	[‘Drama’]	[‘Drama’]
2	[‘Comedy’]	[‘Comedy’,‘Action’,‘Adventure’,‘Science Fiction’,‘Fantasy’]
3	[‘Adventure’,‘Family’,‘Children’s’,‘Animation’]	[‘Family’,‘Children’s’,‘Animation’,‘Musical’]
4	[‘Action’,‘Adventure’,‘Science Fiction’,‘Sci-Fi’]	[‘Thriller’,‘Action’,‘Adventure’,‘Science Fiction’,‘Sci-Fi’]
5	[‘Thriller’,‘Action’]	[‘Comedy’,‘Music’,‘Musical’]
6	[‘Drama’,‘Romance’]	[‘Comedy’,‘Adventure’,‘Family’,‘Children’s’,‘Animation’]
7	[‘Comedy’,‘Family’,‘Children’s’]	[‘Comedy’,‘Family’,‘Children’s’]
8	[‘Thriller’,‘Crime’]	[‘Thriller’,‘Action’,‘Crime’]
9	[‘Drama’,‘Thriller’,‘Mystery’]	[‘Thriller’,‘Horror’,‘Mystery’]
10	[‘Drama’,‘Comedy’,‘Romance’]	[‘Adventure’,‘Family’,‘Fantasy’,‘Children’s’]

Table 4.3: k-modes resulting centroids composition for 10 clusters and the genre feature, with two alternative methodologies for the initialisation of the position of the centroids

It is legit to wonder why certain categories are not at all represented in the centroids of Tables 4.2 and 4.3, see for instance ‘Documentary’, ‘Western’, ‘Film-Noir’ or even ‘Foreign’. For this question the answer lies in the training dataset, using the dissimilarity measure adopted by the k-modes algorithms leads, by construction, to substantial frequency-based overweighting of the categories in centroids, which might also explain the recurrent presence of the most frequent categories as well as the lack of presence of less frequent categories. However, we argue that these labels should indeed be retained as they may represent specific niche of users preferences, and are required in the recommendation items coordinates as shown later in the research.

A similar study of the reverse ‘elbow’ is presented for the keywords feature in Figure 4.22. It is striking to note that whilst the cost function obtained by Huang’s initialisation methodology displays a reverse elbow, even if a weak one at around (k= 20), that is not the case for the Cao initialisation procedure.

A look at the composition of the cluster centroids for (k= 20) demonstrates that, for both initialisation procedures, the presence of ‘other keywords’ (to represent any keyword that does not pass the filter of at least twenty occurrences as previously discussed) become a dominating presence in almost all centroids. In fact, all centroids of this analysis result to be such that they were either just composed of ‘other keywords’ or composed by one of the keywords that can be seen in Figure 4.12 and ‘other keywords’, hence invalidating any conclusions that could be extracted.

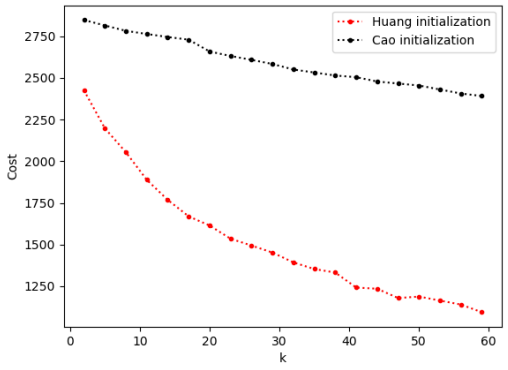


Figure 4.22: Inverse elbow methodology applied to the k-modes clustering of the keywords feature with two alternative methodologies for the initialisation of the position of the artefact centroids

	Centroid Composition (Huang) (Order not relevant)
1	['Revenge']
2	['Father son relationship']
3	['Sequel']
4	['Women director','Friends']
5	['Drug']
6	['Sport']
7	['Female Nudity','Nudity']
8	['Friendship']
9	['Paris']
10	['Murder']
11	['Suspense']
12	['Prison']
13	['Alien']
14	['Monster']
15	['Robbery']
16	['Gay']
17	['London England']
18	['Musical']
19	['Sport']
20	['Death']

Table 4.4: Centroid composition identified by k-modes for the feature keywords with Huang initialisation methodology and k= 20

A repeat of the k-modes clustering removing the 'other keywords' entry, and focusing only on the keywords that are most frequent in the training data, leads to identical inverse elbow plots as in Figure 4.22. The analysis of the centroids for (k= 20) and Huang's initialisation reveals that (see Table 4.4) almost all centroids but two are constituted by a *single* keyword. Overall this finding makes the application of k-modes in this context questionable. The algorithm seems to be very sensitive to the initial conditions chosen, with the centroids changing substantially as new random artefacts are used.

Our empirical experience led us to favour our suggested algorithm 1 over k-mode for the stereotype construction of complex categorical features.

4.3.2 Results with Numerical Features

The concept of persistence and barcode calculation suggested in Section 4.2 was implemented in Python for a one-dimensional real valued sequence via an iterative search process based on sorting the sequence and applying the concept of island birth, death based on jumps vs continuity of the indices. The application of the search for persistence to the features described in Figures 4.1 to 4.7 leads to the results in the Tables 4.5 to 4.18. The tables present for each feature the modes that satisfy the barcode criterion (greater than 2/B) and for such modes the respective estimation of the area obtained via the Riemann's approximation, displayed in the third column, as an estimate of the sample population that can be attributed to the mode under exam. Given that our numerical discretisation of the probability denisity function is done using between 20 and 40 bins, we can estimate that a jitter of +/- 2.5% is in our case the limit between signal and noise, and therefore we disregard as not significative all modes associated to a population of less than 4%. Hence, two criteria that needs to be satisfied for a mode to be considered significative. The barcode needs to be greater than the threshold dictated by the number of bins used in the discretisation; and the area of population that can be attributed to such barcode needs to be also significative based on the numerical approximation.

The algorithm successfully finds and ranks the modes in the desired order, this can be observed for example by looking at some of the features like 'cast gender bias' or 'normalised country distance' (Figures 4.2 and 4.3 vs Tables 4.11 and 4.8). However, the algorithm also discovers some modes that are not significant and that are most likely associated with either noise in the sample (and/or in the numerical discretisation) or are associated with irrelevant structures.

For some examples of structures whose persistence is associated to noise or irrelevant structures one can look at the following: Table 4.17, extremely long movies above 200 minutes in runtime (probably a genuine but irrelevant structure) versus movies of 0 minutes (induced by movies whose runtime is not available); or Table 4.15, movies that have received a vote near to 10 (probably a genuine but irrelevant structure) versus movies that have received a 0 vote (also a genuine structure which mixes poorly liked movie and movies that are yet to receive a vote).

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
0.000	0.4903	0.52
16.096	0.1481	0.42
4.024	0.0013	not significative

Table 4.5: Filtered modes discovered and ranked via the persistence algorithm for feature: log(budget+1)

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
0.000	0.4861	0.51
16.844	0.1414	0.39
2.246	0.0013	not significative

Table 4.6: Filtered modes discovered and ranked via the persistence algorithm for feature: log(revenue+1)

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
0.000	0.3338	0.19
0.574	0.1692	0.13
0.957	0.1389	0.32
1.340	0.0955	not significative
3.446	0.0156	not significative
2.297	0.0063	not significative

Table 4.7: Filtered modes discovered and ranked via the persistence algorithm for feature: director popularity

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
-0.261	0.6856	0.72
0.895	0.1389	0.14
0.580	0.0644	not significative
-0.787	0.0194	not significative

Table 4.8: Filter modes discovered and ranked via the persistence algorithm for feature: country distance

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
0.946	0.0833	0.17
0.000	0.0400	not significative
0.676	0.0076	not significative
1.217	0.0004	not significative

Table 4.9: Filtered modes discovered and ranked via the persistence algorithm for feature: cast popularity

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
7	0.9107	0.91
4	0.0295	not significative
3	0.0206	not significative

Table 4.10: Filter modes discovered and ranked via the persistence algorithm for feature: language popularity

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
-0.684	0.3241	0.31
-0.368	0.2971	0.27
-0.053	0.1486	0.15
-1.000	0.1347	0.13
0.263	0.0417	not significative
0.579	0.0093	not significative

Table 4.11: Filter modes discovered and ranked via the persistence algorithm for feature: cast gender bias

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
0.000	0.5989	0.68
37.092	0.0007	not significative
59.347	0.0006	not significative
89.021	0.0004	not significative
133.532	0.0004	not significative

Table 4.12: Filtered modes discovered and ranked via the persistence algorithm for feature: popularity of movie

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
0.000	0.1473	0.38
5.052	0.0362	not significative
0.561	0.0332	not significative
4.210	0.0194	not significative

Table 4.13: Filtered modes discovered and ranked via the persistence algorithm for feature: production company popularity

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
-0.053	0.1949	0.19
-0.579	0.1633	0.16
0.474	0.1625	0.16
-0.895	0.1574	0.24
0.789	0.1435	0.21

Table 4.14: Filtered modes discovered and ranked via the persistence algorithm for feature: release time of the year

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
5.789	0.2243	0.89
0.000	0.0130	not significative
9.474	0.0004	not significative

Table 4.15: Filtered modes discovered and ranked via the persistence algorithm for feature: vote average

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
1995	0.3519	0.78
1982	0.0168	not significative
1969	0.0028	not significative

Table 4.16: Filtered modes discovered and ranked via the persistence algorithm for feature: release year

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
89.158	0.4861	0.94
0.000	0.0029	not significative
229.263	0.0008	not significative

Table 4.17: Filtered modes discovered and ranked via the persistence algorithm for feature: runtime

Feature Mode Location	Persistence Barcode (Probability)	Population %attributable to the Mode
4.347	0.1094	0.99

Table 4.18: Filtered modes discovered and ranked via the persistence algorithm for feature: log(vote count)

Table 4.19 summarises the features that can be categorised as belonging to the Type I versus those belonging to Type II based on the algorithm discussed in Section 4.2. For features whose distribution is not multimodal (Type II), and also for features of Type I where a zone of the distribution is 'too wide' and still attributable to a single mode (for example the log (budget) area of mode 16.84 contains 42% of the population sample and it spans several orders of magnitude in budget), stereotypes can be introduced via percentile driven intervals as discussed.

Percentile interval driven stereotypes can be obtained by identifying the lower and upper bounds of the feature value corresponding to a given percent level of the population (for example 20% or 25%). The features of Type II, which by definition do not exhibit clear bumps in their distributions, hence clusters, will be stereotyped via the percentile driven approach, the same approach can be applied also to portions of Type I features when needed as discussed previously. Tables 4.20 to 4.33 present the stereotypes that are obtained with the methodology illustrated. For each table the column relative to the mode location is provided only for stereotypes that had been extracted using the modes discovered for features of Type I. The mode location is omitted for areas of the distribution that are stereotyped not by mode plus or minus interval but by a percentile driven interval.

Type I Features	Type II features
Log (budget+1)	Language Popularity
Log(revenue+1)	Popularity (of movie)
Director Popularity	Production company popularity
Country Distance	Vote average
Release Time of Year	Release Year
Cast Gender Bias	Runtime
	Log (vote count)
	Cast Popularity

Table 4.19: Classification of numerical features between Type I (stereotyping can be done via the modes) and Types II (stereotyping can be done via percentile intervals)

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	0.000	0.000	1.000	0.52
2	-	1.000	16.096	0.21
3	-	16.096	∞	0.21

Table 4.20: Stereotypes for feature: log(budget+1)

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	0.000	0.000	1.000	0.51
2	-	1.000	16.844	0.22
3	-	16.844	∞	0.22

Table 4.21: Stereotypes for feature: log(revenue+1)

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	0.000	0.000	0.286	0.23
2	0.574	0.286	0.765	0.15
3	0.957	0.765	∞	0.41

Table 4.22: Stereotypes for feature: director popularity

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-0.261	-∞	0.000	0.78
2	0.895	0.000	∞	0.19

Table 4.23: Stereotypes for feature: country distance

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0.000	0.555	0.25
2	-	0.555	1.011	0.25
3	-	1.011	1.456	0.25
4	-	1.456	∞	0.25

Table 4.24: Stereotypes for feature: cast popularity

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	7	6.9	7.1	0.91
2	-	0	6.9	0.09

Table 4.25: Stereotypes for feature: language popularity

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-0.684	-0.857	-0.432	0.31
2	-0.368	-0.432	-0.226	0.27
3	-0.053	-0.226	+∞	0.25
4	-1.000	-∞	-0.857	0.13

Table 4.26: Stereotypes for feature: cast gender bias

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0.000	2.500	0.25
2	-	2.500	6.204	0.25
3	-	6.204	9.770	0.25
4	-	9.770	$+\infty$	0.25

Table 4.27: Stereotypes for feature: popularity of movie

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0.000	0.895	0.25
2	-	0.895	2.117	0.25
3	-	2.117	3.518	0.25
4	-	3.518	$+\infty$	0.25

Table 4.28: Stereotypes for feature: production company popularity

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-0.053	-0.1	0.1	0.2
2	-0.579	-0.675	-0.1	0.16
3	0.474	0.1	0.628	0.16
4	-0.895	-1.000	-0.675	0.24
5	0.789	0.628	1.000	0.21

Table 4.29: Stereotypes for feature: release time of the year

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0.000	5.7	0.25
2	-	5.7	6.4	0.25
3	-	6.4	7.3	0.25
4	-	7.3	10	0.25

Table 4.30: Stereotypes for feature: vote average

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	1918	1980	0.25
2	-	1980	1993	0.25
3	-	1993	1998	0.25
4	-	1998	2007	0.25

Table 4.31: Stereotypes for feature: release year

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0	93	0.25
2	-	93	102	0.25
3	-	102	117	0.25
4	-	117	300	0.25

Table 4.32: Stereotypes for feature: runtime

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0	3.135	0.25
2	-	3.135	4.356	0.25
3	-	4.356	5.568	0.25
4	-	5.568	10	0.25

Table 4.33: Stereotypes for feature: log(vote count)

4.4 Summary

The problem of complex and rich item data is made more challenging, but at the same time more representative of real-world applications of stereotypes to recommender systems, by the fact that the most important categorical variables are multi-entry categorical in a non-strict sense. By that we mean that the entry of an item for a given categorical features, genre for example, is not simply one category, e.g. 'drama', but it can take several entries without any preassigned number, e.g. for one item the genre may be categorised as 'drama', for another item it may be 'drama' plus 'romance' plus 'historic'.

Multi-entry categorical features do not just bring an extra level of complexity, they also enable the researcher to gain an understanding of the intrinsic relationships between categories, relationships that come directly from the process that generated the data. In this chapter, we have proposed an algorithm for the automatic identification of stereotypes for both complex categorical and numerical features. The approach proposed consists in clustering and grouping generic metadata features (based on their type) to provide new metadata features that can be viewed as a stereotypical representation of the original data. The work in this chapter address the research question:

Can item-based stereotypes, not based on rating, be constructed automatically?

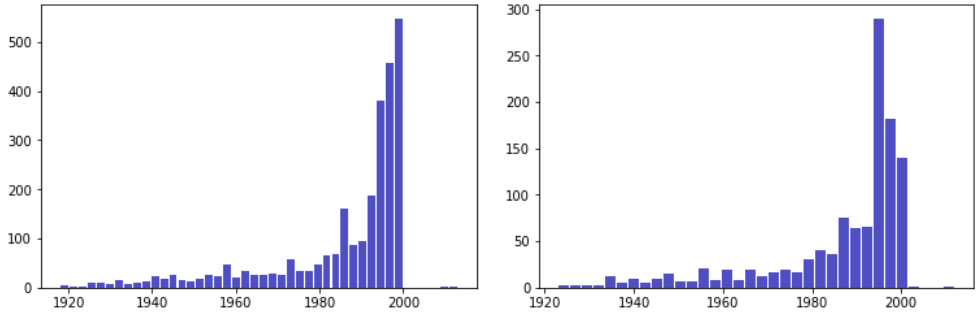
Before discussing how the automatic generated stereotypes can be used in recommender system, it is important to evaluate the stereotypes obtained for their stability and accuracy, which will be carried out in the next chapter.

CHAPTER 5

Preliminary Evaluation of Stereotypes

In Chapter 4, we proposed an algorithm for the automatic identification of stereotypes for both categorical and numerical features. Before discussing how stereotypes can be used in a recommender system, it is important to investigate possible ways to evaluate the stereotypes. It is of paramount importance to check that the results obtained during the stereotype creation phase are stable and truly representative of real data associations. Previously it was also discussed how, given that the problem of assembling stereotypes is an unsupervised learning problem, the absence of 'true' labels makes the evaluation of the appropriateness of the stereotypes a difficult and some what subjective task.

Upon inspection of the stereotypes generated in the previous chapter, it was easy to agree that the great majority of associations discovered by the algorithm were in line with our subjective judgment. Nevertheless, it is possible to take the evaluation of the stereotypes one step further by checking their stability and consistency on out-of-sample data. For this scope all the data used up to this stage to perform calculations and derivation of the stereotypes can be labelled as training data, and new unseen data, labelled as test data, can be used to evaluate consistency and stability of the relationships discovered. In this chapter, we first validate training/test homogeneity in Section 5.1. Then Section 5.2 formulate three different statistical tests for stereotype evaluation. The presentation that follows has been formulated for this thesis, it is believed that the structures discovered should be examined via appropriate tests before being used in a RS. This approach and the tests carried out can be viewed as an element of novelty arising from the present research.



(a) Training set (b) Test set

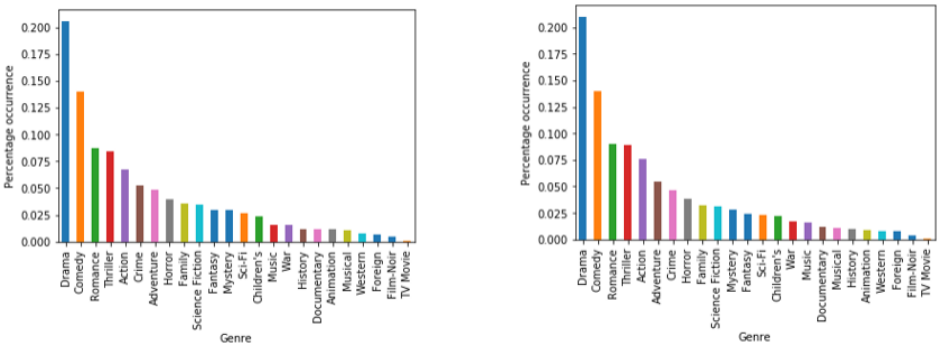
Figure 5.1: Histogram distribution of the movie's production year

5.1 Homogeneity of Training and Test Datasets

For each stereotype created on the in-sample data, a number of statistical tests can be formulated to evaluate its stability, accuracy and predictive content. However, the first step to perform in order to proceed with the training/test validation is to confirm that the data is somewhat 'homogeneous', i.e. the movies contained in the unseen test data - albeit different movies from those contained in the training data - can be thought of as being extracted from the same unknown population as that of the training data.

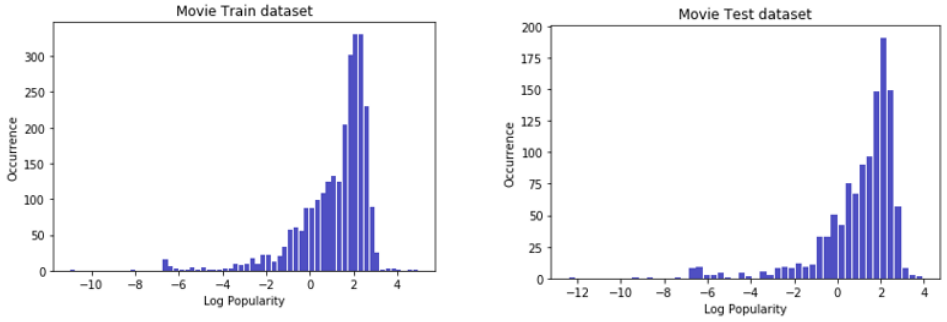
The two simplest driving features that can be checked to confirm or disprove qualitatively the statement that the test data can be seen as representative of the training data are the item (movie) year distribution, and the item (movie) genre distribution. If the distribution of the year of production of movies in the training and the test sets is very different, for example if the training movies were all movies prior to 1975 and the test movie all movies after 1975 it would be possible to observe differences in the stereotypes just as a by-product of the evolution of taste and association of genres. For example, 'Horror-Comedy' would be an easy association to find in the 90s but not so in the 50s. Therefore, several preliminary checks can be performed to confirm whether or not there is ground to believe that conclusions drawn on one dataset may not be applicable to the other. Figures 5.1 display the distribution of the movie's production year for the two sets. The first thing to notice is that statistically the test data, albeit of a smaller size than the train data, still contains enough samples to draw reasonable conclusions: 1149 for the test set versus 2678 for the training set used up to this point.

From Figure 5.1 it is also possible to observe that qualitatively the movies in the two sets have a similar distribution over the years, with the only major difference being that the most frequent movie production years in the test dataset are 1995 to 1997 whereas for the training dataset are 1998 to 2000. This difference should not be enough to invalidate the analyses that will follow, and we can conclude that the two datasets have a similar spectrum of movies over the years.



(a) Training set (b) Test set

Figure 5.2: Histogram distribution of the movie's genres percentage occurrence



(a) Training set (b) Test set

Figure 5.3: Histogram distribution of the movie's popularity feature

The next check performed consists in verifying that the distribution of genres for the movies in the two datasets is not dissimilar to the point of invalidating the ability to extend conclusions drawn on one set to the other set. Figures 5.2 display the percentage of occurrence of each genre label in the movies of both training and test datasets. It is possible to observe how the test dataset is indeed very much representative of the training dataset, with almost a perfect relative match in frequency and order of the most important labels. For instance, one needs to look at the 6th and 7th label in order of relative occurrence to find the first label mismatch between the two datasets, in the training case the 6th category is 'Crime' followed by 'Adventure', and in the test data these two features are swapped in terms of importance.

In theory, all other features should be checked in terms of their distribution similarity in order to make sure that the two datasets can be thought of as homogeneous as possible. For example Figure 5.3 displays the histogram distribution of the movie popularity feature (via the log transformation to compress a large range of popularities). We can see how also the popularities have a very similar distribution in their occurrence between the two sets, with the larger corpus

of log popularities occurring between the values of 0 and 3, and the most frequent log popularity equal to 2 for both sets.

We can observe that the test dataset is indeed representative of the training dataset, and can be used to infer conclusions about the algorithms under investigation.

5.2 Stereotypes Evaluation

There are two logical ways to check in the unsupervised learning situation at hand whether the relationships that have been learnt by the algorithm (in this context the stereotypes) are truly representative of the population: a hard test and a soft test.

5.2.1 A Hard Test

The most severe test is to check if on 'unseen' data - a test datasets - the same relationships as those learnt from the training dataset would be discovered in an independent fashion; and then measure how far apart the relationships learnt by the algorithm from both sets are. This test is an extremely severe test, in the sense that it relies on homogeneity between the two datasets. There is an extra reason, which applies especially to keywords, that makes the hard test extremely severe. Given the very large number of keywords, in principle there is a staggering amount of possible ways in which 'keywords' (but also genres) can combine, making any true relationship extremely noisy and potentially weak.

Complex Categorical Features

The hard test consists of comparing the stereotypes obtained over the training data with the stereotypes obtained independently over the test data. For complex categorical features, this is carried out scoring how close two sets of labels are to each other, by looking at one minus the ratio of the number of labels in the set difference between the stereotype examined and the reference stereotype to the total labels of the reference group; this provides a measure of precision.

A classical simple example from statistics can be used to illustrate the above test: suppose you are interested in measuring the mean of a population, to do so you collect first a training sample of data and successively a smaller test set of data. The hard test proposed for stereotypes can be thought of being akin to estimating the mean of the population from both datasets, and then comparing and measuring how different the means are.

The mathematical formulation of the hard test can be obtained by considering the composition of the stereotypes generated on the training and on the test data. Let S_i^T be the i-th stereotype generated on the training data and composed by a certain number of n_i labels, and let the S_j^t be the j-th stereotype generated on the test data and composed by a certain number of m_j labels. Given that stereotypes are an un-ordered set, and that the stereotypes obtained using the training data may in principle differ from those obtained using the test data, a procedure to find the best match between stereotypes obtained in the two sets is required. For any S_i^T a simple search is performed to find the stereotype S_j^t that shares the highest number of common components

(labels). For example, suppose that over some fictitious training data the stereotypes identified were:
$S_1^T = [a, b, c]$,
$S_2^T = [d, e, f]$,
$S_3^T = [g, h]$;
and suppose that the stereotypes identified over the test data were instead:
$S_1^t = [a, b, d]$,
$S_2^t = [g, h]$,
$S_3^t = [c, e, f]$.
Then the search would find that the maximum number of common labels is obtained when S_1^T is matched with S_1^t, S_2^T with S_3^t and S_3^T with S_2^t, with only the last match being a perfect match of the two stereotypes. How well the stereotypes obtained on the test data are representative of the stereotypes on the training data can be analysed using a metric for the degree of similarity of the matches discovered. For each matched pair of stereotypes (S_i^T , S_j^t) having a count of matching labels equal to e_{ij} the similarity metric can be defined as:

$$\mu_{(S_i^T, S_j^t)} = \frac{e_{i,j}}{max(n_i, m_j)} \tag{5.1}$$

In the case of a perfect match ($e_{ij} = n_i = m_j$), the metric 5.1 will be 1.0; when the stereotypes identified are not identical then the μ_{ij} will be less than 1.0. The statistics of the μ_{ij} will provide an objective way to evaluate the stereotypes.

In order to perform the hard test for the feature: **genre**, the procedure described in algorithm 1 (i.e. the algorithm to discover stereotypes in complex categorical features) is applied to the feature 'genre' for the items in the test data. Figure 5.4 displays the dendrogram for the feature genre over the test dataset. A qualitative comparison with the corresponding Figure 4.15 which was obtained for the training dataset reveals striking similarities: the strongest merges happen in the same order (SciFi-Science Fiction), (Children's-Family), (Music-Musical), (Action-Adventure). And also, the first non-simple merge is the same ((Children's-Family)-Animation). Looking for dissimilarities one can observe how the genre Documentary is connected TV-Movie and Foreign on the training data, whilst on the test data is most connected to the Comedy-Romance-Drama-Horror branch. Also, the genres Western and Fantasy locate in different branches of the dendrograms in the training and test datasets.

Figure 5.5 shows the dendrogram iteration ratio, i.e. the metric that allows one to define the iteration at which to cut the dendrogram automatically in order to generate the stereotypes; Figure 5.5 can be viewed as the corresponding of Figure 4.20, but whilst Figure 4.20 performs the iterations on the training data, Figure 5.5 performs them on the test data (two disjoint sets). A striking similarity between the two graphs can be readily observed, with the algorithm identifying the same iteration (iteration 13) as the highest iteration with a local minimum, therefore cutting

the dendrogram at the same iteration (14) in both cases. The resulting stereotype elaborated on the test data only are reported in Table 5.1 and can be compared directly to those discovered over the training data in Table 4.1.

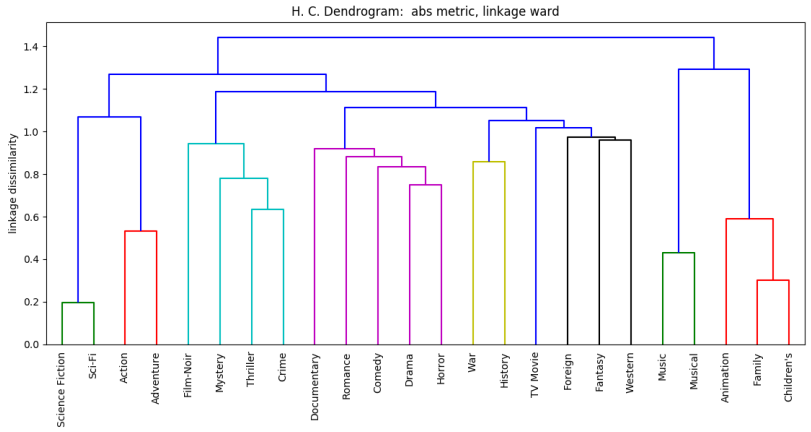


Figure 5.4: Hierarchical dendrogram resulting from the abs metric and the Ward linkage for the genre feature over the test dataset

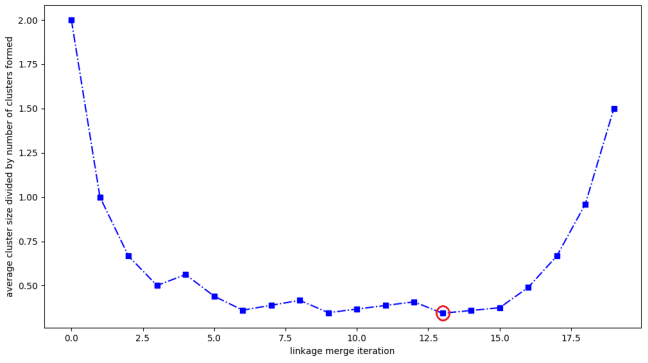


Figure 5.5: Genre feature hierarchical cluster of the correlation matrix of the test dataset, assembly iterations. Dendrogram iteration ratio using dissimilarity metric 4.4, linkage criterium Ward. The red circle indicates the local minimum which is most to the right

The objective analysis of the differences between the two sets of stereotypes (those of Table 4.1 and Table 5.1 constitutes the main result of the hard test and can be performed using the metric introduced in equation 5.1. Table 5.2 reports the results of the comparison between the 8 stereotypes for genre: there is an 88% accuracy (average match), with a median match of 100% - indicating that in most of the stereotypes there is a perfect one-to-one match between the stereotype composition derived on the training data vs those derive on the test data.

	Stereotypes - Genre - Test Data
1	[‘Music’,‘Musical’]
2	[‘Animation’,‘Family’,‘Children’s’]
3	[‘Action’,‘Adventure’]
4	[‘War’,‘History’]
5	[‘TV Movie’,‘Foreign’,‘Fantasy’,‘Western’]
6	[‘Film Noir’,‘Crime’,‘Thriller’,‘Mystery’]
7	[‘Romance’,‘Comedy’,‘Drama’,‘Horror’,‘Documentary’]
8	[‘Science Fiction’,‘Sci-Fi’]

Table 5.1: Stereotypes automatically generated using algorithm 1 for the feature genre over the test dataset only

Descriptive Statistics of μ_{ij}	**Value**
Average (Accuracy)	88%
Median	100%
Minimum	66.6%

Table 5.2: Hard Test comparison statistics for the stereotypes of the feature genre generated over the training vs the test datasets

The first problem that is encountered when trying to perform the hard test for the evaluation of the stereotypes of the feature **keywords** is the fact that the feature is constituted by a much sparser set of values than genre; for this reason, the keywords satisfying the occurrence filter in the test data may not necessarily be the same as those that were identified in the training data. The way to proceed in this case is to generate first the list of keywords that fulfil either the filter condition over the training data or the filter condition over the test data (the aggregate set of keywords will be referred to as the enlarged keywords), and then assemble the stereotypes for the training data with the enlarged set of keywords, and finally proceed with the hard test. Table 5.3 shows the stereotypes for the feature keywords computed using the enlarged set of keywords, the keywords highlighted are the ones that have been added by meeting the filter condition over the test data; it can be noted how such keywords often embed in already existing stereotypes and in other situations their very existence changes slightly the shape of the older stereotypes.

The right column of Table 5.3 shows the stereotypes obtained using the set of enlarged keywords over the test dataset. It can be noted by comparing both columns in Table 5.3 that in the case of the test data there is one more stereotype than in the case of training. The objective analysis of the differences between the stereotypes sets is performed using the metric introduced in equation 5.1. Table 5.4 reports the results of the comparison, there is a 65.3% accuracy (average match), with a median match of 66%. This means that both on average and in median, $\frac{2}{3}$ of the composition of a stereotype is being matched - i.e. two labels out of three. This is an excellent and surprising result, given:

- The sparsity of the keywords features,
- The fact that the training and test datasets are completely disjoint, and

	Stereotypes - Keywords - Training Data	Stereotypes - Keywords - Test Data
1	[Violence, Explosion,Shootout, Flashback]	[Explosion,Shootout, Violence, World war ii]
2	[Nudity, Sex, Female nudity]	[Nudity, Female nudity, Sex]
3	[Based on play or musical, Musical]	[Woman director, Musical]
4	[Lawyer, Rape]	[Friends, Teacher]
5	[Film noir, Obsession, Suspense]	[Detective, Obsession, Suspense]
6	[Blood, Murder, Prostitute, Revenge]	[Adultery, Prostitute, Jealousy, Dying and death]
7	[High school, Teenager, Party]	[High school, Teenager]
8	[Independent film, Gay, Woman director]	[Gay, Based on Play or Musical]
9	[Based on Novel, Extramarital affair]	[Rape, Extramarital affair, Based on novel]
10	[Jealousy, Adultery,Divorce]	[Revenge, Flashback]
11	[Sequel, Monster, Dystopia, Alien]	[Film noir, small town]
12	[Cat, Father son relationship, Dog]	[Cat, Dog, Father son relationship]
13	[Los angeles, Detective, Corruption, New York]	[Corruption, Police, Investigation]
14	[World war ii, Biography, Journalist]	[Lawyer, Biography]
15	[Prison, Escape]	[Prison, Escape, Los Angeles, Dystopia]
16	[Love, Friends, Sport]	[Love, Alien]
17	[After credits stinger, During credits stinger]	[After credits stinger, During credits stinger]
18	[Serial killer, Slasher]	[Serial killer, Monster, Sequel,Slasher]
19	[Drug, Police, Robbery]	[Drug, Independent film]
20	[Death, Investigation]	[New York, Journalist]
21	[Suicide, Teacher, Dying and death]	[Paris, London England, Robbery]
22	[Money, London England, Paris, Wedding]	[Money, Party, Suicide, Sport]
23	[Hostage, Kidnapping]	[Hostage, Kidnapping, Murder, Violence, Blood]
24	[Widow, Small town]	[Widow, Death]
25	[Daughter, Mother daughter relationship, Family]	[Daughter, Mother daughter relationship, Wedding]
26		[Family, Divorce]

Table 5.3: Stereotypes for the keywords feature using the training dataset and the list of enlarged keywords are shown on the left. The keywords highlighted are those that meet the filter condition in the test dataset. The right column show the stereotypes for the keywords using test dataset

Descriptive Statistics of μ_{ij}	Value
Average (Accuracy)	65.3%
Median	66.6%
Minimum	33%

Table 5.4: Comparison statistics for the stereotypes of the feature keywords generated over the test vs the training datasets

- The large feature dimension is in principle leading to a much higher potential number of combinations of the labels, reducing the probability of obtaining matches.

Numerical Features

The evaluation of the numerical stereotypes can be conducted in a similar fashion to that performed for the complex categorical stereotypes. The hard test requires a measure of similarity between the stereotypes that are meant to represent the same group of the population, in the present analysis such stereotypes are generated independently from two non-overlapping data-

sets for any given feature. For this reason, the similarity/dissimilarity will need to quantify firstly whether or not the number of stereotypes discovered over the training data, and then independently over the test data, match before attempting a comparison between pairs of stereotypes.

Figure 5.6 provides an illustration of the type of differences between stereotypes that one needs to quantify in the case of numerical features; the figure depicts the imaginary sample distribution of a feature, the distribution above obtained from a different sample for the same feature than the one below. Using the procedure discussed in Section 4.2 the stereotypes are discovered as intervals of the distribution. One can represent each interval via its 'centre' and probability importance. The dots in the figure represent the 'centres' of each stereotype, the size of each dot is proportional to the probability 'mass' that the stereotype represents and, given the analogy between probability density and mass density, the location of the dots can be placed at the centre of mass of the stereotype (which is analogous to the definition of the expected value of stereotype interval):

$$X^c_{sj} = \int_{S^L_j}^{S^U_j} x\, p(x)\, dx \tag{5.2}$$

$$P^c_{sj} = \int_{S^L_j}^{S^U_j} p(x)\, dx \tag{5.3}$$

Where $p(x)$ represents the probability density of the given feature on the values x. X^c_{sj} is the centre of the j-th stereotype spanning from the below limit of S^L_j to the upper limit of S^U_j, and it can be thought of as the single best point for a given stereotype that one could select if one had to replace the entire stereotype with a single value of the feature. P^c_{sj} represents the probability mass associated to the stereotype.

As seen from the figure, there may be several types of discrepancies arising when comparing numerical stereotypes via the hard test:

- An additional stereotype discovered using one set of data but not the other (example of the brown dot to the right),

- Stereotypes whose probability 'mass' are very close but their centres locations are off (example of the red dots),

- Situations where the centres may be not identical but also the probability of the masses are different (example of the green dots).

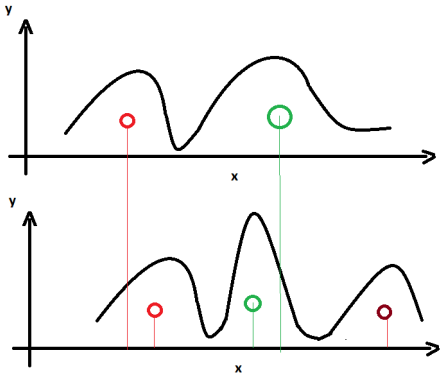


Figure 5.6: Fictitious comparison of numerical stereotypes set up. Stereotypes mass, centre of mass and numbers are identified by the dots

In terms of percentage accuracy, for each stereotype of a given feature, firstly one needs to pair stereotypes between the two distinct runs (note that in the example of Figure 5.6 there would be two pairs and one single stereotype), then the following formula can be used to evaluate the accuracy of the pair:

$$
\begin{aligned}
\delta P_{sj}^{T,t} &= |P_{sj}^{Train} - P_{sj}^{test}| \\
\delta X_{sj}^{T,t} &= |X_{sj}^{Train} - X_{sj}^{test}| \\
\mu_{sj}^{T,t} &= \begin{cases} \left(1 - \frac{\delta P_{sj}^{T,t}}{0.5(P_{sj}^{Train}+P_{sj}^{test})}\right)\left(1 - \frac{\delta X_{sj}^{T,t}}{0.5(X_{sj}^{Train}+X_{sj}^{test})}\right) \\ \\ 0 \quad \text{if no stereotype can be paired} \end{cases}
\end{aligned}
\tag{5.4}
$$

Where the first two equations of relationships 5.4 define the deltas existing between the probability masses and the probability centres respectively, and the third expression defines a percentage accuracy between the two stereotypes obtained via the train and test datasets. It is easy to verify that in case of 0 deltas the formula will give a 1 (100%) accuracy. As soon as either the probability masses or the probability centres exhibit deviations the accuracy will be less than 1. In the limiting case of a stereotype generated in the training or test datasets that cannot be put into correspondence with another then the accuracy will be at 0 (lower branch of the third equation).

Table 5.5 displays the quantities of the set of relationships 5.4 for all stereotypes of all numerical features (features ordered alphabetically). For almost all the features the number of stereotypes discovered over the training and over the test data match; only for the feature 'log revenue' there is one less stereotype discovered in the test data over the training data, and the effect for the non-paired stereotype is reflected in the entry with 0 accuracy. Of all the thirteen

Feature Name		$\delta P_{sj}^{T,t}$	$\delta X_{sj}^{T,t}$	$\mu_{sj}^{T,t}$	**Statistics**
cast gender bias	1	0.01	0.0	0.92	mean accuracy: 0.93 minimum accuracy: 0.86 stdv accuracy: 0.05
cast gender bias	2	0.0	0.0	1.00	
cast gender bias	3	0.02	0.0	0.94	
cast gender bias	4	0.01	0.01	0.86	
cast popularity	1	0.0	0.21	0.23	mean accuracy: 0.43 minimum accuracy: 0.23 stdv accuracy: 0.15
cast popularity	2	0.0	0.32	0.39	
cast popularity	3	0.0	0.42	0.55	
cast popularity	4	0.0	0.63	0.58	
country distance	1	0.0	0.03	0.89	mean accuracy: 0.94 minimum accuracy: 0.89 stdv accuracy: 0.05
country distance	2	0.0	0.01	0.98	
director popularity	1	0.18	0.0	0.58	mean accuracy: 0.66 minimum accuracy: 0.58 stdv accuracy: 0.09
director popularity	2	0.08	0.03	0.61	
director popularity	3	0.03	0.12	0.79	
log (budget)	1	0.02	0.0	0.96	mean accuracy: 0.84 minimum accuracy: 0.75 stdv accuracy: 0.08
log (budget)	2	0.05	1.11	0.75	
log (budget)	3	0.023	1.56	0.83	
log (revenue)	1	0.02	0.03	0.74	mean accuracy: 0.52 minimum accuracy: 0.0 stdv accuracy: 0.38
log (revenue)	2	0.10	0.78	0.82	
log (revenue)	3	Nan	Nan	0.0	
log (vote count)	1	0.0	0.08	0.95	mean accuracy: 0.96 minimum accuracy: 0.94 stdv accuracy: 0.02
log (vote count)	2	0.0	0.19	0.94	
log (vote count)	3	0.0	0.11	0.98	
log (vote count)	4	0.0	0.14	0.97	
popularity	1	0.0	0.08	1.0	mean accuracy: 0.96 minimum accuracy: 0.94 stdv accuracy: 0.02
popularity	2	0.0	0.17	0.96	
popularity	3	0.0	0.27	0.95	
popularity	4	0.0	0.45	0.94	
Prod. comp. popularity	1	0.0	0.08	0.48	mean accuracy: 0.66 minimum accuracy: 0.48 stdv accuracy: 0.12
Prod. comp. popularity	2	0.0	0.36	0.62	
Prod. comp. popularity	3	0.0	0.69	0.70	
Prod. comp. popularity	4	0.0	0.9	0.77	
release time of the year	1	0.0	0.0	1.0	mean accuracy: 0.94 minimum accuracy: 0.79 stdv accuracy: 0.08
release time of the year	2	0.0	0.0	1.0	
release time of the year	3	0.0	0.0	1.0	
release time of the year	4	0.01	0.07	0.79	
release year	1	0.0	2.93	0.98	mean accuracy: 0.99 minimum accuracy: 0.98 stdv accuracy: 0.01
release year	2	0.0	1.23	0.99	
release year	3	0.0	0.55	0.99	
release year	4	0.0	0.0	1.0	
runtime	1	0.0	0.35	0.99	mean accuracy: 0.96 minimum accuracy: 0.88 stdv accuracy: 0.04
runtime	2	0.0	0.75	0.99	
runtime	3	0.0	1.13	0.98	
runtime	4	0.0	15.14	0.88	
vote average	1	0.0	0.06	0.99	mean accuracy: 0.99 minimum accuracy: 0.99 stdv accuracy: 0.005
vote average	2	0.0	0.0	1.0	
vote average	3	0.0	0.0	1.0	
vote average	4	0.0	0.03	0.99	

Table 5.5: Hard test metrics for the evaluation of the numerical stereotypes. The table reports for all features and all stereotypes the dissimilarity in probability mass and centre, and the accuracy from relationships 5.4. The last column gives standard metric of the accuracy

numerical features, nine of them have an hard test resulting average accuracy of over 80%. For the features whose average stereotypes accuracy is below 80%, one can see that only the feature 'cast popularity' overall is characterised by low accuracy across all stereotypes. For 'log revenue' the average is brought down by the unpaired stereotype, the two paired stereotypes do indeed exhibit an accuracy in the region of 80%. The fact that the features with the lowest accuracy are those in the group of popularity derived features (cast popularity, director popularity, production company popularity) put some doubts on to whether the 'popularity' transformation suggested earlier may be responsible for the lower - on average - accuracy of the stereotypes derived on two independent datasets. This fact should be investigated further, and potentially the definitions of popularities revised.

Excluding this finding, that affects three features out of thirteen, for the remaining features the numerical stereotypes are very stable (both in terms of probability 'mass' and in terms of their numerical ranges and hence their centres) therefore making them truly representative of the items population.

5.2.2 A Soft Test

The soft test is a less stringent test and it is perhaps more representative of the use that stereotypes have in a recommender system. In this test, the test data is first used to create aggregate clusters together with the training data, and 'true' labels (coordinates in the stereotypes space) are created for each of the elements of the test data. Then the stereotypes that had been defined over the training set are used to label the items in the test data, therefore obtaining another view of the labels for each point. The statistical metrics of the errors between the two stereotypical representations obtained for each test item can be used for evaluation purposes of the training stereotypes. This test is a precursory analysis of the validity in the use of stereotypes in a recommender system: if the stereotypes learnt on the training dataset prove to be able to give item representations which hold if one 'knew' the true substereotypical representation of the item, it would mean that the relationship learnt can indeed be used to provide recommendations.

Complex Categorical Features

The formulation of the soft test consists of transforming the unsupervised learning problem into a problem that resembles a supervised learning one: first the Train dataset (T) and test dataset (t) are assembled together in a dataset called Full dataset (F). For a given complex categorical feature the stereotyping identification procedure is repeated for the Full dataset (F) obtaining a set of stereotypes S_i^F; using these stereotypes S_i^F it is possible to derive, for each item in the test dataset (t), the degree of belonging of each movie to each stereotype. For example, suppose the fictitious stereotypes were:
$S_1^F = [a, b, c]$,
$S_2^F = [d, e, f]$,
$S_3^F = [g, h]$;

then a test item having as labels (a,b,d) would have [2,1,0] as a vector of degrees of belonging to each stereotype. For each item in the test data, the vector representing the degrees of belonging to each of the S_i^F stereotypes can be regarded as the 'true' stereotypical representation of the item. The degrees of belonging vector representation can be obtained for all items in the test data also using the stereotypes that were discovered in the training data only. For item p, let $\underline{v}^{(F)}(p)$ be its degree of belonging representation to the stereotypes extracted from the Full dataset (F) and let $\underline{v}^{(T)}(p)$ be the degree of belonging representation of the item to the stereotypes extracted from the training dataset (T); a metric can be established by looking at the vector difference: $\underline{v}^{(F)} - \underline{v}^{(T)}$.

It is important to observe that in order to be able to take the difference between the two degree of belonging vectors, the two vectors must be of the same size, and given that the two sets of stereotypes are not ordered, they also need to be rearranged so that the degree of belonging can be compared along the same stereotype's coordinates. These two steps are performed as follow: first the vectors are sorted guaranteeing that there is as close as possible match between the F and T stereotypes at a given index in the vectors. The match is defined via the number of labels overlapping between each of the F and T stereotypes, a minimum of 50% of matching labels is taken as a pre-requisite to declare two stereotypes as corresponding. When there is no direct correspondence, i.e. when a pair of F, T stereotypes cannot be linked between each other because the minimum match between labels is not meeting the required minimum, then the dimension of the vectors is augmented adding both coordinates as a penalty. An example using fictitious data can help understand the sorting and resizing procedure; let us suppose that the stereotypes discovered over the Full data (F) were:
$S_1^F = [a, b, c]$,
$S_2^F = [d, e, f]$,
$S_3^F = [g, h, i]$;
and those discovered in the training data were:
$S_1^T = [c, d, e]$,
$S_2^T = [a, b]$,
$S_3^T = [f, h]$,
$S_4^T = [g, i]$.
The sorting and matching goes as follow: the most similar stereotype to S_1^F is S_2^T with two out of three matching labels; the most similar stereotype to S_2^F is S_1^T with two out of three matching labels. The most similar to S_3^F is S_4^T and S_3^T does not have a corresponding stereotype and it is left last as a dummy coordinate for F stereotypes.

Suppose item A is characterised by the labels: (a,b,d); this item would have [2,1,0,0] as a vector of belonging representation toward the F stereotypes, and [2,1,0,0] over the resorted T stereotypes. A second item B described with labels (c,g,h) would have [1,0,2,0] as a vector representation over the Full F stereotypes and [0,1,1,1] as vector representation over the T ste-

Stereotypes - Genre - Full Data	
1	[‘Music’,‘Musical’]
2	[‘Animation’,‘Family’,‘Children’s’,‘Fantasy’]
3	[‘Action’,‘Adventure’]
4	[‘War’,‘History’]
5	[‘TV Movie’,‘Foreign’,‘Western’]
6	[‘Film Noir’,‘Crime’,‘Thriller’,‘Mystery’]
7	[‘Romance’,‘Comedy’,‘Drama’,‘Horror’,‘Documentary’]
8	[‘Science Fiction’,‘Sci-Fi’]

Table 5.6: Stereotypes automatically generated using algorithm 1 for the feature genre over the Full dataset

reotypes. In the case of item A the vector difference between the two representations would be [0,0,0,0], and for item B instead would be [1,-1,1,-1].

In the first item there is no mismatch between the F and T stereotypical representations of the item, while the stereotypical representation of item B has a substantial mismatch. In particular the mismatch can be quantified by introducing a metric (the mismatch ratio): counting the sum of the labels non-matching, given by the sum of the positive entries in the vector difference, divided by the number of labels for the item. In the case of item A the mismatch ratio would be 0, for item B it would be $\frac{2}{3}$. Investigation of the distribution of the mismatch ratio over as many items as those in the test dataset (t) constitutes the soft test evaluation of how good and stable the stereotypes discovered on the Training data (T) are. In particular it is very indicative of the stability of the stereotypes the number of zero mismatches versus non-zero mismatches, as it can be seen as a simplified representation of how good the stereotypes constructed would be on out-of-sample recommendations.

For the feature **genre**, after assembling the Full dataset (Full= Training + test) and re-computing the stereotypes, one obtains those illustrated in Table 5.6.

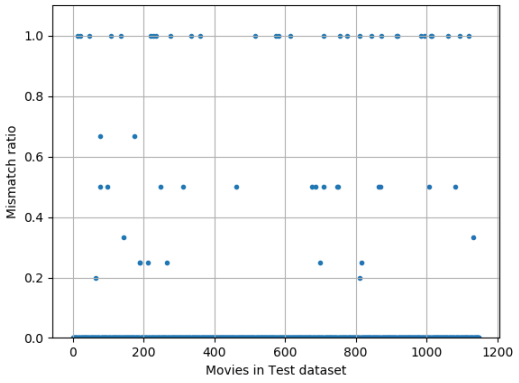


Figure 5.7: Mismatch ratio for the genre stereotyping representations of 1149 movies in the test dataset using Full versus Training stereotypes

	Stereotypes - Keywords - Full Data
1	[Explosion,Shootout, Violence, Flashback]
2	[Nudity, Female nudity, Sex]
3	[Biography, Sport, Musical]
4	[Suicide, Teacher, Dying and death]
5	[Film Noir, Obsession, Suspense]
6	[Adultery, Jealousy, Extramarital Affair, Divorce]
7	[High school, Teenager, Party]
8	[Gay, Based on Play or Musical]
9	[Journalist, World war ii, Based on novel]
10	[Revenge, Blood, Sequel]
11	[Widow, small town]
12	[Cat, Dog, Father son relationship]
13	[Investigation, Detective, Murder, New York]
14	[Lawyer, Rape, Prostitute]
15	[Prison, Escape]
16	[Police, Robbery, Corruption]
17	[After credits stinger, During credits stinger]
18	[Serial killer, Slasher]
19	[Woman director, Independent film]
20	[Friends, Love]
21	[Paris, London England]
22	[Money, Drug]
23	[Hostage, Kidnapping]
24	[Alien, Dystopia, Monter]
25	[Daughter, Mother daughter relationship, Wedding]
26	[Family, Los Angeles]

Table 5.7: Stereotypes automatically generated using algorithm 1 for the feature keywords over the Full dataset and the keywords enlarged set

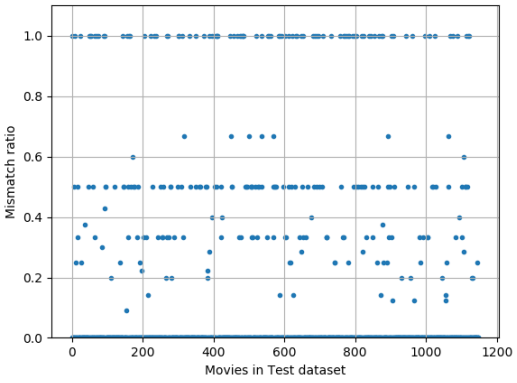


Figure 5.8: Mismatch ratio for the keywords stereotyping representations of 1149 movies in the test dataset using Full versus Training stereotypes

For each of the items in the test data (1149 movies) their stereotypical representations using the stereotypes F and T is performed, and the mismatch ratios computed. Figure 5.7 displays the mismatch ratios for all the samples in the test data for the feature genre. It is possible to

Descriptive Statistics of the mismatch ratios	Genre Value	Keywords Value
Average mismatch	3.5%	14%
Median mismatch	0%	0%
Number of non-zero mismatch	56	267
Proportion of non-zero mismatches	4.8%	23.2%
Mean conditioned on mismatch begin > 0	73%	61%

Table 5.8: Statistics describing the mismatch ratios for the stereotypical representations of complex categorical features for the items in test dataset using Full vs the Training stereotypes

observe how the mismatch ratios of the stereotypical representations of the items is zero in most cases (1093 items have a 0 mismatch) but when not zero, the mismatch is substantial and likely to be near 100%. As for the **keywords**, Table 5.7 represent the stereotypes assembled from the Full dataset, while Figure 5.8 illustrated the mismatch ratio of the stereotypical representations using the stereotypes F and T. The statistics of the mismatch ratios for both complex categorical features are highlighted in Table 5.8

The best way to express the results of the soft test is as follow: over a large sample of test items for the feature genre, there is a chance greater than 95% that the 'true' stereotypical representation of the item can be captured using the stereotypical representation of the item via the Training dataset. There is a 4.8% chance to commit an error, and in the few cases of error, the error is quite sizeable. What the above means is that if an unseen item was to be categorised via its genre into a stereotypical representation, then it is very likely that the categorisation will be right (95.2% chance), however in the case of a mistake, the categorisation will be substantially wrong.

Numerical Features

In a similar fashion to what has been done in the soft test evaluation of the stereotypes generated for complex categorical features, for numerical features the set of labels is achieved by creating the stereotypes on the union of the test and Training data. Once the synthetic 'true' labels are created for the items of the test data, the quality of the classification is investigated for each feature via the confusion matrix of the classification. The confusion matrix offers a summary of the correct vs incorrect classifications performed; the following example will review the idea behind the confusion matrix. Suppose a classification method is trying to predict the stereotype to which an item belongs to, and suppose that there are three stereotypes s1, s2, s3, and we know the 'true' value of the stereotype for all the items. The confusion matrix may look as that of Table 5.9. In this fictitious example the classifier makes a prediction for 100 items, and it predicts that 30 of them belong to the class s1 (where 30 is the sum of the elements in the first row, 20+2+8). In reality of the 30 predicted to be s1, 20 are truly s1, but 2 are in fact belonging to s2 while 8 to s3. The same can be done for the other rows, the classified predicts 25 elements to be of type s2, when in reality 20 are indeed of type 2, but 5 are of type 3.

For each stereotype prediction, it is possible to obtain the total of true positives (TP): the

Items= 100	**True s1**	**True s2**	**True s3**
Predicted s1	20	2	8
Predicted s2	0	20	5
Predicted s3	5	10	30

Table 5.9: An example of confusion matrix. Highlighted the areas of True Positives, True Negatives, False Positives, False Negatives for the prediction of s1

Items= 1014	**True s1**	**True s2**	**True s3**	**True s4**
Predicted s1	126	0	0	0
Predicted s2	0	335	0	0
Predicted s3	0	0	335	0
Predicted s4	0	0	0	218

Table 5.10: Confusion matrix for the feature: cast gender bias

number of items that were predicted correctly (for the example of Table 5.9 and the prediction of s1 it is 20, the cell highlighted in light green). The total of false positives (FP): the number of items that were predicted to belong to s1 but they actually belong to other stereotypes (for the example of Table 5.9 and the prediction of s1 is 10, the sum of the cells highlighted in yellow). The total of false negatives (FN): the number of items that were not predicted to be s1 but should have been s1 (for the example of Table 5.9 and the prediction of s1 is 5, the sum of the cells highlighted in orange). The total number of true negatives (TN): the numbers of items that were correctly predicted not to be s1 (for the example of Table 5.9 and the prediction of s1 is 65, the sum of the cells highlighted in dark green).

For each stereotype, as well as for the entire features, the accuracy of the stereotypes can be defined as the number of true positives (TP) divided by the number of items. Another alternative metric to evaluate the classifier is via the F1 score, which is the harmonic average of precision and recall, where the recall is defined as the number of true positives (TP) divided by the number of positives (TP + FP); the precision is defined as the number of true positives (TP) divided by the number of true positives and false negatives (TP+FN).

Tables 5.10 to 5.22 display the confusion matrices for all the stereotypes of the numerical features. For each feature the total number of items is fewer than the 1149 items comprising the full test dataset. The difference in items arises because many items have invalid/missing entries for the numerical feature presented. It is believed to be safer to remove such entries from each evaluation set, rather than create an ad hoc methodology to handle missing values. Such a methodology would be required if the algorithms were to be used in a recommender system, but for the current scope of providing an evaluation stage any superimposed methodology for missing or invalid entries could increase the risk of obfuscating the results. In the Tables 5.10 to 5.22, the incorrect predictions are highlighted in orange.

Items= 1014	True s1	True s2	True s3	True s4
Predicted s1	478	112	0	0
Predicted s2	0	220	118	0
Predicted s3	0	0	51	27
Predicted s4	0	0	0	8

Table 5.11: Confusion matrix for the feature: cast popularity

Items= 1014	True s1	True s2
Predicted s1	754	0
Predicted s2	0	260

Table 5.12: Confusion matrix for the feature: country distance

Items= 1014	True s1	True s2	True s3	True s4
Predicted s1	532	6	0	0
Predicted s2	0	254	115	0
Predicted s3	0	0	44	49
Predicted s4	0	0	0	14

Table 5.13: Confusion matrix for the feature: director popularity

Items= 1014	True s1	True s2	True s3
Predicted s1	515	0	0
Predicted s2	0	228	0
Predicted s3	0	0	271

Table 5.14: Confusion matrix for the feature: log budget

Items= 1014	True s1	True s2	True s3
Predicted s1	516	0	0
Predicted s2	0	371	0
Predicted s3	0	0	127

Table 5.15: Confusion matrix for the feature: log revenue

Items= 1014	True s1	True s2	True s3	True s4
Predicted s1	264	0	0	0
Predicted s2	17	243	0	0
Predicted s3	0	4	243	0
Predicted s4	0	0	9	234

Table 5.16: Confusion matrix for the feature: log vote count

Items= 1014	True s1	True s2	True s3	True s4
Predicted s1	254	0	0	0
Predicted s2	1	271	0	0
Predicted s3	0	7	242	0
Predicted s4	0	0	6	233

Table 5.17: Confusion matrix for the feature: popularity

Items= 1014	True s1	True s2	True s3	True s4
Predicted s1	332	54	0	0
Predicted s2	0	253	64	0
Predicted s3	0	0	192	32
Predicted s4	0	0	0	87

Table 5.18: Confusion matrix for the feature: production company popularity

Items= 1014	True s1	True s2	True s3	True s4
Predicted s1	308	0	0	0
Predicted s2	0	229	0	0
Predicted s3	0	0	257	0
Predicted s4	0	0	0	220

Table 5.19: Confusion matrix for the feature: release time of the year

Items= 1014	True s1	True s2	True s3	True s4
Predicted s1	227	14	0	0
Predicted s2	0	227	0	0
Predicted s3	0	0	278	0
Predicted s4	0	0	0	268

Table 5.20: Confusion matrix for the feature: release year

Items= 1014	True s1	True s2	True s3	True s4
Predicted s1	214	0	0	0
Predicted s2	0	272	0	0
Predicted s3	0	0	245	15
Predicted s4	0	0	0	268

Table 5.21: Confusion matrix for the feature: runtime

Items= 1014	True s1	True s2	True s3	True s4
Predicted s1	252	0	0	0
Predicted s2	0	247	0	0
Predicted s3	0	0	250	0
Predicted s4	0	0	0	265

Table 5.22: Confusion matrix for the feature: vote average

Feature Name	F1-score	Accuracy
cast gender bias	1.0	100%
cast popularity	0.77	75%
country distance	1.0	100%
director popularity	0.86	83%
log(budget)	1.0	100%
log(revenue)	1.0	100%
log(vote count)	0.98	98%
popularity	0.99	98%
Prod. comp. popularity	0.86	85%
Release time of the year	1.0	100%
Release year	0.98	98%
runtime	0.98	98%
vote average	1.0	100%

Table 5.23: Numerical features stereotypes evaluation, soft test. F1-score and accuracy metrics for the classification problem of the test items using the stereotypes generated on full items

Table 5.23 displays the accuracy and F1-score metrics derived from the confusion matrices for the numerical features. Cast popularity is the feature with the lowest accuracy of prediction, at 75%, all other features have an accuracy well above 80%, with six features having perfect accuracy of 100%. The soft test overall confirms a remarkable stability, thus paving the road to using such stereotypes in the context of recommendation.

5.2.3 Predictive Power of Stereotypes

Before concluding this chapter, one last statistical test will examine how much user's preference traits can indeed be described via the stereotypes discovered in the test set. One can test how much biased user's selection were: does the user display a statistically significant positive or negative bias toward a stereotype compared to the item's population distribution?

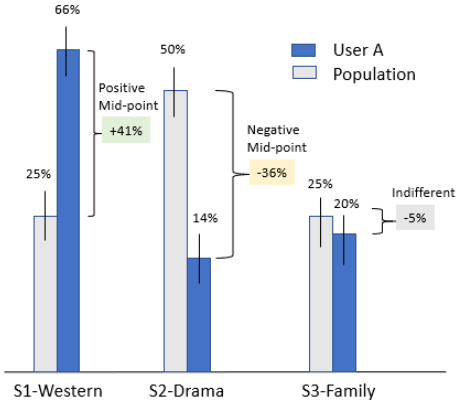


Figure 5.9: Example of the Agresti-Coull test for an imaginary user in the example described in the text

A simple example can aid to explain the test procedure more closely, let us suppose that the universe of movies was constituted only by 300 movies and these had been stereotyped for the genre feature in 75 western movies, 150 drama movies, and 75 family movies. With these numbers the proportions of each stereotype over the population of movies are 25% western, 50% for drama, and 25% for the family type. Suppose that user A rates 30 movies, and the movies rated are 20 western, 4 drama, 6 family; considering user A's own 'universe' of chosen movies roughly 66% are westerns, 14% drama and 20% family. The situation for user A is also depicted in Figure 5.9. Another user, user B, also rates 30 movies: 5 western, 16 drama and 9 family. For user B the proportions are: 17% western, 53% drama, 30% family. Using the fictitious numbers of the example user A has seen and rated 66% western, while the population of westerns was 25%, thus a difference in proportion of +41%; we can say that westerns is an important stereotype for user A, (suggesting that the user seeks such movies). The difference in drama for user A is (14%-50%), or -36%, a negative excess, or a negative mid-point of the difference, and that allows us to say that this user avoids the stereotype drama. The difference in family movies is -5%, being smaller in magnitude, we cannot conclude anything about family, i.e this user does not show a statistically significant preference or dislike for movies of the family stereotypes.

For user B (not depicted in the figure), by computing the difference between proportions and comparing the differences obtained, we see that there we can draw no conclusions for all three stereotypes, the user B does not show a preference or dislike for movies based on the stereotypes created. Therefore, the stereotypes of this fictitious example seem to represent well the like/dislike of user A, but not that of user B. This type of reasoning can be done via a statistical test whose Null Hypothesis is: 'for the stereotype investigated the user consumed a proportion of items that is similar to the proportion expected if the stereotype had no influence in the user's choice'. Rejecting the Null means that the stereotype has indeed an influence in shaping that user's preferences.

The mechanics of the test is carried out via the calculation of Confidence Intervals for the difference of two proportions arising from binomial and multinomial distributions. This statistical problem was studied by Agresti and Coull [170], and a formula for the Confidence interval corresponding to a given statistical significance level proposed in the same reference. By examining the results of the Agresti-Coull test across the thousands of users of the test dataset, and across all features and stereotypes, one can develop an understanding of whether or not stereotypes are descriptive of user's preferences. Preferences here indicate both positive and negative biases, for example the fact that the Agresti-Coull test shows that it is statistically significant that a given user has not consumed an amount of items falling in a particular stereotype (for example low budget movie items) still indicates precious information for a recommender system.

Table 5.24 displays the results of the application of the Agresti-Coull test with 99% confidence. The table is ordered by the proxy of how significant the feature is for the user's population

Feature	% users not different	% users with L.P.P	% users with L.N.P	Avg of significant stereotypes	Total stereo-types
Log (Vote count)	1.1%	97%	95%	3.0	4
Popularity	3.0%	91%	92%	2.6	4
log (budget)	4.3%	85%	93%	2.0	3
log (revenue)	4.4%	85%	93%	2.2	3
Genre	12.3%	26%	30%	3.3	8
Vote Avg	12.7%	66%	65%	2.0	4
Keywords	18.0%	53%	46%	3.0	25
Release year	22.1%	49%	57%	2.1	4
Director popularity	23.4%	19%	0%	2.03	5
Cast popularity	24.7%	43%	47%	2.2	4
Runtime	25.7%	53%	28%	2.0	4
Prod. comp. popularity	26.7%	22%	43%	1.7	4
Release time of the year	65.4%	13%	2%	1.3	5
Cast gender bias	70.2%	6%	3%	1.3	4
Country	75.6%	5%	5%	2.0	2

Table 5.24: Summary for all features of the search for explanatory power of stereotypes via the Agresti-Coull test. Confidence level of 99%

and can be read as follow: for the feature genre only 12.3% of users display no significative positive or negative preferences toward at least one of the genre stereotypes. Of the users displaying at least one positive preference (87.7% of the population), only 26% display a large positive preference (L.P.P) toward at least one stereotype, and 30% display a large negative preference (L.N.P) toward at least one (the two may not be mutually exclusive). The stereotypes discovered are indeed capable of describing positive and negative preference traits for over 70% of the users.

5.3 Summary

This chapter concludes with two observations; the first is that, in the experiment under examination, the stereotypes obtained via the proposed methodology have been shown to be stable on out-of-sample data in both the hard and soft tests (i.e. they accurately describe the item population metadata with a reduced set of dimensions, and they are capable of describing users' positive and negative preferences). The result of this chapter is the output of phase 2 as depicted in research methodology in Chapter 3.

The result of the three statistical tests proposed in this chapter are the keys to confirming that, for the problem at hand, one can proceed to embed the stereotypes as the base coordinates in an RS which will be covered in the next chapter.

The second observation is that the suite of test presented as a way to aid the preliminary evaluation of the stereotypes can be considered among the contributions of this research.

CHAPTER 6

Stereotype-Based Recommendation Performance

The analysis carried out in the previous Chapter 5 have clearly demonstrated that the stereotypes discovered via the proposed automated procedure are capable of representing users' preferences. A natural next step is to test the power of the stereotypes in the recommendation context.

In the existing literature that deals with the recommender systems, the majority of the studies focus on predicting ratings. While the rating is the ultimate variable expressing user's preferences, it can be seen as incorporating two stages of the user decision process: first the user chose to consume the item, then decides to express a rating conveying his/her appreciation of the item. This chapter aims to benchmark the use of stereotypes against the recent advancement in recommender systems in the cold start phase using different metrics, more details are provided in Section 6.1.

The chapter main contribution is to provide answers to the following research questions:

- *Can automatically constructed item-based stereotypes improve recommendations during the cold-start phases?*
- *How do stereotype-based item models compare to other state-of-the-art recommender systems models in terms of predictive performance during the cold-start phases?*
- *Apart from recommendation accuracy, what are the other benefits of item-based stereotypes?*

6.1 Experimental Evaluation

In this section, we present the assessment of recommendations driven by stereotypes during cold starts in three stages:

1. Cold start user-to-item pairing (or consumption). The aim is to predict, hence recommend, which items a user is likely to consume. The effects and potential benefits of introducing stereotypes over using the primitive metadata features are investigated in Section 6.1.1.

2. Cold start user-to-item rating. The aim is to predict the rating that a user would attribute to an unconsumed item, and hence recommend the items with the highest predicted ratings. The effects of introducing stereotypes over using the primitive metadata features are investigated in Section 6.1.2.

3. Cold start general benchmarking of a RS driven by stereotypes against a RS driven by singular value decomposition (SVD) with metadata. The aim is to benchmark the cold start recommendations produced by the stereotype-based system with those produced by a state-of-the-art RS on the same data. This evaluation is carried out in Section 6.1.3.

The results reported in the following experiments are the average over six experiments in which the dataset was split 70% in training and 30% in test for ML/IMDb. The test data was effectively split in six successive experiments with some overlap of the test data allowed across experiments. In the item's consumption case the variable predicted is a binary variable expressing whether a user consumed an item or not, leading to a 'user-to-item consumption matrix'. In the rating case, the variable predicted is the MovieLens rating, which is reported on a 1 to 5 scale.

Recommendation metrics can be broadly classified into three classes: classification accuracy metrics, predictive accuracy metrics, and rank accuracy metrics [35, 144]. In Section 6.1.1, the objective is to predict consumption, hence classification accuracy metrics are adopted. Predictive accuracy metrics are used to measure how well stereotype-based models predict ratings compared to models using the original metadata (Section 6.1.2). Finally, in Section 6.1.3, the cold start recommendations of the stereotype-based system are assessed versus the SVD-based RS (with metadata). In addition to the standard prediction accuracy metrics, several most important rank accuracy metrics are examined including: hit rate (HR), mean reciprocal rank (MRR), mean average precision (MAP), normalised discounted cumulative gain (nDCG) and half-life utility metric (HLU). Rank accuracy metrics measure the ability of a recommendation algorithm to produce a list of items that meets the specific user preferences in terms of hits and ordering (i.e. the items are all interesting to the user, the list replicates how the user would have ordered the same items). In the same section an attempt is also made to measure recommendation's novelty/serendipity.

6.1.1 Cold Start Assessment of Item Consumption

When performing predictions for item's consumption one is not just interested in the class label (0,1), but also in obtaining an estimate of the probability of how likely it is that a user consumes an item. For such an experiment, a simple neural network with a single layer of neurons and a softmax layer to rescale the output to a probability density was chosen as a classifier. Subsequently this classifier will be referred to as the neural network with softmax recommender (NNSR).

Baseline Model and Stereotype-Based Models

Given the different nature of stereotypes for complex categorical features and for numerical features, in this preliminary phase the evaluation of recommendations is done for the two types of stereotypes independently. This aims to demonstrate that performance improvements are intrinsic of the stereotypes approach, and not due to one particular type of feature, or any feature in particular. Then a combination of categorical and numerical stereotypes leading to a complete stereotype-based system is also examined to study the impact on recommendations of the two classes. Therefore, in the experiments conducted four recommendation models were tested:

- A baseline model ($NNSR_b$), which uses all features available in the item and user metadata as they are in the original data.
- A complex categorical stereotype model ($NNSR_c$) which uses the stereotypes for the complex categorical features and revert to the standard features for the rest.
- A numerical stereotype model ($NNSR_n$) which uses the stereotypes for the numerical features and revert to the standard features for the categorical ones.
- A combined stereotype model ($NNSR_{com}$) which uses the mixture stereotypes for the categorical and numerical features.

For this section, the baseline model is the reference model in terms of performance. For each model two experiments are performed:

- Experiment A (new user) - The models are trained over a subset of users. The remaining users are used to test the model performance, treating them as if they were new unknown users, and checking whether the item predicted to be consumed were effectively consumed.
- Experiment B (new item) - The models are trained over preferences expressed for a subset of items. The remaining items are used to test the model performance treating them as if they were new, and checking the predicted item consumption with the real consumption in the data.

Metric	New User Experiment				New Item Experiment			
	$NNSR_b$	$NNSR_c$	$NNSR_n$	$NNSR_{com}$	$NNSR_b$	$NNSR_c$	$NNSR_n$	$NNSR_{com}$
Accuracy	71.49%	71.68%	71.30%	71.34%	70.8%	71.3%	71.4%	71.4%
Percision	30.27%	30.49%	30.71%	30.65%	29.6%	30.2%	30.8%	30.7%
T.P. Rate	73.32%	73.55%	76.6 %	76.14%	73.2%	73.8%	76.6 %	76.2%
F. P Rate	28.82%	28.64%	29.6%	29.48%	29.6%	29.1%	29.5%	29.4%
ROC AUC	79.6%	80.0%	79.8%	80.65%	79.3%	80.1%	80.9 %	80.7%
PRC AUC	41.1%	41.8%	41.9 %	41.44%	40.2%	42.2%	41.9 %	41.5%

Table 6.1: Classification-prediction metrics derived from the confusion matrices, including the area under the curve (AUC) for both the receiver operating characteristic (ROC) and the precision-recall curve (PRC) for the new-user and new-item experiments in the ML/IMDb. (T.P. refers to true positive, and F.P. refers to false positive)

Recommendation Results: New User and New Item Experiments

Table 6.1 shows the metrics derived from the confusion matrices for the 'new user' and 'new item' experiments. To evaluate the model's skills the area under the curve (AUC) for both the receiver operating characteristic (ROC), and for the precision-recall (PRC) curves are reported. When the classes predicted are very unbalanced, as it is the case in situations where users have consumed only few items compared to the population of items (e.g. unbalanced presence of 0s over 1s in the data), predicting rare '1' events (true positives) becomes more important than predicting 0 (true negatives). In such cases as prescribed by [171] the AUC for the PRC may be a better indication of model's skills, although the latter is less easy to interpret. ROC curve is a plot of the true positive rate versus the false positive rate for the predictions of a model for multiple thresholds between 0.0 and 1.0. Herlocker et al [32] recommend to use ROC to evaluate RSs.

It can be observed that the stereotypes-based system provides an improvement in predicting items consumption, especially for the True Positive metric and the PRC AUC. This despite using features with much lower dimensions compared to the original metadata. This first analysis demonstrates that the improvements come from both numerical and categorical stereotypes in an independent manner, hinting that the dimension reduction process intrinsically embeds elements of increased predictability. Coupling the two stereotypes categories leads to a mixture of the two which is expected.

In most commercial applications it is customary to take the top-N recommendations predicted by the RS. The $NNSR$ systems by construction can predict, for every item, the probability of consumption by a given user. For every new user the top-N items ranked by probability of consumption are selected and cross checked for actual consumption, and the precision metrics computed. Tables 6.2 and 6.3 show the sample statistics of precision. The tables also provide the *p*-value of the statement 'the stereotype-based model performs better when predicting the top-N items than the baseline model using the original metadata'. For example, one can state that for

Model	**Top-N Recommendation**	**Top 50**	**Top 100**	**Top 150**
$NNSR_b$	Avg Precision	25.70%	42.44%	56.53%
	Precision Std	15.9%	14.2%	13.2%
$NNSR_c$	Avg Precision	26.16%	43.29%	57.13%
	Precision Std	16.0%	14.3%	13.3%
	Precision % improvement	1.79%	2.03%	1.07%
	P-value	0.36	0.04	0.21
$NNSR_n$	Avg Precision	25.82%	44.55%	58.33%
	Precision Std	16.8%	14.9%	13.6%
	Precision % improvement	0.47%	4.85%	3.18%
	P-value	0.78	<0.01	0.01
$NNSR_{com}$	Avg Precision	25.50%	43.82%	58.10%
	Precision Std	16.86%	14.85%	13.74%
	Precision % improvement	-0.8%	3.3%	2.8%
	P-value	0.57	0.03	0.07

Table 6.2: New User: Top-N Recommendations - performance metrics of stereotype models $NNSR_c$, $NNSR_n$, $NNSR_{com}$ vs baseline model $NNSR_b$, plus performance increase and *p*-value of the test on the significance of the increased performance due to stereotypes

Model	**Top-N Recommendation**	**Top 50**	**Top 100**	**Top 150**
$NNSR_b$	Avg Precision	22.87%	41.93%	55.12%
	Precision Std	17.8%	14.8%	13.3%
$NNSR_c$	Avg Precision	25.49%	43.08%	56.96%
	Precision Std	16.0%	14.1%	13.1%
	Precision % improvement	11.4%	2.74%	3.34%
	p-value	<0.01	0.09	0.04
$NNSR_n$	Avg Precision	24.72%	44.74%	58.50%
	Precision Std	16.8%	14.9%	13.4%
	Precision % improvement	8.09%	6.73%	6.13%
	p-value	0.04	<0.01	< 0.01
$NNSR_{com}$	Avg Precision	25.58%	44.08%	58.21%
	Precision Std	16.79%	14.80%	13.42%
	Precision % improvement	11.85%	5.13%	5.61%
	p-value	0.065	<0.01	0.01

Table 6.3: New Item: Top-N Recommendations - performance metrics of stereotype models $NNSR_c$, $NNSR_n$, $NNSR_{com}$ vs baseline model $NNSR_b$, plus performance increase and *p*-value of the test on the significance of the increased performance due to stereotypes

the top 100 items we are more than 96% (i.e. 1 - 0.04 'p-value') confident that the stereotype-based model $NNSR_c$ for new user experiment performs better than the baseline model, and we are more than 99% confident that the stereotype model $NNSR_n$ performs better than the base model, and our estimated improvement in precision per user is in the region of 4.85% for the $NNSR_n$.

Handling of imbalanced data

A dataset is imbalanced if the classification classes are not approximately equally represented. In recent years there has been an increased interest in applying machine learning techniques to difficult real-world problems, many of which are characterised by extremely imbalanced dataset, where the occurrence of an event is particularly rare. In addition, the distribution of the testing data may vary from that of the training data, and the true misclassification costs may be unknown at learning time. There are two main problems arising from data with unequal class distribution, as follows:

1. Machine learning (ML) algorithms are built to minimise errors. Since the probability of samples belonging to the majority class is substantially higher in imbalanced dataset, the algorithms are much more likely to classify new observations to the majority class. For example, given an imbalanced dataset where the majority class represent the 98% of the observations (against a 2% of the minority class-which is also usually the class of interest), we could have a classifier which achieves a prediction accuracy of 98% by classifying all instances as the majority class and eliminating the 2% minority class observations as noise. Such a classifier would be obviously wrong. In other words, predictive accuracy, a common option to evaluate the output of a classifier, may not be appropriate when data is imbalanced and/or the cost of different errors varies considerably. This is known as the accuracy paradox [172].

2. In real life, costs of different errors vary markedly. The cost of False Negative is usually much larger than False Positive, yet ML algorithms penalise both types with a similar weight. Credit scoring is a typical example, if the model predicts that a loan will default, and the loan is not granted, yet it turned out not to be the case, the maximum loss the lender is subjected to is the profit they would have generated by issuing that loan. On the other hand, if the model classifies a loan that defaults as being safe, the cost is substantially larger, as part of the principal may not be recovered. This illustrates an example where one should not weight False Positive and False Negative equally.

For the ML/IMDb dataset, the data is very sparse: any given person has seen only a small fraction of all movie. On average the user has rated about 115 movies and a median of 67 movies

(i.e. the user rated less than 5% on average). The result is that our dataset is heavily unbalanced. One way to overcome this issue is to use the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as:

$$\frac{n_{samples}}{n_{samples} * freq_{class(y)}} \tag{6.1}$$

Where $n_{samples}$ are the number of samples, $n_{classes}$ is the number of classes (which in our case is two) and $freq_{class(y)}$ is the frequency of occurrence of a class.

We used this method to deal with imbalanced dataset in the previous experiments. Beside the above method which introduce a penalty function to adjust class weights inversely proportional to class frequencies to handle imbalanced data, in this section we are also looking to:

- Validate the results obtained i.e. show that the model with stereotypes has better performances than the base model regardless of the method we use to treat imbalance in the dataset.
- Compare different common methodologies to deal with imbalanced datasets.

Techniques to deal with imbalanced data. The most common techniques used to deal with imbalanced datasets are:

1-Resampling techniques

- Undersampling is the process of deleting some of the observations from the majority class in order to match the numbers with the minority class. The main method to select which observation to delete/keep are:
  - Random undersampling: as the name suggests this a purely random selection.
  - ClusterCentroids [173], in the majority class cluster of samples are replaced by the cluster centroid of a k-means algorithm.
  - NearMiss [173], in order to overcome the issue of potential information loss, 'near neighbour' method and its variations have been proposed. The basic algorithms of the near neighbour family consist of: first, the method calculates the difference between all instances of the majority class and those of the minority class. Then k instances of the majority class which have the smallest distances from those of the minority class are chosen. If there are n instances in the minority class, the 'nearest' method will result in $k * n$ instances of the majority class.

  As discussed by [174] the main advantages and disadvantages of the undersampling techniques were shown to be that it can help improve model runtime and solve the memory problems by lowering the number of training data samples when the training dataset is extremely large. However, it can potentially discard useful information about the data itself that might be needed to build rule-based classifiers. Methods such as ClusterCentroid or NearMiss are in general preferable to that of random undersampling.

- Oversampling is the process of generating synthetic data that attempts to randomly generate a sample of the attributes from observations in the minority class. There are a range of methods used to oversample a dataset for classification problem. From random-sampling to the most common technique used that is called synthetic minority over-sampling technique (SMOTE) [175]. It basically make the minority class equal to the majority class by creating synthetic (not duplicate) samples of the minority class. The advantages of oversampling using the SMOTE technique are its simplicity and no lost of information. However, it is not very practical for high dimensional data.

  2- Training techniques: Cost-Sensitive Training, see for example [176]: a number of learning models provide some built-in support to deal with imbalance data such as class weighting to automatically adjust weights inversely proportional to class frequencies in the input data as explained in equation 6.1 and implemented in the previous experiments.

Comparison of techniques on the ML/IMDb dataset. In order to select the most appropriate method to deal with imbalance dataset, and validate the results obtained so far, the main techniques listed above were tested over the training of both the baseline system and the combined numerical plus categorical stereotype system for the new user experiment. The first test we performed consists in using the $NNSR$ on the imbalanced dataset without applying any correction of the imbalance. Considering in our dataset each user has rated less than 5% of the movies (i.e. our dataset is highly imbalanced) we would expect to obtain fairly poor results from this first test.

Table 6.4 shows the key metrics obtained using the different techniques to treat imbalanced data. When using an unbalanced model (i.e. without correcting the imbalance) the accuracy seems very high, around 86% for both models. Precision, TP/(TP+FP), seems as well fairly high but that is due to the very low number of false positives (FP) and true positives (TP). However, other metrics such as recall, TP/(TP+FN), and false positive rate are extremely low highlighting the inherent problems in the model. The no correction model tends to assign every movie to the predominant class. This constitutes another example of the accuracy paradox [172]. Interesting, using the imbalanced dataset, without correction, penalises the model with stereotypes over the base model (i.e. using stereotype does not seem to reduce imbalance). Overall, the results obtained confirm the improvement in predictions obtained when employing stereotypes.

Using the methods SMOTE and Penalised seems to lead to the best results for both base model and model with stereotypes confirming the improvements obtained using the model with stereotypes. Figure 6.1 shows a plot of the True Positive Rate (Recall) for the various techniques employed to treat imbalance of the important class on the two models i.e. base model and model with stereotypes. While when using an imbalanced dataset, the base

Base Model on New User					
Metric	No correction	Penalised	Rnd Undersmp	Near Miss	SMOTE
Accuracy	86.15%	71.49%	71.8%	73.2%	72.0%
Precision	60.70%	30.27%	30.5%	31.5%	30.7%
Recall	14.28%	73.32%	73.4%	71.4%	73.2%
False Positive Rate	1.58%	28.82%	28.53%	26.46%	28.17%
F1-score	23.12%	42.85%	43.09%	43.71%	43.25%
Stereotype Model on New User					
Accuracy	85.96%	71.34%	71.3%	73.8%	71.5%
Precision	59.41%	30.65%	30.6%	31.7%	30.7%
Recall	12.35%	76.14%	76.0%	68.4%	76.0%
False Positive Rate	1.44%	29.48%	29.46%	25.25%	29.29%
F1-score	20.45%	43.71%	43.63%	43.32%	43.73%

Table 6.4: Handling imbalanced dataset for new user problem

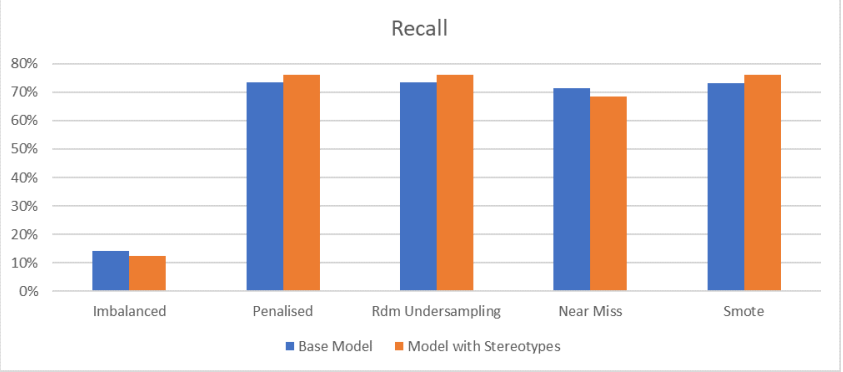


Figure 6.1: Comparison between methods to treat imbalance on base model and model with stereotypes

model performs slightly better than the model with stereotypes, correcting the imbalance leads to a marked improvement in performances for both models and demonstrate as the model with stereotypes does indeed brings benefit to the prediction.

6.1.2 Cold Start Assessment of Item Rating

Having demonstrated in Section 6.1.1 that the use of stereotypes improves the cold start predictions for item consumption, and having also demonstrated that both numerical and complex categorical stereotypes provide independent sources of improvement, this section focuses on predicting rating, with the full range of features stereotyped.

Given the nature of the rating variable, generally represented as a discontinuous number with R possible values, one has two options available: to use a classification approach using R buckets, or to predict the normalised/scaled dependent variable using a regression like algorithm. In the literature there are examples of both methods, see [177] for a clas-

sification and [1] for a regression example. In the present work it is demonstrated how stereotypes can be used in a classification approach in Section 6.1.1, while this section focus on the evaluation of the potential benefit of using stereotypes vs original metadata using regression like approaches.

Generally, user-to-item ratings exhibit biases [178], for example some users always tend to give higher or lower ratings, or ratings restricted to a narrower range. Several techniques have been proposed in the literature to account for such biases, for example subtracting from the original entries user-averages and/or subtracting item-average [1, 178]. In the present study, ratings are normalised per user by converting them to standard scores:

$$\tilde{r} = \frac{(r - \mu_u)}{\sigma_u} \tag{6.2}$$

Where μ_u is the mean rating of the user considered, and σ_u is the standard deviation of such user.

For each of the two experiments (new user and new item as explained in Section 6.1.1) several machine learning algorithms capable of predicting a numerical target variable are tested, where the only difference between the setups evaluated consists in how the predictor features are treated. In the baseline model, all features are treated as they are in the original dataset. In the stereotype model, all features are treated via their stereotypes representation. Our proposed representation of stereotyping allows the application of virtually any machine learning algorithm. The algorithms tested and presented for this evaluation cover the full spectrum of algorithm complexity:

- A naive approach where a system is metadata unaware and involves either a) predicting a rating that equals the average rating for the item considered (new user) with no regard to the specific user or b) predicting a rating that equals the average rating of the user considered, with no regard to the specific item (new item).
- A simple linear regression approach where the target value, in this case an item's rating, is expected to be assembled as the linear combination of the effects of each feature. The algorithm fits a linear model to minimise the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation [179].
- A neural network regression approach which is based on a single-layer neural network with a softmax layer. Neural networks process information in a similar way as the human brain learning by example, they cannot be programmed to perform a specific task.
- A XGBoost driven regression where XGBoost stands for eXtreme Gradient Boosting

and it was developed by Tianqi[1] as an implementation of gradient boosted decision trees classifiers and regressors [180, 181]. XGBoost combines a few advantages, including an ability to find non-linear transformations, ability to handle skewed variables without requiring transformations and high scalability.

It should be noted that, for numerical regression style models, there is no guarantee that the rating will fall between 1 and 5, therefore the resulting prediction, when transformed back (rescaled) into a rating 'r', must be capped/floored in the following manner:

- 1 if r $\leq$ 1
- 5 if r $\geq$ 5
- r otherwise

We do not, however, constrain our predicted ratings to be integer numbers. We note that in practical applications of recommender systems, the fractional parts of the prediction can be helpful in defining a ranking order among items (i.e. an item whose predicted rating is 3.9 should rank lower than an item whose predicted rating is 4.2, however if predicted ratings were rounded to the nearest integer they would have the same value, 4). Additionally, when the scoring metric is the mean squared error (one of the most popular reported metrics - see also Netflix challenge [98]), rounding predictions almost always affects the error.

Rating Predictions and Recommendation Results: New User and New Item Experiments

Tables 6.5 and 6.6 show the key performance metrics obtained for the new user and new item experiments respectively. The tables display prediction accuracy metrics for the naive system and for the RS derived via the three different regression approaches, as well as the different treatments of the metadata each regressor uses (original metadata are indicated as base features vs stereotyped features).

As expected, using a RS that is able to extract rating relationships from metadata improves substantially the error in the cold start rating predictions compared to the naive approach. Also, increasing the ability of the regressor to make use of the metadata improves the rating prediction (i.e. moving from a simple linear model to a more complex neural network-driven regression).

Contrary to what intuition might have suggested, reducing the metadata feature space via the use of stereotypes not only does not make the rating prediction worse, but they are in fact improved. For instance, in the new item experiment, the improvement in precision metrics that can be gained using stereotyped features is equal or larger than the improvement in the same metrics that arises from switching from a simple regression model to a

[1] https://tqchen.com/

more complex one, like neural network regression. Also, and probably most importantly, the benefit obtained in extra precision using stereotypes does not depend on the regression model used; this suggests that stereotypes offer an extra dimension for improvement to the problem of better recommendations (at least in cold start phases) that is *independent* of the rating prediction algorithm used. Moreover, the complexity reduction in the metadata features, once stereotypes are adopted, can be appreciated via the reduction in the CPU time (in seconds) required for training a given regression approach. For the most complex regressors the stereotyped metadata allows in excess of 20% improvement in CPU time. This is an extra benefit of the proposed approach. All experiments are run on a Intel Core i7 -7700K CPU @ 4.2 GHz with 64.0 GB RAM.

Finally, the improvement of prediction accuracy and computational efficiency can be evaluated by introducing an ad-hoc metric, we call it Efficiency metric, as follow:

$$Efficiency = \frac{RMSENaive}{RMSE} * (1 + k * \frac{CPUTimeNaive}{CPUTime}) \tag{6.3}$$

The metric aims at weighting the effects of an increase in prediction accuracy as measured by the relative improvement of RMSE compared to the naive RMSE together with the improvement in computational efficiency. The k of the formula is a parameter, set at 0.03, that aims at under weighting the improvements in computational efficiency, which are important, but of a lower importance compared to improvements in RMSE.

Looking at the percentage improvements obtained in the efficiency metrics in the new user case, one can see that the improvements in using stereotypes (0.6%) are higher than the improvements in increasing the model complexity from a simple linear regression to XGBoost (0.4%). Therefore, providing more grounded evidence for the use of stereotypes in cold start phases.

To gain a view of the goodness of the cold start predictions for the new user and new item problems we have collected and analysed the distribution of the error in the final rating as integer values from 1 to 5 obtained by rounding the values deriving from the XGBoost model with stereotypes.

Figure 6.2 displays an histogram of the distribution of such rating error. For example, an error of +2 means that the predicted rating was two steps higher of the actual rate given by the new user (to the new item). We can see that 43% of the predictions have no error, and that if one considers also predictions where the error amount is of +/- 1 (a small error) - i.e. the predicted rating was off only by one level up or down - then 91% of the population predicted. This is an impressive result, demonstrating how the model suggested is indeed capable of predicting the preference/dispreference in cold start phases over 9 times out of 10.

New User Experiment		
RMSE (Naive: 0.963)	**Base features**	**Stereotyped features**
Linear R.	0.940	0.938
Neural Net R.	0.918	0.906
XGBoost R.	0.913	0.901
MAE (Naive: 0.772)		
Linear R.	0.743	0.742
Neural Net R.	0.724	0.712
XGBoost R.	0.721	0.710
Cpu Time (Naive: 1)		
Linear R.	10.7	9.1
Neural Net R.	69.5	55.6
XGBoost R.	90.5	73.2
Efficiency Metric		
Linear R.	1.054	1.060
Neural Net R.	1.053	1.069
XGBoost R.	1.058	1.073

Table 6.5: Performance metrics: new user problem

New Item Experiment		
RMSE (Naive: 1.01)	**Base features**	**Stereotyped features**
Linear R.	0.939	0.934
Neural Net R.	0.928	0.917
XGBoost R.	0.926	0.918
MAE (Naive: 0.81)		
Linear R.	0.740	0.736
Neural Net R.	0.735	0.727
XGBoost R.	0.738	0.729
Cpu Time (Naive: 1)		
Linear R.	10.8	8.6
Neural Net R.	56.8	34.9
XGBoost R.	90.5	71.6
Efficiency Metric		
Linear R.	1.104	1.118
Neural Net R.	1.093	1.110
XGBoost R.	1.093	1.104

Table 6.6: Performance metrics: new item problem

Comparison with existing studies

Despite the majority of the existing studies focused on the movie recommendation domain and a large number of studies have used the well-known MovieLens and Netfix datasets, there are fewer published works using the combination of IMDb and MovieLens features (an enlarged dataset) to predict MovieLens' users ratings. One such work is [1], where a hybrid method combining collaborative and content-based recommendation is used to predict rating of the MovieLens dataset enhanced by the IMDb movie attributes. In order to

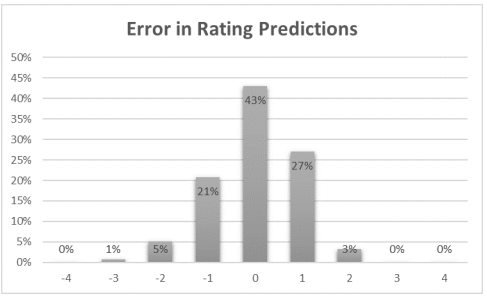


Figure 6.2: Distribution of rating error - new user problem

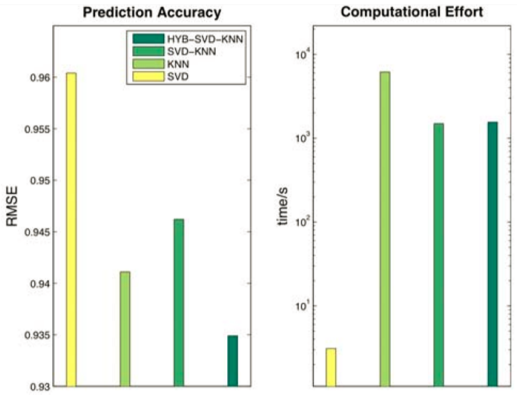


Figure 6.3: Performance comparison between different algorithm implementation. Source [1]

decrease system runtime and to reveal latent user and item relations, Spiegel et al. [1], use various methods from the simple K-nearest neighbours (KNN), to a single value decomposition (SVD) and finally a combination of the two methods with their hybrid filtering approach which they refer to as 'HYB-SVD-KNN'. The novelty in their paper lies in using a hybrid model (combining collaborative and content-based filtering) and in applying a combination between SVD and KNN method.

Figure 6.3 shows the results obtained by Spiegel et al. [1] in terms of both prediction accuracy (RMSE) and computational effort. Unfortunately, the authors do not specify on whether the results refer to the new user or to the new item problems so, to benchmark against our tests, we would assume it refers to a mixed case. The RMSE they obtain with the various methods range between 0.961 and 0.935 as summarised in Table 6.7.

Methods	RMSE
SVD	0.961
KNN	0.941
SVD-KNN	0.946
HBY-SVD-KNN	0.935

Table 6.7: RMSE obtained by [1]

	Baseline	Numerical only	Categorical only	Cat and Num Stero.
RMSE	0.921	0.910	0.910	0.910
MAE	0.727	0.715	0.716	0.716

Table 6.8: Performance metrics XGBoost regression for the mixed case

For a direct comparison with [1] we ran our model for a mixed case (new user and new item) using the XGBoost algorithm. The results are reported in Table 6.8. As shown comparing Tables 6.7 and 6.8 the combined use of stereotypes and the XGBoost algorithm led to an increase in prediction accuracy of 2.5% (0.935-0.910) RMSE compared to the best model proposed by [1]. Unfortunately, we are not able to also compare the performances in terms of computational costs as the machine used for the computation have different performances and there are not enough details on the paper to be able to create an estimate of how the CPU time that we have measured in our experiment could rescale in terms of those experienced by Spiegel et al. [1]. However, in the next section we are implementing alternative solution methods including the SVD, to evaluate the effectiveness of the stereotypes in improving the prediction accuracy.

6.1.3 Cold Start Assessment of Recommendations Driven by Stereotypes versus SVD-Based RS (with metadata)

The previous sections show a broad evaluation of the benefits of introducing stereotypes over the original metadata during cold start. Matrix-factorisation techniques, and in particular SVD and SVD++ methods, have gained substantial popularity. For completeness, a comparison of such techniques with the stereotype-driven approach is necessary. The standard, classic, formulation of the SVD algorithm does not include the user or item metadata. However it was shown in the previous section that without such information, particularly during cold start phases, the recommendations provided would be extremely naive. Therefore a fair comparison with SVD/SVD++ should incorporate the user and item metadata in the factorisation procedure. For this comparison, however, there is no available off-the-shelf solution in Python. To this end we used a new Python library called Surprise[2]. The SVD/SVD++ methods as implemented in Surprise take only a list of user ids, item ids and the corresponding user-to-item ratings as inputs. We needed to extend the basic algorithm to account for the extra implicit knowledge embedded in ratings by using the user and item metadata.

The main ideas behind SVD and SVD++ and the incorporation of metadata are revised first, followed by an in-depth analysis of the recommendation quality of the two approaches (stereoptypes and factorisation methods). The analysis is not limited to recommendation

[2]http://surpriselib.com

accuracy, but attempts an investigation on other aspects and desirable properties of recommendations, such as serendipity.

SVD/ SVD++ with metadata

The intuition behind SVD, and in general matrix-factorisation methods [100], is that there should exist a latent space (P_f), of dimensionality f, that determines how a user rates an item. User-item interactions (i.e. ratings) are modelled as inner products in that space. For example the user u's rating of item i, which is denoted by r_{ui}, can be represented as the inner product of two arrays of length f leading to the estimate:

$$r_{ui} = q_i^T * p_u \tag{6.4}$$

Where each item i is associated with a vector $q_i \in P_f$, and each user u is associated with a vector $p_u \in P_f$. To learn the factor vectors (p_u and q_i), and therefore the latent space representations, the system minimises the regularised squared error on the set of known ratings. Regularisation is typically introduced to avoid overfitting. Two approaches implemented in Surprise to minimise the regularised error are stochastic gradient descent proposed in [182] and alternating least squares (ALS) proposed in [178].

It is important to stress here that these characteristics are latent characteristics, and do not necessarily correspond to the user and item metadata.

Three enhancements have been proposed and applied to equation 6.4 to improve the performance in cold start phases: i. introduce user and item specific biases in the ratings; ii. add user and item metadata; iii. introduce implicit feedback.

Enhancements i. and ii. lead to the decomposition of the ratings as illustrated in equation 6.5:

$$r_{ui} = \mu + b_u + b_i + (q_i + \sum_{a \in A(i)} y_a)^T * (p_u + \sum_{b \in B(u)} y_b) \tag{6.5}$$

where the terms $\mu + b_u + b_i$ represent overall mean, user bias and item bias, respectively. In order to include the item's metadata, a term is added to the right of q_i. The metadata is encoded to a 1 to n encoding giving item i a set of attributes $A(i)$ (for example genres, movie budget, revenue, cast popularity) via the vector $y_a \in P_f$. This latter term represents the effect of the item metadata. In a similar fashion a term for the user metadata representation via a set of attributes $B(u)$ is added to the right of p_u.

To give an example of how item's metadata are incorporated in equation 6.5, lets assume we have 4 users and 5 items, and we are dealing with movies with the rating matrix de-

picted in Figure 6.4, where ratings are on a scale 1 to 5 and a question mark indicates an unknown or unconsumed/unrated item-to-user paring. Suppose that two of the items are movies belonging to the genre comedy, and therefore we want to add the item metadata information: 'genre equals comedy'. We have a vector y_a that is composed of 1 (for comedy movies) and zeros (not comedy movie). We add this vector to q_i as an item factor as indicated in equation 6.5.

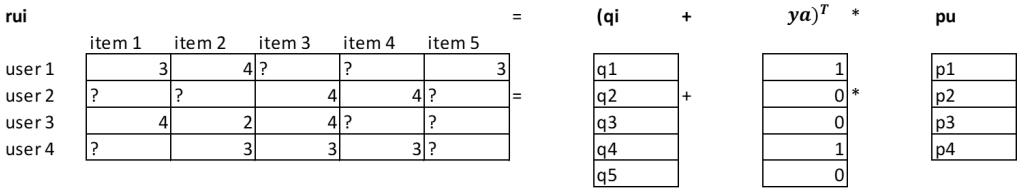


Figure 6.4: Example of adding item factors to Equation 6.5

A third enhancement, that is independent of the two just discussed, is the one that led to the technique called SVD++. As highlighted by [100, 98], recommender systems can use implicit feedback to gain insight into user preferences. Items that a user has not rated have an implicit feedback content. If a user has not rated an item this carries more information than simply not rating, inferring a potential dis-preference. The literature points toward evidence suggesting that incorporating implicit feedbacks improves the prediction accuracy of a recommender system.

SVD/ SVD++ with metadata vs Stereotypes Recommendations: New User and New Item Experiments

In this section the results obtained via SVD with metadata and SVD++ with metadata are compared to the stereotype-based system driven by the XGBoost regression. It is important to note that in the new user experiment, for the 30% of users in the test set we assumed that no rating are available, hence the technique SVD++ is not applicable. It should be noted and remarked that in the current evaluation, the implicit feedback is incorporated for the factorisation-driven RS, *but it is not used* for the stereotype-driven RS.

Table 6.9 shows basic prediction accuracy metrics for both the new user and new item experiments. The stereotype-based model outperforms all of the SVD driven methods in both RMSE and MAE with the only exclusion being the SVD++, i.e. the addition of the implicit feedback information in the new item problem. Note that the stereotypes-driven models do not make use of implicit feedback information. This prove that the introduction of implicit feedback is indeed beneficial as highlighted by [98].

Going past the simplest predictive accuracy metrics, rank accuracy metrics measure the

New User Experiment				
	Stereotype	**SVD without metadata**	**SVD with metadata**	**SVD++ metadata**
RMSE	0.901	0.961	0.924	x
MAE	0.710	0.768	0.736	x
New Item Experiment				
RMSE	0.918	1.059	0.932	0.905
MAE	0.729	0.8607	0.748	0.727

Table 6.9: New user and new item cold start comparisons between the recommendation models: stereotypes and SVD with and without metadata

ability of a recommendation algorithm to produce a recommended ordering of items close to what a user would express. Ranking metrics are more appropriate to evaluate algorithms used in domains where the user's preferences in recommendations are non binary. In what follows the most important rank accuracy metrics, e.g. HR, MRR, MAP, nDCG, HLU as well as serendipity are considered in turn.

Hit Rate. The simplest way to evaluate top-N recommendations is the hit rate (HR), which measures the proportion of successfully recommended items in top-N recommendations. In other words, if a user rated one of the top-N we recommended, we consider it a 'hit'. HR is evaluated at different N (10, 20 and 30) and the results are shown in Table 6.10.

	Model with stereotype		**SVD with features**		***p*-value of difference in HR**	
Hit Rate @ N	**New User**	**New Item**	**New User**	**New Item (SVD++)**	**New User**	**New Item**
HR @ 10	34.7%	23%	26%	20%	< 0.0001	<0.0001
HR @ 20	29.3%	21%	22%	14%	<0.0001	<0.0001
HR @ 30	26.4%	21%	21%	11%	<0.0001	< 0.0001

Table 6.10: Hit rate for top-N recommendation list

As shown in Table 6.10 both for the new user and new item cases the model with stereotypes has a higher percentage of hits in comparison to the SVD with metadata (for the new user problem) and SVD++ with metadata and implicit feedback (for the new item problem). Recall for the new item case, SVD++ with metadata and implicit feedback has a higher RMSE and MAE than stereotype-based model as shown in Table 6.9, however the user satisfaction as measured by the HR is not really improved.

Table 6.10 also shows the *p*-values of the test with null hypothesis: 'The difference of hit rate is not statistically significant'. Given the extremely low *p*-values the stereotype-based models offer statistically significant improvements for both experiments.

Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP). Mean Reciprocal Rank is another measure to evaluate systems that return a ranked list [183]. which takes into account the rank of the position of the *first* correctly identified recommendation. While MRR can be thought of as a score to evaluate only the top hit, the mean average precision (MAP) provides a more suitable measure in the cases where we are interested in the ranking quality of a list rather than just the highest-ranking hit. MAP provides a single summary of the user's ranking preferences as described by [183]. The terminology used is 'MAP @N' to describe how relevant the list of the N recommended items is.

The results for the MRR and MAP in the cold start experiments are shown in Table 6.11. If only the quality of the top hit is examined (MRR) the systems perform in an equivalent manner in the new user case (there is no statistical significant difference). While for the new item case the SVD++ with metadata displays a higher quality of the top hit, suggesting that the use of implicit feedback has indeed valuable information in improving on the quality of the top recommendation. As soon as the focus is extended past the single top recommendation to a basket of recommendations (HR and MAP) then the recommendations provided by the stereotype-based approach are significantly improved over the SVD techniques. This leads to the conclusion that not only stereotype-based model leads to a lower accuracy error but also to *a higher user satisfaction as measured by HR and MAP*.

	Model with stereotype		**SVD with features**		***p*-value of difference**	
	New User	**New Item**	**New User**	**New Item (SVD++)**	**New User**	**New Item**
MRR	66%	51%	66%	55%	0.86	0.03
MAP @ 10	22%	12%	15%	9%	$<$ 0.0001	$<$0.0001
MAP @ 20	17%	10%	11%	6%	$<$0.0001	$<$0.0001
MAP @ 30	14%	9%	9%	4%	$<$0.0001	$<$ 0.0001

Table 6.11: Mean reciprocal rank (MRR) and mean average precision (MAP)

Normalised Discounted Cumulative Gain (nDCG). Normalised discounted cumulative gain (nDCG) is a single-number measure of the effectiveness of a ranking algorithm that allows non-binary judgments of relevance. nDCG uses graded relevance, which is accumulated starting at the top of the ranking and may be reduced, or discounted, at lower ranks [184].

It is the ratio of two measures: discounted cumulative gain (DCG) and Ideal DCG (IDCG). Discounted cumulative gain (DCG) measures the usefulness, or gain, of an item based on its position in the result list. The gain is accumulated from the top of the result list to the bottom, with the gain of each result being discounted at lower ranks. The traditional formula of DCG accumulated at a particular rank position N is defined as, see [184]:

$$DCG@N = \sum_{i=1}^{N} \frac{rel_i}{log_2(i+1)} \tag{6.6}$$

To summarise this value across users, all the items in the top N are sorted by their relative relevance, producing the maximum possible DCG@N, also called Ideal DCG (IDCG). The normalised discounted cumulative gain, or nDCG, is then computed as:

$$nDCG@N = \frac{DCG@N}{IDCG@N} \tag{6.7}$$

Where

$$IDCG@N = \sum_{i=1}^{rel_N} \frac{2^{rel_{i-1}}}{log_2(i+1)} \tag{6.8}$$

The example will explains how nDCG is calculated. Suppose a list of 10 recommended items i (e.g. movies), where only the first, fourth, sixth and eight are relevant (e.g. of interest) for a specific user. Following equation 6.6 we can calculate DCG@10 as in Table 6.12. In order to evaluate nDCG we have to evaluate the maximum possible DCG i.e. IDCG. To do so we sort the list by relevance, see Table 6.13, and we use equation 6.8 to evaluate IDCG. Finally, we use equation 6.7 to calculate nDCG@10 for that specific user that is equal to nDCG@10 = $\frac{DCG@N}{IDCG@N} = \frac{2.10}{2.56} = 0.82$

Note that in a perfect ranking algorithm, the nDCG will be the same as the IDCG producing an nDCG of 1.0. nDCG calculations span from a value of 0.0 (no match) to 1.0 (perfect ranking).

The results obtained for the two cold start scenarios comparing the stereotypes-based models and the matrix-factorisation with metadata based models are reported in Table 6.14. nDCG confirms the results obtained with the other ranking metrics analysed by measuring how useful are our recommendation in average to users. The model with stereotypes outperforms the SVD with metadata for the new user case with a level of confidence well over 98% across all the nDCG tests. Instead, on the new item case, the stereotype-based system performs in a slightly poorer manner. As N grows the statistical confidence on the nDCG of the SVD++ outperforming that of stereotypes wanes. Once again, as seen on MRR, we attribute the slightly better performance on the new item to implicit feedback. By giving a score of 0 to items that no users in the training sample watched prevent these items from being recommended and this is what we believe drives this result. For the new item case this comparison highlights the importance of introducing the implicit feedback that would form part of the future research.

DCG@10			
i	rel_i	$\log_2(i+1)$	$\text{rel}_i/\log_2(i+1)$
1	1	1.00	1.00
2	0	1.58	-
3	0	2.00	-
4	1	2.32	0.43
5	0	2.58	-
6	1	2.81	0.36
7	0	3.00	-
8	1	3.17	0.32
9	0	3.32	-
10	0	3.46	-
DCG@10			2.10

Table 6.12: Example evaluation of DCG@10 following equation 6.6

IDCG@10				
i	pos	rel_i	$\log_2(i+1)$	$2^{rel_{i-1}}/\log_2(i+1)$
1	1	1	1.00	1.00
4	2	1	1.58	0.63
6	3	1	2.00	0.50
8	4	1	2.32	0.43
2	5	0	2.58	-
3	6	0	2.81	-
5	7	0	3.00	-
7	8	0	3.17	-
9	9	0	3.32	-
10	10	0	3.46	-
IDCG@10				2.56

Table 6.13: Example evaluation of IDCG@10 following equation 6.8

Half-Life Utility Metric (HLU). The HLU was introduced by Breese et al. [36] on the premise that a user presented with a ranked list of results, is unlikely to browse deeply into the list. The HLU evaluation metric postulates that the probability of a user selecting a relevant item drops exponentially as they move further down the list. The metric examines an unbounded recommendation list containing all the items. Given such a list, an item at position j has a probability of $\frac{2^{(j-1)}}{(\alpha-1)}$ of being selected, where α is a half-life parameter, specifying the location of the item in the list.

The utility is defined as the difference between the user's rating for an item and the 'default rating' for an item [36], with the default rating generally assumed to be a neutral or slightly negative rating. The expected utility of recommendations given to user u, R_u is represented in equation 6.9. r_{uj} represents the rating of user u on item j of the ranked list, d is the default rating, and α is the half-life factor (or decay factor).

	Model with stereotype		SVD with features		*p*-value of difference	
nDCG @ N	**New User**	**New Item**	**New User**	**New Item (SVD++)**	**New User**	**New Item**
nDCG @ 10	61%	49.2%	57%	52%	0.012	0.001
nDCG @ 20	66%	49.2%	55%	51%	< 0.0001	0.01
nDCG @ 30	60%	49.7%	54%	51%	<0.0001	0.13

Table 6.14: Comparison nDCG for model with stereotype and SVD with metadata

$$R_u = \sum_j \frac{max(r_{uj} - d, 0)}{2^{(j-1)/(\alpha-1)}} \tag{6.9}$$

Figure 6.5 display HLU for the new user and new item experiments for the stereotype model using a decay factor α ranging between 3 and 10 and a default rating equal to the median rating in the dataset (3 in the data used). The half-life utility increases with the decay factor i.e. with the assumption that a user is also interested in items further down the list.

Table 6.15 shows the comparison between the half-life utility value at a decay factor equals to 3 using the model with stereotypes versus the SVD with metadata. While for the new user case we can assert that the model with stereotypes outperforms the SVD model with metadata with more than 95% of confidence, for the new item case the HLU values are much closer and given the *p*-values we cannot assert that the values are statistically different. This behaviour can be ascribed once more to the presence of implicit feedback.

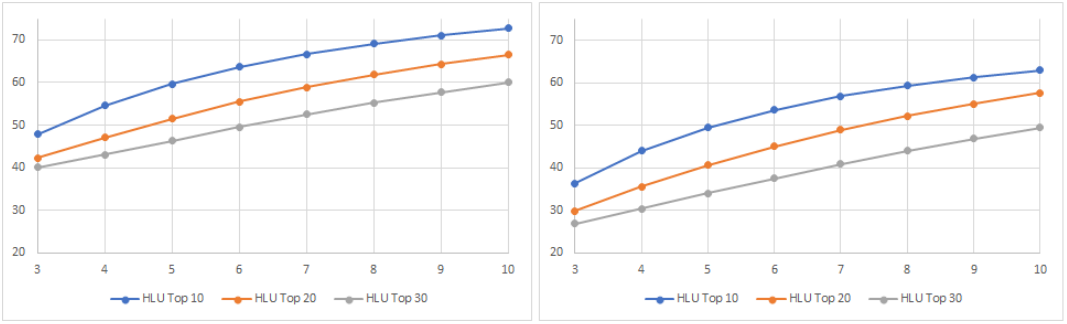


Figure 6.5: Half-life utility (R) new user and (L) new item cases as a function of the α decay factor (x-axis)

Serendipity. Various definitions are proposed for this concept in the recommender system domain. For example, serendipity is defined as a measure of the extent to which the recommended items are both attractive and surprising to the users [35]. To date, various definitions and evaluation metrics to measure serendipity have been proposed, and there is no wide consensus. Kotkov et al. [24] have provided a comprehensive review of the

	Model with stereotype		SVD with features		*p*-value of difference in HLU	
HLU @ N	**New User**	**New Item**	**New User**	**New Item (SVD++)**	**New User**	**New Item**
HLU @ 10	47.917	36.306	44.324	32.444	0.0353	<0.001
HLU @ 20	42.306	29.889	38.927	28.719	0.0340	0.1127
HLU @ 30	40.096	26.895	35.289	26.812	0.0014	0.5840

Table 6.15: Comparison half-life utility for model with stereotypes and SVD with metadata using a decay factor α equal to 3

various definitions and challenges. They suggested that the definition should adapt to the field of application.

Using stereotypes created independently of the ratings to classify user preferences enable the discovery of associations between items that are not necessarily obvious or expected. Given that stereotypes are not generated using rating matrices, such association may carry an element of novelty/serendipity when recommended. In this experiment a proxy for serendipity can be selected by finding how variegate the metadata features were in the top-N recommendation lists.

In the ML/IMDb data under examination, one can think of genre as the perfect example to introduce a measure of serendipity. It is easy to agree on the fact that when selecting a movie, the genre(s) of the movie play a key role in the selection. If a system obtained high prediction accuracy but did so by recommending always the same genre to a given user that has been stereotyped (e.g. male in his 40s like only thriller and action), then recommendations would not be very variegate despite the high accuracy that may have been achieved. One complication is constituted by the fact that there are some items that are categorised as belonging to many genres.

It is possible to argue that there is more information content into an item with a well specified label (for example movie A: drama) than an item categorised with many labels (many genres, for example movie B: drama, war, history, romance, documentary). The weight of a movie in representing a genre should be inversely proportional to the number of labels used. For example, a movie categorised as representative of all 24 genres would add a weight of 1/24 to each genre (a movie with many genre does not represent any single genre highly). With this in mind one can compute for all the items in the top-N list, the sum of the weight contributions to each label (genre in our example).

A recommender system will be more novel/serendipitous than another recommender system if its top-N list will cover more of those labels. A parameter k is introduced to represent the minimum value of the score required to claim that its corresponding genre label was represented. For example suppose one considers top-N with N= 10, and suppose one finds that with the score introduced above the genre label comedy has a score of 0.5. That

means that comedy could have been 1 of the two genres of an item recommended or maybe that comedy was 1 of 4 of the 4 genres in two of the 10 recommendations. At some low value the score should discarded, and only the genres whose score is above k should be considered as being represented in the top-N items.

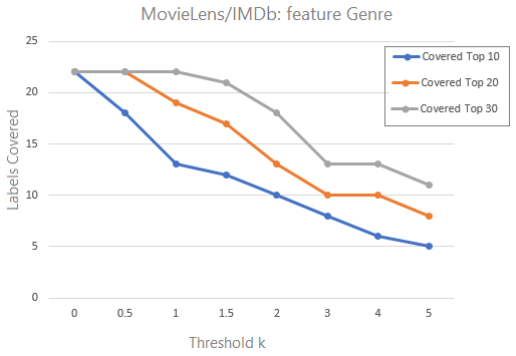


Figure 6.6: Genre diversity (number of distinct genres recommended for the model with stereotype)

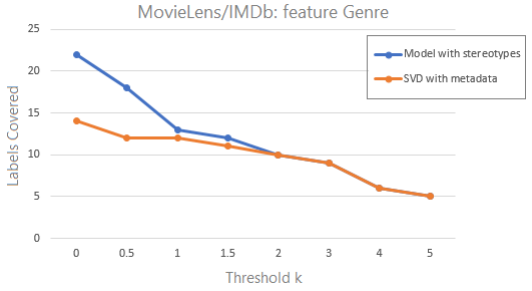


Figure 6.7: Comparison genre diversity for the models: stereotype and SVD with metadata

Figure 6.6 shows the number of genres represented in the top-N recommendation list (y-axis) as a function of the growing value of k (the genre significance cut off, x-axis) for the stereotype-based models. To note that genre has 24 labels in total. Figure 6.6 shows the results obtained for the model with stereotypes for the new user case for 3 different N values, namely Top 10, 20 and 30. The parameter k ranges between 0 (i.e. no limit, all the genres associated with the recommended movies, no matter how small their weight, are counted as being represented) and 5 (i.e. only genre that have a weighted score above 5 are counted).

Figure 6.7 shows the comparison in genre diversity for the top 10 recommendation lists produced by the model with stereotypes and the SVD with metadata. If one agrees that a low k value should be in the region of 0.5 to 1 as intuition suggests (for instance for top 10 a k=1 would mean that there must be at least 1 movie that represents such genre

in full, or perhaps two movies with such genre represented in half, etc.) then the model with stereotypes outperforms the SVD model in this proxy of novelty/serendipity. The two tend to align for higher values of k as expected. These findings allow us to conclude that a stereotype-based recommendation should be more serendipitous than (or at least equally serendipitous as) SVD-based recommendations without sacrificing the quality of recommendation.

6.2 Summary

The presented research so far solve the gap that we found in the literature which are: i) extension of the concept of stereotype to complex categorical features, ii) automation of the procedure to generate stereotypes independently of the RS algorithm chosen and for any feature type, and ultimately iii) the possibility to build stereotypes independently of users-to-item preferences, in order to discover patterns of preferences that may be otherwise lost in the usual clustering approaches driven by rating matrices.

In Sections 6.1.1 and 6.1.2, we demonstrated how the dimensionality reduction performed by automated item-based stereotyped, in a rating agnostic context, not only does not degrade performance, but improve the performance of the recommendations compared to a baseline system that uses primitive item metadata in the new user and new item problems. These experiments answer the research question:

- *Can automatically constructed item-based stereotypes improve recommendations during the cold-start phases?*

The results demonstrate that the improvements, which might not be noticeable in general precision metrics, are indeed present in the metrics that matter most in predicting consumption in the case where the consumed class is rare compared to the catalogue, hinting that the dimension-reduction process intrinsically embeds elements of increased predictability during cold-start. This can be viewed as supporting evidence toward the use of stereotypes in cold start phases.

Most importantly, the added precision obtained by using stereotypes does not seem to depend on the regression model used, suggesting that stereotypes offer an extra dimension for improving the quality of recommendations (at least in cold-start phases) that is independent of the rating-prediction algorithm used. This finding is one of the most important findings of the research justifying the use of stereotypes in the RS community as an extra 'direction' for improvement during cold-start phases.

In Section 6.1.3 we have conducted an in-depth comparison of the recommendations produced by a stereotype-based RS and a SVD-based RS in which we have embedded the user and item metadata. We have shown how stereotypes-based recommendations improve the

rating prediction accuracy metrics (MAE and RMSE) compared to SVD/SVD++ with item and user features. Thus we answered the research question:

– *How do stereotype-based item models compare to other state-of-the-art recommender systems models in terms of predictive performance during the cold-start phases?*

It is important to note that the MovieLens 1M dataset has basic user demographic features and somehow those features (i.e. gender and age) were already stereotyped, i.e. divided in generic groups in the original data.

We demonstrated how stereotypes driven recommendations have a superior performance, with an overall high statistical confidence, according to the most widespread metrics for evaluation of ranked list; for each metric we present the rationale of the metric and discuss the results. For the case of 'serendipity', where there is little agreement in the RS community on how to quantitatively frame such concept, we introduce a novel definition of serendipity (or list variety) that fits well the scope of item complex categorical features - which are to be seen often as the most descriptive features of items metadata. Also according to our definition of serendipity we obtain further evidence corroborating the adoption of stereotypes in RS addressing cold starts. This answers our last research question:

– *Apart from recommendation accuracy, what are the other benefits of item-based stereotypes?*

To generalise the findings, the next chapter demonstrates how the methodology is applicable to a different dataset from the retail sector.

CHAPTER 7

Validation of the Stereotype-Driven Methodology

The Amazon product dataset[1] is the second dataset that will be used to further validate the stereotype-based approach for cold start recommendations. The item features embed the typical complexity that can be encountered in the stereotyping process in modern data, with complex categorical, categorical and numerical features. The dataset contains product reviews and metadata from Amazon including 142.8 million reviews spanning the period from May 1996 to July 2014. Amazon.com has been the subject of intensive investigations, due to the extremely high sparsity, both in the normal recommendation context [127, 185], and in cold-start situations [84]. Section 7.1 will discuss more on this dataset with the focus on the two groups 'Sport and Outdoors' and 'Clothing, Shoes and Jewellery'.

The settings and results of using the Amazon preferences dataset are explained in this chapter. The experiments in Section 7.2 were carried out by using the proposed algorithms as described in Chapter 4. The evaluation of item-based stereotypes in Section 7.3 is following the statistical tests described in Chapter 5. Lastly, Section 7.4 describes the results of using stereotypes in RS and benchmarks the results against the SVD method.

In particular, this chapter aims to further validate our answers to the research questions:

– *Can item-based stereotypes, not based on rating, be constructed automatically?*

[1] http://jmcauley.ucsd.edu/data/amazon/links.html

- *Can automatically constructed item-based stereotypes improve recommendations during the cold-start phases?*
- *How do stereotype-based item models compare to other state-of-the-art recommender systems models in terms of predictive performance during the cold-start phases?*
- *Apart from recommendation accuracy, what are the other benefits of item-based stereotypes?*

7.1 Amazon Dataset

The Amazon product data[2] is selected as a second dataset to further validate the stereotype-based approach for cold start recommendations. It is focused on retail purchases and it is used in modern recommendation research [127, 84, 43, 185, 186].

The dataset contains product reviews and metadata from Amazon including 142.8 million reviews. For convenience, given the size of the full database (i.e. over 20GB compressed) the dataset is also divided in smaller datasets for independent product categories such as books, movies & TV and electronics. The focus has been restricted to two groups: 'Sport and Outdoors' and 'Clothing, Shoes and Jewellery'. The reasons for selecting these two groups are twofold, on one hand the data for these two groups is sufficiently clean, and on the other hand given the validation objective that this second investigation is pursuing, it is believed that taking on retail items that are very different from movies would give more breadth to the current research.

For the items available in these two categories, the dataset includes reviews and product metadata. However, no user/reviewer data such as age, gender, or location is available.

The item metadata contains the following fields:

- asin - ID of the product, e.g. 0000031852
- title - name of the product
- price - price in US dollars (at time of crawl)
- imUrl - url of the product image
- related - related products (also bought, also viewed, bought together, buy after viewing)
- salesRank - sales rank information (numeric).
- brand - brand name

[2] http://jmcauley.ucsd.edu/data/amazon/links.html

– categories - list of categories the product belongs to, e.g. Sports & Outdoors, Hunting & Fishing, Fishing, Accessories, Charts & Maps

Data issues (mostly missing data) were found to affect all dataset groups. For example, for the 'Sport and Outdoors' dataset, which is composed by 530,000 distinct products, while the 'categories' feature is available for all the products in the dataset, price and brand are available only for a subset of products (54% and 28% respectively) as reported in Table 7.1.

The same is applied to the second group under investigation: 'Clothing, Shoes and Jewellery'. In addition, the feature sales rank contains multiple ranks (such as the rank in Sport & Outdoors, and Music even if the item does not strictly belong to the Music group). For consistency, our stereotypes will be calculated only on the sales rank for its respective group, in the example Sport & Outdoors.

The number of products for which the three metrics (price, brand and sales rank) are all available is reported in the last row of the Table 7.1. This dataset will attest the effectiveness of the proposed approach and generalise the results obtained in a completely different domain. The statistics of this dataset is summarised in Table 7.2.

Feature	**Sport & Outdoors**		**Clothing, Shoes and Jewellery**	
	Products	**Percentage**	**Products**	**Percentage**
Total	530,000	100%	1,503,384	100%
Price	287,792	54%	574,882	38%
Brand popularity	150,380	28%	96,730	6%
Sales Rank	370,419	70%	657,651	44%
Combined	113,276	21%	38,803	3%

Table 7.1: Proportion of products with features available for Amazon dataset

Dataset	**Amazon** Sport & Outdoors	**Amazon** Clothing, Shoes & Jewellery
no. of Items	530,000	1,503,384
no. of Users	2,000,000	3,000,000
no. of Rating	3,268,695	5,748,920

Table 7.2: Statistics of Amazon dataset

7.2 Constructing Item-Based Stereotypes

An algorithm for the automatic identification of stereotypes for both categorical and numerical features was proposed in Chapter 4 and then used on the ML/IMDb dataset. For the Amazon dataset the same automated procedure has been used to define the stereotypes.

The results are discussed in the following sections. In order to validate the automatically generated stereotypes and then use them in a recommender system, the first step is to divide the data in test and train datasets. The partition chosen consists of taking 70% of the item metadata as the training set and 30% as a test set, the two are selected via random sampling.

7.2.1 Stereotypes for Complex Categorical Features

In the dataset, the feature 'categories' is composed of multiple aggregated lists of categories and requires de-vectorisation and cleaning. For example, a product can be in categories [Sports & Outdoors, Accessories, Sport Watches], [Clothing, Shoes & Jewellery, Sport Watches] and [Clothing, Shoes & Jewellery, Men, Watches, Wrist Watches]. In order to recreate correlation-based stereotypes the data needs to be pre-processed first by eliminating group entries and exploding such entries into a list of unique categories (de-vectorisation). In the example above the pre-processing leads to the following distinct and unique labels: [Sports & Outdoors, Accessories, Sport Watches, Clothing, Shoes & Jewellery, Men, Watches, Wrist Watches]. This approach, when applied to the data, results in a relative high number of unique labels. In order to reduce the number of clusters and to remove the noise introduced by categories that appear in a relatively small number of products only, the categories with more than 2% of the products were retained. For the 'Sport and Outdoors' product group, this resulted in selecting 72 distinct categories. For the 'Clothing, Shoes and Jewellery' product group, the application of such a filter resulted in selecting 79 distinct categories.

For the two product groups chosen, the feasibility of using the stereotyping algorithm 1 proposed in Chapter 4 has been explored by generating and investigating the correlation matrixes of the filtered feature 'categories' for the training dataset. The correlation matrices have been grouped via the greedy search algorithm that was developed earlier in Chapter 4. Figures 7.1 and 7.2 show the correlation matrices for the feature 'categories' in the 'Sport and Outdoors' and 'Clothing, Shoes and Jewellery' product groups, respectively.

The correlation matrices in Figures 7.1 and 7.2 display well defined area of positive correlation (denoted by the green colour), as well as negative correlation (denoted by the red colour). Such behaviour (present in both product groups) indicates that the stereotyping procedure, based on the hierarchical clustering of the correlation matrix and subsequent dendrogram truncation should work well for these datasets and should lead to stereotypes that effectively reduce the dimensionality of the problem. The typical correlation value found in the matrices in the off-diagonal terms tend to be low, on average less than 0.4;

therefore as suggested in Section 4.1, the algorithm 1 for the generation of stereotypes will use the absolute value dissimilarity metric defined in equation 4.4.

Figures 7.3 and 7.4 display the dendrograms resulting from the hierarchical clustering of the correlation matrices using the absolute dissimilarity metric defined in Equation 4.4 for the product categories. These figures also confirm that there are clearly identifiable and segregated groups of labels that appear to be well suited for the stereotyping algorithm. Proceeding with the algorithm of Section 4.1 Figures 7.5, 7.6, 7.7 and 7.8 show the average cluster size, and the number of clusters formed up to a given iteration, for the 'Sport & Outdoors' and 'Clothing, Shoes & Jewellery' product group, respectively.

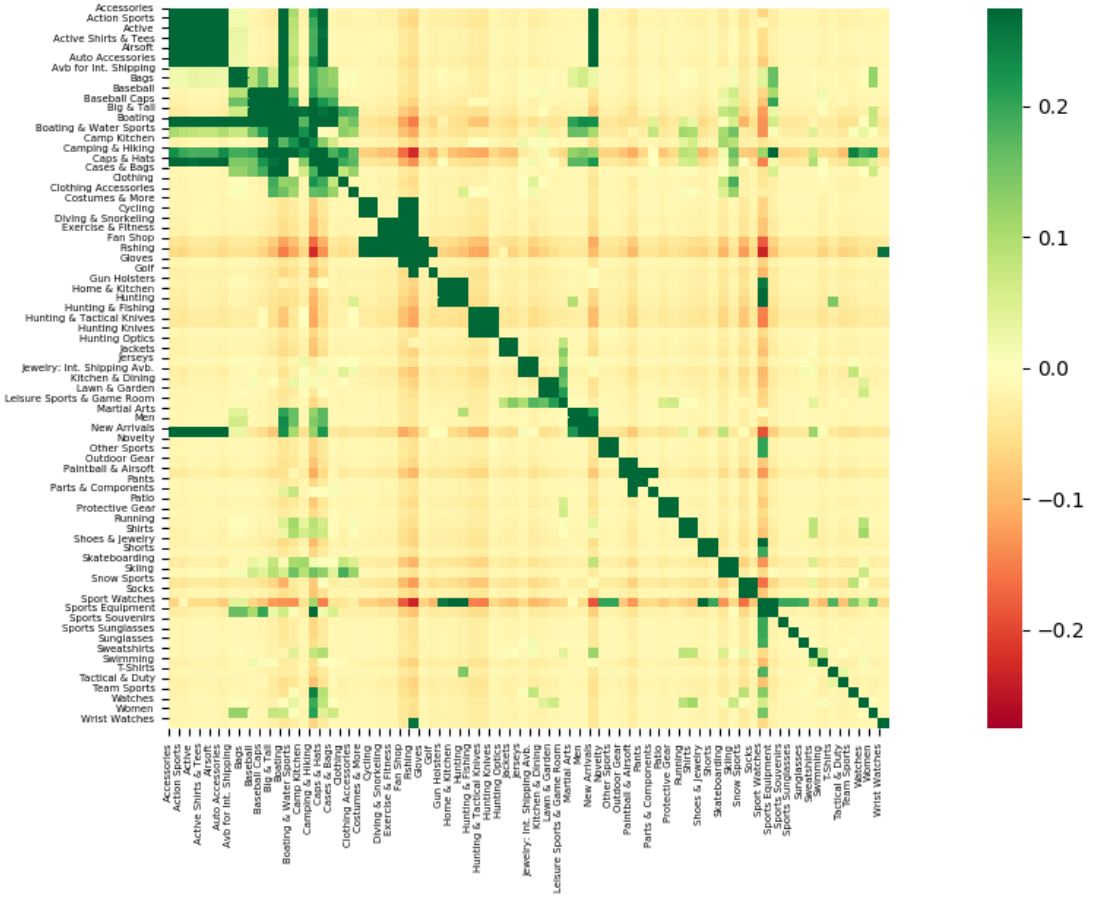


Figure 7.1: Correlation matrix for the complex categorical feature 'categories' in Amazon product group: 'Sport and Outdoors'

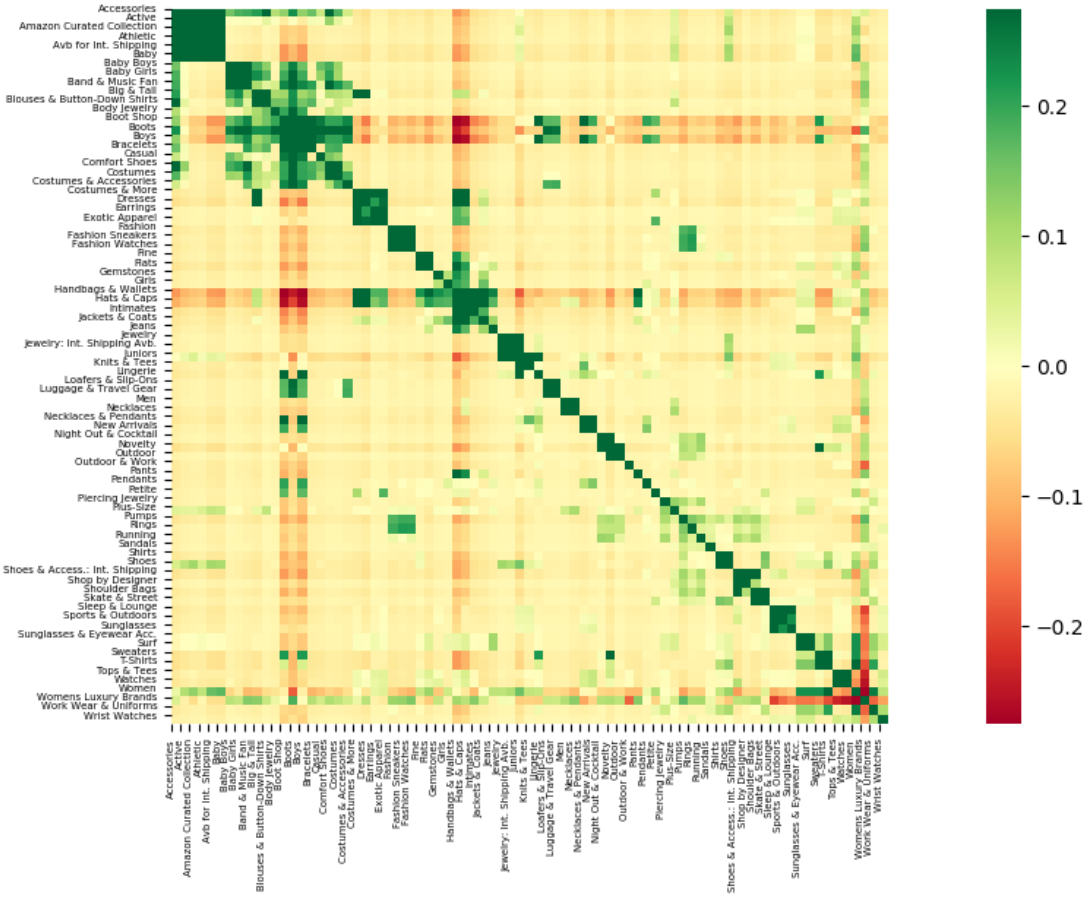


Figure 7.2: Correlation matrix for the complex categorical feature 'categories' in Amazon product group: 'Clothing, Shoes and Jewellery'

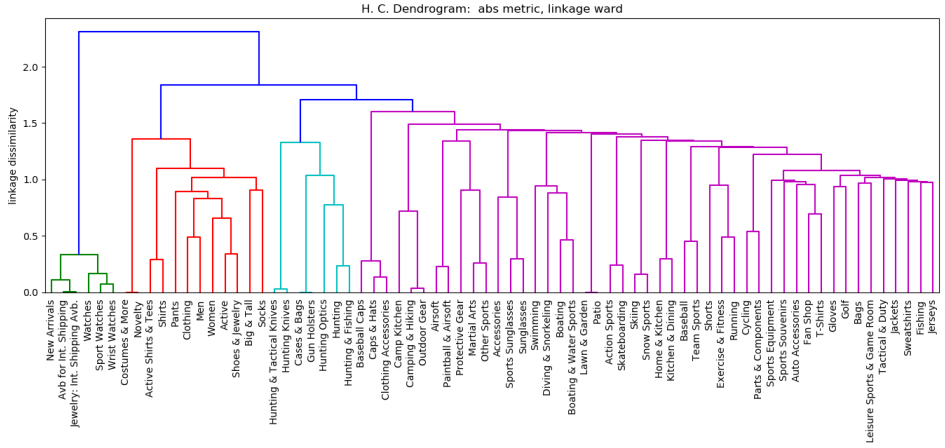


Figure 7.3: Dendrogram for the hierarchical clustering of 'categories' feature in Amazon product group: 'Sport and Outdoors'

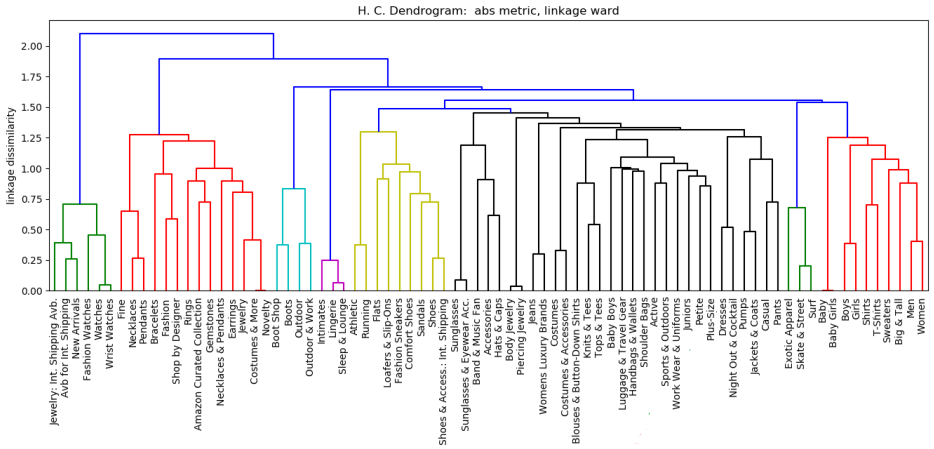


Figure 7.4: Dendrogram for the hierarchical clustering of 'categories' feature in Amazon product group: 'Clothing, Shoes and Jewellery'

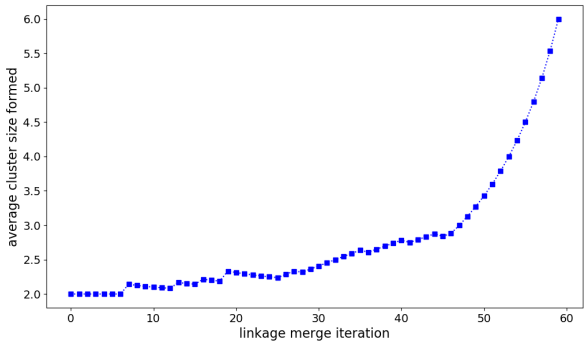


Figure 7.5: Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Average cluster size over the clusters formed at each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward. Amazon product group: 'Sports & Outdoors'

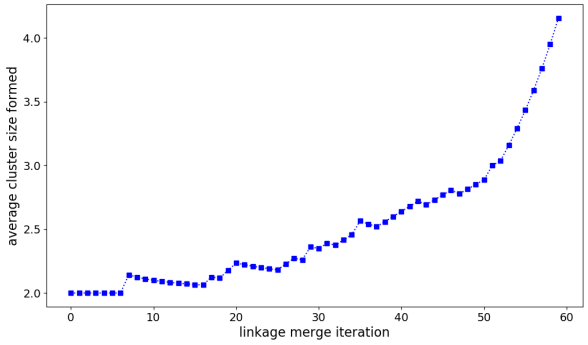


Figure 7.6: Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Average cluster size over the clusters formed at each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward. Amazon product group: 'Clothing, Shoes & Jewellery'

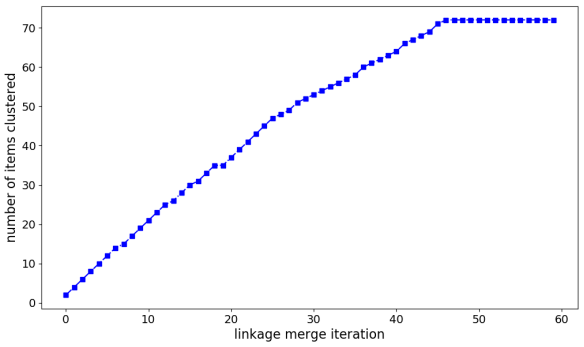


Figure 7.7: Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Total number of clusters present after each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward. Amazon product group: 'Sports & Outdoors'

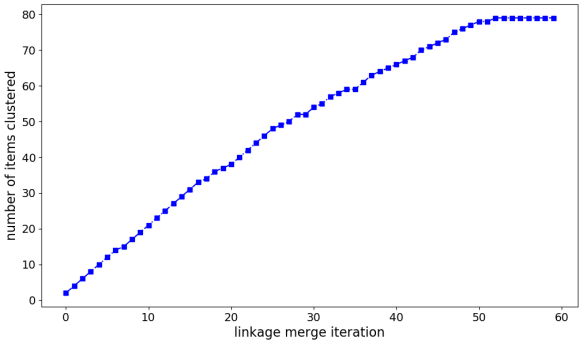


Figure 7.8: Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Total number of clusters present after each iteration of the linkage merge. Dissimilarity metric 4.4, linkage criterium Ward. Amazon product group: 'Clothing, Shoes and Jewellery'

As discussed in Chapter 4, the two criteria (i.e. the average cluster size and the number of clusters formed up to a given iteration) are coupled by taking the ratio, at any iteration, of the average cluster size divided by the number of clusters, a quantity that is referred to as the dendrogram iteration ratio. This quantity is shown in Figures 7.9 and 7.10 for the two product groups. The figures also show the point at which the algorithm 1 developed in Chapter 4 suggests to cut the dendrogram.

Tables 7.3 and 7.4 show the stereotypes obtained for the 'categories' feature of the two product groups. It is possible to see how the procedure identifies many strong relationships that we can agree with as humans. For example the associations of Lawn & Garden with Patio, or that of Caps & Hats with Clothing Accessories, as well as some relationships that are embedded in the data but as humans it would have been less obvious to discover,

when not even counter intuitive, like for example the association of Watches with Martial Arts.

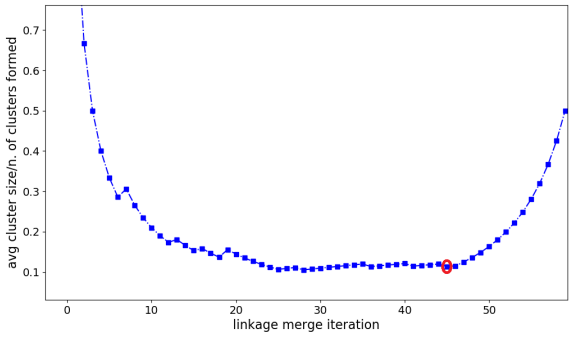


Figure 7.9: Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Dendrogram iteration ratio using dissimilarity metric 4.4, linkage criterium Ward. The red circle indicates the local minimum which is most to the right. Amazon product group: 'Sports & Outdoors'

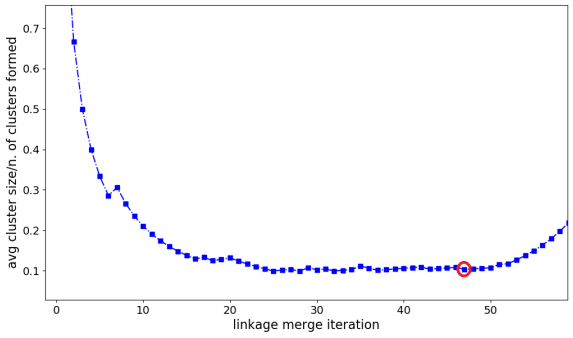


Figure 7.10: Categories feature hierarchical cluster of the correlation matrix, assembly iterations. Dendrogram iteration ratio using dissimilarity metric 4.4, linkage criterium Ward. The red circle indicates the local minimum which is most to the right. Amazon product group: 'Clothing, Shoes & Jewelry'

7.2.2 Stereotypes for the Numerical Features

Each product (identified by the number in column 'asin') has numerical metadata features such as 'price' and 'sales rank' as well as a simple categorical feature, 'brand', that can be used to calculate 'brand popularity'. Following the definition of the popularity features in Chapter 3 we can introduce 'brand popularity' as the natural logarithm of the occurrence of a brand within the training dataset. Given that the feature 'sales rank' spans several orders

	Stereotypes - Categories (Order not relevant)
1	[‘Lawn & Garden’,‘Patio’]
2	[‘Cases & Bags’,‘Gun Holsters’]
3	[‘Costumes & More’,‘Novelty’]
4	[‘Hunting & Tactical Knives’,‘Hunting Knives’]
5	[‘Skiing’,‘Snow Sports’]
6	[‘Airsoft’,‘Paintball & Airsoft’]
7	[‘Action Sports’, ‘Skateboarding’]
8	[‘Caps & Hats’, ‘Clothing Accessories’, ‘Baseball Caps’]
9	[‘Active Shirts’ & ‘Tees, Shirts, Sweatshirts’]
10	[‘Home & Kitchen’, ‘Kitchen & Dining’]
11	[‘Sport Watches’, ‘Wrist Watches’, ‘Watches’, ‘Available for International Shipping’, ‘Jewelry: International Shipping Available’, ‘New Arrivals’]
12	[‘Baseball’, ‘Team Sports’]
13	[‘Cycling’, ‘Parts & Components’]
14	[‘Exercise & Fitness’, ‘Running’]
15	[‘Camping & Hiking’, ‘Outdoor Gear’, ‘Camp Kitchen’]
16	[‘Sports Sunglasses’, ‘Sunglasses’, ‘Accessories’]
17	[‘Active’, ‘Shoes & Jewelry’, ‘Women’, ‘Pants’]
18	[‘Martial Arts‘, ’Other Sports‘, ‘Protective Gear’]
19	[‘Gloves’, ‘Golf’]
20	[‘Boating’, ‘Boating & Water Sports’, ‘Diving & Snorkelling’, ‘Swimming’]
21	[‘Clothing‘, ‘Men’, ‘Big & Tall’, ‘Shorts’, ‘Socks’, ‘Jackets’]
22	[‘Hunting’, ‘Hunting & Fishing’, ‘Hunting Optics’, ‘Fishing’, ‘Tactical & Duty’]
23	[‘Bags’, ‘Leisure Sports & Game Room’, ‘Jerseys’]
24	[‘Fan Shop’, ‘T-Shirts’, ‘Auto Accessories’, ‘Sports Souvenirs’, ‘Sports Equipment’]

Table 7.3: Stereotypes automatically generated using algorithm 1 for the feature: Categories. Amazon product group: ‘Sports & Outdoors’

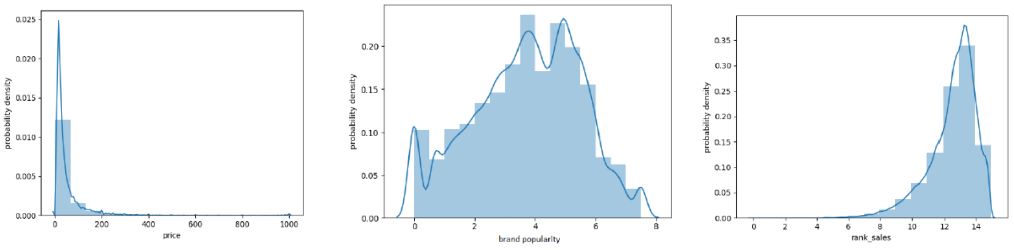
of magnitude, for example from item ranked no.1 to item ranked no. 25,800 it is sensible to take the natural logarithm of such a feature.

In order to group the numerical features into stereotypes, a preliminary examination of the probability distribution of each feature needs to be carried out. Figures 7.11 and 7.12 display three features for the two product groups with two approximate representations for each individual feature’s probability distributions: a standard histogram representation and a kernel density estimation (KDE) [164] performed using a gaussian kernel.

By applying the criterion presented in Section 4.2 to distinguish between features of Type I (i.e. features whose distribution is multi-modal and for which a set of modes can be used to define stereotypes) and features of Type II (i.e. features whose distribution does not allow one to define stereotypes via the use of modes, or where the modes identified are not significative) it was discovered that all the numerical features in the Amazon ‘Sports & Outdoors’ dataset are categorised as Type II. The same applies to the ‘Clothing, Shoes and Jewellery’ dataset where all the numerical features fall within Type II.

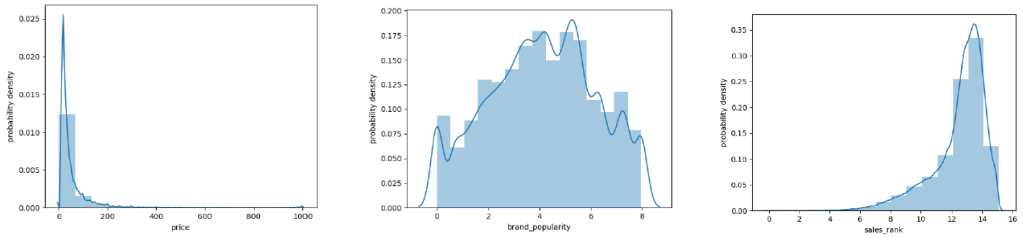
	Stereotypes - Categories (Order not relevant)
1	[‘Skate & Street’, ‘Surf’, ‘Exotic Apparel’]
2	[‘Body Jewellery’, ‘Piercing Jewellery’]
3	[‘Sunglasses’, ‘Sunglasses & Eyewear Accessories’]
4	[‘Lingerie’, ‘Sleep & Lounge’]
5	[‘Handbags & Wallets’, ‘Shoulder Bags’, ‘Luggage & Travel Gear’]
6	[‘Costumes’, ‘Costumes & Accessories’]
7	[‘Athletic’, ‘Running’]
8	[‘Shirts’, ‘T-Shirts’, ‘Sweaters’]
9	[‘Dresses’, ‘Night Out & Cocktail’, ‘Pumps’]
10	[‘Active’, ‘Sports & Outdoors’, ‘Work Wear & Uniforms’]
11	[‘Fashion’, ‘Shop by Designer’, ‘Pumps’]
12	[‘Necklaces’, ‘Pendants’, ‘Fine’, ‘Bracelets’]
13	[‘Baby’, ‘Baby Girls’]
14	[‘Boys’, ‘Girls’]
15	[‘Available for International Shipping’, ‘New Arrivals’, ‘Jewellery: International Shipping Available’, ‘Watches’, ‘Wrist Watches’, ‘Fashion Watches’]
16	[‘Casual’, ‘Pants’, ‘Jackets & Coats’]
17	[‘Shoes’, ‘Shoes & Accessories: International Shipping Available’, ‘Sandals’, ‘Comfort Shoes’, ‘Fashion Sneakers’]
18	[‘Boot Shop’, ‘Boots’, ‘Outdoor’, ‘Outdoor & Work’]
19	[‘Jeans’, ‘Women’s Luxury Brands’]
20	[‘Men’, ‘Women’, ‘Big & Tall’]
21	[‘Knits & Tees’, ‘Tops & Tees’, ‘Blouses & Button-Down Shirts’]
22	[‘Costumes & More’, ‘Novelty’, ‘Jewelry’, ‘Earrings’, ‘Necklaces & Pendants’]
23	[‘Amazon Curated Collection’, ‘Gemstones’, ‘Rings’]
24	[‘Accessories’, ‘Hats & Caps’, ‘Band & Music Fan’]
25	[‘Flats’, ‘Loafers & Slip-Ons’]
26	[‘Petite’, ‘Plus-Size’, ‘Juniors’]

Table 7.4: Stereotypes automatically generated using algorithm 1 for the feature: Categories. Amazon product group: ‘Clothing, Shoes and Jewellery’



(a) Price (b) Brand Popularity (c) Sales

Figure 7.11: Probability density approximation via histograms and KDE for the features: Price, Brand Popularity and log (sales rank). Amazon product group: ‘Sports & Outdoors’



(a) Price (b) Brand Popularity (c) Sales

Figure 7.12: Probability density approximation via histograms and KDE for the features: Price, Brand Popularity and log (sales rank). Amazon product group: 'Clothing, Shoes and Jewellery'

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0.000	12.95	0.25
2	-	12.95	23.99	0.25
3	-	23.99	49.97	0.25
4	-	49.97	∞	0.25

Table 7.5: Stereotypes for feature: Price. Amazon product group: 'Sports & Outdoor'

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0.000	2.48	0.25
2	-	2.48	3.83	0.25
3	-	3.83	5.06	0.25
4	-	5.06	∞	0.25

Table 7.6: Stereotypes for feature: Brand Popularity. Amazon product group: 'Sports & Outdoor'

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0.000	11.85	0.25
2	-	11.85	12.85	0.25
3	-	12.85	13.55	0.25
4	-	13.55	∞	0.25

Table 7.7: Stereotypes for feature: log(sales rank). Amazon product group: 'Sports & Outdoor'

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0.000	12.98	0.25
2	-	12.98	23.16	0.25
3	-	23.16	48.97	0.25
4	-	48.97	∞	0.25

Table 7.8: Stereotypes for feature: Price. Amazon product group: 'Clothing, Shoes and Jewellery'

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0.000	2.56	0.25
2	-	2.56	4.14	0.25
3	-	4.14	5.55	0.25
4	-	5.55	∞	0.25

Table 7.9: Stereotypes for feature: log(brand popularity). Amazon product group: 'Clothing, Shoes and Jewellery'

Stereotype	Mode Location (if any)	Lower Interval	Upper Interval	Population %attributable
1	-	0.000	11.85	0.25
2	-	11.85	12.97	0.25
3	-	12.97	13.68	0.25
4	-	13.68	∞	0.25

Table 7.10: Stereotypes for feature: log(sales rank). Amazon product group: 'Clothing, Shoes and Jewellery'

For features whose distribution is not multimodal (i.e. categorised as Type II), stereotypes are generated via percentile driven intervals as shown in Tables 7.5 to 7.10.

This section validated the novel approach for constructing stereotypes automatically and therefore the research question:

Can item-based stereotypes, not based on rating, be constructed automatically?

Before validating the remaining research questions, it is important to evaluate the stereotypes obtained for their stability and accuracy, which will be carried out in the next section.

7.3 Evaluation of Stereotypes

As discussed in the methodology illustrated for the ML/IMDb dataset, the stereotypes constitute the building blocks of the explanatory power of the RS extracted from the user and item metadata. Therefore it is of paramount importance to check that the results obtained are stable and truly representative of real data associations. All the data used up to this stage to perform calculations and derivation of the stereotypes can be labelled as training data, and new unseen data labelled as test data can be used to evaluate the consistency and stability of the relationships discovered. In what follows the stability of the stereotypes generated is evaluated as well as the use of stereotypes in capturing user preference traits is investigated following similar steps as in Chapter 5.

7.3.1 Homogeneity of Training and Test Datasets

The first step to perform in order to proceed with the training/test validation is to confirm that the data is somewhat 'homogeneous', meaning that the statistical distribution of the

features for the two samples is similar. The two simplest driving features that can be checked to confirm or disprove qualitatively the statement that the test data can be seen as representative of the training data are the price distribution and the product categories distribution.

If the distribution of the price of the products in the training and the test sets was very different, for example if the majority of training products were expensive product, above a certain threshold, while the majority of the test products were generally non expensive products, it would be reasonable to assume that there may be differences in the stereotypes generated. Figures 7.13 and 7.14 display the distributions of price for the training and test dataset for the two Amazon product groups, qualitatively confirming that the distributions of the test and training data are not dissimilar enough to expect any effect on the stereotypes inferred on the data.

The second check performed consists in verifying that the distribution of categories in the two datasets is not dissimilar to the point of invalidating the ability to extend conclusions drawn on one set to the other. Figures 7.15 and 7.16 display the occurrence of most frequent category label in the products of both training and test datasets. It is possible to observe how the test dataset is indeed very much representative of the training dataset, with only small variations in the relative occurrence of certain category labels and almost a one to one perfect relative match in frequency and order of the most important labels. In the figures, the histograms of the training data have been sorted, and displayed with the histogram for the same category labels of the test data. Only small variations in relative frequency are recorded and visible in the graphs. It is therefore possible to argue that the test dataset is indeed representative of the training dataset and can be used to infer conclusions about the stereotypes that have been automatically generated.

Figure 7.13: Histogram distribution of the price for the training set (in blue) with 371k products and the test set (in red) with 159k products. Amazon product group: 'Sports & Outdoors'

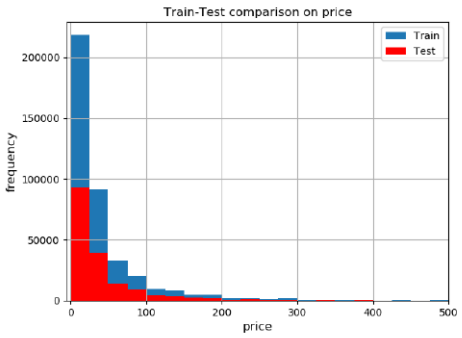


Figure 7.14: Histogram distribution of the price for the training set (in blue) with about 1,052k products and the test set (in red) with about 450k products. Amazon product group: 'Clothing, Shoes and Jewellery'

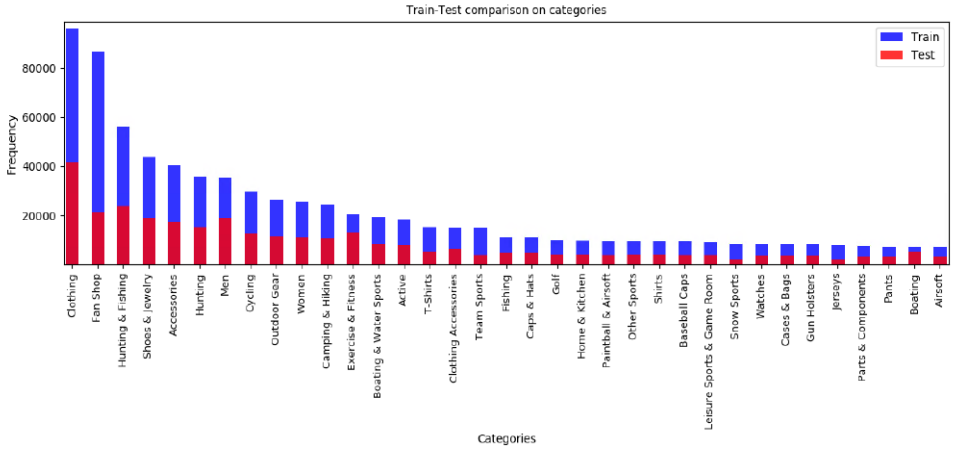


Figure 7.15: Histogram distribution of the 'categories' frequency in the training set (in blue) and the test set (in red). Note that only the most frequent categories are being displayed to facilitate comparison. Amazon product group: 'Sports & Outdoors'

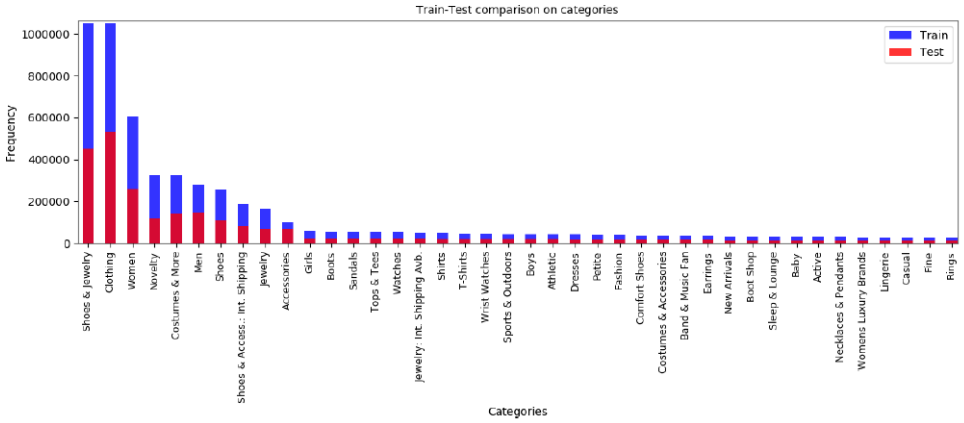


Figure 7.16: Histogram distribution of the 'categories' frequency in the training set (in blue) and the test set (in red). Note that only the most frequent categories are being displayed to facilitate comparison. Amazon product group: 'Clothing, Shoes and Jewelry'

7.3.2 A Hard Test

As discussed in Section 5.2, a hard test is to check if from unseen data - namely the test dataset - the same relationships as those learnt from the training dataset would be discovered in an independent fashion. Then measure how far apart the relationships learnt by the algorithm from both datasets are.

In order to perform the hard test for **complex categorical feature**, Figures 7.17 and 7.18 display the dendrogram for the feature 'categories' over the test datasets derived independently from the training data for two Amazon product groups.

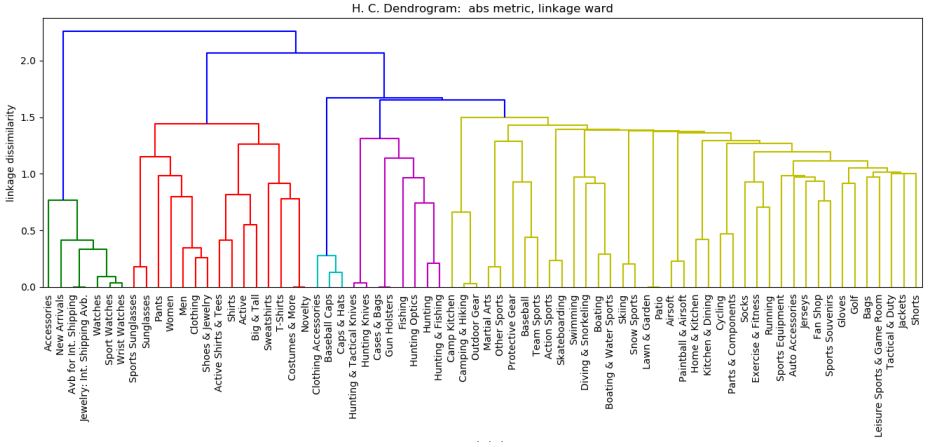


Figure 7.17: Dendrogram for the hierarchical clustering of the 'categories' feature in test dataset. Amazon product group: 'Sports & Outdoors'

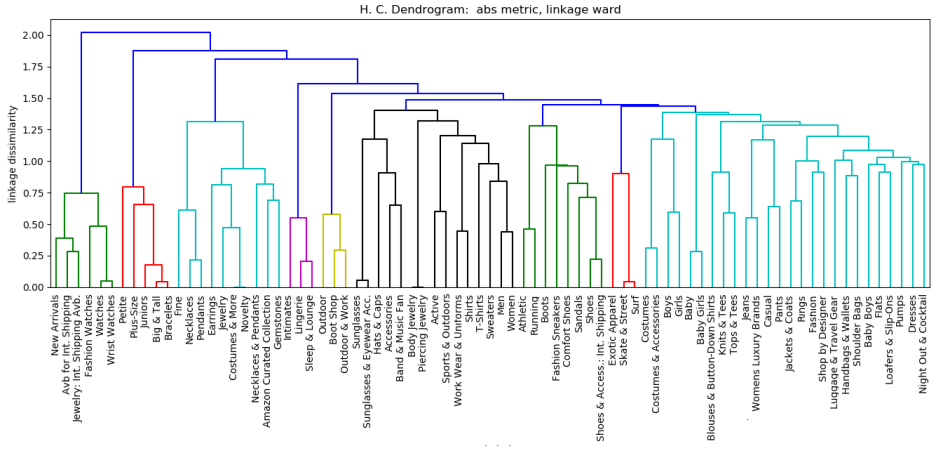


Figure 7.18: Dendrogram for the hierarchical clustering of the 'categories' feature in test dataset. Amazon product group: 'Clothing, Shoes and Jewellery'

For both product groups for both datasets (train and test) the categories composing the

branches, in general, present a very good resemblance between the two, and this qualitative fact will be later confirmed by the stereotype creation and the stereotype hard and soft tests.

Figures 7.19 and 7.20 show the dendrogram iteration ratio, i.e. the metric that allows one to define the iteration at which to cut the dendrogram automatically in order to generate the stereotypes. Figures 7.19 and 7.20 can be viewed as corresponding to Figures 7.9 and 7.10, whilst Figures 7.9 and 7.10 perform the iterations on the training data, and Figures 7.19 and 7.20 perform them on the test data (two disjoint sets). The resulting stereotype elaborated on the test data are reported in Tables 7.11 and 7.12 and can be compared directly to those discovered over the training data in Tables 7.3 and 7.4 respectively.

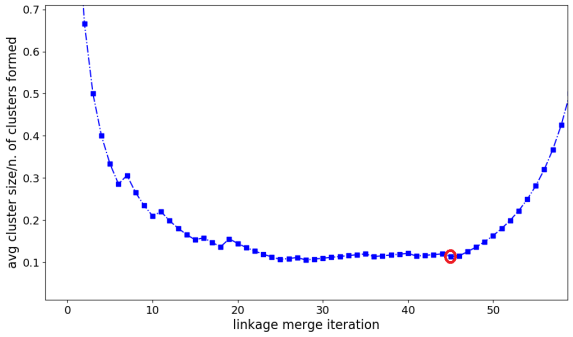


Figure 7.19: Dendrogram iteration ratio using dissimilarity metric 4.4, Linkage criterium Ward for the test dataset. The red circle indicates the local minimum which is most to the right. Amazon product group 'Sports & Outdoors'

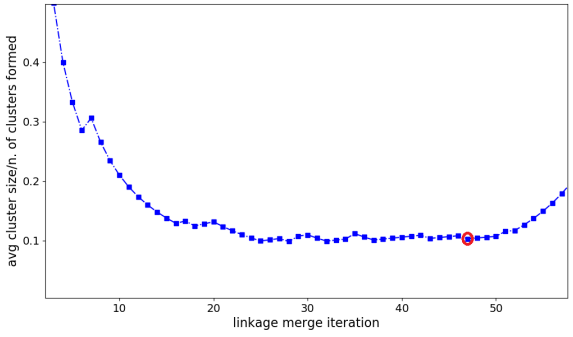


Figure 7.20: Dendrogram iteration ratio using dissimilarity metric 4.4, linkage criterium Ward for the test dataset. The red circle indicates the local minimum which is most to the right. Amazon product group 'Clothing, Shoes and Jewelry'

The objective analysis of the differences between the two sets of stereotypes (those of Tables 7.3 and 7.11, and Tables 7.4 and 7.12) constitutes the main result of the hard test

	Stereotypes - Categories - Test data (Order not relevant)
1	[‘Lawn & Garden’,‘Patio’]
2	[‘Cases & Bags’,‘Gun Holsters’]
3	[‘Costumes & More’,‘Novelty’]
4	[‘Hunting & Tactical Knives’,‘Hunting Knives’]
5	[‘Skiing’,‘Snow Sports’]
6	[‘Airsoft’,‘Paintball & Airsoft’]
7	[‘Action Sports’, ‘Skateboarding’]
8	[‘Sports Sunglasses’, ‘Sunglasses’]
9	[‘Caps & Hats’, ‘Clothing Accessories’, ‘Baseball Caps’]
10	[‘Active Shirts & Tees’, ‘Shirts’, ‘Sweatshirts’]
11	[‘Home & Kitchen’, ‘Kitchen & Dining’]
12	[‘Baseball’, ‘Team Sports’]
13	[‘Cycling’, ‘Parts & Components’]
14	[‘Exercise & Fitness’, ‘Running’]
15	[‘Camping & Hiking’, ‘Outdoor Gear’, ‘Camp Kitchen’]
16	[‘Sport Watches’, ‘Wrist Watches’, ‘Watches’, ‘Available for International Shipping’, ‘Jewelry: International Shipping Available’, ‘New Arrivals’]
17	[‘Active’, ‘Shoes & Jewelry’, ‘Women’, ‘Pants’]
18	[‘Martial Arts’, ‘Other Sports’, ‘Protective Gear’]
19	[‘Gloves’, ‘Golf’]
20	[‘Clothing’, ‘Men’,‘ Big & Tall’, ‘Shorts’, ‘Socks’, ‘Jackets’]
21	[‘Boating’, ‘Boating & Water Sports’, ‘Diving & Snorkeling’, ‘Swimming’]
22	[‘Hunting’, ‘Hunting & Fishing’, ‘Hunting Optics’, ‘Fishing’]
23	[‘Bags’, ‘Leisure Sports & Game Room’, ‘Jerseys’]
24	[‘Fan Shop’, ‘T-Shirts’, ‘Auto Accessories’, ‘Sports Souvenirs’, ‘Sports Equipment’]

Table 7.11: Stereotypes automatically generated using Algorithm 1 for the feature ‘categories’ in the test data. Amazon product group: ‘Sports & Outdoors’

and can be performed using the metric introduced in Equation 5.1. Table 7.13 reports the results of the comparison between the 24 stereotypes for ‘categories’ of the product group Sports & Outdoors. There is 93% accuracy (average match), with a median match of 100% - indicating that in most of the stereotypes there is a perfect one to one match between their composition derived on the training data vs that derived on the test data. The same table reports results for the product group Clothing, Shoes and Jewellery, which exhibit even higher average accuracy

The evaluation of the **numerical stereotypes** can be conducted in a similar fashion to that performed for the complex categorical stereotypes. The formulation is re-adapted to numerical based nature, by adopting definitions of probability mass and location accuracy, mismatch ratios as explained in Chapter 5 in Equation 5.4. The reader is referred to that part of the thesis to refresh the meaning of the various terms and quantities discussed below.

Tables 7.14 and 7.15 display the quantities of the set of Equation 5.4 for the stereotypes of the numerical features for the two product groups under study. For the product group Sports & Outdoors for almost all features the number of stereotypes discovered over the

	Stereotypes - Categories - Test data (Order not relevant)
1	[‘Skate & Street’, ‘Surf’, ‘Exotic Apparel’]
2	[‘Body Jewellery’, ‘Piercing Jewellery’]
3	[‘Sunglasses’, ‘Sunglasses & Eyewear Accessories’]
4	[‘Lingerie’, ‘Sleep & Lounge’]
5	[‘Handbags & Wallets’, ‘Shoulder Bags’, ‘Luggage & Travel Gear’]
6	[‘Costumes’, ‘Costumes & Accessories’]
7	[‘Athletic’, ‘Running’]
8	[‘Shirts’, ‘T-Shirts’, ‘Sweaters’]
9	[‘Dresses’, ‘Night Out & Cocktail’, ‘Pumps’]
10	[‘Active’, ‘Sports & Outdoors’]
11	[‘Fashion’, ‘Shop by Designer’, ‘Pumps’]
12	[‘Necklaces’, ‘Pendants’, ‘Fine’, ‘Bracelets’]
13	[‘Baby’, ‘Baby Girls’]
14	[‘Boys’, ‘Girls’]
15	[‘Available for International Shipping’, ‘New Arrivals’, ‘Jewellery: International Shipping Available’, ‘Watches’, ‘Wrist Watches’, ‘Fashion Watches’]
16	[‘Casual’, ‘Pants’, ‘Jackets & Coats’]
17	[‘Shoes’, ‘Shoes & Accessories: International Shipping Available’, ‘Sandals’, ‘Comfort Shoes’, ‘Fashion Sneakers’]
18	[‘Boot Shop’, ‘Boots’, ‘Outdoor’, ‘Outdoor & Work’, ‘Work Wear & Uniforms’]
19	[‘Jeans’, ‘Women’s Luxury Brands’]
20	[‘Men’, ‘Women’]
21	[‘Knits & Tees’, ‘Tops & Tees’, ‘Blouses & Button-Down Shirts’]
22	[‘Costumes & More’, ‘Novelty’, ‘Jewelry’, ‘Earrings’, ‘Necklaces & Pendants’]
23	[‘Amazon Curated Collection’, ‘Gemstones’, ‘Rings’]
24	[‘Accessories’, ‘Hats & Caps’, ‘Band & Music Fan’]
25	[‘Flats’, ‘Loafers & Slip-Ons’]
26	[‘Petite’, ‘Plus-Size’, ‘Big & Tall’,‘Juniors’]

Table 7.12: Stereotypes automatically generated using Algorithm 1 for the feature: ‘categories’ in the test data. Amazon product group: ‘Clothing, Shoes and Jewellery’

Descriptive Statistics of μ_{ij}	**Sports & Outdoor Value**	**Clothing, Shoes and Jewellery Value**
Average (Accuracy)	93%	96%
Median	100%	100%
Minimum	78%	84%

Table 7.13: Hard test comparison statistics for the stereotypes of the feature ‘categories’ generated over the training vs the test datasets

Feature Name		$\delta P_{sj}^{T,t}$	$\delta X_{sj}^{T,t}$	$\mu_{sj}^{T,t}$	Statistics
price	1	0.01	0.015	0.998	mean accuracy: 0.999 minimum accuracy: 0.998 stdv accuracy: 0.001
price	2	0.0	0.015	0.999	
price	3	0.0	0.010	1.000	
price	4	0.0	0.057	0.999	
brand popularity	1	0.011	0.594	0.850	mean accuracy: 0.875 minimum accuracy: 0.818 stdv accuracy: 0.059
brand popularity	2	0.016	0.802	0.860	
brand popularity	3	0.024	0.724	0.818	
brand popularity	4	0.004	0.620	0.973	
sales rank	1	0.0	0.350	0.943	mean accuracy: 0.985 minimum accuracy: 0.943 stdv accuracy: 0.025
sales rank	2	0.0	0.005	1.000	
sales rank	3	0.0	0.0	1.000	
sales rank	4	0.0	0.005	1.000	

Table 7.14: Hard test metrics for the evaluation of the numerical stereotypes. The table reports for all numerical features and all stereotypes the dissimilarity in probability mass and centre, and the accuracy from equation 5.4. The last column gives standard metric of the accuracy. Amazon product group: 'Sports & Outdoors'

Feature Name		$\delta P_{sj}^{T,t}$	$\delta X_{sj}^{T,t}$	$\mu_{sj}^{T,t}$	Statistics
price	1	0.0	0.005	0.999	mean accuracy: 0.995 minimum accuracy: 0.988 stdv accuracy: 0.005
price	2	0.0	0.220	0.988	
price	3	0.0	0.230	0.994	
price	4	0.0	0.049	0.999	
brand popularity	1	0.0179	0.4127	0.2347	mean accuracy: 0.731 minimum accuracy: 0.235 stdv accuracy: 0.291
brand popularity	2	0.0053	0.5834	0.9629	
brand popularity	3	0.0125	0.7718	0.8163	
brand popularity	4	0.0091	0.6062	0.9098	
sales rank	1	0.001	0.005	0.996	mean accuracy: 0.939 minimum accuracy: 0.759 stdv accuracy: 0.104
sales rank	2	0.002	0.273	0.759	
sales rank	3	0.0	0.005	1.000	
sales rank	4	0.0	0.002	1.000	

Table 7.15: Hard test metrics for the evaluation of the numerical stereotypes. The table reports for all numerical features and all stereotypes the dissimilarity in probability mass and centre, and the accuracy from equation 5.4. The last column gives standard metric of the accuracy. Amazon product group: 'Clothing, Shoes and Jewelry'

training and over the test data match. Of the 3 numerical features, 2 of them have a hard test resulting average accuracy of well above 90%. The other feature, brand popularity has an average stereotype accuracy of 87%. For all the features examined, the numerical stereotypes are very stable (both in terms of probability 'mass' and in terms of their numerical ranges and hence their centres), thus making them truly representative of the item population. This is the further evidence of the stability for the sample population statistical distributions for the numerical features.

For the product group Clothing, Shoes and Jewellery, of the 3 numerical features, 2 of them have a hard test resulting an average accuracy of well above 90%. The other feature, brand popularity, has an average stereotypes accuracy of 73%. This is mainly due to the misalignment of cluster 1, the lowest popularity items that has an accuracy of just 23%. Nonetheless, the numerical stereotypes can be viewed as being quite consistent across the two datasets (training and test) making them a very good and stable representation of the item population for both Amazon product groups.

7.3.3 A Soft Test

As discussed in Chapter 5, the soft test consists in evaluating how suitable the stereotypes created on the training data are for new products. In the unsupervised learning context, one does not have the 'true' stereotypical labels of the items. In order to obtain what we can regard as the 'true' stereotypical representation of the items in the test dataset, we first recreate the stereotypes over the Full dataset (F= training dataset + test dataset).

For each of the items in the test dataset for both product groups, the statistics of the samples' mismatch ratios of **complex categorical feature 'categories'** using the stereotypes F (full dataset) and T (training dataset) are analysed in Table 7.16. The mismatch ratio was introduced formally in Chapter 5 as a metric to quantify how similar/dissimilar the representation of an item is between two sets of stereotypes as coordinates: the ratio represents the sum of the non-matching labels divided by the total number of labels for the item. The soft test is highlighting a very high number of perfect alignment. Only less than 10% of the items representations in stereotypes for the soft test display a non zero mismatch. 9.7% for Sports & Outdoors, 6.7% for Clothing Shoes and Jewelery. The average mismatch is low, but that is also a result of the median being 0 (perfect match across most of the items).

As for the **numerical features**, the quality of the classification is investigated for each feature via the confusion matrix of the classification. For each stereotype, as well as for the entire features, the accuracy of the stereotypes can be defined as the number of true positives (TP) divided by the number of items. Another alternative metric to evaluate the

Descriptive Statistics of the mismatch ratios	Sports &Outdoor Value	Clothing, Shoes and Jewellery Value
Average mismatch	4.3%	2.9%
Median mismatch	0%	0%
Number of non-zero mismatch	15,505	37,689
Proportion of non-zero mismatches	9.7%	6.7%
Mean conditioned on mismatch begin > 0	73%	61%

Table 7.16: Statistics describing the mismatch ratios for the stereotypical representations of 'categories' feature for the items in test dataset using full vs the training stereotypes

Items= 85,941	True s1	True s2	True s3	True s4
Predicted s1	21066	0	0	0
Predicted s2	0	21701	0	0
Predicted s3	0	0	21365	0
Predicted s4	0	0	422	21387

Table 7.17: Confusion matrix for the feature 'price'. Amazon product group: Sports & Outdoors

classifier is via the F1 score, which is the harmonic average of precision and recall. The recall is defined as the number of true positives (TP) divided by the number of positives (TP + FP). The precision is defined as the number of true positives (TP) divided by the number of true positives and false negatives (TP + FN). The reader is referred to Chapter 5 where these concepts were first introduced.

Tables 7.17 to 7.19 display the confusion matrices for all the stereotypes of the numerical features in the product group Sports & Outdoors. While Tables 7.20 to 7.22 display the confusion matrices for all the stereotypes of the numerical features of the product group Clothing, Shoes and Jewellery.

For each feature in the product group Sports & Outdoors, the total number of items is fewer than the 159k items comprising the test dataset (total number of items is fewer than the 450k for product group Clothing, Shoes and Jewellery). Such difference in items arises because many items have invalid/missing entries for the numerical feature presented. In the tables, the incorrect predictions are highlighted in orange.

Table 7.23 displays the accuracy and F1-score metrics derived from the confusion matrices for the numerical features of both product groups. All the features have an accuracy above 90%.

Items= 45,080	True s1	True s2	True s3	True s4
Predicted s1	17173	2674	0	0
Predicted s2	0	9875	3616	0
Predicted s3	0	0	7653	1664
Predicted s4	0	0	0	2425

Table 7.18: Confusion matrix for the feature 'brand popularity'. Amazon Product Group: Sports & Outdoors

Items= 110,593	**True s1**	**True s2**	**True s3**	**True s4**
Predicted s1	27342	0	0	0
Predicted s2	174	27601	0	0
Predicted s3	0	0	27653	0
Predicted s4	0	0	0	27823

Table 7.19: Confusion matrix for the feature 'sales rank'. Amazon product group: Sports & Outdoors

Items= 172,507	**True s1**	**True s2**	**True s3**	**True s4**
Predicted s1	42715	120	0	0
Predicted s2	0	43098	103	0
Predicted s3	0	0	43176	18
Predicted s4	0	0	0	43277

Table 7.20: Confusion matrix for the feature 'price'. Amazon product group: Clothing, Shoes and Jewellery

Items= 28,855	**True s1**	**True s2**	**True s3**	**True s4**
Predicted s1	10297	1864	0	0
Predicted s2	0	6447	1875	0
Predicted s3	0	0	4365	579
Predicted s4	0	0	0	3428

Table 7.21: Confusion matrix for the feature 'brand popularity'. Amazon product group: Clothing, Shoes and Jewellery

Items= 197,342	**True s1**	**True s2**	**True s3**	**True s4**
Predicted s1	49480	0	0	0
Predicted s2	0	49586	0	0
Predicted s3	0	0	48960	0
Predicted s4	0	0	0	49316

Table 7.22: Confusion matrix for the feature 'sales rank'. Amazon product group: Clothing, Shoes and Jewellery

Product Group	**Feature Name**	**F1-score**	**Accuracy**
Sport & outdoors	price	0.995	99.8%
	brand popularity	0.794	91.2%
	sales rank	0.998	99.9%
Clothing, Shoes and Jewellery	price	0.999	99.9%
	brand popularity	0.849	92.5%
	sales rank	1.000	100%

Table 7.23: Numerical features stereotypes evaluation, soft test. F1-score and accuracy metrics for the classification problem of the test items using the stereotypes generated on training items

It has been demonstrated how the stereotypes discovered for the Amazon dataset via the automated procedures are indeed representative of the population of items, and over a large and heterogeneous set of items the stereotypes are remarkably stable. In particular, it was shown how the stereotypes generated independently on some unseen data are very similar, when not outright identical, to those obtained on the training dataset (hard test). And it was also shown how the stereotypes generated over the training data when used to make inferences/predictions about group belonging of new unseen items (test data) do produce very accurate predictions (soft test). The only stereotype and dataset where the resulting accuracy was a bit lower, but still very high, is the one for brand popularity; this could be attributed to the very skewed characteristic of this feature (i.e. few brands with very high popularity vs many with average to low popularity).

Having demonstrated the stability and ability to represent well the characteristics of the items, it is now necessary to demonstrate whether or not the characteristics of the stereotypes discovered are capable of representing groups of users' preferences. This question is answered in the following section following the steps of Chapter 5.

7.3.4 Predictive Power of Stereotypes

In this section the stereotypes discovered in the previous sections for the Amazon dataset are examined for their ability to represent user preferences. If the stereotypes were proved to be able to model user preferences for a non-trivial portion of the user population then this would indicate a suitable set to be used for recommendation.

Up to this point for both Amazon groups, only the item metadata has been used. In order to formulate the test that will examine which stereotypes and which features are most suited to be used as representative of a user preferences, it is necessary to introduce the rating data. For the Sports & Outdoor product group, the dataset contains the ratings provided by approximatively two million users for about 500k items in the training and test data, although only about 20k users have reviewed more than 10 products.

For the Clothing, Shoes and Jewellery product group, the dataset contains the ratings provided by over three million users for over 1.5 million items in the training and test data, although only about 30k users reviewed more than 10 products. Note that not all of the products in the dataset (either training or test dataset) have reviews, a large number of them are unreviewed.

As discussed and demonstrated in Chapter 5, the Agresti-Coull test can be used to build a methodology and a metric on how a stereotype is capable of influencing a single user's preference. This provides us with a way to verify whether the stereotypes created are capable of capturing and representing users preference traits in a population of users. The reader is directed to Section 5.2.3 to refresh the main ideas behind the Agresti-Coull test.

Feature	% users not different	% users with L.P.P	% users with L.N.P	Avg of significant stereotypes	Total stereo-types
Categories	32%	33%	0%	1.93	24
Price	43%	35%	15%	1.39	4
Brand Popularity	52%	33%	14%	1.36	4
Sales Rank	58%	29%	11%	1.35	4

Table 7.24: Summary for all features of the search for explanatory power of stereotypes via the Agresti-Coull test. confidence level of 95%. Amazon product group: Sports & Outdoors

Feature	% users not different	% users with L.P.P	% users with L.N.P	Avg of significant stereotypes	Total stereo-types
Categories	16%	11%	5%	1.94	26
Sales Rank	22%	56%	23%	1.54	4
Brand Popularity	29%	42%	19%	1.52	4
Price	52%	34%	14%	1.35	4

Table 7.25: Summary for all features of the search for explanatory power of stereotypes via the Agresti-Coull test. Confidence level of 95%. Amazon product group: Clothing, Shoes and Jewellery

Table 7.24 displays the results of the application of the Agresti-Coull method for all the stereotypes of all features in product group Sports & Oudoors, and it is evaluated with 95% significance. The column '% users not different' represents the percentage of the users - over the entire sample of users - that do not display any statistically significant preference or dis-preference for at least one of the stereotypes of that particular feature under examination.

In the table, features are ordered by significance of the stereotypes. For example, for the feature categories, 32% of the users in the sample do not present at least one statistically significant preference/dis-preference for a specific stereotype of the categories feature. On the contrary 68% of the users appear to have at least one significant preference or dis-preference in at least one of the labels of the categories driven stereotype.

The columns with '%users with L.P.P or L.N.P' (i.e. large positive preference or large negative preference) display how many users have what can be considered a strong like/dislike preference +/-15% for at least one stereotype of the feature examined. For example, for the feature categories we see that 68% of the users display a statistically significant positive or negative preference for at least one stereotype, however no users display a strong negative preference for one or more stereotypes and 33% of users display a strong positive preference. The last two columns of the table display the typical number of stereotypes triggering positive/negative preferences for users on average, as well as the number of stereotypes that a given feature has. For example, for categories there are 24 stereotypes and a given user is affected on average by less than 2 of those. This indicates that users have exhibited a specific preference for only a few labels /categories that are represented by a

small number of stereotypes.

The other three numerical features seem to be able to have an impact on users' preferences on average for 50% of the users. The main conclusion that can be drawn is that we are more than 95% confident that the stereotypes created for the complex categorical feature are capable of capturing and representing users positive and negative preferences for 2/3 of the user population, and the numerical feature stereotypes capture the preference traits of one in two users. While the stereotypes for 'categories' seem to have only positive large influences, the numerical feature stereotypes describe both positive and negative large preferences.

Table 7.25 displays the results of the application of the Agresti-Coull method, for all stereotypes of all features in product group Clothing, Shoes and Jewellery, and it is evaluated with 95% significance. It is important to note that in both Tables 7.24 and 7.25, the percentage of total indifferent users is calculated on a different sample size (i.e. the sample on which the stereotypes could be calculate). While for the complex categorical feature we could use reviews on all the products in the training dataset, for the numerical features we could use only the products for which the feature under study is available which represent a subset of the whole training dataset.

Similarly, across the two product groups the 'categories' driven stereotype is the one ranking highest in determining positive or negative statistically significant user preferences, but it is at the same time also the weakest in determining very large positive/negative preferences.

Ultimately, we can conclude that in both datasets the stereotypes generated are capable of describing user preference traits in substantially large portions of the user population. Interestingly the stereotypes of the dataset 'Clothing, Shoes and Jewellery' capture a larger share of the user population preferences. Perhaps it is worth highlighting a very different type of behavior for retail customers although further investigation is outside the scope of the RS development of this research. For instance the population of users purchasing an item in the 'Sports and Outdoors' group seem to be more affected by price, while price seems to be the least important characteristics driving a user decision in the 'Clothing, Shoes and Jewellery'. The stereotype system that will be built onto these two datasets is indeed capable of capturing and modelling such behavior in its recommendations.

7.4 Recommendation Performance

The prediction of item consumption during the research on the ML/IMDb dataset in Chapter 6 demonstrated that such a prediction, rarely done in the literature compared to the prediction of ratings, is strongly affected by the imbalance between the class predicted (i.e. consumed) and the majority of the observations (i.e. not consumed). In the movie dataset,

each user has rated an average of 115 movies (less than 5% of the items). The imbalance proved challenging when training the classifier as we discussed in Section 6.1.1.

In the case for highly imbalanced datasets a classifier will classify items into the predominant class as this would minimise the error. Special techniques were used in Section 6.1.1 to treat imbalanced datasets with good success.

However, in the case of Amazon datasets, the imbalance is so extreme that the prediction attempts may be worthless. In the Amazon datasets, we have approximately 530,000 items for the Sport and Outdoors group, and roughly 1,500,000 items for the Clothing, Shoes and Jewellery group, with the typical user having reviewed on average 1.7 items. Even if the dataset was cut down to retain only the items that have more than 10 reviews, the average user-to-item review percentage would be in the region of 0.01% which is 400 times more unbalanced than the ML/IMDb data.

For the above reason the investigation of the item consumption for the Amazon dataset was not carried out in this research. Later when the investigation of the model performance against SVD is carried out, information regarding the ability to correctly rank consumption will arise from measures like MRR and HLU.

In the remainder of this section, we will present the assessment of recommendations driven by stereotypes during cold starts in two stages:

1. Cold start user-to-item rating.
2. Cold start assessment of recommendations driven by stereotypes versus SVD-based RS (with metadata).

7.4.1 Cold Start Assessment of Item Rating

In Section 6.1.2, we explained how user-to-item ratings exhibit different kinds of global biases. For instance, some users consistently give higher ratings on items than other users, as well as some items on average receive higher feedback than other items. In order to compute accurate rating predictions such bias effects (user and item biases) need to be removed from the data.

The dependent variable in this experiment is the user rating, which is expressed as an integer value ranging from 1 to 5. We normalised rating as follow:

$$\tilde{r}_{ui} = r_{ui} + b_i + b_u \tag{7.1}$$

Where r_{ui} is the actual rating, b_i represents the item bias and b_u represents the user bias. The item bias is obtained by subtracting the item average from the mean ($b_i = \mu - \mu_i$ where μ is the overall mean across users and items, and μ_i is the mean of item i). The user

bias, b_u is obtained by subtracting the user average from the mean ($b_u = \mu - \mu_u$ where μ_u is the mean of user u).

Note that the distribution of ratings for Amazon dataset is highly unbalanced toward high rating. Almost 60% of the ratings are equal to 5, and another 20% are equal to 4, leaving only 20% for ratings between 1 and 3. This characteristic 'high rating concentration' is depicted in Figure 7.21, where the cumulative proportion of ratings is displayed. The high rating concentration is a characteristic that makes this dataset completely different from the previous dataset investigated and, as shown later in the research, it is going to have a specific impact on the rating predictions. This characteristic may be interpreted as one further piece of evidence showing how the Amazon and ML/IMDb datasets are different.

The difference between the ML/IMDb and the Amazon datasets may be driven by the fact that the movie rating attributed by a single user has little impact on the commercial fortune or misfortune of the movie. However in the Amazon case, sellers may be encouraged to produce or influence user ratings (i.e. unreal reviews), as such ratings directly reflect on their ability to generate further sales in a very tangible and immediate manner.

This 'high rating concentration' might be the reason why Amazon was recently pushed to introduce 'Verified Purchases Reviews[3]'. The data in the public dataset does not contain any information on whether a review was verified or not, and it is easy to assume it was not verified as most reviews are pre-2014 but the verified review flag was introduced in Nov-2016. This may be the reason why, to the author's knowledge, many scientific papers use the ML dataset more often than the Amazon dataset. However, it is important to note that this important difference highlighted between the two datasets constitutes further supporting evidence for the general validity of the stereotype-based methodology.

As a measure of quality of the results, the root mean squared error (RMSE) is reported along with the mean absolute error (MAE).

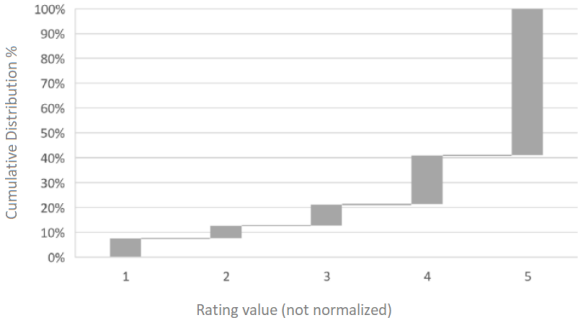


Figure 7.21: Composition of rating - Amazon dataset

[3]https://www.amazon.com/gp/help/customer/display.html?nodeId=202076110

Baseline and Stereotype-Based Models

Two recommendation models are considered: a 'baseline model' which uses all features available in the data in the way they are provided (this is the model against which we benchmark our results), and a 'stereotypes-based model' where the item features are treated via the rating independent stereotypes generated previously.

In order to measure the impact of replacing the original features with the stereotypes, and simulate cold start situations (new user and new item), the dataset has been split into two alternative experimental sets:

- Experiment A (new user) - For each item in the dataset the models are trained on the preference set expressed by a subset of users (randomly selected). The remaining users are left out, and used to test the accuracy of the models.
- Experiment B (new item) - For each user in the dataset the models are trained on all preferences expressed for a subset of items (randomly selected). The remaining items are left out, and used to evaluate the accuracy of the models.

For each of the two experiments several machine learning algorithms are employed from the simplest methods (naive method and linear regression) to the most popular and complex (neural network and XGBoost) with the aim to improve model performances and to confirm that our conclusions related to the application of stereotypes do not depend on the particular model chosen. The reader is referred to Section 6.1.2 for more details on the adopted regression techniques. All experiments are run on an Intel Core i7 -7700K CPU @ 4.2 GHz with 64.0 GB RAM.

Rating Predictions and Recommendation Results: New User And New Item Experiments

The results reported in Tables 7.26 and 7.27 are the average of a six experiments where the dataset was split 70% in training and 30% in test. The test data was effectively split in six successive experiments with some overlap of the test data allowed across experiments. The predicted ratings in the Tables 7.26 and 7.27 are subject to the clamping and flooring described in Section 6.1.2.

The simple prediction rules, also called naive estimators, which predicts new ratings to be the mean rating of items for the new user case (i.e. average rating among existing users) or as the mean rating by user across all items for the new item case, result in RMSE= 0.76 and RMSE= 0.80, respectively. Notice that the first rule represents a sensible 'best sellers list' approach.

The linear regression method constitutes a substantial improvement compared to the naive methods, resulting in an improvement of at least 19% and 26% in RMSE over the baseline model in the new user and new item problem respectively. Using stereotypes results in a further improvement over the baseline model in excess of 1.9% and 2.57% for new user and new item problem respectively. Once more, the use of stereotypes appears to bring an increase in performances. A similar pattern applies to MAE.

As discussed for ML/IMDb dataset, the neural network regression approach is employed as one of the more sophisticated and powerful predictions approaches. Using a neural network regressor further improves the rating prediction performance. The RMSE improves by about 0.5% and 0.2% compared to the linear regression baseline value for new user and new item problems respectively. This small improvement in performance comes at the cost of around 26% increase in computational time. Applying stereotypes leads to a further 2.6% and 3.5% improvement in the RMSE with respect to the neural network baseline model for new user and new item problems respectively. The MAE follows a similar pattern. The performance increase is obtained with a reduction of roughly 30% of the computational time.

In a similar fashion as seen for linear regression, neural network regression also confirms that: a) stereotypes improve performance by a few percentage points compared to the same model that uses basic features, and b) stereotypes reduce computational time.

Using a XGBoost regressor lead to the highest prediction improvement compared to the naive estimation. While the performance for the baseline XGBoost model is only slightly better than the baseline linear regression model, adding stereotypes lead to an improvement of 3.1% with respect to the XGBoost baseline model for the RMSE in the new user experiment. The MAE follows a similar pattern.

For the new item experiment, the XGBoost algorithm is the one that leads to the highest prediction accuracy improvement, with an increased performance of well over 5% for both RMSE and MAE compared to the XGBoost baseline model.

One more datum regarding the benefit of using stereotypes, reported only for the XGBoost new user and new item tests, consists in the number of times the predicted rating value for the baseline model is closer to or further away from the real observed rating compared to the predicted rating value for the stereotype-based model. For the new user experiment it was found that 56.8% of the times the prediction using stereotypes is closer to the actual rating than the prediction of the baseline model. For the new item experiment the stereotype-driven predictions were closer to the actual rating 59.7% of the times.

It is important to note that for this particular dataset improving the regression model does not bring much improvement in prediction performance. However, more sophisticated models (neural network and XGBoost) do substantially improve performance metrics com-

New User Experiment		
RMSE (Naive: 0.76)	**Base features**	**Stereotyped features**
Linear R.	0.616	0.604
Neural Net R.	0.614	0.598
XGBoost R.	0.612	0.593
MAE (Naive: 0.48)		
Linear R.	0.530	0.523
Neural Net R.	0.523	0.515
XGBoost R.	0.522	0.512
Cpu Time (Naive: 1)		
Linear R.	1.534	1.147
Neural Net R.	27.89	18.055
XGBoost R.	12.77	5.472
Efficiency Metric		
Linear R.	1.565	1.710
Neural Net R.	1.252	1.296
XGBoost R.	1.279	1.375

Table 7.26: Performance metrics for new user problem - Amazon dataset

New Item Experiment		
RMSE (Naive: 0.80)	**Base features**	**Stereotyped features**
Linear R.	0.585	0.570
Neural Net R.	0.584	0.563
XGBoost R.	0.585	0.550
MAE (Naive: 0.53)		
Linear R.	0.515	0.492
Neural Net R.	0.511	0.485
XGBoost R.	0.510	0.474
Cpu Time (Naive: 1)		
Linear R.	1.193	1.094
Neural Net R.	26.239	18.383
XGBoost R.	10.779	5.786
Efficiency Metric		
Linear R.	1.836	1.928
Neural Net R.	1.386	1.447
XGBoost R.	1.413	1.553

Table 7.27: Performance metrics for new item problem - Amazon dataset

pared to simpler models (linear regression). Regardless of this, the improvement and value added by stereotypes is fully demonstrated. In addition, the extra predictive content arising when using stereotypes seems to be orthogonal (i.e. independent) of the machine learning approach used (i.e. the solver used to train the features-rating relationship). Similar finding is also observed in the ML/IMDb dataset. When stereotypes are used in more complex models the percentage improvement in the error metrics increases over their respective baseline models.

We can speculate the ability of more complex models, in their baseline form, to perform substantially better than simpler ones (i.e. neural network performing only marginally better than linear regression) can be related to the very characteristic discussed previously, namely the extreme concentration of high reviews in the dataset. Investigation of this particular aspect is outside the scope of this research, and it does not affect our conclusion on the value added by stereotypes.

The two side benefits of using stereotypes (error reduction and computational speed improvement) can be evaluated by using the ad-hoc Efficiency metric 6.3 which was discussed for the same experiments in the movie case in Chapter 6. The metric aims at weighting the effects of an increase in prediction accuracy, as measured by the relative improvement of RMSE compared to the naive RMSE together with the improvement in computational efficiency. The parameter k of the formula is set at 0.03, aiming at under weighting the improvements in computational efficiency, because of a lower importance compared to improvements in RMSE. The metric confirms what has already been discussed, namely that for the Amazon dataset there is little benefit in using complicated prediction models. When the little RMSE improvement is balanced against the higher computational requirements, stereotypes are instead a valuable way to increase efficiency for all models.

In both of the example analysed we have demonstrated how stereotypes enable to improve the accuracy and at the same time reduce the computational time by a substantial amount. The improvement in accuracy cannot be ascribed to a better fit, as it is independent of the regressor model chosen. We can summarise our findings as follows:

- There is a consistent improvement in prediction accuracy between 2% and 5% of RMSE when using stereotypes compared to using the original baseline features during cold start experiments.
- The improvement is consistent across the regressors. Linear, neural network and XGBoost regressions all perform better using stereotypes than using the original features.
- The improvement in prediction accuracy due to stereotypes is larger than the improvement that can be harvested by choosing incrementally more complex regression techniques.
- Using stereotypes, especially in the most powerful regression techniques (NN and XGBoost) leads to more than 30% computational time improvement over using the base features.
- The improved accuracy (via a smaller RMSE) and improved CPU time can be incorporated into the efficiency metric and it shows the real advantage in favour of the stereotype approach.

– Looking at the percentage improvements obtained in the efficiency metric, one can see that the improvements in using stereotypes are higher than the improvements in increasing the model complexity from a simple linear regression to XGBoost. Therefore, providing more grounded evidence for the use of stereotypes in cold start phases. In particular for the Amazon dataset, the Efficiency metric 6.3 suggests that one may settle for simpler models like linear regression with stereotypes as the most efficient approach.

To illustrate another consequence of the fact that the dataset contains a vast majority of top positive reviews, Figure 7.22 displays the histogram for the distribution of the predicted rating error. For example, an error of -1 means that the predicted rating was one step higher than the actual rating given by the new user (to the new item). We can see that 93% of the predictions have an error between +/- 1 (a small error) - i.e. the predicted rating was off only by one level up or down, demonstrating how the model suggested is indeed capable of representing the preference (positive or negative) in cold start phases over 9 times out of 10.

Note that the distribution of the error is skewed toward the negative side. This happens as a result of the top rating concentration. The models calibrated have very little incentive in predicting lower ratings (as the training ratings are extremely dense in the 4 and 5 area). The stiffness of the fit derives the small quantity of large errors. These are low reviews that were not predicted to be so low.

The experiments in this section verified the research question:
Can automatically constructed item-based stereotypes improve recommendations during the cold start phases?

In the next section, the matrix-factorisation methods (SVD and SVD with metadata also known as Factorisation Machine) that were explored and compared with the proposed stereotype approach for the ML/IMDb data are applied to the Amazon data and the results compared to the ones obtained using stereotypes.

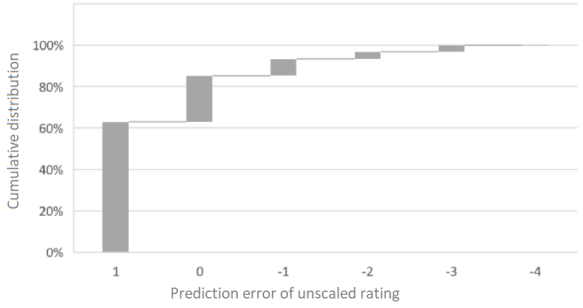


Figure 7.22: Distribution of rating error, new item case, XGBoost

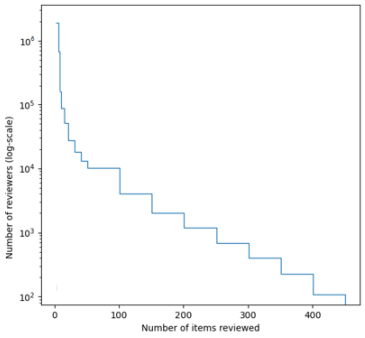


Figure 7.23: Distribution of number of users by number of items reviewed for the Sports & Outdoors product group

7.4.2 Cold Start Assessment of Recommendations Driven by Stereotypes versus SVD-Based RS (with metadata)

The general theory and model set up for the SVD and SVD with metadata features were discussed in section 6.1.3. There is one general complication with the Amazon dataset that required a change in the underlying implementation. The Amazon dataset is much larger and much sparser than the ML/IMDb dataset. Figure 7.23 shows the histogram (in log scale) representing how many reviews typical users have. There are millions of users, i.e. the vast majority, with fewer than 3 reviews. Only at two orders of magnitude below, about ten thousand users have 30 reviews or more, and only less than one thousand users that have more than 100 reviews. This is a very different picture than that of the ML/IMDb case.

To handle the sparsity, it was decided to test two alternative sparse matrix approaches: list of list (LIL) and compressed row storage (CRS) or Yale approach. For a description of both approaches see [187]. Ultimately while LIL provides more flexibility, it is less efficient when handling matrix operations. The CRS method was adopted. Despite the fewer features of the Amazon dataset, compared to the ML/IMDb case, the substantially larger number of users and much larger number of items would render the use of matrix-factorisations with metadata very cumbersome if not outright unfeasible on machines with standard specifications without the use of a sparse matrix treatment.

SVD with metadata vs Stereotypes Recommendations: New User and New Item Experiments

In this section we will compare the results obtained from the factorisation methods (SVD

and SVD with metadata) with the best results obtained from machine learning techniques (XGBoost) and the use of stereotypes.

Table 7.28 compares key accuracy metrics obtained for both the new user and new item experiments. The results reported are the average of six experiments in which the dataset was split 70% in training and 30% in test. The test data was effectively split in six successive experiments with some overlap of the test data allowed across experiments. It is important to note that for the 30% of users (items in new item case) in the test set we assumed that no ratings were available. That means that no previous information was assumed to be available for those users (items), which were used to assess the new user (new item) behaviour. Table 7.28 demonstrates the model with stereotypes outperforms the SVD method in both RMSE and MAE in the cold start phase. The improvement arising by using stereotypes seems to be independent and captured by the stereotypes only, confirming the findings observed in the ML/IMDb dataset. In comparison to the new user tests performed for the ML/IMDb dataset, the performance improvement offered by the stereotype-based model compared to the SVD with metadata is even larger.

In the new item experiment, using implicit feedback, i.e. SVD++, in the form of data on other items rated does not constitute a real benefit as it leads to a negligible improvement of just 0.1% in the RMSE compared to the SVD (not shown) at the cost of more than doubling the computational time. This is probably attributable to the extreme sparsity of the data, where the implicit feedback (i.e. disliking all items not consumed/reviewed) is a poor assumption given the vastity of the items catalogue and the extremely low number of reviews per user.

New User Experiment				
	Base Model	**Stereotype**	**SVD without metadata**	**SVD with metadata**
RMSE	0.612	0.593	0.733	0.613
MAE	0.522	0.512	0.542	0.534
New Item Experiment				
RMSE	0.585	0.550	0.773	0.588
MAE	0.510	0.474	0.529	0.511

Table 7.28: New user and new item cold start comparisons between the recommendation models: stereotypes and SVD with and without metadata

In order to evaluate the rank accuracy metrics, the test sets had to be modified as follows:

- For the new user case we divided users 70%-30% and trained the model on the known rating for 70% of the users, as we did for the previous analyses. For the remaining 30% of the users the test set included all the items in the dataset whether the user rated them or not. In this way our top-N recommendation list includes also items that have not been rated by the user.

– For the new item case we divided items 70%-30% and trained the model on all users but just on 70% of the items. The test set included for each user all of the items in the test set (30% of the total) regardless of whether that user had rated the item or not. In this way our top-N recommendation list includes also items that have not been actually rated by the user.

In addition, due to the sparsity of the dataset analysed, we had to restrict our analysis only to users with a minimum number of reviews in order to obtain meaningful rank accuracy metrics. For the Sports and Outdoors and the Clothing Shoes and Jewellery subsets of the Amazon dataset, each reviewer has rated on average just 1.7 items with the majority of the user having reviewed only one item. Figure 7.23 shows the histogram for the number of items rated by reviewers for the Sports and Outdoors portion of the data where the fact just quoted is self-evident in log scale. In order to evaluate meaningful rank accuracy metrics, only reviewers with more than 10 reviews were retained for this part of the study. It would be difficult to assemble any statistically significant metric if the test set had not been restricted to such reviewers.

Because of the extreme sparsity of the data, and because in the test set while we retain only users with more than 10 reviews we also retain and potentially recommend all items (also the ones that on the dataset have no reviews), we expect the ranking metrics to be lower than those achieved for the ML/IMDb dataset. A different point of view could be obtained by only providing recommendations for the items that have also received a minimum number of reviews. While this would improve the rank accuracy metrics, we wish to point out that what we are interested in this experiment as a relative comparison among techniques, and also showing and preserving the most extreme cold start approach.

Hit Rate. The simplest metric to evaluate the accuracy of the top-N recommendations is the hit rate. The hit rate measure was discussed in details in Section 6.1.3. We evaluated HR at different N (10, 20 and 30) for the two cold start cases using the two methods under comparison i.e. model with stereotypes and SVD with metadata. The main results are shown in Table 7.29.

	Model with stereotype		**SVD with features**		***p*-value of difference in HR**	
Hit Rate @ N	**New User**	**New Item**	**New User**	**New Item**	**New User**	**New Item**
HR @ 10	3.16%	2.53%	1.50%	1.40%	$\ll 0.01$	$\ll 0.01$
HR @ 20	3.13%	2.54%	1.47%	1.45%	$\ll 0.01$	< 0.01
HR @ 30	2.96%	2.66%	1.44%	1.61%	$\ll 0.01$	< 0.01

Table 7.29: Hit rate for Top-N recommendation list - Amazon dataset

As shown in Table 7.29 for both the new user and the new item cases the model with

stereotypes has a higher percentage of hits with respect to the SVD with metadata. The table also shows the p-values of the test with null hypothesis: 'The difference in the mean of hit rate is not statistically significant'. Given the extremely low p-value obtained we are confident at 99% that the improvements in HR given by the stereotype-based models are significant for both the new user and new item case. In the table the symbol ($\ll$) is used to indicate that the p-value is significantly less than the value shown as to be on a smaller order of magnitude.

An important observation for the new user case is the fact that as N increases the HR decreases, this is due to the number of hits identified being those ranking high, and not identifying further hits down in the list, also probably due to the larger catalogue of unreviewed items that the RS is drawing its recommendations from. It is worth noting that despite the slightly higher RMSE of the new item case, with respect to the new user case, the latter displays a higher HR. This is explained as follows: while in the new user case each single user is scored against all of the items (whether items were rated or not), because of the train test split in the new item case only 30% of the items are retained as new item. Therefore, it is less likely that, when a new item is recommended to all users, that item was actually reviewed by any of those users. For this very same reason, in all of the rank accuracy metrics here analysed the new item case is likely to score lower than the new user case.

Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP). MRR is another measure to evaluate systems that return a ranked list of answers to queries while MAP provides a more suitable tool in the cases where we are interested in the ranking quality of a list rather than just the highest-ranking item. It can be viewed as an extension of both MRR and HR and it was discussed in Section 6.1.3.

Table 7.30 reports both MRR and MAP for the two cold start cases. While HR measured the ability to provide more and more relevant recommendations when doing top-N, and we concluded that stereotypes-based models contributed to a better system, the MRR measures the quality of only the first 'hit' score, and in a similar way the first hit score deriving by the use of stereotypes is almost twice that of the SVD with features for the new user case and more than 50% better for the new item case. The same observation conducted when discussing HR about the underlying reason of the new user case being much better than the new item case also applies for this metric. Given the sparsity of the data, i.e. most users rank very little number of items, we feel that it is sufficient to evaluate the MRR for the top 10 list.

MAP can be viewed as an extension of both MRR and HR. For both the new user and the new item cases, MAP is higher with statistical confidence well above 95% that the model

with stereotypes does better than the SVD with features in the new user case. For the new item case the performance improvement of the stereotypes have a stronger confidence. In a similar way as we found analysing the ML/IMDb dataset, not only stereotypes lead to a lower accuracy error but also to a higher user satisfaction as measured by HR and MAP.

	Model with stereotype		**SVD with features**		***p*-value of difference**	
	New User	**New Item**	**New User**	**New Item**	**New User**	**New Item**
MRR @ 10	4.97%	3.66%	2.8%	2.28%	$\ll$0.01	$\ll$0.01
MAP @ 10	2.89%	2.52%	2.25%	2.08%	0.03	$<$0.01
MAP @ 20	2.99%	2.71%	2.55%	2.13%	0.04	$<$0.01
MAP @ 30	3.22%	2.77%	2.66%	2.66%	0.02	$<$ 0.05

Table 7.30: Mean reciprocal rank (MRR) and mean average precision (MAP) - Amazon dataset

Normalised Discounted Cumulative Gain (nDCG). nDCG is a single-number measure of the effectiveness of a ranking algorithm that allows non-binary judgments of relevance. nDCG has a discounting function as discussed in Section 6.1.3. The results obtained for the two cold start scenarios using the stereotypes-based models and the matrix-factorisation with metadata based models are reported in Table 7.31.

	Model with stereotype		**SVD with features**		***p*-value of difference**	
nDCG @ N	**New User**	**New Item**	**New User**	**New Item**	**New User**	**New Item**
nDCG @ 10	5.8%	3.31%	3.4%	2.9%	$<$0.01	0.08
nDCG @ 20	7.5%	4.47%	4.2%	4.1%	$<$ 0.01	0.07
nDCG @ 30	8.7%	6.0%	4.8%	5.6%	$<$0.01	0.08

Table 7.31: Comparison nDCG for model with stereotypes and SVD with metadata - Amazon dataset

nDCG confirms the results obtained from the other ranking metrics analysed by measuring how useful are our recommendation on average to users. The model with stereotypes outperforms the SVD with metadata according to the nDCG values; for new user the improvement is with a very high statistical confidence (above 99%); for new item the improvement is lower and, given the variability embedded in the problem, we are around 92% confident that such improvement is statistically significant.

Half-life Utility Metric (HLU). While the ranking metrics we have examined so far were all adapted from different fields for use in the recommender systems domain, the half-life utility metric was designed explicitly for recommender systems by Breese et al. [36]. The metric was discussed in detail in Section 6.1.3. The HLU postulates that the

probability that a user will select a relevant item drops exponentially down the list of recommendation, and this reflects the smaller and smaller likelihood that the users will browse down deeply in the list. Table 7.32 shows the comparison between the half-life-utility value at a decay factor equal to 3 using the model with stereotypes versus the SVD with metadata. For both tests we can assert that the model with stereotypes outperforms the SVD model with metadata with more than 95% of confidence for the new user and more than 99% confidence for the new item. This confirms what we found with the other measures presented in the above sections where using the model with stereotypes appears always outperforming the SVD model.

	Model with stereotype		**SVD with features**		***p*-value of difference in HLU**	
HLU @ N	**New User**	**New Item**	**New User**	**New Item**	**New User**	**New Item**
HLU @ 10	2.52	2.33	1.54	1.14	<0.05	<0.01
HLU @ 20	2.41	2.30	1.49	1.13	<0.05	<0.01
HLU @ 30	2.29	2.29	1.44	1.12	<0.05	<0.01

Table 7.32: Comparison half-life utility for model with stereotypes and SVD with metadata using a decay factor α equal to 3 - Amazon dataset

Serendipity. As discussed in Section 6.1.3, various definitions have been proposed for the concept of serendipity in recommender systems. We discussed how a serendipitous recommendation should possess the property of being relevant as well as unexpected to the user. Given that in the literature there is no accord in a universal definition of serendipity, when we analysed the ML/IMDb dataset we introduced a metric of serendipity based on the variety of labels being reccommened within the top-N recommendation for a complex categorical feature, and in the case of the ML/IMDb database we examined genre. The same definition can also be used for the Amazon dataset and its complex categorical feature: product category. The underlying idea for Amazon is that if a recommender system obtains high prediction accuracy, but it does so by always recommending the same product category (e.g. baseball caps) this would not be a serendipitous system, despite the high accuracy that one may achieve.

Using the top-N recommendation lists produced for the new user case from both SVD with metadata and stereotypes, we can measure how many distinct product categories are recommended to a user. However, when we look at the top-N items, we are constrained in proposing a predefined number of items for each user as, for this dataset, we do not have any user metadata to differentiate between users, and this greatly limits the serendipity of our predictions. Nevertheless, a comparison is possible on how much variety in product category, derived from the stereotypes and the SVD with metadata based systems.

One complication that was also discussed for ML/IMDb consists in the fact that when examining how a multi-value feature like product category is 'variegate' we need an operative way to address items that are belonging to many product categories simultaneously. The same operative definition as that introduced previously for the ML/IMDb can be employed, in fact such an operative way of computing the variety in a complex categorical feature can be employed for any complex categorical feature. To generalise the definition of k introduced previously, when considering the top-N recommendation for a complex categorical feature, the number of times of a label appearing in one or more recommended items in the top-N is counted and we called this the score of the label. We can then claim that a label is fully represented in the recommendation list if that label has a score larger than a threshold value. The smaller the threshold k the more labels are represented. By comparing, for a given threshold k, the number of labels covered by the top-N recommendations by the two alternative RS, it is possible to indirectly conclude which one has more variety in its recommendations, and hence be more serendipitous.

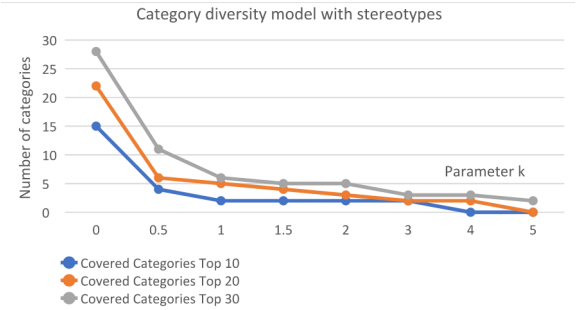


Figure 7.24: Product category diversity (number of distinct product categories recommended in the top-N list) for the model with stereotypes

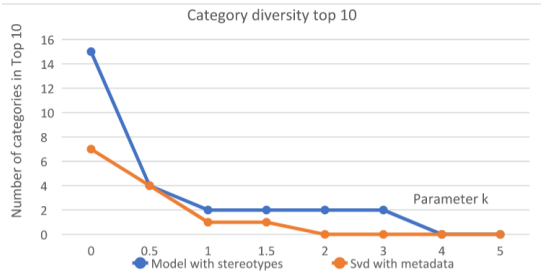


Figure 7.25: Comparison product category diversity for the model with stereotypes and the SVD with metadata

In Figure 7.24 we show the number of product categories represented in the top-N recommendation list (y-axis) as a function of the growing value of the threshold k (x-axis) for the stereotype-based models. In Figure 7.24 we show the results obtained for the stereotype

model for the new user case for 3 different N values, namely Top 10, 20 and 30. To note that the feature product category has 72 distinct labels in total, so in this case, only less than half of those categories are represented asymptotically when the top-N increases. The parameter k ranges between 0 (i.e. no limit, all the product categories associated with the recommended items are counted as being represented even if their score is really small) and 5 (i.e. only product categories that have a weighted score above 5 are counted). As the list of recommendation increases in size, moving from top 10 to 20 and 30, the variety increases, as expected from a serendipitous RS.

As done for the ML/IMDb data, we also compare the variety (proxy for serendipity) metric between the stereotype-based system and the SVD with metadata model. Figure 7.25 shows the comparison in the product categories diversity for the top 10 recommendation lists produced by the model with stereotypes and the SVD with metadata. Across the full spectrum of thresholds the category diversity of the stereotype-based RS is larger or equal to the category diversity of the SVD-based RS with metadata. This observation allows us to conclude that a stereotype-based RS is more serendipitous than (or at least equally serendipitous as) SVD-based RS.

This section benchmarked the use of stereotypes against the recent advancement in recommender systems when predicting cold start phase. Hence, it verified the research questions:

- *How do stereotype-based item models compare to other state-of-the-art recommender systems models in terms of predictive performance during the cold-start phases?*
- *Apart from recommendation accuracy, what are the other benefits of item-based stereotypes?*

7.5 Summary

In this chapter, we have constructed stereotypes automatically on the Amazon dataset and shown a broad range of evaluations of the benefits of introducing stereotypes over the original metadata during the cold start phase. Our empirical experience proves the underlying ideas on how to generate rating and preference-independent item-based stereotypes automatically. We followed a statistically driven approach to the evaluation of the stereotypes. The results confirmed a remarkable stability and consistency, thus paving the road to using stereotypes in the context of recommendation.

We have benchmarked our methodology against the state-of-the-art techniques. SVD representation provides an ideal framework for dimensionality reduction and, for completeness we presented an investigation and comparison between such techniques and the stereotype-driven approach. An in-depth analysis was carried out of the recommendation quality of the two approaches (stereoptypes and factorisation methods); such analysis is

not limited to recommendation accuracy, but also other aspects and desirable properties of recommendations, such as user utility and novelty. We have shown how stereotypes-based recommendation improves the rating prediction accuracy metrics (MAE and RMSE) as well as rank metrics compared to SVD with metadata. We also highlighted the extra benefits of the proposed approach in improving computational efficiency and CPU run time.

The conclusions drawn after the ML/IMDb investigations have all been validated by the experiments of the second dataset Amazon. Thus, we further validated answers to the research questions:

- *Can item-based stereotypes, not based on rating, be constructed automatically?*
- *Can automatically constructed item-based stereotypes improve recommendations during the cold-start phases?*
- *How do stereotype-based item models compare to other state-of-the-art recommender systems models in terms of predictive performance during the cold-start phases?*
- *Apart from recommendation accuracy, what are the other benefits of item-based stereotypes?*

The results on the second dataset Amazon provide validation that the methodology to automatically construct stereotypes we proposed can be generalised to other datasets, retaining a similar performance improvement.

CHAPTER 8

Conclusion and Future Work

8.1 Conclusion

When a set of users and a set of items are well known in terms of their metadata and other implied or expressed preferences, that may span from the past ratings to item selections and browsing histories, the recommender system community has developed very accurate techniques that can provide well targeted recommendations. A problem that have not been as widely studied is that of cold start, which is defined as the problem of providing meaningful recommendations when a new user or a new item joins the platform. The cold start remains an important area of improvement for modern RS, especially given the fact that a new user or a new item content provider are likely to form a view of the usefulness of the RS during the first use sessions, making the first impact of the system potentially even more important than the recommendation accuracy tailored to well-known users and items.

In the literature there is not a unique agreement on the definition, but cold start has been further split into extreme cold start and non-extreme cold start. The first is defined as the situation in which the new user has consumed fewer than a handful of items (or a new item has been consumed by fewer than a handful of users). As a new user and a new item become known via more and more rating or consumption actions the RS gradually shifts away from cold start and is able to apply a more personalised recommendation behaviour. Several works discussed in the literature reviewed in Chapter 2 have demonstrated that during cold start and especially during extreme cold start, before any meaningful person-

alisation can be applied by the RS, it is extremely difficult for the most sophisticated RS techniques (for example those using deep learning algorithms) to outperform naive and baseline methodologies.

It is now widely acknowledged that user and item metadata can play a key role during cold start, when nothing else is known about a new user or a new item, other than the metadata accompanying it. The RS should use such metadata to provide a generalised level of recommendation personalisation. Most of the recent research has focused on adapting existing recommendation algorithms or defining new recommendation algorithms to cope with cold start by resorting to user and item metadata [69, 88]. The main research efforts still seem to be tailored mostly to improving the underlying recommendation technique, embedding the metadata of users and items in the recommendation process, and fitting patterns of recommendation based on the metadata from past ratings.

This thesis have researched and quantified the possibility of improving modern RS performance during cold start using a different point of view, that of rating agnostic stereotypes. The underlying hypothesis driving this research stems from the simple observation that, when nothing else is known other than user and item metadata, it should be possible to discover (using known users and items as a training set) whether there exists some stereotypical relationships between particular sets of users and items that may explain some general preference patterns. If that was the case it would lead to the ability of providing a first level of personalisation when only metadata is available for the new user and the new item. While this may seem not too dissimilar from what other researched recommendation approaches have done there is a key underlying difference. In the literature the training set of ratings are used to drive user and item metadata clustering, namely the associations of metadata that will be used to provide cold start recommendations are discovered directly via the ratings. In our proposed approach we take a step back and we suggest that we should first discover associations between users and items driven by metadata similarities/dissimilarities independently of past ratings. These associations are what we have defined as stereotypes in the course of this work. Only once the stereotypes have been formed one should then train a cold start dedicated RS to find relationships between stereotypes and preferences. In fact, the cold start dedicated RS is trained on the stereotyped users and items rather than on the raw data of users and items. While this may superficially appear to be only a small variation, this thesis discusses how such an approach seems to have been neglected in the literature and we have demonstrated that it does have a statistically significative positive effect on the resulting cold start recommendation performance as well as system efficiency.

The aim of this thesis, as stated in Chapter 1 was to fulfil the following:
How stereotypes can be built *automatically*, and most importantly in a way that is *inde-*

pendent of user preferences and to demonstrate how stereotypes can effectively improve recommendations during the cold start phase.

The research contributions will be outlined below, to show how the the research questions presented in Chapter 1 are addressed.

1. Can item-based stereotypes, not based on rating, be constructed automatically?
2. Can automatically constructed item-based stereotypes improve recommendations during the cold-start phases?
3. How do stereotype-based item models compare to other state-of-the-art recommender systems models in terms of predictive performance during the cold-start phases?
4. Apart from recommendation accuracy, what are the other benefits of item-based stereotypes?

Unlike previous work where stereotypes were built using expert knowledge or based on user-to-item rating matrix, a methodology has been proposed to demonstrate the benefit of building stereotypes automatically and independently of the user-to-item rating matrix. Thus answering our first research question in Chapter 4. Our method consists of creating clusters among items and users that provide the ability to fit general patterns that may not be discoverable through the use of ratings. Chapter 5 discussed a statistically-driven approach to the evaluation of the stereotypes and can be viewed as an element of novelty arising from the present research.

The application of the stereotypes allow for better recommendation for new user and new item and it is given in Chapters 6 and 7. While the same chapters provided an assessment of the proposed systems against recommendations driven by standard factorisation methods as well as factorisation methods with the embedded item and user metadata. Thus answering second and third research questions.

Traditionally, RS research has focused on forecasting the rating that users would give to each item. Rating prediction continues to be an important performance evaluation aspect of RS and has been adopted by recent research [124, 127]. Nevertheless, researchers have acknowledged that accuracy of rating predictions alone is not sufficient for identifying a quality RS, especially from a user-satisfaction and user-retention perspective. The ongoing trend is to evaluate ranked lists of items, presenting users with ranked lists of items and evaluating which RS-derived lists possess qualities such as being relevant and novel to the user. Our presentation of results includes a wide range of metrics, from precise ratings predictions and metrics describing the quality of ranked lists to metrics attempting to capture the serendipity of recommendations. The final research question is answered in Chapters 6 and 7 where in-depth analysis of the recommendation quality is presented.

8.2 Contribution to knowledge

The output of the academic research has a direct impact on the advancements in industry. Recommending technologies are gaining growing industry interest. Amazon has been using collaborative filtering for a decade to suggest products to their customers, and Netflix appreciated improvements to the recommender technology behind their movie service.

Rapid growth of e-commerce business in recent years has generated an urgent need for recommendation techniques. Recommendation algorithms occupy an important role to recommend products for new user, or recommend newly added product to the catalogue. Stereotyping in RS is known for its efficiency in solving cold start problem. However, to the best of our knowledge, efforts in created them are limited to manual and/or user-based approach. This thesis is proposing an automated approach for creating item-based stereotypes for solving user and item cold start problems. The main contribution of the present work is highlighted below.

Constructing stereotypes automatically and independently from user-to-item rating

The first aspect of novelty of our research in terms of importance, is the proposed algorithm for the automated creation of stereotypes. Previous attempts at using clustering or other unsupervised learning algorithms for creating stereotypes were all based on the user-to-item rating matrix, or were done manually via expert knowledge of the problem at hand. We have discussed and demonstrated how certain relationships arising between features might have not been readily discoverable with expert knowledge of the problem. To accomplish the automatic creation of stereotypes we have discussed a structured approach that separates the features of the metadata in three large groups, namely numerical, categorical and complex categorical. During Chapters 3 and 4 we have discussed why certain algorithms have been preferred to others (for example the use of hierarchical clustering over k-mode type of algorithms), and how the algorithm chosen have led to the formulation of stereotypes.

Stereotypes significantly reduces the dimension of the recommendation model

One of the novelty aspects of this thesis, rests on the result that stereotypes, when generated using logical relationships between items built in a way that is independent of past user-to-item preferences, provide a reduced coordinate system that can aid the machine learning model underlying the RS to discover patterns that would have not otherwise been discovered if the raw users and items representations were employed. We have demonstrated how the resulting cold start recommendation quality when employing stereotypes is improved compared to both naive, baseline models and alternative models that make

use of the primitive metadata features. In particular we have demonstrated how the improvement in recommendation quality reach beyond the simple improved accuracy, and it instead embraces several aspects that are deemed to determine positive recommendation characteristics.

In the body of the thesis we have discussed the effects of using stereotypes versus standard metadata when predicting cold start item consumption, when predicting cold start ratings, and also on predicting the quality of ranked lists produced for cold start new user and new item experiments. In all these cases stereotypes add a statistically significant improvement, therefore suggesting that stereotype lead to better cold start recommendation performance, with 'better' referring to a wide range of metrics assessing the quality of recommendations.

Stereotypes improvements in cold start recommendations is independent of the recommendation technique used

The third finding, and probably one of the most important finding and contribution of this work, was encountered and discussed during the cold start recommendation experiments. We routinely observed that the improvement in recommendation quality deriving from using stereotypes was, at a first order of approximation, independent of the machine learning algorithm employed to fit the effect of the ratings over the stereotyped user and item coordinates. This observation, albeit with some noise, was drawn in a similar manner for both of the datasets investigated and it leads us to suggest that the performance gains via the stereotypes in cold start recommendations are independent of the machine learning technique used to fit the user-to-item rating matrix. We deem this finding as one of the most important findings of this work as it provides RS researchers with an extra dimension for improvement. As discussed earlier in this thesis and in the present conclusions, most efforts in the literature seem to have been dedicated to improve the machine learning algorithm that connects users with ratings and make use of metadata. Such attempts could be revisited with further aid from the stereotype-based coordinates.

Applicability of the approach on different features

Over the course of the thesis we demonstrated the flexibility of the approach by producing stereotypes and stereotypes driven recommendations on two datasets that are extremely different in their nature, that of movies (MovieLens coupled with IMDb extra metadata) and that of retail purchases (Amazon). Both datasets have different features types like numerical, categorical and complex categorical. Our suggested approach can be easily and readily generalised to other datasets.

In a speculative way, one may also hypothesize that such stereotypical recommendation framework could be created also for features that are not readily represented as one of our three features group, but that could be transformed to such a state with the aid of an extra

dedicated algorithm. For example, if a feature included images (for example a database of vacation hotspots or of houses for sale) we could imagine using an image recognition software that extracts 'keywords' from an image. Such keywords would be naturally used for stereotyping the feature via a complex categorical feature approach. In the example of houses for sale keywords collected via the image recognition software could be 'wooden floor', 'carpet', 'swimming pool', 'modern', 'wallpaper', 'fire place' and 'garden'. Such keywords could also be parsed from the human readable text description of items. This observation suggests that the scope and potential uses of the proposed algorithms is broader than that investigated in this thesis.

A formal test framework to evaluate stereotypes before using them in a RS

The following aspect of novelty introduced by this thesis rests on our formulation of a series of statistical tests that are aimed at evaluating whether or not the stereotypes discovered are stable (when regenerated over new unseen data) and whether they are capable of representing known preferences from an existing population of user-to-item ratings. In the literature, when clustering of the rating matrix is performed and projected onto the metadata as a way to improve cold start recommendations, no much is said about stability of the clusters discovered. Somewhat this approach is justified by the use of unsupervised learning algorithms for which we have no way to really know if the associations learnt are 'correct' or 'wrong' apart from the fact of whether they are helpful in producing good or poor recommendations. In our context we argue that stereotypes must represent general traits of users or items, and hence, if we assume that the dataset is homogeneous, we should be able to discover extremely similar stereotypes if the procedure was carried out on a separate partition of the data. If that was not the case, we argue that any relationship across stereotypes learnt on the data would not really represent a real underlying phenomenon in the dataset. For our tests dataset we have demonstrated how stereotypes were indeed extremely stable. A situation where stereotypes would not be stable can be imagined as one in which user tastes change over time. For example, if a long time history of reviews of music were used one would be able to see that relationships discovered across stereotypes representing general taste might change over time; like users in their teens would have preferred pop music in the 90s and perhaps 'rap' in the 2000s. Albeit we have discussed the homogeneity of the datasets under investigation, we have not observed this characteristic with our experimental datasets, we argue that we would still be able to cope with such a situation. The system could be readapted for such case by using a time windowed approach to data; stereotypes would be generated for each rolling time window (for example for the last decade) and then patterns fit on such window.

The second part of the proposed stereotype testing phase tries to quantify whether each stereotype is able to represent user preferences, this was done by examining how each

user's preferences, both excessive positive and negative preferences when compared to that expected in the population, are actually captured by one or more stereotypes. We demonstrated that stereotypes do indeed capture the in-sample user preferences, with all stereotypes contributing to the preference representation. Some users may be more sensitive on average to the stereotypes of a feature (for example driven by movie genre, the western and war movies are positively preferred by men in their 50s) and other users to the stereotypes of a different feature (teenagers have preference for high budget movies).

A metric to evaluate serendipity for complex categorical features

Given the nature of stereotypes construction, we have introduced a proxy for measuring how variegate and diverse a ranked list of items is; we have argued that such a metric can be used directly as a representation of recommendation novelty and serendipity of the ranked list presented to a new user. We have also confirmed that according to such serendipity/novelty proxy metric the stereotype-driven recommendations prove to be of a higher quality than the recommendations produced using the original metadata features. The formulation of the proxy metric constitutes also a minor aspect of novelty, and it was dictated by the lack of an agreed and quantifiable measure that represents novelty/serendipity in the literature, which appears to have several authors introducing ad-hoc qualitative definitions depending on the field of application.

8.3 Limitations and Future Work

As the number of ratings expressed by a new user, or given to a new item grows the stereotype-based system in this current formulation is not designed to embed them. In fact, in the formulation obtained the stereotypes are static and any RS driven by them can be viewed as a specialised machinery to address the cold starts. It is however easy to envisage possible ways to remedy this. The simplest way would be to mediate and mix stereotype-driven recommendation with the personalised recommendations that a collaborative filtering driven RS would give as ratings by the new user or for the new item become available. More sophisticated ways could also be obtained, by introducing an intelligent way that re-evaluate the stereotype classes of the new user and new item based on the new acquired ratings recommendation. These points could be viewed as future areas of research.

We have identified also other areas of future research stemming from the present work; for instance, one more problem known to affect RS is that of overspecialisation in the recommendations once a user or item becomes well known. In the thesis we have observed that the ranked recommendation lists provided by stereotypes have a higher variety in terms of the type of metadata features that they embrace. This finding was discussed when we established that such higher variety is an indication of higher serendipity/novelty compared

to a non stereotype-based system. Such property of the recommendations could be used to reintroduce novelty and variety when overspecialisation is detected.

Other minor tasks identified in the course of the research as future work would be to experiment the use of stereotypes in more sophisticated RS algorithm, and experiment with the embedding of implicit feedback when the user-to-item ratings are introduced and used to train the stereotype-driven RS. Ultimately more research should be dedicated to user-driven stereotypes. Both datasets used in this research were extremely rich in item metadata but less so with the regard to the user metadata. A research conducted in publicly available datasets dedicated to the RS tasks revealed that most datasets are poor or totally lacking in user metadata. This may be the result of publishing data trying to avoid any possible privacy concern, thereby applying data anonymisation. Privately available datasets would be the ideal candidate for the next steps of this research. Finally, as indicated in Chapter 2, RS can be evaluated using either online methods or offline methods. The recommender system evaluation conducted in the current research are based on offline methods. It will be interesting to test serendipity and novelty of the proposed approach through online user tests.

Overall, the work presented highlights a novel dimension for improvement in cold start problems that is applicable to other RS techniques.

Availability of code and data. In case of future enquiries, the code and datasets (ML and Amazon) are uploaded into a cloud folder: MovieLens and Amazon.

References

[1] S. Spiegel, J. Kunegis, and F. Li, "Hydra: a hybrid recommender system [cross-linked rating and content information]," in *Proceedings of the 1st ACM international workshop on Complex networks meet information & knowledge management*. ACM, 2009, pp. 75–80.

[2] E. Rich, "User modeling via stereotypes," *Cognitive science*, vol. 3, no. 4, pp. 329–354, 1979.

[3] C. C. Aggarwal *et al.*, *Recommender systems*. Springer, 2016.

[4] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender systems handbook*. Springer, 2011, pp. 1–35.

[5] K. Hosanagar, D. Fleder, D. Lee, and A. Buja, "Will the global village fracture into tribes? recommender systems and their effects on consumer fragmentation," *Management Science*, vol. 60, no. 4, pp. 805–823, 2013.

[6] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, no. 1, pp. 76–80, 2003.

[7] Z. Lock, "Performance and flexibility of stereotype-based user models," Ph.D. dissertation, University of York, 2005.

[8] J. Kay, Z. Halin, T. Ottomann, and Z. Razak, "Learner know thyself: Student models to give learner control and responsibility," in *Proceedings of International Conference on Computers in Education*, 1997, pp. 17–24.

[9] J. Kay, "The UM toolkit for cooperative user modelling," *User Modeling and User-Adapted Interaction*, vol. 4, no. 3, pp. 149–196, 1994.

[10] G. Brajnik and C. Tasso, "A shell for developing non-monotonic user modeling

systems," *International Journal of Human-Computer Studies*, vol. 40, no. 1, pp. 31–62, 1994.

[11] B. Lamche, E. Pollok, W. Wörndl, and G. Groh, "Evaluating the effectiveness of stereotype user models for recommendations on mobile devices." in *UMAP Workshops*. Citeseer, 2014.

[12] J. Beel, S. Dinesh, P. Mayr, Z. Carevic, and J. Raghvendra, "Stereotype and most-popular recommendations in the digital library sowiport," in *Proceedings of the 15th International Symposium of Information Science (ISI)*, 2017.

[13] Q. Chen, A. F. Norcio, and J. Wang, "Neural network based stereotyping for user profiles," *Neural Computing & Applications*, vol. 9, no. 4, pp. 259–265, 2000.

[14] T. George and S. Merugu, "A scalable collaborative filtering framework based on co-clustering," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*. IEEE, 2005, pp. 4–pp.

[15] J. Das, P. Mukherjee, S. Majumder, and P. Gupta, "Clustering-based recommender system using principles of voting theory," in *2014 International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2014, pp. 230–235.

[16] J. Avila, X. Riofrlo, K. Palacio-Baus, D. Astudillo, V. Saquicela, and M. Espinoza-Mejla, "Decategorizing demographically stereotyped users in a semantic recommender system," in *2016 XLII Latin American Computing Conference (CLEI)*. IEEE, 2016, pp. 1–7.

[17] F. Ricci, L. Rokach, B. Shapira, and P. Kan-tor, "Recommender systems handbook: A complete guide for research scientists and practitioners," 2011.

[18] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, J. Riedl *et al.*, "Combining collaborative filtering with personal agents for better recommendations," *AAAI/IAAI*, vol. 439, 1999.

[19] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proceedings of the 1st ACM conference on Electronic commerce*. ACM, 1999, pp. 158–166.

[20] T. Mahmood and F. Ricci, "Learning and adaptivity in interactive recommender systems," in *Proceedings of the ninth international conference on Elec-tronic commerce*. ACM, 2007, pp. 75–84.

[21] F. Ricci and Q. N. Nguyen, "Acquiring and revising preferences in a critique-based mobile recommender system," *IEEE Intelligent systems*, vol. 22, no. 3, 2007.

[22] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[23] D. M. Fleder and K. Hosanagar, "Recommender systems and their impact on sales diversity," in *Proceedings of the 8th ACM conference on Electronic commerce*. ACM, 2007, pp. 192–199.

[24] D. Kotkov, J. Veijalainen, and S. Wang, "Challenges of serendipity in recommender systems," in *Proceedings of the 12th International conference on web information systems and technologies*. SCITEPRESS, 2016, pp. 251–256.

[25] W. Wu, L. Chen, and Y. Zhao, "Personalizing recommendation diversity based on user personality," *User Modeling and User-Adapted Interaction*, vol. 28, no. 3, pp. 237–276, 2018.

[26] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer support-ed cooperative work*. ACM, 1994, pp. 175–186.

[27] D. Jannach, P. Resnick, A. Tuzhilin, and M. Zanker, "Recommender systems beyond matrix completion," *Communications of the ACM*, vol. 59, no. 11, pp. 94–102, 2016.

[28] J. Bennett, S. Lanning *et al.*, "The netflix prize," in *Proceedings of KDD cup and workshop*, vol. 2007. New York, NY, USA., 2007, p. 35.

[29] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.

[30] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl *et al.*, "Item-based collaborative filtering recommendation algorithms." in *Proceedings of the 10th international conference on World Wide Web*. Hong Kong, Hong Kong, 2001, pp. 285–295.

[31] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–71, 1992.

[32] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Berkeley, California, USA, 1999, pp. 230–237.

[33] J. L. Herlocker, J. A. Konstan, and J. Riedl, "Explaining collaborative filtering recommendations," in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. Philadelphia, Pennsylvania, USA, 2000, pp. 241–250.

[34] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Information retrieval*, vol. 5, no. 4, pp. 287–310, 2002.

[35] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[36] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann, Burlington, 1998, pp.

43–52.

[37] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *information retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[38] D. Billsus and M. J. Pazzani, "Learning collaborative information filters." in *Proceedings of the fifteenth international conference on machine learning*, vol. 98. San Francisco, CA, United States, 1998, pp. 46–54.

[39] C. Basu, H. Hirsh, W. Cohen *et al.*, "Recommendation as classification: Using social and content-based information in recommendation," in *Aaai/iaai*, 1998, pp. 714–720.

[40] C. C. Aggarwal, Z. Sun, and S. Y. Philip, "Online generation of profile association rules." in *KDD*, 1998, pp. 129–133.

[41] R. Bhaumik, C. Williams, B. Mobasher, and R. Burke, "Securing collaborative filtering against malicious attacks through anomaly detection," in *Proceedings of the 4th Workshop on Intelligent Techniques for Web Personalization (ITWP'06), Boston*, vol. 6, 2006, p. 10.

[42] C. C. Aggarwal and S. Y. Philip, "A general survey of privacy-preserving data mining models and algorithms," in *Privacy-preserving data mining*. Springer, 2008, pp. 11–52.

[43] E. Frolov and I. Oseledets, "Hybridsvd: when collaborative information is not enough," in *Proceedings of the 13th ACM Conference on Recommender Systems*. Copenhagen, Denmark, 2019, pp. 331–339.

[44] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Incremental singular value decomposition algorithms for highly scalable recommender systems," in *Fifth international conference on computer and information science*, vol. 27. Citeseer, 2002, pp. 27–28.

[45] S. S. Anand and N. Griffiths, "A market-based approach to address the new item problem," in *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 2011, pp. 205–212.

[46] M. Sollenborn and P. Funk, "Category-based filtering and user stereotype cases to reduce the latency problem in recommender systems," in *European Conference on Case-Based Reasoning*. Springer, 2002, pp. 395–405.

[47] P. Lops, D. Jannach, C. Musto, T. Bogers, and M. Koolen, "Trends in content-based recommendation," *User Modeling and User-Adapted Interaction*, vol. 29, no. 2, pp. 239–249, 2019.

[48] I. Nunes and D. Jannach, "A systematic review and taxonomy of explanations in decision support and recommender systems," *User Modeling and User-Adapted Interaction*, vol. 27, no. 3-5, pp. 393–444, 2017.

[49] E. Peis, J. M. del Castillo, and J. A. Delgado-López, "Semantic recommender systems. analysis of the state of the topic," *Hipertext. net*, vol. 6, no. 2008, pp. 1–5, 2008.

[50] M. Pozo, R. Chiky, and E. Métais, "Enhancing collaborative filtering using implicit relations in data," in *Transactions on Computational Collective Intelligence XXII*. Springer, 2016, pp. 125–146.

[51] M. I. Martín-Vicente, A. Gil-Solla, M. Ramos-Cabrer, J. J. Pazos-Arias, Y. Blanco-Fernández, and M. López-Nores, "A semantic approach to improve neighborhood formation in collaborative recommender systems," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7776–7788, 2014.

[52] F. S. Gohari and M. J. Tarokh, "New recommender framework: combining semantic similarity fusion and bicluster collaborative filtering," *Computational Intelligence*, vol. 32, no. 4, pp. 561–586, 2016.

[53] Q. Shambour and J. Lu, "Government-to-business personalized e-services using semantic-enhanced recommender system," in *International Conference on Electronic Government and the Information Systems Perspective*. Springer, 2011, pp. 197–211.

[54] G. Adomavicius, N. Manouselis, and Y. Kwon, "Multi-criteria recommender systems," in *Recommender systems handbook*. Springer, 2011, pp. 769–803.

[55] G. Adomavicius and Y. Kwon, "New recommendation techniques for multicriteria rating systems," *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 48–55, 2007.

[56] Q. Shambour, M. Hourani, and S. Fraihat, "An item-based multi-criteria collaborative filtering algorithm for personalized recommender systems," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 8, pp. 275–279, 2016.

[57] Z. Sun, G. Guo, and J. Zhang, "Exploiting implicit item relationships for recommender sys-tems," in *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 2015, pp. 252–264.

[58] M. H. Rimaz, M. Elahi, F. Bakhshandegan Moghadam, C. Trattner, R. Hosseini, and M. Tkalčič, "Exploring the power of visual features for the recommendation of movies," in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*. Larnaca, Cyprus, 2019, pp. 303–308.

[59] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender systems handbook*. Springer, 2011, pp. 73–105.

[60] J. Wasilewski and N. Hurley, "Bayesian personalized ranking for novelty enhancement," in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*. Larnaca, Cyprus, 2019, pp. 144–148.

[61] M. Kaminskas and D. Bridge, "Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 7, no. 1, pp. 1–42, 2016.

[62] M. Eirinaki, M. Vazirgiannis, and I. Varlamis, "Sewep: using site semantics and a taxonomy to enhance the web personalization process," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 99–108.

[63] B. Magnini and C. Strapparava, "Experiments in word domain disambiguation for parallel texts," in *Proceedings of the ACL-2000 workshop on Word senses and multilinguality-Volume 8*. Association for Computational Linguistics, 2000, pp. 27–33.

[64] M. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites," *Machine learning*, vol. 27, no. 3, pp. 313–331, 1997.

[65] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial intelligence review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[66] B. Krulwich, "Lifestyle finder: Intelligent user profiling using large-scale demographic data," *AI magazine*, vol. 18, no. 2, pp. 37–37, 1997.

[67] D. Asanov *et al.*, "Algorithms and methods in recommender systems," *Berlin Institute of Technology, Berlin, Germany*, 2011.

[68] O. Barkan, N. Koenigstein, E. Yogev, and O. Katz, "Cb2cf: a neural multiview content-to-collaborative filtering model for completely cold item recommendations," in *Proceedings of the 13th ACM Conference on Recommender Systems*. Copenhagen, Denmark, 2019, pp. 228–236.

[69] I. Fernández-Tobías, I. Cantador, P. Tomeo, V. W. Anelli, and T. Di Noia, "Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization," *User Modeling and User-Adapted Interaction*, vol. 29, no. 2, pp. 443–486, 2019.

[70] R. Burke, "Hybrid web recommender systems," in *The adaptive web*. Springer, 2007, pp. 377–408.

[71] G. Shani, A. Meisles, Y. Gleyzer, L. Rokach, and D. Ben-Shimon, "A stereotypes-based hybrid recommender system for media items," in *Workshop on Intelligent Techniques for Web Personalization, Vancouver, Canada*, 2007, pp. 76–83.

[72] S. Middleton, D. De Roure, and N. Shadbolt, "Ontology-based recommender systems. handbook on ontologies. international handbooks on information systems," 2009.

[73] N. N. Qomariyah and D. Kazakov, "Mixed type multi-attribute pairwise comparisons learning," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 1094–1097.

[74] N. N. Qomariyah, "Pairwise preferences learning for recommender systems," Ph.D. dissertation, University of York, 2018.

[75] S. Jiang, X. Wang, C. Yuan, and W. Li, "Mining user preferences for recommendation: A competition perspective," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Springer, 2013, pp. 179–189.

[76] Y. Fang and L. Si, "A latent pairwise preference learning approach for recommendation from implicit feedback," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 2567–2570.

[77] D. Jannach, M. Zanker, M. Ge, and M. Gröning, "Recommender systems in computer science and information systems–a landscape of research," in *International Conference on Electronic Commerce and Web Technologies*. Springer, 2012, pp. 76–87.

[78] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. Tampere, Finland, 2002, pp. 253–260.

[79] M. Elahi, M. Braunhofer, T. Gurbanov, and F. Ricci, "Collaborative recommendations: Algorithms, practical challenges and applications," B. Shlomo, C. Ivan, and T. Domonkos, Eds. Singapore: World Scientific Publishing, 2018, ch. User Preference Elicitation, Rating Sparsity and Cold Start, pp. 253–294.

[80] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge & Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[81] L. Cella, S. Cereda, M. Quadrana, and P. Cremonesi, "Deriving item features relevance from past user interactions," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. Bratislava, Slovakia, 2017, pp. 275–279.

[82] D. Cohen, M. Aharon, Y. Koren, O. Somekh, and R. Nissim, "Expediting exploration by attribute-to-feature mapping for cold-start recommendations," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*. Como, Italy, 2017, pp. 184–192.

[83] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: introduction and challenges," in *Recommender systems handbook*. Springer, 2015, pp. 1–34.

[84] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," in *Proceedings of the 12th ACM Conference on Recommender Systems*. New York, NY, USA, 2018, pp. 311–319.

[85] C. Z. Felício, K. V. Paixão, C. A. Barcelos, and P. Preux, "A multi-armed bandit

model selection for cold-start user recommendation," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. Bratislava, Slovakia, 2017, pp. 32–40.

[86] D. Kluver and J. A. Konstan, "Evaluating recommender behavior for new users," in *Proceedings of the 8th ACM Conference on Recommender Systems*. Foster City, Silicon Valley California, USA, 2014, pp. 121–128.

[87] N. Mirbakhsh and C. X. Ling, "Improving top-n recommendation for cold-start users via cross-domain information," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 9, no. 4, pp. 1–19, 2015.

[88] Y. Deldjoo, M. F. Dacrema, M. G. Constantin, H. Eghbal-Zadeh, S. Cereda, M. Schedl, B. Ionescu, and P. Cremonesi, "Movie genome: alleviating new item cold start in movie recommendation," *User Modeling and User-Adapted Interaction*, vol. 29, no. 2, pp. 291–343, 2019.

[89] S. Sedhain, S. Sanner, D. Braziunas, L. Xie, and J. Christensen, "Social collaborative filtering for cold-start recommendations," in *Proceedings of the 8th ACM Conference on Recommender systems*. Foster City, Silicon Valley, California, USA, 2014, pp. 345–348.

[90] D. H. Alahmadi and X.-J. Zeng, "Twitter-based recommender system to address cold-start: A genetic algorithm based trust modelling and probabilistic sentiment analysis," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. Vietri sul Mare, Italy, 2015, pp. 1045–1052.

[91] X. Du, H. Liu, and L. Jing, "Additive co-clustering with social influence for recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*. Como, Italy, 2017, pp. 193–200.

[92] M. Enrich, M. Braunhofer, and F. Ricci, "Cold-start management with cross-domain collaborative filtering and tags," in *International Conference on Electronic Commerce and Web Technologies*. Prague, Czech Republic, 2013, pp. 101–112.

[93] I. Fernández-Tobías, M. Braunhofer, M. Elahi, F. Ricci, and I. Cantador, "Alleviating the new user problem in collaborative filtering by exploiting personality information," *User Modeling and User-Adapted Interaction*, vol. 26, no. 2-3, pp. 221–255, 2016.

[94] M. Nasery, M. Braunhofer, and F. Ricci, "Recommendations with optimal combination of feature-based and item-based preferences," in *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. Halifax, Canada, 2016, pp. 269–273.

[95] S. Kalloori and F. Ricci, "Improving cold start recommendation by mapping feature-based preferences to item comparisons," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. Bratislava, Slovakia, 2017, pp.

289–293.

[96] M. Elahi, F. Ricci, and N. Rubens, "Active learning in collaborative filtering recommender systems," in *International Conference on Electronic Commerce and Web Technologies*. Munich, Germany, 2014, pp. 113–124.

[97] M. Braunhofer, M. Elahi, and F. Ricci, "User personality and the new user problem in a context-aware point of interest recommender system," in *Information and Communication Technologies in Tourism 2015*. Lugano, Switzerland, 2015, pp. 537–549.

[98] Y. Koren, "Factorization meets the neighborhood: a multifaceted collabora-tive filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 426–434.

[99] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," in *Proceedings of ACM WebKDD Workshop*. ACM, 2000.

[100] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[101] Y. Koren, "The BellKor solution to the netflix grand prize," https://www.netflixprize.com/assets/GrandPrize2009_BPC_Bellkor.pdf, 2009, accessed: 2019-11-05.

[102] J. Kay, *Lies, damned lies and stereotypes: pragmatic approximations of users*. Basser Department of Computer Science, University of Sydney, 1994.

[103] G. Paliouras, V. Karkaletsis, C. Papatheodorou, and C. D. Spyropoulos, "Exploiting learning techniques for the acquisition of user stereotypes and communities," in *UM99 User Modeling*. Springer, 1999, pp. 169–178.

[104] F. Eskandanian, B. Mobasher, and R. Burke, "A clustering approach for personalizing diversity in collaborative recommender systems," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. Bratislava, Slovakia, 2017, pp. 280–284.

[105] M. Khalaji, K. Mansouri, and S. J. Mirabedini, "Improving recommender systems in e-commerce using similar goods," *Journal of Software Engineering and Applications*, vol. 5, no. 02, pp. 96–101, 2012.

[106] L. H. Ungar and D. P. Foster, "Clustering methods for collaborative filtering," in *AAAI workshop on recommendation systems*. AAAI, 1998, pp. 114–129.

[107] M. OâConnor and J. Herlocker, "Clustering items for collaborative filtering," in *Proceedings of the ACM SIGIR workshop on recommender systems*, vol. 128. Berkeley, 1999.

[108] I. Koprinska, J. Poon, J. Clark, and J. Chan, "Learning to classify e-mail," *Information Sciences*, vol. 177, no. 10, pp. 2167–2187, 2007.

[109] E. RICH, "Building and exploiting user models [ph.d. thesis]," 1979.

[110] E. Rich, "Users are individuals: individualizing user models," *International journal of man-machine studies*, vol. 18, no. 3, pp. 199–214, 1983.

[111] L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Difino, and B. Negro, "User modeling and recommendation techniques for personalized electronic program guides," in *Personalized Digital Television*. Springer, 2004, pp. 3–26.

[112] R. Davis, "Interactive transfer of expertise: Acquisition of new inference rules," *Artificial intelligence*, vol. 12, no. 2, pp. 121–157, 1979.

[113] S. M. Beitzel, O. Frieder, E. C. Jensen, D. Grossman, A. Chowdhury, and N. Goharian, "Disproving the fusion hypothesis: an analysis of data fusion via effective information retrieval strategies," in *Proceedings of the 2003 ACM symposium on Applied computing*. ACM, 2003, pp. 823–827.

[114] J. Orwant, "Heterogeneous learning in the doppelgänger user modeling system," *User Modeling and User-Adapted Interaction*, vol. 4, no. 2, pp. 107–130, 1994.

[115] G. Brajnik, G. Guida, and C. Tasso, "User modeling in expert man-machine interfaces: A case study in intelligent information retrieval," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 1, pp. 166–185, 1990.

[116] B. Shapira, P. Shoval, and U. Hanani, "Stereotypes in information filtering systems," *Information Processing & Management*, vol. 33, no. 3, pp. 273–287, 1997.

[117] M. Kamitsios, K. Chrysafiadi, M. Virvou, and E. Sakkopoulos, "A stereotype user model for an educational game: Overcome the difficulties in game playing and focus on the educational goal," in *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, 2018, pp. 1–6.

[118] N. ALRossais and D. Kudenko, "Evaluating stereotype and non-stereotype recommender systems." in *Proceedings of the First Workshop on Knowledge-aware and Conversational Recommender Systems co-located with the 12th ACM Conference on Recommender Systems KaRS@ RecSys*. Vancouver, Canada, 2018, pp. 23–28.

[119] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications*. Siam, 2007, vol. 20.

[120] S. Dolnicar, "Using cluster analysis for market segmentation-typical misconceptions, established methodological weaknesses and some recommendations for improvement," 2003.

[121] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[122] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on pattern analysis and machine intelli-gence*, vol. 22, no. 1, pp. 4–37, 2000.

[123] C. Z. Felício, K. V. Paixao, C. A. Barcelos, and P. Preux, "Preference-like score to

cope with cold-start user in recommender systems," in *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. San Jose, CA, USA, 2016, pp. 62–69.

[124] N. Mauro and L. Ardissono, "Extending a tag-based collaborative recommender with co-occurring information interests," in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*. ACM, 2019, pp. 181–190.

[125] N. Mirbakhsh and C. X. Ling, "Leveraging clustering to improve collaborative filtering," *Information Systems Frontiers*, vol. 20, no. 1, pp. 111–124, 2018.

[126] D. Sacharidis, "Group recommendations by learning rating behavior," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. Bratislava, Slovakia, 2017, pp. 174–182.

[127] A. T. Wibowo, A. Siddharthan, J. Masthoff, and C. Lin, "Incorporating constraints into matrix factorization for clothes package recommendation," in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*. Singapore, Singapore, 2018, pp. 111–119.

[128] B. Li, Y. Liao, and Z. Qin, "Precomputed clustering for movie recommendation system in real time," *Journal of Applied Mathematics*, vol. 2014, 2014.

[129] C.-Y. Lee and E. Antonsson, "Dynamic partitional clustering using evolution strategies," in *Industrial Electronics Society, 2000. IECON 2000. 26th An-nual Con-fjerence of the IEEE*, vol. 4. IEEE, 2000, pp. 2716–2721.

[130] A. K. Jain and R. C. Dubes, "Algorithms for clustering data," Prentice-Hall, Inc., 1988.

[131] M. R. Anderberg, "Cluster analysis for applications," Office of the Assistant for Study Support Kirtland AFB N MEX, Tech. Rep., 1973.

[132] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 321–352.

[133] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data mining and knowledge discovery*, vol. 2, no. 3, pp. 283–304, 1998.

[134] R. S. Sangam and H. Om, "The k-modes algorithm with entropy based similarity coefficient," *Procedia Computer Science*, vol. 50, pp. 93–98, 2015.

[135] F. Cao, J. Liang, D. Li, L. Bai, and C. Dang, "A dissimilarity measure for the k-modes clustering algorithm," *Knowledge-Based Systems*, vol. 26, pp. 120–127, 2012.

[136] J. Ji, T. Bai, C. Zhou, C. Ma, and Z. Wang, "An improved k-prototypes clustering algorithm for mixed numeric and categorical data," *Neurocomputing*, vol. 120, pp. 590–596, 2013.

[137] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[138] X. Liu, Q. Yang, and L. He, "A novel dbscan with entropy and probability for mixed data," *Cluster Computing*, vol. 20, no. 2, pp. 1313–1323, 2017.

[139] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," *Pattern recognition*, vol. 30, no. 7, pp. 1109–1119, 1997.

[140] Y. Leung, J.-S. Zhang, and Z.-B. Xu, "Clustering by scale-space filtering," *IEEE Transactions on pattern analysis and machine intelli-gence*, vol. 22, no. 12, pp. 1396–1410, 2000.

[141] G. J. Hamerly and C. P. Elkan, "Learning structure and concepts in data through data cluster-ing," Ph.D. dissertation, University of California, San Diego, 2003.

[142] M. G. Omran, A. P. Engelbrecht, and A. Salman, "An overview of clustering methods," *Intelligent Data Analysis*, vol. 11, no. 6, pp. 583–605, 2007.

[143] P. Sneath and R. Sokal, "Numerical taxonomy. freeman, london," 1973.

[144] A. Gunawardana and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *Journal of Machine Learning Research*, vol. 10, no. Dec, pp. 2935–2962, 2009.

[145] S. Shinde and M. Potey, "Survey on evaluation of recommender systems," *International Journal Of Engineering And Computer Science*, vol. 4, no. 2, pp. 10 351–10 355, 2015.

[146] . Cleverdon and E. Keen, "Factors determining the performance of indering systems, vol. 2, test results, aslib-cranfield research project, cranfield, 1966. salton, g, and me lesk, computer evaluation of indexing and text processing," *Journal of the ACM*, vol. 15, pp. 68–8, 1968.

[147] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, pp. 1–19, 2016.

[148] N. ALRossais and D. Kudenko, "iSynchronizer: A tool for extracting, integration and analysis of movielens and imdb datasets," in *UMAP'18 Adjunct: 26th Conference on User Modeling, Adaptation and Personalization Adjunct, July 8-11, 2018, Singapore, Singapore*, 2018, p. 5.

[149] J. Jotheeswaran and Y. Kumaraswamy, "Opinion mining using decision tree based feature selection through manhattan hierarchical cluster measure." *Journal of Theoretical & Applied Information Technology*, vol. 58, no. 1, 2013.

[150] V. Schickel-Zuber and B. Faltings, "Using an ontological a-priori score to infer user" s preferences," in *Proceeding of the Workshop on Recommender Systems-ECAI06*, no. LIA-CONF-2006-006, 2006, pp. 102–106.

[151] F. Eskandanian, N. Sonboli, and B. Mobasher, "Power of the few: Analyzing the impact of influential users in collaborative recommender systems," in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*. Larnaca, Cyprus, 2019, pp. 225–233.

[152] A. Rana and D. Bridge, "Explanations that are intrinsic to recommendations," in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*. Singapore, Singapore, 2018, pp. 187–195.

[153] L. Jing, P. Wang, and L. Yang, "Sparse probabilistic matrix factorization by laplace distribution for collaborative filtering," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[154] E. Palumbo, G. Rizzo, and R. Troncy, "Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation," in *Proceedings of the eleventh ACM conference on recommender systems*, 2017, pp. 32–36.

[155] B. Abdollahi and O. Nasraoui, "Using explainability for constrained matrix factorization," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 79–83.

[156] S. Zhang, L. Yao, and X. Xu, "AutoSVD++ an efficient hybrid collaborative filtering model via contractive auto-encoders," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. Shinjuku, Tokyo, Japan, 2017, pp. 957–960.

[157] Z. He, S. Deng, and X. Xu, "Approximation algorithms for k-modes clustering," in *International Conference on Intelligent Computing*. Springer, 2006, pp. 296–302.

[158] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[159] M. Friendly, "Corrgrams: Exploratory displays for correlation matrices," *The American Statistician*, vol. 56, no. 4, pp. 316–324, 2002.

[160] A. Zimek, "Correlation clustering," Ph.D. dissertation, University Munchen, Munchen, 2008.

[161] J. Podani, *Introduction to the exploration of multivariate biological da-ta*. Backhuys Publishers, Kerkwerve, 2000.

[162] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer, Heidelberg, 2001, vol. 1, no. 10.

[163] B. Everitt, "Cluster analysis heinemann educ," *London, Books*, 1974.

[164] Y.-C. Chen, "A tutorial on kernel density estimation and recent advances," *Biostatistics & Epidemiology*, vol. 1, no. 1, pp. 161–187, 2017.

[165] A. Ambardar *et al.*, *Analog and digital signal processing*, 1995.

[166] H. Edelsbrunner and J. Harer, *Computational topology: an introduction*. American Mathematical Society, Providence, 2010.

[167] A. Ginzburg, *Calculus: problems and solutions*. Courier Corporation, 2003.

[168] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.

[169] F. Cao, J. Liang, and L. Bai, "A new initialization method for categorical data clustering," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10 223–10 228, 2009.

[170] A. Agresti and B. A. Coull, "Approximate is better than exact for interval estimation of binomial proportions," *The American Statistician*, vol. 52, no. 2, pp. 119–126, 1998.

[171] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, p. e0118432, 2015.

[172] B. Abma, "Evaluation of requirements management tools with support for traceability-based change impact analysis," *Master's thesis, University of Twente, Enschede*, 2009.

[173] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study in-volving information extraction," in *Proceedings of workshop on learning from imbalanced da-tasets*, vol. 126, 2003.

[174] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 875–886.

[175] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[176] G. King and L. Zeng, "Logistic regression in rare events data," *Political analysis*, vol. 9, no. 2, pp. 137–163, 2001.

[177] M. H. Latif and H. Afzal, "Prediction of movies popularity using machine learning techniques," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 16, no. 8, pp. 127–131, 2016.

[178] R. M. Bell and Y. Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights." in *Proceedings of the 2007 seventh IEEE international conference on data mining*, vol. 7. Omaha, NE, USA, 2007, pp. 43–52.

[179] D. Gujarati, "Basic econometrics mcgraw," hill publishers, New york, 2004.

[180] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[181] ——, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[182] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Scalable collaborative filtering approaches for large recommender systems," *Journal of machine learning research*,

vol. 10, pp. 623–656, 2009.

[183] R. Baeza-Yates and B. Ribeiro-Neto, "Modern information retrieval: the concepts and technology behind search," *New Jersey, USA: Addison-Wesley Professional*, 2011.

[184] K. Jarvelin and J. Kekalainen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.

[185] C. Musto, M. de Gemmis, G. Semeraro, and P. Lops, "A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews," in *Proceedings of the eleventh ACM conference on recommender systems*. Como, Italy, 2017, pp. 321–325.

[186] D. Paul, S. Sarkar, M. Chelliah, C. Kalyan, and P. P. Sinai Nadkarni, "Recommendation of high quality representative reviews in e-commerce," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 311–315.

[187] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2012, vol. 3.