Solving Cold Start WithLebiD2

Lebi Jean Marc Dali and Qin Zhi Guang
Department of Computer Science and Engineering, University of Electronics, Science and Technology of China
Email: lebijean@qq.com

***Abstract*—Cold-Start is one of the most difficult problems faced by web companies today in the domain of recommendation system. Cold-Start problem refers to predicting the behavior of a new user having no history. Common algorithms used to predict user's behavior fail at addressing the cold-start problem because the algorithms are based on the user history with the company. But in this paper, we successfully solve the cold-start problem by applying the model based paradigm to the trust based methodology. In this paper, we explain our method in detail.**

***Index Terms*—cold start, recommenders, model-based RS, trust based algorithm, social network**

I. Introduction

Recommender systems are used to help in decision making. They suggest items a particular user may be interested in or they predict the rating a user will give to a particular item. Recommendation systems are divided into two groups namely content based recommendation system and Collaborative Filtering recommendation system. Content based methods use information pertaining to the user like his interests, occupation, age, etc. and information pertaining to the item like title, topic etc to make predictions. On the other hand, Collaborative Filtering (CF) predicts the rating of the user based solely on the available rating matrix in the database. The latest method known as trust based CF make use of the CF method and of the social network information of the user. We know that today almost everyone is associated to asocial network. Hence by considering the information of the social network, we can successfully predict the user behavior on a particular item. This method is very effective in addressing the cold start problem. Indeed the cold start problem refers to predicting the behavior of a novel user with no history. Here we describe our technique LebiD2which apply the model based methodology into the trust based technique. The result is outstanding. We have a better performance at solving the cold start problem than we did with the former technique LebiD1 [1] which will be briefly described here. LebiD2 will be described extensively in this paper. This paper has 3 sections. In section 2, we discuss related works in this area of study. In section 3, we explain our method "LebiD2" in detail, then in section 4 we evaluate LebiD2 against other well known methods for solving the cold start problem and finally we conclude by showing the advantages of our method.

Manuscript received December 1, 2014; revised June 15, 2015.

II. Related Work

The field of recommendation system has attracted a lot of attention from the research community mainly due to its obvious potential of maximizing the profit of internet companies. One method used most often is memory based [2] like the nearest neighbor technique. This technique analyzes the entire rating matrix to find the close friends of the active user based on the user history and hence predict the behavior based on these data. It is not a good technique for real life interactive applications because it is very slow in processing and can't address the cold start problem since no user information is available then. The next method in recommendation is model based technique [2]. In this method, we first learn a model for the problem and then predict the user behavior using the model computed. We have two types of model based systems: parameterized and non-parameterized model based system. In the parameterized model based system, we first assume a model and according to this model we find the parameters for the model based on the training data. To find the parameters we can used the famous Least Square Method where we use the gradient technique. In this technique, our objective function is the error function in terms of the parameters. We find the gradient of the error function in terms of these parameters by setting the gradient to zero. We solve resulting equation and derive the parameters. In case of a univariate model, the solutions to the parameters will be found directly. However if we are dealing with the multivariate model we may apply a technique like SVD in order to find the final parameters. In case of a discrete model, the model that is considered most often is the Bernoulli distribution or the binomial distribution for a univariate model. And for a multi variate model, we often go for the multinomial model. Where as in the univariate continuous system, we often choose the Gaussian model and with the multi variate system we go often with the multivariate Gaussian model. One major problem with the parameterized technique is that the assumed model may not be the correct one and hence the parameters obtained will not be give a good prediction. For example we may consider a multivariate Gaussian model, where the actual model needed is the Dirichlet model. This is very tricky and we need to be very careful in selecting the model. In the case of the non-parameterized model, we don't assume a model apriori and we allow the model to rise from the training data. One important method used in this category is the Kernel method wherein the dataset is

doi: 10.18178/ijmerr.4.3.255-258

divided into volumes of fixed size. We consider the region of our active data (the user for which prediction is desired). Given for instance two classes where we need to predict in which class the data belongs. We compute two posterior distributions one for each class given our data, the volume of the region and number of elements of each class in that region. We compare both posterior distributions. The one having the largest distribution is considered to be the class of our active user. A word must be said on the Least Square method stated above. Care must be taken when using the Least Square method to find the parameters. Because it often leads to overfitting specially when we have a restricted number of training data. If the number of training data is large, we must not worry about overfitting. Overfitting is not desirable because it hinders generalization. And generalization that is the possibility to use the model to predict for new data is the main purpose of recommendation. In case we have a limited training data, we can use regularizer to limit overfitting. Indeed the regularizers will shrink the magnitude of the parameters. Hence overfitting will avoided. Also here a balance is needed because a too large value for the regularizer will generate a over-smoothed model which is equally as bad as the overfitting problem. The Least Square technique is equivalent to the maximum likelihood technique. Here also care must be taken to avoid overfitting in case we have a limited training data. The technique used to suppress the overfitting problem in case of maximum likelihood is the addition of a prior. Indeed the addition of a prior distribution has the ability to suppress the overfitting. The resulting posterior distribution will have the same model has the prior model. This is now a full Bayesian treatment of the problem. The Least Square method in case of the multivariate problem makes use of matrix factorization which is a very expensive process and is no guarantee of success due to the overfitting problem stated earlier. The latest technique in this area is the trust based system which combines the rating matrix and the social network database for better prediction. One such method is the TidalTrust recommendation system [3], which performs a modified breadth-first search in the system. It computes the trust value based on all the raters at the shortest distance from the target user. The trust between users $\mathbf{u}$ and $\mathbf{v}$ is given by:

$$t_{\mathbf{u},\mathbf{v}} = \frac{\sum_{\mathbf{w}\in\mathbf{N}} (t_{\mathbf{u},\mathbf{w}} * t_{\mathbf{w},\mathbf{v}})}{\sum_{\mathbf{w}\in\mathbf{N}} t_{\mathbf{u},\mathbf{w}}} \quad (1)$$

where $\mathbf{N}$ denotes the set of the neighbors of $\mathbf{u}$.

And the trust depends on all the connecting paths.

The predicted rating is computed as:

$$r_{\mathbf{u},\mathbf{i}} = \frac{\sum_{\mathbf{v}\in raters} (t_{\mathbf{u},\mathbf{v}} * r_{\mathbf{v},\mathbf{i}})}{\sum_{\mathbf{v}\in raters} t_{\mathbf{u},\mathbf{v}}} \quad (2)$$

where $r_{\mathbf{v},\mathbf{i}}$ denotes rating of user $\mathbf{v}$ for item $\mathbf{i}$.

This technique is very efficient in addressing the cold start. Our technique will be evaluated against this technique.

We also have the MoleTrust technique [4]. It is very similar in its operation to the previous technique (TidalTrust). The only difference is that instead of focusing on users at the shortest distance, it rather consider raters up to a maximum-depth d. Tuning appropriately d gives a very good result. Hence the MoleTrust is better than TidalTrust. We will test our technique (lebiD1) against this method as well.

The last but not the least technique to evaluate is the TrustWalker method [5]. Also it is similar in its proceedings to the MoleTrust but instead of the far friends who have rated the target item, we use the near neighbors who have rated similar items. The question here is how do we define similarity between two items. The similarity is given by:

$$sim(\mathbf{i},\mathbf{j}) = \frac{corr(\mathbf{i},\mathbf{j})}{1+e^{-\frac{|corr(\mathbf{i},\mathbf{j})|}{2}}} \quad (3)$$

This technique will also be matched against our method.

And finally we have LebiD1 which is also a trust based technique. It combines the memory based technique with the social network information to solve the cold start problem.

When applied on the QQ social network and the movielens database [6], the result outclass the previous methods as will be seen later. The formula used in LebiD1 is:

$$r_{\mathbf{a},\mathbf{i}} = \frac{\sum_{\mathbf{u}\in\mathbf{F}} r_{\mathbf{u},\mathbf{i}}}{card(\mathbf{F})} \; \mathbf{i}\in\mathbf{I} \quad (4)$$

where $\mathbf{F}$ represents the closest friends of active new user $\mathbf{a}$, $\mathbf{I}$ is the set of all the items rated by all the users $\mathbf{u}$ of $\mathbf{F}$.

where *r***a**,**i** denotes the predicted rating of novel user **a** to item **i**, **F** denotes the set of social network friend shaving rated item **i**, *r***u**,**i** denotes the rating of friend user **u** to item **i** and finally *card* (**F**) denotes the number of such friends.

III. LebiD2

Cold start has been a problem that has troubled researchers for years. Our method solves the cold start problem effectively. Already in LebiD1 [1], we solved the cold start problem using a memory-trust based methods with QQ social network and the movielens database [6]. The same databases are used in LebiD2 but instead of using the memory-trust based LebiD1 [1], we instead use the model-trust based method LebiD2. Using this technique as we will see in the next section, we have a better performance in solving the cold start problem. In LebiD1, we have a movie database like Movielens [6] and we want to predict the movies that a new user will be interested in. So here we have two problems: first that of

predicting the movies the new user will like and the second providing a rating for these movies such that we can rank them with the first movie having the highest rank and the last movie the lowest. Each position determines the level of importance. In LebiD1, we consider a second database called social database. For LebiD2 the same premises above will be kept but we will use the model based framework to solve the cold start problem. The model formula for LebiD2 is:

$$\mathbf{W} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T.t \quad (5)$$

where $\mathbf{W}$ represents the model needed and it is solved using LebiD2 decomposition. $\mathbf{X}$ represents the user dataset and $\mathbf{t}$ represents the movie database.

LebiD2 is an SVD like decomposition except it is more stable and less time consuming. The algorithm is as follows:

N: represents the number of friends on the social network having rated item i

-For k=1 to n

-For i=k to m

$$S(k) = \sqrt{S(k)^2 + A(i,k)}$$

$$A(i,k) = A(i,k)/S(k)$$

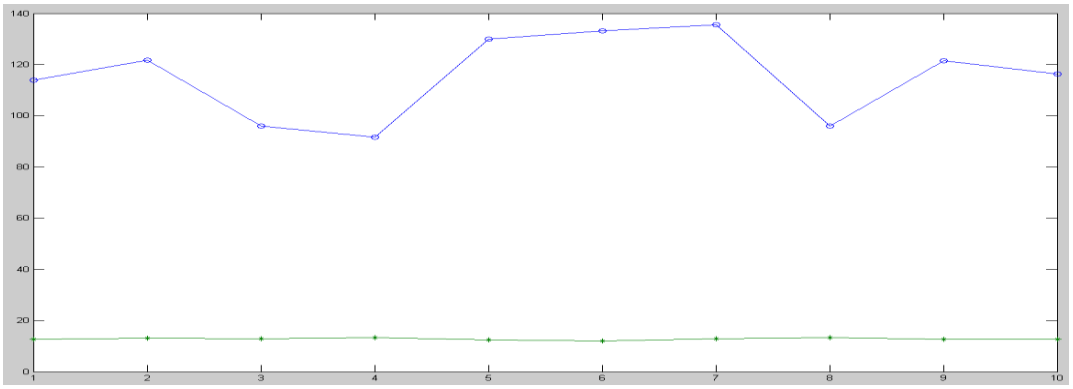
-For j=k+1 to n

-For i= k to m

$$A(i,j) = (A(i,j) + A(i,k)^2)/A(k,k))$$

$$S(i) = A(i,i)$$

Comparing our LebiD2 with the matlab SVD we obtain:



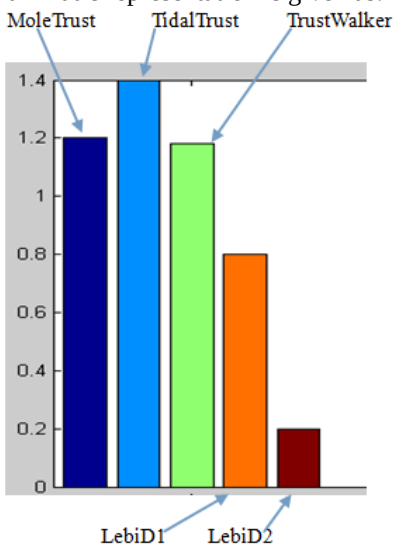
LebiD2 is in green. We can see that our method LebiD2 is very stable compared to the matlab SVD which means it is less time consuming, which makes it ideal for real time application. Indeed we could have used the standard SVD for solving the model but using the standard SVD will be impractical due to the time involved in computing. Hence our technique even though less effective as the matlab SVD is very good for the application of the cold start problem. The reason is that here the only thing we need is only an approximation of the singular results. But the results must be generated as soon as possible. So we can be lenient on the effectiveness of the singular values however we cannot compromise on the speed since users don't normally like to waste their time on a web page. So speedy result means we are more likely to keep our novel user instead of great performance at a longer time which we can't afford in our framework. Also as we will see in the next section, the results are spectacular in addressing the cold start problem.

IV. Experimental Evaluation

Our method (LebiD2) has been evaluated against well known techniques like MoleTrust, TidalTrust and TrustWalker and our former technique LebiD1. And the RMSE [2] for Cold Start users is given as per the following table:

Method	RMSE results
MoleTrust	1.400
TidalTrust	1.200
TrustWAlker	1.180
LebiD1	0.800
LebiD2	0.200

The diagrammatic representation is given as:



We can clearly see that our model-trust based approach LebiD2 (the last bar in the figure) has a better error rate than any of the previous methods.

RMSE (Root Mean Squared Error)

The RMSE is a rating metric that is used to test the accuracy of the recommender technique. Its formula is given by:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{\{i,j\}}(p_{i,j} - r_{i,j})^2} \quad (6)$$

n is the total number of ratings over all users, $p_{i,j}$ is the predicted rating for user Ion item j, $r_{i,j}$ is the actual rating.

RMSE amplifies the contributions of the absolute errors between the predictions and the true values.

V. Conclusion

LebiD2 is definitely a major breakthrough in dealing with the Cold Start problem. The combination of model based approach and trust based technique has proven to beindeed very effective in dealing with the cold-start problem. Also the accuracy is increased as can be seen with the RMSE test. However LebiD2 is time consuming since it is based on the model based architecture. We encourage researchers to focus on the relationship between the user and Social network [7] in order to design better recommender that solve efficiently the cold start (0% error rate) in an acceptable time delay. Indeed Social Network is the next big thing in the world of recommenders. Hence we encourage researchers to focus more on the social network relation of the users.

Acknowledgement

This research was supported by the NSFC under grant No. 61133016, 61202445, 61103206.

References

[1] L. J.M. Dali and Q. Zhi Guang, "Cold start mastered: LebiD1," in *Proc. International Conferences on Computational Science and Engineering*, Chendu China, Dec. 2014

[2] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, January 2009.

[3] J. Golbeck, "Computing and applying trust in web-based social networks," PhD thesis, University of Maryland College Park, 2005.

[4] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. Acm Conference on Recommender Systems*, 2007.

[5] M. Jamal and M. Ester, "TrustWalker: A random walk model for combining trust-based and item-based recommendation," *KDD* 2009.

[6] MovieLens Data. [Online]. Available: http://www.grouplens.org/

[7] A. Sharma and D. Cosley, "Do social explanations work? Studying and modeling the effects of social explanations in recommender systems," in *Proc. International Conference on World Wide Web*, 2013.

Lebi Jean-Marc Dali was born in Abidjan, Ivory Coast, in 1987. He received the B.C.A. (Bachelor Computer Application) degree from Bangalore University (India) in 2010 and the M.C.A (Master Computer Application) degree from the same University, in 2013. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, UESTC (University of Electronic, Science and Technology of China). His research interests include machine learning, recommendation systems and information security.

Prof Qin Zhi Guang received his PhD from UESTC (University of Electronic, Science and Technology of China) in 1996. He is currently the Dean of the Computer Science Department. His research interests include machine learning, recommendation system and information security.