IADIS International Journal on WWW/Internet
Vol. 12, No. 2, pp. 149-167
ISSN: 1645-7641

EVALUATING THE IMPACT OF DEMOGRAPHIC DATA ON A HYBRID RECOMMENDER MODEL

Edson B. Santos Junior. *Institute of Mathematics and Computer Science – Sao Paulo University (ICMC-USP), 400 – Trabalhador São-carlense Ave., São Carlos – SP, Brazil*

Marcelo Garcia Manzato. *Institute of Mathematics and Computer Science – Sao Paulo University (ICMC-USP), 400 – Trabalhados São-carlense Ave., São Carlos – SP, Brazil*

Rudinei Goularte. *Institute of Mathematics and Computer Science – Sao Paulo University (ICMC-USP), 400 – Trabalhados São-carlense Ave., São Carlos – SP, Brazil*

ABSTRACT

One of the major challenges in Recommender Systems is how to predict users' preferences in a group context. There are situations in which a user could be recommended an item appropriated for one of their groups, but the same item may not be suitable when interacting with another group. There are situations in which a user could be recommended an item appropriated for one of their groups (e.g. poetry friends), but the same item may not be suitable when interacting with another group (e.g. soccer team). We note that recommender systems should try to satisfy the group's demands, but it should also respect the user's individuality. The demographic data are an effective way to consider users' characteristics, enabling analysis about group of users and their contextual constraints. In our past work we have proposed a multifaceted hybrid recommender model, which integrates a set of different user's inputs into a unified and generic latent factor model, achieving better results than the other state-of-the-art approaches. The recommender exploits users' demographics, such as age, gender and occupation, along with implicit feedback and items' metadata. Depending on the personal information from users, the recommender selects content whose subject is semantically related to their interests. In this paper we evaluate our model, aiming to analyze the impact of demographic data on hybrid recommenders, in order to understand how these systems are beavering in terms of the recommendation accuracy, precision and recall, computational cost and improvements of technical results.

KEYWORDS

Group recommender systems, matrix factorization, collaborative filtering, demographic data, implicit feedback, latent factor model.

1. INTRODUCTION

The increase in data availability on the Web has required additional efforts for the development and enhancement of Recommender Systems (RS). These tools assist in the Information Retrieval field, being an important branch of the Personalization research field, which aims to help users to find content in large data sources meeting individual needs [Ricci et al., 2011] and generate personalized recommendations. Consequently, companies as Google1, Facebook2 and NetFlix3 make significant investments on the creation and deployment of Collaborative Systems.

Two traditional recommendation mechanisms are reported in the literature: content-based and collaborative [Adomavicius and Tuzhilin, 2005]. In the content-based approach, the data are selected by attribute associations, where each item is structured of its own characteristics. However, if only metadata are considered, there may occur overspecialization, not allowing users to receive new and/or diverse content, in particular when their profile is restricted to descriptions of similar items that were visited [Burke, 2002].

In the Collaborative Filtering (CF) approach, two models are studied: neighborhood and latent factors. In the first case, clusters of items are formed to recommend items which are similar to the ones preferred by users in the past. In the second one, the recommendation can be computed by uncovering latent associations among users or items. An alternative path is to transform both items and users into the same latent factor space, allowing them to be directly comparable [Bell and Koren, 2007]. It is also possible to reduce the dimensionality of the matrices by gathering only the most relevant information associated with the users and items.

In the real-world recommendation scenario, suggestions for a group of users are even more complicated, since managing the variety of profiles and relationships is extensive due to the dynamics of memberships. A challenge is how to classify users in a network [Ricci et al., 2011], where there are situations which users may be interested to receiving an specific content, but his group maybe not. Group recommenders should aim at satisfying the group's demands, respecting the user individuality. For example, the interests shared with group members could be a correct recommendation at an inopportune moment. Under these circumstances users do not need or do not want to receive these contents, since the system knows their preferences, but does not know when to recommend to them [Ricci et al., 2011].

Considering the different user's input ways to construct an accurate profile, based on the benefits of content-based and CF, hybrid recommenders are a good alternative for RS. However, recent works which exploit the latent factor model as recommender basis does not considers neither metadata nor demographic data associated to the content. In fact, semantic and known descriptions of users and items could be added into the recommendation process, in contrast to the obscure and incomprehensive relations of latent factors. Moreover, the most of group RS only analyze interactions based on the trivial: user–link–item [Ricci et al., 2011]. In this case, the relationships among different nodes in a network are not analyzed, hindering to acquire relations data from users who have different preferences. In face of the given scenario, we argue that additional demographic data are an effective alternative in order to enable systems to discover and analyze contextual constraints in a real-world scenario.

[1] http://www.google.com/
[2] http://www.facebook.com/
[3] http://www.netflix.com/

In past works we have presented a multifaceted recommender, in order to solve these lacks and improve the group recommendation accuracy. This hybrid recommender captures the user's preferences according to the semantics associated with the content. Our model has shown better results than the state-of-the-art, but we had not analyzed the impact of demographic data on these systems, yet. In this direction, we could note which these comparative analyses are being a relevant issue, helping us to understand how these systems behave in terms of the recommendation accuracy, improvements of the technical results, precision, recall, and computational cost.

In this paper we propose an analysis on our hybrid recommender model, which considers personal information from users, items metadata and implicit feedback into a unified and generic model. We aim to demonstrate how demographic data could assist to be an effective way to consider users' characteristics, enabling analysis about groups of users and their constraints. The evaluations demonstrate how our algorithm acts with or without users' personal data, and the achievements regarding demographic information on hybrid recommenders.

The paper is organized as follows: Section 2 addresses the related work; Section 3 describes the past models explored in this study; Section 4 presents our hybrid recommender model in detail; Section 5 describes the model's evaluation; finally, Section 6 provides the final remarks.

2. RELATED WORK

This section summarizes three features as basis of our hybrid recommender model: implicit feedback, which provides additional mechanisms to characterize the users' interests; items' metadata and its factorization, in order to adjust their relative importance in face of the users' preferences; and, users' personal information, which is an additional cue used to infer the particular likes and dislikes.

Implicit feedback – is an important mechanism to extract user's preferences mainly when explicit feedback is unavailable or incomplete [Joachims et al., 2005; Agichtein et al., 2006]. One ancient work in this field is proposed by Oard and Kim [1998], who identify different types of implicit feedback and how they could be exploited in a couple of recommendation strategies. More recent work include the proposal of Hu et al. [Hu et al., 2008], which is a tool to transform users' implicit feedback into a training data in a preference-conference format.

A relevant model was also proposed by Koren [2008; 2010], who integrated indirect information into a neighborhood latent factor model to improve recommendation performance. From this work, the author also proposed the SVD++ algorithm, which is a latent factor model that exploits implicit feedback gathered from the user's rental history. Still concerning implicit feedback, Yang et al. [2012] proposed a simple and effective local implicit feedback model mining users' local preferences to get better results. Their work consisted in extending Koren's algorithm by incorporating the notion of rating time interval to gather local and momentary interests.

Metadata incorporation – there is a variety of content-based methods available in the literature [Adomavicius and Tuzhilin, 2005] that design matching mechanisms between the item's metadata and the user profile, making the recommender to act similarly to information retrieval systems. However, usually content-based filtering algorithms have associated problems such as the occurrence of cold start and overspecialization, which may decay the recommendation results. In order to reduce the aforementioned problems, CF systems is used, whose main idea is to compute similarities of users/items in order to predict new items to users. One efficient way to compute such similarities is by using matrix factorization models in order to reduce the dimensionality of the user-item matrix [Koren, 2008; 2010]. However, if adopted Singular Value Decomposition (SVD) to factorize the users vs. items matrix, imputation methods will have to be incorporated to reduce the sparsely effects, but at the cost of distorting and/or over fitting the training data.

Regardless of the possibility to use only the observed ratings in order to reduce the sparsely effects [Funk, 2006; Koren, 2010], one alternative way to compute users' similarities by means of factorization techniques was proposed by Manzato [2012]. In this specific work, the author created a user-category matrix factorization model in order to extract user's preferences about movies. With the preferred categories, the system computes the users' similarities to support CF systems. However, such approach is not accurate because it uses Singular Value Decomposition to factorize the matrix, being also necessary an imputation method which leads to over fitting. In a more recent work, Manzato [2013] proposed a way to incorporate implicit feedback into a latent factors model with metadata awareness; however, it does not support user's personal information, such as demographic data. The idea of factoring a matrix associated to metadata (e.g. movies' genres) was also considered by Gantner et al. [2011], who described a method that maps user or items attributes to the latent features of a matrix factorization model. With such mappings, the factors of the model trained by standard techniques can be applied to the new user and the new item problems, while retaining its advantages, in particular speed and predictive accuracy.

Personal information – is related to demographic filtering [Pazzani, 1999; Krulwich, 1997]. This approach is based on the principle 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 fact is by using CF boosted by demographic information. Chen and He [2009], for instance, propose a CF algorithm that computes users' similarities based on three demographic attributes and ratings of items separately, and then, a new similarity is generated by combining the previous results. Lee and Lee et al. [2002] first segment all users by demographic characteristics, and later, apply in each segment a user clustering procedure according to the preference of items using a Self-Organizing Map (SOM) neural network. Yapriady and Uitdenbogerd [2005] propose a simple measure for combining demographic data with traditional CF techniques in order to improve group recommendation precision. Vozalis and Margaritis [2007] propose a CF approach which uses SVD, as an augmenting technique, and demographic data, as a source of additional information, in order to improve the quality of the generated predictions. In general, a modularity function adapted to social networks can also be used as an alternative for hybrid recommenders, indicating the possibility of building virtual communities [Expert et al., 2011]. In this direction, some works that consider human factors [Cremonesi et al., 2012; Rodriguez et al., 2012], for instance, despite of adopting a user-centered approach, do not regard the formation of groups of users. Gartrell et al. [2010], on the other hand, propose a group recommendation model that incorporates both social recommendation and content interest information to generate consensus among a group (the group consensus function), thereby identifying items that are most suitable for a group. In addition, they present a detailed analysis of key group characteristics and their impacts on the group decision making.

The aforementioned techniques are related to the proposed one in certain aspects: on the one hand, there are methods which exploit implicit feedback, but don't consider the available user and item metadata; on the other hand, there is a set of models which exploit the metadata and its factorization, but don't support implicit feedback.

In our model [Santos Junior et al., 2013], we have been addressed both issues improving the prediction results with a user's and item's descriptions aware model. Moreover, our work do not consider local implicit feedback neither neighborhood models, but incorporate item's metadata and user's personal information to improve accuracy. Furthermore, our approach differs from theirs because, in addition to metadata awareness, we cluster into unified model information from different sources and types: items' metadata, users' personal information, implicit feedback and latent factors. This unique model, in turn, is learnt from the observed ratings through a gradient descent scheme.

3. PAST MODELS

In this section, we describe in more details the models reported previously in the literature, which are explored by the one proposed in this paper.

3.1 Notation

Following the same notation of Koren [2010], we use special indexing letters to distinguish users, items and metadata: users are indicated as u and v; items are referred as i and j; and users and items' descriptions as d and g, respectively. As these descriptions may be of different types, we also use the index letters z_u and z_i to refer, respectively, to different types of users and items' metadata. A summary of these indexes and also the sets used in this paper are defined in Table 1 [Santos Junior et al., 2013].

Table 1. Notation for different sets used in this paper

Notation	Indexes	Definition (set of)
K	(u,i)	known ratings
$R(u)$	i,j	items rated by user u
$R(i)$	u,v	users who rated item i
$N(u)$	j	items for which user u provided an implicit feedback
$N(j)$	u	users who provided an implicit feedback to item j
$Z(u)$	z_u	different personal information considered in the system
$Z(i)$	z_i	different items metadata considered in the system
$G(u; z_u)$	d	descriptions of type x associated to user u
$G(i; z_i)$	g	descriptions of type y associated to item i

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

Because the rating data is sparse, the models are prone to over fitting. Thus, to address this issue, regularization is applied so that estimates are shrunk towards baseline defaults. Similarly to Koren [2010], we denote $\lambda_1, \lambda_2, \ldots$ the constants used for regularization. The values of these constants are defined in Section 5.1, which describes the experiments with the dataset adopted to the evaluation proposed in this paper.

3.2 Baseline Estimates

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

In order to overcome such differences, baseline estimates are used to adjust the data by accounting for these effects [Koren, 2010]. A baseline estimate for an unknown rating r_{ui} is denoted by b_{ui} and is defined as Equation 1, where μ refers to the overall average rating; and the parameters b_u and b_i indicate the observed deviations of user u and item i, respectively, from the average.

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

To estimate these parameters, usually two methods can be adopted. The first one consists of decoupling the calculation of the item biases from the user biases. For each item i, the bias is computed and the, b_i is used to calculate the user bias (b_u).

The second and more accurate method to estimate b_u and b_i is by solving the least squares problem, where the first term before regularization aims to find the user and item biases that fit the given ratings. The second term avoids over fitting by penalizing the magnitudes of the parameters [Herlocker et al., 2002; Koren, 2008; Koren, 2010].

3.3 Implicit Feedback

In RS, an important issue is how to integrate different forms of user input into the models to precisely reflect the user's preferences [Bell and Koren, 2007]. The algorithms usually rely only on explicit feedback, which includes ratings assigned by users to items they have visited in the past. A good example is Netflix[4], which allows users to choose and assign an amount of stars for movies they have watched. The system constructs and controls the user's profile by considering each rating into their personal interests.

On the other hand, one can argue that explicit feedback is not always available due to cold start, or simply because users may not assign any ratings to their preferences. Consequently, the implicit feedback could be explored, since it is an abundant source of information and indirectly reflects 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 [2008; 2010] proposed a set of models which faces implicit feedback when explicit feedback is also available, which integrates both types of feedback by considering ratings

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

assigned by users to visited items and also the set of movies rented in the past. The adopted dataset (Netflix) lacks this second 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 [2010] is the SVD++ algorithm[5], which integrates explicit and implicit feedback into a factorization model representing the user's preferences. Each user u is associated with a user-factors vector $p_u \in R^f$ and each item i with an item-factors vector $q_i \in R^f$. A popular prediction rule would be:

$$\hat{r}_{ui} = b_{ui} + p_u^T q_i \quad . \tag{2}$$

Based on Equation 2, Koren extended this basic model in order to consider implicit information. In fact, he used an additional factors vector $y_i \in R^f$ and also considered set $N(u)$, which contains all items for which u provided an implicit preference. Thus, the SVD++ model is defined as:

$$\hat{r}_{ui} = b_{ui} + q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) \quad . \tag{3}$$

The preferences of a user u are represented by a combination of explicit and implicit information. The user-factors vector p_u is learnt from the given explicit ratings, and complemented by the sum of y_j, which represents the implicit feedback. Again, the parameters are learnt by minimizing the associated squared error function through gradient descent [Koren, 2008; Koren, 2010; Funk, 2006; Paterek, 2007].

Still concerning implicit feedback, Manzato [2013] proposed an extension of the SVD++ model taking into account the items' metadata, when available. This extension was denominated "gSVD++" and considers set $G(i; z_i)$, which contains the descriptions of type z_i associated with item i. It also defines a metadata factors vector $x_g \in R^f$ containing the factors for each possible description. In this direction, the Equation 3 was rewritten in order to complement the items factor q_i with the available metadata, as follows:

$$\hat{r}_{ui} = b_{ui} + \left(q_i + \sum_{z_i \in Z(i)} |G(i; z_i)|^{-\alpha} \sum_{g \in G(i; z_i)} x_g \right)^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) \quad . \tag{4}$$

The original Koren's solution considered the y_j factor vector to represent the indirect user's information (e.g. rental history). As an extension to this model, another factor vector x_g was added to represent the items metadata (e.g. genres), incorporating the feature of metadata awareness.

[5] In fact, Koren proposed a more accurate algorithm than SVD++, which integrates implicit feedback with a neighborhood model. However, this method is not in the scope of our original proposal, as it also incorporates similarities of users and items.

4. HYBRID RECOMMENDER

Differently from the previous approaches, our method proposed in Santos Junior et al. [2013] consists of a latent factor approach which integrates users' and items' metadata and implicit feedback into a unified model. Users and items metadata are incorporated into a hybrid approach to improve the prediction accuracy by detecting such small and semantic associations among the involved entities.

Thus, depending on the semantics associated with the user's profile and demographic data, the recommender selects those items whose subject is meaningfully related to the individual's tastes. The next subsections describe the proposed model in details, by gradually incorporating the various components which constitute the final schema.

4.1 Baseline Revisited

In the Subsection 3.2 we described the baseline estimates, which model systematic tendencies according to users' and items' intrinsic characteristics. Such a technique can be slightly improved by also incorporating the global effects of how users rate items depending on the contextual environment or demographic data. Considering the age group, for instance, children may rate an item differently from adults; similarly, if the user is interacting with their church friends, they may rate an item differently than when cycling. In order to model such possibilities, our model extends the baseline estimates by also considering the contextual environment or demographic information:

$$b_{ui}^{demo} = \mu + b_u + b_i + \sum_{z_u \in Z(u)} |G(u; z_u)|^{-1} \sum_{d \in G(u; z_u)} b_d \quad . \tag{5}$$

In this case, $Z(u)$ is the set of different types of user's information considered in the system and $G(u; Z_u)$ represents all data of type Z_u associated with user u. An example is when using the demographic information from the MovieLens dataset[6]. We have denoted $Z(u)$=*{occupation, age group, gender, zip code}* and Z_u=*{occupation}* of a user u as $G(u;$ *occupation)={programmer}*, for instance. Similarly, considering all metadata types available in the dataset, we denote $Z(i)$=*{title, genre, year of release, IMDB URL}*; and when Z_i=*{genre}* of an item i, we denote $G(i; Z_i)$=*{action, science fiction}*, for instance. It is worth mentioning that in most cases $|G(u; Z_u)|=1$, but we preferred to keep generality by using more than one piece of information associated with Z_u.

The contextual biases b_d can be estimated by solving a least squares problem. In the experiments reported in Section 5, we employ a simple gradient descent scheme using the observed data to change the parameters in the opposite direction of the gradient.

4.2 Incorporation of Item's Metadata

In addition to the global effects modeled in our extended estimated baseline, we aimed to create associations between the demographic data and the content metadata available for each

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

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 certainly prefers children's films, female users will probably like romantic and drama films, and a group of cyclists will enjoy sports and adventure content.

In order to capture such associations between users' and items' metadata, we incorporated another set of parameters h_{dg} to Equation 5, as follows:

$$\hat{r}_{ui}^{meta} = b_{ui}^{demo} + \sum_{z_i \in Z(i)} |G(i;z_i)|^{-1} \sum_{z_u \in Z(u)} |G(u;z_u)|^{-1} \sum_{d \in G(u;z_u)} \sum_{g \in G(i;z_i)} h_{dg} \quad . \tag{6}$$

Here set $Z(i)$ denotes all different types of items' metadata, such as genres, list of actors and keywords. Set $G(i; z_i)$ represents all pieces of information of the same type z_i associated with item i. For instance, we could instantiate $Z(i)=\{genre\}$ and then an item i entitled "Star Wars" that would have a set of $G(i; genres)$ composed of "Science Fiction", "Action" and "Adventure". The parameters represented by h_{dg} capture the weights of a user's demographic data d associated with an item description g. Again, such weights are learned from the observed data through gradient descent.

4.3 Factorization and Implicit Feedback

The last refinement of our model refers to the incorporation of latent factors and implicit feedback regarding users. By using latent factors, it is possible to capture a user's preference to different features, which characterize the whole item to be recommended. On the other hand, by using implicit feedback, it is possible to capture the user's preferences even if they have few ratings.

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-factors vector $p_u \in R^f$ and each item i with an item-factors vector $q_i \in R^f$. A popular prediction rule would be: $\hat{r}_{ui} = b_{ui} + p^{T}_{u} q_i$. Koren extended this basic model in order to consider implicit information. In fact, he used an additional factors vector $y_i \in R^f$ and also considered set $N(u)$, which contains all items for which u provided an implicit preference.

The preferences of a user u are represented by a combination of explicit and implicit information. The user-factors vector p_u is learnt from the given explicit ratings, and complemented by the sum of y_j, which represents the implicit feedback. The parameters are learnt by minimizing the associated squared error function through gradient descent [Koren, 2008, 2010; Paterek, 2007].

Such an enhancement in our model is dictated by the combination between our approach (Equation 6) and Koren's SVD++ model [Koren, 2010]. Concretely, our unified model is defined by the following equation:

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

Instead of summing the items' metadata features with the factors vector q_i, the information is added as global effects into the baselines. Concomitantly, we combine users' demographics and items' metadata in order to capture the relationship between users' personal information and items' descriptions.

Similarly to the previous formulations, the parameters are learnt by minimizing the regularized squared error function associated with Equation 6 as follows:

$$\begin{aligned} \min_{b_*, h_*, q_*, p_*, y_*} \sum_{(u,i) \in K} \Bigg(& r_{ui} - \mu - b_u - b_i - \sum_{z_u \in Z(u)} |G(u; z_u)|^{-1} \\ & \sum_{d \in G(u; z_u)} b_d - \sum_{z_i \in Z(i)} |G(i; z_i)|^{-1} \sum_{g \in G(i; z_i)} b_g - \sum_{z_i \in Z(i)} |G(i; z_i)|^{-1} \\ & \sum_{z_u \in Z(u)} |G(u; z_u)|^{-1} \sum_{d \in G(u; z_u)} \sum_{g \in G(i; z_i)} h_{dg} - q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) \Bigg) \\ & + \lambda \Bigg(b_u^2 + b_i^2 + ||p_u||^2 + ||q_i||^2 + \sum_{z_u \in Z(u)} \sum_{d \in G(u; z_u)} b_d^2 + \sum_{z_i \in Z(i)} \sum_{g \in G(i; z_i)} b_g^2 \\ & + \sum_{z_u \in Z(u)} \sum_{z_i \in Z(i)} \sum_{d \in G(u; z_u)} \sum_{g \in G(i; z_i)} h_{dg}^2 + \sum_{j \in N(u)} y_j^2 \Bigg) \quad , \end{aligned} \tag{8}$$

Using the same strategy adopted by other authors [Koren, 2008, 2010], we employ a simple gradient descent scheme to solve the system indicated in Koren [2010]. Let us consider $e_{ui} \stackrel{def}{=} r_{ui} - \hat{r}_{ui}$. Using the training dataset, we loop over all known ratings in K. For a given training example r_{ui}, we change and move the parameters in the opposite direction of the gradient, as illustrated in Algorithm 1.

Input: Set of known ratings $(u, i) \in K$
Output: Learnt parameters $b_u, b_i, b_d, b_g, h_{dg}, p_u, q_i, y_j$
for *count = 1,...,#Iter.* **do**
  foreach $(u, i) \in K$ **do**
    $\hat{r}_{ui} \leftarrow$ Prediction according to Equation 3; $e_{ui} \leftarrow r_{ui} - \hat{r}_{ui}$; $b_u \leftarrow b_u + \gamma(e_{ui} - \lambda_1 b_u)$;
    $b_i \leftarrow b_i + \gamma(e_{ui} - \lambda_2 b_i)$; **foreach** $z_u \in Z(u)$ **do**
      foreach $d \in G(u; z_u)$ **do**
        $b_d \leftarrow b_d + \gamma(e_{ui} - \lambda_3 b_d)$;
      end
    end
    foreach $z_i \in Z(i)$ **do**
      foreach $g \in G(i; z_i)$ **do**
        $b_g \leftarrow b_g + \gamma(e_{ui} - \lambda_3 b_g)$;
      end
    end
    foreach $z_u \in Z(u)$ **do**
      foreach $z_i \in Z(i)$ **do**
        foreach $d \in G(u; z_u)$ **do**
          foreach $g \in G(i; z_i)$ **do**
            $h_{dg} \leftarrow h_{dg} + \gamma(e_{ui}|G(u; z_u)|^{-1}|G(i; z_i)|^{-1} - \lambda_3 h_{dg})$;
          end
        end
      end
    end
    $p_u \leftarrow p_u + \gamma_2(e_{ui}q_i - \lambda_4.p_u)$; $q_i \leftarrow q_i + \gamma_2(e_{ui}(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j) - \lambda_4.q_i)$; **foreach** $j \in N(u)$ **do**
      $y_j \leftarrow y_j + \gamma_2(e_{ui}|N(u)|^{-\frac{1}{2}}q_i - \lambda_4.y_j)$;
    end
  end
  $\gamma \leftarrow \gamma * 0.9$; $\gamma_2 \leftarrow \gamma_2 * 0.9$;
end

Algorithm 1. Learning the factorized model through gradient descent.

5. EVALUATION

The evaluation presented in this paper consists in comparing our model with other methods available in the literature. We also evaluate the different parts of the model to check the contribution of each aspect to the final recommendation improvement.

5.1 Experiments and Dataset

The experiments were conducted with the well-known MovieLens-1M dataset[7]. It consists of 6000 users, who assigned 1 Million ratings to 1700 movies. The dataset provides metadata regarding those users and items. Demographic data are also provided, such as age, occupation, gender and Zip code. In this evaluation, we consider all types of demographic data, except Zip code. Regarding the users' age, we pre-processed such information in order to cluster those users in the same age group. Tables 1, 2 and 3 shows how was configured experimentally in order to associate the same information with all users that belong to that group.

[7] http://www.grouplens.org/datasets/movielens/

Table 1. Users occupation.

Occupation	Users	Dataset (%)
"other" or not specified	711	11.77%
academic/educator	528	8.74%
artist	267	4.42%
clerical/administrator	173	2.86%
college/grad student	759	12.56%
customer service	112	1.85%
doctor/health care	236	3.90%
executive/managerial	679	11.24%
farmer	17	0.28%
homemaker	92	1.52%
K-12 student	195	3.22%
lawyer	129	2.13%
programmer	388	6.42%
retired	142	2.35%
sales/marketing	302	5.00%
scientist	144	2.38%
self-employed	241	3.99%
technician/engineer	502	8.31%
tradesman/craftsman	70	1.15%
unemployed	72	1.19%
writer	281	4.65%

Table 2. Users genre.

Sex	Users	Dataset (%)
Male	4331	71.70%
Female	1709	28.29%

Table 3. Users age.

Age	Group	Users	Dataset (%)
0 – 12	infant	222	3.67%
12 – 18	teenager	1103	18.26%
18 – 25	young adult	2096	34.70%
25 – 35	adult	1193	19.75%
35 – 55	mature	550	9.10%
55 – 60	aged	496	8.21%
Over 60	elder	380	6.29%

As a result, our set $Z(u)$ is composed of *{age group, occupation, gender}* and sets $|G(u$, *age group*$)|$, $|G(u$, *occupation*$|)$ and $|G(u$, *gender*$|)$ have all size 1. Regarding items' metadata, the MovieLens dataset provides the movie title, date of release, IMDB URL and the set of associated genres. In our experiments, we considered only the genres items' metadata; consequently, $|Z(i)|=1$, and $|G(i, genre)|\geq 1$, because one or more genres can be assigned to each movie i. In this version of the dataset, there are 19 different genres, all considered in this evaluation. The combination of two or more types of items' metadata is left to future work.

We chose three previous methods to be compared against our model in this evaluation:

- Biased MF: algorithm proposed by Rendle and Lars [2008] reduces the cold start by deriving an online-update algorithm for regularized kernel matrix factorization models. The algorithm is also flexible for nonlinear interactions between feature vectors.
- DemoSVD++: algorithm proposed by Koren [2010] is an accurate model which exploits implicit feedback with demographic data from users and latent factors.

All methods were implemented using MyMediaLite library [Gantner et al., 2011]. The prediction accuracy of all demographic data variations in terms of RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) was compared according to a varying number of factors. Thus, the best number of factors was selected, i.e., the value of f for which the RMSE computed in the previous step was minimal. Finally, we use these specific values of f for each comparison, all methods were compared in terms of precision accuracy and AUC (Area Under Curve) at top 5 and top 10 recommendations.

For all experiments the 5-fold cross-validation was used to improve the results with more confidence. The constants involved in this evaluation were defined experimentally and are summarized in Table 4. The details of their utilization can be found in Algorithm 1, as previously explained.

Table 4. Constant values.

Constant (#Iterations)	**Value** (#50)
γ	0.01
γ_2	0.021
λ_1	$0.025*\lvert R(u)\rvert^{-\frac{1}{2}}$
λ_2	$0.025*\lvert R(i)\rvert^{-\frac{1}{2}}$
λ_3	0.025
λ_4	1

This occurs because the first layer of our model (i.e. the value of $\hat{r}_{meta,ui}$) is not based on latent factors, although its significance is high when the final prediction is computed. An advantage of such an approach is that we can provide good recommendations for low values of f. Thereby, the computational cost becomes significantly lower in comparison with the related approaches. Table 5 shows the RMSE results for specific values of f.

5.2 Results

We can note that our model achieved the best accuracy when *f*=20, whereas all other techniques yielded the best results when *f*=10 for BiasedMF and *f*=20 for DemoSVD++. However, our algorithm outperformed DemoSVD++ (0.8434 against 0.8464) and Biased MF (0.8434 against 0.8495). In fact, the accuracy curve tends to be better when more factors are added, showing improved results for the hybrid recommender against other approaches. Furthermore, the hybrid recommender showed better results for all of the *f* values, when compared with other state-of-the-art approaches, as shown in Figure 1.

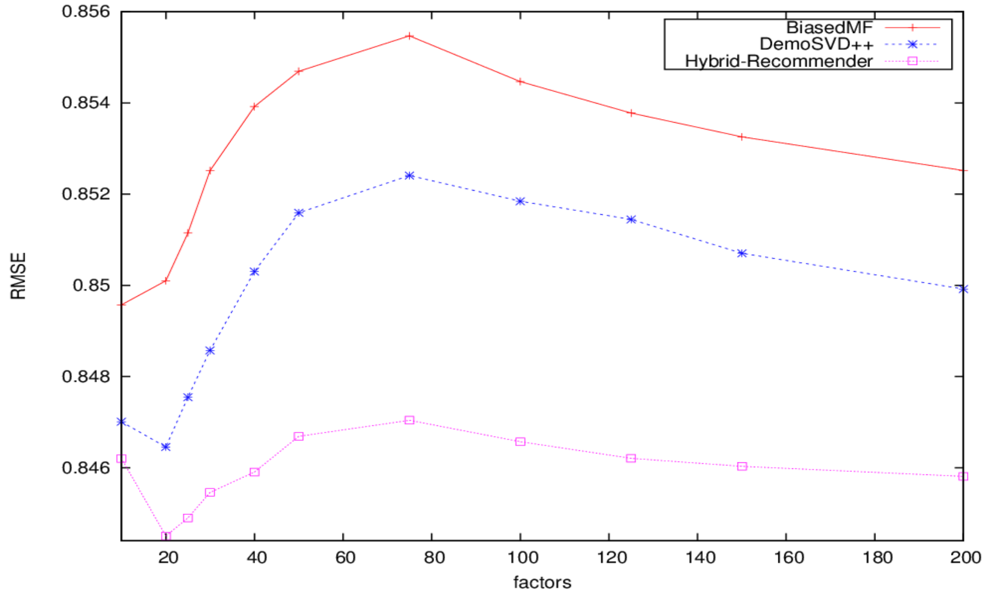

Figure 1. RMSE comparison of the base method with other approaches.

Furthermore, we measured our evaluation with the quality of top-N recommendations for all techniques considered. This study was important because, as argued in the literature [Cremonesi et al., 2010; 2012], the RS aim to find few specific items which are supposed to be most appealing to the user, and common strategies, such as computing prediction accuracy in terms of RMSE are not a natural fit for evaluation.

Table 6 shows the AUC, precision@5 and precision@10 for the algorithms. It also shows the training and testing time captured on an Intel Core i7-3930K CPU @ 3.20GHz x 12 and 8GB RAM machine equipped with Ubuntu 13.10 operating system. Our algorithm could recommend better items, which were more significant to the users' interests. It is worth mentioning that such results were obtained with only 20 factors, being that Biased MF with 10 and DemoSVD++ with also 20 factors (Table 6). Our method needed almost 1:15 hour (more than other techniques) to train the model, but their precision is most expressive, being better than other techniques. Regarding the Biased MF, because it processes fewer data, it could achieve the fastest training time even with 10 factors, but computed the worst predictions among all.

Table 5. RMSE comparison for different *f* values

Method	*f'10*	*f'20*	*f'50*	...	*f'100*	*f'150*	*f'200*
Biased MF	**0.8495**	0.8501	0.8546	...	0.8544	0.8532	0.8525
DemoSVD++	0.8470	**0.8464**	0.8515	...	0.8518	0.8507	0.8499
Hybrid Rec.	0.8451	**0.8434**	0.8453	...	0.8462	0.8455	0.8448

Next, we compared all types of user's demographic data: age group, occupation and gender, which were combined with the items' metadata, e.g. genres. Accordingly, Figures 2 and 3 – regarding RMSE and MAE comparisons, respectively – presents seven runs of our algorithm, which differ from each other in terms of the types of demographic data considered. For example, the red curve indicates the RMSE and MAE variation of the model according to the number of factor *f* using only the age group information from users. We clearly see in Figures 2 and 3 that as more demographic data are considered, better is the results regarding the prediction accuracy.

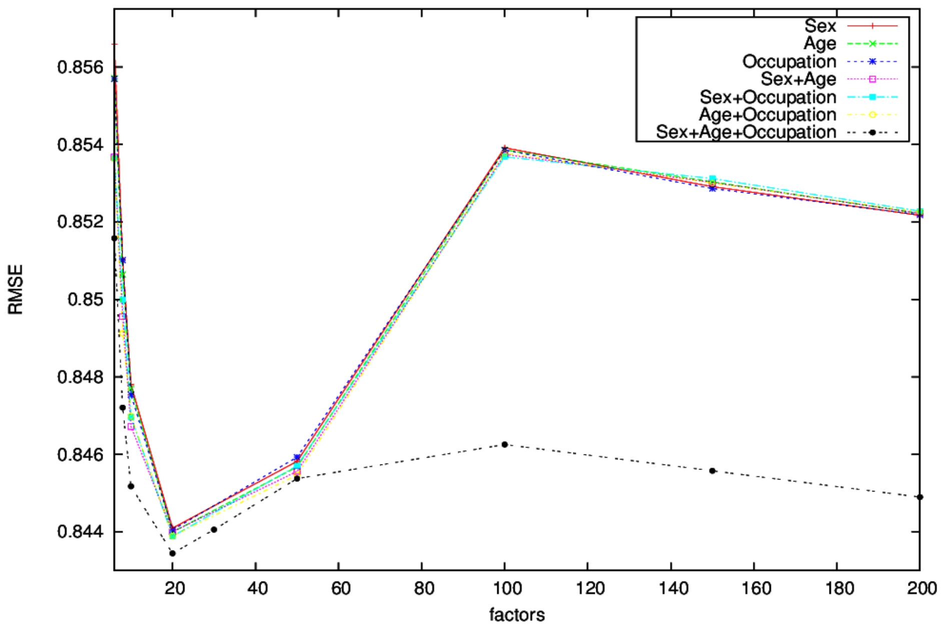

Figure 2. RMSE comparison of the base method with all demographic variations.

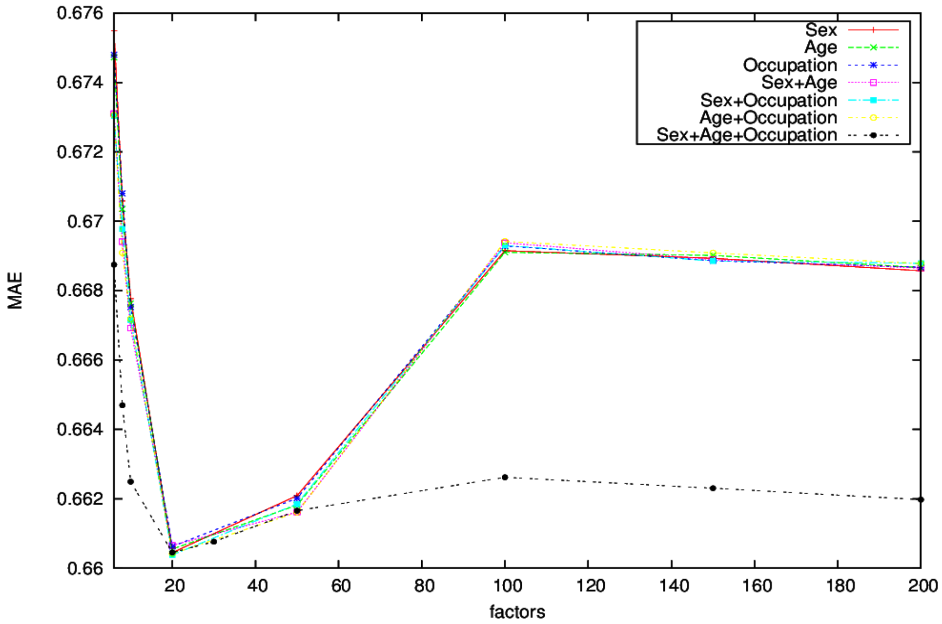

Figure 3. MAE comparison of the base method with all demographic variations.

In addition, among the runs which used only one type of user information (age group, occupation or gender), we realized that occupation was the one that scored the worst results. This is understandable, as two users from completely different jobs can have similar tastes for movies, whereas two users from different age groups or gender may have some divergences about movie genres. We still can note which the different possibilities of combination of only two types of data (age group/occupation, age group/gender and occupation/gender) produces similar results in either case.

Table 6. Comparison of strategies according to top-N and training and testing time

	Algorithms			
	BiasedMF	*DemoSVD++*	*Hybrid Recommender 1*	*Hybrid Recommender 2*
factors	10	20	20	10
AUC	0.6976	0.6993	0.8319	0.709
prec@5	0.1263	0.1313	0.1387	0.1239
prec@10	0.1055	0.1092	0.1202	0.1052
training	00:00:24.08	00:02:51.97	01:15:19.35	01:15:10.27
testing	00:00:06.15	00:00:15.89	00:00:16.00	00:00:07.89

Thus, we can note the effectiveness of incorporating users' personal information into the model and combining it with items' metadata. Although in this evaluation we considered only demographic data, in fact our model is generic enough to consider additional user information, such as the actual context or environment that the interaction is happening at specific time with two or more users who share the same preferences.

Finally, we evaluate the impact of our hybrid recommender model regarding users' characteristics input. For this, we have been used two different models: "Hybrid Recommender 1" (with demographic data) and "Hybrid Recommender 2" (without demographic data). It is worth mentioning that such results were obtained with only 20 factors for both each situations. With demographic data our method needed fewer than 1 minute more to train the model.

However, our hybrid recommender archieved a precision more expressively than model without demographic data in both situations (prec@5 with 0.1387 against 0.1239 and prec@10 with 0.1202 against 0.1052).

6. FINAL REMARKS

This study was based on our last hybrid recommender proposed, that explores latent factors in order to integrate users' demographics, items' metadata and implicit feedback into a unified model. This hybrid recommender aimed to capture the user's preferences according to the semantics associated with the content. The objective was to capture the user's preferences according to personal information and the semantics associated with the content. For instance, a 7-year-old child will probably like items specific for children (e.g. cartoons), whereas adult users will have different preferences, which are captured by latent factors. The technique involves less computational cost than neighborhood models, but at the same time, users and items metadata are incorporated into a hybrid approach to improve the prediction accuracy by detecting small and semantic associations among the entities involved.

In this paper we aimed to evaluate about the impact of demographic data on a hybrid recommender based on different forms of user's input to construct an accurate profile, since hybrid recommenders are a good alternative for RS. Thus, we argue that additional demographic data are an effective in order to enable systems to discover and analyze contextual constraints in a real world recommendations scenario. As results, demographic data showed an effective way to consider users' characteristics, regarding contextual constraints in a group environment, in order to improve the group recommendations.

We demonstrate how demographic data could assist to be an effective way to consider users' characteristics, enabling analysis about group of users and their constraints. At the same time, this paper shows the evaluation of our model also with other techniques, which also demonstrated its effectiveness in this sense.

Finally, we have provided a system's evaluation in terms of prediciton accuracy and precision at top-N recommendations, by comparing users' demographic data into a same contextual group environment. This evaluation shows the model's scalability, regarding the importance of demographic data, regarding expressive precision results that our model was acquired if compared to a without demographic data model.

ACKNOWLEDGEMENT

The authors would like to thank the São Paulo Research Foundation – FAPESP (grants #2011/00422-7; #2013/22547-1) and the Coordination for the Improvement of Higher Educational Personnel – CAPES (grant #4586/2014-00) for the financial support.

REFERENCES

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.

Agichtein, E., Brill, E. and Dumais, S. “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, (New York, NY, USA), pp. 19–26, ACM, 2006.

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

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370.

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.

Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys ’10, pages 39–46, New York, NY, USA. ACM.

Cremonesi, P., Garzottto, F., and Turrin, R. (2012). User effort vs. accuracy in rating-based elicitation. In *Proceedings of the sixth ACM conference on Recommender systems*, RecSys ’12, pages 27–34, New York, NY, USA. ACM.

Expert, P., Evans, T. S., Blondel, V. D., and Lambiotte, R. (2011). Uncovering space-independent communities in spatial networks. *Proceedings of the National Academy of Sciences*.

Funk, S. (2006). Netflix Update: Try This At Home. Available in: <http://sifter.org/~simon/journal/20061211.html>. [Online; accessed 09-September-2014].

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.

Gartrell, M., Xing, X., Lv, Q., Beach, A., Han, R., Mishra, S., and Seada, K. (2010). Enhancing group recommendation by incorporating social relationship interactions. In *Proceedings of the 16th ACM International Conference on Supporting Group Work*, GROUP ’10, pages 97–106, New York, NY, USA. ACM.

Herlocker, J., Konstan, J. A. and Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, vol. 5, pp. 287–310, Oct.

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, (Washington, DC, USA), pp. 263–272, IEEE Computer Society.

Joachims, T., Granka, L., Pan, B., Hembrooke, H. and Gay, G. “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, (New York, NY, USA), pp. 154–161, ACM, 2005.

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

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

Lee, M., Choi, P., and Woo, Y. (2002). A hybrid recommender system combining collaborative filtering with neural network. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2347 LNCS:531–534.

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, (New York, NY, USA), pp. 249–252.

Manzato, M. G. (2013). gSVD++: supporting implicit feedback on recommender systems with metadata awareness. In *28th Symposium On Applied Computing - SAC'13*, Coimbra, PT. ACM.

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

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, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.*, 13(5-6):393–408.

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, F., Rokach, L., Shapira, B., and Kantor, P. B., editors (2011). *Recommender Systems Handbook*. Springer.

Rodriguez, M., Posse, C., and Zhang, E. (2012). Multiple objective optimization in recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems*, RecSys '12, pages 11–18, New York, NY, USA. ACM.

Santos Junior, E. B., Manzato, M. G., and Goularte, R. (2013). Hybrid recommenders: Incorporating metadata awareness into latent factor models. In *Proceedings of the 19th Brazilian Symposium on Multimedia and the Web*, WebMedia '13, pages 317–324, New York, NY, USA. ACM.

Vozalis, M. G. and Margaritis, K. G. (2007). Using svd and demographic data for the enhancement of generalized collaborative filtering. *Inf. Sci.*, 177(15):3017–3037.

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, (New York, NY, USA), pp. 91–98, ACM.

Yapriady, B. and Uitdenbogerd, A. L. (2005). Combining demographic data with collaborative filtering for automatic music recommendation. In *Proceedings of the 9th international conference on Knowledge-Based Intelligent Information and Engineering Systems - Volume Part IV*, KES'05, pages 201–207, Berlin, Heidelberg. Springer.