Predicting your Soulmate

Devishi Kesar[1] and Akarsha Sehwag[1]

Indraprastha Institute of Information Technology, Okhla Phase 3, Delhi, India

Abstract. In modern day's world the World has shifted to finding partners online because it is easy and more informational. The eHarmony[4] dataset used in the paper is one such site which matches you to your partner based on the a structure based SVM model using the answers to various questions asked. We use the DTI-Pred and PUCPI algorithms[2] on the dataset to show how they perform better than standard algorithms using AUC scores.

Keywords: Collaborative Filtering · eHarmony · dating.

1 Introduction

In today's world where people find it socially awkward and resist moving out of their homes, when playing on mobile devices is more popular than outdoor games, finding a partner should be available on these devices too. Utilizing this phenomenon, companies like tinder and dating websites like eHarmony have profited. Yet the methods they use to help predict perfect match for a person based on their answers may not be up to date. We have, in this paper applied novel techniques to help predict the best partner for a user. We used SVD(singular value decomposition), Non-Negative matrix factorization and prediction assuming normal distribution of training data set.

1.1 Dataset

Collaborative filtering approaches are usually designed to work on very large datasets. Therefore, it is crucial that the proposed approach is scalable. To find out the best scalable approach and give a more real-life scenario with millions of users, weve used a huge dataset open-sourced by eHarmony, an online dating platform. The dataset provides user to user matching data to show whether two individuals; given by their respective user IDs matched or not which basically shows their mutual interest. This has been indicated by positive and negative examples in the dataset wherein positive examples have been depicted by one and negative examples as zero. Two users have been considered to be relevant or similar if they match, otherwise, they have been regarded as irrelevant. The users have been kept anonymous to ensure their privacy and therefore numbers have been used instead. Similarly, to describe an individual, his personal choices or profile details have been written in numerical form. The mapping has not been explained by the authors. Every user has been represented as a vector with

59 columns, one of them is their UserID and the other 58 columns have been used to describe their profile including their interests and their personality.

The dataset has been divided according to the matching time of the two users. Training data comprises of the matches of the users in the first half of the time interval and testing data contains the matches in the other half. The training dataset consists of approximately 2,75,000 unique users and approximately half a million matches from the eHarmony app. Such a huge dataset is difficult not just to process but might also contain some users which are not contributing to the discriminative data. Therefore, we sample the data, to get the top 10000 users and their corresponding edges(rating for other users).

2 Algorithms

2.1 SVD

The famous SVD algorithm, as popularized by Simon Funk during the Netflix Prize. When baselines are not used, this is equivalent to Probabilistic Matrix Factorization[5].
The prediction $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^t p_u$$

If user u is unknown, then the bias b_u and the factors p_u are assumed to be zero. The same applies for item i with b_i and q_i.
To estimate all the unknown, we minimize the following regularized squared error:

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2)$$

These steps are performed over all the ratings of the train set and repeated 20 times. Baselines are initialized to 0. User and item factors are randomly initialized according to a normal distribution with γ as 0.005 and λ as 0.02.

2.2 Non-Negative Matrix Factorization

This algorithm[3] is very similar to SVD. The prediction $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = q_i^T p_u$$

where user and item factors are kept positive.
The optimization procedure is a (regularized) stochastic gradient descent with a specific choice of step size that ensures non-negativity of factors, provided that their initial values are also positive.

2.3 Normal Predictor

Algorithm predicting a random rating based on the distribution of the training set, which is assumed to be normal.
The prediction $\hat{r}_{ui}$ is generated from a normal distribution $N(\hat{\mu}, \hat{\sigma}^2)$ where $\hat{\mu}$ and $\hat{\sigma}$ are estimated from the training data using Maximum Likelihood Estimation:

$$\hat{\mu} = \frac{1}{|R_{train}|}$$

$$\sum_{r_{ui} \in R_{train}} r_{ui}$$

2.4 DTIPred

The method as implemented by [1] is based on using random forest to predict drug-target interaction. We thus applied random forest as suggested by the paper over our training data reduced to 30000 samples to allow fast computation over the data set.

2.5 PUCPI

PUCPI or the first compound-protein interaction (CPI) prediction method. It selects the best parameters for c and j followed by the k-fold cross validation on the data set. We reduce the data size to 30000 samples to allow fast computation. The authors have trained their prediction algorithm using libSVM or biased-SVM (Support Vector Machines) for classification. Computation of libSVM-classifier amounts to minimizing the given expression:

$$[\frac{1}{n} \sum max(0, 1 - y_i(w.x_i - b))] + \lambda \left\| \omega^2 \right\|$$

Smaller value of lambda would yield the hard-margin classification for data that can be linearly classified.

3 Results

We observe that both the algorithms, DTI-Pred and PUCPI, are able to perform better with an AUC of 0.5. We improved the AUC by 0.17 as compared to the baseline paper. We also tried to find the RMSE and MAE for some baseline algorithms wherein we got the best RMSE scores best for DTI-Pred and the best MAE for Singular Value Decomposition. Overall, DTI-Pred performed best.

Table 1. Table 1: The table compares the baseline algorithms implemented by us and the novel algorithms suggested.

Baseline	**RMSE**	**MAE**
SVD	0.213	**0.336**
Non-Negative Matrix Factorization	0.179	0.339
Normal Predictor	0.273	0.407
DTI-Pred	**0.121**	0.348
PUCPI	0.273	0.407

Table 2. Table 2: The table compares the algorithms implemented in the initial paper [1] and those implemented by us.

Algorithm	**AUC**
SVM(original paper)	0.336
DTI-Pred	**0.500**
PUCPI	**0.500**

4 Conclusion

We have presented here few novel machine learning algorithms which were written initially for DTI(Drug-Target interactions) data sets. Due to the large expanse of the data set, the data set was pruned and then algorithms were applied. The results have shown how the version of random forest implemented by DTIpred and SVM implemented by PUCPI outperforms the implementation of the paper[1]. Our code is available on https://github.com/devishi/CF_Project.git .

References

1. Coelho, E.D., Arrais, J.P., Oliveira, J.L.: Computational discovery of putative leads for drug repositioning through drug-target interaction prediction. PLoS Computational Biology **12**(11) (2016). https://doi.org/10.1371/journal.pcbi.1005219, https://doi.org/10.1371/journal.pcbi.1005219
2. Ezzat, A., Wu, M., , Li, X.L., Kwoh, C.K.: Computational prediction of drug-target interactions using chemogenomic approaches: An empirical survey. Briefings in Bioinformatics (2018)
3. Luo, X., Zhou, M., Xia, Y., Zhu, Q.: An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Trans. Industrial Informatics **10**(2), 1273–1284 (2014). https://doi.org/10.1109/TII.2014.2308433, https://doi.org/10.1109/TII.2014.2308433
4. McFee, B., Lanckriet, G.R.G.: Metric learning to rank. In: Fürnkranz, J., Joachims, T. (eds.) Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel. pp. 775–782. Omnipress (2010), http://www.icml2010.org/papers/504.pdf
5. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: Cohen, W.W., McCallum, A., Roweis, S.T. (eds.) Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008. ACM International Conference Proceeding Series, vol. 307, pp. 880–887. ACM (2008). https://doi.org/10.1145/1390156.1390267, https://doi.org/10.1145/1390156.1390267