AALBORG UNIVERSITY
COPENHAGEN

Semester: ICTE 4

Title: A Diversity by Design Recommender System for the Movie Application Domain

Project Period: **21/06/17 – 21/09/17**

Aalborg University Copenhagen
A.C. Meyers Vænge 15
2450 København SV

Semester Coordinator: Henning Olesen

Secretary: Maiken Keller

Semester Theme: Master Thesis

Supervisor(s):
Sokol Kosta
Jannick Kirk Sørensen

Project group no.:

Members
(do not write CPR.nr.):

Michele Zanitti

Pages: 76
Finished: 21/09/17

Abstract:

Recommender systems are powerful personalization tools which have seen widespread adoption across the Internet. However, it is thought that by emphasizing personalization through the optimization of accuracy-driven metrics, the issue of overpersonalization emerges with negative effects on the user experience. An acknowledged effect of this problem is the filter bubble, manifested when the recommendations and consumption cover only a selected portion of the catalogue, causing the user experience to narrow down on the long term. An increasingly popular countermeasure to the problem is offered by diversifying the recommendations even at the cost of reducing the accuracy of the recommender system.
In this thesis the possibilities of developing a solution to address the problem are investigated through the proposal of a recommendation system which implements the diversity by design for a movie application domain.
Building on past research, a user-centric framework to enhance the diversity on four related dimension, namely global coverage, local coverage, novelty and redundancy is presented. The proposed solution is designed to diversify users profiles, modeled on categorical preferences, within the same group in the recommendation filtering.
A proof of concept is developed to evaluate the diversification levels reached through the extraction of diverse within-group users against random extraction, with different levels of user diversity.

Table of Contents

List of Figures

List of Tables

1 Introduction

This chapter aims at introducing the reader to the subject covered in this work. With a general overview of the area of research, the motivations of the topic discussed are brought to light. Afterwards, it presents the specific problem hereby examined, together with the general methodology adopted to confront it. Subsequently, the chapter ends with the main contributions to the research field, providing finally the overall structure of this work.

1.1 Background and motivation

Recommender systems (RSs) are software tools developed to assist users when browsing and choosing from a vast collection of alternatives [1]: by exploiting user's preferences, recommender systems filter out irrelevant options and select only a personalized subset of items and their effect on user satisfaction has been proved to be positive in many domains, e.g. online shopping [2] and entertainment [3].

From an user experience-related point of view, RSs try to mitigate the phenomenon known as *information overload*, whereby people have difficulty in performing a decision when faced with an overwhelming amount of choice. This problem has been explored [4][5][6][7] even before the diffusion and subsequent adoption of information technology. Furthermore, since the advent of the World Wide Web and its democratization, the virtually infinite Internet shelf space increased the content production and availability exponentially, exacerbating the problem of information overload to unsustainable levels for the human mind [5]. Thus, recommender systems have been developed to aid the consumption of available information by learning the user's habits or preferences and selecting relevant options for her to be chosen from.

On a more business-related perspective, RSs have been acknowledged for their value of increasing revenue and as a tool to retain a company's customer base [8].

One obvious strength of RSs is the ability to personalize content suggestion to the user's preferences; another factor is defined by the coverage of such recommendations, i.e. how personalization can promote the discovery of niche content that, without personalization, would not contribute much on increasing revenue [8] or how, to put it simply, it can leverage the long-tail distribution of product consumption.

1.1.1 The filter bubble

More recently [9], there has been a debate around the ethics of personalization and the issue of recommender systems being 'black box' tools which, acting as digital editors, select content in place of the users who are oblivious of the inner workings of personalization algorithms.

In fact, as helpful recommender systems are in solving the information overload problem, it is argued that they created new problems when they are overly tailored to the user's habits. It is however still unclear to what extent a product can be considered "personalized". For recommender system, personalization is the result of filtering the items so that they match with the user's preferences or according to other criteria that generally suit the user's needs in a given context.

Moreover, when recommender systems know their users too well, they tend to recommend content which exactly match the user preferences, thereby promoting the consumption of such content and enslaving the users in a positive feedback loop created and maintained by ever-similar recommendations. This problem is the direct effect of over-specialization, also known as over-personalization and recommendation overfitting and has been popularized as *filter bubble* [9].

In essence, the filter bubble problem depicts a scenario in which a system, through personalized algorithms, encourages the selective exposure to information that supports one's own beliefs [9][10], increasing informational polarization. Moreover, as the problem is inherent to personalization, it potentially affects every domain in which recommender systems are operative: from news to social networks to entertainment focused domains and so on.

In political terms, it can be said that when the filter bubble problem is evident, recommender systems tend to become conservative (individually, for each user) than those promoting diverse items (even if these do not match users preferences) such as, for example, news articles having different point of views than those one is accustomed to or disagrees with. As a result, conservative recommender systems tend to *implicitly* suppress the consumption of non-personalized content so that the users will always get more of what fits them, narrowing their perspectives. In fact, the implicit filtering is emphasized as the primary cause for which the filter bubble is considered invisible and insidious, but cozy nonetheless, as it promotes stability (contrary to the dynamicity provided by the unfamiliar): *"the filter bubble doesn't just reflect your identity. It also illustrates what choices you have"* [9].

For users and a psychology perspective, the picture depicted by the filter bubble scenario is rather bleak, given that users get what they want without effort and not what they might need [11]. With apparent references to a Chomskyan dystopia, the filter bubble effect may even be exploited to control the user consumption towards specific contents, thus using recommender systems as propaganda/marketing tools. In this interpretation, the algorithms and data become the panoptic (as in "surveillance") means, necessary to for tailoring the content. However, it is argued that algorithms of data mining and big data already constitute surveillance tools, as they are used to keep track of the user decisions, behaviours.

From a business standpoint the filter bubble represents perhaps an unwanted risk, since it significantly limits the content space to choose from, which the user could exhaust in short time, with negative consequences in terms of customers retention.

In fact, it has been found that the filter bubble caused to narrow down the movie consumption for users who followed recommendations over time, even though users who did not follow recommendations narrowed their consumption more quickly [12].

While information overload has been acknowledged as a major driver of recommender system development, the filter bubble has become a side-effect brought by the (ab)use of personalization which, in conflict with the purpose of liberating the users of the decision cognitive load, it gave no options to choose from, no content to discover, being it already perfectly tailored to the users' preferences.

In this perspective, it can be argued that over-personalized recommendations limit both the user's growth potential in term of choice variety and the recommender system's coverage, thus the discoverability of items within the catalogue.

Moreover, in term of user satisfaction, it has been theorized a relationship between the amount of choices and subjective satisfaction, depicted by the Wundt curve [7], which essentially illustrates how too few choices on one side and too much diversity on the other can cause user dissatisfaction. Schwartz definition of his paradox of choice [6] takes the Wundt curve into consideration; however, he discusses more about information overload and user satisfaction related to too many choices than the problem of having too few options.

Instead, the relation between too few options and satisfaction could be linked to the filter bubble problem which would assume that few choices negatively influence the user satisfaction, since truly homogeneous items recommended by an over-personalizing RS hardly constitute a considerable number of choices. To offer an example, one can consider the genres of movies recommended: if the movies belong to the same genres preferred by the user, as a consequence of the filter bubble, they are treated as more similar than if they belonged to different genres, wth consequences on the user satisfaction. A more in depth elaboration of this argument is offered in the next section.

Nevertheless, the Wundt curve presumes that there exist a middle point that could increase user satisfaction, while both extreme choice scarcity and abundance could lead to decision paralysis. Hence, the answer to information overload, filter bubble and personalization might lie in the divesification of recommended items and how tuned is to a given user, instead of focusing on matching recommendations to user preferences.

1.1.2 Diversity: a countermeasure for the filter bubble

Having a clear idea of the over-specialization issues, their causes are hereby identified as the tendency of recommender systems to maximize their personalization tendency, also known as accuracy.

In evaluating recommender systems, it has ben argued that a great deal of published work focuses on accuracy optimization techniques and accuracy metrics, which generally measure how close are the ratings predicted by an RS compared to the actual user's ratings [13][14][15].

Nevertheless, Herlocker et al. claim that *"good recommendation accuracy alone does not give users of recommender systems an effective and satisfying experience"* [13], suggesting that there exist a trade-off between accuracy and recommendation usefulness and that a RS should provide both in hope to satisfy its users. On the aforementioned RS business value factors, catalogue coverage is perhaps the closest to this trade-off, as it relates to how much content a RS is able to suggest, in contrast to recommendations based on pure accuracy or even on popularity, since unpopular items are more difficult to recommend. Herlocker et al. also argue that accuracy enables a RS to suggest easy-to-predict items, i.e. obvious recommendations that do not have value for users [13], while popularity based recommendations are worse in term of both accuracy and usefulness, since they often include content for which users do not have interest.

It is therefore necessary to understand that whilst accuracy is important for user satisfaction, it is merely one ingredient and that recommender systems need to take other dimensions into account [16]. Perhaps accuracy is one of the most employed metric of RS evaluation, but other dimensions are increasing in recognition, such as novelty, diversity and serendipity which generally promote a more varied item consumption, as opposed to item homogeneity.

McNee et al. claim that *"recommendations that are most accurate according to the standard metrics are sometimes not the recommendations that are most useful to users"* [16] and argue that diversification metrics should be employed in order to judge the quality of recommendations, besides advocating an user-centric evaluation of recommender systems. In fact, an initial intuition of the downsides of optimizing accuracy has been to understand that these metrics penalize recommendations that do not meet the user preferences [16] and that these metrics do not take into consideration the user intentions, i.e. accuracy metrics are not user-centric as they implicitly assume that the user will be always interested in the items she already consumes: *"users often judge recommendations that are for items they would not have thought of themselves. the most common recommendation was for the Beatle's "White Album". From an accuracy perspective these recommendations were dead-on: most users like that album very much. From a usefulness perspective, though, the recommendations were a complete failure: every user either already owned the "White Album", or had specifically chosen not to own it."* [16].

One approach to solve the problem of over-personalization is to diversify the recommendations for the users, so that they do not meet their preferences completely, since they would expect to be recommended new, interesting items, compared to boring (but correct nonetheless) items [13].

Diversification in recommender systems mostly borrows its meaning and purpose from the field of Information Retrieval [17][18], which has been applied as a response to query intent ambiguity and to remove redundancy in the query results.

In Information Filtering (IF), the field of RS, the concept of diversity is similar but fundamentally influenced by the difference it has compared to IR: the IF field has the concepts of user modeling and profile, used to filter and present the content, while in IR these notions are absent. In this light, diversity is a tool that should consider the variety of user interests, given her profile: since in IR the diversity is the mean to provide the user with as many interpretations of a query as possible, i.e. query disambiguation, and there is no notion of profile, the consequence is that user preferences are not stored. Therefore, diversity is equally important to any other user in information retrieval, while this statement proves to be fundamentally wrong for recommender systems.

Considering that the filter bubble is a mostly ill-defined problem by nature, and given the increasing emphasis on recommendation diversity in literature both against it and as a way to increase user satisfaction [19], is what drives this project. The focus of this thesis is to understand the role of diversity in recommender systems not only from the mathematical perspective, but also by integrating the underlying psychological aspects of it, in order to provide a convergence between the two.

The heterogeneity in the definitions and approaches to include diversity and the metrics to evaluate it (in RS), discussed further in chapter 1, provide an extensive background on the problem.

1.2 Problem formulation

From the background on personalization, it has been acknowledged that accuracy is not always a desired outcome of recommender systems and that diversity is intended as a primary measure to prevent over-personalization.
Hence, the main research questions hereby addressed are:

Q1. *To what extent are decision psychology findings applicable in the design of a diversification method, compared to a pure recommendation system perspective?*
Q2. *How can user interests be modeled in the algorithmic solution?*
Q3. *How can diversity be incorporated in the recommendation process to suggest both varied and relevant items to users and to limit the effect of overspecialization?*
Q4. *How can the solution be evaluated?*

These inquiries focus on the actual utility of diversity intended as the desired effect of the recommendation process, therefore a solid foundation is required to determine how this measure affects user preferences and the discovery of new items. Moreover, an experimental evaluation is also needed to provide the answers to the above questions.

1.3 Methodology

In order to investigate the above research goal, the following methodology is adopted.
Firstly, the concept of diversity is studied to gain insights as to how is interpreted in the RS research community and in other fields, and to define how a user-centric approach to recommendation diversity can be pursued.
A comprehensive overview of current diversification solutions and metrics from RS literature is also necessary to understand the problem of diversity related to the field.
Therefore, the literature review takes into account also findings from decision psychology, consumer behaviour psychology and preference and choice psychology, in order to assess the importance of diversity in the recommendation task.

The second step is to analyze the psychology-related findings to gain insights on how diversity can be modeled in recommender system. Subsequently, an analysis of the diversification solutions and metrics explored is required to determine their limitations regarding the psychological meaning of diversity, which serve as the starting point for the creation of the recommendation approach with diversity by design.
The methodology of the proposed approach is described more in detail in the chapter 1.
Finally, a prototype of the approach is implemented in the movie domain to demonstrate the validity of the approach with respect to the diversification level reached and to provide an initial answer to the research questions. To this end, the Movielens dataset [20], containing movie ratings, is used with the metadata from IMDb [21] are utilized (within the allowed conditions of usage), which allow for an offline experimental evaluation.
Within the same chapter, to provide an initial answer to the research questions, the proof of concept is developed and evaluated on the static Movielens dataset.

Subsequently, the experiment's results are analyzed and discussed.

1.4 Delimitations

Here, the scope of the thesis is illustrated by providing the aspects not covered by the methodology formulated as above. In particular, it is personally acknowledged that the validation of the proposed approach requires ulterior work and experimental evaluations. Moreover, as the validation is limited to the proof of concept, reaching the actual recommendations is not feasible and it would be more reasonable to complete the implementation and validate the results with real users.

The research is focused solely on the diversity aspect of RSs, by providing an extensive -although not complete- overview of diversification methods and metrics, as well as relevant psychology motivations for embedding diversity in the recommendations.

The implications of diversity on user satisfaction are not considered, as this study demands an online experiment with actual users, which is left as a future perspective. Moreover, a comparison between the proposed approach to state of the art diversification methods is left as a future work, as the implementation is performed on the proof of concept. Another reason for not considering a comparison between the approaches lies in an apparent lack of generality of state of the art methods, which requires a thorough investigation for implementing and optimizing them to the application domain addressed here.

Other concepts closely related to diversity, such as serendipity, are not covered in detail, given the fact that they require other approaches and evaluation metrics to be incorporated in this work.
Topic areas related to recommender system design, such as the inclusion of contextual or cross domain information, privacy awareness and other issues affecting recommender systems (sparsity, cold start, etc.) are not considered. Nevertheless, it is acknowledged that diversity is also strongly tied to contextual variables [22], and a future development of the proposed approach would undoubtedly benefit from the inclusion of the diversification based on contextual data.

Finally, from the design point of view, only the algorithmic part is considered, while the development of the user interface and recommender optimization are left as possible developments of this work.

1.5 Main contributions

This work suggests proposes the following contributions to the RS field:

- A diversification framework to identify the dimensions on which diversity can be modeled is derived from an established theory and definition of diversity [23] and centered on the psychological factors related to user preferences.
- A diversification approach based on user preferences built on categories of items is proposed to select diverse users and recommends items from their preferred categories.
- As the proof of concept, a filtering method which models diversity by design, contrary to existing post-filtering methods, through the users' diversification is developed to demonstrate the diversification level the proposed approach aims to reach.

1.6 Thesis structure

The thesis structure reflects the methodology hereby adopted.
Chapter 1 focuses on the theoretical background, with an overview of the recommendation task and relevant definitions on diversity, with the application of the concept in recommender systems.

Chapter 2 focuses on the review of relevant diversification techniques and metrics proposed in the field of research. Moreover, also relevant machine learning and data mining methods to support the design and implementation of the proposed approach are presented.

In chapter 3 the motivations for incorporating diversity are discussed in relations to the user preferences from a sociological point of view. From the initial analysis on psychological motives and on the diversity theory from both the sociological and recommendation perspectives, a diversification framework is extracted. The framework, besides the theoretical purpose, serves also as the foundation for the analysis of the techniques and metrics and, more importantly, for the articulation of the proposed approach to diversification. The chapter continues with the analysis of the diversification techniques and metrics, in relation to criteria obtained from the diversification framework and from internal considerations, to determine the shortcomings and advantages offered by the state of the art approaches.

Chapter 4 aims at introducing the proposed recommendation model, which implements diversity by design through a modular procedure. In particular, the designed approach is divided into feature extraction, preprocessing, and recommendation (with embedded diversification) modules, the first two of which are directed to the transformation of the data to a format fitting the proposed diversification framework, while the last group of modules target the creation of a diversified recommendation list, taking into consideration the user current preferences.

Chapter 0 is focused on preparing the data by using the Movielens dataset and the IMDb metadata, relevant for the movie application domain, as the implementation of the first modules of the proposed approach (feature extraction). Therefore, the chapter includes a thorough exploratory analysis of the datasets and as a result, the preliminary user and item profiles are obtained.

Chapter 6 is focused on the implementation of a proof of concept of the proposed approach to evaluate the proof of concept, which concludes the list of the main contributions of this work. In fact, it is proposed to not consider the creation of a recommendation list, which is therefore left as a future recommendation. This chapter is divided to cover the preprocessing modules and the first module of the recommendation procedure. An evaluation, followed by the discussion of the results, of the preprocessing modules is provided to ensure that the hypotheses of this work are valid. Lastly, an experimental evaluation of the diversification method against normal recommendation models is performed on the last considered module.

Lastly, in chapter 0 the results obtained in chapter 5 are discussed in relation to the formulated research questions and proposals for future development of this work are formally provided.

2 Theoretical background

Defining the scope of research requires the knowledge of the theoretical foundations on which the problem has been already handled. Specifically, an understanding of the concept of diversity is essential. Therefore, this chapter aims at providing the foundations for understanding the problem confronted in this work.

2.1 Recommender system theory

As the rationale behind recommender systems has been clarified in ch. 0, here, it is worth to explain the general methods to achieve the recommendation task derived from the acknowledged RS taxonomy[1] provided by Burke [24] and further discussed in [25]:

Content Based Filtering (CBF) is a family of recommendation approaches based on individual user behaviours [25]: CBF methods recommend items matching the user's past preferences, e.g. if an user has a strong preference for a particular type of items, the system recognizes the features of such items as predictors for future preferences and thus recommends similar items based on those features (Figure 2.1, right side).

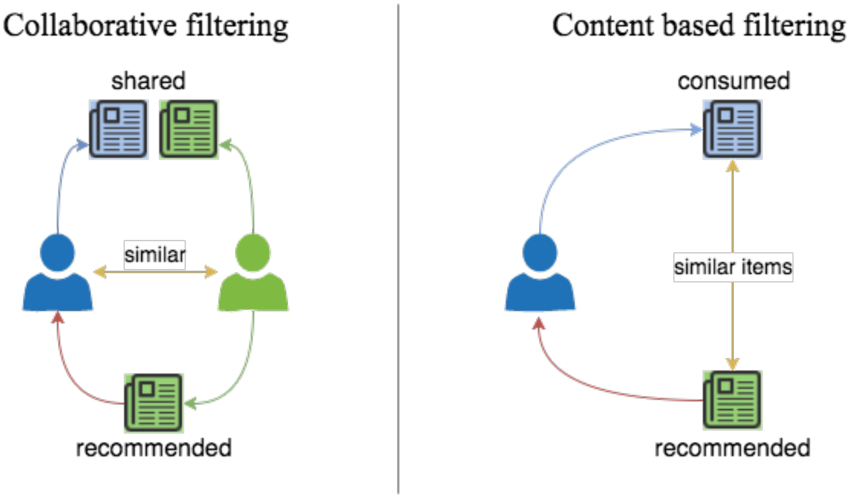


Figure 2.1. Content based (right) and collaborative (left) filtering paradigms.

Collaborative Filtering (CF) approaches instead look at the similarity among users' past behaviours to produce recommendations (Figure 2.1, left side) [25]: users who share habits or have rated the same movies are exploited to recommend unknown items to a given user, using a "word of mouth" approach [26]. CF methods are further divided in memory-based, which look at item-to-item or user-to-user similarities, and model based, which use machine learning models to learn user preferences and recommend useful items. Arguably, CF can be considered as a variant of CBF if users are treated as contents.

Hybrid Recommender Systems: the two approaches suffer from several shortcomings, which also affect the recommendation diversity. CF suffers from cold-start problems for both users and items, since the co-occurrence of items ratings is required, while CBF instead suffers from overspecialization, since it tends to recommend similar items to those liked in the past. These problems, in particular over-specialisation, can be mitigated by combining the strong aspects of both methods [25].

[1] About the complete taxonomy, it has been decided to not explain demographic and knowledge-based RSs, as they are generally suited for particular types of recommendations: demographic RSs are mostly adopted for website personalization, while the latter are useful to recommend complex products or services (e.g. education program, holidays, etc.).

2.2 A definition of diversity

With a framed theory of RS, a general introduction of diversity is required to provide the necessary constraints to the problem formulations. The concept of diversity has been studied in many fields to describe the property of a set of containing distinguishable elements, and is generally understood as *heterogeneity*.
In literature, from information filtering to biology and other fields, several techniques are available to quantify the diversity of a set, depending on the interpretation of the concept itself. Nehring and Puppe [27] suggest to measure it using a multi-attribute approach as the sum of the weights of the attributes of the elements in a set, instead of a simple aggregate pairwise dissimilarity. Junge considers diversity as two dimensional concept that includes the number of categories the elements can be classified into, and the homogeneity of elements across the categories [28].

However, the most complete work on diversity studies has been pursued by Stirling in [23], which identifies three properties related to diversity and proposes a general framework to analyze it:

- Variety: *"is the number of categories into which system elements are apportioned […] 'how many types of thing do we have?'"* [23], i.e., the number of categories present in a set, independently from the elements within each, is a signal of diversity.
- Balance: *"Balance is a function of the pattern of apportionment of elements across categories […] 'how much of each type of thing do we have?'"* [23]. As variety only measures the number of categories, this property assess the extent to which categories are equally represented, through the relative distribution of elements.
- Disparity: *"refers to the manner and degree in which the elements may be distinguished […] 'how different from each other are the types of thing that we have?'"* [23]. This property assesses the specificity of each category (i.e., how can be easily distinguished), determining the dissimilarity as a signal of diversity.

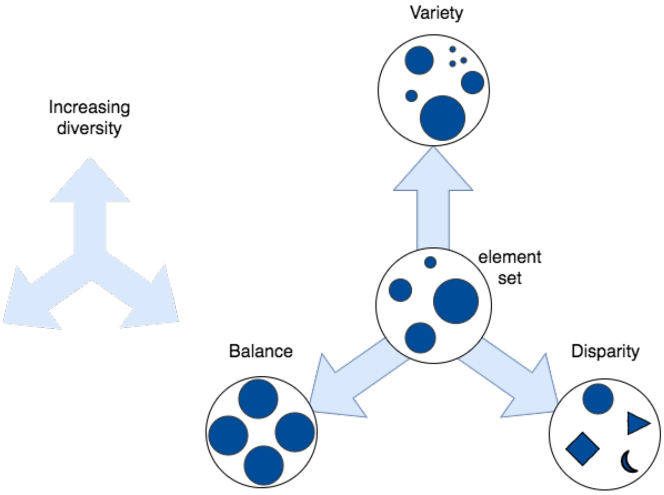


Figure 2.2. Relationships between variety, balance, disparity and diversity.

Figure 2.2 illustrates the properties of diversity as defined in [23], proposing that the diversity increases along with variety, balance and disparity and that taken individually, these properties are insufficient to define diversity.
In the next chapter, some of the relevant approaches to implement and evaluate diversity in recommender systems are introduced.

3 Related work

This chapter provides an overview of the different methods to achieve diversity and metrics to evaluate the resulting recommendations. Moreover, an overview of relevant data mining and machine learning tools and methodologies is delivered, which is necessary to understand the following chapters of the thesis.

3.1 Diversification methods in recommender systems

This section introduces to a selected overview of the state of art methods to incorporate diversity in the recommendation process.
In recommender systems, the diversification problem is defined as finding a set of elements, from the catalogue of items, that is optimized in terms of their quality and [18]. RS literature distinguish two paradigms of diversification methods depending on the level at which is achieved, namely diversity modeling and post-filtering approaches (Figure 3.1) [14].

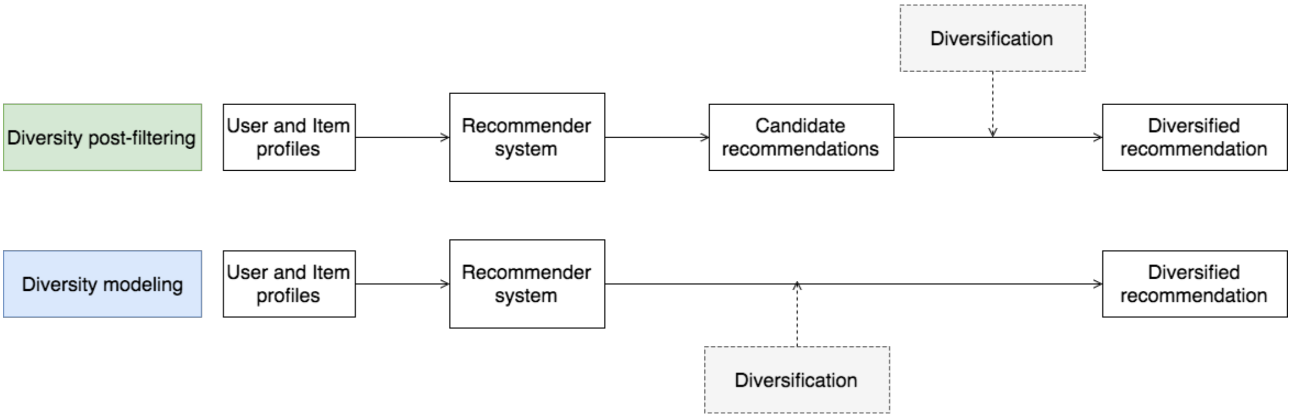


Figure 3.1. Levels of integration of the diversity modeling and post-filtering diversification approaches in the recommendation process.

The former solutions aim to enhance the filtering step by combining a diversification criteria prior to the extraction of a set of recommendations. Instead, post-filtering methods process the set of candidate items after the filtering step through re-ranking strategies in order to extract the subset that satisfy the specified diversification and quality criteria [29].
A similarity between diversity and context modeling in context aware recommender systems (CARSs) is noticeable [30] especially in relation to the step when both can be modeled, which, for CARSs are divided in pre-filtering, context modeling (i.e. included in the filtering method) and post-filtering. However, to the author's knowledge, there are no methods for pre-filtering diversification.

The nature of this work allows to examine a limited number of techniques which do not entail real user evaluations or adaptations to varying user preferences, of which, temporal diversity [31] seems to be the most promising, even though the latter proves to be an interesting line of future research in order to avoid producing the same recommendations, even if already diverse. In particular, the methods have been selected given their performance using datasets belonging to the application domain hereby addressed. It is important to mention that the intent of this review is not to provide a review of the performance, but to identify the key aspects on which each approach intends to enhance the recommendation diversity.
The work from Castells et al. [17] is interesting as it provides a unified understanding of diversity related to both IR and recommender systems, with an extensive survey on evaluation metrics; however, it gives a limited overview of methods to enhance diversity. Hence, the methods have been gathered from a limited source of works, notably [32], which offers an exhaustive survey of both methods and evaluation metrics for diversity in recommender systems.

- Ziegler et al. **topic diversification** [15]: this technique is devised to *"balance and diversify personalized recommendations lists in order to reflect the user's complete spectrum of interests"* [15] and falls into the post-filtering solutions as it takes recommended items in input to re-rank and extract a diversified subset of items. Interesting, but perhaps unsurprising, is the application of item similarity, a content-based metric, to diversify the item set.

- Vargas et al. **binomial diversity** [33]: instead of using item similarity, Vargas et al. proposed a definition of diversity encompassing the genre coverage, genre redundancy and the recommendation list size awareness, in light that the topic diversification does not address such parameters. Still, since it works by re-ranking an initial recommendation, it falls into post-filtering approaches. Here, it is interesting how the technique approaches diversity taking not only the similarity, but also coverage and redundancy: **coverage** is achieved through finding the genres present in a recommendation set, compared to all genres, considering also that users themselves have preferences over certain genres and thus, some are more relevant than others. **Redundancy** in turn is defined as the frequency of each genre in the item set. The resulting method thus aim to maximize coverage of genres considering the user preferences while decreasing redundant genres [33].

- **ClusDiv** [34]: this approach uses clustering to group items in the catalogue from the explicit ratings rather than on item descriptions (although this is not a necessary prerequisite), and recommends items from different clusters. Compared to re-ranking methods such as [15], using item clusters resulted quicker and achieved similar diversification results. The authors employed k-Means as the clustering method to generate clusters which are subsequently used to create a users–to-clusters weights matrix. However, as it takes a precomputed list of recommendations, it is still categorized as a post-filtering approach.

- **Neighbor Diversification** [35]: the diversification paradigm in Yang et al. is shifted from the items to the users, which propose to retrieve a set of diverse users, by using explicit ratings, to an active user; recommendations are then extracted from these distant neighbors. Diversity is evaluated also considering the catalogue coverage, the novelty and the accuracy. An unexpected (and perhaps serendipitous) finding is that the accuracy levels, in terms of precision and recall, as the user diversity threshold increases, do not drop and in some cases can also increase, thus suggesting that the trade-off between accuracy and diversity may hold when considering items, but other factors come into play when users are considered. Moreover, compared to the three methods described above, this is the only diversification modeling technique hereby discussed.

- **XploDiv**: In [18] Stirling's definition has been employed to suggest that the balance affects the trade-off between relevance and diversity and a novel diversification method has been devised to deal with the trade-off and with the user's openness tendency (to explore novel items or to exploit her preferences). This is a promising method which encompass most of the discussions on preference uncertainty, moreover, the parameters to control the two trade-offs are tunable and dynamically learned, to allow a fine grained control of exploitative or explorative diversity. Unlike the previous method, XploDiv is devised as a post-filtering approach and requires a set of recommended items.

3.2 Diversity evaluation metrics in recommender systems

In order to measure the diversity offered by the approaches introduced above, a number of metrics are required. Ge et al. [36] define a number of evaluation metrics that relate to the global coverage of the recommender algorithm and serendipity, described as the tendency to recommend unexpected but satisfactory items. As serendipity can only be evaluated by measuring the satisfaction of real users, the metrics have been selected for their feasibility to evaluate recommender systems with offline experiments, which do not require real users. Given a list of attributes on which items can be compared, the following evaluation metrics are discussed.

3.2.1 Intra list Similarity [15]

Ziegler et al. define this evaluation metric [15] as the aggregate similarity between pairs of items in a set so that the lower it is, the higher the diversity and vice versa. While dependent on a similarity metric, its advantage lies in the evaluation straightforwardness. This metric, however, does not consider the user's interest coverage, as it assumes that the filtering has already produced a personalized set of items only to be re-ranked. For all items in a list L and a pair of items i and j, ILS_L is:

$$ILS_L = \frac{1}{2}\left(\sum\nolimits_{i\in L}\sum\nolimits_{j\in L, j\neq i} sim(i,j)\right) \tag{3.1}$$

3.2.2 Binomial diversity metric [33]

Initially borrowed from IR, these metrics are interesting to the diversity problem, as they measure the heterogeneity and homogeneity of the recommendation set in a different way than ILS: coverage may refer to both the range of user preferences and the item catalogue width, while redundancy refers to how homogeneous are the items within the recommendation sets or between recommended items and previously consumed ones (on the latter, this interpretation of redundancy is closely related to recommendation novelty, since it assumes that a low redundancy means that recommended products do not share many attributes with the items already consumed)). A metric focusing on both coverage and redundancy is defined in [33] as the product of coverage and non redundancy of a recommendation set, which has been adopted to evaluate the binomial diversity approach proposed in the same paper. For a list L of recommendations, the binomial diversity is measured as:

$$BinomDiv_L = Coverage_L \cdot NonRedundancy_L \tag{3.2}$$

3.3 Data mining and machine learning techniques for recommender systems

This section focuses on relevant machine learning techniques for recommender systems which can be used to model user preferences and to implement a prototype of the recommendation module with embedded diversity. While there are many more ML methods and tools, the ones explained here have been chosen for their relative low complexity and the widely gained popularity received in the RS community. Moreover, a methodology to analyze the data in order to to apply these techniques is provided in the next section.

3.3.1 Latent Semantic Analysis

In this section, the technique of LSA is explained, with an emphasis on the words weighting scheme ($TFIDF$) and the dimensionality reduction method (SVD), as they are central to the recommendation technique proposed.

Latent semantic analysis (LSA) is an IR technique introduced by Deerwester et al. [37] used to group similar documents based on the co-occurrences of terms, with the intuition that documents are more similar to each other if they share the same terms. LSA employs a dimensionality reduction technique known as singular value decomposition (SVD) to map both terms and documents into a latent feature space, so that they can be compared more effectively.

LSA has been devised as an approach to retrieve similar documents based on concepts, instead of individual words [37]. By treating documents as bags-of-words, LSA performs the following steps:

1) First, the term-document matrix is constructed from the collection of documents, containing the word co-occurrences (term frequencies) of representative terms for each document. This steps assumes that the terms have been preprocessed and only the relevant ones are retained for each document.
2) the matrix is then normalized with the $TFIDF$ scheme, in which weights less the word frequencies (term frequency) shared by more documents (Inverse document frequency).

3) Singular value decomposition is used on the $TFIDF$ term-document matrix to extract a conceptual space where terms and documents that are closely related are also represented close to one another.
4) SVD reduces the dimensionality necessary to compare the documents to the number of "concepts" in the semantic space and the last step involves computing the similarity indexes between pairs of documents.

3.3.1.1 TF-IDF weighting scheme

Term weights in LSA are defined by the $TFIDF$ scheme which is explained in detail below.

The first step in LSA includes the construction of the term-document matrix containing the occurrences of terms for each document.

There are several ways to calculate term frequencies, the simplest is by mean of raw counts of terms. However raw frequencies can produce biases toward longer documents, in which same terms occurring more often than in shorter terms are weighted more. If the documents lengths are comparable, raw occurrences can be used, without any risks of such bias. Nevertheless, for simplicity only the raw frequency is considered. For a given term t and a document d, the frequency $tf_{t,d}$ can be calculated as:

$$tf_{t,d} = f(t,d) \tag{3.3}$$

However, weighting terms solely on their absolute frequency results in giving more weights to less discriminating terms, shared by many documents. Therefore, the second step in LSA focuses on the normalization of the term frequencies to mitigate the weights for terms occurring too often, using the inverse document frequency. Intuitively, the IDF scheme weights terms proportionally to the times they appear in each document, therefore rare terms are weighted more than common (or popular) ones. For a given term t, the IDF score is calculated as follows:

$$idf_t = \log\frac{|D|}{|\{d : t \in d\}|} + 1 \tag{3.4}$$

The idf is augmented with 1 to avoid ignoring words that might have null weight completely and also at the denominator to prevent division by zero [38].

Finally, the weight of the term t in a document d is defined as:

$$tf\text{-}idf_{t,d} = tf_{t,d} \cdot idf_t \tag{3.5}$$

To summarize, the tf-idf scheme defines how terms can be representative (term frequency) for a document and discriminant among documents (inverse document frequency).

3.3.1.2 Singular Value Decomposition

$TFIDF$ alone is useful to extract relevant features, however their high number often represents a performance obstacle when comparing documents.

Singular value decomposition is a matrix factorization technique adopted in LSA for its power to represent documents and terms together into a space of lower dimensionality. Given an m by n matrix X, it is possible to decompose it into three matrices (Figure 3.2): U and V containing left and right singular vectors are orthonormal, and S, containing the singular values along the diagonal entries: in particular, the columns of U and V correspond to the eigenvectors of XX^T and X^TX and the singular values in S are the square roots of the eigenvectors of XX^T and X^TX [39].

$$X_{m,n} = U_{m,r} \times S_{r,r} \times V^T{}_{r,n} \tag{3.6}$$

$$\underset{m\;x\;n}{\overset{X}{\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}}} = \underset{m\;x\;r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mr} \end{pmatrix}}} \underset{r\;x\;r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \dots \\ 0 & \ddots & \\ \dots & & s_{rr} \end{pmatrix}}} \underset{r\;x\;n}{\overset{V^T}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{r1} & \cdots & v_{rn} \end{pmatrix}}} \tag{3.7}$$

These matrices can subsequently be used to approximate the original matrix retaining a much smaller dimensionality.

It is important to note that the singular values in S explain the variation in the eigenvectors and represent the overall rank of X, which can be up to $\min(m, n)$.

Moreover, the peculiarity of SVD consists in how the singular values in S are ordered in their descending order of magnitude [39], meaning that the variance is explained more by the first singular values than by the last ones. Therefore, it is possible to find experimentally the number k, where $k < r$, of singular values which are more important in describing the latent dimensions.

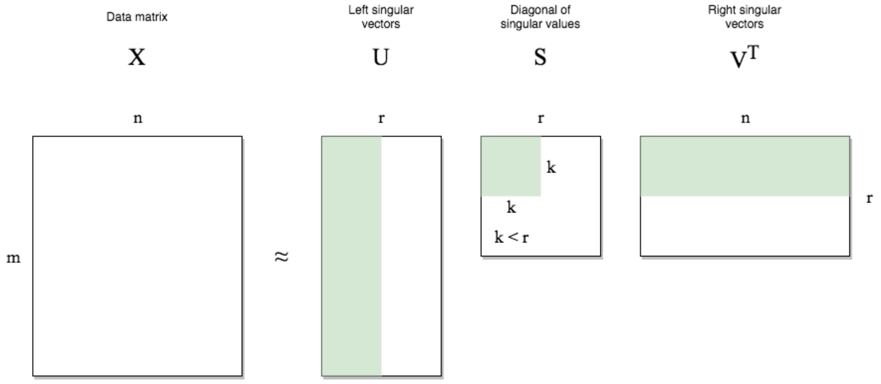


Figure 3.2. Singular Value Decomposition of X.

In LSA, SVD is thus exploited to capture the latent relationships between documents and terms, allowing a better extrapolation of document similarity. Intuitively, the retained singular values can be considered as the 'types' of documents, an abstraction statistically extracted from the single documents.

Nevertheless, the downside of SVD lies in the impossibility to explain the latent features, since intuitively, they are combinations of the original features and cannot be controlled directly; in this light, SVD can be considered as a black box method.

3.3.1.3 Similarity/distance metrics

The output of SVD is an approximation of the original matrix, however to define a document's neighborhood, a similarity (or distance) metric is required. The last step in the LSA algorithm entails this process, in order to extract the pairwise similarity between documents, with the the aim of immediate classification through nearest neighbors search, or to cluster similar documents together.

Therefore, a number of metrics useful for extracting document similarities are reviewed [40][41][42]. Other similarities such as Jaccard and Hamming, dealing with discrete data, are not considered since the documents (after SVD) are described by continuous features. Nevertheless, different similarity measures may yield different results, so the choice of a measure must be defined taking into consideration its purpose, the type of data and, specific to this work, the problem formulation.

From hereon, since similarity and distances are inverse to one other, the metrics are discussed as distance metrics, on prior understanding that the longer the distance between two documents, the lower their similarity and vice versa. Specifically, given two vectors u and v, the following distance metrics are examined (Table 3.1):

- *Euclidean*: the euclidean distance (also known as L2 distance) between u and v is defined in the euclidean space defined by their dimensionality k (after the truncation to the largest k singular values) as the square root of the sum of squared differences between the elements of the vectors [40], according to [41], it works well when the data presents compact structures.

- *Cosine*: unlike the euclidean distance, cosine measures the distance between u and v as the cosine of the angle between them, ranging from -1 for opposite vectors, to 1 for the same vector. It is important to note that this metric removes the bias depending on the vector magnitude, i.e. regardless of the proportion of the features. For this reason, cosine is also treated as a correlation metric. In fact, the cosine distance between u and v is defined as the dot product of the normalized u and v vectors. As cosine is originally a similarity metric, it can be transformed into a distance metric by subtracting the angle to 1 (Table 3.1). Nevertheless, according to [43] [44] an interesting relationship between the cosine and euclidean metrics is defined on the basis of the geometrical interpretation of cosine theorem applied to the normalized u and v vectors: in accordance to the cosine theorem $a^2 = u^2 + v^2 - 2uv \cdot \cos(\alpha)$, if $\bar{u}, \bar{v}$ are the normalized (that is, if it has unit length according to the Euclidean norm [44]) counterparts of u and v, then the formula becomes $a^2 = 1 + 1 - 2\cos(\alpha) = 2(1 - \cos(\alpha))$. Taking the square roots on each side, the euclidean distance can now be defined as $a = \sqrt{2(1 - \cos(\alpha))} = \sqrt{2(d_c(\bar{u}, \bar{v}))} = d_e(\bar{u}, \bar{v})$ [43]. Hence, the euclidean distance is equal to the square root of the doubled cosine distance between two normalized vectors, recalling that the cosine, in this case, is given by a simple dot product. Following this reasoning, the cosine distance gives the same answer to the euclidean distance in terms of ranked similarities, provided the application is on normalized vectors [44]:*"the order between document vectors is fixed notwithstanding the distance measure applied, either the cosine or Euclidean distance [...] it is possible to use some clustering and instance-based learning techniques with the Euclidean distance function so that results computed are consistent with those given by these methods with the cosine distance from the original data"* [43].

Distance metrics d(u,v)	**Formula**	
Euclidean	$d_e(u, v) = \sqrt{\sum_i^k (u_i - v_i)^2}$	(3.8)
Cosine	$d_c(u, v) = 1 - \dfrac{\sum_i^r u_i \cdot v_i}{\sqrt{\sum_i^r u_i^2} \cdot \sqrt{\sum_i^r v_i^2}}$	(3.9)

Table 3.1. Euclidean and cosine distance metrics

3.3.2 Discussion on LSA advantages and disadvantages

Because it looks at documents as bags of words and assumes that the document similarity is defined when same words are contained in different documents, LSA has been devised to deal with two issues in document retrieval: *polysemy* and *synonymy*. Synonymy occurs when different words may be used to describe the same concept and thus used in one document but not in another, thereby the model is unable to capture this similarity. Polysemy instead happens when the same word is used to describe different concepts (e.g. bank can have different meanings depending on the context). The application of SVD on the weighted term-document matrix has proven to mitigate word synonymy, however it is still prone to polysemy since *"the failure comes in the fact that every term is represented at just one point in space [...] as a weighted average of the different meanings"* [37].

It is argued that the polysemy and also the synonymy issues can be managed in the preprocessing step, that is in the extraction of meaningful terms for documents, perhaps by mapping each term to a cluster of synonyms, however, this kind of optimization is not the main focus of this thesis and such inquiry is left as a future recommendation.

Moreover, another issue inherent to bag of word models is the irrelevance of term order in documents, as only the occurrences are captured; for cases where documents are indeed treated as bags of words, such as metadata, this issue does not poses a problem since documents are already structured. However, for unstructured documents (e.g. articles in plain language) the order of terms may be interesting to keep.

As this work deals with structured documents, i.e. movies described by precise features and metadata, this issue, at this point, is not expected to impact the proposed approach.

3.3.3 Clustering algorithms

Since the grouping of documents can be treated as a problem for unsupervised learning, because there are no previously available document categories, it is possible to capture the patterns extracted by LSA (i.e. the groups of similar documents) utilizing clustering algorithms, illustrated in Figure 3.3. Finding clusters of documents requires a prior step of calculating the similarities among pairs of documents (recalling that it is the last step in LSA). Therefore, the last section of machine learning theory focuses on relevant clustering algorithms which use the previously explained distance metrics to group similar documents together.

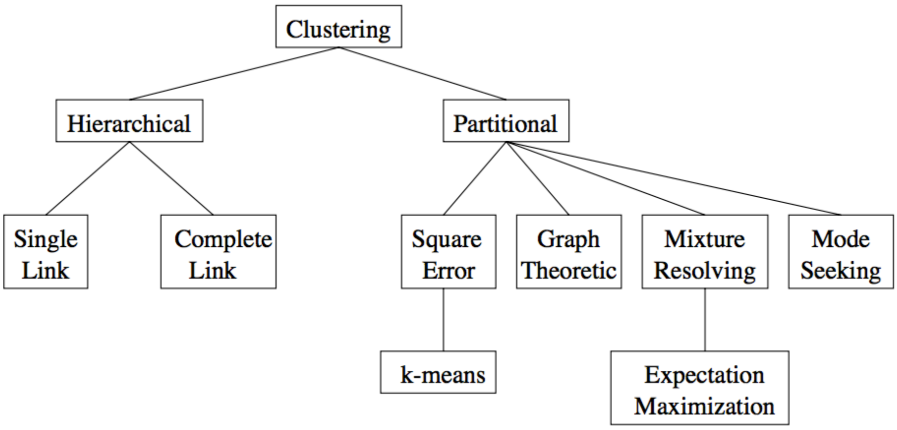


Figure 3.3. Clustering methods taxonomy [41].

Among these, the following sections focus on k-means and hierarchical clustering methods.

3.3.3.1 K-means

K-means [45] is a type of partitional clustering which tries to separate the data points into k mutually exclusive partitions, given the specified number of clusters k in input. The algorithm begins by choosing k arbitrary cluster centers (centroid) randomly or through an heuristic (k-means++) [46] for the initialization. Then, it assigns each data points to the nearest calculated center and recalculates the cluster centroids so that the within sum of squares function is minimized. This step is then performed iteratively until convergence is reached, that is, until the centroids stop moving significantly.

A drawback of k-means is its assumption that cluster have a spherical or convex shape, thus it performs poorly with elongated structures or irregular shapes. Moreover, the initialization of k-means using random centroids strongly affects the resulting clusters if the data does not form regular shapes.

3.3.3.2 Hierarchical clustering

Hierarchical clustering [47] is a family of techniques which permits to generate a different number of partitions depending on the level of granularity of each clusters, finding also the between-cluster affinity. This type of technique is often employed when the number of categories is unknown, thus allowing more freedom than partitional methods such as k-means, thanks to the visualization of the resulting tree structure (dendrogram). Hierarchical clustering methods are divided in agglomerative and divisive; due to the constraints in this work, only the agglomerative clustering is considered.

This type of clustering uses a matrix of previously computed pairwise distances as an input and starts with considering each element as singletons (single-element clusters), progressing in a bottom-up fashion by merging clusters together (by means of a linkage method) and creating super clusters until there are no more clusters to merge, in other words, the root has been reached.

Resulting clusters, depending on the linkage method for updating them, may merge differently; the most commonly used types (the reference to the original paper is provided for further understanding on the inner workings of the algorithm and the linkage methods) are [47]:

- *Single*: it is also called nearest neighbor, since it uses the smallest distance between the elements in two given clusters.
- *complete*: is instead called furthest neighbor, as it uses the largest distance between the elements in the clusters.
- *average*: calculates the average distance between all pairs of elements within the clusters.
- *Ward*: it assumes that the input distances are in the euclidean form and may produce incorrect results otherwise; understood that, the method minimizes the total increment of sum of squares (as the sum of the squares of the distances between the cluster's element and its centroid) at each agglomeration. Intuitively, this method has many affinities with k-means, as it operates in the euclidean space and they share a similar objective function (minimization of sum of squares), but without requiring the explicit number of clusters.

3.3.3.3 Clustering analysis and evaluation methodology

It is important to restate that different similarity/distance measures can yield different clusters [41], therefore also the method must be chosen according to the same rationale as in choosing the similarity metric.
The process of clustering follows the methodology as suggested in [41] and is shown in Figure 3.4. Clustering analysis methodology *[41]*.

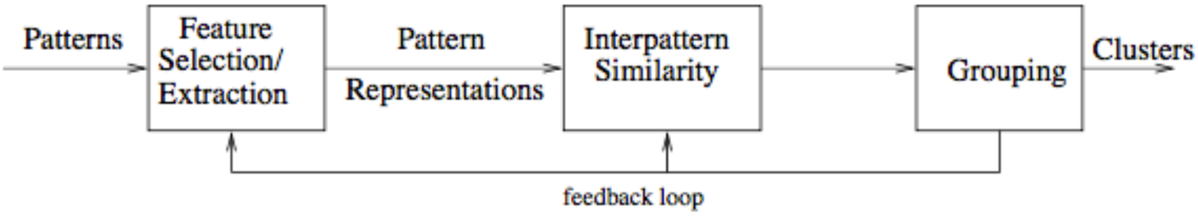


Figure 3.4. Clustering analysis methodology [41].

In particular, the parameters which allow to control the quality of the clusters is determined by the similarity/distance metrics employed (which may also depend on the type of clustering algorithm, since k-means does not work with distance matrices), the type of clustering algorithm, and the desired number of clusters. However, apart from the creation of clusters, ensuring their quality is vital.

3.3.3.4 Clustering evaluation as part of the clustering analysis

The problem of unsupervised learning methods lies in their power to accurately generalise the information contained in a dataset, in order to predict the best fitting category for unknown data. That is, if the trained model does not find a specific category for unknown data or always places data in the same category, there may be a fitting problem. Overfitting occurs when the model is too specialised on the training data to respond to new data [48]. Underfitting is the opposite problem, when the model is still unable to provide an accurate prediction because of its "shallow" understanding on the training data, therefore it performs poorly on both training and test data.
In order to mitigate overfitting, the evaluation of clustering algorithms is required. This kind of evaluation may differ depending on the method used, however it remains an important step in guaranteeing that the clusters have a good quality and can be adopted in real-world scenarios. The evaluation criteria include estimating the "right" number of clusters (by means of the "elbow" method) and finding the ratio between within-cluster and between-cluster distances (a measure defining how homogeneous and well separated are the clusters). Nevertheless, it is argued that these types of evaluations work better when the data points form noticeable structures.

According to [41], there are three types of validation studies: *"an external assessment of validity compares the recovered structure to an a priori structure. An internal examination of validity tries to determine if the structure is intrinsically appropriate for the data. A relative test compares two structures and measures their relative merit"*. General evaluation critera exist as a mean to automatically find the optimal number of cluster which, depending on the dataset distribution may yield different answers [49]: some criteria are prone to error when the dataset is skewed or presents noise and therefore, automatic evaluations of the clusters presents reasonable drawbacks.

An initial inspection of the clusters can instead be performed by employing an hierarchical clustering to subsequently look at the resulting dendrogram. The tree structure provides interesting insights regarding overfitting and underfitting: at the bottom the clusters are highly specialized and cannot be used to categorize data with enough generalization power, thus can lead to overfit with high probability. On the other hand, clusters high in the dendrogram represent very coarse categories which may include different groups of elements, therefore a model trained with few clusters may as well have a poor categorization power.
Hence, by analyzing the dendrogram and more specifically the distances between the merging points, could result into finding clusters of better quality.

In order to directly assess the quality of clusters, one might look at the elements contained in each cluster and determine the average distances between the elements, as well as the average distance between the elements of other clusters. Other measures can look at the inner distances and the intra-distances as defined prior to the application of the clustering method, in order to understand how, on average, closely related are the elements in a group, compared to other groups. This, intuitively may look at the types of elements contained so that if two clusters share similar elements (i.e., are similar to one another), they might or should belong to the same cluster [50].
Silhouette analysis can be taken into consideration provided that the clusters are well separated and densely populated; as these hypotheses may not work depending on the data nature, the comparison of cluster tightness and separation is still an open ended question.

4 A proposal for a user-centric diversification framework

As the concept of diversity has been investigated from the pure recommendation system perspective, in this chapter the implications of psychological insights on user preference learning are analyzed. As a primary contribution of this work, a novel, user-centric diversification framework is proposed, following the directions offered in [16] and inspired by Stirling's definition of the concept [23], adapted to the recommendation task: four general dimensions are thereby identified where diversity can be enhanced for the active user namely, local coverage, global coverage, redundancy and novelty.

4.1 Diversity in Recommender systems: an user-centric overview of motivations

The motivations of incorporating diversity have been largely discussed in chapter 0, in relation to the filter bubble effect of over-specialisation and to the poor quality (i.e. obviousness) of recommendations resulting from methods which prioritize accuracy over other measures. Here, the concept is taken from the user's psychology perspective, in order to provide a foundation as to why diversity can be beneficial for the user experience, in terms of discovering new items and of limiting the choice complexity in decision processes.

Recommender systems are a reality already embodied in people's life; as such, understanding how people develop preferences for certain items, behave with recommender systems and consume content is needed to provide valuable insights to the RS community, given the relatively dim presence of psychology concepts to support recommender algorithms. Therefore, this work provides an overview with relevant concepts with the aim of designing the recommender module, taking diversity as a central standpoint.

4.1.1 On user preference uncertainty and preference learning

Generally, the rationale of recommender systems is to suggest items personalized to the users tastes; however, it is also acknowledged that user preferences are not everlasting and are more likely to change over time. Castells et al. also have specified that *"user interests are complex, highly dynamic, context-dependent, heterogeneous and even contradictory"* [17]. Therefore, the uncertainty on user preferences plays a significant role and RSs should work toward the interpretation of the available evidence of user preferences, which may come either explicitly (ratings) or implicitly (retention time and click-stream measures).

Moreover, as RSs encourage users to browse instead of searching (as in IR), such systems are mostly used by users without clear needs, who simply like to get entertained, informed or educated without putting excessive cognitive efforts.

As such, RSs should allow users to find items which conform with their range of tastes (once they have clear preferences) and discover items they did not know existed or wanted. Hence, RSs potentials can be uncovered through the discovery and the satisfaction it comes after having stumbled upon an unexpectedly good product.

In this light, diversity can be understood as a paramount aspect to support these values. Hence, two interpretations of the concept can emerge from this discussion: diversity related to one's preferences is mostly an internal measure, as it is related to how a RS can reflect such range of interests, or **local coverage** [14], whereas diversity related to item discovery corresponds with the concept of **novelty**, the external measure of how different the items are compared with the user profile [14].

Therefore, the diversity of a recommended set of items in relation to the user profile, can be evaluated to measure both how well her interests are covered (local coverage) and how the recommendations differ from her interests (novelty).

For example, if the set has items from same or similar categories, they may reflect the user's strongest preferences, but fail in covering other categories not been explored extensively and in providing "anchoring points" for future preferences, or for establishing blurred interests. Related to the user tendency to discover novel items, psychology of choice and preferences suggests that people can be divided as maximizers, who always try to make the best possible choice across every option available, and satisficers, who settle for good enough, sub-optimal options [51]. Moreover, the internal diversity of the same set of items can be measured without referring to the user profile. This third metric can be interpreted as the **global coverage** of the recommended items [14], i.e. how well the system can cover the inherent heterogeneity of the items' catalogue, and is highly related to the novelty it could bring to the user since it does not consider user categories of preferences, but the full range of item types. As the catalogue often encompasses a high variety of items, measuring the general coverage of the recommended set may turn unfeasible and counterproductive when the user profile is taken into consideration: given that users may have clear negative preferences for specific categories, a logical assumption is to not recommend such types of items, in order to ensure a general level of item relevance.

It is argued that diversity, as a general heterogeneity of the recommended items (thus focusing on the global coverage), provides an interesting value, provided that some criteria are met regarding the items' nature. Depending on the application domain and the type of decision process, both a high and low item heterogeneity are considered as a value:

- For informational goods (hence, the realm of books, movies, news, etc.), high diversity is acknowledged as a value both because an heterogeneous set is attractive *per se* [52] and because deciding on which item to select is regarded as a low cognitive effort.
- While domains where items are concrete products and for which decisions are likely to impact the lives of the users (cars, houses, jobs, holidays, etc.), diversity is perhaps regarded as a riskier addition to the recommender system [53], hence, the user may be more content if there are less options to choose.

In [19], the effect of increasing item similarity in the recommended set of movies has proven that similarity is directly proportional to the cognitive effort in the decision process, notably due to the subtler comparisons on the items attributes. In fact, a notable result from their first experiment has been to understand that the higher the diversity of the recommended items, the lower is the perceived choice difficulty. For example, the user might benefit for lists of items that are mutually diverse, as this kind of heterogeneity allows the user to compare items on more superficial attributes (e.g. genres, cast, etc. for movies) than if the list was composed of more similar items. This thesis reasonably illustrates the choice difficulty resulting from highly homogeneous recommendation, which are already personalized but often not relevant for the user's needs.

Diversification is highly subjective and users may have different degrees of openness towards it [54][55], nevertheless the concept has been largely studied in relation to consumer behaviour and particularly, to the tendency of seeking variety [56][22], which can be explained through internal psychological factors (such as the desire for the unfamiliar) as well as external contextual variables. An important result, perhaps expected, is that the diversity seeking behavior grows as the user becomes satiated with the types of products she has experienced, in other words, when the user becomes bored, she searches for new stimuli in order to establish new preferences [56]. This is in line with the study on maximizers vs. satisficers, as the two profiles can be explained by respectively having high and low points of stimulation.

For example, watching solely comedy movies represents a strong preference for that particular genre and is important to consider, however if the user continues to consume similar titles, she might grow bored and instead seek different titles to watch, if only internal motivation factors are considered. Therefore, recommending comedy movies may seem the obvious response in order to personalize the movie catalogue for that user; in reality, suggesting the same content will confine the user to the already discussed filter-bubble, with no room for discovering new potentially interesting content.

In other words, there seems to exist a relationship between familiarity with a kind of content and preference, however it is also true that familiarity does not always translate into the user's current needs. Hence it is posited that the more the user is familiar with a particular kind of content, the lower is the point of satiation for it: as the user consumes many similar items, she becomes satiated more quickly. On the other hand, if the user has discovered a surprisingly interesting title, it would be useful to recommend more similar items to that than to individually popular ones, as she might be more interested into developing the new preference, as she has not yet reached a satiation point for it. Nevertheless, the user's openness to new experience must also be taken into consideration in this discussion, as some users may simply have high satiation points, i.e. are strongly tied to the familiar. An intuitive explanation of the openness to new experiences is suggested by the interpretation of recommendation novelty: it is hypothesized that more open users may be more prone to consume unfamiliar contents than "closed" ones, that is, they may experience a vast amount of items and in this case, their openness

can be detected by their tendency to consume always diverse content. On the other hand, closed users feel more safe when consuming constantly similar items, hence, their detected variation in which types of items they consume is presumably low.
As such, diversity can be regarded also as an important factor for the generation of new preferences in RSs when users become bored with the same recommendations.

4.2 A revised user-centric diversification framework

From the discussion on user preferences, it seems relevant to understand the irrationalities of user preferences, notably on their formation and development related to kinds of items. In this light, Stirling's definition of diversity [23] perhaps requires an adaptation to the human factor. From the original definition, the absolute diversity of a set increases as variety, balance and disparity of categories increase; however, it is argued that in an user-centric framework, diversity must be connected to the user's preferences. Therefore, in this section, the proposed framework is described.

It is hypothesized that users, as individuals, perceive diversity as satisfactory depending on their needs [56] and their openness to experience varies accordingly. Also, as users are satiated by the consumption of similar products, their needs may also vary in order to allocate space for new interests: concept-drift (the temporal changes of preferences) in user preferences is inherent to the human nature of avoid boredom and seek satisfaction by taking different stimuli when necessary. This is a common scenario when there are no more movies of a particular kind to watch for an user.
The temporal patterns of changes in concept drifts, have been identified in [57], however, an investigation on of such patterns on user preferences is not in the scope of this work.
Nevertheless, by acknowledging the concept drift in user preferences, it is assumed that users can be treated as mixtures of types of items, meaning that their profiles can be partitioned according to categories of elements, which are highly related to Stirling's idea of categories.

Hence, Stirling's definition can be adapted to accommodate the idea of users as mixtures and that categories of elements are weighted differently, into the proposed diversification framework (Figure 4.1).

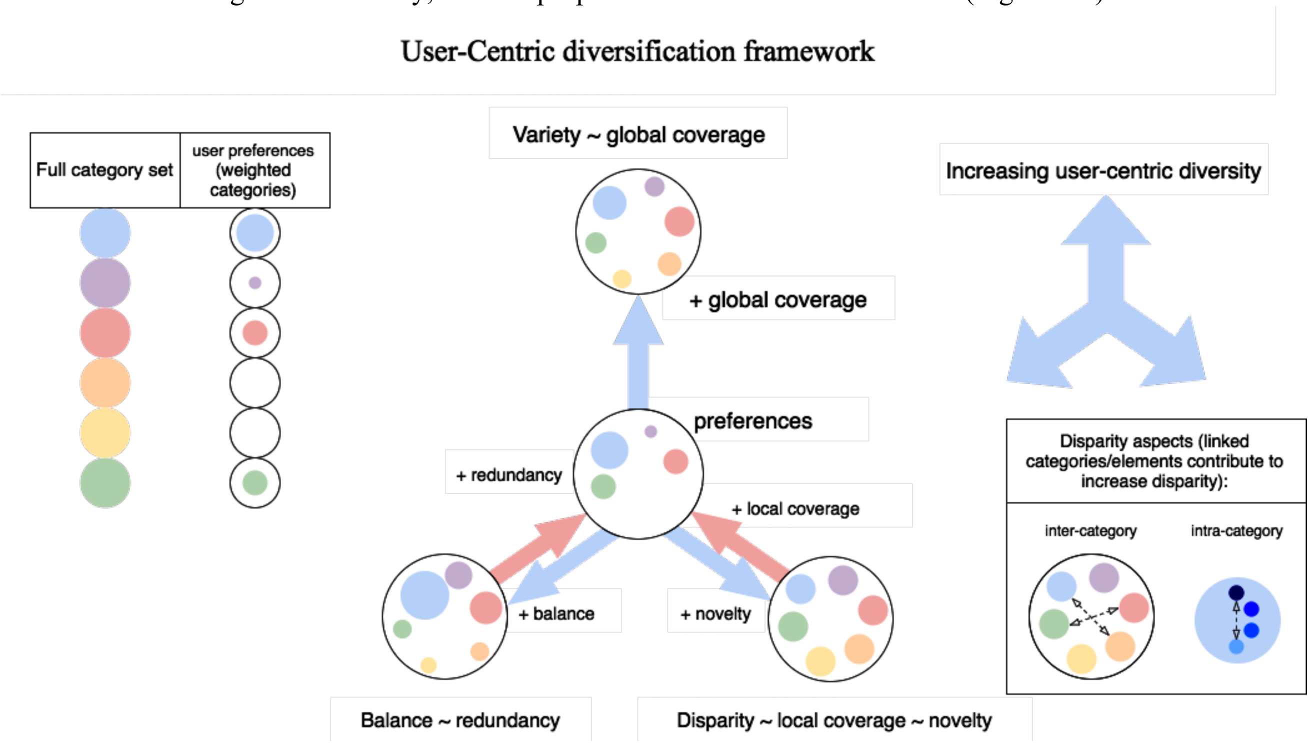

User-Centric diversification framework
Full category set
user preferences (weighted categories)
Variety ~ global coverage
+ global coverage
preferences
+ redundancy
+ local coverage
+ balance
+ novelty
Balance ~ redundancy
Disparity ~ local coverage ~ novelty
Increasing user-centric diversity
Disparity aspects (linked categories/elements contribute to increase disparity):
inter-category
intra-category

Figure 4.1. The proposed user-centric diversity framework, controlling coverage, novelty and redundancy.

In fact, the limitation of Stirling’s definition, which led to the articulation of the proposed framework, lies in how equally important the categories are treated, diversity-wise, whereas users have defined preferences over certain categories either implicitly or explicitly, impacting the perceived level of diversity.

In particular, user preferences would undoubtedly conflict with the property of balance, which instead assumes that *“the more even the balance, the greater the diversity”* [23]. Hence, a relaxed version of the diversity definition would allow to adjust the level of personalization for a particular user (in terms of balance), in relation to her preferences, while maintaining an acceptable level of heterogeneity thanks to the other two properties of variety and disparity.

Therefore, the more skewed the distribution of categories for user preferences, the less the balance and therefore, the apportionment to categories for the recommendation purpose should reflect this distribution, in order to achieve a personalized diversity which allows the existence of “good”[2] redundancy.

On the other two properties of diversity (variety and disparity), it can be said that they are related to the coverage of user’s preferences as well as to the global coverage (the full taxonomy) and to the novelty: the more categories from the taxonomy are present in the recommendations, the greater the variety and the more diverse are the categories to the active user, the greater the disparity, resulting as a greater diversity.

Disparity is also thought to be more related to novelty than to coverage, as there seems to be a trade-off between coverage of user preferences and novelty. In particular, coverage and novelty can be interpreted as measuring the same aspect in opposite ways: the former dictates that the more categories of user experiences are covered, the less the novelty (which is a logical conclusion to the fact that if the preferred categories remain the same, there is no added diversity); novelty, on the contrary, measures the extent to how the categories deviate from the preferred ones: the more novel the categories, the more this property grows.

Moreover, given that categories have been derived from the elements and thus are an abstraction, there can be two fronts on which to consider the novelty (hence, disparity): an high level interpretation considers the inter-categorical added disparity, while a lower level novelty focuses on the heterogeneity within the categories. This last discussion opens new inquiries to understand how categories can be compared to one another, as well as comparing elements within a categories. Presumably, categories form a hierarchical structure which also define the intra/inter-similarities to allow this kind of comparison. If proven, there are at least two levels at which disparity can be measured, as categories that are distant from the preferred ones may suggest an increase in novelty, as well as elements (or better, sub-categories) that belong to a category of interest but that are distant from the experienced elements may increase the overall novelty of the recommendation set.

4.3 Diversity evaluation metrics for the proposed framework

As the metrics introduced in ch. 1 work by taking into consideration their application to *ad hoc* approaches, it is reasonable to assume that they might perform differently depending on the diversification approach, on the application domain and on the type of diversity evaluated. In this section, following the proposed framework, a diversification metric is hereby adapted to meet the concept of a taxonomy of categories. Moreover, as the framework lacks the metrics for considering the identified dimensions, this section completes the framework by proposing additional evaluation metrics focusing on redundancy, novelty and coverages.

4.3.1 Categorical intra list similarity

Following Stirling’s categorical diversity, an adaptation of ILS is proposed to consider the categories in which the recommended items are apportioned. Hence, the category-based ILS for L, given that the categories of a recommended item i is C_i, can be defined as:

$$category\_ILS_L = \frac{1}{2}\left(\sum_{i \in L} \sum_{j \in L, j \neq i} sim(C_i, C_j)\right) \tag{4.1}$$

[2] The extent to which redundancy is considered good depends on the single user’s perceived diversity and satisfaction, however, without online user experiments, understanding this aspect thoroughly is unfeasible.

where the similarity of a pair of categories is defined by how they were constructed (e.g., the similarity metric employed in the clustering analysis, as described in section 3.3.1.3) can be utilized in the same manner, but considering the categories as simple elements. In fact, this metric can consider the disparity contained in the recommendation list, measured by the similarity by the categories contained.

4.3.2 Local and global coverage

In Castells et al. [17], a number of metrics have been reviewed for their specific purposes: some are more diversity oriented, while others are specialized in measuring the novelty (unexpectedness measure) or the coverage brought by the recommendation set (aggregate diversity), hence, as the coverage and redundancy have been thoroughly discussed, the following two metrics are focused on measuring these aspects.

This aspect can be considered either locally, in terms of user experience, or globally as the ability of the recommender to consider multiple item categories. However, for the diversification and personalization trade-off, the first interpretation of coverage may seem more appropriate, as it relates to how many categories of well-perceived items the recommender is able to capture for individual users and follows directly the revised definition of diversity as it can be interpreted with the property of variety. Hence, only the preference-wise definition of coverage is considered and the following metric takes into consideration the categories the user is already familiar with, C_u, and the categories of the recommended items C_L:

$$local\_coverage_{L,u} = \frac{|C_u \cap C_L|}{|C_u|} \tag{4.2}$$

Alternatively, to calculate the coverage on the system level, the following formula can be adopted:

$$global\_coverage_{L,U} = \frac{|C_U \cap C_L|}{|C_U|} \tag{4.3}$$

Where $|C_U|$ is the aggregate number of categories of which all users have experience.

4.3.3 Novelty

While coverage considers user experience categories as the target, novelty takes the categories as a starting point to find new, not experienced categories of items to recommend. Hence, there is a noticeable trade-off between coverage and novelty, since the former aims to find items that are true to the user preferences, while novelty encourages the discovery of new types of elements.

Moreover, without following Stirling's definition, this metric can be considered element-wise, since within the same category, the user may perceive diversity on a lower level than inter-category-wise.

Castells et al. [17] review a novelty metric related to the user profile as the level of item "unexpectedness". In principle, interpreting Murakami et al. [58], the unexpectedness level is defined as the difference of the recommended items to the expected (as obvious) set of recommendations. However, since it works on single items, the metric should be adapted to consider the user profile in terms of item categories. Therefore, the level of unexpectedness for a recommendation set L, considering the expected recommendation categories C_u for user u, is defined as follows:

$$Unexpectedness_{L,u} = \frac{|C_u \setminus C_L|}{|C_u|} \tag{4.4}$$

Where, the item categories of the recommended list are C_L, in which each item is apportioned, and C_u is similarly defined but for the items the user has already experienced.

4.3.4 Redundancy

Lastly, to cover all dimensions of the proposed framework, a metric for estimating the redundancy obtained with a list of recommendations for a given user -recalling that redundancy is equivalent to the property of balance- is hereby formulated as the amount of items falling in the same category. As such, this metric can be regarded as an augmentation of the local coverage, where for each category covered by the user profile, the amount of items relative to the size of the recommendations list is inspected. It can be argued that the category-based ILS can be treated as a redundancy metric if the similarity between categories is a simple match which returns 1 if the categories are the same and 0 otherwise.In this case, the proposed metric considers only the categories C_u covered by u, which are the expected ones (contrary to novelty), and for a recommendation list L, the "good" redundancy can be estimated as follows:

$$redundancy_{L,u} = \frac{1}{2}\left(\sum\nolimits_{i \in L, C_i \in C_u \cap C_L} \sum\nolimits_{c \in C_U} sim(C_i, c)\right) \tag{4.5}$$

$$sim(C_i, c) = \begin{cases} 1, & C_i = c \\ 0, & otherwise \end{cases} \tag{4.6}$$

Where the redundancy is the $category\_ILS_L$ of each covered category with respect to the recommendations list.

4.4 A comparison of diversification techniques to the proposed framework

The relevant diversification methods and metrics explored in ch. 1 are here assessed for their advantages and disadvantages, particularly in relation to the proposed diversification framework.
As Stirling's definition entails the concept of elements and categories, as well as the concepts of redundancy (balance), coverage (variety) and novelty (disparity), it seems reasonable to review the methods and metrics for their ability to manage the multi-faceted aspect of diversity.

However, for the limitations of this work, it has not been possible to pursue the diversification for concept-drift scenarios, which would impact the diversity balance in recommended sets, as this investigation would require different evaluation settings.

The methods illustrated in ch. 1 introduce diversity on different perspectives, from item-to-item dissimilarities to user diversification, incorporating important factors such as coverage, redundancy and user interests range in the extraction of the final recommendation set.
Looking at the diversification methods, it is clear that most of the solutions are post-filtering approaches. However, as suggested in [32], applying diversity within the filtering process may bring benefits in relation to the final output because it is possible to better control both the filtering and the diversification to, for instance, prevent the filtering from producing homogeneous recommendations which are problematic to diversify.
Hence, the criteria to analyse the approaches reflect the advantages and the limitations each one brings and are derived from :

Is it a *post-filtering* or a *diversity modeling* approach?
The distinction on post-filtering and diversity modeling is important, as the former methods require that the filtering produces an already diverse recommendation set, simply to re-rank, which may perform poorly when the candidate set is not diverse [18]. In contrast, diversity modeling, while less popular, is linked to the filtering process and does not require an initial set of recommendations. Moreover, in literature, the diversity is often described as a tunable parameter [18] in both post-filtering and modeling approaches, therefore, with a clear control on the output, diversity modeling techniques could perform better than re-ranking methods.

Nevertheless, suggesting that diversity should be incorporated directly in the recommendation algorithm also results in a tight coupling between filtering and diversity, while post-filtering approaches are considered as expansion modules to the recommendation engine and may allow for more flexibility, specifically for separated control on the filtering and diversification outputs (even if the latter depends on the former). This criterion suggests how generally are devised the methods and may offer further understandings regarding the design of a novel method.

Does the method allow to control the diversity-relevance trade-off for each user?

A pivotal aspect of the diversification method is to allow to balance the trade-off is important to ensure the satisfaction of each user and, when satisfaction is not directly measurable through online experiments, the trade-off should optimize the degree of relevance of an item and the overall diversity that a recommender set brings to the user.

Does the method consider coverage and redundancy of user preferences and item catalogue?

As plenty discussed, these properties are important in the relevance-diversity trade-off, as they control the variety, disparity and balance of diversity of the elements in a set, as such they are paramount in ensuring the user satisfaction in cases when some are more open to new items and vice versa. Nevertheless, an important aspect of diversity is to allow the users to explore novel items, as well as novel item categories.

Does the method consider item categories or item similarities?

According to Stirling's definition, diversity is subordinated to a prior definition of an element taxonomy, which constitutes a set of categories containing relatively homogeneous elements. In the movie domain, such categories may be represented by genres or more metadata, or they may be built upon aggregated sets of metadata, when the sets consist of non disjointed categories, as the case for movie genres [33].

Is the method considering item diversity or user diversity?

This aspect is mostly related to the method's design. From the discussion on the filter bubble, it seems the issue behind it is the polarization of user interests towards particular item categories or other trends. This criterion is supported by the findings in [35], which suggests that diversifying users have the same effect of diversifying items and may improve the relevance of items, thus the accuracy. While user diversity is dependent on the preferences of other users, ensuring item diversity may instead be beneficial for distinct users as it allows a more fine grained control on diversity.

The two approaches can be compared to collaborative and content based filtering methods, as user diversity leverages similarities among users and item diversity exploits item diversities based on either explicit user preferences or metadata.

This criterion may offer interesting insights regarding the design of the diversification approach, taking into consideration the filter bubble problem.

Diversity method	TOPIC DIVERSIFICATION [15]	Binomial diversity	Cluster Diversity	Neighbor diversity	XploDiv
DESCRIPTION	Taking a list in input, re-rank the list.	Includes the user's interests coverage and accuracy.	Applied after recommendation .	Select diverse users	control of exploration/exploitation trade-off, control on relevance/diversity trade-off.
Diversity modeling method	Post-filtering	Post-filtering	Post-filtering	Diversity modeling	Post-filtering
User-based relevance vs diversity trade off control	No	yes	yes	yes	yes
Coverage and redundancy control	yes	yes	No	yes	yes
Item categories or similarity	similarity	Categories (genres)	Categories (clusters)	User similarity	Similarity to recommended set
Item or user diversity	item	item	item	user	item

Table 4.1. Comparison of the diversification techniques on the criteria.

The analysis on the criteria has been summarized and the results are shown in Table 4.1. It can be concluded that most of the selected methods are based on prior filtering algorithm and thus do not allow to understand how the recommendations were created on the first place (as in which filtering method has been employed, for example).

The diversification vs relevance trade-off has been acknowledged by all methods, along with the importance of tuning the trade-off on an individual basis. However, topic diversification requires an extension in order to include this parameter, as already concluded by Ziegler et al. [15].

Increasing importance has been given on the coverage and redundancy aspects of recommendations, which, despite being borrowed from the

Most approaches assume that the diversity works on the basis of categories, supporting Stirling's definition, instead of adopting low level item similarities.

All but one reviewed approaches consist of diversifying items, instead of users, perhaps since the concept of item-based diversification has become an implicit standard in literature. Nevertheless, the user-based approach seems promising and could be investigated further.

4.5 A comparison of diversification metrics to the proposed framework

Hereby, a comparison of the metrics is conducted with respect to the proposed framework; the results of the comparison are summarized in Table 4.2.

The metrics presented in section 3.2, apart from the adapted coverage and novelty metrics, are considered state of the art in evaluating diversification approaches. In fact, most of the other metrics used in recommendation diversity literature generally derive from Ziegler's (ILS itself is just an adaptation of the average pairwise distance [59]), due to its flexibility, while the binomial diversity metric is interesting as it combines coverage and redundancy, allowing to measure diversity in a more user-centric fashion, but is limited in considering only one side of the items, which, in [33] is represented by the item genres. Moreover, it has been hypothesized that the metric may perform better than Ziegler's if is able to work on additional metadata apart from genres [32].

However, it is argued that the ILS can be adapted to cover the similarities between item categories, instead of single items so that, while at the item-ILS may focus on the aggregate diversity of the set on a lower level, category-ILS focuses on the redundancy of item categories.

In fact, categorical-ILS is closely related to the concept of redundancy, as already discussed (sect. 4.3.4), as it measures the aggregate number of the same categories of items in the recommendation set, if the similarity metric between categories is considered as a simple match between them is employed.

On the other hand, a simple match will return the ILS score independently of the inter-relations between categories, since depending on how they have been constructed, some categories may not be completely different. As such, some degrees of similarities are to be expected, since categories form a taxonomy which can be hierarchically organized. For example, again taking the movie domain, a category of comedy movies may be more related to a *drama* category than to a *mystery* or *horror* category, even if *comedy* and *drama* can be perceived as completely different, following a match-based similarity. Hence, the category similarity may be defined after the categories have been constructed.

The discussion on the category-based ILS intuitively brings the analysis on which diversity property is tackled, depending on the adopted similarity metric. Hence, a distinction between "sharp" and "fuzzy" diversity may be interesting to propose, as a matching similarity yields discrete values for determining the diversity of the set's categories, while a different metric may yield continuous values, being able to capture subtler similarities among the categories, which can be interpreted as degrees of the shared features for each item within them. In turn, a sharp ILS is more likely to cover the list's redundancy, while a fuzzy ILS score can better describe the list's disparity, which, contrary to a user-based novelty, it measures the list's diversity independently from the user profile.

The local coverage metric, as adapted from [58], takes the categories of the recommended items to evaluate how faithful they are to the user preferences, hence it operates at the local level of the user. A metric to evaluate the coverage level of the recommender system is also proposed, but for simplicity only the local one is considered for the experimental evaluation. Since coverage has been devised as a metric to cope with query ambiguity in IR, it is argued that a category-based local coverage metric is suited for the recommendation approach in terms of understanding how the system is able to interpret the user profile and generate recommendations that follow user preferences.

The global coverage, instead measures the amount of categories contained in the recommendation list while considering the whole set of users, to understand from how many categories the system is able to recommend items for.

The novelty metric as suggested above considers only the categories of elements, as suggested in the revised Stirling's definition. However, for simplicity, the novelty on the item-basis is not included in the formula. Therefore, it is argued that the high level of generalization offered by the item taxonomy could pose a limitation when the novelty should be evaluated on the level of single items. As a future work, the metric and the recommendation approach should consider item-based novelty , along with the categories in which items are assigned.

Following Stirling's revised definition, it has been argued that the redundancy, coverage and novelty are the corrispective of the balance, variety and disparity properties. Hence, by considering the three aspects individually, the general level of diversity can be evaluated, as an aggregated function of these metrics could potentially be misleading in the evaluation. Moreover, in order to avoid a "black box" diversity measure, it is proposed to adopt the categorical-ILS to measure the recommendation set diversity on itself in terms of non redundancy, while for the evaluation of diversity to the user profile, the, redundancy, coverage and novelty metrics as proposed are selected.

An important consideration on the evaluation metrics is that they are not able to distinguish the positions of the items in the recommendation list. Since the proposed approach currently focuses solely on recommendation filtering. As a future work, an evaluation of this issue is proposed, perhaps employing other diversity metrics which take into consideration the size and the rank of the items. To this end, it is argued that a ranking module is necessary, in order to consider the item positions, as arguably, they cannot be directly controlled by the filtering approach.

metric	Description	Advantages	Disadvantages
Category-based intra list similarity [15]	Aggregate dissimilarity between pairs of categories in a set.	Flexible and well-served to describe the diversity of a set. It can cover redundancy or disparity independently of the user profile and at different levels depending on sharp/fuzzy metrics.	Dependent on the definition of item similarity and similarity metric. No consideration of user coverage.
BinomDiv *[33]*	Combination of genre coverage and non-redundancy of a set	Aggregates coverage and redundancy in one metric. Performs better than ILD [32].	Only considers genre.
Category-based Coverage	Coverage of category-based user preferences.	At category level, it measures the ratio of preferred categories of the set. It can be adapted to measure local and global category coverage.	Considers categories as disjointed, may be improved to consider intra-category similarities to compare on subtler aspects.
Category-based Novelty	Unexpectedness value of item categories for the given user.	At category level, it measures the ratio of unexpected categories in the set, compared to the preferred ones.	Different assumptions are required to measure novelty at inter and intra-categories levels.
Redundancy	The redundancy is based on category based-ILS, with match-based similarity.	Calculates a simple aggregate of the occurrences of categories covered by the user and those present in the list.	Same as category-based coverage.

Table 4.2. Comparison of the diversification metrics

5 Design of the recommender model based on the proposed diversification framework

In this chapter the proposed approach is explained in detail, following the analysis of the proposed framework on the diversification approaches. This chapter covers the last contribution of this work, namely the algorithmic recommendation procedure with embedded diversity which has been devised after the analysis of the shortcomings of related diversification methods. For this work, the proposed approach is tailored to the application domain already introduced, movie recommendation. The generality of applicability is not considered, as other types of dataset are required. As the diversification framework propose to utilize categories of items, the design of the approach is divided into 3 phases, which cover the required transformations of the user and item profiles to meet the hypotheses of the framework, from feature extraction to the actual profile models and the full recommendation procedure. To understand this approach, the required machine learning tools and methodologies are explained in section 3.3.

5.1 Overview of the proposed diversification approach

The proposed approach is introduced and defined as a combination of recommendation filtering and diversification; the motivations which brought to its construction follow the proposed framework (section 4.2) based on Stirling's [23] definition, which describes diversity on the basis of a taxonomy of elements (in this case, the elements consist of movies) as well as the analysis of the limitations related to other diversification methods (section 4.4).
The proposed diversification technique is tightly coupled to the recommendation filtering, therefore can be formulated as a diversity modeling approach, in contrast to post-filtering.
A preliminary understanding of the rationale behind the diversity modeling approach is shown in Figure 5.1.

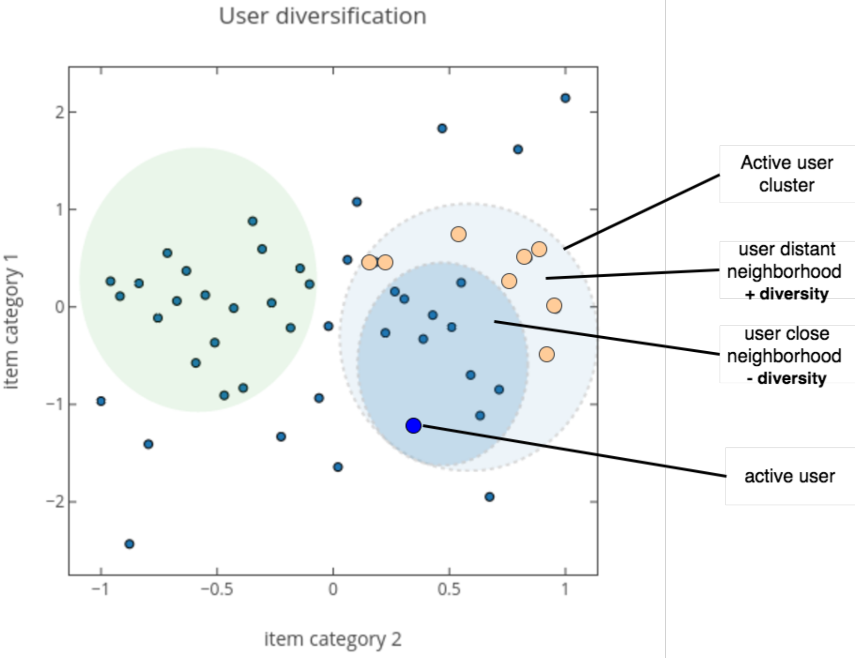


Figure 5.1. Proposed diversification approach: filtering distant neighbors (orange points) in the same group of users (blue cluster).

In detail, the approach suggests to utilize the categories of items as a mean to model the user preferences and then, compute a list of recommendations by filtering distant neighbors (in orange) for the active user (in saturated blue). Moreover, in order to maintain a level of accuracy in the neighbor filtering, it is proposed to group users together: from the users within the same cluster (in blue), a number of distant ones is retrieved, while the nearest neighbors are filtered out. Subsequently, the list of recommendations is generated from the preferred categories of the distant neighbors.

It is argued that the assumed existence of the filter bubble can be mitigated by receiving recommendations from users who do not mirror the active user's preferences. In fact, in collaborative filtering, the normal heuristic to provide recommendations is to extract the similarities among users and recommend from the most similar users, found through classification techniques such as k-Nearest Neighbor. While by searching for the most similar users may be beneficial for the accuracy of the system, it is argued that providing recommendations from distant users may increase the likelihood that the items are diverse and therefore, the chances for serendipitous encounters with novel items. Moreover, the conventional interpretation of the filter bubble is in regards of the polarization of interests –in this case, the user preferences– based on an overly personalizing system. Diversifying users thus would help to decrease the preference polarization by allowing more opportunities from the experience of different types of items.

As a first idea, the approach would retrieve the distant neighbors from the same cluster, since it is expected that the baseline within-group similarity can be a threshold for considering common preferences and therefore, the accuracy VS diversity trade-off, so that the users may not be completely dissimilar. On the other hand, it may be interesting to see the effects of the proposed approach when the neighbors are retrieved from different groups of users. Nevertheless, only the first idea is discussed in this work.

As the diversity framework has been explained, it is important to recall that diversity is applied to the active user profile; hence, the proposed approach takes into consideration five dimensions on which trade-offs between the user preferences and the other users' can appear, namely:

- *Neighbor distance*: The accuracy VS diversity tradeoff is controlled by the pairwise diversity between the active user and the other users in the same group. As such, the retrieved users should follow a parameter which can be tuned externally. As the distance between neighbors is the same that has been applied for the clustering analysis, a less naïve approach to the neighborhood formation taking into consideration each other user's contributions (e.g., which user contributes the most in terms of novel categories for the active user) could be proposed as a future development of this work.

- *Variety of categories (global coverage)*: the number of categories that should appear in the recommendation list (without considering their disparity or the novelty to the user profile) depends on the length of the desired recommendation list and should be controlled to ensure that not too many or too few categories will appear. Intuitively, by controlling the variety of categories, the user will have the opportunity to navigate a list of recommendations with items belonging to many or few categories, hence, it can also be understood as the property of *global coverage*. Also, the variety implicitly controls the category disparity, as it seems reasonable that with low variety of categories comes a low disparity and vice versa.

- *Disparity of categories (novelty and local coverage)*: the heterogeneity of the categories is the result of the novelty VS local coverage trade-off between the favourite categories for the active user and the distant neighbors'; therefore, it considers how many of the novels and how many of the covered categories should be used to select the items.

- *Variety of items*: similarly to the selection of the number of categories for the recommendation list, the variety of items considers how many items to show in the list. Intuitively, the variety of items is proportional to the variety of categories, as the number of categories increases, also the number of items should increase. The variety of items, in terms of list size, has been acknowledged to influence the coverage and redundancy of the recommended set [33]. In fact, it seems reasonable to generate recommendations from a number of categories proportional to the desired list length (i.e., for short lists, few categories and viceversa); as such it is proposed to control the variety of categories accordingly.

- *Item recommendation list balance (redundancy)*: the last trade-off is controlled by the number of items belonging to the same category that should appear in the final list of recommendations. Intuitively, this trade-off is between balance and redundancy to control the visibility of each category; the more the redundancy, the more items belonging to the same category and vice versa.

With this method, the goal is to construct a set of recommendations that can be diverse and at the same time qualitatively relevant for the active user. As a matter of fact, it is argued that the application of this approach to clusters of users allows to control the diversity VS relevance trade-off, by filtering distant or close neighbors. Intuitively, the designed technique takes inspiration from [35], whose idea of providing recommendations on the basis of distant (as diverse) neighboring users has proven to be unconventional as well as promising, given the results of the study. However, the aspects on which the proposed approach differs from [35] are as follows:

- In [35], the neighborhood is produced by maximizing the significance between the active user and the users in the neighborhood and subsequently selected. The proposed approach instead includes an additional step, the formation of groups of users, which considers a baseline similarity among the users within the same group.
- In [35], the significance between two users is calculated so take into consideration the accuracy-diversity trade-off, controlled by a tunable parameter α. The proposed approach naively extracts the users within the same group and, using a similar method, it controls the similarity VS diversity trade-off.
- The technique illustrated in [35] utilizes solely the rating dataset, whereas here, both ratings and metadata are exploited.
- In [35] the objective is to predict the ratings for unseen items for the active user while here the ratings are the starting point to categorize user preferences. Nevertheless, there can be an additional step in the proposed approach to predict the relevance of a certain category for the active user, given the similarity between the user and the distant users and the preferences for their categories.
- However, the major difference lies in the categorization of user preferences (apart from the creation of groups of users), not included in [35].

5.2 Recommendation procedure modularity

Nevertheless, for the proposed approach to work, a further preprocessing step is required to transform item and user profiles in order to meet the concept of taxonomy and can be considered as part of the technique.
The overall recommendation process can be modularized as shown in Figure 5.2 , in which the modules are distributed according to three categories of tasks: feature extraction, preprocessing and recommendation.

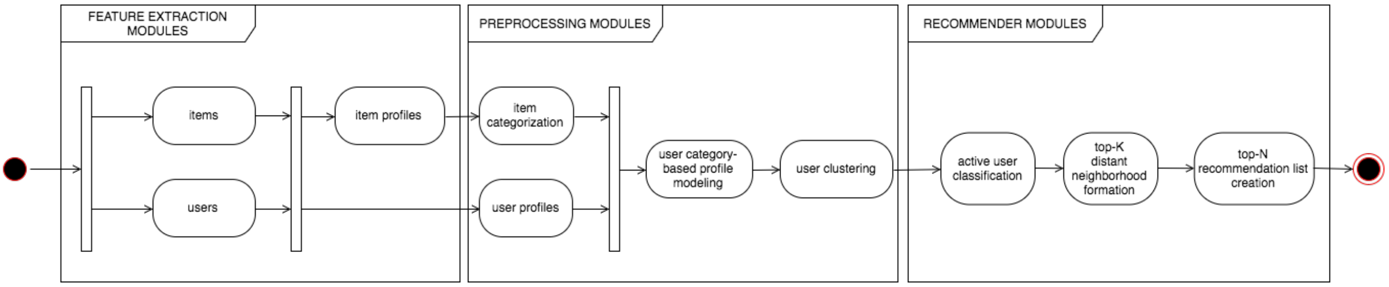

FEATURE EXTRACTION MODULES
items
users
item profiles
PREPROCESSING MODULES
item categorization
user profiles
user category-based profile modeling
user clustering
RECOMMENDER MODULES
active user classification
top-K distant neighborhood formation
top-N recommendation list creation

Figure 5.2. Diagram of the feature extraction, preprocessing and recommendation procedure

Figure 5.1 covers only the explanation of the filtering modules, whose processes define the last step of the recommendation procedure. The next sections focus on the modules according to their task.

5.3 Feature engineering of user and item profiles: data preparation

After having understood the machine learning methods and tools, the following section illustrates the general methodology applied to the data, defining the logical flow to transform the raw data into appropriate features that better represent users, items and user preferences to be used by the proposed technique. Given the Movielens dataset, the process includes the extraction of metadata associated with each rated movie with the subsequent modeling of the user and item features.

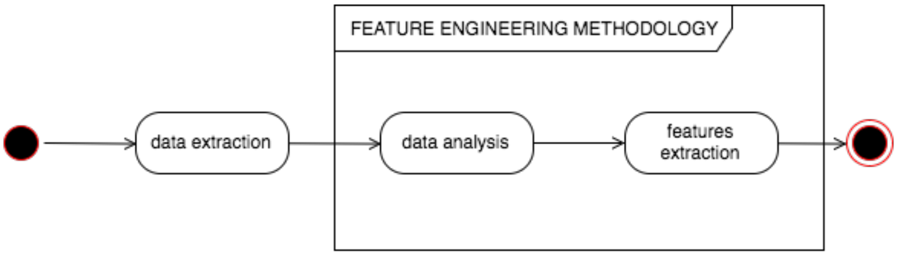


Figure 5.3. General feature engineering pipeline.

The feature engineering step (Figure 5.3) entails a thorough analysis of both datasets as the parallel steps following the extraction of raw data (a default step), to understand the best course of action for extracting relevant features, since a first glimpse to the data has determined the requirement of a preprocessing step.
Arguably, the dataset analysis represents the most important process as it offers the essential knowledge of the application domain.
The second step focuses on the additional transformation of available features in order to extract a set of meaningful features for users and items as previously defined.
Since the process of extracting meaningful features may vary depending on the dataset utilized, this process is explained more in detail in chapter 0.
The next section illustrates in detail the preprocessing modules: from the transformation of user and item profiles to the creation of an item taxonomy, as well as the analysis of the clustering results.

5.4 Preprocessing modules design

In this section, the preprocessing modules delegated to the transformation of the user and item profiles to structures suitable for the application of the recommendation procedure are described, as the current profiles cannot be utilized as is. Following Figure 5.2, the modules of interest are: item categorization, user preliminary profile modeling, user category-based profile modeling and user groups formation.
The preprocessing modules inner workings (item and user profiles modeling) focused on items and users are more visible in the following diagrams.

5.4.1 Item categories construction

The first step in the recommendation preprocessing methodology applies the vision of the diversity definition to compose a taxonomy of item categories (Figure 5.4).

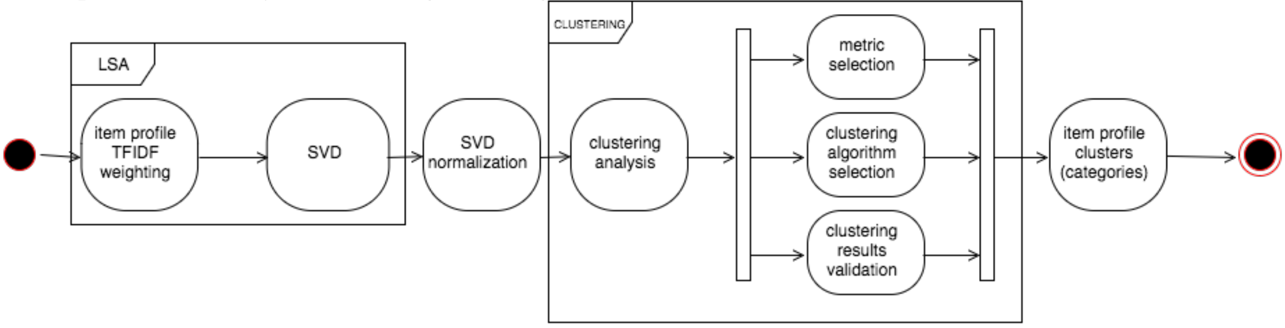

LSA
item profile TFIDF weighting
SVD
SVD normalization
CLUSTERING
clustering analysis
metric selection
clustering algorithm selection
clustering results validation
item profile clusters (categories)

Figure 5.4. Item categories construction.

In particular, the LSA methodology is applied to the item profiles to find the latent similarities based on the original extracted features: the feature occurrence matrix is computed, then TF-IDF transformed to find a potentially unbiased subset of features and calculate their relevance scores for each item. In fact, TF-IDF is argued to unbias the dataset by decreasing or annulling the relevance for features shared by many items, thus less discriminant, while maintaining the defining features for single items.
Secondly, the item profiles dimensionality is inspected and SVD is applied to reduce the dataset dimensionality in order to extract latent features upon which items are compared.
Prior to the clustering analysis, an additional step is the normalization of the SVD rank reduced matrix, as it allows the application of the euclidean distance which is comparable to the cosine distance for normalized vectors.

Next, clustering analysis is performed on the factorized item profiles, according to the criteria of similarity/distance metrics, clustering algorithm and number of desired clusters; specifically, the number of clusters is subordinated to a qualitative and quantitative analysis of the intra-cluster density and inter-cluster distance, in order to limit the creation of redundant clusters (i.e., having similar items in separate clusters), which can also be highly specialized and a potential cause of overfitting.
As a result, the item categories are created and each item is labeled accordingly. As this module requires new information to grow, it is suggested to provide periodic updates of the clusters so that new movies can be classified according to new terms and keywords.

5.4.2 Preliminary user profile modeling

Parallelly to the item categories construction, users are modeled into preliminary profiles which encode each individual positive preferences for the rated items by applying an extracted rating threshold (for the Movielens dataset, this step is explained in section 6.1.3). This module assumes that the ratings are explicit, however the same reasoning can be adopted for implicit ratings, such as retention time or other measures of implicit preferences.

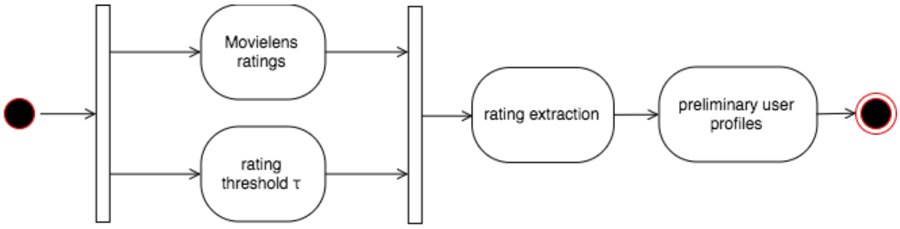


Figure 5.5. Modeling of the preliminary user profiles.

Figure 5.5 illustrates the process in detail. In order to extract the individual threshold, each user rating is standardized to z-scores and the threshold, set to the centered average rating, is compared to the centered mean and the mean prior to standardization, so to extract both positive preferences for enthusiastic users and other users (see section 6.1.3 for the complete analysis on the ratings).

5.4.3 Category-based user preference modeling

After having modeled the users into preliminary profiles and the taxonomy of items is prepared, the next module in the preprocessing procedure involves the user profiles modeling according to the extracted ratings and the item categories.

Figure 5.6 illustrates the process, which uses the output of the previous preprocessing modules as input to model user preferences and extracts the category-based user profiles.

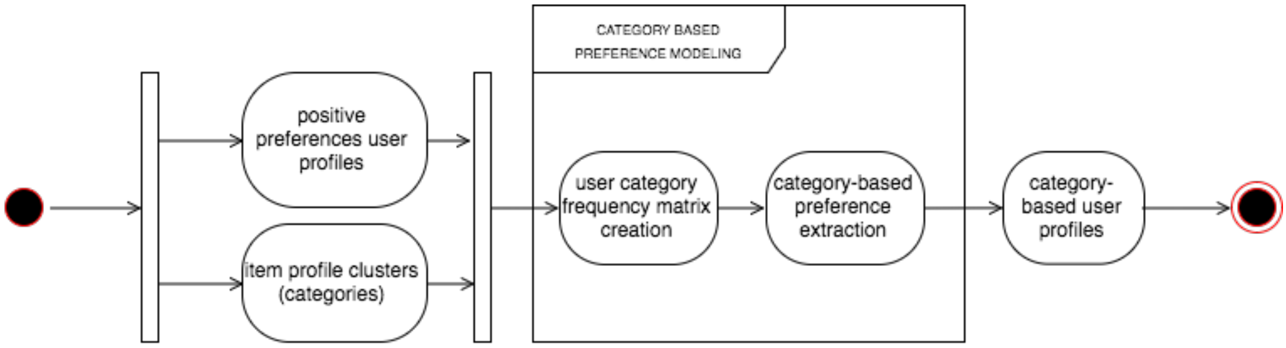

CATEGORY BASED
PREFERENCE MODELING
positive preferences user profiles
item profile clusters (categories)
user category frequency matrix creation
category-based preference extraction
category-based user profiles

Figure 5.6. Modeling the category-based user profiles.

Two matrices are used for this purpose: R, containing the positive ratings of users u on items i, and the cluster matrix C, a boolean item i to cluster c association matrix.

The profile of user u is constructed with the matrix $P_{u,c}$, where c is a cluster (or category) of items, and each element of the preference matrix is defined as follows:

1. from the preliminary user profiles, the categories to which positively rated items are apportioned are extracted and the profiles are initially encoded with each category raw frequency (as the numerator of (5.1).
2. Then each user profile is divided by the number of categories experienced. This step is not required, but is argued that by transforming the raw category frequencies into their proportion to the user profile, the differences towards users having experienced more categories and users having more focused interests may be more comparable using profile proportions, rather than raw frequencies and average ratings.

$$P_{u,c} = \frac{\sum_{i \in I_{u,c}} C_{i,C}}{\sum_{c \forall C_u} \sum_{i \in I_{u,c}} C_{i,C}} \qquad (5.1)$$

Where:

- R_u is the set of ratings for user u.
- $I_{u_\tau} = \{i, \forall r_{u,i} \in R_u | r_{u,i} \geq \tau\}$ is the set of items rated positively by u.
- $I_{u,c} = \{i, \forall i \in I_{u_\tau}, c \in C | C_{i,C} \neq 0\}$ is the set of items rated by u, belonging to category c. (use for average rating and for category frequency)
- $C_u = \{c, \forall c \in C, \forall i \in I_{u_\tau} | C_{i,C} \neq 0\}$ is the set of categories for which u has experience.
- $C_{i,c}$ denotes the occurrence of cluster c for each item i in R_u rated above threshold τ and
- the denominator consists of a normalization parameter to bring the preference scores within [0,1].

5.4.4 User groups formation

The last process included in the preprocessing procedure entails the formation of groups of users.
Intuitively, it is hypothesized that the users can form groups based on their similar preferences, otherwise collaborative filtering would not be possible. Moreover, by clustering the users, a basic similarity to the active user can be guaranteed, which defines also an implicit measure of recommendation relevance. Thus, by receiving the output of the category-based user profile modeling process, the steps are illustrated in Figure 5.7.

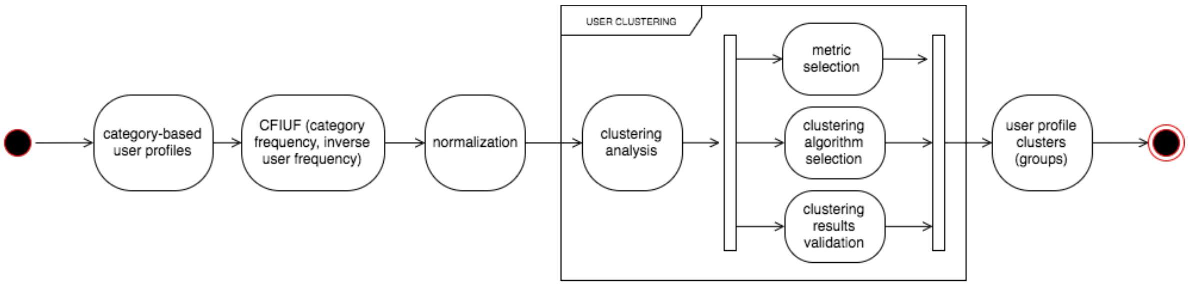

USER CLUSTERING
category-based user profiles
CFIUF (category frequency, inverse user frequency)
normalization
clustering analysis
metric selection
clustering algorithm selection
clustering results validation
user profile clusters (groups)

Figure 5.7. User clustering process.

This process, similarly to the item categorization one applies TFIDF to the category-based profiles, thus can be renamed CFIUF (category frequency, inverse user frequency), aiming to model the similarities of users by weighting the category scores for each user. Moreover, the same reasoning regarding the choice of clustering algorithm and distance/similarity metric can be applied here and it is argued that the normalization of the CFIUF-weighted user profiles allow the clustering to be performed with euclidean metrics without incurring in the skewness issue to which k-means is affected.
Subsequently, the normalized CFIUF-weighted user profiles are used for the clustering analysis, whose methodology is identical to the item categorization process.
As a result, the groups of users are created and the recommendation procedure can be explained in detail.

Specifically, categories of movies, groups of users and the item and user profiles as derived by the previous processes are the starting point in the design of the proposed approach, for which the next session proposes the general process of recommendation and embedded diversification.

5.5 Recommendation & diversification modules design

The methodology of the proposed recommendation and diversification approach is divided in four major steps: the user profile construction, the user clustering step and the formation of the user neighborhood with successive extraction of a set of recommendations.
In particular, the process entails several sub-operations (Figure 5.8), which require the presence of the active user:

1. Transformation of the active user to her category-based profile.
2. Classification of the active user to a group of similar users.
3. Formation of the distant neighborhood.
4. Creation of the list of recommended items.

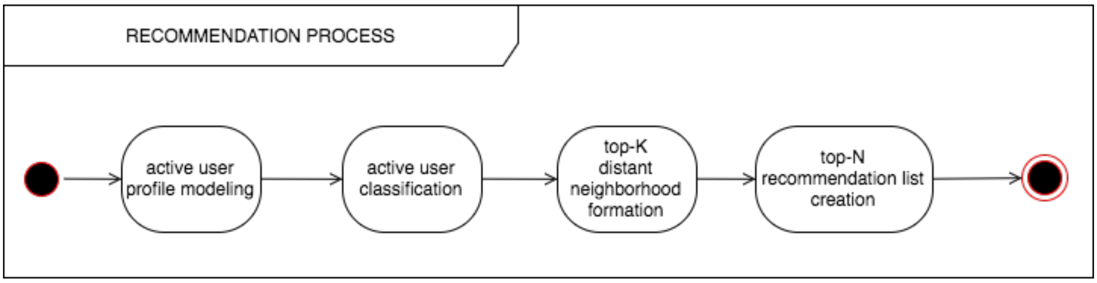

RECOMMENDATION PROCESS
active user profile modeling
active user classification
top-K distant neighborhood formation
top-N recommendation list creation

Figure 5.8. Recommendation procedure in detail.

In the following sections, each of the illustrated modules composing the recommendation procedure is discussed in detail.

5.5.1 Active user profile modeling

This module is required as the active user profile is expected to contain the raw ratings for each consumed items, in this case in the original Movielens format. Here, the role of the preprocessing modules described in the previous section is to transform the active user profile into the category-based one, following the process described in Figure 5.9.

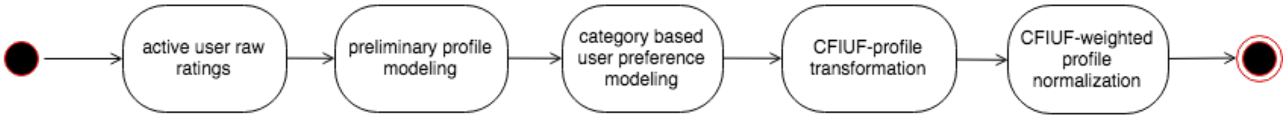

active user raw ratings
preliminary profile modeling
category based user preference modeling
CFIUF-profile transformation
CFIUF-weighted profile normalization

Figure 5.9. Transformation of the active user profile into CFIUF-weighted format.

This process is equivalent to the transformation of the user profiles to category-based and subsequently CFIUF-weighted profiles, prior to the formation of user groups.

5.5.2 Active user classification

Once the active user has been transformed into the CFIUF-weighted profile, the classification to a group of users is achieved through the selection of the nearest cluster, which can be computed using k-nearest neighbor on the clusters.

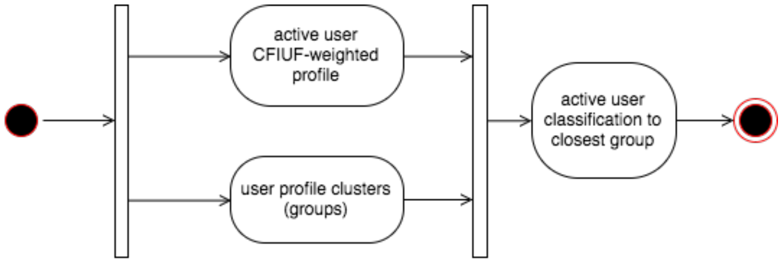

active user CFIUF-weighted profile
user profile clusters (groups)
active user classification to closest group

Figure 5.10. Active user classification.

Therefore, the classification process simply involves the labeling of the active user according to the most similar group of users (Figure 5.10).

5.5.3 Distant neighborhood formation: accuracy VS diversity

This is the pivotal step of the recommendation procedure, as it involves the control of the diversity between the users within the same cluster and arguably, the control of the recommendation list diversity.

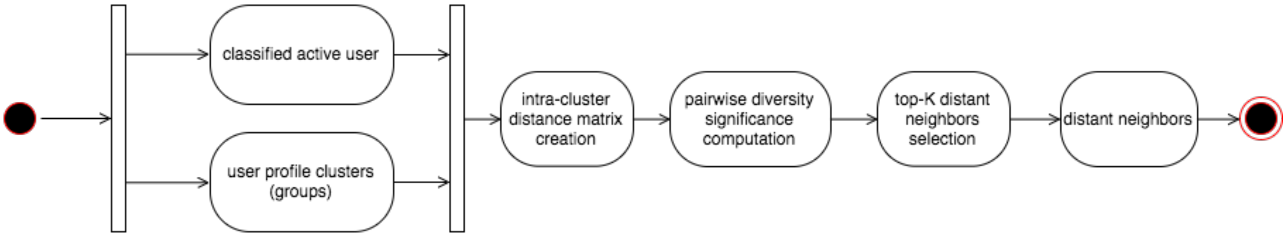

classified active user
user profile clusters (groups)
intra-cluster distance matrix creation
pairwise diversity significance computation
top-K distant neighbors selection
distant neighbors

Figure 5.11. Distant neighborhood filtering flow.

In this step (Figure 5.11), the k most significantly diverse users within the same cluster are filtered, as opposed to nearest neighbors, according to the pairwise diversity significance between users. As such, it is expected that the categories of items from distant users will appear different but not radically, from the active user (since the prerequisite of the recommendations is to be still accurate to the active user preferences). Nevertheless, it may be interesting also to select users from different clusters, as they may provide further discovery opportunities for the active users, since it is expected that users from different clusters will have preferences for categories that do not overlap (or at least that are more different) with the active user's.

Essentially, in order to control the diversification of the distant neighborhood formation, an external parameter α is introduced to determine the pairwise diversity significance, similarly to [35]. for a given user u and another user v belonging to the same cluster U, the diversity significance $s(u,v)$ is calculated as follows:

$$s(u,v) = (1-\alpha)\cdot(1-d(u,v)) + \alpha d(u,v) \tag{5.2}$$

Where:

- The diversity is proportional to the growth of α; hence, the larger α is, the more the diversity.
- $d(u,v)$ is the distance between the active user and v.
- $1-\alpha$ and α respectively measure the similarity and the distance trade-off between the two users.

Hence, the k most significant users are extracted so that the significance between u and v is maximized as:

$$V = argmax_{v\in V}\big(s(u,v)\big) \tag{5.3}$$

Where the resulting set V of distant neighbors is the result of the maximized diversification significance.

5.5.4 Top-N recommendation list generation

The final step in the recommendation procedure is to select the categories and the items from the distant neighbors (Figure 5.12) and the item categories.

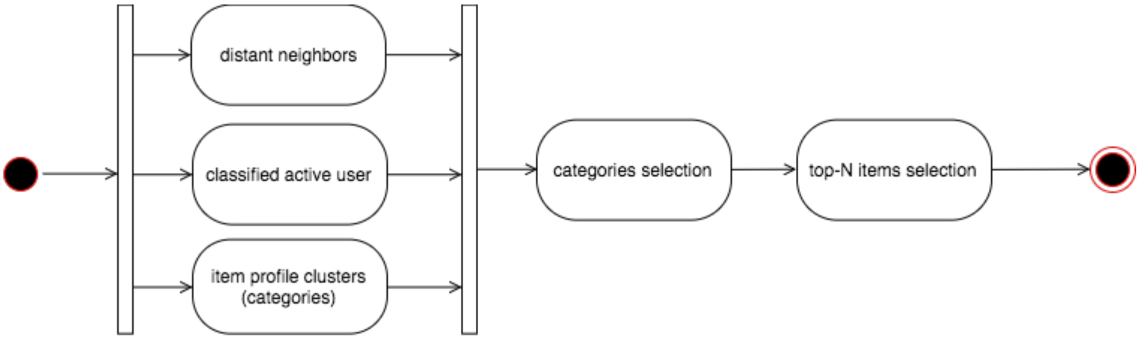


Figure 5.12. Recommendation list generation process.

As such, there are two additional processes in order to extract the categories of the distant neighbors and exploiting them to create a list of items, which are explained in the next two subsections.

5.5.4.1 Categories selection: variety and disparity and size awareness

In this step, the control on variety, disparity and size awareness is defined to produce a number of categories proportional to the number of recommendations.

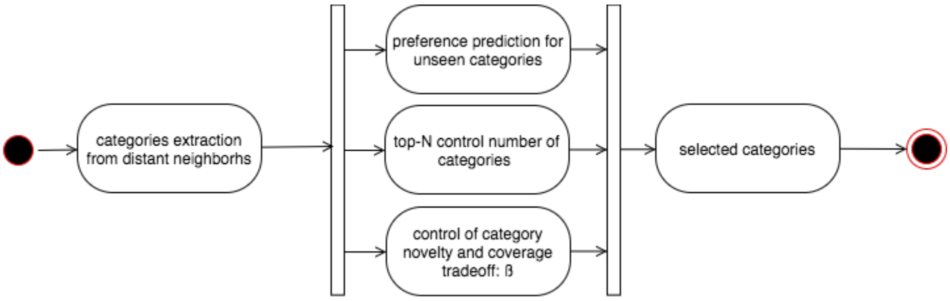

categories extraction from distant neighborhs
preference prediction for unseen categories
top-N control number of categories
control of category novelty and coverage tradeoff: ß
selected categories

Figure 5.13. Categories selection process.

The categories selection step (Figure 5.13) receives the necessary data to predict the relative category frequency from the original category-based user profile set P. Specifically, for a given user u and a distant neighbor $v, v \in V$, with V as the set of selected distant neighbors, it is proposed to predict the preference score for a category c not experienced by u with the average preference augmented with the diversity significance:

$$\hat{p}_{u,c} = \frac{\sum_{v \in V} p_{v,c} \cdot s(u,v)}{|V|} \tag{5.4}$$

Where:

- $C_V = \bigcup_{v \in V} C_v$ is the union of all categories for which the distant neighbors have a preference.
- $p_{v,c}$ is the preference score for a distant user v and a category c.
- $s(u,v)$ is the diversity significance between the active user and v.

In particular, the preference score for unseen categories is proportional to the preference for the distant neighbor v and the similarity between the active user and v.

Following this step, the unseen categories are ranked according to the descending predicted preference.
Parallelly, the size of the recommendation list is chosen as $topN$ and, recalling that the number of categories should be proportional to the recommendation set size, it is proposed to control the number of categories, defined as $topC$ as follows:

$$topC = \lfloor \gamma \cdot topN \rfloor \tag{5.5}$$

Where $\gamma \in]0,1]$ is an external parameter controlling the proportion relative to the recommendation list, so that the relative number of categories is actually $1/\gamma$ of the number of items in the list. For example, when $\gamma = 1$, the recommendation list's categories are one for each item, whereas if $\gamma = 1/3$, the number of categories is $\lfloor topN/3 \rfloor$ and if $\gamma = 1/topN$, only one category dominates the list. Therefore, as γ grows, so does the relative number of categories.

After the unseen categories have been ranked, the selection of categories from the set of distant users is devised according to the number of desired categories to show in the recommendation lists. In principle, the control of category variety is operated with the $topC$ categories.
The final step is to select the categories as to overlap with the user profile so that the preferred categories of the active user are preserved by selecting items from them, therefore, coverage. Nevertheless, in the case of overlap, to ensure that the recommendation list is diverse with respect to the user profile, it would be more intuitive to select the novel categories for which the active user has no relative frequency of consumption, therefore, novelty.
Hence, by extracting two initial $topC$ lists of candidate categories, divided in novel and covered categories, the proportion between coverage and novelty brought by the categories to the final $topC$ list of categories is controlled by the another external parameter β, given as:

$$\bar{C}_{novel} = \beta(|topC_{novel}|)$$
$$\bar{C}_{cover} = (1-\beta)\cdot(|topC_{cover}|) \tag{5.6}$$

Where:

- $C_{novel} = C_V \setminus C_u$ is the set of categories for which the active user has no experience.
- $topC_{novel}$ are the $topC$ categories ranked by predicted preference score belonging to C_{novel}.
- $topC_{cover}$ are the $topC$ categories ranked by preference score belonging to C_u.
- as β grows, the proportion of novel categories increases accordingly and the proportion of covered categories decreases.

Hence, it may be more reasonable to define the proposed approach as a dual step diversification modeling, applied first to retrieve distant neighbors and then, to control the proportion of covered VS unseen categories for the creation of the recommendation list.
The final set of selected categories is simply given by the formula as follows:

$$topC_u = \bar{C}_{novel} \cup \bar{C}_{cover} \tag{5.7}$$

5.5.4.2 Top-N recommendation list generation

Lastly, from the selected categories, the $topN$ items unseen by the active user u are retrieved to generate the recommendations. For this purpose, the $topC_u$ categories, with relative preference scores are used to balance the category redundancy. In particular, the balance follows the rank of the preference scores so that highly ranked categories have more redundancy and vice versa. Therefore, the rationale is simply to recommend more items from highly ranked categories. However, this approach may have implications on the user satisfaction, whose evaluation with real users is left as a future development of this work.

To generate the recommendation set, the following methodology is proposed.
Firstly, the empty list is partitioned according to the weight of a given category. In fact, the category score has to be transformed to the frequency relative to the recommendations size. Recalling that the preference score for given user and category is $p_{u,c} \in [0,1]$, the proportion of the category in the recommendation list is:

$$freq_c = \frac{p_c}{\sum_c p_c} \tag{5.8}$$

Where:

- $c \in topC_u$ and p_c is the preference score (either associated to or predicted for the active user).

Once the basic relative frequency has been computed, the next step is to define the number of elements from the same category that will appear in the recommendation list, transforming the frequency (as the partitioned item set), to its discrete counterpart:

$$\eta_c = \lfloor freqc \cdot topN \rfloor \tag{5.9}$$

Therefore, η_c becomes the actual number of items from the same category c, derived from the partioned set frequency. The resulting total number of items is given as $topN = \sum_{c \in topC_u} \eta_c$.

After deciding how many items should appear from the selected categories, the final part of the recommendation procedure is to select which items from those categories. A naïve approach would be to suggest the most popular η_c items from each selected category c without considering the items in u's history so that the user will have a general idea of what to watch. Another approach could entail the selection of items from the distant neighbors' history. Moreover, the ranking of the items can include a mixture of global rating, popularity and freshness, to ensure that they are not completely unknown or too niche. Nevertheless, the decision of which items to recommend is still an open ended question; as such, a proper answer to this step is left to the future development of this work.

The following chapters are therefore focused on the implementation of the proposed approach, focusing on the modules up to the distant neighborhood formation. As such, an initial evaluation, proposed in this work, is focused in chapter 1 on understanding the diversity brought by the selection of distant neighbors as previously defined.

6 Feature engineering: data preparation for the recommendation model

In this chapter, the machine learning methodology proposed in section 5.3 to analyse and extract the data for the intended application domain is followed. Beginning with the analysis of the users, the Movielens dataset is utilized. As the output of the rating analyis, the threshold is specified for each user. The chapter continues with the exploratory analysis of the movies contained in Movielens on the metadata extracted from IMDb, to prepare the initial item profiles. As a result of this process, the initial user and item profiles are modeled and ready to be preprocessed in the second phase according to the full recommendation procedure.

6.1 Movielens user data analysis

The ratings for movies are gathered through the Movielens dataset [20], maintained by the GroupLens Research, which contains 100k ratings from 671 users for around 9000 movies, between 1995 and 2016. From the documentation, the 100k dataset results small, all-purpose and useful for education purposes and not for research purposes, due to the relatively short-lived existence (as it is updated every year). Nevertheless, it has been adopted for research purposes given its very small dimensions to test the diversification approach and not to test scalability or accuracy issues.
In particular, the exploratory data analysis is performed to answer the following questions:

- How are ratings distributed across users? to understand the effect of power users.
- How many movies are rated in common? To understand the sparsity level of the dataset.
- How differently rated are the movies? How many users have low/high mean low/high variance in ratings? To understand the individual scale of ratings adopted by each user and to identify rating biases.
- Which are the most rated movies? To identify popularity biases.

The methodology proposed to analyse the Movielens dataset is illustrated in Figure 6.1.

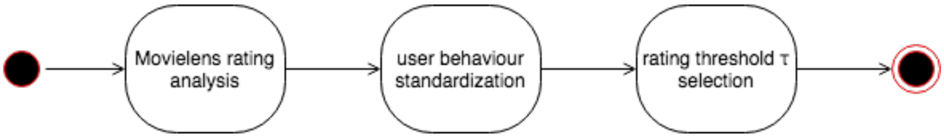

Movielens rating analysis
user behaviour standardization
rating threshold τ selection

Figure 6.1. Rating analysis methodology.

The first step is entails the analysis of ratings and rating behaviour, followed by the feature extraction step, defined as the standardization of each user on the basis of her rating behaviour. Lastly, a rating threshold is defined after the standardized users, for the later preliminary user profile modeling.

6.1.1 Rating distribution analysis

Figure 6.2 illustrates the sparsity of ratings for the first 100 users (columns) and each movie (rows). Ratings has been color-coded to understand user patterns and the general rating distribution.

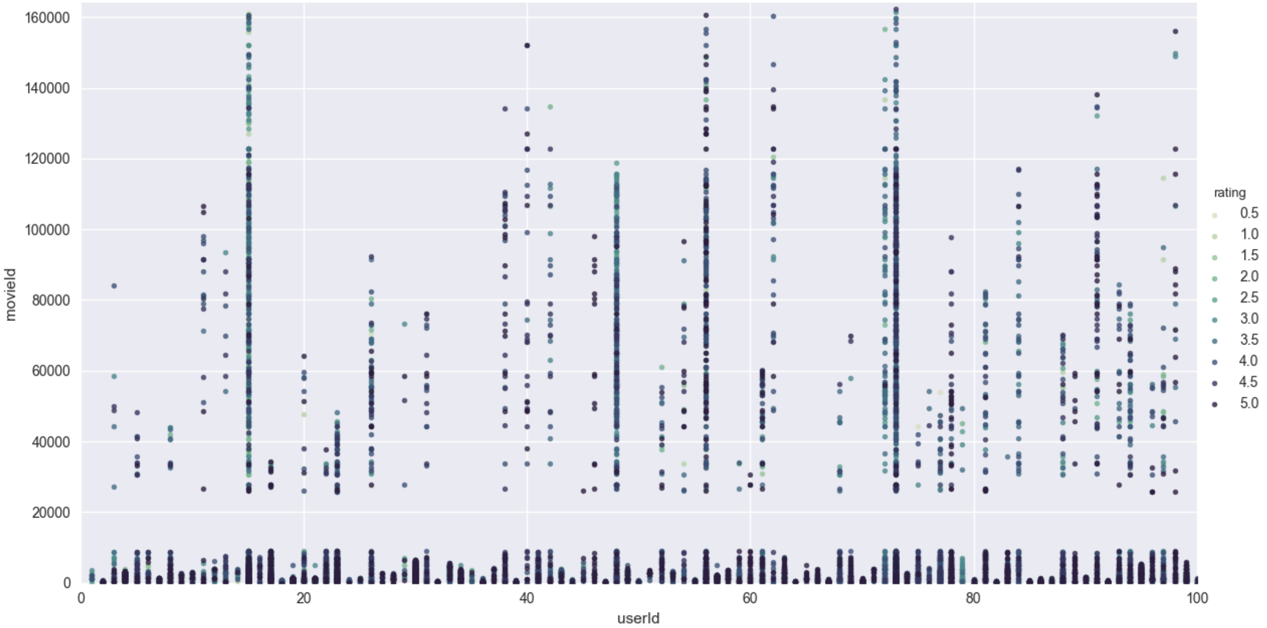


Figure 6.2. Rating sparsity of the dataset for the first 100 users (as columns) and all movies (as rows).

The abnormal gap between the ratings is due to the change of movie identifiers found in Movielens, presumably because of the periodic updates to the dataset, which yet conserve the ratings for the first movies that have been added by Movielens.

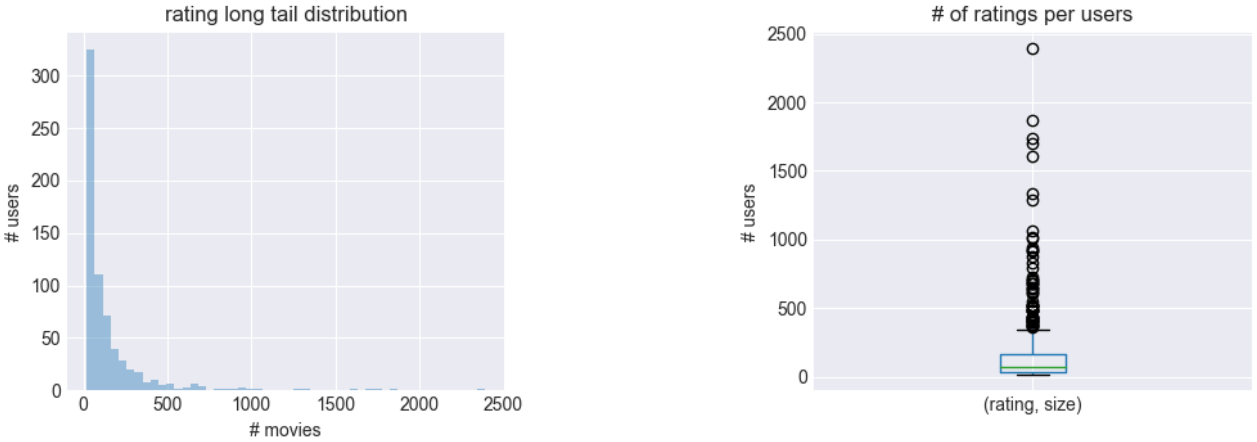


Figure 6.3. Rating long tail distribution.

Users have rated the movies on a clear long-tail distribution, which is better depicted in Figure 6.3.

What is interesting to notice is the amount of ratings per users, which shows that the majority have rated less than 1000 movies, while very few have rated above 1000 movies and up to ~2300 movies (in fact, only 10 users have rated more than 1000 movies). An inspection of the timestamps at which ratings were recorded has shown that users with a large amount of ratings were presumably filtering agents, namely *filterbots*, which have been introduced by GroupLens in order to mitigate the cold start problem for new users by providing "power" users with many ratings modeled after other users' behaviours [60]. Nevertheless, the effect of filterbots on the recommendation module may provide even more insights for diversification purposes, thus these surrogate users have not been removed.

Moreover, the rating sparsity could pose a problem for the recommendation algorithm, therefore it has been proposed to inspect movie metadata, instead of ratings, to model the user preferences. It is expected that a content-boosted collaborative filtering approach to the recommendation engine and diversification method allows to find similarities among users on a more fine-grained level, compared to simple rating similarity. The next section presents also how ratings can pose a problem due to individual rating biases and proposes a way to preprocess raw ratings, to select only the positive ones for each user, leaving out "below individual average" for the user preference modeling purpose.

6.1.2 User rating behaviour analysis

The individual rating patterns are depicted in Figure 6.4, where each point represents the mean rating for a specific user. Generally, individual averages tend to be more compact near 3.5 and 4 (in rating scale)

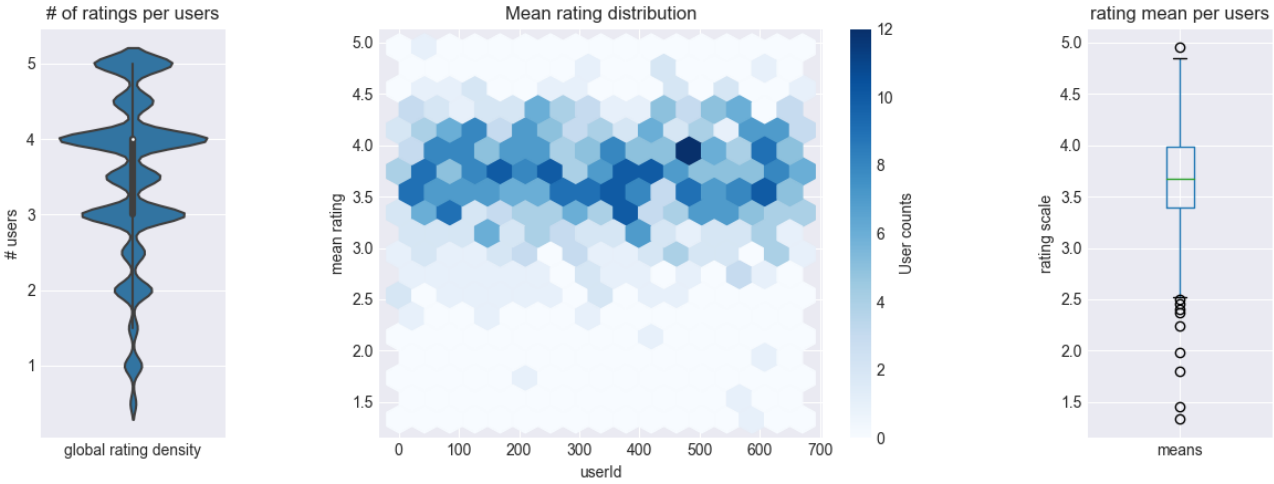


Figure 6.4. Ratings distribution and differences in individual mean ratings.

The box and violin plots (Figure 6.4, each side) respectively shows the global rating density and how the individual user's average ratings vary in a more compact way: from these figures, users have in general a tendency to rate movies well (with average mean around 3,5) and with a relative low variance (~1 on average), suggesting that many users may have similar rating patterns. However, in the case of rating behaviour, the rating means need to be considered individually, since there is evidence of different rating scales for each users. For example, while most of the users tend to rate movies around the average mean, other users have rating means above or below the average. Having understood this, an analysis of how many users have different rating patterns may shed some light on how a movie is considered "good" on an individual basis.

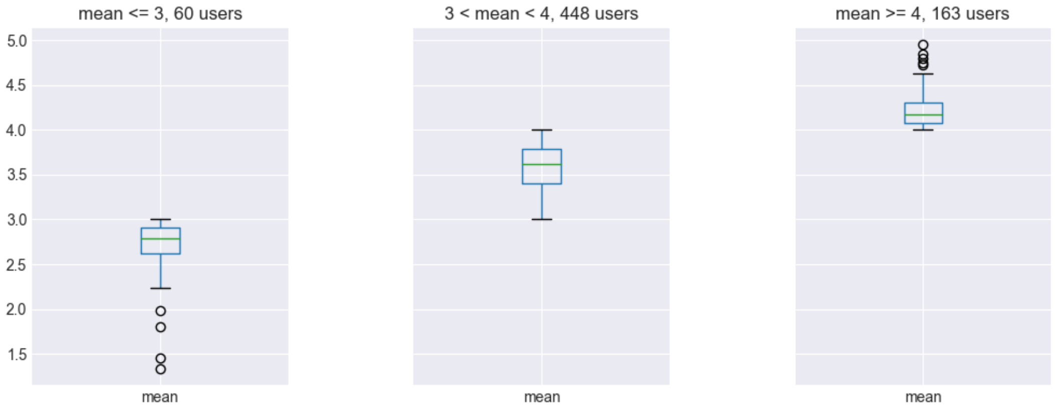


Figure 6.5. Rating patterns (means) divided by user mean rating.

Figure 6.5 shows the rating patterns along with the number of users following them, which have been splitted to consider only the individual rating mean below 3, within 3 and 4 and above 4. These values have been chosen for the following reasons: users who rate below 3 on average may have high rating standards, on the other hand, users only with ratings around 4 may focus only on rating good movies and ignore bad ones, while balanced users may enjoy a high number of movies and rate consistently.
The same discussion holds when analysing the standard deviation (depicted as the error in the box plots) for the same individual means: users with means below 3 tend to have higher standard deviations, which is understandable due to their probably high standards; balanced users (middle box plot), the expected majority, have also balanced rating standard deviations (around 1) even if there are some outliers; the left-hand side shows how "low-standard" users also tend to rate closer to their respective mean than the rest (with a standard deviation below 1).
To support the existence of varying rating scales, Schafer et al. argue that *"one optimistic happy user may consistently rate things 4 of 5 stars that a pessimistic sad user rates 3 of 5 stars. They mean the same thing ("one of my favorite moves"), but use the numbers differently"* [61], reinforced also by an analysis performed on the effect of the rating scale granularity (i.e. how many ratings are allowed) [62].

The analysis on individual rating means and standard deviations uncovered interesting aspects on user rating behaviours which have also been acknowledged by the research community [63], in particular the existence of subjective scales at which users adhere when rating movies: i.e. as movie preferences are perceived individually, not impersonally, there is a need to standardize all users so that individual biases can be removed and rating scales can be compared objectively. The purpose of user standardization is required also for the sake of modeling user profiles, in order to extract only the positive instances of movie ratings as starting points to model user preferences.

6.1.3 User standardization and rating threshold selection

As found in the analysis of the Movielens dataset, users have individual rating scales which do not allow for an objective extraction of well-rated movies on a single-user basis and increase the complexity when comparing users to one another. The analysis thus suggested to perform a standardization step on the raw ratings for each user to mitigate the effect of rating biases, calibrating their "enthusiasm" and finding the neutral ratings, with the aim to comparing better each user.
From the rating matrix, which encodes raw user preferences for items, the standardization procedure is performed on each user (not on movies). The standardization technique chosen for this step is the z-score standardization, which returns the ratings with zero user's mean and unit user's standard deviation [63] is as follows:

$$(u, r, i) \mapsto (u, r', i), \qquad r' = \frac{r - \mu_u}{\sigma_u} \tag{6.1}$$

Where r' is the standardized rating, while the user means and standard deviations are calculated as follows:

$$\mu_u = \frac{1}{N}\sum_{i=0}^{N} r_{u,i} \tag{6.2}$$

$$\sigma_u = \sqrt{\frac{1}{N}\sum_{i=0}^{N}\left(r_{u,i} - \mu_u\right)^2} \tag{6.3}$$

Where $N = |R_u|$ is the number of ratings given by u.
By standardizing user ratings, it is expected that the comparison between users may proceed efficiently, especially in the extraction of positive preferences therefore items, independently from the individual rating scales.

However, applying the centered mean as a baseline threshold may leave out important preferences for users who tend to rate movies generally well since, for these users the threshold may extract only exceptionally good items, while also items whose ratings are slightly below the threshold consist of valuable information. There might be also the case of users ignoring to rate negatively items and only rate favourite ones, which has been already discussed.

On the other hand, it is argued that the same reasoning does not hold against users with higher standards (i.e., a lower average rating), since they are less likely to rate a movie highly, therefore, in order to provide them with a neutral rating baseline, an individual threshold is extracted using their mean after the standardization.

Therefore, as an additional step, it is proposed to differentiate the users whereas the rating mean before normalization is above 4 in rating scale. In other words, the threshold τ for these users is by default 4. The value of 4 has been chosen after an inspection of the recommendation methods used by other works [64][65], which are used for either rank a list of recommendations or to extract positive preferences, by applying a static threshold to all users and ratings. As a static threshold may appear useful, it is important to consider individual rating scales and rating behaviours, for which a dynamic threshold may serve as a better baseline rating to extract individually positive preferences.

Hence, the formula to calculate the threshold for an user is given as follows:

$$\tau_u = \begin{cases} \sigma'_u, & if\ \sigma_u < 4 \\ 4, & if\ \sigma_u \geq 4 \end{cases} \qquad (6.4)$$

Where σ_u is the average rating of u prior to normalization, so that the threshold can assume values 4 where the mean is above or equal, σ'_u (as the mean rating after normalization, which is always 0) for the other users. User ratings are standardized on an individual basis to guarantee a more straightforward comparison of user profiles, having now a neutral rating baseline given by τ_u.

6.2 IMDb-retrieved movie features analysis

Using the identifiers of the movies rated in Movielens (9125 in total), it has been possible to extract descriptive metadata from IMDb through the available external interfaces[3]. It has been decided to utilize metadata as movie features, instead of ratings, because the former are more expressive (even without considering their sparsity), thus better exploitable when comparing movies (and ultimately the users) for the diversification step: *"if two movies have similar user rating vectors, that means they are similarly liked, not that their content is similar"* [12].

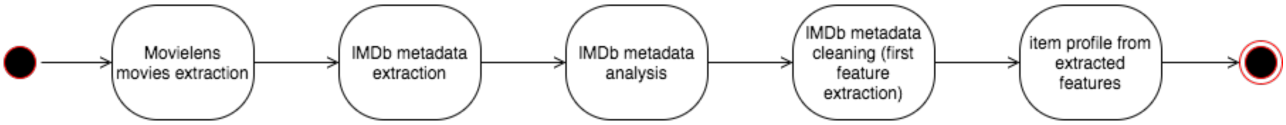

Movielens movies extraction
IMDb metadata extraction
IMDb metadata analysis
IMDb metadata cleaning (first feature extraction)
item profile from extracted features

Figure 6.6. Item metadata analysis and profile extraction.

The methodology followed in this section is illustrated in Figure 6.6, aiming at the extraction of meaningful features describing the content of the items. The proposed methodology to model item profiles first identifies the set of input features, which are analyzed and cleaned of missing or erroneous values (already covered by the feature engineering process, section 5.3). It is important that this step ensures that the original dataset is preserved and that no items are removed as a result. The remaining features compose the terms used to describe the items (item profiles from the extracted features).

The exploratory data analysis was limited on the metadata extracted from IMDb, provided that the type of feature has been considered for their descriptive value, hence other types of features, such as technical specifications, gross revenues, etc. have been ignored. Other interesting metadata such as user reviews or PG

[3] IMDb does not allow scraping their websites for obvious reasons and the extraction of metadata is only allowed for research purposes, which are according to this work.

ratings were not considered, as it has been deemed that the retrieved set of metadata was sufficient to have an exhaustive while limited set of structured features(explained in Table 6.1).

Feature name	Description
Cast	Names of the first 10 leading actors
Company	Name(s) of the production company/companies
Countries	Country/countries of production
Director	Name(s) of the movie director(s)
Genres	Genres of the movie
Keywords	Keywords associated with the movie plot
Languages	Spoken language(s) of the movie (of subtitles if silent)
Original music	Composer(s)
Release date	Year of the movie release
Writer	Name(s) of the movie writer(s)

Table 6.1. Description of the extracted features.

Therefore, a movie m can be represented as a set of the extracted metadata. In detail, a movie $m|m \in M$ can be structured as:

$$m_i = \begin{pmatrix} Cast_i \\ Company_i \\ Countries_i \\ Director_i \\ Genres_i \\ Keywords_i \\ Languages_i \\ Original\_music_i \\ Release\_date_i \\ Writer_i \end{pmatrix}, \quad m \in M \qquad (6.5)$$

The analysis on movie metadata is motivated by the sparsity of ratings and to extract movie similarities based on content. In turn, this step is subordinated to finding meaningful features which can express movie contents. As such, the metadata analysis seeks insights on:

- How are movie metadata distributed? What kind of features occurr the most in the dataset? Are there any noticeable patterns that could bias the proposed recommendation approach?
- What is the relative importance of single features to movies?
- How can missing metadata be handled?

6.2.1 Metadata extraction rationale

The drivers that led to the selection of the above features considered mostly the value each one brings to the description of a movie.

While *genres* are described as universally accepted categories to classify movies, they only works at a broad level and leave aside other meaningful aspects. Moreover, **genres** are not mutually exclusive and often, movies are the result of genre contamination, leading to assume that movies exhibit aspects from more than one genre at once. Nevertheless, the reliability of genres allow to find similarities among movies, even if on a shallow level.

Hence, other types of metadata have been considered, to grasp the subtle aspects which genres are not able to.

One metadata similar to genre but more fine grained consist on **keywords**, which represent descriptive, stylilistical and sub-topical aspects of genres[4]. Some have been extracted from movie plots, while others have been manually inserted by IMDb users and for this reason, keywords are akin to tags. Nevertheless, there are important considerations to be made: in IMDb, keywords are represented in a descending order of relevance for movies, thus the first ones are more descriptive than the rest, and they are inputted by IMDb users and are subsequently reviewed to ensure their quality, the last important point is that there are standard keywords shared by movies, while many others are present in fewer movies.

In terms of metadata deemed important, **director**, **cast**, **composer** and **writer** consist of names of people who were involved in the creation of movies. The choice of director and cast is perhaps obvious when categorizing movies, as influent director(s) and cast often evoke the general atmosphere of movies even regardless of the genres. However, there are also instances of cast and directors being involved in totally different works, suggesting that while some control every aspect of the movies and have "trademark" styles, others are more adaptive. On the choice of composers and writers, the criteria were similar: (screen)writers control the storytelling of movies, they implicitly influence the direction and the *pathos* represented by dialogues (if present) or by the story itself. Composers have a similar role to writers in that they are responsible to create the atmosphere of the movie, encompassing both music, sound effects and often dialogues, thus being an important aspect of movie aesthetics. **Production companies**, on the other hand, also influence the stylistic choices in movie direction and often, influent ones are associated with a particular evocative style.

The other metadata, **countries**, **languages** and **release dates**, instead are not related to stylistical choices, but can be interesting for categorizing movies from other points of view: countries and languages for describing movies produced also evoke stereotypes, such as the cast dominant ethnicity or the sceneries given by shooting locations; on the other hand, release dates, while less descriptive, are interesting especially when considering movies from past decades, which evoke distinct impressions on the movie settings.

6.2.2 Exploratory metadata analysis

Having understood the rationale behind the selection of metadata, an analysis of the metadata distribution is useful to understand at first which general types of movies are available in the dataset.

6.2.2.1 Movie genres

The exploratory analysis begins with the inspection of genres. It is important to state that genres are not mutually exclusive categories and most of the movies present several genres at once.

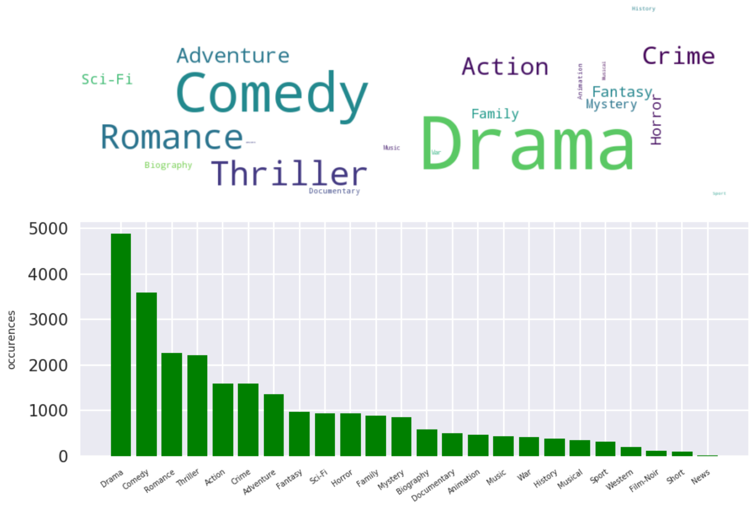


Figure 6.7. Genre distribution per movies.

[4] A keyword is a word (or group of connected words) attached to a title (movie / TV series / TV episode) to describe any notable object, concept, style or action that takes place during a title. The main purpose of keywords is to allow visitors to easily search and discover titles [66].

The distribution shows that there are dominant genres (Figure 6.7), which are also considered the most general, as they do not add meaningful insights on movies apart from the overall theme [33] following also the observations of the previous section: drama, comedy, romance and thriller consist on the most occurring genres in the dataset. From this distribution, it is clear that the majority of movies belong to these genres which, even if not disjointed, may confirm that the movies in the dataset are biased towards these genres and, given that they have been extracted from the Movielens dataset, may suggest that they are also the most popular among users.

6.2.2.2 *Keyword analysis*

Keywords, on the other hand are more fine grained, but given the prevalence of the aforementioned genres, it is expected that the most popular keywords may be related to them.

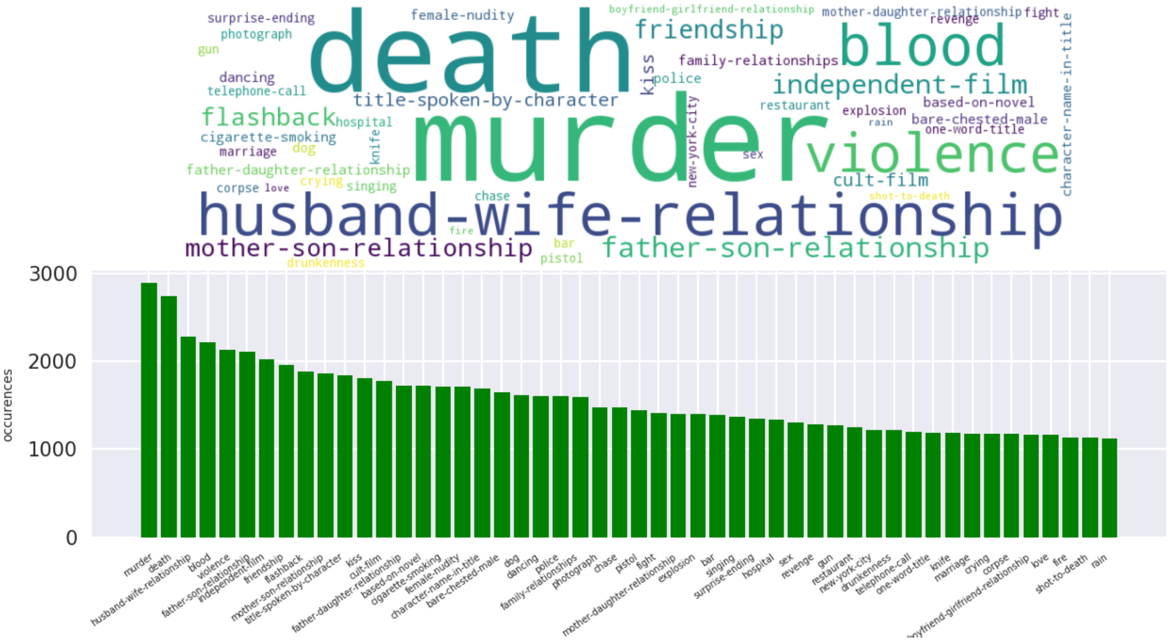


Figure 6.8. Popularity of the first 50 keywords.

In fact, from what can be seen in Figure 6.8, the most occurring keywords are highly related to thriller (murder and death) and drama/comedy (family relationships) genres. However, an important consideration is that keywords are widespread (shown by the slow decrease in number of occurrences), allowing leverage their discriminating aptitude to compare movies more effectively than with genres only.

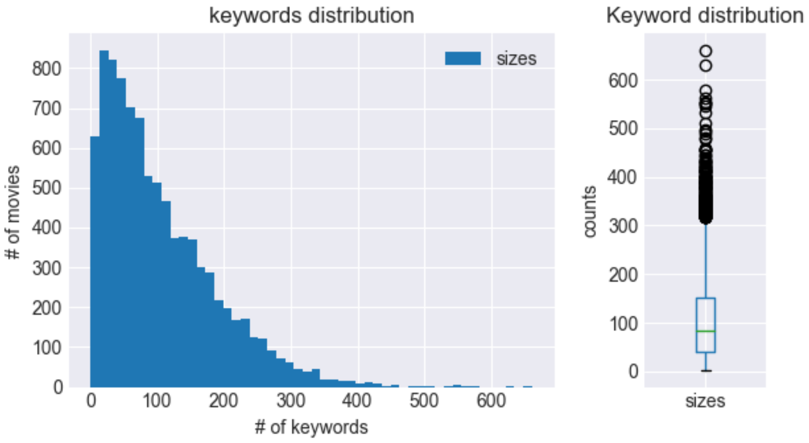


Figure 6.9. Keyword distribution for all movies in the dataset, prior to cleaning.

Another aspect of keywords may result from their distribution across movies (Figure 6.9), which exhibits a long-tail figure: there are few movies with a high number of associated keywords (presumably due to their popularity), while the majority has less than 100. Analysing this aspect may suggest that the more keywords are associated to movies, the less relevant are, given that they are ordered by relevance. Thus, a naive approach to this issue would be to filter out keywords above a certain threshold, which can be established empirically in order to maintain both descriptive granularity and generality.

Given that keywords are more descriptive than genres, now a more in depth analysis on the latter can be performed to understand their facets by inspecting the most common keywords associated with each genre.

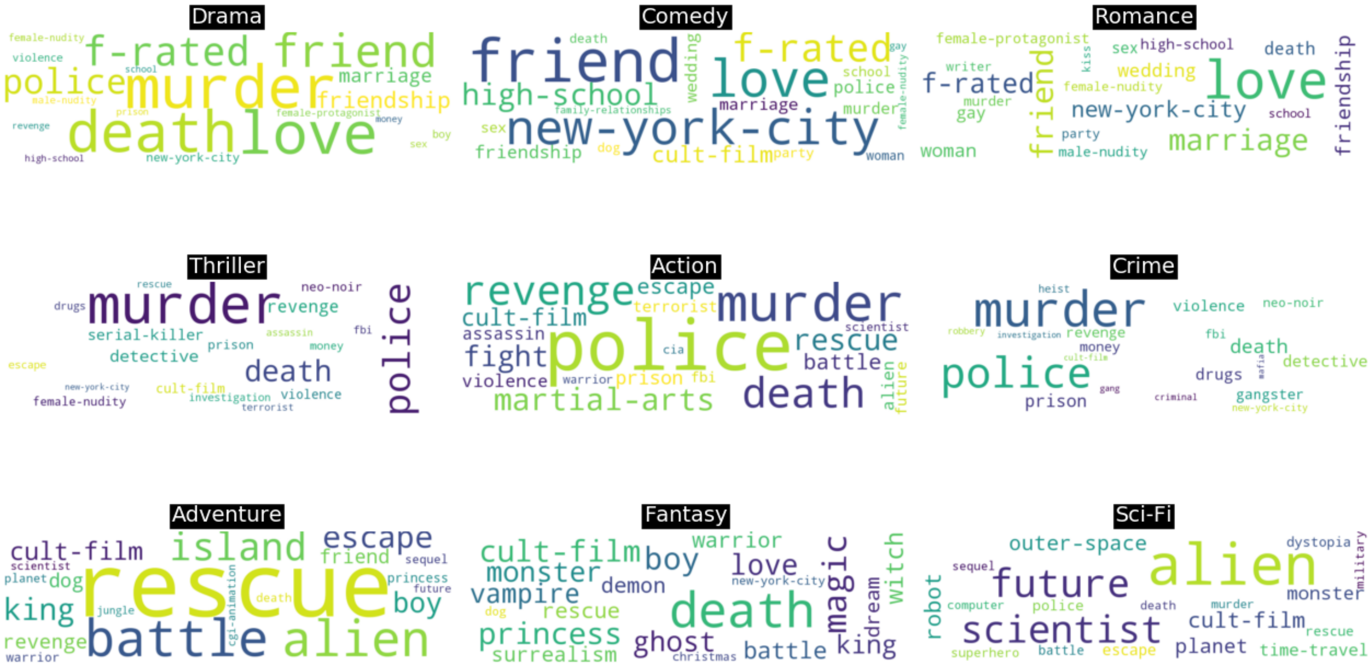


Figure 6.10. Most common keywords associated to the 9 most occurring genres.

Figure 6.10 shows how, unsurprisingly (but interesting nonetheless), the most common keywords vary for the same genre. Interestingly, the same keywords occur also in different genres (e.g. *police* and *murder* for the *thriller* and *crime* genres), perhaps because the movies, from which the keywords have been extracted, belong to multiple genres.

Thus, by inspecting slightly less common keywords for each of these genres, the subtler differences in each genre facets can be uncovered (Figure 6.11).

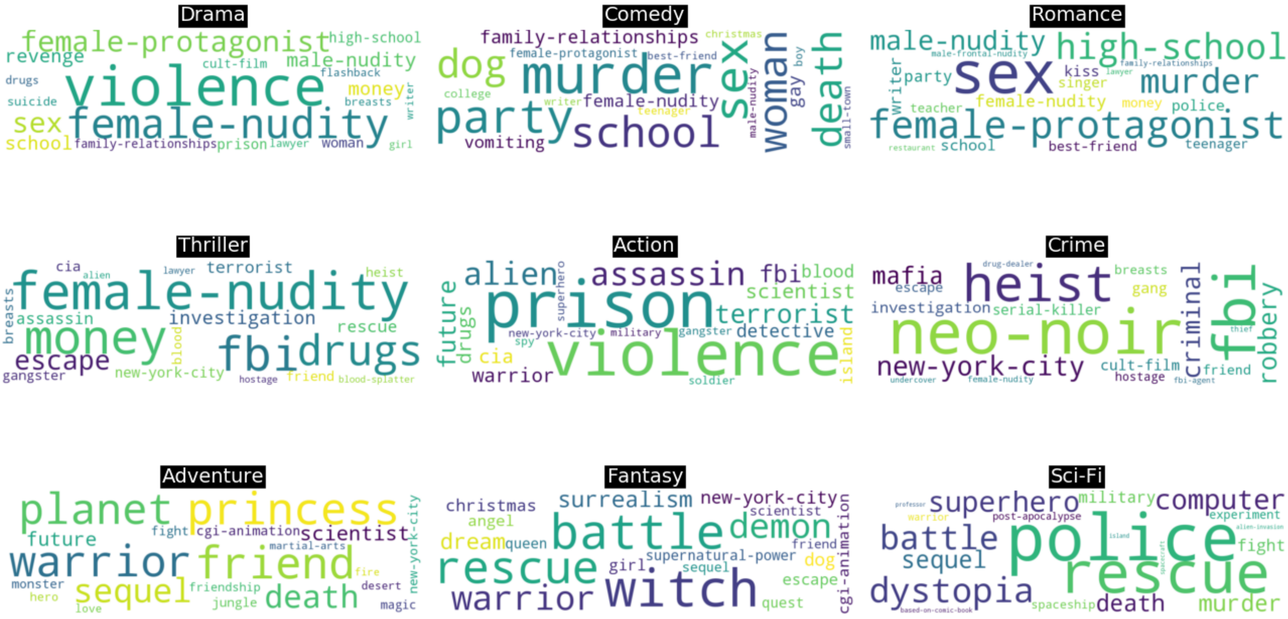


Figure 6.11. Less occurrent keywords (the top 10 removed) for the top 9 genres.

Indeed, removing the 10 most general keywords for each genre, the keywords appear to be more genre-specific, while also showing already divergent aspects within them (e.g., for the *fantasy* genre, keywords like *battle* and *witch* appear to form different "subgenres").

6.2.2.3 Release, country and language analysis

An analysis on movie distribution by decade (Figure 6.12), taken by the year of **release**[5], allows to understand better how movies are distributed in time.

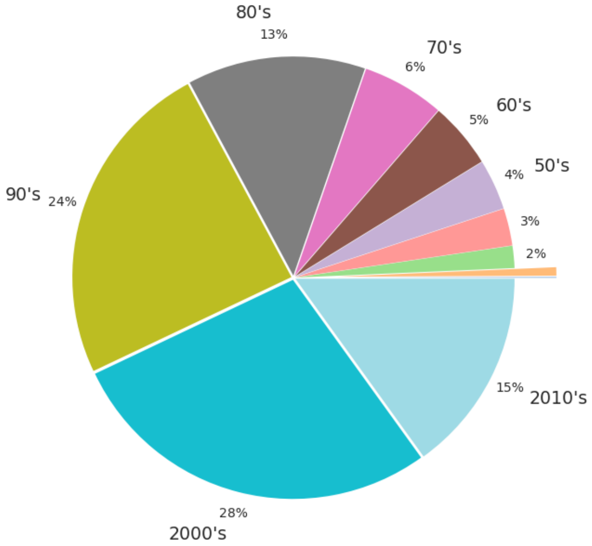


Figure 6.12. Movie distributed by decade of release.

[5] At this stage, for exploratory analysis the release dates were taken from Movielens if missing during the metadata extraction.

While the release date may not give particular insights on the movie content, the chart interestingly shows that users tend to rate more recent movies, as more than 70% have been released from the 80' up to now. Also, an increasing amount of movies per decade can be noticed.

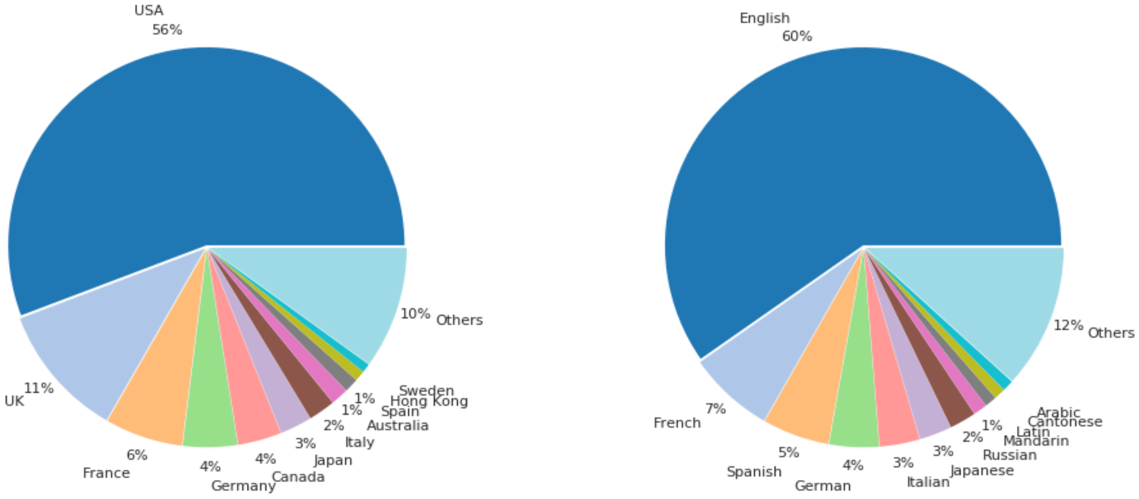


Figure 6.13. The 12 most occurring countries and languages.

Analysing **countries** and **languages** of production shows a dominance for English speaking countries and, suggesting that implicitly, users tend to rate more movies from these countries and potentially are of the same nationality. For limitations, only the first 12 countries and languages have been shown (Figure 6.13), as there are many more, but with a much lesser incidence.

Even though an analysis on user nationalities is not the focus of this section, there might be a correlation between mother country or mother tongue of users and those of the rated movies; specifically, there is a strong correlation between the countries and their respective most common spoken language (i.e. *English* for *USA* and *UK, Japanese* for *Japan*, etc.).

6.2.2.4 Remaining metadata analysis

An analysis on **cast, directors, writers** and **composers** may give insights on how proficient are each category of person, by finding how many movies they have worked in (Figure 6.14).

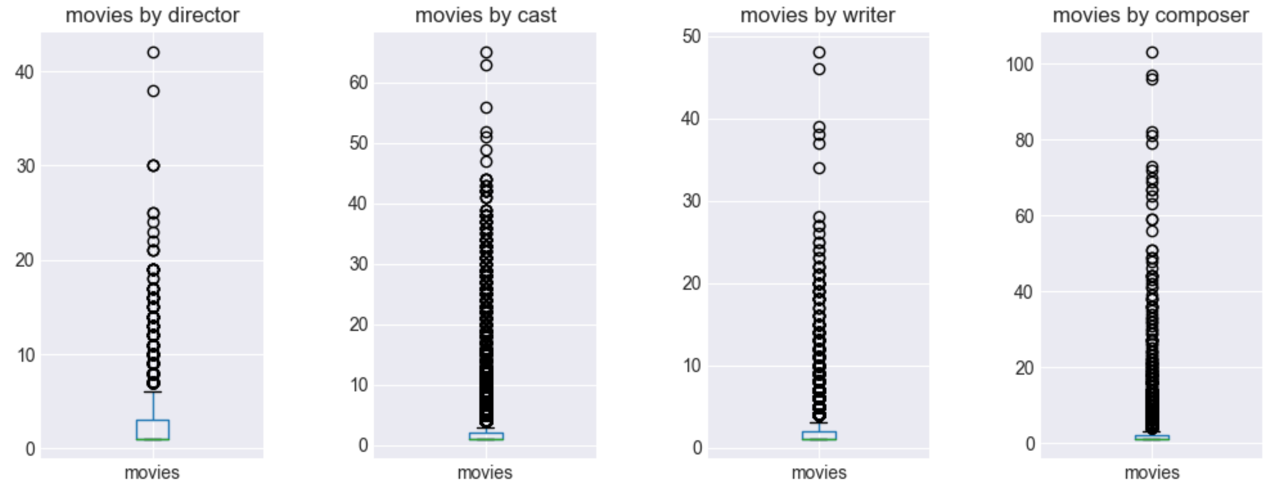


Figure 6.14. Distribution of proficiency for people involved in movie creation.

The last four metadata (Figure 6.14) show the general proficiency for individual directors, actors/actresses, writers and composers (for this purpose, movies in which these features were missing have not been considered): apart from noticeable outliers, most tend to have low proficiencies regarding the number of movie productions they have been involved into. It is perhaps spontaneous to assume that the "outliers" represent celebrities in their fields, given their many occurrences. This assumption is better supported by Figure 6.15, depicting the names of the most occurring persons having worked at the movies in the dataset.

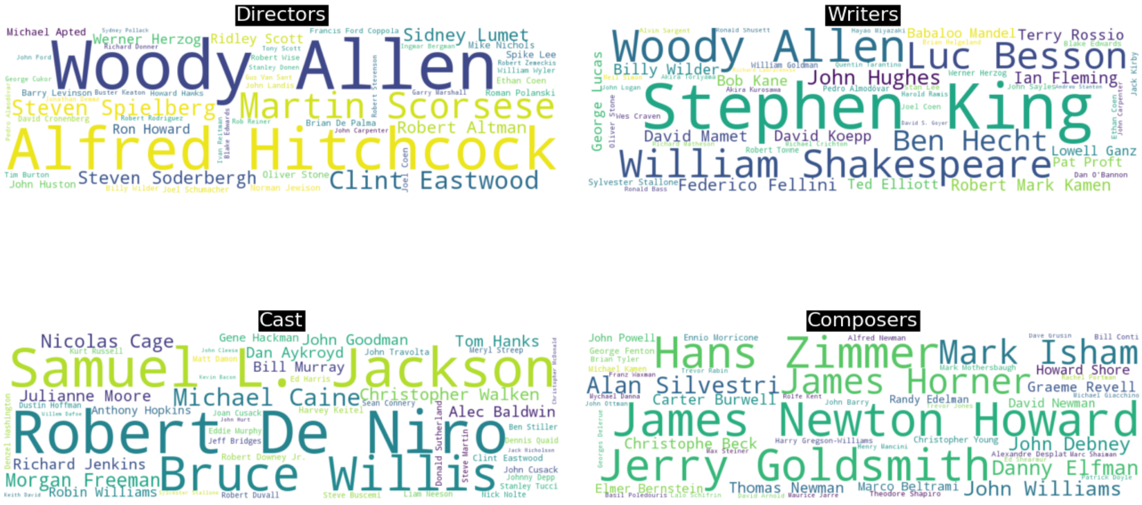


Figure 6.15. A closer look to the most proficient figures.

Looking at the figure, many contemporary celebrities can be noticed, as well as icons from the past decades with the exception of writers (while there are recent names here as well, some others come from past centuries, such as Shakespeare). To the author's knowledge, most of the above names can be related to drama and thriller movies, suggesting that there might be a correlation between their occurrences in the movies rated and the types of popular movies users tend to watch.

Lastly, an exploration of the distribution of production companies (Figure 6.16).

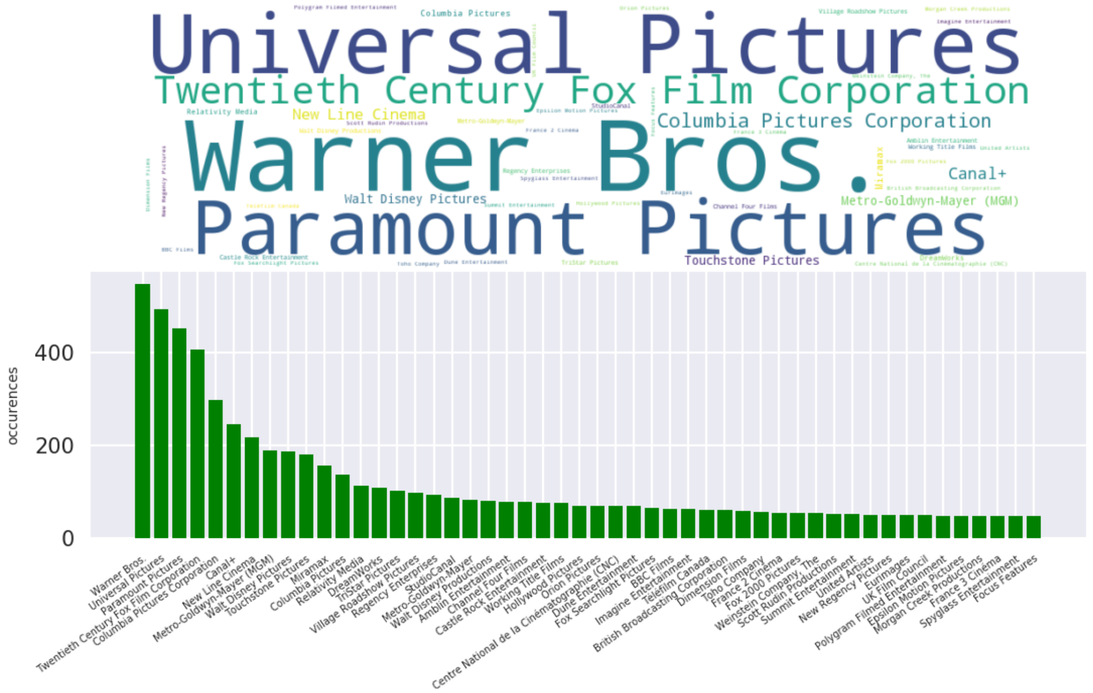


Figure 6.16. Production companies popularity and distribution.

As the production company does not add much knowledge to the domain, apart from what can be derived by the movies usually produced, they can be associated to other aspects such as movie budget and gross revenue, since the most producing companies also tend to have the most incidence of high revenues, as they effectively add as sponsors for the movies. Thus, the advantages to include this feature are mostly implicit but can be more expressive than including gross revenue and budget features. Moreover, to a single movie there are often more than one associated production company.

Apart from that, most of the occurring companies are associated to theatrical movies, as they produce only for the cinema industry. Nevertheless, there can be also companies producing movies for home entertainment, video-on-demand and independent movies and thus, can be less popular than those in the industry. Also, from what can be seen, the companies depicted are all coming from English-speaking countries and could be distributed either in USA or UK, suggesting that the users have clear preferences for established companies and for English-speaking movies.

6.2.2.5 Exploratory analysis summary

The metadata exploratory analysis served the purpose of inspecting the movie contents to clarify that the dataset (recalling that movies have been derived by the Movielens dataset, the exploring IMDb metadata may also point to further insights on rating behaviours of the users) has the following properties:

- There is a prevalence of drama/comedy and thriller movies, as shown by genre and keywords inspection.
- A closer look at genres has shown that each has different facets according to the most occurring keywords.
- Movies have a generally low number of associated keywords, some of which are consistently present throughout the movies.
- According to the decade, there is a prevalence for more recent movies, compared to decades prior to the 60'.
- The countries and languages movies have been produced the most consist generally of English-speaking countries, followed by European countries and languages and lastly by Asian ones.
- Most of the movies have generally once-performing casts, directors, writers and composers; however, there are also recognizable names in their respective fields. The same holds for production companies.

6.2.3 Metadata cleaning

With an understanding of the metadata included in the IMDb dataset, this section follows the last two steps of the feature engineering process in order to select the most descriptive features from all metadata.
The first step is to analyse the missing metadata and generate a set of criteria to be followed during the cleaning step.

6.2.3.1 Missing metadata analysis

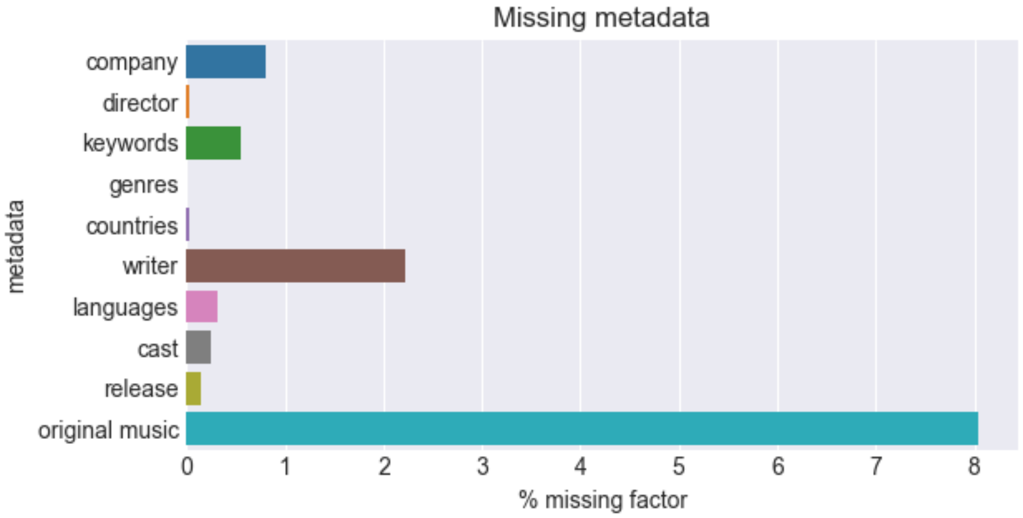


Figure 6.17. Percentages of missing values for each metadata in the original dataset.

Figure 6.17 shows relative missing percentage for all metadata considered, prior to the cleaning step. Out of all movies (9125), the missing values do not have a strong incidence and it can be seen that genres are globally present. However, an important reflection upon the chosen metadata is needed before performing on the dataset: from an initial inspection of missing values of particular metadata (directors, composers, writers, cast, companies and languages), it has been understood that some movies do not include some kinds of aspects by design and instead, is important to note that the lack of such features is not an error but rather a peculiarity of these movies, thus missing values can be features in themselves, depending on the kind of movie.
For example, considering that there are movies from the first decades of the past century, missing values can mostly include soundtracks and languages for silent movies (although intertitles can be considered as the featured language of the movie, if present), while documentaries and animated pictures may lack the cast or a director, a soundtrack or a scenography or even the support of a production company (e.g. for independently produced movies).

Thanks to this insight, there is a possibility to distinguish metadata as essential features for the movies from other metadata which has been erroneously omitted in IMDb[6].

Therefore, the following cleaning criteria have been considered:

- A lack of features cannot be naively interpreted as an error.
- The above argument is not valid for all features, especially for those of high importance for a movie (such as keywords).
- For some movies, the lack of a particular feature is considered itself a feature (only regarding the *cast*, *company*, *director*, *writer* and *composer* features).
- There are features (keywords) with more descriptive, less ambiguous power than others (directors, writers, composers, etc.).
- If unavoidable, movies with features that should be present while they were already absent prior to the metadata extraction and cannot be reinserted automatically, should be removed from the dataset.

[6] The extraction of metadata has been a one time operation (and has not been optimized for periodic extractions). It is therefore likely that movies with missing features were filled after the mining.

The following section thus focuses on the cleaning step, considering the above criteria in order to avoid introducing biases towards specific categories of movies.

6.2.3.2 Metadata cleaning methodology

The cleaning process is vital to the creation of a more compact dataset to be used in the feature extraction step and the model training part.

In detail, the following cleaning steps, explained in the next sections, are:

1. Preprocess features so that they can be extracted using the same or the Movielens dataset
    a. Cleaning of release dates.
    b. Cleaning of countries and languages of production.
    c. Preliminary cleaning of keywords.
2. Keep the remaining features even if missing, given that there is a possibility that missing features are a peculiarity of some movies.

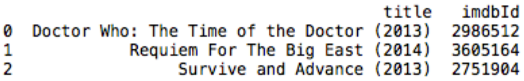
6.2.3.3 Cleaning of release date

This is the step that has been performed prior to the analysis of the decades: it was required since the format extracted from IMDb included the complete date, which was too fine grained. Hence, only the years have been extracted from the release dates. Regarding missing release dates, the Movielens dataset has been exploited through extraction of the years of release contained in each movie title.

6.2.3.4 Cleaning of countries and languages of production

If the language is not present, it is either because the movie is silent or because it is not included in IMDb. A look at some IMDb pages in comparison with their Wikipedia pages has shown that for some affected movies, Wikipedia includes languages unlike IMDb.

The analysis of countries and languages suggest that there is a strong correlation between the two variables, supported by the fact that spoken languages mostly come from the movie production country. This insight, combined with the fact that there are more missing languages than countries and that the intertitles of silent movies are highly related to the same countries of production, have been decisive in this step. In order to fill the missing languages on a country basis, for each country, all the movies in which it occurred have been selected and subsequently the most common language has been chosen (e.g. *English* is the most common language for movies produced in *USA* or *French* for *France* and so on).



```
                                      title   imdbId
0  Doctor Who: The Time of the Doctor (2013)  2986512
1             Requiem For The Big East (2014)  3605164
2                  Survive and Advance (2013)  2751904
```

Figure 6.18. Movies without language and country.

At the end of this process, only three movies (Figure 6.18) remained without both language and country of production, which, after a manual inspection, have been filled with the most common country and language in the dataset (*USA* and *English*).

6.2.3.5 Cleaning of movie keywords

As derived by the analysis of genres and keywords, the two features seem related in terms of descriptive granularity, moreover some keywords have the same depth as genres. For these reasons, genres can be utilized as keywords when the latter are absent for their respective movies, also because genres are ubiquitous.

Thus, a first part of this procedure consisted in filling the missing keywords by exploiting the those associated with each encountered genre of the affected movies; since this step may result in biasing the dataset towards some genres even more, for a genre, only the 15 most common keywords have been extracted to describe the movie on a high level.

Secondly, in order to remove noise caused by redundant terms, each keyword has been reduced to its root form (stem). As seen in Figure 6.9, most movies have less or around 100 associated keywords, while the rest have abnormal numbers of keywords. Nevertheless, it has been chosen to describe each movie with the full keywords dimensionality: while many keywords can be unique, maintaining all keywords can be beneficial in finding the subtler aspects on which movies can be compared, moreover, this step is only part of the construction of item profiles, which will be weighted in the preprocessing module to further remove noisy features.

6.2.3.6 Discussion on the other metadata

Regarding the remaining metadata (*cast*, *company*, *director*, *writer* and *composer*), the following represent the rationale for not removing affected movies: unlike the other metadata, these cannot be replaced by averages or common values since they are unique and represent important aspects of individual movies.

Moreover, given that the absence is also a feature, there is an added complexity to filling missing values. Also, since these movies were extracted from the Movielens dataset, users have clearly defined some preferences for them, which may bring an advantage for the recommendation approach: while users have predominantly expressed preferences towards a particular actor or actress, if the cast is present, other users do have preferences for affected movies and can result more similar to one another.

Hence, it has been decided neither to remove movies with these missing values nor to attempt to fill these, as the latter would require an extensive analysis of the affected movies.

6.2.3.7 Results of the metadata cleaning step

After the cleaning step, each movie has been filled where applicable and the resulting dataset missing factors have been updated, as shown in Figure 6.19.

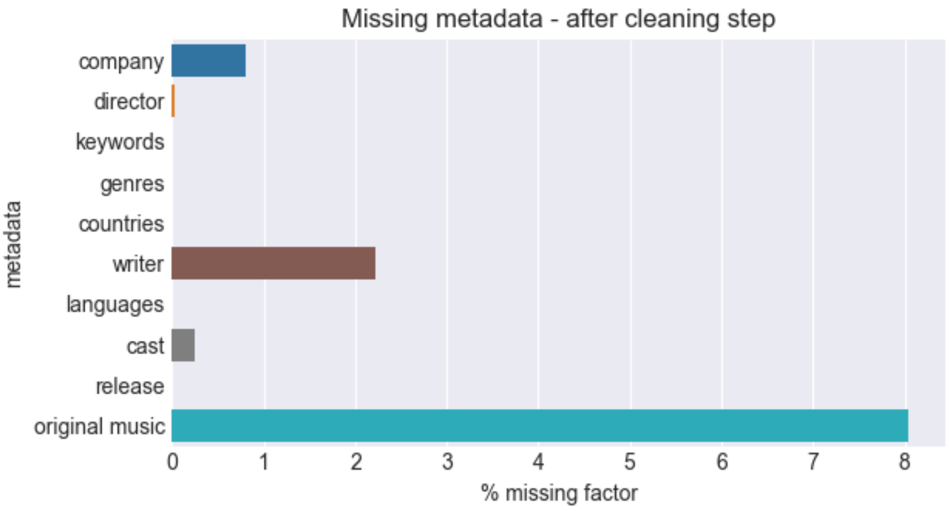


Figure 6.19. Percentages of missing values for each metadata after the cleaning step.

As expected, the only missing features are the most important for the movie description, along with keywords (which have been filled completely).

Hereby, a summary of the cleaning procedure is provided:

1. Find the number of missing values for every metadata.
2. Transform the release date in only the year of release, from the IMDb format, complete with country, day and month of release. For movies without release date from IMDb, use the release given in the Movielens dataset.
3. Fill missing keywords using common instances from the first occurring genre of affected movies and determine a relevance threshold.
4. Fill missing languages and countries of production using common instances from unaffected movies.
5. Keep remaining missing features.

The result of this step provides a list of selected features assigned to each movie to compose the initial item profiles.

7 Implementation and evaluation of the proof of concept

This chapter is focused on the implementation of the proposed approach. However, for academic constraints, only a small part of the full recommendation procedure can be implemented and discussed in detail, which serves as the proof of the concept hereby proposed. Therefore, as explained at the end of chapter 4, the implementation is aimed at the formation of the distant neighborhood and the evaluation of the user diversity. The next sections target the following implementation in depth:

- Item categorization, with clustering analysis.
- Category-based preference modeling and user groups formation, with clustering analysis.
- Setup of the experimental evaluation to be performed on the distant neighborhood formation.

Specifically, since the proposed approach is established on the concept of a taxonomy of items, the aim of each section is to understand that the proposed approach meets the following hypotheses:

- That the items can form categories more or less homogeneous.
- That the users can be clustered according to the found categories, and finally,
- that selecting the distant users within the same group of users can fulfill the diversity VS accuracy trade-off.

7.1 Item categorization

The item categorization follows the process described in section 5.4.1. However, the popular items have been sampled from the full dataset so that they can be easily recognizable for the clustering analysis. As such, after the items selection, the implementation is divided in two subsections, focusing on LSA and clustering analysis respectively. As a last section, the results of the clustering analysis are presented and analyzed.

It is hypothesized that items can be clustered together with the structured metadata associated; for this reason, items can be treated as documents and LSA can be performed as a first step to extrapolate the item similarities based on the latent features. Nevertheless, after the exploratory analysis, there are still many uncertainties regarding the expected number of categories for the movie application domain, given the limited knowledge of the domain.

Since the user preferences are built from the categories, the cluster quality has to be carefully inspected.

Hence, the clustering analysis is performed to select a reasonable number of clusters given the distribution of associated terms, as a manual inspection of cluster quality. Also, the elbow method may prove to be useful. In fact, it is argued that for this application domain, an appropriate heuristic would be to analyze the result of the dendrogram obtained through hierarchical clustering.

An important consideration is necessary, regarding the metadata format and its implication on LSA: metadata are already structured, therefore movies cannot be treated as unstructured documents, as originally envisioned in LSA. Moreover, most of the associated metadata, after an initial inspection, are not duplicated for a single movie (even though there might be cases where the same director also is the writer, or where the name of an actor/actress is repeated as a keyword).

7.1.1 Selecting the items for the categorization

In order to show how the proposed item categorization works, a subset of items has been selected under the following criteria:

- Each movie has at least 3 ratings.
- Each movie has an average rating equal or above 3.

Since the ratings follow a long tail distribution, both users and items are affected. In order to obtain clusters of high quality, it has been necessary to limit the coverage of the categorization to both generally popular and generally well rated movies. Nevertheless, the clustering can be generalized to all movies, in a full scale solution, since periodic updates of the categorization module are essential to allow new movies to be classified or new clusters to be formed. As such, a total of 3685 movies out of the 9125 present have been retrieved.

7.1.2 LSA on movie metadata: TFIDF and SVD

Hereby, the decisions regarding the implementation of TFIDF and dimensionality reduction are presented. After further preparing the data so that each movie is represented as a bag of words without duplicates, applying TFIDF results in merely an IDF of the binary matrix of occurrences.

The choice of IDF formula for the implementation is given by equation (3.4), which allows to not ignore completely common words.

The item profiles, composed of metadata extracted after the cleaning step are TFIDF-weighted. To this purpose, the profiles have been further cleaned of the remove unique or very rare terms by controlling the minimum document frequency.

It is argued that as the document frequency (DF) threshold increases, only the popular terms are kept, leaving out unique words such as one-time performant directors, actors/actresses, unique keywords, etc. However, it is also argued that a large threshold may filter out important information related to subtler similarities among items (e.g., among niche movies, which have particular themes), while preferring popular terms: mainly, popular cast and the likes, more generic keywords and mainstream genres. More specifically, the minimum document frequency can control the generality/specificity of the terms. In fact, it seems reasonable to work at a low minimum frequency, to ensure the accuracy of the similarities among movies, since the analysis on metadata, in particular on the cast, directors, writers and composers (Figure 6.14), has shown that the majority has worked in less than 10 movies. Nevertheless, it is reasonable to filter out unpopular terms, since they are so sporadic to represent mainly noise. In fact, out of 91343 terms, 52114 (~57%) appear to be unique (except for the feature *English*, which appears in at least 90% of the movies and has been removed); therefore, have been filtered out for the analysis.

Successively, SVD is applied on the TFIDF matrix, preserving the 500 largest singular values.

In order to evaluate the performance of SVD, it is proposed to extrapolate the explained variance after the application on different instances of the TFIDF matrix, by tuning the minimum document frequency threshold. The immediate analysis of the application of SVD on the TFIDF-weighted matrix is provided.

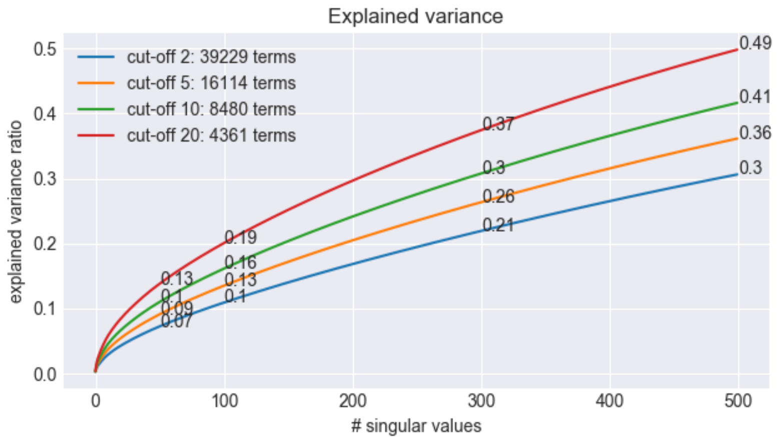


Figure 7.1. Explained variance after SVD on TFIDF with different minimum document frequency thresholds.

Figure 7.1 illustrates the amount of information retained after the application of SVD on different instances of TFIDF matrices, which have been selected by determining the document frequency threshold (in the legend), so that terms occurring in at least that number of documents are preserved, reducing the total amount of terms for the dimensionality reduction. In detail, each curve shows the cumulative variance annotated for the first 50, 100, 300 and 500 singular values.

As can be seen, the variance retained after the application of SVD increases as the minimum DF increases, which is expected, given that the amount of features on which dimensionality reduction is applied decreases accordingly.

Furthermore, Figure 7.1 points to a more critical issue: the maximum number of largest singular values retained here (500) is unable to explain more than half of the variance contained in the original TFIDF matrix, suggesting that a more proper feature extraction of the metadata or a more sound term weighting scheme are necessary to extract the preliminary feature profiles. Moreover, it is expected that the more singular values are kept, the better SVD is able to approximate the TFIDF matrix; however, increasing the number of latent dimensions would result computationally heavy for the model.

Nevertheless, it has been chosen to utilize 2 as the minimum document frequency, as it gives the maximum amount of information to compare the movies at subtle levels, to understand the similarity obtained by varying the amount of singular values retained.

Therefore, an initial examination has been performed to manually evaluate the results of applying cosine and euclidean similarity metrics by taking the product of U and S, the first of which relates the items to the latent features and the second encodes the importance of each latent feature. As a result, the matrix US allows to find similarities based on the latent features with most magnitude. In particular, Table 7.1 and Table 7.2 report the results of the 10 most similar movies to the query after having normalized the output of SVD and using the Euclidean distance from (3.8) as a metric to find the closest movies. Two popular movies, *The Matrix*[7] and *Mad Max*[8] have been chosen to understand the performance of LSA by controlling the k largest singular values. The first movie has been chosen for its popularity and the easily recognizable themes (*cyberpunk, dystopia, sci-fi, action*), while the latter is a relatively recent movie but with equally distinguishable themes (*post-apocalyptic world*). A comparison on such titles is therefore more straightforward if the movies are popular.

	The Matrix			
k	**50**	**100**	**300**	**500**
1	*Matrix Reloaded, The (2003)*	*Matrix Reloaded, The (2003)*	*Matrix Reloaded, The (2003)*	*Matrix Reloaded, The (2003)*
2	*Matrix Revolutions, The (2003)*	*Matrix Revolutions, The (2003)*	*Matrix Revolutions, The (2003)*	*Matrix Revolutions, The (2003)*
3	*I, Robot (2004)*	*Animatrix, The (2003)*	*Animatrix, The (2003)*	*Animatrix, The (2003)*
4	*Terminator, The (1984)*	*Terminator, The (1984)*	*Terminator, The (1984)*	*Terminator, The (1984)*
5	*Terminator 3: Rise of the Machines (2003)*	*Terminator 2: Judgment Day (1991)*	*Terminator 3: Rise of the Machines (2003)*	*Terminator 3: Rise of the Machines (2003)*
6	*Terminator 2: Judgment Day (1991)*	*Terminator 3: Rise of the Machines (2003)*	*I, Robot (2004)*	*Assassins (1995)*
7	*Animatrix, The (2003)*	*I, Robot (2004)*	*Terminator Salvation (2009)*	*Tron (1982)*
8	*One, The (2001)*	*Blade Runner (1982)*	*Tron (1982)*	*Blade (1998)*
9	*Total Recall (1990)*	*Terminator Salvation (2009)*	*Blade Runner (1982)*	*One, The (2001)*
10	*Blade Runner (1982)*	*Island, The (2005)*	*Terminator 2: Judgment Day (1991)*	*I, Robot (2004)*

Table 7.1. 10 most similar movies to: The Matrix, retrieved from number of singular values retained.

As can be seen, there are substantial differences in the similarity ranks for the movies considered in the comparison, according to the number of latent features retained. For *The Matrix*, while all resulting movies can be considered related to the title, with 50 and 100 singular values (SV) there is a slight decrease in the perceived ranked similarity. At 300 and 500 SV, the sequels and spin-off are grouped together. For *Mad Max*, instead the most similar movies are less obvious: all movies are somewhat related to the *post-apocaliptic world* theme but do not cover the same topics (e.g., there are also *horror, super-hero* and *action* movies.

[7] http://www.imdb.com/title/tt0133093/

[8] http://www.imdb.com/title/tt1392190/

	Mad Max: Fury Road			
k	**50**	**100**	**300**	**500**
1	*Running Man, The (1987)*	*Maze Runner: Scorch Trials (2015)*	*Road Warrior, The (Mad Max 2) (1981)*	*Road Warrior, The (Mad Max 2) (1981)*
2	*American Ultra (2015)*	*Dredd (2012)*	*Maze Runner: Scorch Trials (2015)*	*Mad Max Beyond Thunderdome (1985)*
3	*Resident Evil: Afterlife (2010)*	*Dark Knight Rises, The (2012)*	*Mad Max Beyond Thunderdome (1985)*	*Dark Knight Rises, The (2012)*
4	*Maze Runner: Scorch Trials (2015)*	*Surviving the Game (1994)*	*Dark Knight Rises, The (2012)*	*Maze Runner: Scorch Trials (2015)*
5	*Omega Man, The (1971)*	*Omega Man, The (1971)*	*Omega Man, The (1971)*	*Vampire Hunter D: Bloodlust (Banpaia hantâ D) (2000)*
6	*Surviving the Game (1994)*	*Book of Eli, The (2010)*	*Dredd (2012)*	*Omega Man, The (1971)*
7	*Commando (1985)*	*Road Warrior, The (Mad Max 2) (1981)*	*Vampire Hunter D: Bloodlust (Banpaia hantâ D) (2000)*	*Mad Max (1979)*
8	*Expendables 2, The (2012)*	*Resident Evil: Afterlife (2010)*	*Terminator Salvation (2009)*	*Expendables 2, The (2012)*
9	*Dark Knight Rises, The (2012)*	*Running Man, The (1987)*	*First Blood (Rambo: First Blood) (1982)*	*Dredd (2012)*
10	*Dredd (2012)*	*American Ultra (2015)*	*Expendables 2, The (2012)*	*Resident Evil: Afterlife (2010)*

Table 7.2. 10 most similar movies to: Mad Max: Fury Road, retrieved from number of singular values retained.

On the basis of this comparison, it is argued that retaining 500 SV could be reasonable, as the similar movies are more compact and homogeneous.

Another comparison has been focused on the performance of the distance metrics, since the above tables are the results of similarities on the normalized output of SVD. Hence, Table 7.3 summarizes the similar movies for *The Matrix*, retrieved with the 500 largest singular values, without normalized distances.

	The Matrix	
metric	**Euclidean**	**Cosine**
1	*Matrix Reloaded, The (2003)*	*Matrix Reloaded, The (2003)*
2	*Matrix Revolutions, The (2003)*	*Matrix Revolutions, The (2003)*
3	*Animatrix, The (2003)*	*Animatrix, The (2003)*
4	*Assassins (1995)*	*Terminator, The (1984)*
5	*Blade Runner (1982)*	*Terminator 3: Rise of the Machines (2003)*
6	*Tron (1982)*	*Assassins (1995)*
7	*Terminator Salvation (2009)*	*Tron (1982)*
8	*My Man Godfrey (1957)*	*Blade (1998)*
9	*Blade (1998)*	*One, The (2001)*
10	*One, The (2001)*	*I, Robot (2004)*

Table 7.3. A comparison of similarity queries with Euclidean and Cosine distances.

As expected, the similarities retrieved with the cosine metric are more coherent than with Euclidean distances as the latter introduced an irrelevant result (8th) already within the 10 most similar movies, suggesting that cosine represents a valid metric to extract item similarities. The reason why Euclidean distances are not performing well for this particular task may lie in the different vector magnitudes, which affect the pairwise distances, whereas the cosine only calculate the angle between the vectors regardless of their magnitude. Moreover, the output of cosine is the same as with the normalized Euclidean distances in terms of ranked movies, as all vectors have unit norm; therefore, the Euclidean metric can be adopted for the clustering analysis without incurring in the problem of skeweness.

To complete the LSA section, a visual analysis of the SVD output is provided.

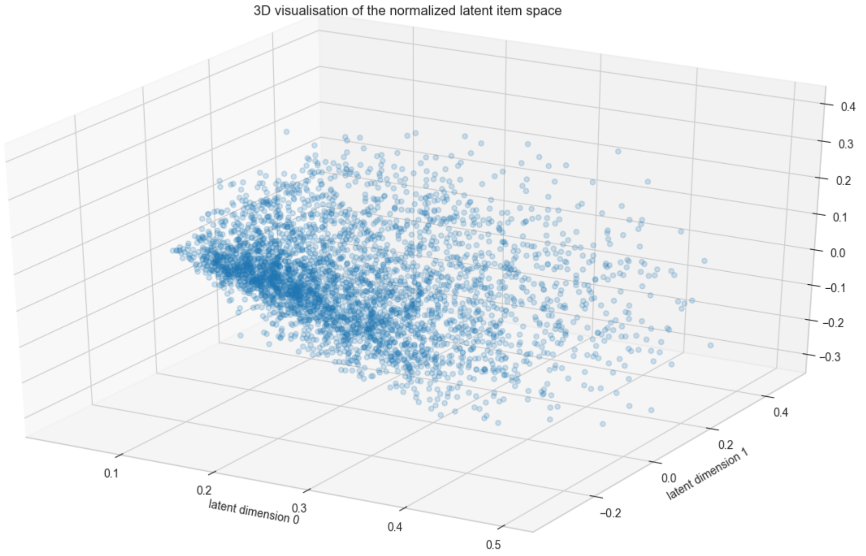


Figure 7.2. Full latent item space after normalization of SVD.

Beginning with Figure 7.2, the full latent space (after the normalization to unit norm of the SVD output) suggests that movies do not form compact structures in three dimensions. Nevertheless, in Figure 7.3 similar movies appear to be closer in this space even in the fourth (3) and second (1) latent dimensions: the red point represents the query for which similar movies have been retrieved.

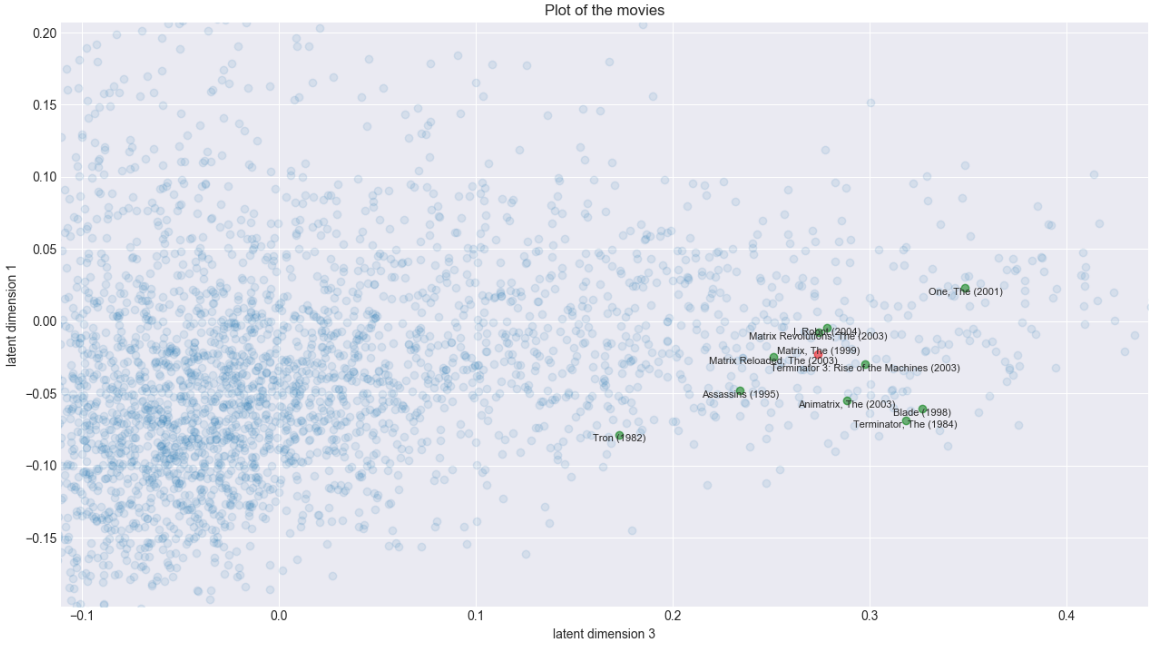


Figure 7.3. The 20 most similar movies to The Matrix in the 0 and 1 latent dimensions.

A wider-range visual inspection of the SVD output (Figure 7.4) can help in disclosing the distributions of different movies in the same latent feature space.

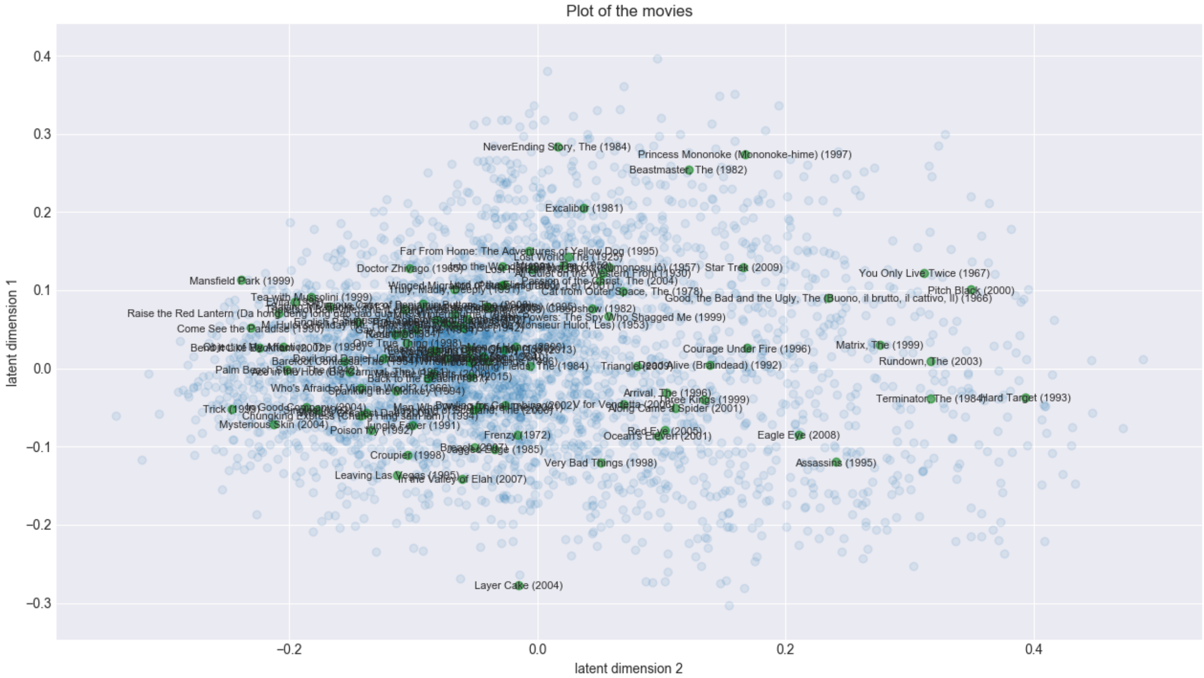


Figure 7.4. General visualization of the movies in the latent dimensions 1 and 2.

As an be seen, some *thriller* and *action* movies tend to group togheter in the right area of the plot, *fantasy* movies can be recognized in the upper section of the plot, while the majority of movies form an agglomeration around the 0 in both dimensions.

7.1.3 Item clustering analysis

After having inspected that the similarities captured by retaining 300 singular values are indeed easily recognizable, the clustering analysis focuses on the creation of the categories. To evaluate the number of clusters, a manual inspection through hierarchical clustering is proposed, since it has been found that movies do not form compact structures in 3 dimensions. Moreover, the number of clusters is not known *a priori*, therefore beginning with an analysis with hierarchical clustering seems a reasonable choice since k-means requires the number of clusters as an input.

7.1.3.1 Selection of the clustering algorithm

The clustering algorithm and distance metric, if applicable, are necessary to understand the problem of categorizing the items from the latent features extracted after SVD. Therefore, as the number of categories is not known *a priori* and k-means requires the number as an input, it is reasonable to employ a hierarchical clustering method.

7.1.3.2 Visual inspection of the clusters with hierarchical clustering

This step is required to visually estimate a suitable number of clusters from by inspecting the average distance within each agglomerative step. It is argued that a low number of clusters may result in creating very general categories, with potential negative interpretation of user preferences. On the other hand, the specificity of the clusters ,given by more compact categories, is favoured and therefore the clustering analysis is focused on finding where this trade-off is considered acceptable.

Firstly, to understand how the similarities vary according to the number of $topN$ most similar movies, the average distances for all movies, shown in Figure 7.5, can be analyzed prior to the clustering analysis.

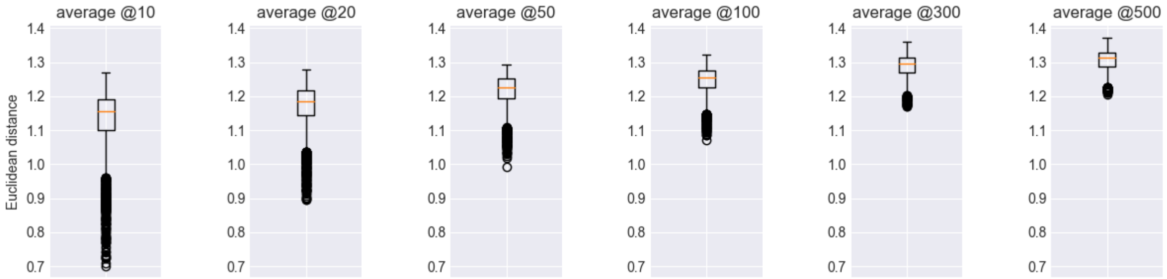


Figure 7.5. Average distances for all considered movies and different top-N configurations.

As can be seen, the movies appear to be quite distant even by selecting the 100 most similar movies, with an average distance around 1.27 which, in cosine distance is equivalent to ∼0.8. These results are expected as it seems unreasonable for the movies to be completely identical; instead, they appear almost orthogonal (at 300 most similar movies an average of 1.3 is equivalent to ∼0.84 in cosine distance).

A visualization of the full item space (with the movies selected to perform LSA) is provided in Figure 7.6, which has been derived after an extensive exploration of different linkage methods, arriving at the conclusion that Ward's method provides a better visualization, after the considerations on the average distances (since with average, single and complete linkages, the movies tend to be at the same height).

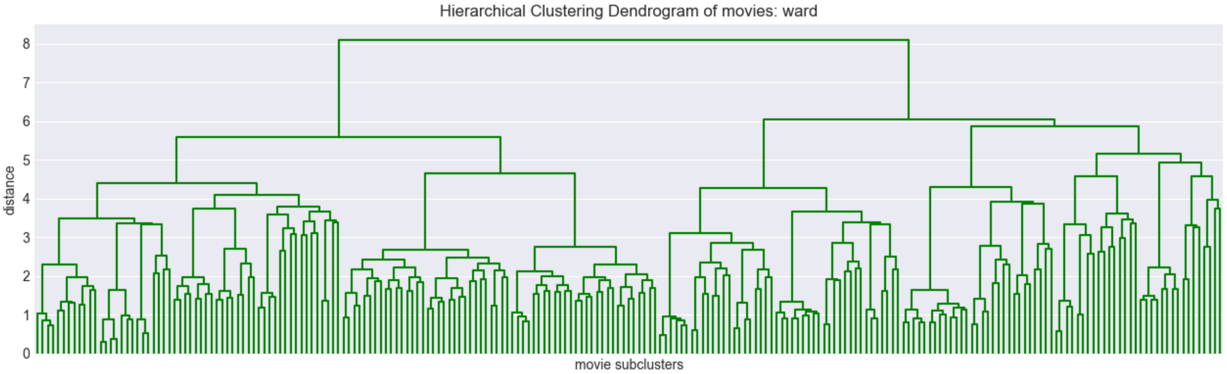


Figure 7.6. Dendrogram after hierarchical clustering (Ward, Euclidean distance) of the subset of movies.

With over 3000 movies, a full hierarchical visualization hardly provides an accurate reading of the estimated number of clusters. For this reason, Figure 7.6 shows the dendrogram at only seven depth levels (the divisions from the root) to provide a better understanding of the item space. Nevertheless, by contracting the dendrogram it is impossible to understand how movies are grouped. A closer inspection is thererfore necessary and has been delivered in Figure 7.7 which focus on two more detailed groupings.

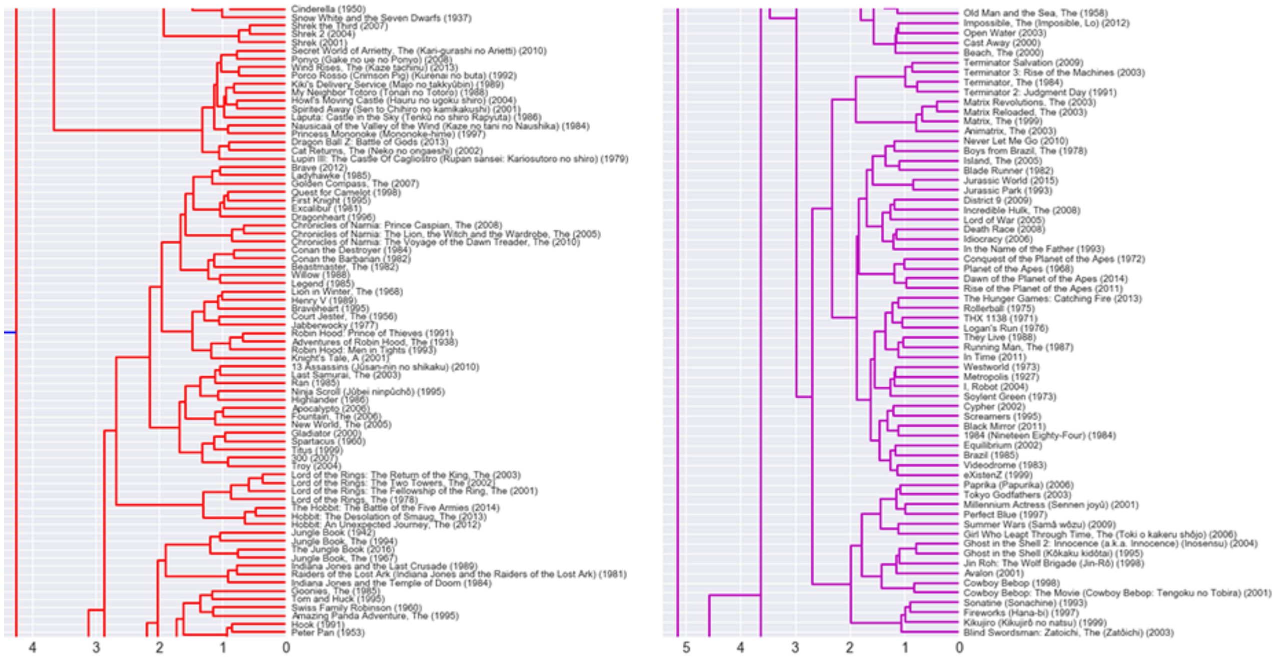


Figure 7.7. Closer inspection of the dendrogram: fantasy (left) and sci-fy (right) movies.

By closely examining the dendrogram, it is possible to formulate an initial hypothesis regarding where clusters can form homogeneous groups.

7.1.3.3 Selection of the optimal number of clusters k

After an extensive exploration of the dendrogram, for which Figure 7.7 only provides a limited perspective, it is expected that more specific clusters can form where the meaningful agglomerations have height (according to the Ward's method [47]) around 3, while more general clusters have height between 3 and 4.
To support this hypothesis, the optimal number of clusters is shown in Figure 7.8, which illustrates how the number of estimated clusters changes according to the height of the dendrogram.

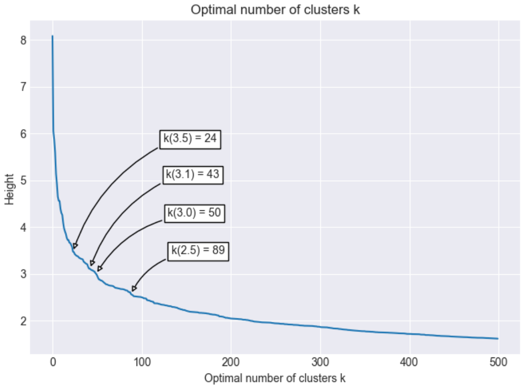


Figure 7.8. Optimal number of clusters given the dendrogram height.

It has been chosen to limit the elbow analysis to up to 500 clusters, since at that point, the height is equivalent to the average distances among the 500 most similar movies, according to Figure 7.5 and the elbow is not visible any longer. In the plot, each k represents the optimal number of clusters at height $3.5, 3, 3.1, 3, 2.5$ respectively, as the result of the agglomeration of sub-clusters. Therefore, each chosen k accounts for high generality or for high specificity depending on the height. Interestingly, at height 3.5, the number of clusters coincides with the number of total IMDb genres, indicating that the clusters may have the same properties as genres. For $k = 43$, the truncated dendrogram becomes as shown in Figure 7.9.

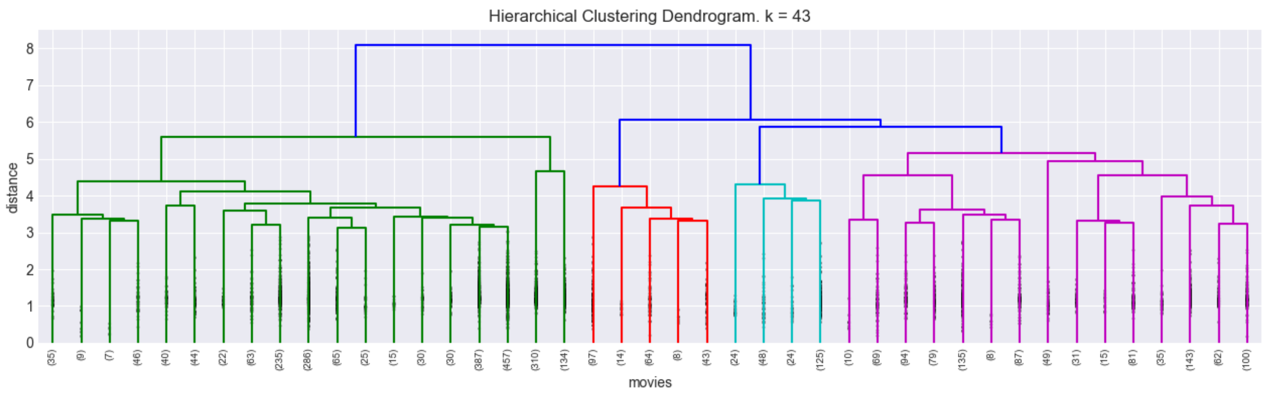


Figure 7.9. Truncated dendrogram for k = 43 clusters.

Unlike in Figure 7.6, here the dendrogram is truncated for the actual number of clusters, regardless of the depth level. The dendrogram shows the distribution of movies for their associated features: a closer inspection has identified that the green clusters have in common features associated to *drama, comedy, documentary, war* and *romance* genres, red clusters include *action, thriller* movies and the remaining clusters are associated to *fantasy, animation, horror and sci-fy* movies. Moreover, what can be noted is the specificity of the clusters: besides some dominant clusters having size greatly above 100 movies, some of the remaining clusters are very specific, with around 10 or fewer movies. In fact, small sized clusters account for movies with many sequels (i.e. *Star Wars, Star Trek, Harry Potter,* etc.). The same happens for greater k, for which the amount of specific clusters increases. Therefore, it is argued that having $k \geq 43$ may result in obtaining very fine grained clusters.

7.1.3.4 Inspection of the retrieved clusters for k = 43

Similarly to how genres have been analyzed (section 6.2.2.2), the same approach can be adopted to inspect the movies included in each cluster, by retrieving the most descriptive features associated. For this purpose, $k = 43$ clusters have been selected after the analysis of the elbow plot (Figure 7.8). Nevertheless, some of the k $(24, 43, 89)$ are used to understand how optimal is the number of clusters for the user profile modeling, user group formation and evaluation of the user diversity.

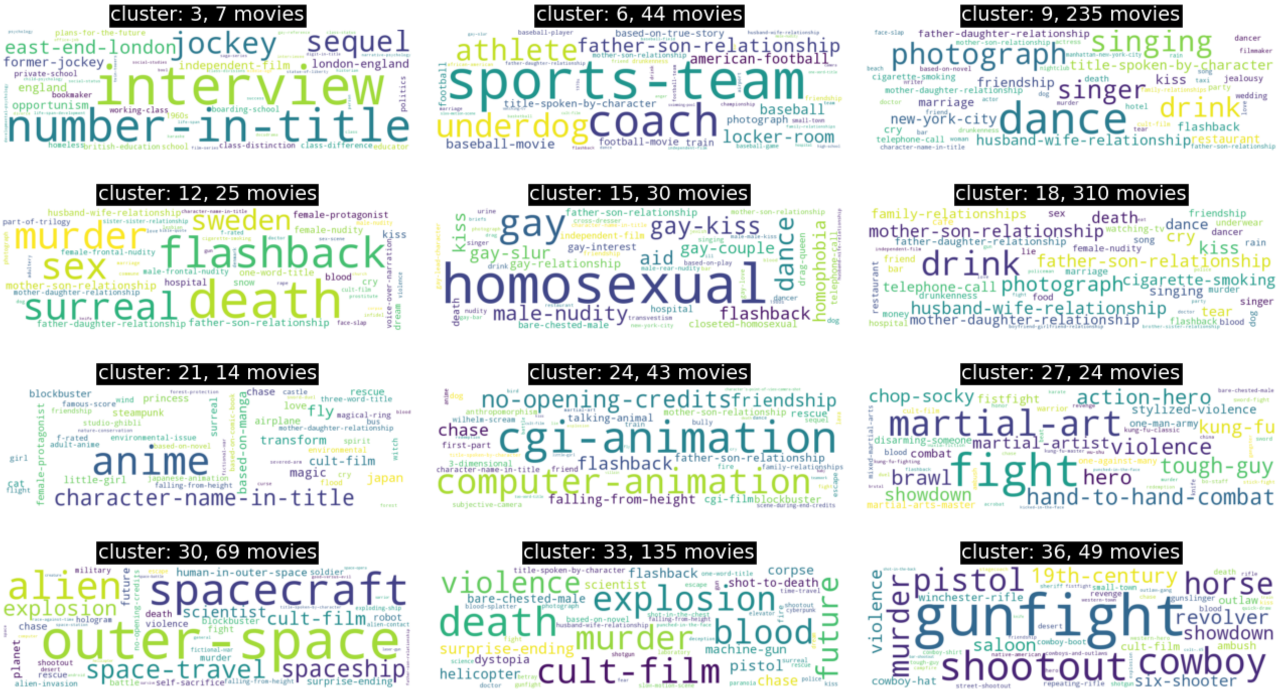


Figure 7.10. Some of the retrieved movie clusters described by keywords.

By examining the associated features (Figure 7.10), in this case the most occurring keywords, it is possible to find that the clusters are ordered following the dendrogram (a complete exploration of the keywords is provided in Appendix A). In particular, also at $k = 47$, some of the retrieved clusters still appear highly specific (here, clusters 3 with 7 movies) and small sized, while few others are highly dominant and coincide with the biased distribution of the movie dataset for *comedy, drama* and *romance*-based movies (Figure 7.11). Nevertheless, similarities among clusters are more easily uncovered through the inspection of genres across clusters (Figure 7.11), making more noticeable both the common themes addressed by the movies in each cluster and a general tendency for clusters to distribute the themes from *drama* to *comedy* and *action* (see Appendix A).

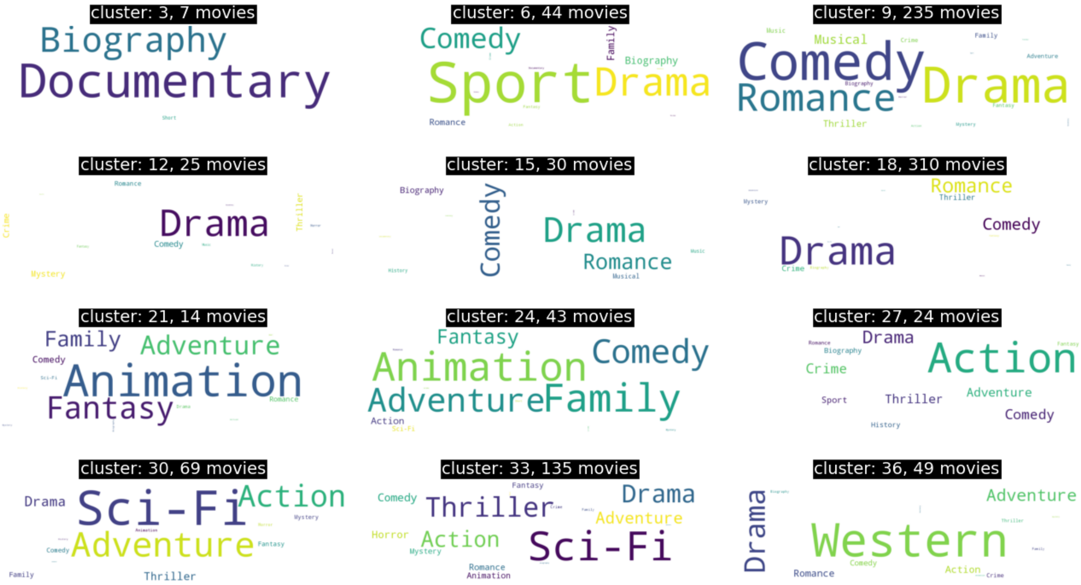


Figure 7.11. Some of the retrieved movie clusters described by genres.

7.2 Category-based preference modeling and user groups formation

In the previous section the item categories have been extracted from a list of popular movies. Here, the same set is used to model the user profiles, disregarding the other movies not sampled for the categorization. In particular, with the popular movies, the number of preserved ratings for all users accounts for 82600, with a loss of ~17% compared to the original 100004 ratings. This aspect is interesting, as it shows that the long tail including the remaining 17% ratings are given to more than half of the movies (~59%). However, it can be argued that the same process can be generalized to include the remaining movies, as initial explorations (Appendix A) with the full item set have concluded that the dendrogram maintains the same structure as in Figure 7.6, thus revealing that while the heights vary, the thresholds can be derived similarly as in Figure 7.8 and be valid for the categorizations of all movies in the set.

Therefore, the user profile modeling and group formation follow the modules as described in sections 5.4.3 and 5.4.4.

7.2.1 User profile modeling

Following the hypothesis that user preferences can be transformed for categories of movies, which is more intuitive than having them for individual items, the profile modeling follows the formula (5.1), exploiting the taxonomies prepared for each k clusters and the rating threshold τ that has been used to extract the positive preferences for each user (section 6.1.3).

Out of the total 82600 ratings, 55416 (~67%) are above the individual rating threshold. With these, the user profiles have been modeled so that each category is weighted by the relative number of items over the total amount of items rated for the given user.

7.2.2 User groups formation

Subsequently, the modeled profiles are CFIUF-weighted, without considering a minimum user frequency threshold as for the item categorization, and finally normalized to unit norm. For the user clustering analysis, the same criteria as item categorization can hold: as there is no prior understanding on the groups of users, hierarchical clustering is adopted to discover the distribution of user similarities by inspecting the distances at which they merge and form super-clusters. Hence, the next sub-sections of the analysis focus on the follwing steps:

1. Analyze the average distances among users as performed in Figure 7.5 to establish a minimum distance threshold (which serves as the maximum similarity between pairs of users) for the analysis of the dendrogram.
2. Inspect the merge heights in the dendrogram and understand which heights form too specific (few users) or too generic (too many) clusters and analyze the height "elbow" point to define the number of user groups.
3. Inspect the user groups for the most occurring categories.

Regarding the size of the groups and following the design of the diversification solution, it is hypothesized that the groups should allow a certain level of heterogeneity. In fact, unlike for the item categorization, the size of user groups is important, as the cluster size implicitly controls the user diversification. Selecting a high k may incur in producing fine grained clusters, without allowing internal heterogeneity among users. On the other hand, selecting a small k may produce highly generic clusters, with the shortcoming of including many diverse users within the same group. In fact, it is reasonable to assume a direct proportionality between the cluster size and the intra-cluster heterogeneity. Therefore, selecting a k so that the cluster size is balanced is the target of this analysis. Nevertheless, given the bias present in the Movielens dataset for particular kinds of movies, it is presumed that the same effect is visible in the sizes of the resulting clusters.

7.2.2.1 Average user pairwise distance analysis

To determine the average minimum similarity between any pair of users from the CFIUF-weighted and normalized profiles, the same process as in Figure 7.5 can be adopted.
Therefore, the average distances are analyzed for each number of item categories as found in section 7.1.

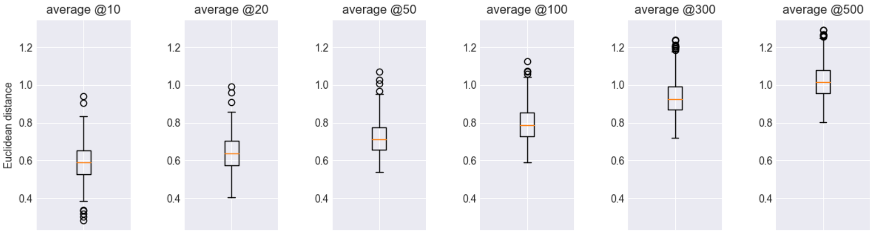


Figure 7.12. Average distances for different topN similar users, on 24 item categories.

As can be seen, the users have an average distance lower than the movies and varies highly depending on the upper distance limit with profiles having either 24 (Figure 7.12) or 43 categories (Figure 7.13).

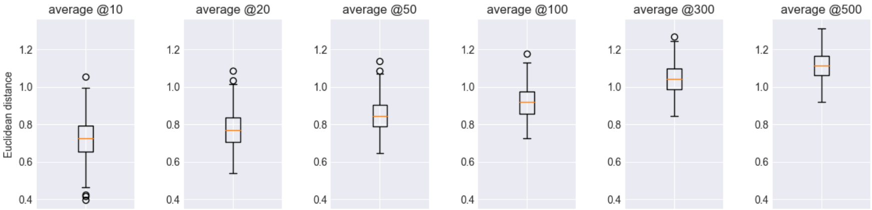


Figure 7.13. Average distances for different topN similar users, on 43 item categories.

It is noted that the Euclidean distance for normalized vectors is bounded in $[0, \sqrt{2}]$, therefore the maximum distance is around 1.41, which represents the maximum cosine distance as well.
Nevertheless, the average distances seem to increase as the user profile dimensionality increases (from 24 to 43 item categories) and this trend is easily visible in Appendix B, suggesting that for each application of hierarchical clustering, different height thresholds should be applied.

A visualization of the full user space is provided similarly to Figure 7.6: Figure 7.14 illustrates the dendrogram resulted by applying hierarchical clustering with Ward linkage on the weighted user profiles modeled with 24 item categories at six depth levels from the root to allow a better visualization of the heights. Hence, the leaves may consist in either single users or merged users (within parenthesis).

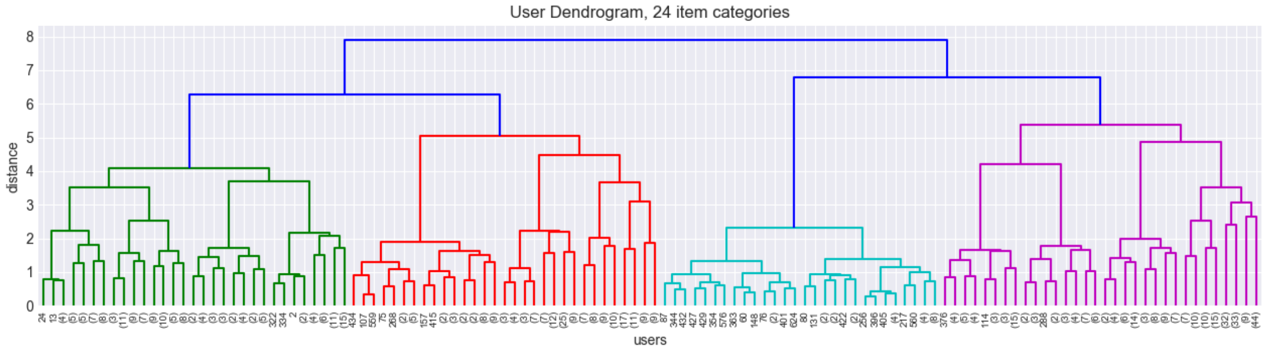


Figure 7.14. Dendrogram of users on k = 24 *item categories.*

For completeness, also the dendrogram with 43 item categories is provided (Figure 7.15) to evaluate the optimal merge height. The dendrogram built on 89 categories can be found in Appendix B, together with the average distance analysis.

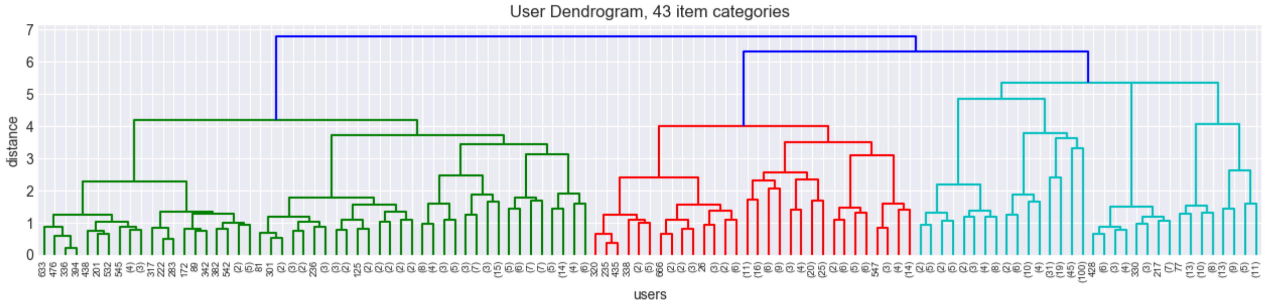


Figure 7.15. Dendrogram of users on k = 43 *item categories.*

As can be noted, the dendrograms appear similar in the groupings at six depth levels (a close inspection reveals that the two dendrograms are almost specular), with minor variations in the merge heights: with 43 categories, the minimum merge height increases, following the interpretation of the average distances (Figure 7.12, Figure 7.13). However, in some cases the successive merges have a lower height. Hence, the merge thresholds should take also this aspect into consideration.
A closer inspection is therefore required to define the threshold at which clusters can considered meaningful, recalling that the goal of this analysis is to find clusters as balaced as possible.

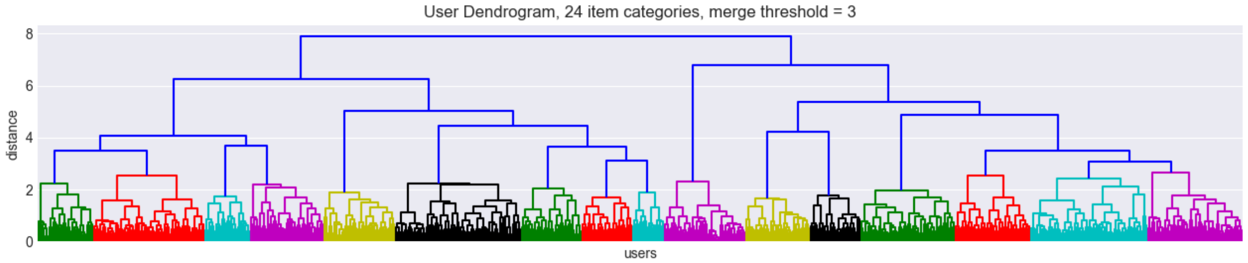


Figure 7.16. Dendrogram with 24 item categories and merge threshold.

Figure 7.16 illustrates the resulting groups if with height threshold 3: while some groups appear quite small, it is argued that overall, the sizes are balanced. In fact, by increasing the threshold by 1 some groups become large, while other groups maintain the original size as with threshold 3. The change in size, however, reflects the popularity bias in the Movielens dataset given by many users having favoured movies from the same categories. Moreover, the heights appear more stable with a higher threshold, threrefore, it seems also reasonable to consider 4 as the merge threshold for the dendrogram built on 25 item categories.

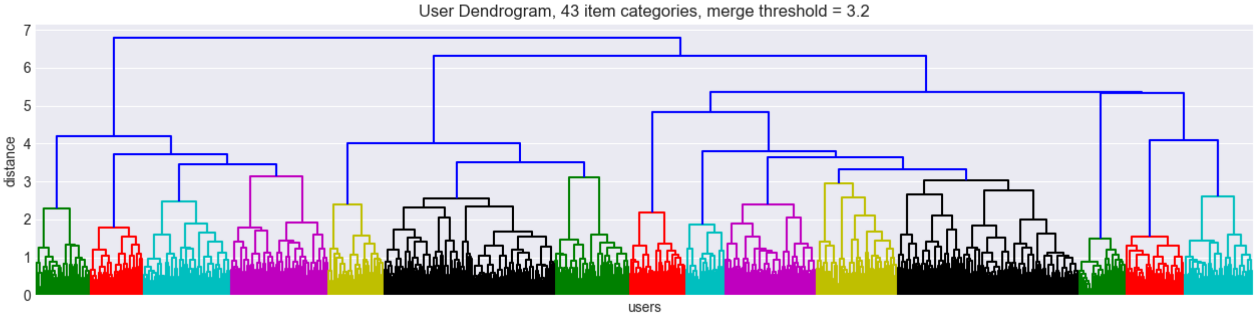


Figure 7.17. Dendrogram with 43 item categories and merge threshold.

The same reasoning can be applied to the dendrogram built on 43 item categories: as can be noted in Figure 7.17, meaningful groups appear already at a low threshold, but are unstable. On the other hand, at thresholds 3 to 4 the heights are more stable, producing both broad and specific clusters.

An extensive exploration of the merge heights also for the other dendrograms (see Appendix B) has concluded that balanced clusters form already at small heights, around 2. With larger heights, the resulting groups tend to have uneven sizes which conform to the popularity bias.

With an inspection of the merge heights, it has been possible to formulate the hypotheses regarding the number of balanced clusters which allow user heterogeneity, taking into consideration the popularity bias.

7.2.2.2 Selection of the optimal number of user clusters k

The selection of the optimal k has been determined from the merge thresholds at which the heights remain stable and form balanced clusters, neither fine grained nor broad. At this stage, however, it is beneficial to evaluate the hypotheses following the height elbows, shown in Figure 7.18 for the dendrograms built on different item categorizations places the optimal number of clusters below 100. Also, since the profile dimensionality slightly influences the merge heights, the differences are noticeable in the elbows, which follow the same trend nonetheless.

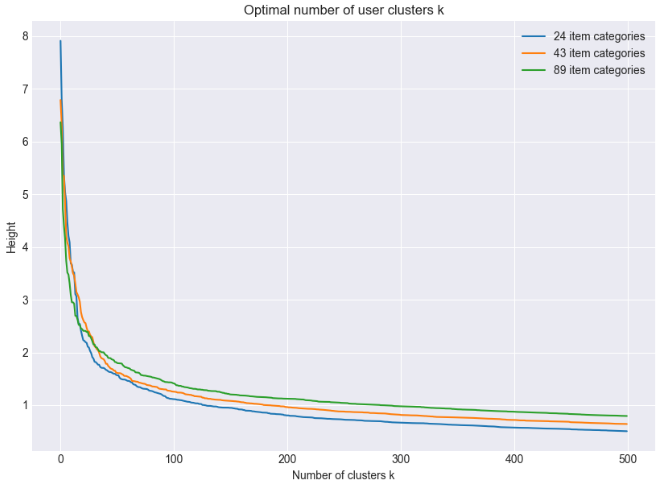


Figure 7.18. Elbow plot of the optimal number k of user clusters for all item categories.

In particular, the differences are more noticeable in Figure 7.19, showing up to $k = 50$ clusters. The dendrogram of profiles built on 24 categories shows an elbow slightly greater than the remaining ones, which instead have an increasing acceleration and form the same number of clusters at a lower merge height.

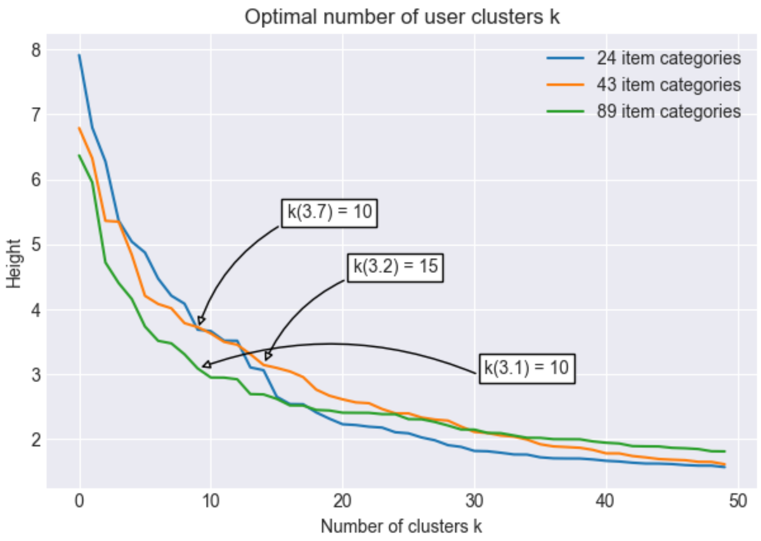


Figure 7.19. Elbow plot of the optimal number k of user clusters for all item categories up to 80 clusters.

In fact, the elbows are more visible at $k < 50$, although the visual inspection of the merge threshold suggests to keep a low number of clusters, around 3.7, for the users built on 24 item categories, and around ~3.2 for the others (see Appendix B).

It is interesting to notice that the for different thresholds, the optimal number of clusters does not vary for users profiles built on 24 and 89 item categories, even though the user preferences granularity differs. On the other hand, the profiles built on 43 categories seem to yield more clusters at the optimal merge threshold (~3.2). Nevertheless, a small number of clusters is expected, given the dimensions of the Movielens dataset: in fact, it is argued that the thresholds and therefore, the optimal number of clusters may vary for available larger datasets.

For $k = 15$, the truncated dendrogram is shown in Figure 7.20.

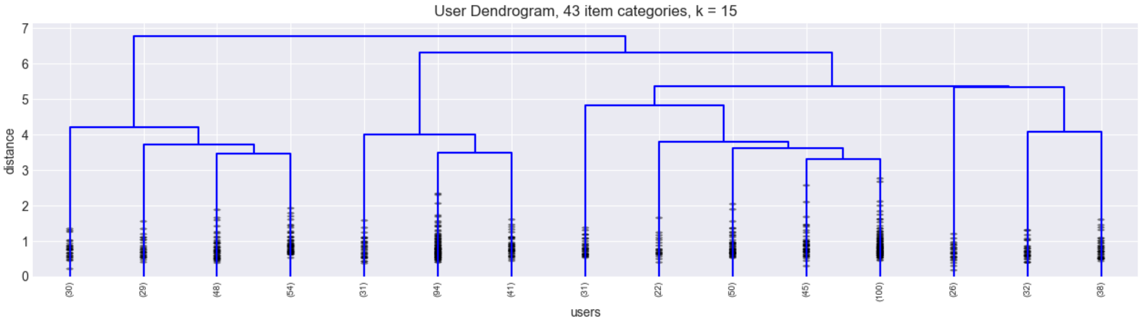


Figure 7.20. Truncated user dendrogram, k = 15 and 43 item categories.

Unsurprisingly, the bias in Movielens is present, as one particularly large group (100 users) is formed. The restant groups are generally smaller but arguably heterogeneous enough to allow meaningful intra-cluster differences for the diversification purpose: in fact, Most of the users form evenly distributed agglomerations between height 1 and 2 (as shown by the black points).

7.2.2.3 Inspection of the retrieved user groups for k = 15 and 43 item categories

Again, replicating the same inspection of the item categories, the groups of users can be analyzed for their descriptive features associated. The inspection is focused on the clusters of user profiles built on 43 categories, as they represent a compromise of between fine grained and generic user preferences, an extensive exploration is however available in Appendix B.

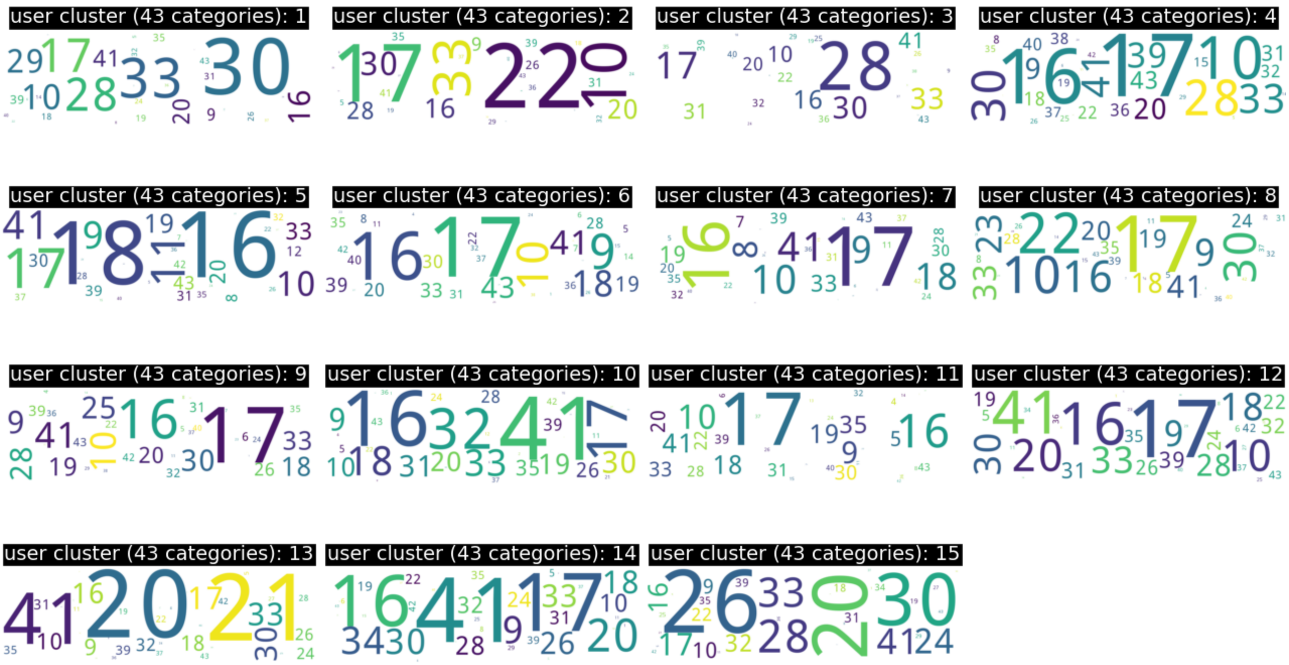


Figure 7.21. Description of the clusters of users, using the item categories as features.

While Figure 7.21 does not provide a detailed description of the clusters in terms of the metadata as for the inspection of item categories, some patterns can be uncovered also by analysing the categories *per se*. In particular, this analysis follows the types of categories described in Appendix A. As can be seen, all clusters have in common the item category 17, which coincides with the largest item category (~400) containing *romance, drama, comedy* movies. This bias was expected and supports the fact the Movielens dataset is biased towards movies of those genres. A more in depth analysis permits to find some clusters of users where some categories seem to be correlated: for example, in group 13, the most common item categories are associated to *animation* and *mystery* movies, as well as containing categories related to *sci-fy* movies in smaller portions. Groups 1 and 2 are also similar as they share *animation* and *sci-fy* related categories. Clusters 3 and 4 instead share categories related to *thriller* and *action* movies. *Drama* movies are more popular in groups 5, 6, 7 and 8, as the first 19 item categories are related to that genre, to a certain extent. The other groups appear different, focusing on *sci-fy*, *animation* and other genres.

7.3 Diversification evaluation of the distant neighborhoods

This section focuses on the actual evaluation of the proof of concept built on the item categories and on the found user groups. Nevertheless, in order to understand the diversification reached through the user clustering and distant neighborhood formation, the experimental evaluation requires the setup explained in the next section.

The aim of this section is to understand to what extent the proposed approach answers the following hypotheses:

- The approach can tune the diversification among users within the same clusters and therefore, control the trade off between user similarity and distance.
- The approach performs similarly to baseline diversification aproaches, such as random diversification.

7.3.1 Evaluation procedure of the distant neighborhood formation

The user dataset from Movielens requires a training phase and a test phase for the offline evaluation.

1. In the training phase, the users undergo the clustering process for each of the item categories with the optimal number of groups as already discovered in section 7.2.2.2.

2. Next, the test user profiles are classified on the clusters created at the training phase, by performing kNN on the closest 11 users to determine the suitable cluster.

3. Lastly, for each of the classified test users, the distant neighborhood is formed for different values of α, with the formula (5.2) on the cosine distances to find the significance scores. Also, different numbers of neighbors are retrieved according to (5.3), which allows to control the trade-off between similarity and distance between the active user and any other user in the same cluster. The cosine distance is utilized as it can be easily integrated in the significance score formula, which requires both similarities and distances and more importantly, since the Euclidean distance does not have an opposite metric to calculate the similarities.

4. Finally, the proposed diversification approach, for now limited to the extraction of distant users, is evaluated against random user diversification, used as a baseline method using the Intra Set Similarity (ISS), as introduced in [35] to evaluate the diversity of users within a set of neighbors, as the chosen metric. As a future experimental evaluation, it will be possible to reach the actual recommendations and fully evaluate the proposed framework, by employing the diversification metrics defined in section 4.3.

7.3.2 Expected outcomes of the evaluation

Following the hypotheses, it is expected that the formula (5.2) will allow to control the user diversity considering that the evaluation at this stage allows only to measure the diversity on the distant neighbors and to reach actual recommendations, different evaluation procedures are required. In particular, it is expected that the yielded diversity of the distant neighbors to the test users will be directly proportional to α.

It is also expected that since the distant neighbors are selected from the same cluster, the resulting diversity will not be as high as with random neighborhood formation (e.g. with no criteria in selecting the significant users to the active one).

7.3.3 Evaluation procedure setup

As introduced in the previous section, the user dataset is splitted so that 80% of the users are used to train the algorithm, learn the user preference categories and form the clusters of users. The remaining 20% of users are kept for the testing phase.

Since there are three datasets for which the user clusters have been derived, the experimental procedure can focus on the user profiles built on 43 item categories.

Regarding the diversification parameter α, the behaviour of the proposed approach is studied for values: $\alpha = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. The values range from no diversification (traditional similarity based neighbor formation, therefore expected low heterogeneity) to varying diversification (expected low to high heterogeneity), and full diversification (where the farthest users within the group are selected, therefore, expected high heterogeneity), respectively.

Moreover, for each α, the size of the distant neighborhood retrieved is $k = \{5, 10, 15\}$, following the assumption that this number is reasonable given the sizes of the groups of users, which contain at least 22 users, for the clusters built on 43 categories of items (Figure 7.20), which may be also smaller for the clusters yielded from the training users.

For the evaluation metric, it is proposed to use Intra List Similarity as the diversification metric to provide an initial understanding of the diversification based on the significance of the distant users. In particular, the ISS metric is derived from [35] and adapted to consider the dissimilarity between pairs of users, instead of similarity. Henceforth, the ISS metric is regarded as Intra Set Diversity metric (ISD), which is directly proportional to the pairwise dissimilarity between the users.
The following section provides the empirical results of the evaluation and the subsequent analysis.

7.3.4 Empirical results and analysis of the user diversification approach

The results of the experiment on a single instance of user profiles (built on 43 item categories) can be inspected in Figure 7.22. The ISD metric has been chosen instead of ILS for the possibility of comparing the diversification levels for different neighborhoods with increasing size: in fact, an initial experimentation with ILS has concluded that this metric increases as the neighborhood is larger, therefore does not allow an objective comparison of the diversification levels.

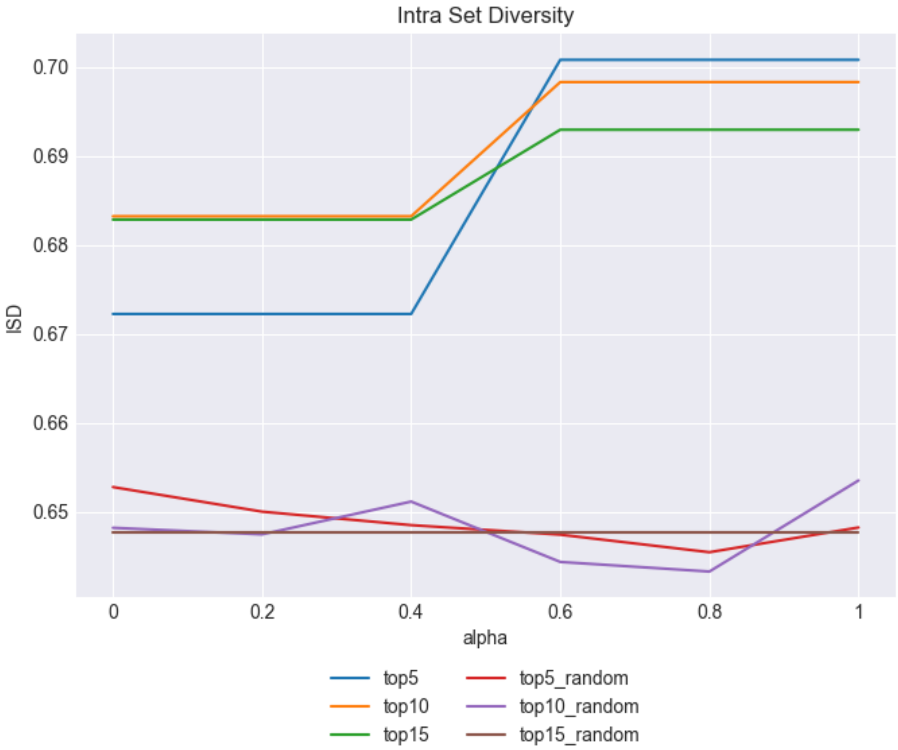


Figure 7.22. Intra Set Diversity for different neighborhoods (topN retrieved within the same cluster and topN randomly extracted from the test set) and user profiles built on 43 item categories.

The x-axis represents the α values for which the user diversification has been conducted. On the y-axis, the resulting ISD scores are produced and represent the overall diversity of the users.
Each curve has been labeled considering the significant neighbors for varying neighborhood sizes (5, 10, 15), together with the baseline random diversified users (again using the same neighborhood sizes), which serve as an anchoring measure for the analysis of the approach.

A general trend can be noted for the distant neighbors extracted from the same groups: the first three curves show a crescent trend from low to high diversity, which tend to smoothen as the neighborhood size increases. Moreover, the ISD for the small neighborhood (not random) follows an intuitive trend from generally low to high diversity, even if the scale is not as high as expected (ISD ranges between 0.67 and 0.70 for the blue curve. Surprisingly, the diversification levels do not chance as expected: instead of following a smooth curve, the changes in diversity are moderately sudden, especially for the small neighborhood (in blue). In particular, there seems to be a rift between $\alpha = 0.4$ and $\alpha = 0.6$ which causes this trend, as with lower and smaller α, the ISD does not capture any other variation in the neighbor diversity.

On the other hand, the α-diversification on random users appears more erratic but with smaller variations in the diversity, with contradictory results as the diversification level increases, producing an initially descending ISD, which ultimately increases when at full diversification ($\alpha = 1$).

7.3.4.1 Analysis of the results on the α-diversification levels

The results of this experiments may suggest to increase the neighborhood size, as there are no substantial divergences below 15 most significant neighbors. Nevertheless, the hypothesis concerning the proportionality between the diversification levels and the diversity reached has been confirmed, even if with limitations on the validation (the experiment on the other user profiles has not been performed at this stage), suggesting that the proposed user within-cluster diversification may be worth of consideration in the next stage of this work. Moreover, the hypotheses on the increasing diversification levels for distant neighbors within the same cluster are valid to the extent of the maximum dissimilarity between any pair of users in the same cluster: in fact, as the users within the same groups share a theoretical baseline similarity, it would be unreasonable to expect a constant increment of the distant neighbors diversification, also taking the group sizes in consideration.

Regarding the hypothesis on the levels of diversification reached through ISD, it can be concluded that the results are not as expected, as the ISD metric favours the distant neighbors, which have a greater overall diversity than the random users.

8 Conclusions

The last chapter considers the last experimental evaluation as the method to answer the research questions stated in the beginning of this work. The previous chapters have been focused on the examination of the problem related to the RS field, with the insights offered by the psychological understanding on the diversification concept, along with how it has been addressed by the research community.
A user-centric framework of diversity has been presented following the analysis of the psychological meaning of diversity (hence the meaning as user-centric) as the primary contribution of this work and is argued that it could be efficiently adopted as a model to evaluate recommender systems for the diversity related to individual users, rather than the diversity of the recommender system *per se*.
The framework has been adopted in the analysis of current approaches to diversification, in particular, to assess their shortcomings and focuses: it has been found that most of the works, compared to the framework, do not take into consideration the full dimensionality of the diversity (in terms of the identified properties on which diversification can be tuned: coverage, redundancy and novelty).
In turn, the analysis has been of primary importance to design a novel recommendation approach, directly embedding the diversification into the recommendation filtering. The proposed approach has been modularized and the development of a proof of concept has served to answer the research questions.

8.1 Discussions on the problem formulation

The proposed approach has been designed as a possible solution to mitigate the filter bubble problem for individual users. Still, the existence of the filter bubble itself has to be proven on the first place.

Nevertheless, by looking at the primary cause of filter bubble, over-specialization, a novel user-centric diversification framework has been proposed, taking into consideration the aspects which may be important to evaluate against individual preferences and regarding the user satisfaction (local and global coverage, novelty and redundancy).
Moreover, the insights gained from the analysis of relevant works in sociology, preference and choice psychology and decision making psychology have been a fundamental requirement for the creation of the diversification framework, which tries at its best to encode the motivations for incorporating diversity in recommender systems, even without requiring real user validations.
Therefore, to answer the first research question, the findings from psychology and sociology greatly surpass the constraints of this work, which has been limited to considering internal motivations for the diversification of recommendations: there are still many other variables that the proposed framework has to address, as the diversity is highly dependant on contextual variables (mood, location, company, etc.) as well as on internal factors and changes in the preferences [56]; the proposed framework only considers the internal motivations for diversity, such as the desire for the unfamiliar (novelty) against the monotony of similar recommendations, the individual preference magnitudes as weights for the recommendations as well as for the varieties in types of recommendations.

On modeling of the user interests (second research question), the framework has been devised to transform user preferences into category based ones, without losing the relationships between users and items. The benefits of this operation, related to diversity, lie in the generality and on an easier understanding of the resulting preferences, which are not expressed on individual items, but on classes of items sharing relevant attributes. As such, the preferences encoded into categories allowed to mitigate the rating sparsity and to find more easily the similarities (and differences) among users.

While this work has provided a limited analysis of related works, it still has to address completely the research field, by considering state of the art diversification frameworks and other solutions, in order to validate the proposed framework. Nevertheless, a diversity by design recommender system has been designed following the principles of the framework, so that the diversification can be tunable and the recommendations can be compared to the active user preferences.

As the main contribution after the proposed framework, the recommendation procedure has been designed to take in consideration the diversification prior to the extraction of the items, which has been argued to be more flexible than more popular post-filtering diversification approaches: the proposed approach considers the diversification framework to find distant users from which to extract novel types of items for the active user, which may increase the chances that the user will get interested in unfamiliar categories, as well as to mitigate the over-specialization tendency through a “diverse” approach to personalization.
In fact, a diversified personalization goes against the collaborative filtering rationale that the most useful recommendations come from similar users: instead, it is argued that recommendations may be more relevant when users have different (but not contradictory) preferences.
However, since over-specialization is considered as a dynamic effect of the recommendations (a positive feedback loop), the validity of the designed recommendation system still needs clearer answers through real user testing.
Nevertheless, the third research question can be answered on a theoretical level, considering that the proof of concept has been evaluated only on the diversification level reached from the distant neighborhood formation. For this question, diversity has been incorporated following the analysis of related work on the principles of the proposed framework: it is expected that the designed approach may retrieve interesting users from the tunable trade off in the distant neighbors extraction, from which it is definitely possible to find novel categories as a mean against over-specialization. Nevertheless, the proposed approach requires a necessary augmentation when users have experienced all categories, in which case there would be no more novel categories: a proposed solution therefore would entail in controlling the preference scores for the categories both covered by the active user and by distant neighbors.

Lastly, to completely answer the second question (regarding user profile modeling), the proposed approach is still limited in encoding the user interests through item categories, as a more in depth analysis and a better item categorization is required. Moreover, at this stage the user preferences merely consist of the occurrences of item categories, while a better modeling would be interesting to pursue (e.g., by modeling the occurrences along with the ratings for each item), hence, this question requires additional investigations to find a suitable user interest modeling procedure.
Nevertheless, with the user profiles modeled through simple occurrences of favourite item categories, the proposed approach has been evaluated for the level of diversity yielded by extractin significant neighbors for active users within the same groups.

Finally, the proposed approach has been evaluated on a proof of concept to answer the last research question, regarding the influence of the distant within cluster neighbors on the overall Intra Set (as in user neighborhood) Diversity, by tuning the external parameter α. From the experiment, it has been possible to evaluate positively the results concerning the expected diversification levels reached from the selection of distant users. Nevertheless, as the experimental evaluation has been limited to only one instance of user modeled profiles, still it requires a generalization to other profiles built on greater item categories.

8.2 Future work

The proposed approach represents one of the many possible implementations of diversity within the recommendation filtering process. By employing the proposed framework, it is suggested to validate its applicability through real users experiments (e.g. A/B testing), which is also required for the framework to effectively be called user-centric, since for now, it has only been applied on the Movielens dataset with static users.

Nevertheless, as discussed in the delimitations and in the previous section, the proposed framework should be extended by covering additional aspects of diversity, such as contextual variables and other external factors, for which the Movielens dataset has also been limited in the provision: with only the timestamps of the ratings, there are not many conclusive answers regarding the contextual variables which brought to the choice of a given movie and, arguably, as the Movielens dataset is focused on explicit ratings, it would be more interesting to analyze other datasets with implicit ratings, such as retention time when watching a movie, which may encode more information than a simple rating.

Regarding the technical implementation of the approach, it is also proposed to utilize an unbiased dataset of movies to generate the item categories, as in this thesis, the analysis and the validation of the proposed approach was limited to the movies extracted from Movielens. It has been discussed how the exploratory analysis of the extracted movies offered insights of general tastes of the users and for this reason, the movie dataset has been deemed biased towards particular kinds of movies. In order to mitigate this issue, a feature extraction methodology has been pursued, however, there is still a need to provide an objective categorization of movies based on their content. Perhaps, an extensive extraction of metadata from IMDb might be suitable for this task.

Moreover, another type of item categorization can be pursued to group the movies according to specific types of metadata (e.g., clustering according to the cast, directors and writers, according to keywords and genres, according to decade and language, etc.), since a naïve clustering approach with all metadata has the disadvantage to not encode user preferences that are based on precise metadata, such as some actors or actresses, or a paricular composer, etc. In fact, it is not uncommon for people to be conditioned by actors or actresses, especially if they are celebrities. As a future development, a better item categorization and encoding of the user preferences would take into consideration this particular aspect.

One limitation of the approach is that the preprocessing modules also require an item classification module, necessary when adding new items and the clusters have been already computed, which will be addressed in the future.
Another limitation of the proposed approach regards the fact that it is currently not able to evaluate the actual volume of each item category. Since the categories found after the clustering analysis are evidently unbalanced, the proposed approach would address this issue as a future development, to consider the baseline volume of all categories into the recommendation procedure and, in particular, in the extraction of unseen movies for the active user.

References

[1] P. Resnick and H. R. Varian, “Recommender systems,” *Commun. ACM*, vol. 40, no. 3, pp. 56–58, Mar. 1997.

[2] G. Häubl and V. Trifts, “Consumer Decision Making in Online Shopping Environments: The Effects of Interactive Decision Aids,” *Mark. Sci.*, vol. 19, pp. 4–21.

[3] L.-L. Wu, Y.-J. Joung, and J. Lee, “Recommendation Systems and Consumer Satisfaction Online: Moderating Effects of Consumer Product Awareness,” in *2013 46th Hawaii International Conference on System Sciences*, 2013, pp. 2753–2762.

[4] N. E. Bingham, “Toffler, Alvin. Future shock. New York: Bantam Books, Inc., 1971 (540 pages),” *Sci. Educ.*, vol. 56, no. 3, pp. 438–440, Jul. 1972.

[5] C. Speier, J. S. Valacich, and I. Vessey, “The Influence of Task Interruption on Individual Decision Making: An Information Overload Perspective,” *Decis. Sci.*, vol. 30, no. 2, pp. 337–360, Mar. 1999.

[6] B. Schwartz, *The paradox of choice: Why more is less.* New York, NY, US: HarperCollins Publishers, 204AD.

[7] M. Piaseki and S. Hanna, “A redefinition of the paradox of choice,” in *4th Design Computing and Cognition Conference - DCC’10*, 2010, pp. 347–366.

[8] B. Hallinan and T. Striphas, “Recommended for you: The Netflix Prize and the production of algorithmic culture,” *New Media Soc.*, vol. 18, no. 1, pp. 117–137, 2016.

[9] E. Pariser, “The Filter Bubble: What the Internet Is Hiding from You,” *ZNet*, p. 304, 2011.

[10] Q. V. Liao and W.-T. Fu, “Beyond the filter bubble,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI ’13*, 2013, p. 2359.

[11] E. Pariser, “Eli Pariser: Beware online &quot;filter bubbles&quot; | TED Talk Subtitles and Transcript | TED.com,” *TED Talks*, 2011. [Online]. Available: https://www.ted.com/talks/eli_pariser_beware_online_filter_bubbles/transcript?language=en.

[12] T. T. Nguyen, P.-M. Hui, F. M. Harper, L. Terveen, and J. A. Konstan, “Exploring the filter bubble,” in *Proceedings of the 23rd international conference on World wide web - WWW ’14*, 2014, pp. 677–686.

[13] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[14] M. Kaminskas and D. Bridge, “Diversity, Serendipity, Novelty, and Coverage,” *ACM Trans. Interact. Intell. Syst.*, vol. 7, no. 1, pp. 1–42, Dec. 2016.

[15] C.-N. C. N. Ziegler, S. M. S. M. McNee, J. a. J. a. Konstan, and G. Lausen, “Improving recommendation lists through topic diversification,” *Proc. 14th Int. Conf. World Wide Web WWW 05*, no. January, p. 22, 2005.

[16] S. M. McNee, J. Riedl, and J. A. Konstan, “Being accurate is not enough,” in *CHI ’06 extended abstracts on Human factors in computing systems - CHI EA ’06*, 2006, p. 1097.

[17] P. Castells, N. J. Hurley, and S. Vargas, “Novelty and Diversity in Recommender Systems,” in *Recommender Systems Handbook*, Boston, MA: Springer US, 2015, pp. 881–918.

[18] B.-U. A. ; H. B. ; H. C. ; R. A. Carrillo, “XploDiv: Diversification Approach for Recommender Systems,” Jan. 2015.

[19] M. C. Willemsen, M. P. Graus, and B. P. Knijnenburg, “Understanding the role of latent feature diversification on choice difficulty and satisfaction,” *User Model. User-adapt. Interact.*, vol. 26, no. 4, pp. 347–389, Oct. 2016.

[20] F. M. Harper and J. A. Konstan, “The MovieLens Datasets: History and Context,” *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19:1--19:19, 2015.

[21] “IMDb - Movies, TV and Celebrities - IMDb.” [Online]. Available: http://www.imdb.com/. [Accessed: 20-Jul-2017].

[22] B. E. Kahn, “Consumer variety-seeking among goods and services,” *J. Retail. Consum. Serv.*, vol. 2, no. 3, pp. 139–148, Jul. 1995.

[23] A. Stirling, “A general framework for analysing diversity in science, technology and society,” *J. R. Soc. Interface*, vol. 4, no. 15, 2007.

[24] R. Burke, “Hybrid Web Recommender Systems,” in *The Adaptive Web*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 377–408.

[25] F. Ricci, L. Rokach, and B. Shapira, “Recommender Systems: Introduction and Challenges,” in

Recommender Systems Handbook, Boston, MA: Springer US, 2015, pp. 1–34.

[26] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, “Collaborative Filtering Recommender Systems,” *Found. Trends® Human–Computer Interact.*, vol. 4, no. 2, pp. 81–173, 2011.

[27] K. Nehring and C. Puppe, “A Theory of Diversity,” *Econometrica*, vol. 70, no. 3, pp. 1155–1198, May 2002.

[28] K. JUNGE, “Diversity of ideas about diversity measurement,” *Scand. J. Psychol.*, vol. 35, no. 1, pp. 16–26, Mar. 1994.

[29] G. Adomavicius and Y. O. Kwon, “Toward more diverse recommendations: Item re-ranking methods for recommender systems.” Social Science Research Network, 2009.

[30] G. Adomavicius and A. Tuzhilin, “Context-Aware Recommender Systems,” in *Recommender Systems Handbook*, Boston, MA: Springer US, 2015, pp. 191–226.

[31] N. Lathia, S. Hailes, L. Capra, and X. Amatriain, “Temporal diversity in recommender systems,” in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’10*, 2010, p. 210.

[32] M. Kunaver and T. Požrl, “Diversity in recommender systems – A survey,” *Knowledge-Based Syst.*, vol. 123, pp. 154–162, 2017.

[33] S. Vargas, L. Baltrunas, A. Karatzoglou, and P. Castells, “Coverage, redundancy and size-awareness in genre diversity for recommender systems,” in *Proceedings of the 8th ACM Conference on Recommender systems - RecSys ’14*, 2014, pp. 209–216.

[34] T. Aytekin and M. Ö. Karakaya, “Clustering-based diversity improvement in top-N recommendation,” *J. Intell. Inf. Syst.*, vol. 42, no. 1, pp. 1–18, Feb. 2014.

[35] C. Yang, C. C. Ai, and R. F. Li, “Neighbor Diversification-Based Collaborative Filtering for Improving Recommendation Lists,” in *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, 2013, pp. 1658–1664.

[36] M. Ge, C. Delgado-Battenfeld, and D. Jannach, “Beyond accuracy,” in *Proceedings of the fourth ACM conference on Recommender systems - RecSys ’10*, 2010, p. 257.

[37] S. Deerwester, S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *J. Am. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391--407, 1990.

[38] D. Sarkar, “Text Classification,” in *Text Analytics with Python*, Berkeley, CA: Apress, 2016, pp. 167–215.

[39] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Application of Dimensionality Reduction in Recommender System - A Case Study.” 2000.

[40] S.-H. Cha and S.-H. Cha, “Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions,” *Int. J. Math. Model. METHODS Appl. Sci.*, 2007.

[41] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[42] S. D. Sondur, S. Nayak, and A. P. Chigadani, “Similarity Measures for Recommender Systems: A Comparative Study,” *Int. J. Sci. Res. Dev.*, vol. 2, no. 3, pp. 76–80, Jun. 2016.

[43] T. Korenius, J. Laurikkala, and M. Juhola, “On principal component analysis, cosine and Euclidean measures in information retrieval,” *Inf. Sci. (Ny).*, vol. 177, no. 22, pp. 4893–4905, Nov. 2007.

[44] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT Press, 1999.

[45] S. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[46] D. Arthur, D. Arthur, and S. Vassilvitskii, “K-means++: the advantages of careful seeding,” *Proc. 18TH Annu. ACM-SIAM Symp. Discret. ALGORITHMS*, 2007.

[47] D. Müllner, “Modern hierarchical, agglomerative clustering algorithms,” Sep. 2011.

[48] Mathworks, “Applying Supervised Learning,” *What is Machine Learning*. Mathworks, p. 20, 2016.

[49] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, “Understanding of Internal Clustering Validation Measures,” in *2010 IEEE International Conference on Data Mining*, 2010, pp. 911–916.

[50] T. Calinski and J. Harabasz, “A dendrite method for cluster analysis,” *Commun. Stat. - Theory Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[51] B. Schwartz, A. Ward, J. Monterosso, S. Lyubomirsky, K. White, and D. R. Lehman, “Maximizing versus satisficing: happiness is a matter of choice.,” *J. Pers. Soc. Psychol.*, vol. 83, no. 5, pp. 1178–97, Nov. 2002.

[52] J. Berger, M. Draganska, and I. Simonson, “The Influence of Product Variety on Brand Perception and Choice,” *Mark. Sci.*, vol. 26, no. 4, pp. 460–472, Jul. 2007.

[53] D. McSherry, “Diversity-Conscious Retrieval,” Springer, Berlin, Heidelberg, 2002, pp. 219–233.

[54] L. Chen, W. Wu, and L. He, “How personality influences users’ needs for recommendation diversity?,” in *CHI ’13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA ’13*, 2013, p. 829.

[55] L. Chen, W. Wu, and L. He, “Personality and Recommendation Diversity,” Springer, Cham, 2016, pp. 201–225.

[56] L. McAlister and E. Pessemier, “Variety Seeking Behavior: An Interdisciplinary Review,” *J. Consum. Res.*, vol. 9, pp. 311–322.

[57] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, Mar. 2014.

[58] T. Murakami, K. Mori, and R. Orihara, “Metrics for Evaluating the Serendipity of Recommendation Lists,” in *New Frontiers in Artificial Intelligence*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 40–46.

[59] B. Smyth and P. McClave, “Similarity vs. Diversity,” Springer, Berlin, Heidelberg, 2001, pp. 347–361.

[60] B. M. Sarwar, J. A. Konstan, A. Borchers, J. Herlocker, B. Miller, and J. Riedl, “Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system,” in *Proceedings of the 1998 ACM conference on Computer supported cooperative work - CSCW ’98*, 1998, pp. 345–354.

[61] J. Ben Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative Filtering Recommender Systems,” in *The Adaptive Web*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 291–324.

[62] F. Cena, C. Gena, P. Grillo, T. Kuflik, F. Vernero, and A. J. Wecker, “How scales influence user rating behaviour in recommender systems,” *Behav. Inf. Technol.*, pp. 1–20, May 2017.

[63] T. Hofmann and Thomas, “Latent semantic models for collaborative filtering,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 89–115, Jan. 2004.

[64] G. Adomavicius and YoungOk Kwon, “Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 896–911, May 2012.

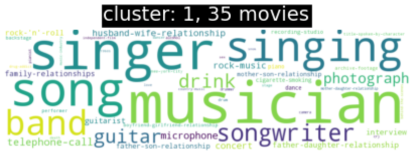
[65] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, “Feature-Weighted User Model for Recommender Systems,” in *User Modeling 2007*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 97–106.

[66] IMDb, “Submission Guide: Keywords.” [Online]. Available: http://www.imdb.com/help/search?domain=helpdesk_faq&index=2&file=keywords. [Accessed: 14-Sep-2017].

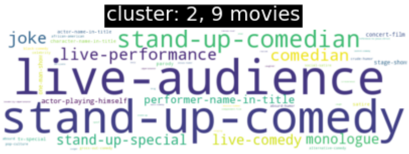
Appendix A

Here a complete overview of the item categories for $k = 43$ clusters is provided. The clusters are described by the most frequent keywords within each one.

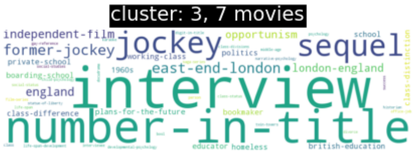
cluster: 1, 35 movies



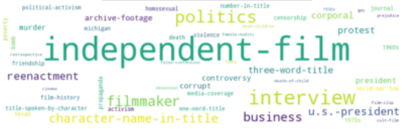
cluster: 2, 9 movies



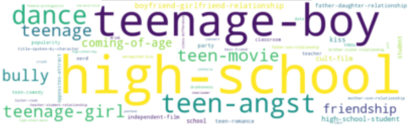
cluster: 3, 7 movies



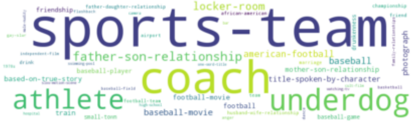
cluster: 4, 46 movies



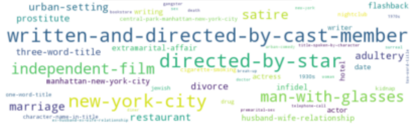
cluster: 5, 40 movies



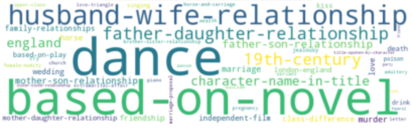
cluster: 6, 44 movies



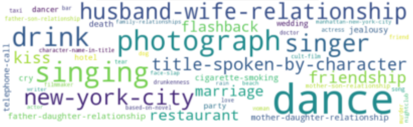
cluster: 7, 22 movies



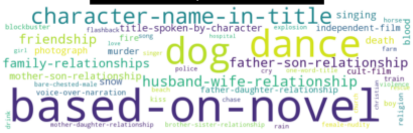
cluster: 8, 63 movies



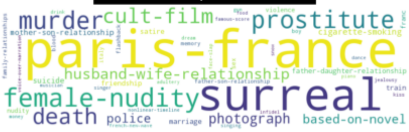
cluster: 9, 235 movies



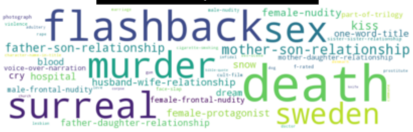
cluster: 10, 286 movies



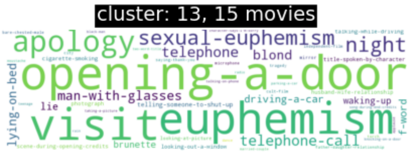
cluster: 11, 65 movies



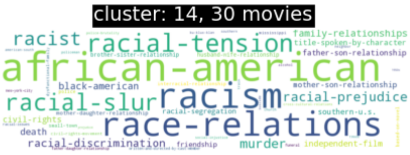
cluster: 12, 25 movies



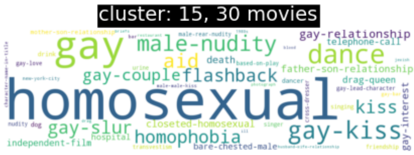
cluster: 13, 15 movies



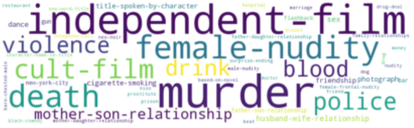
cluster: 14, 30 movies



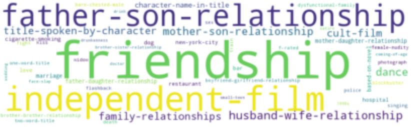
cluster: 15, 30 movies



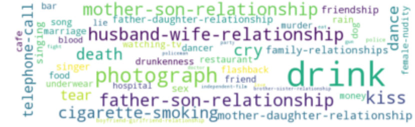
cluster: 16, 387 movies



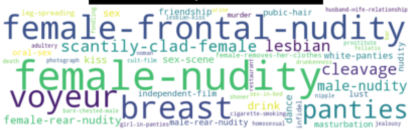
cluster: 17, 457 movies



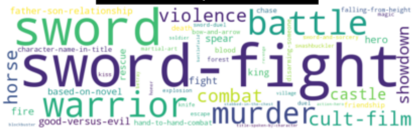
cluster: 18, 310 movies



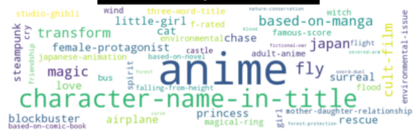
cluster: 19, 134 movies



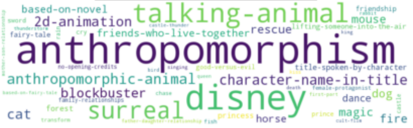
cluster: 20, 97 movies



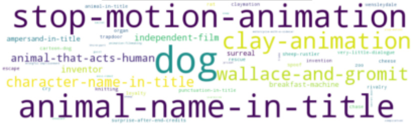
cluster: 21, 14 movies



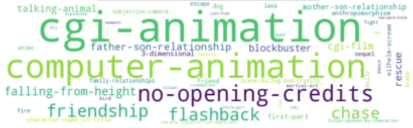
cluster: 22, 64 movies

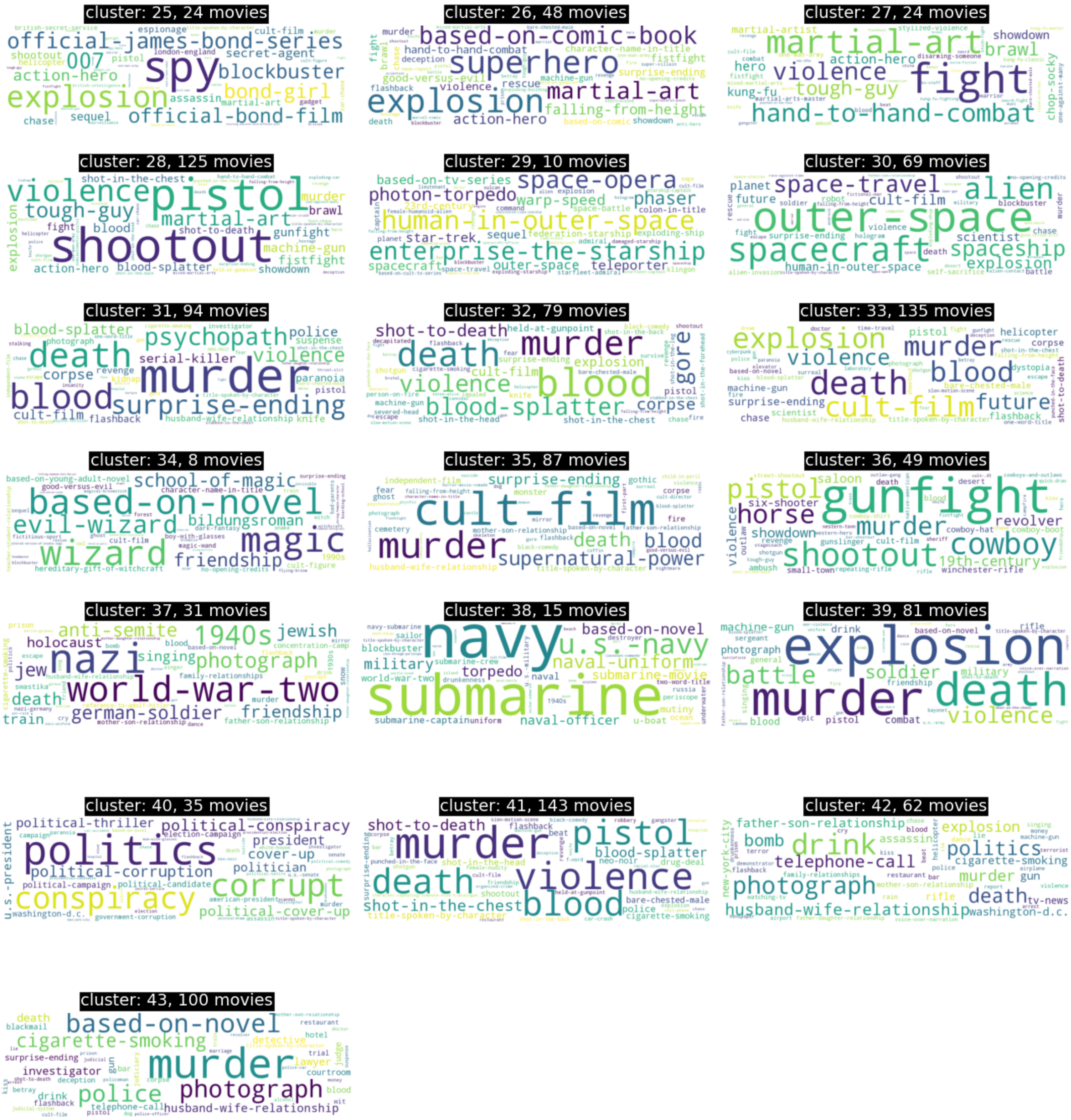


cluster: 23, 8 movies



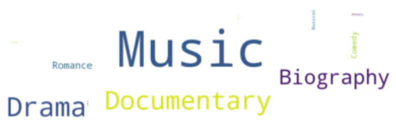
cluster: 24, 43 movies



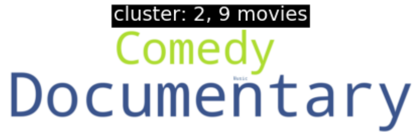


The same description is provided for the genres, which are more understandable than specific keywords.

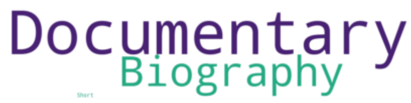
cluster: 1, 35 movies



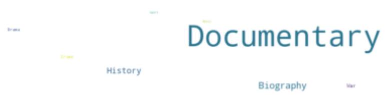
cluster: 2, 9 movies



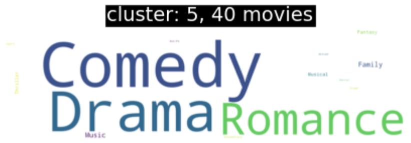
cluster: 3, 7 movies



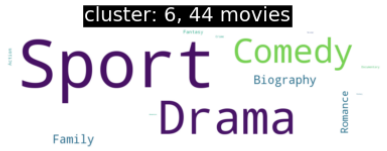
cluster: 4, 46 movies



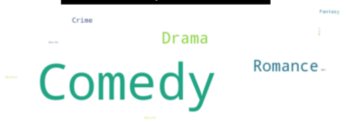
cluster: 5, 40 movies



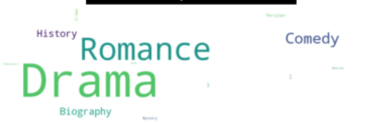
cluster: 6, 44 movies



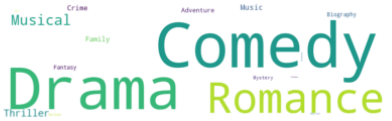
cluster: 7, 22 movies



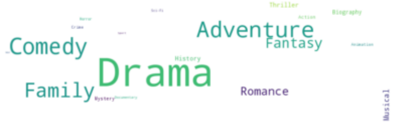
cluster: 8, 63 movies



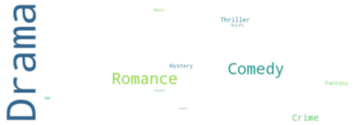
cluster: 9, 235 movies



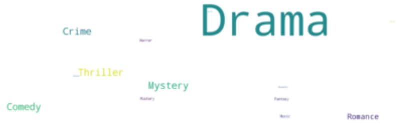
cluster: 10, 286 movies



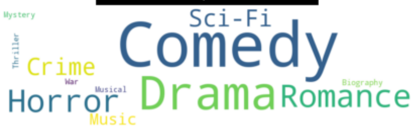
cluster: 11, 65 movies



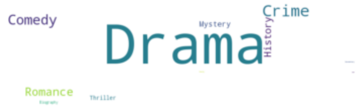
cluster: 12, 25 movies



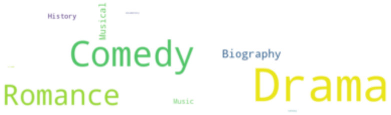
cluster: 13, 15 movies



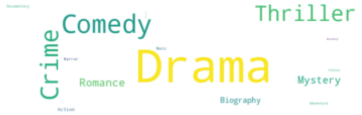
cluster: 14, 30 movies



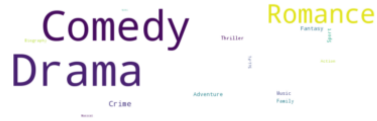
cluster: 15, 30 movies



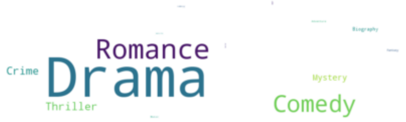
cluster: 16, 387 movies



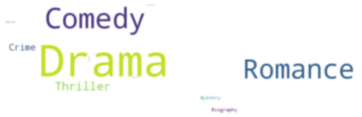
cluster: 17, 457 movies



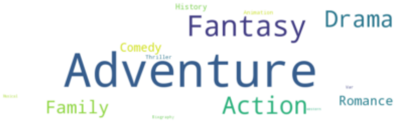
cluster: 18, 310 movies



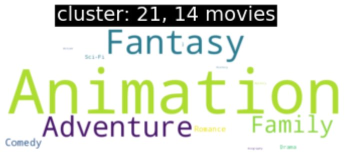
cluster: 19, 134 movies



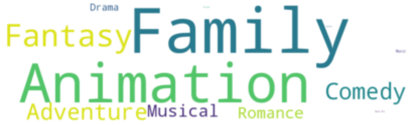
cluster: 20, 97 movies



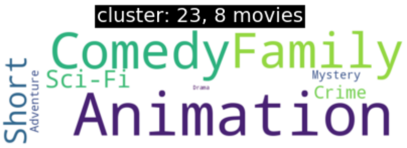
cluster: 21, 14 movies



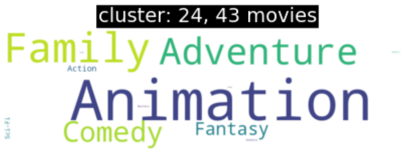
cluster: 22, 64 movies



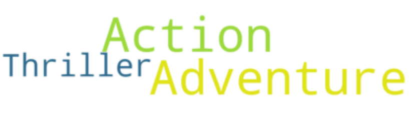
cluster: 23, 8 movies



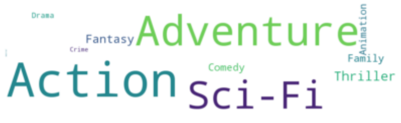
cluster: 24, 43 movies



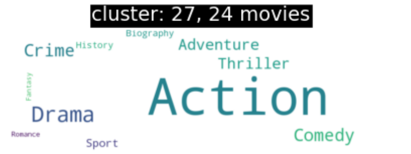
cluster: 25, 24 movies



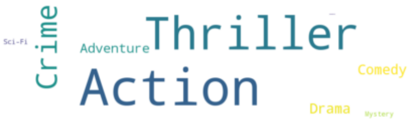
cluster: 26, 48 movies



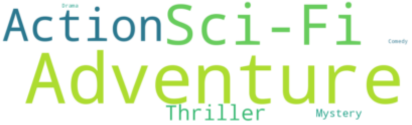
cluster: 27, 24 movies



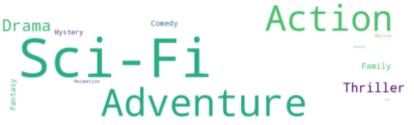
cluster: 28, 125 movies



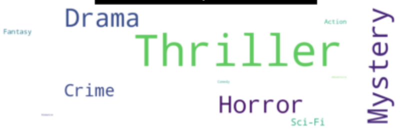
cluster: 29, 10 movies



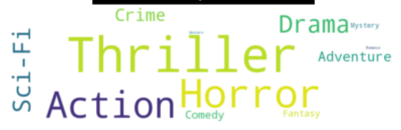
cluster: 30, 69 movies



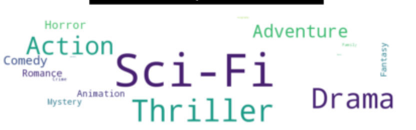
cluster: 31, 94 movies



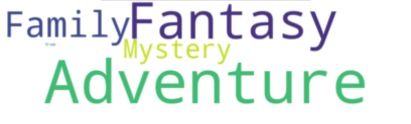
cluster: 32, 79 movies



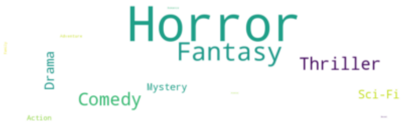
cluster: 33, 135 movies



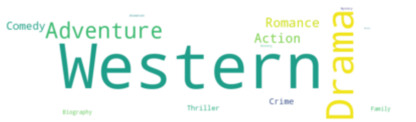
cluster: 34, 8 movies



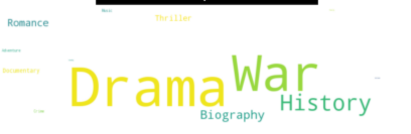
cluster: 35, 87 movies



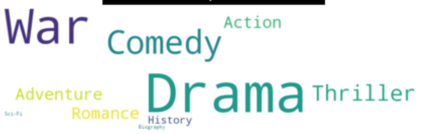
cluster: 36, 49 movies



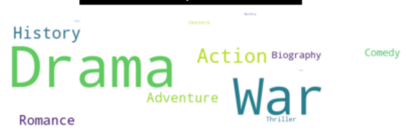
cluster: 37, 31 movies



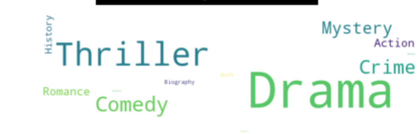
cluster: 38, 15 movies



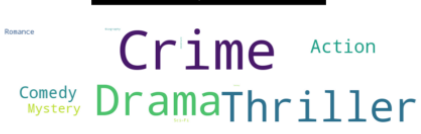
cluster: 39, 81 movies



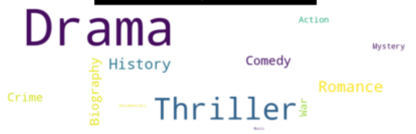
cluster: 40, 35 movies



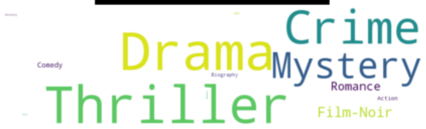
cluster: 41, 143 movies



cluster: 42, 62 movies

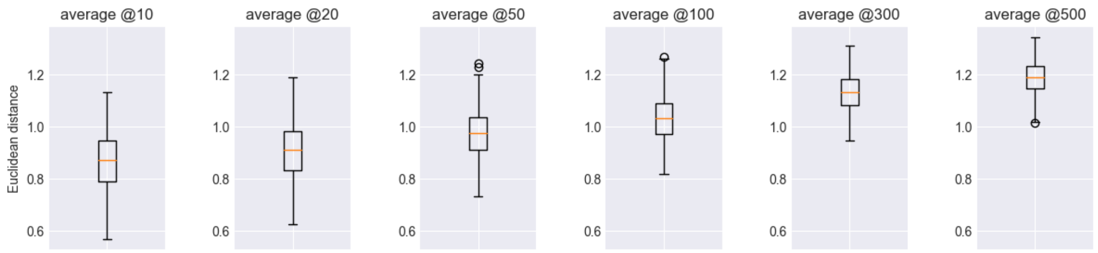


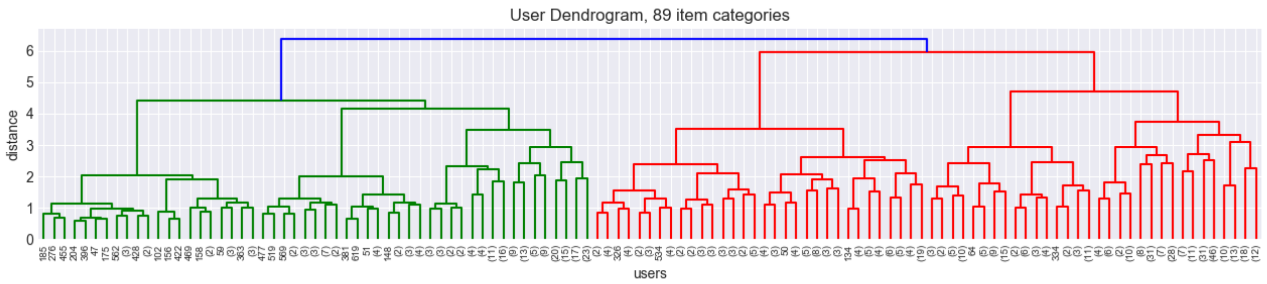
cluster: 43, 100 movies

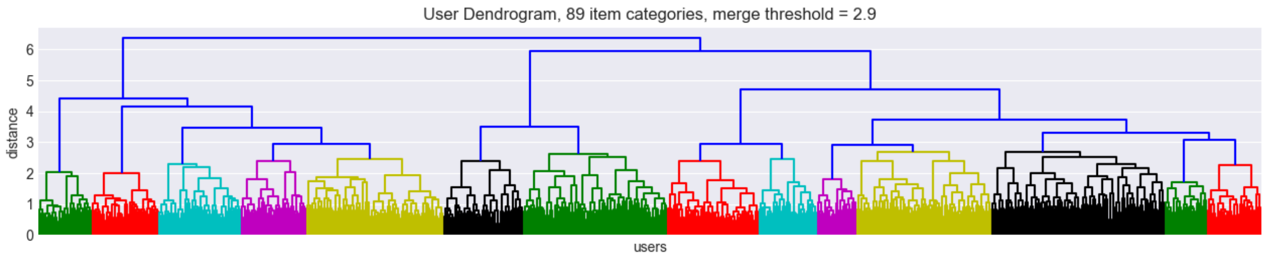


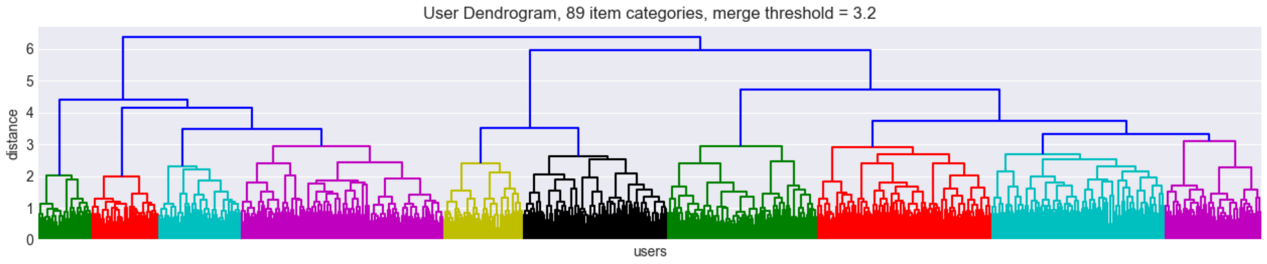
Appendix B

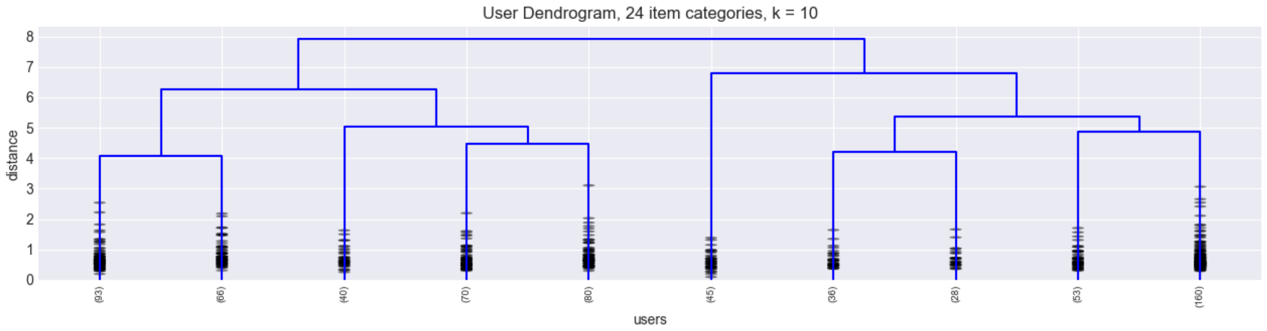
Here, the full examination on clusters of users is provided for $k = 89$ of item categories.

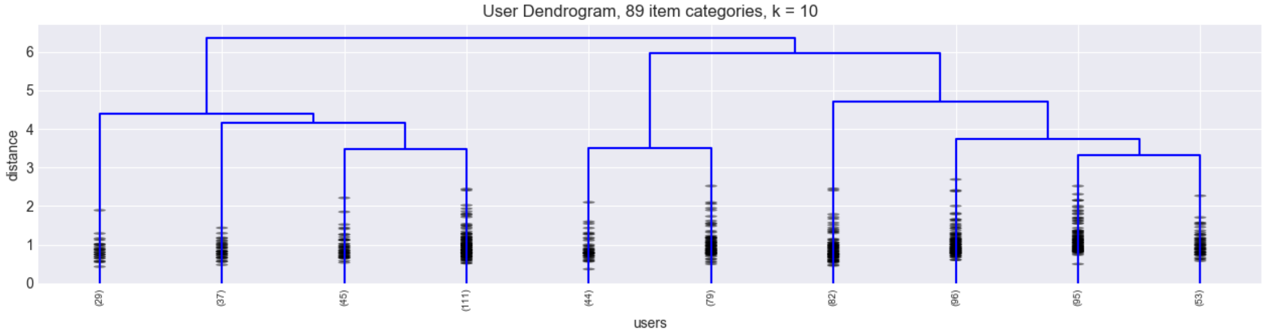



User Dendrogram, 89 item categories


User Dendrogram, 89 item categories, merge threshold = 2.9

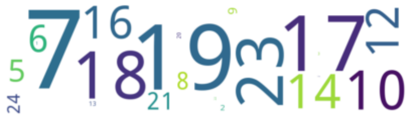

User Dendrogram, 89 item categories, merge threshold = 3.2


User Dendrogram, 24 item categories, k = 10

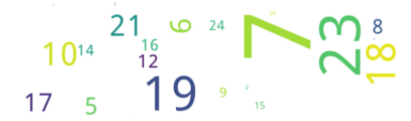


User Dendrogram, 89 item categories, k = 10

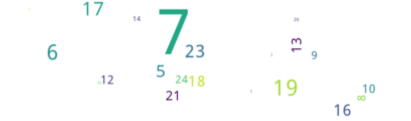
user cluster (24 categories): 1



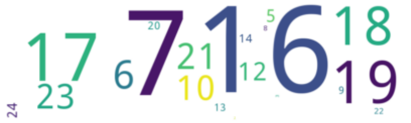
user cluster (24 categories): 2



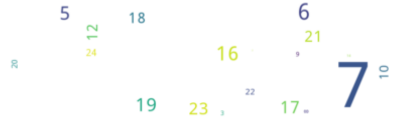
user cluster (24 categories): 3



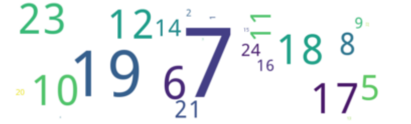
user cluster (24 categories): 4



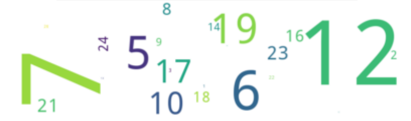
user cluster (24 categories): 5



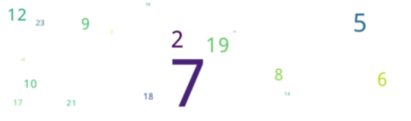
user cluster (24 categories): 6



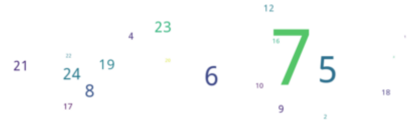
user cluster (24 categories): 7



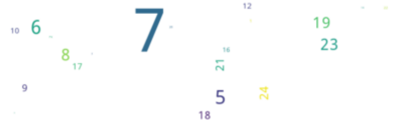
user cluster (24 categories): 8



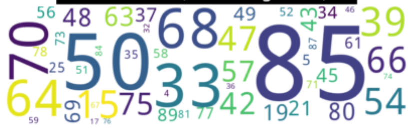
user cluster (24 categories): 9



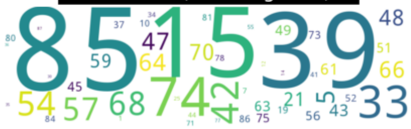
user cluster (24 categories): 10



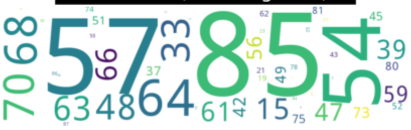
user cluster (89 categories): 1



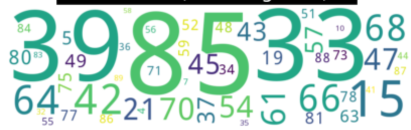
user cluster (89 categories): 2



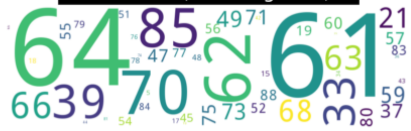
user cluster (89 categories): 3



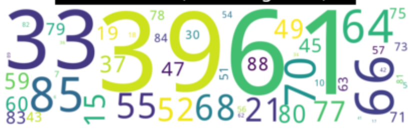
user cluster (89 categories): 4



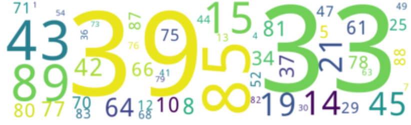
user cluster (89 categories): 5



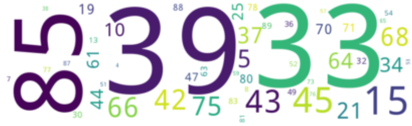
user cluster (89 categories): 6



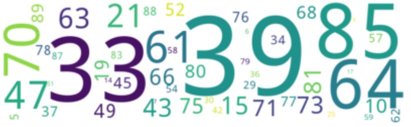
user cluster (89 categories): 7



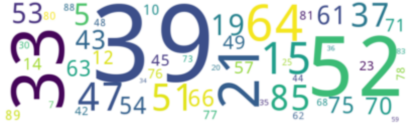
user cluster (89 categories): 8



user cluster (89 categories): 9



user cluster (89 categories): 10



53
5
10
39
19
64
61
37
33
43
49
71
14
63
12
45
73
1
15
52
23
57
47
54
51
66
2
85
75
70
78
89
42
77
62