A New Modeling for Item Ratings Using Landmarks

Gustavo R. Lima
PESC, COPPE
UFRJ
Rio de Janeiro, Brazil
grlima@cos.ufrj.br

Carlos E. Mello
PPGI, CCET
UNIRIO
Rio de Janeiro, Brazil
mello@uniriotec.br

Geraldo Zimbrao
PESC, COPPE
UFRJ
Rio de Janeiro, Brazil
zimbrao@cos.ufrj.br

***Abstract*—Collaborative Filtering (CF) has been a widely used approach for personalized recommendations. In this context, model-based CF algorithms have been studied extensively in the literature and have shown higher accuracy in rating prediction than memory-based ones. The major approach regarding model-based CF is Matrix Factorization (MF). It uses the item ratings given by users to predict unknown ratings, which are posteriorly used in recommendations. Usually, MF learns a data model through optimization algorithms by requiring too much time to process. To overcome this issue, we propose a novel modeling for item ratings that allows one to apply Supervised Learning (SL) algorithms to predict unknown ratings. This modeling consists in computing item similarities to a preselected set of items, namely landmarks. Then, we build a model for each user using his/her ratings as labels and the corresponding vector of item similarities to landmarks as samples. In this work, it was applied Support Vector Regression to learn the data model. We compared our proposal against 5 state-of-the-art CF techniques in 6 different databases. The results show the proposal is able to reduce computational time and still keep competitive accuracy compared to the experimented CF algorithms.**

***Index Terms*—Recommender Systems, Collaborative Filtering, Landmarks, Support Vector Regression**

I. Introduction

The emergence of new technologies and the increasing in internet speed turn the urgency felling more and more common among different people. But the technology advances also bring several challenges by putting available a huge volume of data, which must be processed to understand users' behavior and, consequently, increase the performance of several systems. Recommender systems are one of these that benefit themselves from the increase of data available and have an important role in helping users to find desired items among too many options.

In this context, Collaborative Filtering (CF) [1] has been used to provide users with personalized item recommendations. It considers the history of purchases and users' tastes to identify items that are likely to be acquired. In general, this information is represented by item ratings given by users, which is stored in a rating matrix. Each rating matrix row represents a user, its columns correspond to items and the cells hold the rating value associated to a specific user and item. Thus, the CF algorithms' objective is to predict the missing values of the rating matrix, which are posteriorly used in personalized item recommendations.

In spite of many researchers seek to improve accuracy of CF algorithms, a well known issue faced by them is how to improve algorithms' scalability, since even more data are available. For instance, memory-based CF algorithms require more time to build similarity matrix as the number of users/items increase [2], while model-based CF ones usually learn the data model offline, since many Matrix Factorization (MF) apply optimization algorithms in learning phase [3]. Thus, scalability is another aspect of CF which requires the attention of researchers.

In this paper, we propose a new item modeling such that Supervised Learning (SL) algorithms may be applied to rating prediction. The proposed modeling consists in representing items through their similarities to preselected items, namely landmarks. This representation holds information about users' taste, since the similarity measures consider user ratings during calculation. Then, a model is learned for each user using his/her ratings as labels and the corresponding item representation, which is a vector of similarities (features), as samples. We applied Support Vector Regression (SVR) [4] in our experiments.

The proposed modeling enables one to apply more scalable SL techniques in item recommendation, which is not possible if users or items are represented by rating vectors, since many ratings are missing. Furthermore, the results showed that our proposal outperformed the experimented CF algorithms in terms of computational performance, while keeping competitive accuracy.

The main contribution of this work is an item modeling scheme, which incorporates users' taste, and allows one to apply SL algorithms to rating prediction. Besides, it investigates how to use landmarks in CF context that, for the best of our knowledge, has not been done yet.

This work is organized in six sections, where this is the first one. Section 2 reviews the literature and presents the related work. Section 3 presents some CF definitions, introduces our proposal and depicts a toy example. Section 4 describes the databases and evaluation metrics used in experiments, and defines the experiment objectives. Section 5 starts by investigating the parameters of the proposed modeling, and

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

follows by comparing our proposal against other CF algorithms. Finally, Section 6 points out conclusions and future work.

II. Related Work

The objective of Collaborative Filtering (CF) algorithms is to predict the ratings a specific user would give to not yet consumed items and, consequently, determine which items this user would prefer rather than others [1]. For this purpose, such algorithms try to learn users' behaviour from past data represented by a rating matrix R. In this matrix, users and items are represented by rows and columns, respectively, and the cells hold item ratings whenever given by users. Otherwise, the cell is empty and the unknown ratings must be predicted by CF algorithms.

CF literature usually classifies these algorithms in Memory-based and Model-based [1]. Memory-based algorithms have two phases: (1) neighborhood identification and (2) rating prediction. In the first phase, a similarity measure is applied to identify the k most similar users (*i.e.* the neighbors) of a particular user u in a user-based CF approach. Then, the similarities and ratings of neighbors are aggregated in the second phase to estimate the unknown ratings of user u. The most representative algorithm of this class is k-Nearest Neighbors, which was adapted to work with missing ratings [1]. The memory-based CF algorithms may also be applied in item-based scheme, but instead of finding the neighbors of a specific user, they find the items that are most similar to those consumed by the user [5], [6].

Likewise, model-based algorithms also have two phases: (1) learning a model and (2) rating prediction. In the first one, the available ratings are used to learn a model, which keeps the information about user taste. Often, optimization techniques are applied in learning phase, *e.g.* Stochastic Gradient Descent (SGD) or Alternating Least Squares (ALS) [3], [7], which makes it expensive in terms of computational cost [8]. In the second phase, the learned model is able to generate accurate recommendations, but it becomes outdated since new information (*e.g.* ratings) are available, which requires it to be retrained [9].

The most known model-based CF algorithms are based on Matrix Factorization (MF) [3], which is an adaptation of dimensionality reduction algorithms for the recommender system context (*e.g.* Regularized SVD [10], Improved Regularized SVD [10], Probabilistic MF [11], Bayesian Probabilistic MF [12], Poisson MF [13], Non Negative MF [14], [15]). Usually, these algorithms are more accurate [16] and scalable than the memory-based ones [17], but the former may easily explain recommendations, since they are obtained from the most similar users/items [18].

This work fits into the group of model-based techniques by proposing to build a Support Vector Regression (SVR) model [4] for each recommender system's user, which is able to make accurate recommendations. To learn SVR model, we use item feature vectors as samples and the item ratings given by the user as labels. We consider the item similarities to few preselected items, namely landmarks, as features. Therefore, instead of representing items by a vector of user ratings, we represent them by their similarities to other items (the landmarks). In landmark space, the item position depends on user taste, once similarities to landmarks are computed considering the ratings given by users.

To the best of our knowledge, there is no previous work in recommender system that uses landmarks to model users/items or to improve CF algorithms. The main contributions concerning landmarks are in the area of dimensionality reduction [19], [20].

In this context, Silva and Tenenbaum proposed to reduce computational cost of Multidimensional Scaling (MDS) through landmarks [19]. They presented an algorithm, namely Landmark MDS (LMDS), that uses few data observations as landmarks and, instead of computing the full similarity matrix of all observations, it computes the landmark similarity matrix which is used to obtain the lower dimensional vector space. Then, the remaining observations are mapped to this new space, considering their similarities to the landmarks.

The main advantage of using LMDS is the trade off between accuracy and runtime [21]. In one hand, if one needs to perform LMDS faster, it is possible to sacrifice accuracy by reducing the number of the landmark. On the other hand, if one needs to improve the algorithm's accuracy, it is required to increase the number of landmarks.

Another algorithm that uses landmarks to reduce computational cost is Landmark Isometric feature map (or Landmark Isomap) [20], [22]. Similarly, it chooses few samples as landmarks and computes the similarity between all data points and the landmarks. Then, it applies LMDS to this similarity matrix in order to find an Euclidean embedding of data, which saves a great amount of time if the number of landmark is much smaller than the number of samples.

Finally, an important issue in landmark approach is to choose the most informative instances as landmarks, since data representation depends on the similarity to these points. Several selection strategies are proposed in literature and most of them are related to Landmark Isomap (*e.g.* [23]–[25]).

III. Proposal

Here, we start by presenting some Collaborative Filtering (CF) definitions. Then, we introduce our proposal to modeling item ratings, which allows one to apply more scalable Supervised Learning (SL) techniques in the context of CF. Finally, we present a toy example to illustrate the process.

A. Collaborative Filtering Definitions

Collaborative Filtering (CF) takes into account only the available ratings in rating matrix R to estimate users tastes and, then, makes recommendations of items for a particular user. For this purpose, many CF algorithms were proposed (*e.g.* k-Nearest Neighbors [1], Regularized Singular Value Decomposition [10], Non Negative Matrix Factorization [14], etc). These are based on different assumption, but all of them undergo a data sparsity issue, since the less ratings are known, the harder is to learn the model [26].

Formally, let $\mathbf{U}$, $\mathbf{P}$, and $\mathbf{R}$ be the set of users, the set of items, and the rating matrix, respectively. Yet, let $\mathbf{V}$ be the set of possible rating values in the recommender system. Thus, the rows of $\mathbf{R}$ represent users, the columns correspond to items and each cell at row u and column v holds the rating value $r_{uv} \in \mathbf{V}$, if a user $u \in \mathbf{U}$ evaluated an item $v \in \mathbf{P}$. Otherwise, the cell is empty.

Typically, the rating matrix $\mathbf{R}$ is highly sparse, since most of the possible ratings are unknown and the dimensions (*i.e.* $|\mathbf{U}| \times |\mathbf{P}|$) tend to be huge [1].

Therefore, the recommender system aims at finding for a particular user u the item $v \in \mathbf{P} \setminus \mathbf{P}_u$ to which the user u is likely to be most interested. For this purpose, CF algorithms estimate a function $f : \mathbf{U} \times \mathbf{P} \rightarrow \mathbf{V}$ that predicts the rating value $f(u, v) = \hat{r}_{uv}$ for a user u and an item v [27].

B. A New User Rating Modeling

We propose a new modeling of item ratings that allows one to use more scalable Supervised Learning (SL) algorithms to predict unknown ratings, since the volume of data tends to be huge. It consists in associating the ratings given by a specific user u to the features of the rated items. The features contain information about the ratings given by other users, *i.e.* they hold information about users' taste, and are used as samples to train a Support Vector Regression (SVR) [4]. Then, after the SVR model is built for user u, it is able to make recommendations.

The features we propose to use are the similarities between the items and a preselected item set $\mathbf{L}$, namely landmarks set. Therefore, for each item v rated by user u with value r_{uv}, it is associated a feature vector $w_v = (s(v, l_1), s(v, l_2), ..., s(v, l_{|L|}))$, where $l \in L$ and $s(v, l)$ is a function that computes the similarities between item v and landmark l.

Firstly, to obtain the features, one must select the k most rated items in recommender system, *i.e.* the landmarks, which compose L. Then, a matrix containing item similarities to landmarks is built, using Cosine similarity, showed in (1):

$$cos(v, p) = \frac{\sum\limits_{u \in U_{vp}} r_{uv} r_{up}}{\sqrt{\sum\limits_{u \in U_{vp}} r_{uv}} \sqrt{\sum\limits_{u \in U_{vp}} r_{up}}}, \tag{1}$$

where v and p are items, and $U_{vp} = U_v \cap U_p$ is the set of users who co-rated items v and p.

Next, we apply a min-max normalization in landmark similarities, obtaining a dense matrix S with dimension $|P| \times |L|$. Its rows contain the features of the items and are used as samples to train the SVR model. The sample labels are the ratings given by the user to the corresponding items. Therefore, for each user u of the recommender system, we learn a model using the features of rated items as samples and the rating values as labels, as may be seen in the algorithm 1.

Algorithm 1: Training a SVR model for each user u

```
Data: rating matrix R, user set U, item set P,
      parameters of SVR param
Result: array of models M
L ← selectLandmarks(n)
/* Computing Cosine similarities       */
for v ∈ P do
    for l ∈ L do
        S[v, l] ← cos(v, l)
    end
end
/* Normalizing similarities            */
for l ∈ L do
    a ← min(S[:, l])
    b ← max(S[:, l])
    for v ∈ P do
        S[v, l] ← (S[v, l] − a)/(b − a)
    end
end
/* Learning SVR model per user         */
for u ∈ U do
    samples ← S[P_u, :]
    labels = R[u, :]
    M[u] = SVR(param)
    M[u].fit(samples, labels)
end
```

We decided to choose the most popular items to compose the landmark set L in order to maximize the number of co-rated users considered in similarity calculation. Since small number of co-rated users may lead to poor similarity accuracy [28]. Additionally, the number of landmarks is empirically adjusted according to each database experimented.

The parameters of SVR may also be varied in order to obtain higher accuracy in rating prediction. In this work, we used Radial Basis Function (RBF) as kernel function, showed by (2), to obtain non-linear separation of samples. This function has the parameter γ which may be interpreted as the inverse of the influence radius of samples selected as support vectors. It means, that the higher is γ, more support vectors are used by SVR. This parameter was empirically tuned for each database.

$$K(x, x') = exp(\gamma||x - x'||^2) \tag{2}$$

C. Toy Example

Fig. 1 depicts a toy example to illustrate the proposed modeling. In Fig. 1(a), the matrix $\mathbf{R}$ contains the item ratings given by users A, B, C, D, E and F, while missing ratings are indicated by empty cells. The first step of the proposed schema is to select the $|L|$ most rated items as landmarks. In this example, we took $|L| = 2$ for illustrative reasons and selected items 3 and 5 as landmarks. These appear highlighted in yellow.

In the next step, it was computed Cosine similarity between the items and the landmarks. These were normalized consid-

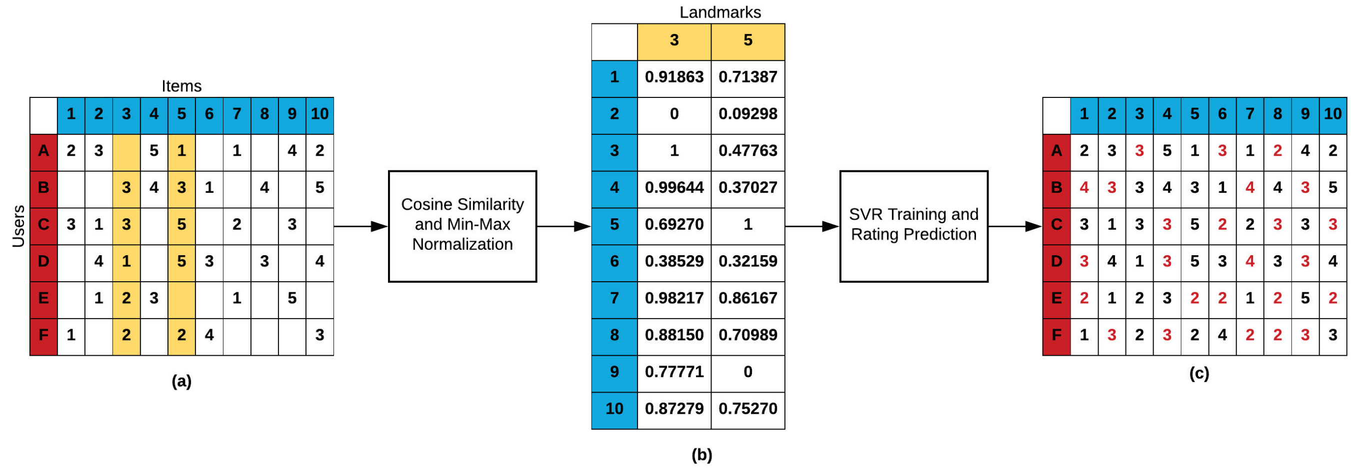

Fig. 1. A toy example to illustrate the proposed modeling. In **(a)**, the rating matrix is presented where the rows represent users $A - F$, the columns correspond to items $1 - 10$ and the cells hold rating value. The columns 3 and 5 are the selected as landmarks, which are highlighted in yellow. After computing Cosine similarity between items and landmarks and applying a min-max normalization, the similarity matrix is depicted in **(b)**. Finally, the SVR model is learned for each user and the rating matrix is filled with predictions (highlighted in red) in **(c)**.

ering the maximum and minimum values of similarities for each landmark. Thus, Fig. 1(b) depicts 1 and 0 as the highest and lowest similarity values in each column, which represent the landmarks.

Further, we learned a Support Vector Regression (SVR) model for each user considering his/her ratings as labels and the corresponding vectors of item similarities as samples. For example, user A rated items 1, 2, 4, 5, 7, 9 and 10 with values ranging from 1 to 5. Thus, we selected the similarity matrix rows corresponding to these items as samples (*i.e.* the rows 1, 2, 4, 5, 7, 9 and 10 from matrix illustrated by Fig. 1(b)) and the rating values as labels.

The SVR model positions the item ratings (samples) into landmark space, considering the vector of item similarities to landmarks (features), and tries to separate them according to the labels (*i.e.* rating values). For this purpose, it separates the samples with different labels by edges, trying to group the ones with the same label in the same region. We applied Radial Basis Function (RBF) kernel to obtain a non-linear separation of samples and, consequently, regions defined by curves.

Fig. 2 shows the decision boundaries of SVR learned for user A. As may be seen, only four regions were identified – the ones corresponding to rating values 1, 2, 3 and 4. The dark blue represents ratings with value 1, while light blue indicates ratings with value 2. Additionally, light red represents ratings with value 3 and dark red indicates the ones with value 4. It is worth noting that the ratings with value 4 and 5 of user A were grouped in the same region, which corresponds to value 4 (dark red), because of the small number of samples in the toy example and also because of the number of landmarks. In a real case scenario, we expected that all rating values will be mapped in different regions by SVR.

Finally, after learning the user models, it is possible to complete the rating matrix by presenting the similarity vector

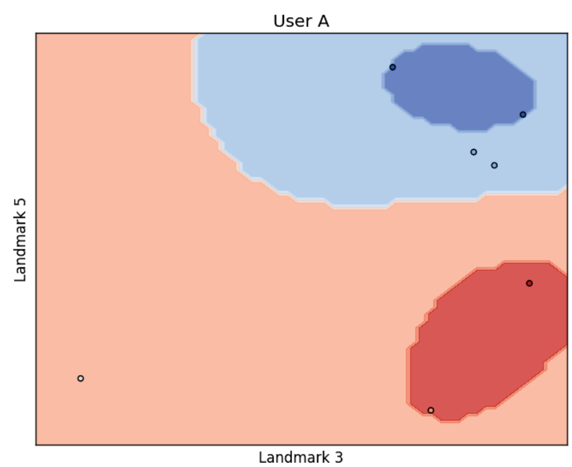

Fig. 2. SVR decision boundaries for user A in toy example. The dark blue, light blue, light red and dark red regions represent rating values classified as 1, 2, 3 and 4, respectively.

of items which were not rated by the user. In Fig. 1(c), we filled the rating matrix $\mathbf{R}$ with predicted ratings (highlighted in red).

IV. Experiments

In order to analyze the proposed modeling scheme performance, we conduct experiments on four well-known databases: MovieLens [29], [30], Netflix [31], Yahoo! Movies [32], [33], and FilmTrust [34], [35]. There were considered from MovieLens both MovieLens100k and MovieLens1M [36]. For Netflix, two cuts were made w.r.t. the chronological order, obtaining two data sets, Netflix100k and Netflix1M, containing 100,000 and 1,000,000 ratings, respectively. The other two data sets were maintained as original.

We had two objectives in the experiments: (1) to investigate the parameters (*i.e.* the number of landmarks and the parameter γ of Radial Basis Function (RBF) kernel) that lead the best accuracy for each database; and (2) to compare our proposal against the state-of-the-art CF techniques.

All experiments were carried out with 10-fold cross validation. As evaluation measures, we record the time (in seconds) required to learn the models and predict ratings, while the prediction accuracy was measured by calculating Mean Absolute Error (MAE), whose formula is presented in (3):

$$MAE = \frac{\sum_{u \in U} \sum_{i \in P} |r_{ui} - \hat{r}_{ui}|}{|T|}, \tag{3}$$

where $|T|$ stands for the number of predicted ratings in test set T and $(u, i, r_{ui}) \in T$.

V. Results and Discussion

A. Parameter Investigation

Here, we aim at finding the settings of parameters that yield the best accuracy for each database. For this purpose, we varied the number of landmarks and the Radial Basis Function (RBF) kernel's parameter γ in each database, computing the accuracy of the proposed modeling associated to each parameter value. The results presented in this section were obtained using MovieLens100k. For sake of simplicity, we omit the parameter fitting for other databases.

For determining the number of landmarks, we selected 10 to 500 most rated items, as may be seen in Fig. 3. As the number of landmarks increases, MAE decreases until it reaches 250 items. Upon this limit, increasing the number of landmarks also increases MAE. The best accuracy was yielded by selecting the 250 most rated items as landmarks. However, the difference in MAE by selecting 200 or 250 items is quite small.

As regards the RBF kernel's parameter γ, we adjusted it empirically by taking values spaced evenly on a logarithm scale from an interval starting at 10^{-4} and ending at 10^{4}. The number of landmarks was fixed in 20 items. We adopted the same γ in the Support Vector Regression (SVR) models of all user. The results are presented in Fig. 4.

As may be seen, MAE decreases with γ in interval $[10^{-4}, 12]$ and increases as the parameter assumes values greater than 12. The γ value that yields the best SVR accuracy is 11.288 and, if all SVR models were set up with this value, the achieved MAE would be 0.6805.

However, each user model is subjected to different samples during learning phase and, consequently, it is likely that different models would yield the highest accuracy with different value of γ. Thus, after adjusting the parameter for each user model and using 200 landmarks, we were able to achieve a MAE of 0.6626 in MovieLens100k.

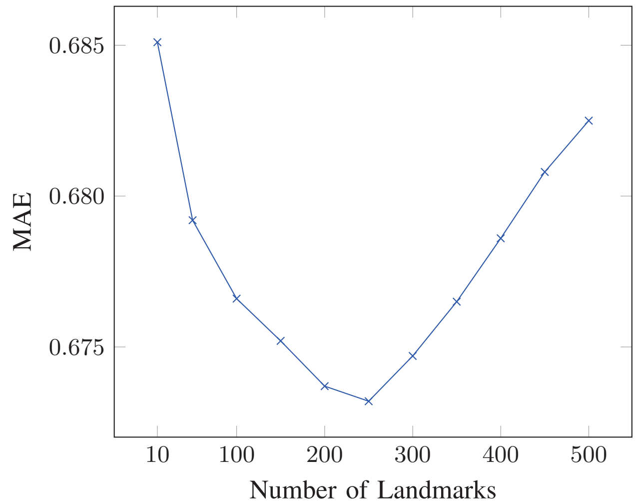

Fig. 3. Varying the number of landmarks in MovieLens100k. The RBF kernel's parameter γ was fixed in 10 for all user models.

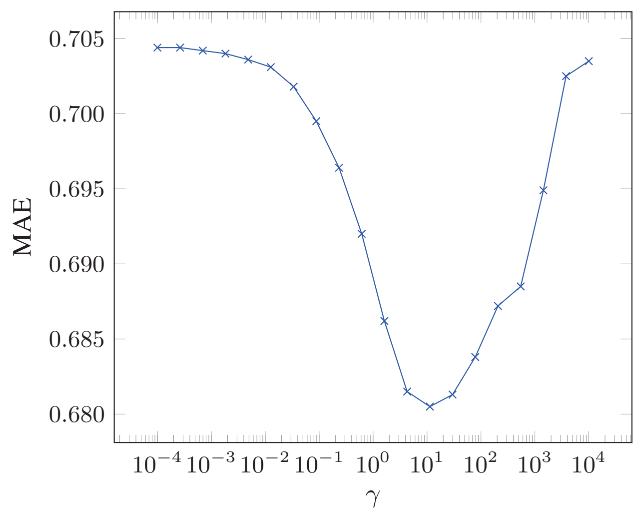

Fig. 4. Varying RBF kernel's parameter γ in MovieLens100k. The number of landmarks was fixed in 20.

B. Comparative Analysis

After adjusting parameter γ for user models and defining the number of landmarks for each database, we compared our proposal against the following state-of-the-art Collaborative Filtering (CF) algorithms: Regularized Singular Value Decomposition (RSVD) [10], Improved Regularized Singular Value Decomposition (IRSVD) [10], Probability Matrix Factorization (PMF) [11], and Bayesian Probability Matrix Factorization (BPMF) [12]. We also experimented a variation of Matrix Factorization using Alternating Least Squares (ALS) [3].

Table I presents the number of landmarks used for each database. Regarding the RBF kernel's parameter γ , it was adjusted by user model (*i.e.* γ assumes different values in different user models) and its value was chosen as described in the section before.

TABLE I
NUMBER OF LANDMARKS FOR EACH DATABASE.

Database	Number of Landmarks
MovieLens100k	200
Netflix100k	100
MovieLens1M	100
Netflix1M	100
FilmTrust	50
Yahoo	450

Table II presents MAE achieved by our proposal, namely *Landmarks SVR*, and the CF algorithms. Despite the simplicity of the proposed modeling, *Landmarks SVR* yielded the lowest MAE in three of the six experimented databases (*i.e.* MovieLens100k, Netflix100k and Yahoo). In other databases (MovieLens1M, Netflix1M and Filmtrust), RSVD or IRSVD achieved a better performance, but the difference in MAE compared to our proposal is quite small. Besides, if the algorithms were ranked by their accuracy, the proposed one would be among the top three most accurate ones.

The main advantage of the proposed modeling is the computational time required to train the models. While standard CF algorithms, like RSVD and IRSVD, take more than 30 minutes to learn the data model in MovieLens1M and Netflix1M, *Landmarks SVR* requires less than 4 minutes, as may be seen in Table III. In most of the databases, ALS was the fastest CF technique, but it was less accurate than *Landmarks SVR*.

In spite of *Landmarks SVR* learns a SVR model for each user, the number of samples used in training phase of models is quite small, which makes them converge fast. Thus, the amount of time required for learning the model is lower than other algorithms. Besides, it is easy to speed up the proposed modeling, since user models are independent and may be trained in parallel.

Therefore, we conclude that the proposed modeling is able to reduce computational time, while maintain accuracy comparable to state-of-the-art CF techniques, like RSVD and IRSVD.

VI. CONCLUSIONS

In this paper, we proposed modeling items through their similarities to preselected items, namely landmarks. Thus, instead of representing items by the ratings given by users, we represented them by their similarities to landmarks. This new modelling incorporates users' taste, since similarity measures consider the item rating during calculation.

Once items are represented by a vector of similarities (*i.e.* features), we learned a Support Vector Regression (SVR) for each user, considering the vector of item similarities, that were rated by him/her, as samples and the given ratings as labels. After the model is trained, it is able to predict ratings that is posteriorly used in item recommendation.

The proposed modeling allows one to apply Supervised Learning (SL) algorithms in the context of Collaborative Filtering (CF). We decided to use SVR, since it may be applied with different kernels to fit the data and it is easy to explain the predicted ratings through classification regions. However, other SL algorithms must be evaluated considering the proposed modeling, like Artificial Neural Networks. It must be addressed as a future work, since our objectives in this work were propose and evaluate the new modelling.

Here, we apply Cosine to compute similarity between items and landmarks. Besides, we selected the most rated items as landmarks, in order to maximize the number of co-rated users. Thus, the obtained similarities are more consistent, since more ratings are used in their computation [28]. Nevertheless, it is important to investigate other strategies for selecting landmarks and understand how these influence item representation and, consequently, the accuracy of rating prediction.

Furthermore, we varied the number of landmarks and the Radial Basis Function (RBF) kernel's γ – parameters of our proposal – to achieve higher accuracy in rating prediction. The results showed the proposed modeling is able to reduce computational runtime, while maintains competitive accuracy. It yielded higher accuracy than the state-of-the-art CF algorithms in three of the six databases experimented and it was among the top three most accurate algorithms in all databases.

As future work, a theoretical investigation should be addressed so as to determine the number of landmarks that guarantee accuracy bounds. We hope to determine lower bounds for the number of landmarks given a fixed approximation error. Besides, it may be interest to generate artificial landmarks with higher number of ratings and relevant statistical properties in order to improve similarity calculation in very sparse databases.

ACKNOWLEDGMENT

Our thanks to CNPq/CAPES for funding this research.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, June 2005.

[2] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 3:1–3:45, May 2014.

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug 2009.

[4] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Proceedings of the 9th International Conference on Neural Information Processing Systems*, ser. NIPS'96. Cambridge, MA, USA: MIT Press, 1996, pp. 155–161.

[5] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, Jan 2003.

[6] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decision Support Systems*, vol. 74, no. Supplement C, pp. 12 – 32, 2015.

[7] Y. Ning, Y. Shi, L. Hong, H. Rangwala, and N. Ramakrishnan, "A gradient-based adaptive learning framework for efficient personal recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, ser. RecSys '17. New York, NY, USA: ACM, 2017, pp. 23–31.

[8] B. Joshi, F. Iutzeler, and M.-R. Amini, "Asynchronous distributed matrix factorization with similar user and item based regularization," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys '16. New York, NY, USA: ACM, 2016, pp. 75–78.

TABLE II
MAE ACHIEVED BY OUR PROPOSAL (LANDMARKS SVR) AND THE STATE-OF-THE-ART CF ALGORITHMS IN DIFFERENT DATABASES.

Database	Landmarks SVR	ALS	RSVD	IRSVD	PMF	BPMF
MovieLens100k	**0.6624**	0.6870	0.6673	0.6697	0.6882	0.6862
Netflix100k	**0.7147**	0.7531	0.7320	0.7155	0.7324	0.7399
MovieLens1M	0.6252	0.6538	**0.6210**	0.6229	0.6480	0.6437
Netflix1M	0.6711	0.7013	0.6670	**0.6624**	0.6888	0.6904
FilmTrust	0.6115	0.6666	0.8003	**0.6001**	0.6856	0.6335
Yahoo	**0.6201**	0.6805	0.9012	0.6430	0.6866	0.6862

TABLE III
TIME (IN SECONDS) REQUIRED BY THE PROPOSED MODELING AND OTHER CF ALGORITHMS TO LEARN THE MODEL AND PREDICT RATINGS.

Database	Landmarks SVR	ALS	RSVD	IRSVD	PMF	BPMF
MovieLens100k	22.9	**14.0**	189.1	670.2	89.3	84.5
Netflix100k	13.0	**12.9**	130.0	434.1	63.7	94.3
MovieLens1M	218.9	**109.7**	2605.0	8052.9	1342.5	153.8
Netflix1M	223.4	**169.9**	1982.3	6979.2	982.8	244.9
FilmTrust	**8.0**	15.7	68.0	246.9	29.2	143.9
Yahoo	192.9	**63.7**	497.2	1436.4	247.9	675.2

[9] J. Bobadilla, F. Ortega, A. Hernando, and G. G. de Rivera, "A similarity metric designed to speed up, using hardware, the recommender systems k-nearest neighbors algorithm," *Knowledge-Based Systems*, vol. 51, no. Supplement C, pp. 27 – 34, 2013.

[10] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop*, vol. 2007, 2007, pp. 5–8.

[11] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, ser. NIPS'07. USA: Curran Associates Inc., 2007, pp. 1257–1264.

[12] ——, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 880–887.

[13] P. Gopalan, J. M. Hofman, and D. M. Blei, "Scalable recommendation with hierarchical poisson factorization," in *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, ser. UAI'15. Arlington, Virginia, United States: AUAI Press, 2015, pp. 326–335.

[14] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a bayesian probabilistic model," *Knowledge-Based Systems*, vol. 97, no. Supplement C, pp. 188 – 202, 2016.

[15] V. Sindhwani, S. S. Bucak, J. Hu, and A. Mojsilovic, "One-class matrix completion with low-density factorizations," in *2010 IEEE International Conference on Data Mining*, Dec 2010, pp. 1055–1060.

[16] V. Formoso, D. Fernndez, F. Cacheda, and V. Carneiro, "Using profile expansion techniques to alleviate the new user problem," *Information Processing and Management*, vol. 49, no. 3, pp. 659 – 672, 2013, personalization and Recommendation in Information Access.

[17] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, Aug 2017.

[18] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutirrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, no. Supplement C, pp. 109 – 132, 2013.

[19] V. De Silva and J. B. Tenenbaum, "Sparse multidimensional scaling using landmark points," Technical report, Stanford University, Tech. Rep., 2004.

[20] V. d. Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *Proceedings of the 15th International Conference on Neural Information Processing Systems*, ser. NIPS'02. Cambridge, MA, USA: MIT Press, 2002, pp. 721–728.

[21] J. C. Platt, "Fast embedding of sparse music similarity graphs," in *Proceedings of the 16th International Conference on Neural Information Processing Systems*, ser. NIPS'03. Cambridge, MA, USA: MIT Press, 2003, pp. 571–578.

[22] A. Babaeian, M. Babaee, A. Bayestehtashk, and M. Bandarabadi, "Nonlinear subspace clustering using curvature constrained distances," *Pattern Recognition Letters*, vol. 68, no. Part 1, pp. 118 – 125, 2015.

[23] H. Shi, B. Yin, Y. Bao, and Y. Lei, "A novel landmark point selection method for l-isomap," in *2016 12th IEEE International Conference on Control and Automation (ICCA)*, June 2016, pp. 621–625.

[24] H. Shi, B. Yin, X. Zhang, Y. Kang, and Y. Lei, "A landmark selection method for l-isomap based on greedy algorithm and its application," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 7371–7376.

[25] C. Orsenigo, "An improved set covering problem for isomap supervised landmark selection," *Pattern Recognition Letters*, vol. 49, no. Supplement C, pp. 131 – 137, 2014.

[26] F. Braida, C. E. Mello, M. B. Pasinato, and G. Zimbro, "Transforming collaborative filtering into supervised learning," *Expert Systems with Applications*, vol. 42, no. 10, pp. 4733 – 4742, 2015.

[27] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*. Boston, MA: Springer US, 2011, pp. 1–35.

[28] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," *Knowledge-Based Systems*, vol. 56, no. Supplement C, pp. 156 – 166, 2014.

[29] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 19:1–19:19, Dec. 2015.

[30] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl, "Movielens unplugged: Experiences with an occasionally connected recommender system," in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, ser. IUI '03. New York, NY, USA: ACM, 2003, pp. 263–266.

[31] J. Bennett and S. Lanning, "The netflix prize," in *Proceedings of KDD cup and workshop*, vol. 2007, 2007, p. 35.

[32] T. Ebesu and Y. Fang, "Neural semantic personalized ranking for item cold-start recommendation," *Information Retrieval Journal*, vol. 20, no. 2, pp. 109–131, Apr 2017.

[33] H. P. Vanchinathan, A. Marfurt, C.-A. Robelin, D. Kossmann, and A. Krause, "Discovering valuable items from massive data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: ACM, 2015, pp. 1195–1204.

[34] J. Golbeck and J. Hendler, "Filmtrust: movie recommendations using trust in web-based social networks," in *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.*, vol. 1, Jan 2006, pp. 282–286.

[35] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel bayesian similarity measure for recommender systems," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI '13. AAAI Press, 2013, pp. 2619–2625.

[36] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and*

Development in Information Retrieval, ser. SIGIR ’99. New York, NY, USA: ACM, 1999, pp. 230–237.