Handling Data Sparsity in Collaborative Filtering using Emotion and Semantic Based Features

Yashar Moshfeghi
School of Computing Science
University of Glasgow
Glasgow, UK
yashar@dcs.gla.ac.uk

Benjamin Piwowarski
School of Computing Science
University of Glasgow
Glasgow, UK
bpiwowar@dcs.gla.ac.uk

Joemon M. Jose
School of Computing Science
University of Glasgow
Glasgow, UK
jj@dcs.gla.ac.uk

ABSTRACT

Collaborative filtering (CF) aims to recommend items based on prior user interaction. Despite their success, CF techniques do not handle *data sparsity* well, especially in the case of the *cold start* problem where there is no past rating for an item. In this paper, we provide a framework, which is able to tackle such issues by considering item-related emotions and semantic data. In order to predict the rating of an item for a given user, this framework relies on an extension of Latent Dirichlet Allocation, and on gradient boosted trees for the final prediction. We apply this framework to movie recommendation and consider two emotion spaces extracted from the movie plot summary and the reviews, and three semantic spaces: actor, director, and genre. Experiments with the 100K and 1M MovieLens datasets show that including emotion and semantic information significantly improves the accuracy of prediction and improves upon the state-of-the-art CF techniques. We also analyse the importance of each feature space and describe some uncovered latent groups.

Categories and Subject Descriptors: H.3.3 Information Storage and Retrieval - *Information Search and Retrieval* - *Information Filtering*
General Terms: Performance, Experimentation
Keywords: Semantic, Emotion, Collaborative Recommendation, Collaborative Filtering

1. INTRODUCTION

Recommender systems attempt to alleviate users' information overload by filtering documents that are not relevant to the users' interests [1]. Amongst these, collaborative filtering (CF) systems are the most widely used [19]. CF techniques recommend an item to a user by considering data gathered from other users [7]. Examples of such systems are Amazon.com for products, and Netflix[1] for movies.

In this work, we propose a novel approach for CF by integrating semantic and emotion information along with the rating information. This is motivated by the fact that a user likes a movie for a set of latent reasons, e.g. due to style of its direction. This information can improve the CF system prediction accuracy, especially when data is sparse, i.e. when there is not enough data available (e.g. ratings) to be used by standard CF techniques. Data sparsity is a well-known problem for CF systems [1] and in the extreme case of the so-called "cold start" problem, there is no rating for new users or items, making the prediction process impossible.

From a technical perspective, for each semantic (e.g. actor) and emotion (e.g. plot summary emotion) space we propose to construct latent groups of users. In order to do so, we extend a well-known model-based approach, namely Latent Dirichlet Allocation (LDA) [2]. In each space, we propose a methodology to compute the probability that a given user likes an item. Finally, in order to predict a rating, the information about the different spaces is aggregated using standard machine learning techniques.

Our work also is one of the few that explores the use of emotion in IR. Emotion is being proposed as a good feature to use in IR experiments [21]. Recent advances in psychology and linguistics make emotion extraction from textual documents feasible, thus making it appropriate to consider emotional features in IR. Despite their potential important role, emotions are seldom used in CF. In this work, we extracted emotions using the OCC[2] model (Section 3.1), and developed a sophisticated psycho-linguistic model to extract emotions expressed in movie reviews and plot summaries. We use these as a new source of information that indicates whether a movie will be liked or not by any user.

Finally, from an experimental point of view, we have conducted extensive experiments where we vary the sparsity of the dataset and compare our models to two state-of-the-art CF approaches. Furthermore we present preliminary experiments in an item cold start scenario and analyse qualitatively the latent spaces uncovered by our extended LDA approach.

The rest of the paper is organised as follows: Section 2 discusses related works in CF and emotion extraction techniques, our approach is described in Section 3, experiment methodology in Section 4, results in Section 5, and finally the discussion and conclusion in Section 6.

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

[2] OCC stands for the creators of the model, Ortony, Clore and Collins.

SIGIR'11, July 24–28, 2011, Beijing, China.

2. BACKGROUND

2.1 Collaborative Filtering Categories

CF techniques mostly make use of users' past ratings to predict user's level of interest for an item. Despite their success they face issues of scalability and sparsity [1].

The scalability problem refers to the quantity of data, i.e. the number of users and items: many CF algorithms fail to scale to big datasets. This issue is further stressed by the fact that in real-world systems, new items and new users appear every day, therefore generating a huge amount of data.

The sparsity problem refers to the situations where the ratio of unrated items to rated ones is high therefore not providing enough information for CF systems for their predictions. Data sparsity can be down to many factors. Firstly, judging is a cognitively expensive activity [6]. Secondly, there are unpopular or unseen items [17]. Finally, in special cases, known as the *cold start* problem (i.e. new user and/or new item), no ratings are available at all. In the following, we discuss how previous works in CF have tackled these two challenges.

CF systems can be divided into memory- and model-based approaches [1]. In a memory-based approach, recommendation is made by determining the nearest neighbours of a user and/or item, and then aggregating the ratings of these neighbours. These CF techniques have the advantage of being better adapted to users with unusual tastes, but they are impractical to use [7] due to scalability issues since calculating the neighbourhood for users and/or items can be time consuming, especially in real life (i.e. commercial) datasets.

Model-based techniques learn a user and/or item model from data [7] and are able to scale to large datasets; the model proposed in this paper lies in this category. There are many different model-based approaches such as regression models [27] based on user and item features or clustering of user and/or items, but the most successful ones are based on dimensionality reduction techniques because they deal better with data sparsity. We rely on such techniques in this work.

Dimensionality reduction techniques find lower dimensional latent topics from a higher dimensional information space. The most well known of these approaches are Latent semantic indexing (LSI), Principal Component Analysis (PCA), or probabilistic approaches like the probabilistic LSI (pLSI) [8], nonparametric probabilistic Principal Component Analysis (NPCA) and LDA [2]. A downside of reducing the dimension is that important information (i.e. ratings) that can be useful for predicting unusual user-item ratings may also be discarded [30].

In order to deal with different feature spaces, we base our work on LDA, which can be extended for our purposes and has been shown to be competitive compared to state-of-the-art CF techniques, outperforming the unigram model and pLSI [2]. This approach is also the base of many state-of-the-art CF systems such as the URP model by Marlin [14]. LDA will assign high probabilities to movies that are liked *and* judged by many users thus favouring popularity over the fact of being liked. This "popularity" problem becomes worse when feature spaces are used since some features can be present in just a few movies (e.g. a user likes an actor who has played only in a few not so popular movies). In this work, we show how an extension of LDA can overcome this problem.

In order to cope with data sparsity, it is necessary to resort to external sources of information. This becomes mandatory when dealing with the cold start problem, since the absence of rating hinders the possibility of using CF techniques that rely only on rating information.

A common approach is to use the representation of users and items to predict ratings. For instance, [25] first uses these representations along with a user profile to create new ratings and then applies the standard CF techniques on the denser data. However, in these approaches, the similarity between item and user is based on the actual representations of their contents and not on the latent relation among them.

A second approach tries to use this external information by capturing the latent relationship between items and users. Mobasher [19] uses a memory-based approach that takes advantage of an ontology and of a latent semantic model. However, creating and maintaining the ontology is a very laborious task. Park and Chu [24] use regression techniques where user demographical information and the item meta-data are used as features. However, the user demographic information is unreliable in general and, more importantly, in their approach users are not characterised by the movies they rated. Finally, Moshfeghi et al. [20] proposed to use a memory-based approach where the neighbourhood of an item in semantic spaces was used to predict the rating. However, in our work, we use a model-based technique, which can scale to large datasets and also considers both semantic and emotion information.

2.2 Emotion and Collaborative Filtering

Emotions are considered to be important factors influencing overall human effectiveness including the rational tasks such as reasoning, decision-making, communication and interaction [9]. In recent years there has been an increasing amount of research in both academia and industry to enable computers to detect emotions [3]. This is due to the utility of emotion information in applications such as those in affective computing, opinion mining, market analysis and human computer interaction. For example, Winoto and Tang [29] investigated the effect of user mood on their ratings and consequently modified a collaborative recommender system to depend on user mood. However, mood is difficult to gather since it either needs explicit feedback or relies on unreliable and invasive techniques (e.g. face detection). Fortunately, due to advances in NLP, we now have reliable tools to extract emotion data from textual sources [28]. Our approach exploits such techniques to extract emotional information and apply it in CF algorithms.

Text does not only contain topical but also emotional information. In fact, text contains clues to emotions related to what the writer wants to transmit or what his or her mood was at the time of writing [28]. Emotions extracted from text were first used in a sentiment analysis task [28]. The essential issue in sentiment analysis is to identify the positive (favourable) or negative (unfavourable) opinion towards the topic of a text. Sentiments extracted from movie reviews have been used to infer unknown ratings of users [23, 5]. Existing approaches (using sentiment in CF) are complementary to our work. For example using sentiment as a substitute for ratings [11] or relying on sentiment expressed over movie aspects (e.g. actors in the movie are good/bad) in order to provide opinionated ratings [10] could be used likewise in our work.

Other than using sentiments, none of the works exploit the much richer emotion information for collaborative filtering. In psycholinguistic and its practical applications, emotion sensing requires a significantly more detailed text analysis, since emotions are a finer-grained version of sentiment. We believe that this extra information is useful for CF.

There are multiple views on what emotions are and how they can be represented [9]. Ekman regards emotion as psychosomatic states and categorises them into six discrete categories[3] [3]. Some commercial systems follow this approach in order to classify emails [12]. The alternative OCC model [22] is considered to be more informative than Ekman's by the cognitive psychological community. It specifies 22 emotion types (joy, distress, happy-for, sorry-for, resentment, gloating, hope, fear, satisfaction, fears-confirmed, relief, disappointment, shock, surprise, pride, shame, admiration, reproach, gratification, remorse, gratitude and anger) and two cognitive states (love and hate), based on the valence reaction to agents, events and objects. One of the state-of-the-art emotion extraction methods was introduced by Shaikh et al. [28]. It adapts the OCC model for textual documents employing natural language processing techniques for emotion extraction. We use this method to identify emotions in text.

Given the improvement of emotion extraction systems and the overwhelming engagement of users in providing user-generated content and evidence on the role of emotion in decision making, we firmly believe that it is now appropriate to exploit emotions in recommendation. In contrast to previous work, we do not use sentiments to infer rating associated to a user review, but we instead use emotions as another source of information about an item, and show in the experiments that emotion information extracted from movie reviews and plot summaries can be useful to tackle the data sparsity problem.

3. APPROACH

In our approach, we postulate that CF techniques can be improved by taking into account item semantic and emotion information. In this section we describe our framework showing how it can be applied to movie recommendation.

We first suppose that a movie can be described as a set of features in a so-called feature space s. For example, in the actor space, a movie is described by the set of actors that played in the movie. The different spaces that we consider are further described in Section 3.1.

For a given space s, we first evaluate the probability that a user u, defined by his/her past ratings, likes the movie m that is described by a feature f (e.g. De Niro played in the movie):

$$P(+|f,u,s) \tag{1}$$

where + denotes the event that a user likes a movie. In Section 3.2, we show how LDA was extended in order to compute this value.

As a movie is described by a set of features (e.g. actors in the actor space) and thus it is necessary to aggregate the probability in Eq. (1) over these features, i.e. to estimate the probability that a movie m is liked by user u:

$$P(+|m,u,s) \tag{2}$$

We show in Section 3.3 a way to estimate Eq. (2) given the different probabilities calculated with Eq. (1) for each possible feature where the presence of a feature independently influences the relevance of an item.

Because the underlying characteristics of feature spaces vary, they provide different, and hopefully complementary views on the same object. Therefore we chose to use a machine learning approach to combine the prediction made on individual spaces (Section 4.3.2). The performance of each individual feature space, and discussion of whether they complement each other is in Section 4.

3.1 Feature Spaces

In this paper, we considered three types of feature spaces. The first and simplest type of feature space is movie space where we consider the movie itself as a feature. In this space, the probability of a movie to be liked is directly given by Eq. (1) where f is the movie at hand. It is the space on which most of the CF systems are based since it relates movie and users by their ratings.

The second type of feature space is semantic. In this work, we consider three such spaces, namely actor, director and genre spaces, since they are readily available on the Web and are likely to be good predictors for user ratings. A movie is represented as the set of features that characterise the movie in each of these spaces. For example, in the actor space, the movie *Dr. Mabuse: The Gambler* is associated with the list of its actors, i.e. Rudolf Klein-Rogge, Aud Egede Nissen, etc.

Finally the last type of feature space is the emotion space where for each movie, emotion features are constructed based on the emotion extracted from its reviews or plot summaries. In order to extract emotions from text, we used our implementation[4] of Shaikh et al. [28] which is considered as the state-of-the-art method. Since the extraction is sentence-based, the following explains the method of aggregating the detected emotions for movie related texts.

Both movie reviews and plot summaries are composed of a set of texts, which in turn are composed of a set of sentences. Let T denotes the set of texts associated to a movie (either reviews or plot summaries), and S_t the set of sentences associated to a text $t \in T$.

The emotion classifier of Shaikh et al. [28] makes a binary decision about each emotion for a sentence, i.e. decides whether the emotion is present or not. For each sentence k of a text t, we can use this classifier to construct a vector of 24 components, each of those associated with one of the 22 emotions and two cognitive states (see Section 2.2). Each component can take the value 0 (the emotion is not present in the sentence) or 1 (the emotion is present).

In order to represent the emotions in a text $t \in T$, we sum the emotion vectors of the sentences in t. Since we want to give equal importance to each text, we normalise the values by dividing by the number of sentences. Then, for a set of texts T we sum the vectors corresponding to the individual texts and normalise again, this time by the number of texts.

[3] Specifically happiness, sadness, fear, anger, disgust and surprise

[4] This is due to the fact that some components of Shaikh et al.'s system are not available. Our implementation achieve a better performance than the Liu et al. [13] system, which is considered as another state-of-the-art emotion extraction method [28]

Formally an emotion vector for a set of texts T is defined as

$$emotions(T) = \frac{1}{|T|} \sum_{t \in T} \frac{1}{|S_t|} \sum_{k \in S_t} emotions(k) \quad (3)$$

Since we use a LDA model (see Section 3.2) we eventually need to discretize the components of the emotion vector. As the distribution of values for each emotion can be very different we used a non-parametric way of discretizing by assigning to each value its corresponding quartile.

For example, if the values of the component corresponding to the emotion "fear" are distributed evenly in the four quartiles $[0, 0.3)$, $[0.3, 0.4)$, $[0.4, 0.75)$ and $[0.75, 1]$, then a value of 0.32 for fear would be transformed into 2. From an LDA perspective, this would in turn be represented by "fear-2".

3.2 Building Latent Spaces

In this section, we show how to estimate the probability that a movie is liked or disliked because of a feature. We build upon LDA [2] which is a generative probabilistic model for discrete data collection mainly used for textual corpora.

LDA represents documents as a distribution probability over latent topics, where each latent topic is a distribution over words. Documents that have similar topics should share the same latent topic distribution. This can be paralleled with CF where users who share the same ratings for the same items have related interests, and should thus be in the same *latent groups* that are defined by a similar distribution over features.

In the LDA approach to CF described[5] in [2], a user is defined by a probability distribution over a set of latent groups. Each group in turn defines a probability distribution over the movies that are liked by the users represented by this group. It is then possible to compute the probability that a user likes a movie by marginalising over the different possible latent groups. More formally, LDA defines the probability to observe a series of movies $\mathcal{M}_u^+ = (m_1, \ldots, m_n)$ liked by a user u:

$$p\left(\mathcal{M}_u^+ | \alpha, \beta\right) = \int p\left(\theta|\alpha\right) \left(\prod_{i=1}^{n} \sum_{z=1}^{T} p\left(z|\theta\right) p\left(m_i|\beta_z\right) \right) \mathrm{d}\theta \quad (4)$$

where T is the number of latent groups, θ follows Dirichlet distribution of hyper-parameters α, z (the latent group) follows a multinomial distribution given by θ and finally the probability of liking a movie m given the latent set of users z follows a multinomial distribution given by β_z. The model is fully specified by the α and the β_z for each possible latent group z. Those hyper-parameters are learnt by maximising the likelihood of the dataset (Section 4.3.1).

However, one of the problems of LDA is that it gives high probabilities to popular movies. Let us illustrate this problem with an example. First, consider two movies judged the same number of times. The probability given by LDA will rank the two movies in order of their probability to be liked. But, if the first movie has been judged by all the users and liked half of the time, it will have the same probability as another movie judged by only half of the users but liked all the time. This is not desirable.

This LDA limitation is due to the fact that in the movie space, LDA assigns high probabilities to movies that are liked *and* judged by many users. Thus the probability assigned to a movie m for a given latent group does not correspond to the probability that this movie would be liked by a user of this latent group, but rather to the probability that if we pick at random a movie liked by a user of this latent group, it will be m. Formally, LDA gives us the joint probability $P(+, m|u, s)$ instead of the conditional probability $P(+|m, u, s)$.

This "popularity" problem becomes worse when using feature spaces because some features can be presented in just a few movies (like an actor who has played only in a few not so popular movies).

We propose to alleviate this problem by considering negative information (i.e in movie space, movies that have been disliked, or in semantic or emotion space, features that appear in a movie that has been disliked). That is, we define LDA as a generative process of a series of couples feature-decision $\mathcal{F}_{u,s} = ((f_1, d_1), \ldots, (f_n, d_n))$ where f_i is a feature and d_i its associated decision, either "liked" (+) or "disliked" (-):

$$p\left(\mathcal{F}_{u,s}|\alpha, \beta\right) = \int p\left(\theta|\alpha\right) \left(\prod_{i=1}^{n} \sum_{z=1}^{T} p\left(z|\theta\right) p\left(f_i, d_i|\beta_z\right) \right) \mathrm{d}\theta$$

Let us illustrate how the set of couples is computed in the actor space. Assume that a user has (i) liked a movie with actors a, b and c; (ii) disliked a movie with actors a and b; (iii) liked a movie with actors a and d. This user would be represented by the couples $(a, +)$, $(b, +)$, $(c, +)$, $(a, -)$, $(b, -)$, $(a, +)$ and $(d, +)$.

Besides addressing the popularity problem, this approach also has two advantages. First, we consider more information to learn the LDA latent groups, since negative information is used. Second, user groups reflect not only features (e.g. actors) that appear in the movies they like, but also in the movies they don't like, thus providing richer information.

The LDA model is used to compute the posterior distribution of whether the feature f indicates a movie liked (+) or disliked (-) given the past user interaction $\mathcal{F}_{u,s}$ and the learnt parameters α and β, that is

$$P(\pm, f|u, s) = p\left(\pm, f|\mathcal{F}_{u,s}, s, \alpha, \beta\right)$$

In the next section, this probability is used to derive the final formula corresponding to a movie being liked in a feature space.

3.3 Probability Estimation based on a Feature Space

This section presents our methodology for calculating the probability that a movie is liked given a user and corresponding movie features. Our approach is based on a simple "averaging" method where the probability that the movie is liked is the expectation that the movie is liked because of each of its features. We also tried other forms of aggregation but preliminary results suggested that they would not improve over this simple method.

The probability $P(+|m, u, s)$ that user u likes movie m in

[5] To ease the reading, we adapted here the notations and concepts. For example, we refer to latent topics as *latent groups* since we are performing LDA on non textual information.

the feature space s can be written as

$$P(+|m,u,s) = \sum_{f \in F} P(+,f|m,u,s) = \sum_{f \in F} P(+|f,u,s)P(f|m,s) \qquad (5)$$

where F is the set of possible features for a given movie and where we assumed that (i) features are examined one at a time to make a decision about whether a movie is liked or not. In this case, f and f' are disjoint events whenever $f \neq f'$; (ii) when the feature is known the judgment does not depend any more on the movie, i.e. $P(+|f,m,u,s) = P(+|f,u,s)$; (iii) when there is no judgment involved, the fact that a given user and a given movie are independent, i.e. $P(u,m) = P(u)P(m)$; (iv) the fact that a movie has a given feature is independent from the user, i.e. $P(f|m,u,s) = P(f|m,s)$.

Eq. (5) reduces to the estimation of two quantities: the probability of considering the feature f given a movie m and a space s, i.e. $P(f|m,s)$, and the probability that a user u likes a movie given that it has the feature f, i.e. $P(+|f,u,s)$.

The latter probability is straightforward to estimate, since the probability $P(+|f,u,s)$ can be rewritten as

$$P(+|f,u,s) \quad = \quad \frac{P(+,f|u,s)}{P(+,f|u,s) + P(-,f|u,s)} \qquad (6)$$

where $P(\pm,f|u,s)$ is the probability that the feature f occurs in a movie that is liked (or disliked) by the user u in the space s, which is given by our extended LDA.

Note that when only a few observations are available for a given movie the estimations given by Eq. (6) can be unreliable. This is especially true when the data sparsity is high. We tried different smoothing techniques, and report the best performing one, the Laplace smoothing:

$$P(+|f,u,s) \quad = \quad \frac{P(+,f|u,s) + \epsilon}{P(+,f|u,s) + P(-,f|u,s) + 2\epsilon} \qquad (7)$$

The ϵ value is set to $0.001 \times |s|^{-1}$ where $|s|$ is the number of features of space s. This scaling was necessary in order to adapt to the different spaces where the number of features can vary greatly.

With respect to the probability $P(f|m,s)$, unless we have an a priori reason to give more importance to a feature (e.g. to give a higher importance to the main actors), we can assume a uniform distribution over the feature present in the movie m in the space s. Denoting $F(m,s)$ the set of features present in movie m in the space s and $\#F(m,s)$ the set cardinality, the probability $P(f|m,s)$ is $\frac{1}{\#F(m,s)}$ if f is a feature of space s for the movie m and 0 otherwise.

Putting the derived quantities back into Eq. (5), the final prediction formula is

$$P(+|m,u,s) = \sum_{f \in F(m,s)} \frac{1}{\#F(m,s)} \frac{P(+,f|u,s) + \epsilon}{P(f|u,s) + 2\epsilon} \qquad (8)$$

Note that in the case of the movie space, each movie is defined by one distinct feature and the sum reduces to one term.

Finally, to compute the final rating prediction for a given item we combine the information from the different spaces as given by Eq. (8), using boosted trees (see Section 4.3.2).

4. EXPERIMENTS

4.1 Test Collection

Our approach is evaluated on two MovieLens datasets [26] containing 100,000 ratings for 1682 movies from 943 users (100K dataset) and 1 million ratings for 3900 movies from 6040 users (1M dataset) respectively. In both datasets, there are at least 20 movie ratings per user. The rating scale takes values from 1 (not liked) to 5 (most liked).

We extracted the information needed to define the different semantic and emotion spaces from the IMDb website[6]. We considered the genre, the actors, and the director as our semantic spaces. Emotion extracted from plot summaries and movie reviews was used to define our emotion spaces.

4.2 Evaluation Protocols

The variability of our results was studied by performing a 10-fold cross validation where each time we used 70% of the users to train the LDA (Section 4.3.1), and 20% to identify the number of latent groups for LDA and to train the boosted trees based on LDA output (Section 4.3.2). The remaining 10% was used for performing the test.

In order to study the impact of sparsity on our models, following the standard methodology, we randomly removed some ratings from the training set so that the maximum number of rated items per user is below a given threshold (10, 20 and no limit, coined “full”), where 10 represents the highest sparsity and full the lowest.

The last processing step divides, for each user, the set of rated items into two. One set is used to represent the past history of the user, i.e. to compute the user representation in the various feature latent spaces. The second set of items are held out, and their predicted rating is computed with each model before being compared to the real value in order to measure the performance of the model. We considered the following two splitting methodologies:

Random For each user, we randomly divide the items in two. In doing so, some users might have rated an item that is held out for testing for another user.

Cold Start 10% of the items that have been rated by the test users were randomly selected to be the held out set for *all* users. In order to ensure that it is a cold start, we also removed the ratings of these items in the whole training set.

Metrics.

To measure the performance of the models, we used three different metrics. First, we report a measure of the average error made at the rating level. Two of the most widely used metrics of collaborative filtering, namely MSE (Mean Squared Error) and MAE (Mean Average Error) belong to this category. Results were similar for both, and in this paper, we only report the former due to space limits.

The goal of CF is often to return the relevant items to the user, such as the top rated movies. The performance of a CF algorithm with respect to those movies is better measured by mean average precision (MAP). Due to limited space, we do not include the complete results, but instead report the cases where MAP had a different behaviour than MSE.

[6] The Internet Movie Database (IMDb, http://www.imdb.com/)

Evaluation Methodology.

We tested the performance of our models with different combinations of the features spaces, i.e. Movie (M), Director (D), Actor (A), Genre (G) , Review Emotion (R) and Plot Summary Emotion (P) spaces. The configuration M is similar to LDA, but as explained in Section 3.2, does make use of negative information.

In the first set of experiments, we investigated the effect of each individual space. These models are represented by the initial Letter associated to each space (M, G, A, D, R or P).

In the second set of experiments, we investigated the effect of a combination of spaces. We experimented by using, besides the movie space, only emotion spaces (MPR), semantic spaces (MGAD) and all the spaces (MGADPR).

Finally, we used three different baselines. First, as a threshold, we report the performance of a constant rating estimator that returns the mean of the ratings in the training set. Second, for comparison we also report the performance of the original LDA approach (identified as LDA) along with our model on movie space (identified as M). Third, as a much stronger baseline, we report the performance of nonparametric probabilistic principal component analysis (NPCA) presented by Yu et al. [31], which has been shown to outperform other state-of-the-art approaches in the literature.

In the cold start situation, systems that rely on past item ratings cannot predict ratings for the items that do not provide such information (e.g. new items). This means, we will not be able to employ NPCA, original LDA, or our model based on movie space (M) to address the cold start problem. In the cases where the M space is combined with others, we simply removed the space M, leading to combinations based on emotions (PR), semantic spaces (GAD) and all the spaces (GADPR). It can be argued that the review emotion space (R) should not be used when we are dealing with the cold start problem. However, in our experiments, we consider the reviews as a movie feature rather than a user feature since any individual who is not part of the CF system can give these reviews. Moreover, we are interested to see the effect of utilising the review emotion space when there is no rating available for a movie.

4.3 Optimising Parameters

4.3.1 LDA

We described how we use LDA in Section 3.2. In order to train the LDA model, we need binary relevance judgments, and two sets of hyper-parameters, namely the number of latent groups T and the initial α and β.

With respect to the transformation into binary values, ratings of 3 (neutral) were discarded and we mapped 1-2 to negative, and 4-5 to positive.

We set the initial values of α and β as proposed by Misra et al. [18]. The number of latent groups has great influence on the performance of the LDA approach. We used the standard methods to find the right number of latent groups in dimensionality reduction techniques based on the likelihood over a held out set of training data [18].

In this paper, for each space and dataset, we tried several different quantities of latent groups, namely 3, 5, 10, 20, 35, 50, 100, 120, and 150. The maximum number of latent groups was set to 150 for computational reasons. The number of latent groups was selected by maximising the likelihood of observations over the second subset of the training set.

4.3.2 Boosted Tree

In order to predict the final rating, we use the standard machine learning technique of gradient boosted trees [4]. The features given to the boosted trees are a set of probabilities given by Eq. (8), one for each of the different spaces used in a given model, the output is a predicted rating between 1 and 5.

Note that even in the case of single space-based models, e.g. M, G or A, boosted trees are still useful since they map the probability of a movie to be liked to the rating scale. In order to ensure a fair comparison, we also used boosted trees to predict a final rating for the LDA-based model.

The parameters used for our experiment were found during preliminary experiments. We set the maximum number of trees to 2000, a maximum tree depth of 3 and a gaussian cost function that directly optimises the MSE. We used 65% of the data to train the boosted trees leaving 35% to control for over-fitting.

5. RESULTS & DISCUSSION

In Section 5.1 we analyse the results for models based on individual and multiple spaces, and discuss the latent groups discovered by LDA in Section 5.2.

5.1 Quantitative Study of the Performance of the Models

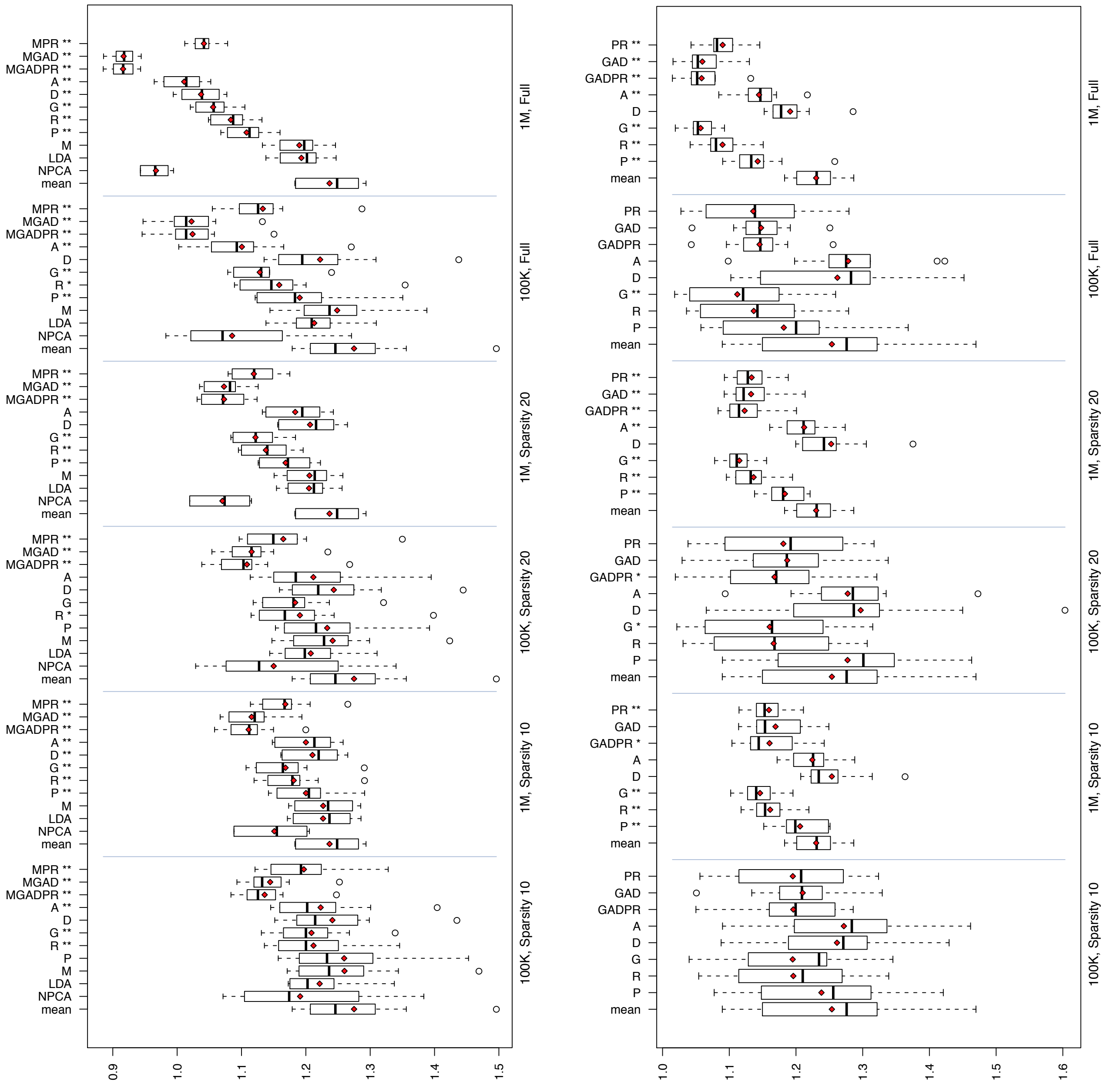
Figure 1a (random split) and 1b (cold start split) show the box plots for the MSE measure, for the two test collections (100K and 1M) and for different levels of sparsity (10, 20, and full). Each box plot reports, over the 10 cross validation sets, five important pieces of information namely the minimum, first, second (median), third, and maximum quartiles[7]. We performed a paired t-test between measures obtained for each user to check the significance of the difference with the baseline (M in Figure 1a and mean in Figure 1b). We use (*) and (**) to denote the fact that a model had results different from that of the baseline in all the cross validation sets with the confidence levels ($p < 0.05$) and ($p < 0.01$) respectively.

Main results.

With respect to models based on several feature spaces, we observed that the proposed model combining all spaces (MGADPR) consistently and significantly outperformed other models, including NPCA (shown in Figure 1a). With low sparsities, it has a better median and similar variance, and with high sparsities, it has a slightly better median but a much lower variance. This shows that substantial performance improvements can be achieved by integrating multiple sources of information for predicting ratings.

We can see that the model using movie space (M) has a slightly better performance (statistically significant) than original LDA (identified as LDA in Figure 1a) with complete data (1M, Full). However, in case of high sparsity, this model performed poorly. Hence for this space, popularity is important when data is sparse. We did not observe the same pattern for the models using semantic or emotion

[7] Further information can be found in [16].



(a) Random split

(b) Cold start split

Figure 1: Box plot of MSE for different sparsities (no sparsity, sparsity=20, sparsity=10) and datasets (100K, 1M). The lower the value, the better the performance. The diamond represents the mean of each model.

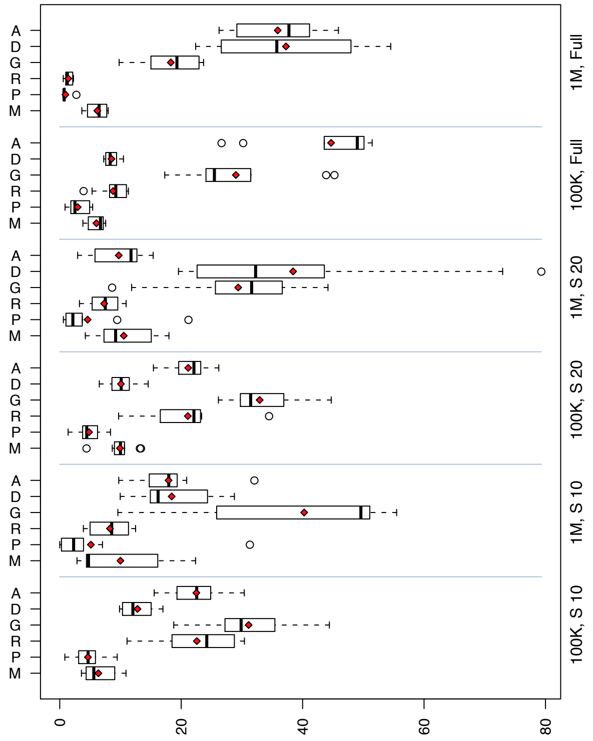


Figure 2: Box plot of the importance of each space in MGADPR model calculated by boosted trees for different sparsities (no sparsity, sparsity=10) , and datasets (100K, 1M). The higher the value, the higher the importance. The diamond represents the mean of each model.

space. In preliminary experiments (not reported here), the models based on our extended LDA (with negative and positive information) did outperform those based on the original LDA model. The improvement of our model, with semantic or emotion spaces, is significantly better than original LDA model.

Cold start.

The state-of-the-art methods discussed in this paper do not handle the cold start problem and we therefore consider the mean of ratings as our baseline. In Figure 1b we present how our proposed approach performs in the case of the cold start problem. There is an overall increase of 0.1 in MSE for GADRP over the baseline, and this holds for different level of sparsity. The comparison between the model using emotion spaces (i.e PR) and the one using semantic spaces (i.e. GAD) shows a comparable performance between them (sparsity "full" and 20), but with high sparsity (10), emotion space based models are performing better.

MAP and MSE.

By comparing the MAP (not shown here) and MSE performances, we can distinguish systems that are good at predicting the rating on average from those that improve the top ranking. We observed that movie and actor feature spaces do perform well at predicting top ranked items (at the price of a higher variability in the 100K dataset). The models based on emotions and to some extent genre (G) do not perform well in terms of MAP compared to the other spaces. This means that while they are useful to predict accurate ratings, they are not good predictors for highly rated items.

Analysing Feature Spaces.

Let us now analyse the performance of each feature space separately. In the case of the full scenario for the 1M dataset, we have a clear ordering of the different spaces based on their performance. As shown in Figure 1a, this order is actor, director genre, plot summary emotion, movie review emotion and finally movie space. When the sparsity of the database increases, the best performing spaces (i.e. actor and director) degrades more than the others. This is due to the fact that these spaces have many more features (i.e. the number of actors or directors is many more than the number of genres), and hence are more likely to lack information when sparsity increases. In the extreme case of the cold start problem, director and actor spaces performed the worst. Another interesting finding is that emotions are useful in sparse scenarios, and especially in cold start situations, as shown by the comparison of movie review emotion with genre spaces (Figure 1b).

In addition, the comparison between the model using emotion spaces (i.e MPR) and the one using semantic spaces (i.e. MGAD) shows a superiority of the latter one in low sparsity situations (Full in Figure 1a). Emotion spaces add an extra dimension of information in high sparsity scenarios as shown with sparsities of 10 and 20 (random and cold start splits) since they decrease MSE and the variance of the performance measure.

Finally, in order to investigate further the importance of the feature spaces, we used the measure of relative influence proposed by Friedman et al. [4] for boosted trees. Each of the features is given an importance between 0 and 100, so that the sum over features equals 100. The importance of a variable reflects how important the drop in performance would be, if the feature was to be removed. We report the box plot of the relative influence over the 10 cross validations for the model (MGADPR) in Figure 2.

We can observe different relative influences depending on the amount of data. In the 1M dataset with no sparsity (1M, Full in Figure 2), the most important spaces are actor, director, and then genre, and movie. This means that emotion spaces are not important when the data provides sufficient information to predict the rating of the movie based on more direct indicators like the actors in the movie.

However, at the other extreme, i.e. the most sparse scenario (100K, 10 in Figure 2), actor, director and movie space do not influence the decision making process of boosted trees as much as the movie review emotion space and genre do. Again we observe that the movie review emotion space acquires more importance than plot summary emotion space.

It can also be observed that there is a connection between the number of different features in a feature space and the use of this feature space for different volumes of data. For example, genre and emotion spaces have less features in comparison to the other ones, and are more likely to be used when there is not much data (i.e. ratings) to train from.

To investigate whether emotions do bring more information than sentiments, we performed experiments using a sentiment feature space[8] instead of emotion feature space (following the methodology in Section 3.1). The improvement is lower, but not significantly, than the one obtained over emotion feature space . However, the usefulness of the emotion feature space is illustrated in the next section where users' latent groups implicitly (through LDA) were created based on the emotions extracted from external movie reviews.

5.2 Analysis of Latent Groups

In this subsection we illustrate the latent groups discovered by LDA with some examples for two of the most important semantic spaces (i.e. actor and genre) and the emotion spaces. Here, we used the data learnt by LDA with the full data (1M, no sparsity) for better interpretability.

We first selected the most important latent groups based on the expectation of the probability $P(z)$ over the set of training users. For each of these latent groups, we calculated the top and bottom five features f for $p(+|f,s)$ and reported them in Table 1. Note that the number after each feature in the table for movie review (R) and plot summary (P) emotion spaces, corresponds to the discretised quartile to which it belongs (Section 3.1), and ranges from 1 to 4. We now discuss our findings, based on the analysis of several latent groups beyond those presented in the table.

In the actor space, we observe that in most of the latent groups, the important actors (i.e. those who play the main role) are separated from other actors (e.g. supporting actors). This is due to the fact that these actors were consistently liked whereas the supporting actors also play in the movies that a user has not liked. The second observation is that the actors who are categorised together in a latent group either play in the same movies or the same genre, or belong to the same period of time. LDA hence did correctly put together actors into coherent latent groups.

In the genre space, we observe that the features within latent groups are also related. For example, the genres with respectively the highest and the lowest probability do define distinct types of genre profiles, and the top movies related to this latent group perfectly match the liked genres.

In the movie review emotion space, we observe that the features within the latent group indicate those movies for which users expressed disappointment or dissatisfaction. By looking at the reviews given for the top movies for this latent group one can observe supporting comments such as "The movie had nothing to do with the title." and "It played off more as a B movie." for *Soft Toilet Seats* and "The story doesn't hit you over the head explaining events like most films" for *New Rose Hotel*. However, positive comments with high ambiguity such as "I saw this movie when I was very young and at first never really understood it." and "I mistake it for a Disney movie a lot of the time but who can blame me" can be considered as reproach and therefore the referring movie (in this case *The Swan Princess*) considered in the same group. Therefore, if a user has an unusual taste and likes the movies that the majority of people don't like then more such movies will be recommended to him or her. This feature is unique to this space. On the other hand, if the interest of the user is similar to the crowd, then the recommendation will also be common to that of the crowd such as the movies *Sneakers*, *Amistad*, and *A Simple Plan* for another important latent group.

In the plot summary emotion space, we observe that the features within latent groups meaningfully select movies based on the emotion interpretation of their storyline. For instance, the top latent group corresponds to the movies that have a twist of story or surprising (or shocking) emotion. This makes the plot summary different from genre space. This can be seen better by looking at the top movies for this latent group. For example, "her character is to be killed off" or "Her life begins to fall apart" for the movie *The Killing of Sister George* or "A . . . mother . . . facing divorce is thrust back in time . . . Given the chance to change the course of her life . . . " for the movie *Peggy Sue Got Married*, and "John Shaft . . . first finds himself up against . . . the leader of the black crime mob . . . finally working with [him] against the white mafia . . . " for the movie *Shaft*.

In order to gain further insight on the relationship between the different feature spaces, we calculated the Pearson's correlation of the predicted ratings for the model based on single features spaces. We observed that genre and emotion spaces (especially reviews that have a correlation around 0.8 for the complete dataset) had the highest correlation at different levels of sparsity. This is an interesting observation as the information present in these spaces is of a very different nature. However, it can be argued that the emotion expressed in the movie reviews are influenced the most by the genre of the movie. Other spaces had rather low correlations (below 0.4), showing that they are more likely to be complementary. A final observation is that correlation decreases when sparsity increases confirming the observation that considering different feature spaces is important in high sparsity situations.

6. CONCLUSIONS

In this paper we studied the effect of emotion and semantic spaces in improving the performance of a model-based CF system, and analysed the effect of sparsity and dataset sizes in rating prediction accuracy and recommendation precision. We observed that movie and actor spaces are the most and least sensitive spaces to sparsity and dataset size. We also observe that the model that uses all spaces is the best performing model over all sparsities and datasets performing better than a state-of-the-art CF system.

The LDA approach to CF was adapted to cluster semantic movie information as well as emotion information based on users' ratings. We proposed to include negative information (movies that have been disliked) into the LDA generation process and for each feature space we calculated the probability that a movie is liked (or not) given a user. Based on this, we predicted a rating using boosted trees.

The results show that emotional features consistently play a role in improving the recommendation quality in comparison to the scenario where only the movie space (i.e. the baseline) is used. Furthermore the effectiveness of emotion spaces increases with the sparsity of the dataset, especially in a cold start situation. This indicates that emotion spaces encapsulate a potential source of information. A comparison between the improvement achieved in MSE and MAP values shows that emotion spaces are more effective in predicting the actual ratings than detecting top rated movies. We also observed that movie review emotion space and genre space based models predict similar ratings, but it is important to note that emotion features are the outcome of an emotion extraction system and not manually created metadata as is

[8] The sentiment values were extracted based on the work presented in [15].

Table 1: The five highest and lowest probability features and the five highest probability of being a feature in a liked movie for the most important latent groups in actor, genre and movie review emotion spaces (No sparsity - 1M dataset)

Space	Features with the highest & lowest probability	Movies with the highest probability
Actor	**Highest:** Tommy Lee Jones, Samuel L. Jackson, Sean Connery, Fred Dalton Thompson, James Earl Jones **Lowest:** K. Baltz, P. T. Vince, E. Izzard, A. Dick, S. Lawrence	In the Line of Fire, Die Hard, The Mask of Zorro, U.S. Marshalls, Twister
Genre	**Highest:** Adventure, Crime, Musical, Mystery, Sci-Fi **Lowest:** Comedy, War, Documentary, Action, Western	The Man in the Iron Mask, Brenda Starr, Let's Get Harry, The Avengers, Indiana Jones and the Temple of Doom
Movie Review Emotion	**Highest:** reproach-1, reproach-4, surprise-1, distress-1, reproach-2 **Lowest:** distress-3, joy-2, joy-3, sorry-for-2, resentment-2	Above the Rim, Power 98, Soft Toilet Seats, New Rose Hotel, The Swan Princess
Plot Summary Emotion	**Highest:** surprise-1, shock-3, surprise-2, gloating-4, distress-3 **Lowest:** joy-1, hate-3, joy-3, hate-2, surprise-3	Peggy Sue Got Married, The French Connection, Shaft, Blast from the Past, The Killing of Sister George

the case for genre, thus they do not require costly and time consuming human intervention.

Emotion extracted from the movie plot summary and movie review emotion spaces affect system performance differently. We believe that this is due to the richer emotional content in opinionated movie reviews than the relatively more objective plot summary texts. It is also important to consider that there is much room for improving the accuracy of emotion extraction techniques.

In future work, we would like to improve our predictions for sparse datasets, relying more on semantic and emotion spaces. In order to do so, we plan to (1) investigate more sophisticated ways to aggregate feature probabilities for semantic and emotion spaces, in order to account for the variance (uncertainty) and importance of each feature (e.g. distinguish between main and secondary actors); (2) investigate the effect of semantic and emotion spaces in other domains such as books and products and (3) more importantly, we plan to continue working on utilising emotions extracted from text to improve the performance of information retrieval tasks such as novelty, diversification and personalisation.

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *TKDE'05*, June 2005.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR'03*, 2003.

[3] P. Ekman and H. Oster. Facial Expressions of Emotion. *Annual Reviews in Psychology*, 1979.

[4] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 2001.

[5] A. B. Goldberg and X. Zhu. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorisation. *TextGraphs-1*, 2006.

[6] M. Hancock-Beaulieu and S. Walker. An evaluation of automatic query expansion in an online library catalogue. *J. Doc.*, 1992.

[7] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. *SIGIR'03*, 2003.

[8] T. Hofmann. Latent semantic models for collaborative filtering. *ACM TOIS*, 2004.

[9] C. Izard. *The Psychology of Emotions*. Plenum Publishing Corporation, 1991.

[10] N. Jakob, S. H. Weber, M. C. Müller, and I. Gurevych. Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. *TSA '09*, 2009.

[11] C. Leung, S. Chan, and F. Chung. Integrating collaborative filtering and sentiment analysis: A rating inference approach. *ECAI'06*, 2006.

[12] H. Liu, H. Lieberman, and T. Selker. A model of textual affect sensing using real-world knowledge. *IUI'03*, 2003.

[13] H. Liu, T. Selker, and H. Lieberman. Visualizing the affective structure of a text document. *CHI'03*, 2003.

[14] B. Marlin. Modeling user rating profiles for collaborative filtering. *NIPS'04*, 2004.

[15] S. M. A. Masum, H. Prendinger, and M. Ishizuka. Sentiment assessment of text by analyzing linguistic features and contextual valence assignment. *AAAI'08*, 2008.

[16] R. McGill, J. Tukey, and W. Larsen. Variations of box plots. *American Statistician*, 1978.

[17] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. *AAAI'02*, 2002.

[18] H. Misra, O. Cappé, and F. Yvon. Using lda to detect semantically incoherent documents. *CoNLL'08*, 2008.

[19] B. Mobasher, X. Jin, and Y. Zhou. Semantically Enhanced Collaborative Filtering on the Web. *LNCS*, 2004.

[20] Y. Moshfeghi, D. Agarwal, B. Piwowarski, and J. M. Jose. Movie Recommender: Semantically Enriched Unified Relevance Model for Rating Prediction in Collaborative Filtering. *ECIR'09*, 2009.

[21] Y. Moshfeghi and J. M. Jose. Role of emotional features in collaborative recommendation. *ECIR'11*, 2011.

[22] A. Ortony, G. Clore, and A. Collins. *The cognitive structure of emotions*. Cambridge University Press, 1990.

[23] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *ACL'05*, 2005.

[24] S.-T. Park and W. Chu. Pairwise preference regression for cold-start recommendation. *RecSys'09*, 2009.

[25] D. Pennock, E. Horvitz, S. Lawrence, and C. Giles. Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. *UAI'00*, 2000.

[26] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. *CSCW'94*, 1994.

[27] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. *WWW'01*, 2001.

[28] M. A. M. Shaikh, H. Prendinger, and M. Ishizuka. A Linguistic Interpretation of the OCC Emotion Model for Affect Sensing from Text. *Affective Information Processing*, 2009.

[29] P. Winoto and T. Y. Tang. The role of user mood in movie recommendations. *Expert Systems with Applications*, 2010.

[30] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. *SIGIR'05*, 2005.

[31] K. Yu, S. Zhu, J. Lafferty, and Y. Gong. Fast nonparametric matrix factorisation for large-scale collaborative filtering. *SIGIR'09*, 2009.