Journal of Universal Computer Science, vol. 21, no. 2 (2015), 223-247
submitted: 10/6/14, accepted: 8/1/15, appeared: 1/2/15 © *J.UCS*

Leveraging Hybrid Recommenders with Multifaceted Implicit Feedback

Marcelo G. Manzato
(Institute of Mathematics and Computer Science
Sao Paulo University (ICMC–USP), Sao Carlos, SP – Brazil
mmanzato@icmc.usp.br)

Edson B. Santos Junior
(Institute of Mathematics and Computer Science
Sao Paulo University (ICMC–USP), Sao Carlos, SP – Brazil
ditusp@icmc.usp.br)

Rudinei Goularte
(Institute of Mathematics and Computer Science
Sao Paulo University (ICMC–USP), Sao Carlos, SP – Brazil
rudinei@icmc.usp.br)

Abstract: Research into recommender systems has focused on the importance of considering a variety of users' inputs for an efficient capture of their main interests. However, most collaborative filtering efforts are related to latent factors and implicit feedback, which do not consider the metadata associated with both items and users. This article proposes a hybrid recommender model which exploits implicit feedback from users by considering not only the latent space of factors that describes the user and item, but also the available metadata associated with content and individuals. Such descriptions are an important source for the construction of a user's profile that contains relevant and meaningful information about his/her preferences. The proposed model is generic enough to be used with many descriptions and types and characterizes users and items with distinguished features that are part of the whole recommendation process. The model was evaluated with the well-known MovieLens dataset and its composing modules were compared against other approaches reported in the literature. The results show its effectiveness in terms of prediction accuracy.

Key Words: recommender systems, implicit feedback, metadata awareness, user demographic, latent factors
Category: H.3.1, H.3.3, H.3.4

1 Introduction

Recommender systems are an important mechanism to enable users to deal with the increasing information overload. They provide suggestions of items and services, which are chosen in a way to match the user's preferences and interests [Adomavicius and Tuzhilin, 2005]. Over the past 20 years, research efforts have been devoted to the development of strategies and methods to automatically recommend items to people. Such studies are related to a variety of topics, including matrix factorization models, constraint-based and context-aware recommenders, social and group recommenders, among others [Shi et al., 2014, Bobadilla et al., 2013, Ricci et al., 2011]. Those advances have also reached the industry, in particular involving the collaboration of other users. Good examples are Amazon[1] and Netflix[2], of which the former provides similar products to consumers according to what they have acquired and the latter improves the user's experience by providing recommendations of movies based on preferences that other users have expressed for those items.

In its basic form, a recommender system can be designed according to three different strategies: i) content-based filtering, in which the user will be recommended items whose metadata are similar to the information stored in his/her profile; ii) collaborative, in which content selection is based on similar items appreciated previously or decisions made by other users with similar preferences; and iii) hybrid approaches, which combine collaborative and content-based recommendations.

Despite the variety of possibilities to implement recommender systems, most efforts have been devoted to the field of collaborative filtering (CF). Such studies have opened opportunities to deal with limitations found in content-based methods, such as limited content analysis and overspecialization [Adomavicius and Tuzhilin, 2005]. Two topics are studied in the CF technique: neighborhood models and latent factors. In the first case, clusters of items are formed to recommend items similar to the ones preferred by the user in the past. Alternatively, clusters of users can be formed to recommend items to a specific user, i.e. items valued by other users with similar preferences. In latent factors, the recommendation can be computed by uncovering latent associations among users or items. Thereby, items and users are transformed into the same latent factor space so that they can be directly comparable [Koren, 2010].

After having participated in the Netflix Prize competition[3], Bell & Koren [Bell and Koren, 2007] reported a number of lessons to guide researchers in the study of CF models. One of these lessons refers to the effectiveness of latent factor models to estimate an overall structure that simultaneously relates most or all items to each other, but at the cost of not detecting strong associations among a small set of closely related items [Koren, 2010]. Although this issue can be addressed by neighborhood models, computing users and items similarities can be a very computationally expensive task. Another lesson reported by Bell & Koren regards the importance of integrating different forms of user's inputs into the models. In fact, many efforts nowadays are related to the integration of explicit feedback with user's demographics and implicit feedback, which are

[1] http://www.amazon.com/
[2] http://www.netflix.com/
[3] http://www.netflixprize.com/

indirect clues users provide about their preferences when interacting with the system [Oard and Kim, 1998]. Examples of explicit, demographic and implicit information are, respectively, ratings manually assigned to visited items, user's gender, and user's purchase history.

Using different forms of user's inputs leads to the construction of a more accurate user's profile [Da Costa et al., 2014]. In this sense, hybrid recommenders play an important role because they group together the benefits of content-based and collaborative filtering. Limitations of both approaches, such as the cold start problem, overspecialization and limited content analysis can be reduced when the two strategies are combined into a unified model [Adomavicius and Tuzhilin, 2005]. However, most systems [Hu et al., 2008, Rendle et al., 2009, Koren, 2010, Yang et al., 2012] that exploit latent factor models and implicit feedback consider neither the metadata associated with the content, nor the user's personal information. In fact, semantic and known descriptions could be added into the profiles of individuals, in contrast to the obscure and incomprehensive relations of latent factors.

In view of the current state-of-the-art of hybrid recommenders, this article proposes a unified model that exploits implicit feedback from users by considering not only the latent space of factors that describes the user and item, but also the available metadata associated with them. Different types of information, such as user's personal data (e.g. gender, age) and items' metadata (e.g. genres, keywords) are carefully modeled to create a more meaningful user's profile containing his/her main interests, resulting in an improvement of prediction accuracy.

The article is structured as follows: Section 2 addresses the related work; Section 3 describes the past models explored in the study; Section 4 presents the proposal; Section 5 describes the evaluation conducted in the system; finally, Section 6 addresses the final remarks and future work.

2 Related Work

The model proposed in this article exploits three features to improve the recommendation accuracy. The first is implicit feedback, which provides additional and indirect clues about the user's preferences to characterize his/her profile. For example, navigation history, clicks with pointing devices, time spent on visits to items are categorized as implicit feedback because the user has not shown his/her preferences explicitly, which in this case, are inferred from the system through a set of heuristics [Oard and Kim, 1998].

The second feature is the incorporation of items' metadata and their factorization to adjust their relative importance in face of the users' preferences – for instance, when watching movies, the system might consider each movie's title, year of production, genres and set of actors as additional information to support the construction of factorization matrices of users and items.

Finally, the third feature is the incorporation of users' personal data, such as demographic information, which is usually provided by the user when signing up to the system – for example, the user's gender, age, location, etc. Sections 2.1 to 2.3 discuss each of these three features in the context of recommender systems.

2.1 Implicit Feedback

Implicit feedback extracts user's preferences mainly when explicit feedback is unavailable or incomplete [Joachims et al., 2005, Agichtein et al., 2006]. Oard & Kim [Oard and Kim, 1998] identified different types of implicit feedback and the way they could be exploited in a set of recommendation strategies. Hu et al. [Hu et al., 2008] developed a tool to transform users' implicit feedback into training data in a preference-conference format. Yang et al. [Yang et al., 2012] proposed a simple and effective local implicit feedback model mining users' local preferences to obtain better results. It extended Koren's algorithm by incorporating the notion of rating time interval to gather local and momentary interests.

Another approach for the extraction of more accurate information about users' interests is to exploit implicit and explicit feedback considering the individual is prone to interact with the content in different ways along its consumption. Koren [Koren, 2008, Koren, 2010] integrated implicit information and ratings into a neighborhood latent factor model to improve recommendation accuracy. In this same study, the author developed the SVD++ algorithm, which is a latent factor model that exploits implicit and explicit feedback gathered from the user's movies rental history and ratings assigned to visited items, respectively. Jawaheer et al. [Jawaheer et al., 2014] and Da Costa et al. [Da Costa et al., 2014] proposed a framework for the use of explicit and implicit user's feedback based on a set of heuristics to gather better insights about personal preferences.

Our study differs from the aforementioned studies because the proposed model incoporates item's metadata, user's personal information and implicit feedback into a unified approach to improve recommendation accuracy.

2.2 Metadata Awareness

A variety of content-based methods regarding metadata incorporation is available in the literature [Shi et al., 2014, Bobadilla et al., 2013, Adomavicius and Tuzhilin, 2005]. In general, they design matching mechanisms between the item's metadata and the user's profile, making the recommender engine to work similarly to search/retrieval systems. However, content-based filtering algorithms have usually associated problems, such as occurrence of cold start and overspecialization, which may affect the recommendation results.

Such problems can be reduced with collaborative filtering, whose main idea is to compute similarities of users/items in order to predict new items to users. An efficient way to compute such similarities is to use matrix factorization models to reduce the dimensionality of the user-item matrix [Koren et al., 2009]. However, if Singular Value Decomposition (SVD) is adopted to factorize the users vs. items matrix, imputation methods must be incorporated so as to reduce the sparsity effects, but at the cost of distorting and/or overfitting the training data.

Regardless of the possibility of using only the observed ratings to reduce the sparsity effects [Funk, 2006, Koren, 2010, Shi et al., 2013], an alternative to compute users similarities by means of factorization techniques was proposed by Manzato [Manzato, 2012]. The author created a user-category matrix factorization model to extract user's preferences about movies. The system computes the users similarities to support collaborative filtering. The idea of factorizing a matrix associated with metadata (e.g. movies' genres) was also studied in other related works [Agarwal and Chen, 2009, Stern et al., 2009, Gantner et al., 2010, Manzato, 2013, Manzato et al., 2014]. However, these studies do not support an integrated model that combines user's personal information with other important features in recommender systems, such as implicit feedback.

2.3 Personal Information

Most studies that consider user's personal information are related to demographic filtering [Pazzani, 1999, Krulwich, 1997]. This approach is based on the assumption that users with common personal characteristics (e.g. country, gender, age, occupation) will also have similar preferences. Consequently, a simple and effective way to explore this idea is to use collaborative filtering boosted by demographic information. Chen & He [Chen and He, 2009], for instance, proposed a collaborative filtering algorithm that computes users similarities based on three demographic attributes and ratings of items separately. A new similarity is then generated by combining the previous results. Lee & Woo [Lee et al., 2002] first segmented all users by demographic characteristics and then applied a user clustering procedure to each segment according to the preference of items and using a Self-Organizing Map (SOM) neural network. Yapriady & Uitdenbogerd [Yapriady and Uitdenbogerd, 2005] proposed a simple measure to combine demographic data with traditional collaborative filtering techniques to improve recommendation precision. Vozalis & Margaritis [Vozalis and Margaritis, 2007] proposed a collaborative filtering approach that uses SVD as an augmenting technique and demographic data as a source of additional information to improve the quality of predictions.

The aforementioned techniques are related to our approach in certain aspects: on the one hand, some methods that exploit implicit feedback do not consider the available user's and item's metadata; on the other hand, a set of models that exploits metadata and their factorization does not support implicit feedback.

A recent model that considers all types of information mentioned above as input data is factorization machines (FM) [Rendle, 2012a, Rendle et al., 2011]. The idea is to estimate interactions among categorical variables combining feature engineering and factorization models. Thereby, personal information, metadata and implicit feedback can be modeled as features, as in other machine-learning approaches, such as linear regression and support vector machines, and the model uses factorized interactions between the variables to learn its parameters. As stated by the author, factorization machines are generic enough to accept any type of information as input data, including metadata related to both users and items, context information, multiple users' feedback, etc. Although generality is an important characteristic in recommender systems, as it can be adapted to different application scenarios, its drawback is that all variables will be treated in the same way. For instance, a prediction rule that will use metadata weights to adjust the items factors prior to computing the final prediction cannot be defined. In contrast, factorization machines will interact with all variables regardless of the type of each input datum.

This article proposes a well-defined prediction rule generic enough to accept different types of users' and items' metadata and implicit feedback. The advantage is all information related to users, items and interactions is treated accordingly, which results in a more accurate rating prediction.

3 Past Models

This section describes the models reported in the literature and explored in this study.

3.1 Notation

Following the same notation of [Koren, 2008, Koren, 2010], we use special indexing letters to distinguish users, items and metadata, i.e., users are indicated as u, items are referred to i and j, and users' and items' metadata are stated as d and g, respectively. As all users' and items' descriptions may be of different types, we also use index letters z_u and z_i to refer, respectively, to each metadata type, such as title, gender, age, and occupation.

A rating r_{ui} refers to the explicit feedback a user u has assigned to an item i and is distinguished from the predicted $\hat{r}_{ui}$, which is a value guessed by the recommender algorithm. The (u, i) pairs for which r_{ui} is known are represented by set $K = \{(u, i)|r_{ui} \text{ is known}\}$.

Because the rating data are sparse, the models are prone to overfitting, therefore, regularization is applied so that estimates are shrunk towards baseline defaults. Similarly to Koren [Koren, 2010], we denote by $\lambda_1, \lambda_2, ...$ the constants used for regularization. Their values are defined in Section 5, which describes the experiments with the dataset adopted for the evaluation of the proposed model.

A summary of the notation and sets used in the article is shown in Table 1.

Table 1: Notation used in this article.

Notation	**Definition**
u	Index letter indicating a user.
i, j	Index letters indicating an item.
d, g	Index letters indicating personal information and item's metadata, respectively.
z_u, z_i	Index letters indicating a personal information type and an item's metadata type, respectively.
f	Number of factors of the model.
μ	Overall average rating.
b_u, b_i, b_d, b_g	Biases related to a user, an item, personal information and an item's metadatum, respectively.
b_{ui}	Baseline estimate (see Equation 1).
b_{ui}^{demo}	Baseline revisited (see Equation 8).
r_{ui}, $\hat{r}_{ui}$	A rating and its prediction, respectively.
$\hat{r}_{ui}^{meta}$	A rating prediction with item's metadata incorporation (see Equation 9).
λ, $\lambda_{1...7}$, α	Regularization constants.
γ, γ_2	Learning rates.
p_u, q_i	User and item factor vectors, respectively.
x_g, y_j	Metadata and implicit feedback factor vectors, respectively.
h_{dg}	A parameter that captures the weights of a user's personal information d associated with an item's description g.
K	Set of known ratings.
$R(u)$	Set of items rated by user u.
$R(i)$	Set of users who rated item i.
$N(u)$	Set of items to which user u provided an implicit feedback.
$N(j)$	Set of users who provided an implicit feedback to item j.
$Z(u)$	Set of different personal information considered in the system.
$Z(i)$	Set of different items' metadata considered in the system.
$G(u; z_u)$	Set of descriptions of type z_u associated with user u.
$G(i; z_i)$	Set of descriptions of type z_i associated with item i.

3.2 Baseline Estimates

Baseline estimates are used to encapsulate systematic tendencies from data according to users' and items' intrinsic characteristics. For example, a user may use value 4 to rate a great movie, whereas another user may adopt value 5 to indicate the same degree of interest. Similarly, an item may be rated differently by users, although some of these ratings may refer to the same likeness.

In order to overcome such differences, baseline estimates are used to adjust the data according to these effects [Koren, 2008, Koren, 2010]. A baseline estimate for an unknown rating r_{ui} is denoted by b_{ui} and defined as

$$b_{ui} = \mu + b_u + b_i \quad , \tag{1}$$

where μ refers to the overall average rating and parameters b_u and b_i indicate the observed deviations of user u and item i, respectively, from the average. Two methods can be adopted for the estimate of these parameters. The first consists in decoupling the calculation of the item biases from the user biases [Herlocker et al., 2002, Koren, 2008, Koren, 2010]. For each item i, the bias is computed as

$$b_i = \frac{\sum_{u:(u,i)\in K}(r_{ui} - \mu)}{\lambda + |\{u|(u,i) \in K\}|} \quad . \tag{2}$$

Then, b_i is used to calculate the user bias

$$b_u = \frac{\sum_{i:(u,i)\in K}(r_{ui} - \mu - b_i)}{\lambda + |\{i|(u,i) \in K\}|} \quad . \tag{3}$$

The second and more accurate method to estimate b_u and b_i solves the least squares problem

$$\min_{b_*} \sum_{(u,i)\in K} (r_{ui} - \mu - b_u - b_i)^2 + \lambda(\sum_u b_u^2 + \sum_i b_i^2) \quad , \tag{4}$$

where the first term before regularization aims at finding the user and item biases that have fitted the given ratings. The second term avoids overfitting by penalizing the magnitudes of the parameters.

3.3 Implicit Feedback

In recommender systems, an important issue refers to the integration of different forms of user's input into the models for a precise reflection of the user's preferences [Bell and Koren, 2007]. Algorithms usually rely only on explicit feedback, which includes ratings assigned by users on items they have visited. A good example is Netflix[4], which enables users to choose and assign an amount of stars to

[4] http://www.netflix.com/

movies they have watched. In response, the system constructs and controls the user's profile by updating his/her personal interests according to the assigned ratings.

On the other hand, one may argue that explicit feedback is not always available due to cold start, or simply because for some reason, users may not provide any ratings for their preferences. Consequently, implicit feedback could be exploited, as it is a more abundant source of information that indirectly reflects on the user's opinion by observing his/her behavior [Oard and Kim, 1998]. Examples of implicit feedback are purchase or rental history, browsing activity, search patterns, etc.

Koren [Koren, 2008, Koren, 2010] proposed a set of models that faces implicit feedback when explicit feedback is also available. The model integrates both types of feedback by considering ratings assigned by users to visited items and also the rental history. The adopted dataset (Netflix) lacks this type of implicit feedback, therefore the author simulated such information by considering the movies rated by the users regardless of how they were rated.

The most accurate model reported by Koren is the SVD++ algorithm, which integrates explicit and implicit feedback into a factorization model representing the user's preferences. Each user u is associated with a user-factor vector $p_u \in \mathbb{R}^f$ and each item i with an item-factor vector $q_i \in \mathbb{R}^f$. The vectors' dimensionality or number of factors is given by f, in which each element $p_{u,f}$ or $q_{i,f}$ corresponds to a latent feature of a preference-relevance model, i.e., how much user u likes a particular feature in item i, and simultaneously, how much such characteristic is relevant to item i.

We describe this technique by first introducing the basic prediction rule:

$$\hat{r}_{ui} = b_{ui} + p_u^T q_i \quad , \tag{5}$$

where the parameters are estimated by the minimization of the associated squared error function:

$$\min_{p_*, q_*, b_*} \sum_{(u,i) \in K} (r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda(b_u^2 + b_i^2 + ||p_u||^2 + ||q_i||^2) \quad . \tag{6}$$

Based on Equation 5, Koren extended this basic model to consider implicit information. In fact, he used an additional factor vector $y_i \in \mathbb{R}^f$ and also considered set $N(u)$, which contains all items for which u provided an implicit preference (for instance, u visited a product, but did not provide a rating). Therefore, the SVD++ model is defined as

$$\hat{r}_{ui} = b_{ui} + q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) \quad . \tag{7}$$

The preferences of a user u are represented by a combination of explicit and implicit information. The user-factor vector p_u is learnt from the given explicit ratings and complemented by the sum of y_j, which represents the implicit feedback. Only after this adjustment can the user-factors be interacted with the item-factors. Again, the parameters are learnt through the minimization of the associated squared error function by gradient descent. For more details about this and other methods and the way of solving the least squares problems above by gradient descent optimization, readers may refer to [Koren, 2008, Koren, 2010, Funk, 2006, Paterek, 2007].

4 Proposed Method

The previous section described some models that integrate different types of feedback towards an accurate user representation. However, the latent factor approaches do not take into account a simpler and straightforward type of information usually associated with users and items, i.e., the metadata themselves. Descriptions, such as user gender, age, genres of movies, list of actors, keywords and producers are not considered in the models so that the recommendation results can be improved. We propose a generic latent factor model able to exploit the available metadata associated with both users and items.

The next subsections describe the proposed model in details, by gradually incorporating the various components that constitute the final schema.

4.1 Baseline Revisited

Subsection 3.2 described the baseline estimates that model systematic tendencies according to users' and items' intrinsic characteristics. Such an idea can be slightly improved through the incorporation of the global effects of how users rate items depending on personal data or items' metadata. For instance, children may rate an item differently from adults. Similarly, action movies may be rated on a higher (or lower) scale than romance.

In order to model such possibilities, we have extended the baseline estimates in Equation 1 by also considering personal information and items' descriptions:

$$b_{ui}^{demo} = \mu + b_u + b_i + \sum_{z_u \in Z(u)} |G(u; z_u)|^{-\alpha} \sum_{d \in G(u; z_u)} b_d + \sum_{z_i \in Z(i)} |G(i; z_i)|^{-\alpha} \sum_{g \in G(i; z_i)} b_g \quad . \tag{8}$$

In this case, $Z(u)$ and $Z(i)$ are, respectively, the sets of different types of users' and items' information considered in the system, and $G(u; z_u)$ and $G(i; z_i)$ represent all information of types z_u and z_i associated with user u

and item i. An example of this representation is the use of metadata information from the MovieLens dataset[5]. We denote $Z(u)$ = {occupation, age group, gender, zip code}, and when z_u = occupation of a user u, we denote $G(u; z_u)$ = {programmer}, for instance. Similarly, considering all metadata types available in the dataset, we denote $Z(i)$ = {title, genre, year of release, IMDB URL}; when z_i = genre of an item i, we denote $G(i; z_i)$ = {action, science fiction}, for instance. In the aforementioned example, although $|G(u; z_u)| = 1$ in most cases, we preferred to keep generality by using more than one piece of information associated with z_u. The regularization constant α from Equation 8 is set to 1 when metadata are available, and 0 otherwise.

The contextual biases b_d and b_g can be estimated by solving a least squares problem similar to Equation 4. In the experiments reported in Section 5, we employed a simple gradient descent scheme using the observed data to change the parameters in the opposite direction of the gradient [Koren, 2010, Paterek, 2007].

4.2 Item's Metadata Incorporation

In addition to the global effects modeled by our extended baseline estimates, we aimed to model the associations between the personal data and the content metadata available for each item. Such an approach is important because, depending on the actual contextual environment, demographic data or personal interests, users may prefer to visit items related to specific subjects. For instance, a 7-year-old user will certainly prefer children's films; female users will probably enjoy romance and drama; a group of cyclists will enjoy sports and adventure content.

In order to capture such associations between users' and items' metadata, we have incorporated another set of parameters h_{dg} to Equation 8, as follows:

$$\hat{r}_{ui}^{meta} = b_{ui}^{demo} + \sum_{z_i \in Z(i)} |G(i; z_i)|^{-\alpha} \sum_{z_u \in Z(u)} |G(u; z_u)|^{-\alpha} \sum_{d \in G(u; z_u)} \sum_{g \in G(i; z_i)} h_{dg} \ . \tag{9}$$

The parameters represented by h_{dg} capture the weights of a user's personal information d associated with an item's description g. Again, such weights are learnt from the observed data through gradient descent.

4.3 Latent Factors

Significant improvement can be achieved from the previous models if we adopt a matrix factorization scheme that maps users and items into a joint latent factor space of dimensionality f. Incorporating Equation 5 to our model, we have:

[5] http://www.grouplens.org/node/73

$$\hat{r}_{ui} = \hat{r}_{ui}^{meta} + p_u^T q_i \quad . \tag{10}$$

Now, for a given item i, vector q_i represents the relevance of each factor to the item itself, which can be positive or negative. For a given user u, vector p_u measures the importance or relevance of those factors to the user's preferences. The user's overall interest in the item's features can be captured by multiplying both vectors. By summing the result to $\hat{r}_{ui}^{meta}$, we can incorporate the biases and descriptions of u and i into a latent factor model.

4.4 User's Implicit Feedback

Subsection 3.3 described SVD++ [Koren, 2008, Koren, 2010], which can integrate explicit and implicit feedback from users into a unique model. Prior to the inner product of the factors, vector p_u is enhanced with implicit feedback from user u, represented by parameter y_j.

This subsection addresses the incorporatation of SVD++ into our latent factors model also for the capture of implicit feedback regarding users. By using latent factors, the user's preferences for different features can be captured, which characterizes the whole item subject to be recommended. On the other hand, by using implicit feedback, additional user's preferences can be captured even if a few ratings have been provided.

Concretely, our enhanced model is defined by the following equation:

$$\hat{r}_{ui} = \hat{r}_{ui}^{meta} + q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) \quad . \tag{11}$$

The items' metadata and users' personal features are added as global effects into the baselines, and, at the same time, the users factors are enhanced with implicit feedback information.

4.5 gSVD++

The SVD++ model (Equation 7) and its extension described in the last subsection (Equation 11) have a characteristic in common, i.e., vector q_i representing the item factors is not enhanced with any additional information. However, additional improvements can be achieved if we incorporate a symmetric items modeling using their metadata when available.

This subsection describes another prediction rule, but users' and items' metadata are not modeled as global effects. Such an approach will be useful i) to describe the unified model in the next subsection, in which all features are combined, and ii) for the evaluation of each module separately, as shown in Section 5.

We denominate this particular extension "gSVD++" [Manzato, 2013]. Let us denote set $G(i; z_i)$, which contains the descriptions of type z_i associated with item i. For instance, the genres can be considered metadata ($Z(i) = \{\text{genre}\}$), consequently, a particular movie i would have set $G(i; z_i)$ storing the related categories, such as action, adventure and science fiction. Furthermore, let us denote a metadata factors vector $x_g \in \mathbb{R}^f$ containing the factors for each possible description. Equation 7 could be rewritten to complement the items factor q_i with the available metadata, as follows:

$$\hat{r}_{ui} = b_{ui} + \left(q_i + \sum_{z_i \in Z(i)} |G(i; z_i)|^{-\alpha} \sum_{g \in G(i; z_i)} x_g \right)^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) . \tag{12}$$

Both user and item factors have been enhanced with implicit feedback. The original Koren's solution considered the y_j factor vector to represent the indirect user's information (e.g. rental history). Such an approach was extended by the incorporatation of another factor vector x_g to represent the item's metadata (e.g. genres). In this metadata awareness model each item set of descriptions $G(i; z_i)$ contributes to adjust the importance of each information g, represented by a factor vector. The weights of this vector are determined through the observation of the known ratings during training. Regularization constant α from Equation 12 is set to 1 when there are metadata associated with item i, and 0 otherwise. When no metadata are available, the model equals SVD++.

4.6 Unified Model

According to the extensions made so far, this subsection describes a unified model which integrates different aspects related to users and items towards an accurate prediction rule. Such a strategy integrates five functionalities described in the previous models:

1. the global effects from users and items through the baseline estimates;
2. the global effects from users' personal information and items' metadata, as defined by our baseline extension;
3. the association between users' and items' metadata that models how particular users rate specific items;
4. the capture of the overall interest of a user in a particular item by means of latent factors;

5. the implicit feedback from users (e.g. rental history) and items (metadata), according to the adjustments made in the users' and items' latent factors prior to their inner product.

The unified model is defined by the following equation:

$$\hat{r}_{ui} = \hat{r}_{ui}^{meta} + \left(q_i + \sum_{z_i \in Z(i)} |G(i; z_i)|^{-\alpha} \sum_{g \in G(i;z_i)} x_g \right)^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) . \tag{13}$$

Similarly to the previous formulations, the parameters are learnt through the minimization of the regularized squared error function associated with Equation 13, as follows:

$$\begin{aligned} \min_{b_*, h_*, p_*, q_*, x_*, y_*} \sum_{(u,i) \in K} \Bigg(r_{ui} - \mu - b_u - b_i \\ - \sum_{z_u \in Z(u)} |G(u; z_u)|^{-\alpha} \sum_{d \in G(u;z_u)} b_d - \sum_{z_i \in Z(i)} |G(i; z_i)|^{-\alpha} \sum_{g \in G(i;z_i)} b_g \\ - \sum_{z_i \in Z(i)} |G(i; z_i)|^{-\alpha} \sum_{z_u \in Z(u)} |G(u; z_u)|^{-\alpha} \sum_{d \in G(u;z_u)} \sum_{g \in G(i;z_i)} h_{dg} \\ - \left(q_i + \sum_{z_i \in Z(i)} |G(i; z_i)|^{-\alpha} \sum_{g \in G(i;z_i)} x_g \right)^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) \Bigg)^2 \\ + \lambda \Bigg(b_u^2 + b_i^2 + ||p_u||^2 + ||q_i||^2 + \sum_{z_u \in Z(u)} \sum_{d \in G(u;z_u)} b_d^2 + \sum_{z_i \in Z(i)} \sum_{g \in G(i;z_i)} b_g^2 \\ + \sum_{z_u \in Z(u)} \sum_{z_i \in Z(i)} \sum_{d \in G(u;z_u)} \sum_{g \in G(i;z_i)} h_{dg}^2 + \sum_{z_i \in Z(i)} \sum_{g \in G(i;z_i)} x_g^2 + \sum_{j \in N(u)} y_j^2 \Bigg) . \end{aligned} \tag{14}$$

We employ a simple gradient descent scheme to solve the system indicated in Equation 14 [Koren, 2008, Koren, 2010, Paterek, 2007]. Let us consider $e_{ui} \stackrel{\text{def}}{=} r_{ui} - \hat{r}_{ui}$. Using the training dataset, we loop over all known ratings in K. For a given training example r_{ui}, we change the parameters by moving them in the opposite direction of the gradient, as illustrated in Algorithm 1.

5 Evaluation

The evaluation consists in comparing our model with other methods available in the literature. The different modules are also evaluated to check the contribution of each aspect to the final recommendation improvement.

ALGORITHM 1: Learning the factorized model through gradient descent.

Input: Set of known ratings $(u, i) \in K$
Output: Learnt parameters $b_u, b_i, b_d, b_g, h_{dg}, p_u, q_i, x_g, y_j$

```
for count = 1,...,#Iter. do
    foreach (u,i) ∈ K do
        r̂_ui ← Prediction according to Equation 13;
        e_ui ← r_ui − r̂_ui;
        b_u ← b_u + γ(e_ui − λ1 b_u);
        b_i ← b_i + γ(e_ui − λ2 b_i);
        foreach z_u ∈ Z(u) do
            foreach d ∈ G(u; z_u) do
                b_d ← b_d + γ(e_ui − λ3 b_d);
            end
        end
        foreach z_i ∈ Z(i) do
            foreach g ∈ G(i; z_i) do
                b_g ← b_g + γ(e_ui − λ3 b_g);
            end
        end
        foreach z_u ∈ Z(u) do
            foreach z_i ∈ Z(i) do
                foreach d ∈ G(u; z_u) do
                    foreach g ∈ G(i; z_i) do
                        h_dg ← h_dg + γ(e_ui |G(u; z_u)|^−α |G(i; z_i)|^−α − λ3 h_dg);
                    end
                end
            end
        end
        p_u ← p_u + γ2(e_ui(q_i + Σ_{z_i∈Z(i)} |G(i; z_i)|^−α Σ_{g∈G(i;z_i)} x_g) − λ4.p_u);
        q_i ← q_i + γ2(e_ui(p_u + |N(u)|^−1/2 Σ_{j∈N(u)} y_j) − λ5.q_i);
        foreach j ∈ N(u) do
            y_j ← y_j + γ2(e_ui |N(u)|^−1/2 (q_i + |G(i)|^−α Σ_{g∈G(i)} x_g) − λ6.y_j);
        end
        foreach z_i ∈ Z(i) do
            foreach g ∈ G(i; z_i) do
                x_g ← x_g + γ2(e_ui |G(i; z_i)|^−α (p_u + |N(u)|^−1/2 Σ_{j∈N(u)} y_j) − λ7.x_g);
            end
        end
    end
    γ ← γ ∗ 0.9;
    γ2 ← γ2 ∗ 0.9;
end
```

5.1 Dataset

The experiments were conducted with the well-known MovieLens 100k dataset. It consists of 943 users, who assigned 100k ratings to 1682 movies. The dataset also provides metadata of the users and items. With respect to users, demographic data, such as age, occupation, gender and Zip code are provided. All types of demographic data, except Zip code, were considered in the evaluation.

Information about the users' age was pre-processed so that users of the same age group could be clustered. Table 2 was configured experimentally to associate the same information with all users belonging to that group.

Table 2: Configuration of age group.

Age	**Group**
0 – 12	children
13 – 18	teenagers
19 – 25	young adults
26 – 35	adults
36 – 55	mature
56 – 60	aged
Over 61	elderly

As a result, set $Z(u)$ is composed of $\{$age group, occupation, gender$\}$ and sets represented by $G(u, z_u)$ have size 1.

Regarding the items' metadata, the MovieLens dataset provides the movie title, date of release, Internet Movies Database (IMDB)[6] URL and set of associated genres. In our experiments, we considered only the genres as items' metadata, consequently, $|Z(i)| = 1$, and $|G(i, z_i)| \geq 1$, because one or more genres can be assigned to each movie i. In this version of the dataset, there are 19 different genres, all of them considered in the evaluation.

5.2 Methodology

This evaluation compares the different modules of the proposed model to check the contribution of each aspect to the final recommendation accuracy. In addition, we also compare the final solution against three related methods available in the literature:

- Biased MF: an algorithm proposed by Rendle & Schmidt-Thie [Rendle and Lars, 2008] that reduces the cold start by deriving an online-update algorithm for regularized kernel matrix factorization models. It is also flexible for nonlinear interactions among feature vectors.
- SVD++: an algorithm proposed by Koren [Koren, 2008, Koren, 2010] that integrates explicit and implicit feedback from users into a latent factors model.

[6] http://www.imdb.com/

– Factorization Machines (FM): a technique proposed by Rendle [Rendle, 2012a, Rendle et al., 2011] that estimates interactions among categorical variables combining features engineering and factorization models.

We used libFM [Rendle, 2012a] and MyMediaLite [Gantner et al., 2011] libraries to implement the methods and the results were measured as follows:

1. the prediction accuracies of all techniques were compared in terms of RMSE (root mean squared error) and MAE (mean average error) according to a varying number of factors ($f = \{10, 50, 100, 150, 200, 250, 300\}$);

2. for each technique, the best number of factors was selected, i.e., the value of f for which RMSE and MAE computed in the previous step was minimal.

Both traditional RMSE (root mean squared error) and MAE (mean squared error) metrics [Ricci et al., 2011] provide information about the average magnitude of the errors. However, they differ from each other because the former is a quadratic scoring which gives a relatively high weight to large errors, and the latter is a linear scoring that gives equal weights to all error predictions. In all experiments, a 10-fold cross-validation procedure was used to separate the dataset into training and test sets. We trained the model using the training set and then evaluated the prediction accuracy with the test set. After ten executions varying the disjoint composition of the sets, the average RMSE and MAE were returned. A two-sided paired t-test with 95% confidence level was used to compare the final prediction rule (unified model) against the three baselines reported in the literature (Biased MF, SVD++ and FM).

5.3 Comparison of Baselines

The baseline estimates were compared with and without demographic data and items' metadata. The involved constants were defined experimentally, as shown in Table 3. Details of their utilization can be found in Algorithm 1, as previously explained.

Table 4 shows the results and corresponding standard deviation. The traditional baseline represented by Equation 1 achieved 0.9437 of RMSE, as it considered only the user and item's biases. When demographic data and items' metadata in terms of global effects were added, the RMSE decreased to 0.9396. Details of this algorithm can be found in Subsection 4.1, Equation 8. Additional improvement was achieved when items' metadata associated with the users' personal information were incorporated into the model, i.e., RMSE was reduced to 0.9380. The algorithm is depicted in Subsection 4.2 and models how users with certain demographic data (e.g. children) rate items with particular genres (e.g. cartoon).

Table 3: Constants used in the evaluation of baselines.

Constant	Value
#Iter.	50
γ	0.02
λ_1	$0.025 * \vert R(u)\vert^{-\frac{1}{2}}$
λ_2	$0.025 * \vert R(i)\vert^{-\frac{1}{2}}$
λ_3	0.025

Table 4: Baselines' results. All values are statistically significant in comparison to the traditional baseline (p-value < 0.05).

Algorithm	RMSE	MAE
Traditional Baseline (Equation 1)	0.9437 ± 0.0061	0.7480 ± 0.0052
Baseline Revisited (Equation 8)	0.9396 ± 0.0059	0.7424 ± 0.0050
Item's Metadata (Equation 9)	0.9380 ± 0.0060	0.7408 ± 0.0051

5.4 Comparison of Latent Factors Approaches

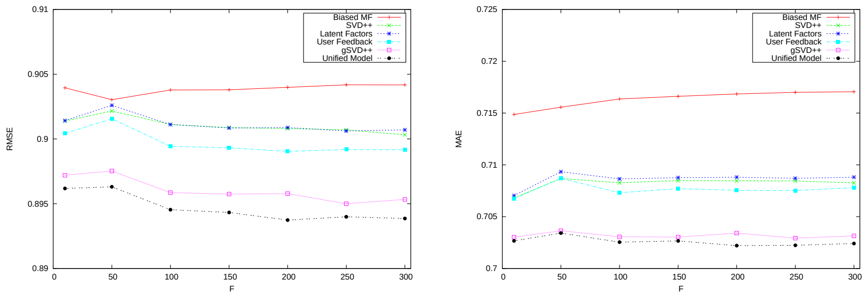
Our unified model was compared against its isolated modules and the related models depicted in Subsection 5.2. The modules that compose the proposed unified model, i.e., Biased MF (Equation 5), SVD++ (Equation 7), Latent Factors (Equation 10), User Feedback (Equation 11), gSVD++ (Equation 12) and Unified Model (Equation 13) were grouped into one single comparison. The set of constants defined in Table 5 was used in the experiment and the details of their utilization can be found in Algorithm 1. Tables 6 and 7 show the RMSE and MAE results and corresponding standard deviation, according to a varying number of factors. The values in bold indicate the best results for each technique using a particular number of factors. The RMSE and MAE values of the proposed unified model were statistically significant in comparison to Biased MF and SVD++ (p-value < 0.05), which are the baselines proposed elsewhere [Rendle and Lars, 2008, Koren, 2008, Koren, 2010].

Figure 1 illustrates the same results. The RMSE and MAE of the proposed unified model (Equation 13) clearly outperformed the related techniques and isolated modules, as it incorporated and combined many aspects related to users and items. Such aspects are described in Subsection 4.6 and include demographic and metadata information, implicit feedback and latent factors.

Biased MF showed the worst accuracy, mainly because it considers only simple matrix factorization to characterize the users' preferences and items' features. Latent Factors and SVD++ achieved a better score, which is a reflection of the combination of implicit feeedback or demographic and metadata information in terms of global effects into a latent factor approach. The comparison

Table 5: Constants used in the evaluation of latent factors approaches.

Constant	Value
#Iter.	50
γ	0.007
γ_2	0.1
λ_1	$0.05 * \|R(u)\|^{-\frac{1}{2}}$
λ_2	$0.05 * \|R(i)\|^{-\frac{1}{2}}$
λ_3	0.05
λ_4	$\|R(u)\|^{-\frac{1}{2}}$
λ_5	$\|R(i)\|^{-\frac{1}{2}}$
λ_6	$\|N(j)\|^{-\frac{1}{2}}$
λ_7	$\|G(i, z_i)\|^{-1}$



(a) (b)

Figure 1: Comparison among Biased MF (Equation 5), SVD++ (Equation 7), Latent Factors (Equation 10), User Feedback (Equation 11), gSVD++ (Equation 12) and Unified Model (Equation 13).

between Latent Factors and SVD++ shows the observed gains were not statistically significant. The difference is Latent Factors do not use implicit feedback, but incorporate metadata information, and SVD++ uses implicit feedback, but does not consider metadata information. The next module, i.e. User Feedback, combines both types of information into a latent factor approach, which results in better RMSE and MAE for all numbers of factors.

The comparison between SVD++ and gSVD++ shows an improvement of gSVD++ over the previous Koren's model, which suggests the effectiveness of incorporating metadata awareness into a latent factor model that supports implicit feedback. In the case of gSVD++, we only added parameter x_g to enhance the item factors with metadata, as described in Subsection 4.5.

The unified model achieved the best accuracy regardless of the number of fac-

Table 6: RMSE comparison among Biased MF (Equation 5), SVD++ (Equation 7), Latent Factors (Equation 10), User Feedback (Equation 11), gSVD++ (Equation 12) and Unified Model (Equation 13).

Method	$f = 10$	$f = 50$	$f = 100$	$f = 150$	$f = 200$	$f = 250$	$f = 300$
Biased MF	0.9039 ±0.0070	0.9030 ±0.0068	0.9038 ±0.0065	0.9038 ±0.0068	0.9040 ±0.0066	0.9042 ±0.0064	0.9042 ±0.0065
SVD++	0.9014 ±0.0065	0.9022 ±0.0067	0.9011 ±0.0067	0.9009 ±0.0067	0.9008 ±0.0065	0.9007 ±0.0067	**0.9003 ±0.0070**
Latent Factors	0.9014 ±0.0071	0.9026 ±0.0071	0.9011 ±0.0070	0.9008 ±0.0068	0.9009 ±0.0070	**0.9006 ±0.0068**	0.9007 ±0.0070
User Feedback	0.9004 ±0.0070	0.9016 ±0.0064	0.8994 ±0.0068	0.8993 ±0.0071	**0.8990 ±0.0065**	0.8992 ±0.0068	0.8992 ±0.0069
gSVD++	0.8972 ±0.0049	0.8975 ±0.0051	0.8959 ±0.0050	0.8957 ±0.0053	0.8958 ±0.0049	**0.8950 ±0.0047**	0.8953 ±0.0052
Unified Model	0.8962 ±0.0060	0.8963 ±0.0056	0.8945 ±0.0061	0.8943 ±0.0057	**0.8937 ±0.0052**	0.8940 ±0.0052	0.8939 ±0.0054

Table 7: MAE comparison among Biased MF (Equation 5), SVD++ (Equation 7), Latent Factors (Equation 10), User Feedback (Equation 11), gSVD++ (Equation 12) and Unified Model (Equation 13).

Method	$f = 10$	$f = 50$	$f = 100$	$f = 150$	$f = 200$	$f = 250$	$f = 300$
Biased MF	**0.7149 ±0.0056**	0.7156 ±0.0056	0.7164 ±0.0053	0.7166 ±0.0055	0.7168 ±0.0054	0.7170 ±0.0053	0.7171 ±0.0053
SVD++	**0.7068 ±0.0054**	0.7087 ±0.0056	0.7083 ±0.0054	0.7085 ±0.0055	0.7085 ±0.0054	0.7084 ±0.0056	0.7083 ±0.0056
Latent Factors	**0.7070 ±0.0056**	0.7093 ±0.0056	0.7086 ±0.0053	0.7088 ±0.0052	0.7088 ±0.0057	0.7087 ±0.0054	0.7088 ±0.0054
User Feedback	**0.7067 ±0.0059**	0.7087 ±0.0054	0.7073 ±0.0052	0.7077 ±0.0057	0.7075 ±0.0050	0.7075 ±0.0053	0.7078 ±0.0057
gSVD++	0.7030 ±0.0043	0.7036 ±0.0047	0.7031 ±0.0043	0.7030 ±0.0045	0.7034 ±0.0045	**0.7029 ±0.0044**	0.7031 ±0.0047
Unified Model	0.7027 ±0.0050	0.7034 ±0.0047	0.7025 ±0.0054	0.7027 ±0.0048	**0.7022 ±0.0042**	0.7022 ±0.0042	0.7024 ±0.0047

tors. As explained in Subsection 4.6, the unified approach combines gSVD++ and User Feedback into a single model, consequently, it can capture most aspects of the recommendation process, including user implicit feedback, demographic information and metadata. As it provided the best results, we can infer each separate technique contributes to a different aspect of the whole system composed of users and items with distinguished characteristics.

5.5 Comparison against Factorization Machines

The proposed unified model was compared against other techniques which also integrate different features into a same prediction rule. Factorization Machines

(FM) [Rendle, 2012a, Rendle et al., 2011] were chosen as a baseline, as they model the features by means of categorical variables, as accomplished in other machine-learning techniques. The main difference, however, is they use factorized interactions among the variables to learn the hyperparameters.

The libFM library [Rendle, 2012a] was configured as follows. The input data were composed of users, items, genres, age group, gender and occupation. Similarly to [Rendle et al., 2011], we modeled the genre information as a boolean vector, as each item may be associated with more than one genre; age group, gender and occupation were modeled using a particular integer value for each datum. We set the number of interactions to 20, learning rate to 0.002 and initial standard deviation to 0.01. All these constants were defined experimentally, following the guidelines described in [Rendle, 2012a]. We adopted stochastic gradient descent (SGD) with adaptive regularization [Rendle, 2012b] to learn the hyperparameters.

After executing FM with a varying number of factors, the best RMSE and MAE results were achieved when 10 dimensions were used for the number of factors. Table 8 shows the comparison between FM and the unified model. The latter was configured according to the best results achieved in the last experiment (200 dimensions, as shown in bold in Tables 6 and 7).

The proposed unified model achieved statistically significant improvements over factoriztion machines. Therefore, the use of different types of information in a generic model, such as FM, was not efficient because different data may influence the recommendation process in distinct ways. Because FM will interact with all features interchangeably, information types, such as implicit feedback and users' and items' metadata will have the same importance in the model. On the other hand, our technique deals with each type of datum in a particular strategy, which improves the prediction accuracy.

Table 8: RMSE and MAE comparison between Unified Model and Factorization Machines.

Technique	***f***	**RMSE**	**MAE**
FM	10	0.9597 ± 0.0237	0.7542 ± 0.0133
Unified Model	200	0.8937 ± 0.0052	0.7022 ± 0.0042

6 Final Remarks

This article proposed a hybrid filtering approach which integrates different aspects of the whole system composed of users and items with distinguished characteristics into a single recommendation rule. The model is based on latent factors, but it also considers metadata awareness to detect strong associations among items and users closely related.

In view of the importance of integrating different ways of user's inputs [Bell and Koren, 2007], our model is generic enough to include as many pieces of information as necessary, depending on the application needs, but at the same time, keeping the identity of each information type and restricting its collaboration when recommendations are computed. This is the main difference of the proposed technique in comparison to Factorization Machines, as the latter processes all information types interchangeably as single input variables.

The generality of our model can be useful for many application scenarios, including group recommendations. For example, given a scenario where the system would recommend items to a group of friends from the church, one could label all individuals as belonging to that group and incorporate such information as an additional attribute type into set $Z(u)$. Later, this attribute would be combined with the content metadata to highlight items whose subject is church-related, such as gospel, saints, etc.

The related techniques were evaluated through the initialization of the involved parameters with items' genres and users' demographics and rental history available in the MovieLens dataset. Our analysis showed each module of the unified model can improve the results in certain aspects. The unified model was also compared against other techniques available in the literature, and the results show that the proposed model achieved the best prediction accuracy.

The contributions of this article are:

1. a mechanism that combines users' and items' descriptions, including a training algorithm to learn the importance of each parameter to the whole recommendation process;

2. several extensions to tackle different limitations of the SVD++ model [Koren, 2008] and capture different aspects of the system, including metadata and demographics awareness, implicit feedback, users' personal information and baselines;

3. incorporation of the extensions into a single and unified model generic enough to deal with different types of information regarding users and items.

As future work, we aim at evaluating the model with other types of user's personal information and item's metadata to find the information that is more relevant to the final filtering procedure. We also plan to extend the technique to incorporate contextual information, which is another source of user's input that should be considered towards further improvements in the prediction accuracy.

Acknowledgments

The authors would like to thank FAPESP, CNPq and CAPES for the financial support.

References

[Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.

[Agarwal and Chen, 2009] Agarwal, D. and Chen, B.-C. (2009). Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 19–28, New York, NY, USA. ACM.

[Agichtein et al., 2006] Agichtein, E., Brill, E., and Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 19–26, New York, NY, USA. ACM.

[Bell and Koren, 2007] Bell, R. M. and Koren, Y. (2007). Lessons from the netflix prize challenge. *SIGKDD Explor. Newsl.*, 9(2):75–79.

[Bobadilla et al., 2013] Bobadilla, J., Ortega, F., Hernando, A., and Gutiérez, A. (2013). Recommender systems survey. *Know.-Based Syst.*, 46:109–132.

[Chen and He, 2009] Chen, T. and He, L. (2009). Collaborative filtering based on demographic attribute vector. In *Future Computer and Communication, 2009. FCC '09. International Conference on*, pages 225–229.

[Da Costa et al., 2014] Da Costa, A., Domingues, M., Rezende, S., and Manzato, M. (2014). Improving personalized ranking in recommender systems with multimodal interactions. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 198–204.

[Funk, 2006] Funk, S. (2006). Netflix Update: Try This At Home. `http://sifter.org/~simon/journal/20061211.html`. [Online; accessed 26-September-2012].

[Gantner et al., 2010] Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., and Schmidt-Thieme, L. (2010). Learning attribute-to-feature mappings for cold-start recommendations. In *2010 IEEE 10th International Conference on Data Mining (ICDM)*, pages 176–185.

[Gantner et al., 2011] Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2011). MyMediaLite: A free recommender system library. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 305–308, New York, NY, USA.

[Herlocker et al., 2002] Herlocker, J., Konstan, J. A., and Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.*, 5(4):287–310.

[Hu et al., 2008] Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, Washington, DC, USA. IEEE Computer Society.

[Jawaheer et al., 2014] Jawaheer, G., Weller, P., and Kostkova, P. (2014). Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Trans. Interact. Intell. Syst.*, 4(2):8:1–8:26.

[Joachims et al., 2005] Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 154–161, New York, NY, USA. ACM.

[Koren, 2008] Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 426–434, New York, NY, USA. ACM.

[Koren, 2010] Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data*, 4(1):1:1–1:24.

[Koren et al., 2009] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

[Krulwich, 1997] Krulwich, B. (1997). Lifestyle finder: Intelligent user profiling using large-scale demographic data. 18(2):37–45.

[Lee et al., 2002] Lee, M., Choi, P., and Woo, Y. (2002). A hybrid recommender system combining collaborative filtering with neural network. *Lecture Notes in Computer Science*, 2347 LNCS:531–534.

[Manzato, 2012] Manzato, M. G. (2012). Discovering Latent Factors from Movies Genres for Enhanced Recommendation. In *Proceedings of the 6th ACM Conferece on Recommender Systems*, RecSys '12, pages 249–252, New York, NY, USA.

[Manzato, 2013] Manzato, M. G. (2013). gSVD++: supporting implicit feedback on recommender systems with metadata awareness. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 908–913, New York, NY, USA. ACM.

[Manzato et al., 2014] Manzato, M. G., Domingues, M. A., Marcacini, R. M., and Rezende, S. O. (2014). Improving personalized ranking in recommender systems with topic hierarchies and implicit feedback. In *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR 2014)*.

[Oard and Kim, 1998] Oard, D. and Kim, J. (1998). Implicit feedback for recommender systems. In *Proceedings of the AAAI Workshop on Recommender Systems*, pages 81–83.

[Paterek, 2007] Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In *Proc. KDD Cup Workshop at SIGKDD'07, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining*, pages 39–42.

[Pazzani, 1999] Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13:393–408.

[Rendle, 2012a] Rendle, S. (2012a). Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22.

[Rendle, 2012b] Rendle, S. (2012b). Learning recommender systems with adaptive regularization. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 133–142, New York, NY, USA. ACM.

[Rendle et al., 2009] Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States. AUAI Press.

[Rendle et al., 2011] Rendle, S., Gantner, Z., Freudenthaler, C., and Schmidt-Thieme, L. (2011). Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 635–644, New York, NY, USA. ACM.

[Rendle and Lars, 2008] Rendle, S. and Lars, S.-T. (2008). Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 251–258, New York, NY, USA. ACM.

[Ricci et al., 2011] Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors (2011). *Recommender Systems Handbook*. Springer.

[Shi et al., 2013] Shi, Y., Larson, M., and Hanjalic, A. (2013). Mining contextual movie similarity with matrix factorization for context-aware recommendation. *ACM Trans. Intell. Syst. Technol.*, 4(1):16:1–16:19.

[Shi et al., 2014] Shi, Y., Larson, M., and Hanjalic, A. (2014). Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1):3:1–3:45.

[Stern et al., 2009] Stern, D. H., Herbrich, R., and Graepel, T. (2009). Matchbox: Large scale online bayesian recommendations. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 111–120, New York, NY, USA. ACM.

[Vozalis and Margaritis, 2007] Vozalis, M. and Margaritis, K. (2007). Using SVD and demographic data for the enhancement of generalized collaborative filtering. *Information Sciences*, 177(15):3017 – 3037.

[Yang et al., 2012] Yang, D., Chen, T., Zhang, W., Lu, Q., and Yu, Y. (2012). Local implicit feedback mining for music recommendation. In *Proceedings of the 6th ACM Conference on Recommender Systems*, RecSys '12, pages 91–98, New York, NY, USA. ACM.

[Yapriady and Uitdenbogerd, 2005] Yapriady, B. and Uitdenbogerd, A. (2005). Combining demographic data with collaborative filtering for automatic music recommendation. In Khosla, R., Howlett, R., and Jain, L., editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 3684 of *Lecture Notes in Computer Science*, pages 201–207. Springer Berlin Heidelberg.