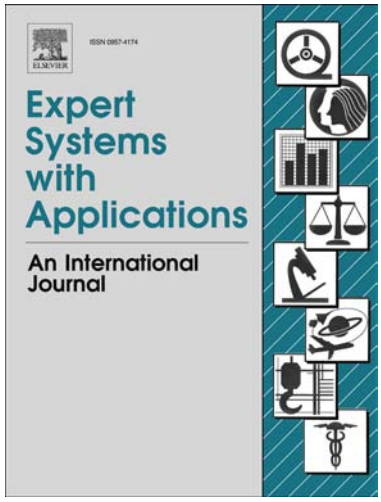

Transforming collaborative filtering into supervised learning problem

Filipe Braida do Carmo, Marden Braga Pasinato, Carlos Eduardo Mello, Geraldo Zimbrão

PII: S0957-4174(15)00038-X
DOI: http://dx.doi.org/10.1016/j.eswa.2015.01.023
Reference: ESWA 9803

To appear in: *Expert Systems with Applications*

Please cite this article as: do Carmo, F.B., Braga Pasinato, M., Eduardo Mello, C., Zimbrão, G., Transforming collaborative filtering into supervised learning problem, *Expert Systems with Applications* (2015), doi: http://dx.doi.org/10.1016/j.eswa.2015.01.023

Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 00 (2014) 1–11

Expert Systems with Applications

Transforming Collaborative Filtering into Supervised Learning Problem

Filipe Braida do Carmo[a,b,*], Marden Braga Pasinato[a], Carlos Eduardo Mello[b], Geraldo Zimbrão[a]

[a] *PESC/COPPE, Universidade Federal do Rio de Janeiro, CT Bloco H, Cidade Universitária - Rio de Janeiro, Brazil, P.O. Box: 68511, +55 21 2562-8672*

[b] *DCC/IM, Universidade Federal Rural do Rio de Janeiro, Nova Iguaçu - Rio de Janeiro, Brazil, Zip-Code: 26020740, +55 21 2667-0105*

Abstract

Collaborative filtering, a well-known approach for recommender system, aims at predicting user ratings for items based on items previously rated by others. Due to the difficulty to represent such ratings in an unique feature space, it becomes infeasible to build supervised training sets from which one is able to induce learning models. There have been several methods in the literature tackling the problem of building a feature space for collaborative filtering. However, those proposals usually rely on domain information which might be not available in all settings. In this context, this work aims at proposing a methodology that transforms the classic collaborative filtering setting into the supervised learning problem addressed by classic machine learning algorithms. For that, the methodology allows to build feature space from latent variables underlying rating space matrices. Experiments have yielded satisfactory performance over the classic collaborative filtering methods that exist in the literature.

Keywords: Recommender System, Dimensionality Reduction, Supervised Learning

1. Introduction

The growth of the Internet has catapulted a myriad of new business and applications by reshaping many human activities. For instance, there has been a significant shift in trading towards more flexibility, availability, and mobility. In this context, e-commerce systems have emerged to offer users all these features. A million of such systems have been deployed throughout the globe each one offering an unlimited catalog of goods and services, overloading users with to many option, driving process of choice become more difficult. This overload has become users anxious by preventing consume (Schwartz, 2005).

In this scenario, recommender systems (RS) have emerged playing an important role by providing recommendations for users. These have been widely adopted by big Web players such as Amazon[1], NetFlix[2], YouTube[3] and others. Besides their obvious virtues of leveraging sales, RS also improve customer loyalty as well as increase cross-sales. In fact, these systems try to efficiently meet needs and interests of users by avoiding the burden of finding a needle in a haystack (Schafer et al., 1999).

As one of the most successful RS approaches, Collaborative Filtering (CF) learn from users evaluations, often represented by rating, in order to make item recommendations. This is often modeled through a $n_{users} \times m_{items}$ matrix R, where each position r_{ij} of R represents rating given by an user i to an item j. Not rare, rating matrix is sparse, in the sense that, the amount of ratings fulfill only a small percentage of all possible positions in the matrix. For instance, one may have a rating matrix containing positions filled by rating values (*e.g.*, ranging from 1 to 5) and unfilled positions a non-existent rating symbol $\varnothing$.

Accordingly, the main issue is to predicting ratings for unfilled position based only on the fulfilled ones (ratings). In other words, CF techniques do not require any other input than the rating matrix itself. Unfortunately, such matrix model has created a barrier for other methods that requires ratings represented in an input space like Supervised Learning.

This rating matrix representation issue avoids with which

*Corresponding author

Email addresses: filipebraida@ufrj.br (Filipe Braida do Carmo), marden@ufrj.br (Marden Braga Pasinato), carlos.mello@ufrj.br (Carlos Eduardo Mello), zimbrao@cos.ufrj.br (Geraldo Zimbrão)

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

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

[3] http://www.youtube.com

one learns a model mapping appoint into the input space to a rating value, once there is no straightforward features associated to a given rating by an user i to an item j.

There have been proposals that attempt to transform the rating matrix into a typical SL dataset suitable for ML methods. Nevertheless, those transformations usually rely on the domain information which can be hard to extract and even misleading (Cunningham & Smyth, 2010; Hsu et al., 2007; O'Mahony & Smyth, 2010, 2009).

Moreover, a transformation process based on available domain information is still a complex task only specialists can execute. The transformations proposed so far are thoroughly crafted using the domain information and are also very specific. Thus it is possible to say that no straightforward domain-independent transformation has even been proposed to allow ML methods to be fully explored as CF techniques.

In this context, this work proposes a straightforward domain-independent transformation in the sense that it is a automatic process that can used by any layperson and only requires the rating matrix as input. The unfilled positions of the rating matrix are mapped into a k-dimensional vector space corresponding to the k most important latent factors of the matrix. Each point in this feature space is associated to its rating value forming a typical SL dataset. ML methods are trained with this dataset and applied to predict the users' unknown ratings. Their performance was given according to metrics such as MAE and RMSE, which indicate that this approach has greatly outperformed classical CF techniques.

This paper is organized in 5 sections of which this is the first one.

2. Related Work

Collaborative filtering has been largely used in recommender systems and the issue of sparse matrix has been addressed and handled with adaptations in classic classification/regression methods. For instance the user-based collaborative filtering predicts ratings based on the k-Nearest Neighbors classifier. In order to deal with the sparsity of users' ratings, the similarity measures between users is adapted to consider only the co-rated items in the computation (Adomavicius & Tuzhilin, 2005).

Due to this sparsity issue, several works have been proposed to tackle this problem. Many works apply dimensionality reduction so as to transform sparse matrix in a fixed space of features. (Sarwar et al., 2000) proposes the use of the Singular Value Decomposition (SVD) and recommendations are generated by operations between the resulting matrices.

Another approach to try to deal with the sparsity problem is to treat it as classification/regression problem. An approach for this is to derive feature from the rating matrix based on the domain knowledge (Cunningham & Smyth, 2010; Hsu et al., 2007; O'Mahony & Smyth, 2010, 2009). For instance, the users' mean of ratings and the standard deviation are used as features to train classifiers.

In (Billsus, 1998), authors propose a domain-independent method in order to predict ratings. For each user a model is induced based on a built matrix of features derived from the original rating matrix. Although this proposal performs good results, it does not work for multiclass problems and is not scalable as long as the number of users increases, the more models we have to induce. This method uses SVD to reduce the dimensionality and build a new feature space to train the models.

To the best of our knowledge there is no work that proposes a general domain-independent transformation of the rating matrix into a training set for applying classic supervised learning method.

3. Proposal

There are two main approaches for collaborative filtering: memory-based and model-based. The former is usually based on lazy supervised learning algorithms, such as k-Nearest Neighbours whereas the latter is usually concerned with building supervised models from the user-item rating matrix.

Memory-based algorithms handle predictions through similarities between either pair of users or items. In this way, ratings are predicted for a certain user on a target item, considering only the ratings of the most similar users who have rated the target item. A wide range of similarity measures has appeared in the literature, to improve rating predictions. This class of algorithms is also mostly named as collaborative filtering based on heuristics (Adomavicius & Tuzhilin, 2005).

In contrast to memory-based algorithms, the model-based collaborative filtering attempts to induce learning models for rating predictions. Machine learning techniques and data mining algorithms are often used to support such models, exploiting and identifying patterns to predict ratings (Adomavicius & Tuzhilin, 2005).

The main issue related to Collaborative Filtering lies in the space matrix representation. As many users do not evaluate the majority of the items, the rating matrix becomes highly sparse. This matrix representation, along with the sparse characteristic, creates a considerable challenge for learning the underlying users' preferences behind the ratings. In fact, one should understand the ratings as outcomes of the underlying preferences of the users. Therefore, a method that aims at exploiting this information behind the data more straightforwardly would lead to better predictions. Moreover, one should explicitly represent such underlying preferences throughout the use of latent semantic variable analysis.

In this context, this work proposes a novel method that transforms the collaborative filtering representation, *i.e.*, the sparse rating matrix, into a classic supervised learning task based on the underlying user preferences behind the

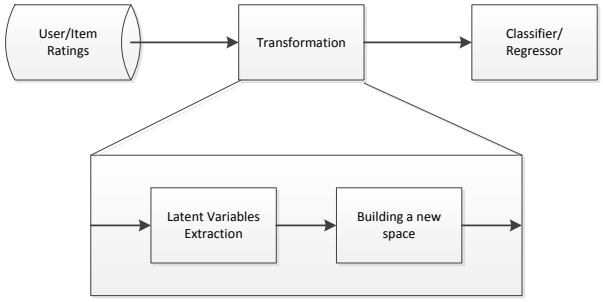

Figure 1. Procedure for transforming the collaborative filtering problem to supervised learning.

rating matrix. To do so, a methodology to transform the collaborative filtering problem into a supervised learning named COFILS – *Collaborative Filtering to Supervised Learning* is defined, as well as its implementation. This transformation consists of applying a set of techniques by building a new input space where users and items are represented throughout their latent variable values. This representation allows us to apply directly classification techniques, such as Logistic regression, to learn models for rating prediction. This procedure is illustrated in Figure 1.

Thus we can define the processing illustrated in Figure 1 which can be divided into two steps. The first is the extraction of the user and items latent variables. The second would be to build a model that would use the information from the first stage and correlate with notes, to allow us to create a training set to apply to a machine learning algorithm.

Latent semantic variables somehow explain the hidden relationships between users and items across the given ratings. In (Koren et al., 2009) an analogy with users and movies is provided. There, a ratings matrix of users-by-movies is decomposed into two matrices (U and V) where U is a matrix of users-by-factors and V is a matrix of movies-by-factors. Thus users and movies are defined as vectors of factors. Each factor might be understood as a hidden movie category such as comedy, horror, drama, etc. Those techniques usually decompose a matrix into other matrices by exploiting linear algebraic decompositions. There are many techniques from liner algebra to decompose a sparse matrix such as SVD, Tensors, QR, PCA , and others (Koren et al., 2009).

This technique allows us to represent users/items in the same space of features, *i.e.* a space of factors. Besides, the space is completely defined according to the number of factors. However, this is not suited yet to predicting ratings as long as we do not have rating representations for the same space of features. As regards the users and item factors the only component that is necessary is a model that correlates these features with the ratings, to build the training set to apply in a classifier or regressor.

In the following subsection, we discuss the two stages of the proposal. The first to be discussed regards the extracting of the latent variables problem. In this stage we discuss the issues involved in the extraction of the features, due to the lack of data and to the characteristics of the problem. Finally we discuss the model construction that uses this information into a ratings prediction using a supervised machine learning algorithm.

3.1. Transformation

3.1.1. Extraction of the Latent Variables

In the context of collaborative filtering, each user u and v item can have a relationship and this observed value is represented by the tuple $\overrightarrow{\{U_i; U_j\}}$. The mapping of user characteristics and the item with that observed value can be used to train a supervised learning algorithm. The major challenge is that the mapping is not explicitly defined, *i.e.*, $U_j \times U_j \rightarrow \overrightarrow{\{U_i; U_j\}}$. There is no observed information about the users and the items, thereby preventing the direct application of this algorithms class.

However it can be considered that the user and item have characteristics that represent one's behaviour, but they cannot be directly observed. In the literature these variables are named latent variables and they are inferred using a mathematical model or derived from other observed variables, which in this case would be the user's preference for the item.

The problem is that the user preferences matrix is sparse. This characteristic prevents the application of the algorithms for extracting latent variables. So it is necessary to somehow represent the unknown ratings.

The first and naive approach is to consider the unknown ratings as a distinct value of the preference set. An example, using integer numbers with values between one and five, it to consider this rating as zero. The downside of this solution is the inclusion of information that clashes within the matrix. This value below the lowest rating value and thus changing the rating semantics, *i.e.*, the unknown item to the user would be considered an awful item. So this solution adds a large error into the extraction of latent variables.

Another similar approach is considering the missing values as the average global ratings. This solution considered that the unknown values tend to average. Another aspect is using the average user or item as a substitute for unknown value and thus minimizing the overall error in comparison with another solution. The main difficulty is when there is an user and/or item that has no ratings or has a small quantity. In this case a global average could be used as a substitute.

So, the problem is to find an unknown values' representation that minimizes the error in the latent variables extraction. The best representation of a rating matrix is in predicting unknown values. In (Candès & Recht, 2009), shows that rating matrix can be approximated by a matrix with a low rank and applying matrix completion techniques to fill this.

Finally, any collaborative filtering algorithm could be used for the purpose discussed. The major problem with

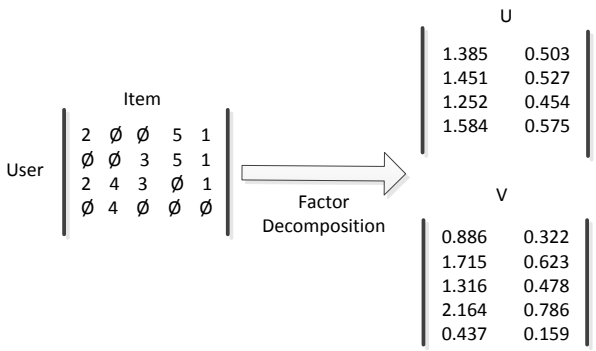

Figure 2. SVD factorization with 2 factors

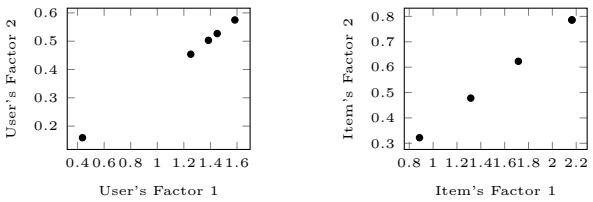

Figure 3. User (left) and item (right) in factor space

this approach would increase the solution complexity, and may not be useful for this purpose if the gain in results does not do justice to this increase. Especially if the collaborative filtering algorithm has a great cost to predict only one rating as that would be applied to the need to provide all unknown values.

Figure 2 illustrates the Singular Value Decomposition factorizing the rating matrix R. For this operation two factors were considered and then two matrices U and V were produced. Note that the two columns represent the two factors and in both matrices U and V each row represents an user and an item, respectively.

Figure 3 depicts the graph representation for matrices U (Figure 3 on the left) and V (Figure 3 on the right) as shown above in Figure 2. There, we can see how users and items are spread over the space built on the factors. In the U matrix graph (Figure 3 on left) the distances among users can be calculated and used as similarities among them. The same might be done for matrix V represented on the right.

3.1.2. Building a new space

In the previous section, we saw that there are latent variables user and item that define the behaviour of each one and there are several methods that extract these variables on a rating matrix. Then, we can define that after applying the techniques of extracting latent variables discussed earlier, for each user i there is an array of features $\bar{C_{Ui}}$ as well as for each user j vector $\bar{C_{Vj}}$.

With these two vectors the relation $U_j \times U_j$ is defined, *i.e.*, with the mapping that correlates the user and item defined by its features $\{\bar{C_{Ui}}|\bar{C_{Vj}}\}$ and this implies a rating $\{U_i; U_j\}$. Through this new space construction it will be

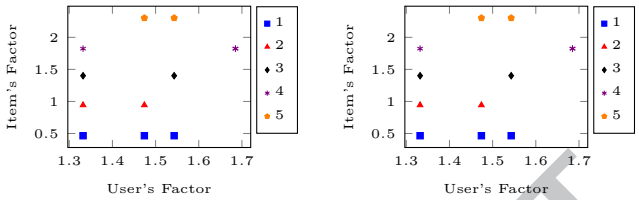

Figure 5. Training set space generated in the previous example (left) and without normalization, with normalization by the average user (right).

possible to build a training set to be used in a supervised learning algorithm.

As seen before, the machine learning algorithms can perform supervised complex mapping runs between input and output, *i.e.*, these algorithms attempt to infer a function Θ. Depending on the supervised learning algorithm used, this function becomes linear or not. Thus, the recommendation problem becomes a supervised learning problem where the goal is to build a model that will find the best mapping to a function Θ. Then we can rewrite the problem as $\Theta\left(\bar{C_{Ui}}|\bar{C_{Vj}}\right) \rightarrow \{U_i; U_j\}$. This process is illustrated in Figure 4.

In Figure 5 (left), the supervised training dataset built from the rating matrix of the toy example is shown. The instances represented are produced from the factor deposition using SVD for one factor. Thus the horizontal axis shows the factor values for users and the vertical axis shows the factors for items. This space is linear-separable and therefore we can induce a linear classifier to make rating predictions.

The parameter variation, *e.g.*, the number of factors for each latent variable, the technique used for the extraction of the variables or the unknown value definition can generate several different spaces. Thus, a classifier/regressor performance can change depending on these parameters as they affect their choice, *e.g.*, the distribution of the feature space.

This parameter-dependence problem used in the transformation is shown in Figure 5 (right). In it we used the previous example, but employing the user's mean rather than the global average zero to replace unknown values. Note that another distribution of instances was generated and, depending on such distribution it is possible to benefit some classifiers / regressors more than others.

4. Experiments

In this section the experiment conducted is described and its results are discussed. Our experiments aim at evaluating the proposed transformation and comparing whether it is possible to outperform classic collaborative algorithms.

In order to evaluate the proposal, two specific experiments were performed. The first focuses on identifying on how different classifiers/regressors behave with different operators by varying the number of factors generated by the SVD. In the second, holding the best parameters

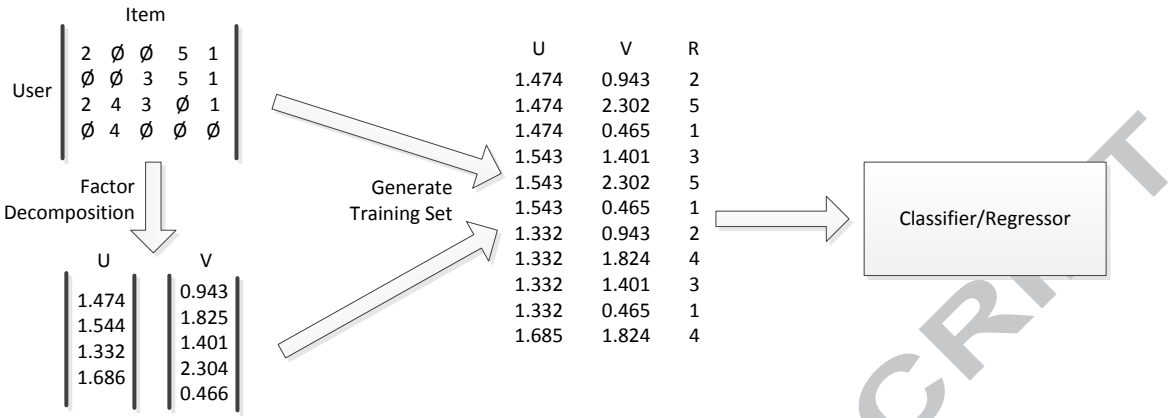

Figure 4. Procedure for transforming the collaborative filtering problem to supervised learning.

from the first experiment, a comparison in terms of prediction accuracy is made among the User-based Collaborative Filtering, Item-based Collaborative Filtering, Improved Regularized SVD (IR SVD) (Paterek, 2007), and the best classifier/regressor, using our proposal.

4.1. Data Base

The MovieLens dataset was used in both experiments (Adomavicius & Tuzhilin, 2005). It contains 100,000 ratings given by users on movies. The rating value varies from 1 to 5. There are 943 users and 1,682 movies. There is also users and items' information such as user age, gender, movie title and release date, although none of them was taken into account. In the last experiment the 1M dataset was used and it exists 6040 users, 3952 movies and 1,000,209 ratings.

4.2. Setup

The methodology consists of experiments to evaluate the technique performance varying parameter combinations. As the method used for the latent variable extraction, the supervised learning algorithm and number of variables, to compare with the recommendation classical techniques system literature as nearest neighbours, Regulared SVD and Improved regularized SVD (Paterek, 2007).

The proposed methodology (COFILS) has three steps: preprocessing, latent variables extraction, and regression / classification. The first step will be responsible for representing the data and giving a solution to the unknown values representation. At this stage three approaches were chosen, the first being to use zero to represent the unknown values and without normalization. The other two normalized the relationship with matrix media user or item and thus the missing data would make the average normalizing factor, *i.e.*, the value zero. This transformation is shown in Figure 6.

In the latent variables extraction a set of three techniques will be used. The first is the singular value decomposition (SVD) using the matrix U as latent variables user and the matrix V as the item, and discarding the singular values, with the same strategy adopted for the Latent Semantic Analysis (Linden et al., 2003).

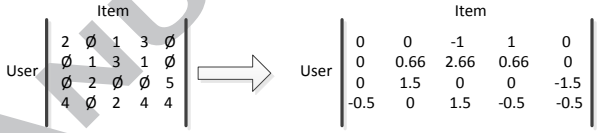

Figure 6. Preprocessing using the normalization user.

The improved regularized SVD technique will also be used. This technique is a linear regression model and its two variables are defined (p and q) that correlate user attributes and items respectively. The last is the Partial SVD of the LingPipe [4] library. It is used in sparse matrices and uses gradient descent technique as training to minimize the error due to unknown values to rebuild the array.

Finally, the last step is to apply the supervised learning algorithm. For this study three techniques will be used: Bayesian learning, artificial neural networks (ANN), and random forest. The first technique was chosen because of its simplicity and it will be possible to evaluate the proposed techniques with a simple machine learning. The second technique was chosen due to its robustness as mentioned in the literature. Random forest was chosen because of its accuracy (Caruana & Karampatziakis, 2008).

In short, there are four parameters that must be defined before being applied to the proposal which are:

1. Technical preprocessing and normalization: normalization by the average user, item and without normalization.
2. Technical latent variable extraction: SVD, improved regularized SVD and Partial SVD
3. Quantity of Latent variables used

[4] http://alias-i.com/lingpipe/

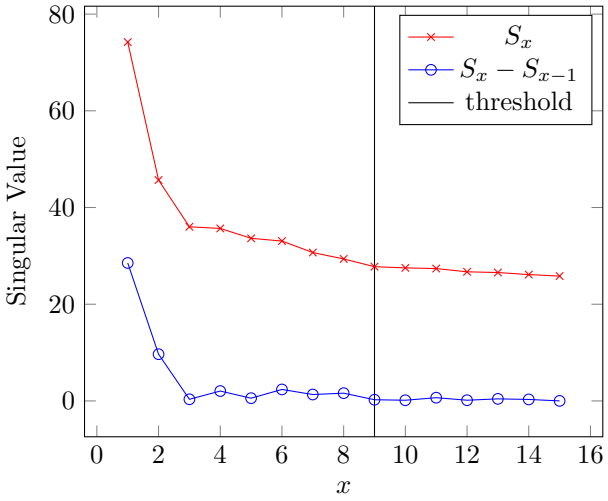

Figure 7. Quantity and gain information that each latent variable.

4. Technical supervised learning: Bayesian learning, ANN, and random forest

In addition to these four parameters, there are many other elements that have to be defined for each machine learning technique. In the case of the neural network, it uses a network of multilayer perceptrons, and the architecture is used with two hidden layers and the transfer function is the sigmoid one. Random forest has an unique parameter that is the amount of decision trees used.

The parameters that can influence the result would require numerous tests. In this work we only validate our proposal, so we are not looking for the optimal parameters but that which leads to better results than those of classical algorithms. This is enough to show the proposal potential.

The first experiment has the purpose to define the supervised learning techniques' configuration to be used in the following experiments. In the case of the neural network it will define the number of neurons in the hidden layer. In the random forest it is the number of decision trees used in boosting. The Bayesian learning algorithm will not be present, and thus requires no parameter to be set.

The problem with this experiment is to define the other three parameters (normalization, extraction technique of latent variable, and latent variables quantity) to be used. The decision was to use a normalization by the average user. The latent variables extraction technique that was used is the SVD. This was chosen because of its importance in the latent semantic analysis area and simplicity. The only missing parameter is the latent variables' quantity.

In order to determine the minimum number of latent variables to be used the SVD algorithm was applied to the user preferences matrix to analyse the singular values S, in Figure 7. For the visualization effect we calculate the difference between each singular value and its predecessor $(S_x - S_{x-1})$ as it can be seen in Figure 7.

There is a large drop in the value of the latent variable octave. So it was determined that this value should be set for the experiment for both the user and as for the movie. Thus, the first experiment was determined, aimed at defining a configuration for supervised learning algorithms.

The second experiment will have to evaluate the proposed by varying the four parameters described above. Through it you can check the behaviour of the supervised learning technique as regards the latent variables number, the extracting latent variables technique, and the matrix preprocessing. There is a relationship between these parameters and the performance of the classifier/regressor as they affect the distribution of the data in the new latent variable space.

After this experiment we will show the performance of some the collaborative filtering main techniques aiming at comparing it with the best result of the previous experiment. Four algorithms were chosen using both the memory and the model approach that are the user and item nearest neighbours and each using the cosine and pearson similarity, Regulared SVD, and Improved Regulared SVD.

Nearest neighbours technique has a weakness: a minimum number of neighbours must to be set, leading to overall coverage decrease. For a fair comparison, an experiment was carried out, using COFILS best configuration from the second experiment and predicting rates only for the users or the films that have a minimum number of neighbours. In this way we get the same coverage for both techniques: COFILS and nearest neighbours. This experiment checks overall impact of user or movies with few neighbours.

With the objective to validate the proposal, we run the experiment in another database. The goal is to demonstrate that the proposal is valid with another dataset that has distinct characteristics compared with the first.

A 10-fold cross validation is performed in each experiment step in order to obtain robust results. The measures of accuracy used in both experiments were the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) (McLaughlin & Herlocker, 2004).

Briefly, five experiments will be to evaluate the proposal of this work:

Experiment I

Evaluation the ANN and random forest settings using the normalization by average users, eight latent variables and extraction through decomposition by SVD.

Experiment II

Evaluation of Bayesian learning algorithms, random forest and ANN using the best configuration of the previous experiment and varying the normalization, the latent variables quantity and extraction technique.

Experiment III

Comparison with the best configuration of the experiment two with four technical literature of collaborative filtering: user and item nearest neighbours and for

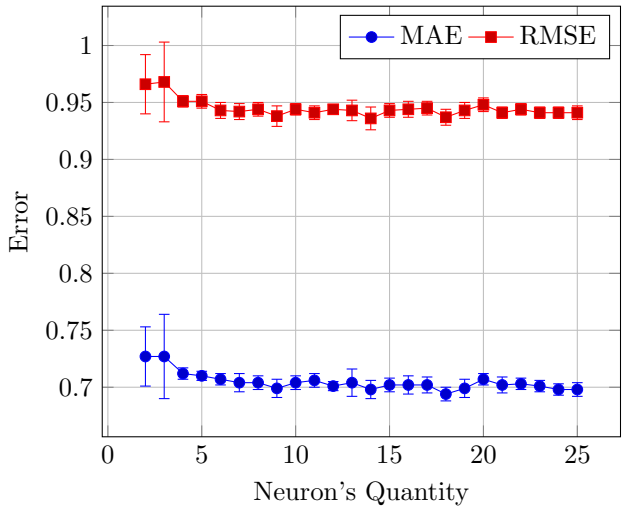

Figure 8. Evaluation of ANN technique varying the neuron's quantity.

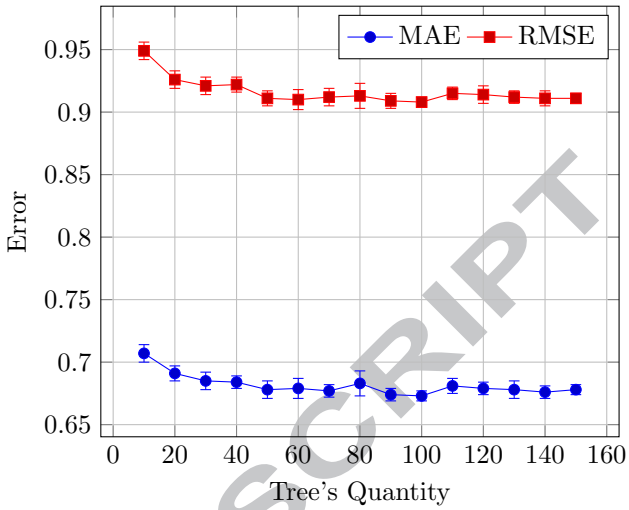

Figure 9. Evaluation of random forest technique by varying the quantity of decision trees.

each using the cosine and pearson similarity, Regulared SVD and Improved Regulared SVD.

Experiment IV

Evaluation varying the minimum neighbours quantity to be predicted using the best setting of the second experiment.

Experiment V

Comparing the best configurations in another base.

4.3. Results

In this section we will discuss the results presented showing the results obtained by varying settings. Finally the results of classic techniques recommendation system will be compared with the best configuration.

In experiment I it was found that there is a downward trend in the error when the neuron quantity in the neural network (Figure 8) is increased as well as decision trees in random forest (Figure 9). In the case of the neural network there is a stabilization of the error from eight neurons and random forest trees from fifty. In both cases, the increased training time is correlated with the increased number of parameters.

In this experiment it was found that from eight neurons in the hidden layer, the error rate tends to fall and not only with increasing the time of learning. It was then used in the following experiments in the configuration of ten neurons. The random forest had a sharp initial fall and then tended to stabilize the error. This stabilization was from the value of fifty decisions trees.

Experiment II aimed at checking possible proposal configurations and the machine learning algorithms parameters were defined in the first. Basically there were two distinct behaviours in these results. The first is when you increase the number of features, which also improves performance, or better, when you increase the knowledge on the data there is an boost in performance up to a limit (neural

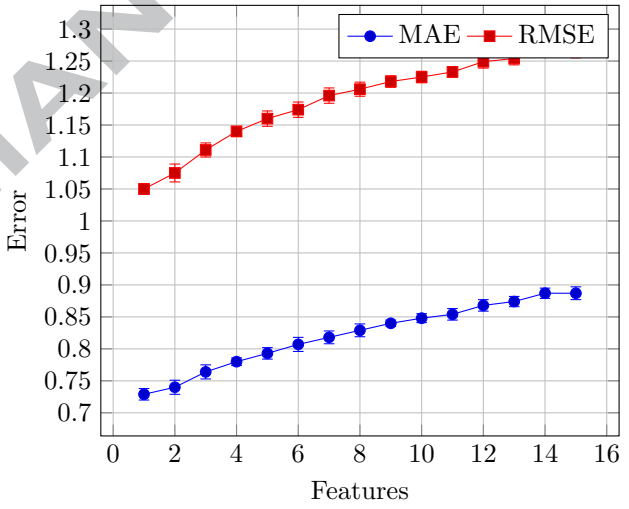

Figure 10. Evaluation of experiment II using Naive Bayes, without normalization and using Partial SVD.

networks and random forest), the other being the opposite, *i.e.*, worsening the outcome (bayesian learning). An example of the performance of the Naive Bayes case can be seen in Figure 10, ANN in Figure 11 and random forest in Figure 12 random forest.

Table 1 shows the best results in Experiment II, separating the best result for the normalization and algorithm supervised learning. Because the choice was to use two metrics for evaluating performance, and not necessarily the best one, the other two columns will be show where the best outcome regarding each metric was presented.

As shown in the results the choice of data representation with latent variables had a different influence on each supervised learning algorithm, showing the hypothesis that the choice of extracting the latent variables technique influences each algorithm differently. In general, the Partial SVD generated the best result for the Bayesian learning

Table 1. The best results of the proposed technique by varying its parameters

		Best MAE				Best RMSE			
Technique	Normalization	MAE	RMSE	Features	L.V.E.T.[1]	MAE	RMSE	Features	L.V.E.T.[1]
N. Bayes	-	0.729	1.050	1	SVDL[2]	0.729	1.050	1	SVDL[2]
ANN	-	0.691	0.930	2	SVDL[2]	0.691	0.930	2	SVDL[2]
	User	0.689	0.932	3	SVDL[2]	0.689	0.932	3	SVDL[2]
	Item	0.700	0.944	2	SVDL[2]	0.703	0.942	6	SVD
Random Florest	-	0.687	0.930	6	SVDL[2]	0.690	0.918	9	SVDL[2]
	User	**0.681**	0.915	7	SVD	0.683	0.914	8	SVD
	Item	**0.681**	0.914	12	SVD	0.682	**0.913**	8	SVD

[1] Latente Variable Extraction Technique;
[2] Partial SVD by LingPipe

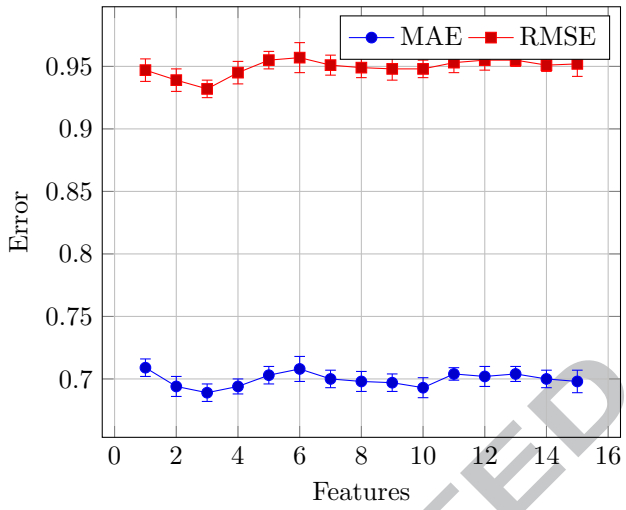

Figure 11. Evaluation of experiment II using ANN, normalization through the average user and using Partial SVD.

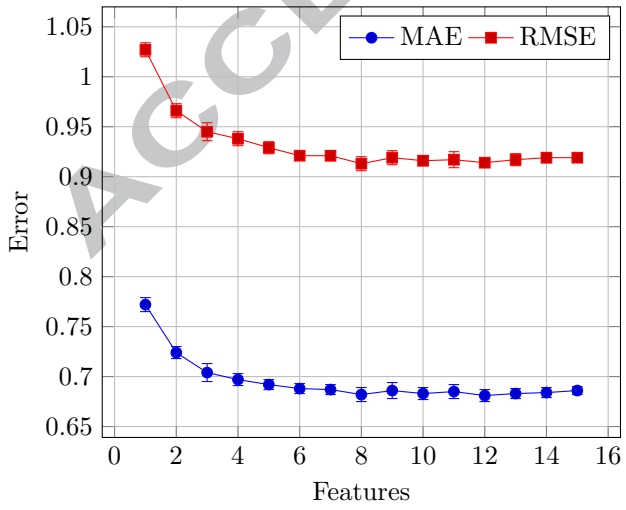

Figure 12. Evaluation of experiment II using random forest, normalization through the average item and using SVD.

and neural network, and the pure SVD in the random forest.

Besides the machine learning technique, the amount of latent variable affects the results in different ways depending on the chosen algorithm. The Bayesian Learning got a better result with a few variables. The neural network had a better result with four, and the random forest with eight variables. When the result is better, using the SVD technique for latent variables extraction, gave the best results with eight variables. The same quantity as was evaluated on the Figure 7.

During the proposal presentation we hypothesized that the way to complete the missing values of user/item matrix directly would impact the results obtained using either zero or some average. As it can be seen from the results, the difference between using the normalization (average user or item) against the simplest way to represent the data (using zero values unknown to us) did not generate a big difference in the result: the average difference was below 1%.

Experiment III was responsible for evaluating the classic techniques of collaborative filtering using the same methodology that was used to evaluate the experiments of our proposal. The Improved Regulared SVD technique obtained a better result compared to the techniques of collaborative filtering based on Regulared SVD and memory approach, but when it is given a limit on the minimum number of neighbours in the memory-based techniques, these get a good result, and even better based on that model. A problem of this limitation is the reduced coverage.

The results obtained using our proposed were consistently better than those obtained with the classic collaborative filtering algorithms. Our proposal generated an improvement of about 5% in MAE and 4% compared with the RMSE of the best collaborative filtering algorithm. These results can be seen in Table 2.

In Experiment IV we evaluate the performance of the best configuration obtained in Experiment II (random forest, using SVD and eight latent variables) varying the minimum number of neighbours required to perform the prediction. The goal is to compare the performance of the proposed technique with the nearest neighbours. The

Table 2. Best results of collaborative filtering techniques

Technique	MAE	RMSE	Coverage	Features
COFILS	0.681 (0%)	0.914 (0%)	100%	12
SVD by Funk	0.722 (-6.0%)	0.980 (-7.2%)	100%	14
Regulared SVD	0.715 (-5.0%)	0.952 (-4.2%)	100%	38

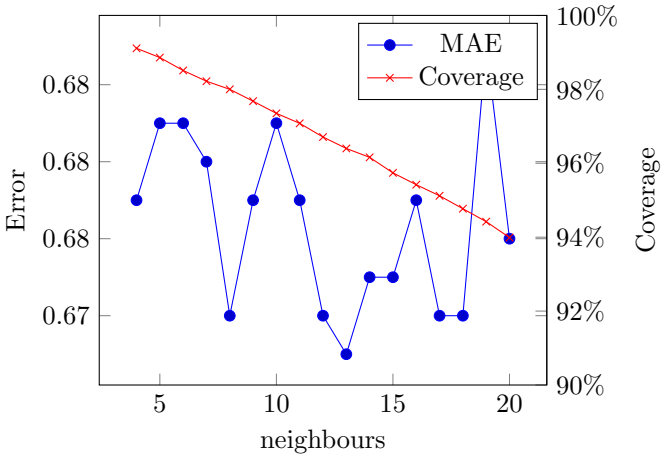

Figure 13. Evaluation of experiment IV using random forest, normalization through the average item and using SVD.

minimum number of neighbours influence on performance when the threshold is set. Even limiting to 50 the number of neighbours in the memory-based techniques, the proposal has received a better and complete coverage of the basic test result. The result can be seen in Figure 13 and a comparison is made with the results obtained in the Table 3.

Limiting the minimum number of neighbours in the proposed algorithm produced a drop of about 1% MAE and 2% in RMSE. The best results occurred when the proposal to limit the number of neighbours came to the value of thirteen. Defining this minimum as a parameter, the proposal received an improvement compared to the best result of the algorithms based on a memory of approximately 6% in MAE and 5% in RMSE.

Moreover, the best model-based techniques (Improved Regulared SVD), generated the best results when it used 38 latent variables while our proposal already obtained a better result with only two. The complexity of the SVD algorithm is directly proportional to the number of latent variables to be calculated, where our proposal reduces the time required for preprocessing (SVD).

Only the MAE or RMSE are not enough to understand the magnitude of the recommendation being made. To this end the confusion matrix of the user-based collaborative filtering algorithm was calculated using cosine similarity and the proposal's best configuration. Table 4 is the user-based collaborative filtering and Table 5 the result of the proposal.

A major difficulty in the recommendation, within the

Table 4. Collaborative filtering based on user using cosine similarity and a minimum of thirteen neighbours.

		Prediction				
		1	2	3	4	5
Truth	1	8.9%	33.5%	45.5%	11.8%	0.4%
	2	1.0%	18.7%	56.2%	22.9%	1.2%
	3	0.3%	7.5%	52.6%	37.9%	1.7%
	4	0.0%	2.2%	35.5%	58.1%	4.2%
	5	0.1%	1.1%	18.6%	66.6%	13.5%

Table 5. Confusion matrix of the proposed algorithm using the random forest, SVD, normalization by the average user and eight latent variables.

		Prediction				
		1	2	3	4	5
Truth	1	9.0%	33.1%	45.1%	12.4%	0.5%
	2	0.8%	18.6%	58.9%	21.0%	0.7%
	3	0.0%	6.8%	53.4%	39.0%	0.8%
	4	0.0%	1.4%	30.6%	63.8%	4.2%
	5	0.1%	0.8%	14.1%	67.5%	17.6%

collaborative filtering is to predict a low rating. The algorithms have a tendency to have a high error in these cases, as they tend to classify the average score and as it can be seen in Table 4. Through the confusion matrix one can see that both techniques error a great deal in this regard. In both cases the trend is to recommend to grade three, *i.e.*, the average grade of the database. A big surprise was the case when the proposed algorithm reports that the recommendation is to rating one, and the probability that the algorithm is correct is high (81.54%) and much higher than the nearest neighbour technique.

In Experiment V we verified the proposed algorithm on another database and it can be seen in Table 6. It was verified that COFILS obtained better results than the classical algorithms. An interesting feature is that there was a percentage increase in the COFILS error compared to the other. One might think that this increase of ratings may have facilitated the generalization of supervised learning algorithms.

Overall the proposal got a better result than the nearest neighbour algorithm predicting even more reviews. In grades four and five there was a superior performance of the proposed compared to the classical algorithm. Another improvement was the reduction of extreme predictions, *i.e.*, when the rating is a value and the algorithm provides the worst possible score, for example, the rating is five and it provides one. The only downside was the case of footnote

Table 3. Best results of collaborative filtering techniques with minimum neighbours

Technique	MAE	RMSE	Coverage	Neighbours
COFILS	0.673 (0%)	0.906 (0%)	96.40%	13
CF. Item Based Cosine	0.723 (-7.4%)	0.963 (-6.3%)	96.42%	13
CF. User Based Cosine	0.725 (-7.7%)	0.963 (-6.3%)	96.45%	13
CF. Item Based Pearson	0.713 (-5.9%)	0.954 (-5.3%)	95.01%	13
CF. User Based Pearson	0.715 (-5.0%)	0.955 (-5.4%)	95.05%	13
CF. Item Based Cosine	0.700 (-4.0%)	0.936 (-3.3%)	83.14%	48
CF. User Based Cosine	0.700 (-4.0%)	0.938 (-3.5%)	84.65%	42
CF. Item Based Pearson	0.683 (-1.5%)	0.922 (-1.8%)	**75.05%**	49
CF. User Based Pearson	0.682 (-1.4%)	0.924 (-2.0%)	**76.95%**	46

Table 6. Best results of collaborative filtering techniques in 1M MovieLens dataset

Technique	MAE	RMSE	Coverage
COFILS[1]	**0.646 (0%)**	**0.871 (0%)**	100%
COFILS[2]	0.670 (-3.7%)	0.899 (-3.2%)	100%
CF. User Based Cosine[3]	0.730 (-13.0%)	0.960 (-10.2%)	99.65%
CF. Item Based Cosine[3]	0.734 (-13.6%)	0.963 (-10.6%)	99.53%
CF. User Based Pearson[3]	0.706 (-9.2%)	0.939 (-7.8%)	99.53%
CF. Item Based Pearson[3]	0.708 (-9.6%)	0.944 (-8.4%)	99.52%
Regulared SVD[4]	0.690 (-6.8%)	0.915 (-5.0%)	100%

[1] random forest, SVD, normalization by the average user and eight latent variables;
[2] ann, SVDL, normalization by the average user and three latent variables;
[3] 13 neighbours;
[4] 10 features;

one. The proposed algorithm gave a more extreme rating in this case than when compared with the classic one.

Through these results we can conclude that there is a considerable decrease in both the MAE as RMSE using the proposed algorithms when compared with the traditional collaborative filtering., as this comparison was made in the classic algorithms within the two classes used in collaborative filtering: memory and model-based. The reduction of the MAE/RMSE was because the algorithm could better generalize the problem, as seen in the confusion matrix.

5. Conclusion and Outlook

In this work, we proposed a new way to deal with the sparse matrix problem. A transformation framework was proposed so as to transform the rating matrix used by recommender systems into a classic training dataset appropriate to apply classic supervised machine learning methods.

The proposed framework consists of a preprocessing step in which the rating matrix is decomposed and latent semantic variables are created. These variables are used to build a new feature space to represent the ratings.

Experiments were conducted in order to evaluate whether our proposal achieves results as good as those of the classic collaborative filtering algorithms. Additionally, an analysis was done on how the number of factors and technique to extract latent semantic variables can influence the prediction accuracy returned by an Artificial Neural Network, a Naïve Bayes classifier and Random Forest. So it is possible to treat this problem as a pattern recognition problem.

The results obtained show that our method is comparable to the classic collaborative filtering algorithm in terms of accuracy, having obtained a reduction in extreme prediction, for example, when the rating is five and it predicts one. This problem is crucial in a recommendation system, as it is worse to recommend an item that one does not like than item that is average for it.

As future work we highlight the study of other latent semantic analysis techniques such as Probabilistic Latent Semantic Analysis. We would also like to point the use of other supervised learning methods or the application of bootstrap to streamline the obtaining of better rating predictions.

A challenge is to use this approach in a real system recommendation. All experiments were conducted in offline recommendation systems. Although not enough to have a good hit rate, the algorithm needs to be scalable. A major problem of supervised algorithms is the time spend to train the model. There are several possibilities to solve this problem,such as, for example, using appropriate algorithms that can quickly converge a model with a large amount of data (Huang et al., 2004, 2011), models with incremental training (Joshi, 2012) or techniques for selecting better

data to be used in training (Mello et al., 2010; Sun & Wang, 2010).

The literature has many different bases and each one has different characteristics such as the size of the set of preferences, the rating of distribution, quantity of reviews, the degree of sparsity, and others. Depending on these characteristics the proposed algorithm can get a better or worse outcome. A future work would be to evaluate the proposal in these other databases.

6. Acknowledgments

Our thanks to CNPq/CAPES (Brazilian Council) for funding this research.

References

Adomavicius, G., & 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*, 734–749. doi:10.1109/TKDE.2005.99.

Billsus, D. (1998). Learning collaborative information filters. *International Conference on Machine Learning*, .

Candès, E. J., & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, *9*, 717–772. doi:10.1007/s10208-009-9045-5.

Caruana, R., & Karampatziakis, N. (2008). An empirical evaluation of supervised learning in high dimensions. *on Machine learning*, (pp. 96–103). doi:10.1145/1390156.1390169.

Cunningham, P., & Smyth, B. (2010). An assessment of machine learning techniques for review recommendation. *Artificial Intelligence and Cognitive Science*, (pp. 241–250).

Hsu, S., Wen, M., Lin, H., Lee, C., & Lee, C.-h. (2007). Aimed-a personalized tv recommendation system. *Interactive TV: a Shared Experience*, (pp. 166–174).

Huang, G., Zhu, Q., & Siew, C. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. *Neural Networks, 2004. . . .*, *2*, 985–990. doi:10.1109/IJCNN.2004.1380068.

Huang, G.-B., Wang, D. H., & Lan, Y. (2011). Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, *2*, 107–122. doi:10.1007/s13042-011-0019-y.

Joshi, P. (2012). Incremental Learning: Areas and Methods – A Survey. *International Journal of Data Mining & Knowledge Management Process*, *2*, 43–51. doi:10.5121/ijdkp.2012.2504.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*, 30–37.

Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, *7*, 76–80.

McLaughlin, M. R., & Herlocker, J. L. (2004). A collaborative filtering algorithm and evaluation metric that accurately model the user experience. *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04*, (p. 329). doi:10.1145/1008992.1009050.

Mello, C., Aufaure, M., & Zimbrao, G. (2010). Active learning driven by rating impact analysis. In *Proceedings of the fourth ACM conference on Recommender systems* (pp. 341–344). ACM.

O'Mahony, M., & Smyth, B. (2009). Learning to recommend helpful hotel reviews. In *Proceedings of the third ACM conference on Recommender systems* (pp. 305–308). ACM.

O'Mahony, M., & Smyth, B. (2010). Using readability tests to predict helpful product reviews. In *Adaptivity, Personalization and Fusion of Heterogeneous Information* (pp. 164–167). LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD Cup and Workshop* (pp. 5–8). volume 2007.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Application of dimensionality reduction in recommender system-a case study. *Architecture*, .

Schafer, J., Konstan, J., & Riedi, J. (1999). Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce* (pp. 158–166). ACM.

Schwartz, B. (2005). *The paradox of choice*. New York: ECCO.

Sun, L., & Wang, X. (2010). A survey on active learning strategy. *Machine Learning and Cybernetics (ICMLC) . . .*, (pp. 11–14).

- We propose an transformation from CF problem to typical Supervised Learning

- The proposed transformation is straightforward and domain-independent

- Our transformation greatly outperforms classical Collaborative Filtering techniques