User Relation Prediction Based on Matrix Factorization and Hybrid Particle Swarm Optimization

Zhenkun Shi
College of Computer Science and Technology, Jilin University, Changchun, 130012, China
shizk14@mails.jlu.edu.cn

Wanli Zuo*
College of Computer Science and Technology, Jilin University, Changchun, 130012, China
wanli@jlu.edu.cn

Weitong Chen
School of Information Technology and Electrical Engineering, the University of Queensland, Brisbane Australia
w.chen9@uq.edu.au

Lin Yue
School of Computer Science and Information Technology, Northeast Normal University, Changchun, 130117, China
yuel031@nenu.edu.cn

Jiayu Han
College of Computer Science and Technology, Jilin University, Changchun, 130012, China
jyhan15@mails.jlu.edu.cn

Lizhou Feng
Tianjin University of Finance and Economics, Tianjin, 300222, China
flzvg@126.com

ABSTRACT
Many real-world domains are relational in nature, consisting of a set of objects related to each other in complex ways. Matrix factorization is an effective method in relationship prediction, However, traditional matrix factorization link prediction methods can only be used for non-negative matrix. In this paper, a generalized framework, itelliPrediction, is presented that is able to deal with positive and negative matrix. The novel itelliPrediction framework is domain independent and with high precision. We validate our approach using two different data sources, an open data sets and a real-word dataset, the result demonstrated that the quality of our approach is comparable to, if not better than, exiting state of the art relation predication framework.

Categories and Subject Descriptors
E.1. [**DATA STRUCTURES**]: Graphs and networks

General Terms
Algorithms, Design, Experimentation

Keywords
Social Networks; Machine Learning; Relation Prediction; Matrix Factorization

1. INTRODUCTION
In nature, networks rarely appear isolate, and objects are interpedently connected. Users are virtually connected through a network of different types of linkages in social media. Thus, there has been an increasing interest in predicting multi-relation data that contain multiple entity type and serval relations. As an example, relation network in social media, such as Weibo, where user relation can be defined as relation network. A user relation network among social media is a kind of social network, use this network online users can get reliable information for e-commerce, trust assessment and garbage behavior detection.

When modelling relationship between pairs of individuals, positive relationships are representative of liking, loving, valuing or approving someone, and negative relationships are representative of disvaluing, disapproving or negatively valuing. For a given directed link from user i to j in a social network, we define its sign to be trust (or distrust) if it expresses a positive (or negative) attitude from user i to j. We call such network with both trust and distrust links relation network.

Perhaps the most basic yet significant belief in relation network is structural balance theory [27] and low-rank matrix factorization model [29]. Structural balance theory states that users in relation network tend to follow patterns such as “an enemy of my friend is my enemy” and “an enemy of my enemy is my friend” and so on [28]. Low-rank matrix factorization model make relation predication in an unsupervised scenario by seeking low-rank representations for users. Structural balance theory aims to infer the unknown relationship between two entities by learning from balance information of relation network. Hence, it is not applicable to the relation network, which contains many isolated islands. However, the iterative method for low-rank matrix decomposition can only decompose non-negative matrices, so we can only predict the trust relation in the relation network. Therefore, these methods have inherent limitations. In our work, we aim to develop a practical framework for the problem of relation prediction among online users in the very near future by using a new framework itelliPrediction. ItelliPrediction extended the none-negative matrix factorization by introducing biological intelligence algorithm, hybrid particle swarm optimization.

In this paper, we did some work in trust link prediction, we predict the type of links among users in social media sites. Fig. 1 shows an example to illustrate the intuition behind this idea. Fig. 1 is the user relation network of wikiElec, the network contains trust relations (mark as +), unknown or missing relations (mark as 0) distrust relations (mark as -). We predict if there is a relationship between node i and node j, and if so, what the relationship is. Our main contributions are summarized as follows:

* Corresponding author

WWW 2017 Companion, April 3-7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3051151

1. We study the effectiveness of none-negative matrix factorization method for the user relationship prediction problem, inspired by their success in trust prediction [9,18] and social recommendation [29].
2. We provide an approach to exploit matrix factorization via biological intelligence.
3. We propose an unsupervised framework, itelliPrediction, for the problem of user relation prediction, incorporating matrix factorization with hybrid particle swarm optimization.
4. We evaluate itelliPrediction extensively using two different data sources, an open data sets and a real-word dataset.

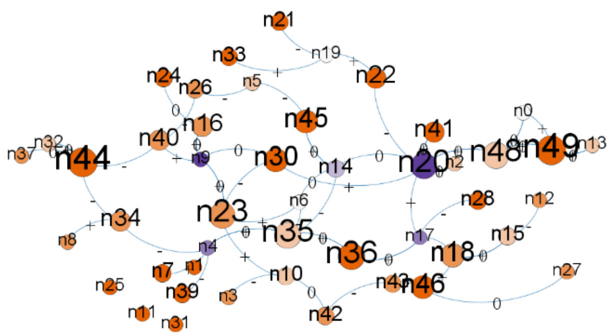


Figure 1. User Relation network of wikiElec.

2. RELATED WORK

Recently, for mining a relation network, researchers have made a variety of studies including but not limited to trust-aware recommendation systems [5, 11, 15, 16], trust diffusion analysis [9, 17] and trust-link prediction [18]. At a high level, existing link prediction models fall into two classes: unsupervised and supervised [25]. Unsupervised models compute scores for pairs of nodes based on topological properties of the graph. Supervised models, on the other hand, attempt to be directly predictive of link behavior by lining a parameter vector θ via

$$\min_{\theta} \frac{1}{|\mathcal{O}|} \sum_{(i,j)\in\mathcal{O}} \ell(G_{ij}, \hat{G}_{ij}(\theta)) + \Omega(\theta) \qquad (1)$$

where $\hat{G}_{ij}(\theta)$ is the model's predicted score for the dyad $\ell(i,j)$, $\ell(\cdot,\cdot)$ is a loss function, and $\Omega(\cdot)$ is a regularization term that prevents overfitting. The choice of these terms depends on the type of model. We list some popular approaches:

Path Probability and Spring Embedding (PPSE) [23]: A method for computing trust and distrust, which is provided by Thomas DuBois, et al. He does that by combining an inference algorithm that relies on a probabilistic interpretation of trust based on random graphs with a modified spring-embedding algorithm.

Transfer Learning (TL) [24]: An algorithm that adapts the transfer learning approach to leverage the edge sign information from the source network. Because the network may have a different related joint distribution of edge instances and their class labels, there is no predefined feature vector for edge instances in a signed network. Ye Jihang, et. al adopt an AdaBoost-like transfer learning algorithm adjoin with instance weighting to utilize more useful training instances in the source network for prediction.

Homophily Effect [9]: Presented by Tang Jiliang, et al, homophily is a social theory, and the effect suggests that users with similar life habit are likely to set up trust relations, while users with trust relations are more similar in behaviors. Matsutani, et al [18] extended the method by adding user activity information for trust-link prediction and applied it to analyze user behavior in an item-review site.

Matrix Factorization for Link Sign Prediction (MF-LiSP) [19]: In view of the global behavior of the different users' needs to be accounted for, albeit the local interactions do play a significant role too [26]. Kuter, et al presents a new method called MF-LiSP, which employs a trace norm regularizer with a particularly suited variation of the pair-wise hinge loose to approximate the given matrix. His experiments show that the method is right and advanced in binary classification, but this method does not apply to multiple classification.

Nevertheless, most of them focus on trust relations and applied it into various applications. However, in reality we should think more, in international relations exist hostile, neutral and alliance, in the field of biology, there exists a promotion and inhibition relationship between neurons, in online voting sites, there is an approval and disapproval standpoint among users. Considering these situations, we should distinguish users what relationship they have established.

3. PRELIMINARIES

3.1 Data Analysis

For this study, we collect two datasets, i.e., wikiElec[1] and Slashdot[2]. For a Wikipedia editor to become an administrator, a request for amidships must be submitted, either by the candidate or by another community member. Subsequently, any Wikipedia member may cast a supporting, neutral, or opposing vote; snap.stanford.edu crawled and parsed all votes as wikiElec. Slashdot is a technology-related news website and in 2002, Slashdot introduced the Slashdot Zoo feature, which allows users to tag each other as friends or foes.

Table 1. Statistics of the Datasets

	WikiElec	Slashdot
# of Users	456	3367
# of Out-trust Relations	26313	126224
# of Neutrality Relations	1858	-
# of Out Distrust Relations	8581	74748
# of In-trust Relations	27952	100241
# of In-distrust Relations	7910	40877
Max # of Out-trust Relations	827	387
Max # of Out-distrust Relations	363	400
Max# of In-trust Relations	417	1389
Max# of In-distrust Relations	173	498
Relation Network Density	0.078303	0.005165

We rank all pairs of users with in-trust (out-trust) relations and in-distrust (out-trust) relations in descending order and we pick up

1 The wikiElec dataset is available from http://snap.stanford.edu/data/wiki-Elec.html

2 The Slashdot dataset is available from http://snap.stanford.edu/data/soc-Slashdot0811.html

users with both in-links and out-links. Next, we filter users with less than two out-link relations and two in-link relations, respectively, so that we can get sufficient historical information for the purpose of evaluation. We found out that, on average users in wikiElec have 20.25 out-trust relations, 7.18 out-distrust relations, 20.49 in-trust relations and 7.00 in-distrust relations. Users in Slashdot have 11.10 out-trust relations, 8.26 out-distrust relations, 10.76 in-trust relations, and 7.59 in-distrust relations.

The distribution of in-trust, in-distrust, out-trust and out-distrust for each user are shown in Fig. 2. We can see that most users have few relations, while few users have high number of in or out relations. It indicates the typical online social networks follow the power-law distribution.

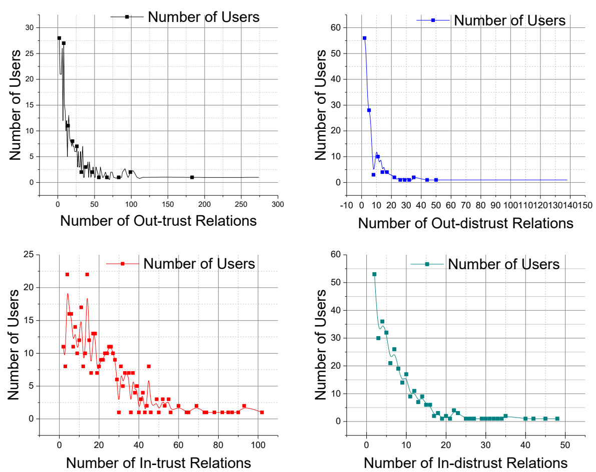


Figure 2. Out-link and In-link Distributions in WikiElec

3.2 Use Low-rank Matrix Factorization Method to Predict User Relation

Low-rank matrix factorization can be used in many occasions like collaborative prediction [1], CT reconstruction [2], deep network training [3], collective filtering [4-5] and document clustering [6-7]. User relation prediction refers to the task of predicting relations in the coming future of every pairs of users. Let $U = \{u_1, u_2, \cdots u_n\}$ be the set of users and n represent the number of users. Let $G \in \mathbb{R}^{n \times n}$ represent the users' relation matrix, matrix element set is as follows:

$$G(i,j) = \begin{cases} 1, & \text{if } u_i \text{ trust } u_j \\ -1, & \text{if } u_i \text{ distrust } u_j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

We note that among all users in the dataset few users can establish relations, result in G very spare and low rank. The low-rank matrix factorization model try to seeks a new representation of matrix G, $M \in \mathbb{R}^{n \times d}$ $(d << n)$ by solving the following problem shows in Eq. (3),

$$\min_{U,V} \left\| G - MNM^{\mathrm{T}} \right\|_F^2 \quad (3)$$

$\|\bullet\|_F^n$ is the n norm of a matrix and $N \in \mathbb{R}^{d \times d}$ captures the correlations among their low-rank representation such as $G(i,j) = M(i,:)NM^{\mathrm{T}}(j,:)$. In order to prevent over fitting we add 2 smoothing factor α and β to M and N, respectively. Then we get a new function,

$$\min_{U,V} \left\| G - MNM^{\mathrm{T}} \right\|_F^2 + \alpha \|M\|_F^2 + \beta \|N\|_F^2 \quad (\alpha > 0, \beta > 0) \quad (4)$$

According to [8], can be explained as an adjoin matrix, which indicates user relation structures.

The matrix factorization method has many advantages [9]: (1) it is very flexible and allow us to include prior knowledge and we will apply it into our future work; (2) it can be applied to find a well-worked optimal solution, among the vast amounts of user relationships; (3) it has a nice probabilistic interpretation with Gaussian noise.

4. FUNDAMENTAL ALGORITHM

There are several matrix factorization methods such as the gradient descent method [9], multiplicative update algorithms and alternating least squares algorithm [10], but these algorithms will convergence only when the matrix is nonnegative. In consideration of our relation matrix containing both positive and negative elements, the methods mentioned above are not directly applicable any more. In this section, we provide a basic matrix factorization scheme, coupled with hybrid particle swarm optimization, that encompasses relation prediction.

4.1 Description of Hybrid Particle Swarm Optimization (HPSO)

Particle swarm optimization (PSO) is a population based algorithm, which exploits a set of potential solutions to the optimization problem, which is developed by Dr. Eberhart and Kennedy in 1995, inspired by social behavior of birds flocking or fish schooling. PSO can deal with non-linear optimization problems in non-convex domains [20]. We called each potential solution as a "particle", and the set of potential solutions in each iteration step forms the swarm [21].

HPSO updated the way from the traditional PSO algorithm by tracking extremum to update the particle's location. HPSO introduced crossover and mutation into PSO. Search for the optimal solution by apply crossover and mutation over the particle extremes and the colony extremes.

4.2 Phases of Hybrid Particle Swarm Optimization Algorithm

In general, the hybrid particle swarm optimization algorithm has the following steps [22]:

Initialization: an initial population is randomly generated, which consists of particles, which represent possible solutions of the problem.

Update particles: update the velocity and position of each particle use crossover and mutation. If their previous velocities are very close to zero, then all the particles will stop moving once they catch up with the global best particle

End condition: the HPSO algorithm will stop when the optimal solution reached or there are little changes of the optimal solution.

4.3 The proposed method

In order to combine the matrix factorization problem with the HPSO problem, we have to process the elements in the matrix factorization problem and make some assumptions. First, we assume that every element of the decomposition matrix represents the gene information of the individual and the decomposition matrix itself is an individual. Next, we assume that a complete iteration of the relationship matrix is the process of the evolution of the individual. Above all, we can preliminarily combine the matrix factorization problem with the HPSO problem.

With the definition of low-rank matrix factorization link prediction is to solve the optimization problem proposed in Eq. (4), and if we remove the constants in the objective function, then we can get Eq. (5),

$$F = Tr(-2G^{\mathrm{T}}MNM^{\mathrm{T}} + MN^{\mathrm{T}}M^{\mathrm{T}}MNM^{\mathrm{T}}) + \alpha Tr(MM^{\mathrm{T}}) + \beta(NN^{\mathrm{T}}) \quad (\alpha > 0, \beta > 0) \tag{5}$$

The coupling between ***M*** and ***N*** makes the problem presented in Eq. (4) difficult to find optimal solutions simultaneously. In this work, we introduced the HPSO for Eq. (5).

First, we initialize the population randomly, then we calculate the fitness of each particle, finally we update the particle by using the framework, itelliPrediction. The itelliPrediction is based on HPSO and matrix factorization. The algorithm will stop when there is little change for the best value or the algorithm reached its predefined. The detailed algorithm of proposed framework, itelliPrediction, is shown in Algorithm 1.

Algorithm 1. The framework of link prediction

Input: ***G***, ***U***, V, *NIDN*, *Pc*, *Pm*, *MAXGEN*, α, β
Output: ***G'***

Initial the population per *NIDN*
Initialize chromosomes
Replace one of the chromosomes randomly with *M* and *N*
while Not reach the *MAXGEN* and not meet the end condition **do**
    Fitness ()
    Record the best chromosome per individual fitness, remember as $\boldsymbol{B}_{individual}$
    Delete the worst chromosome from the population.
    Select ()
    Recombine ()
    ParticleMutate ()
    Insert the $\boldsymbol{B}_{individual}$ into the population
end while
Set $\boldsymbol{G'} = \boldsymbol{MNM}^{\mathrm{T}}$
Return *G'*

Where ***G*** is users' relations matrix. ***M*** and ***N*** are the relative minimum. *NIDN*, *Pc*, *Pm* and *MAXGEN* represent the population size, the probability of recombination, the probability of mutation. Where α and β are introduced to control the capability on ***M*** and ***N***, respectively.

For the function ***Fitness*** (), we first calculate the fitness value of each chromosome C_k, which is represented by ***eval(C)***:

$$eval(C_k) = f(x^k) = \|G - M_k N_k M_k^{\mathrm{T}}\|_F^2 + \alpha\|M_k\|_F^2 + \beta\|N_k\|_F^2 \tag{6}$$

Then we calculate the fitness value of the whole population:

$$F = \sum_{k=1}^{NIDN} \frac{\sum_{k=1}^{NIDN} eval(C_k)}{eval(C_k)} \tag{7}$$

Next we calculate the corresponding selection probability of each chromosome $\boldsymbol{C_k}$:

$$P_k = \frac{\sum_{k=1}^{NIDN} eval(C_k)}{F \bullet eval(C_k)} \tag{8}$$

Finally, we calculate the cumulative probability of each chromosome $\boldsymbol{Q_k}$:

$$Q_k = \sum_{j=1}^{k} P_j \quad j = 1, 2, \cdots \tag{9}$$

For the function ***Select* ()**, on the basis of fitness, calculated above, we adopt a roulette method based on linear ranking selection. For the function ***Recombine* ()**, we bring single-point of intersection to restructure the chromosomes, the probability of recombination Pc is given by users manually. For the function ***ParticleMutate* (),** we adopt the commonly used bit mutation operator.

The optimal ***M*** and ***N***, is a representation of ***G***. As ***M*** and ***N*** contain both positive and negative value, the new low-rank matrix ***G'*** is a representation of a singed network which indicates users' relationship. The relationship of u_i and u_j are indicated by $G'(i,j)^+$ and $G'(i,j)^-$, respectively.

5. EXPERIMENT

We conduct experiments to evaluate the proposed framework. First we give our experiment setting and evaluation metric, then we briefly discuss the baseline algorithm against which we intended to compare the different types of relation prediction methods, finally we conduct the experiment on different datasets to show the adaptability and flexibility of the proposed framework.

5.1 Experiment Settings

The experiment setting of the dataset is shown in Fig. 3, where $A = \{\langle u_i, u_j\rangle | G(i,j) = \pm 1/0\}$ and $B = \{\langle u_i, u_j\rangle | G(i,j) = -9\}$ are the user relation set. ***A*** represents users with relations while *B* indicate users without relations. Conforming the time when users build up their relations, we arranged the pairs both in ***A*** and ***B*** in a descending order. We divided ***A*** into two parts, the first *x*% as training set ***E*** to train data and the remaining 1-*x*% as testing set ***O*** for prediction. We set $G(i,j) = 0, \forall\langle u_i, u_i\rangle \in O$ in ***O*** to remove the user relation type, and ***G'*** is the new representation of ***G***, then ***G'*** will input into each predictor. We varied $\boldsymbol{x}$ as $\{50, 60, 70, 80, 90\}$.

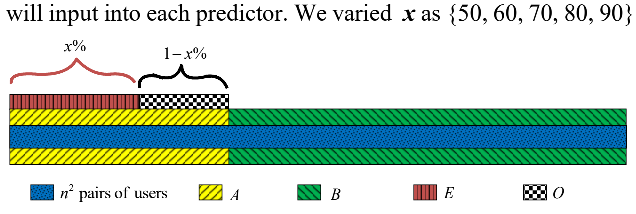


Figure 3. Separation of the dataset. A is the training set of pairs with relations, and B is the testing set without relations.

We rank user couples in $B \cup O$ in decreasing order for each predictor, and pick up the first $|Q_h|$ and the last $|Q_l|$, as the testing set for user relation prediction, denoted as C_h and C_l respectively. Then the prediction accuracy (PA) can be defined as,

$$PA_h = \frac{|Q_h \cap C_h|}{|Q_h|} \qquad\qquad PA_l = \frac{|Q_l \cap C_l|}{|Q_l|} \tag{10}$$

Where $|\cdot|$ represent the size of a set.

5.2 Baseline Methods

Next, we compare our method against various baseline algorithm as follows:

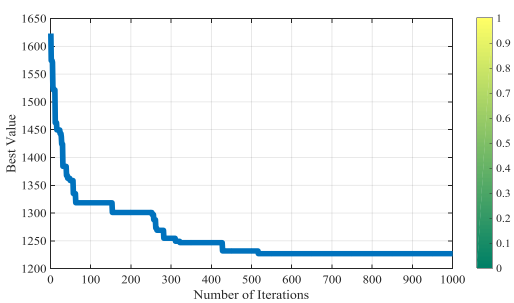
Weighted Random(WR): Note that the distribution of trust and distrust are very spare, we also predicted the type of users' relation as trust or distrust in proportion to the respective share, and we refer to this as the weighted method.

Path Probability and Spring Embedding (PPSE) [23]: A method for computing trust and distrust, which is provided by Thomas DuBois, et al. He does that by combining an inference algorithm that relies on a probabilistic interpretation of trust based on random graphs with a modified spring-embedding algorithm.

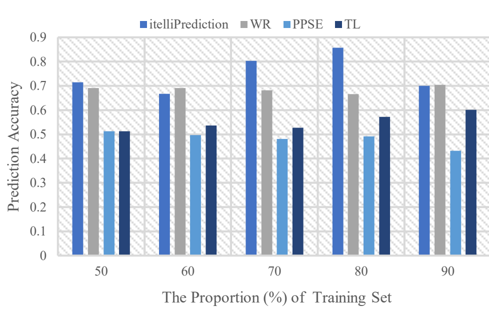
Transfer Learning (TL) [24]: An algorithm that adapt the transfer learning approach to leverage the edge sign information from the source network. Because the network may have a different related joint distribution of edge instances and their class labels, what is more there is no predefined feature vector for edge instances in a signed network. Ye Jihang, et. al adopt an AdaBoost-like transfer learning algorithm adjoin with instance weighting to utilize more useful training instances in the source network for prediction.

5.3 Experimental results

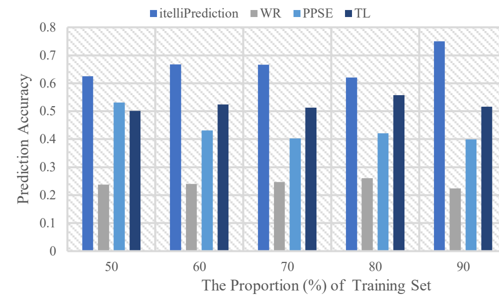
First we conduct the experiment on WikiElec, all experiments are averaged over 30 runs with ***NIDN***=50, ***Pc***=0.9, ***Pm***=0.1, ***MAXGEN***=1000, ***α***=0.5, ***β***=0.5. Results are shown in Figs. 4.



(a)Process of Finding the Optimal Solution



(b) Accuracy of Trust Prediction

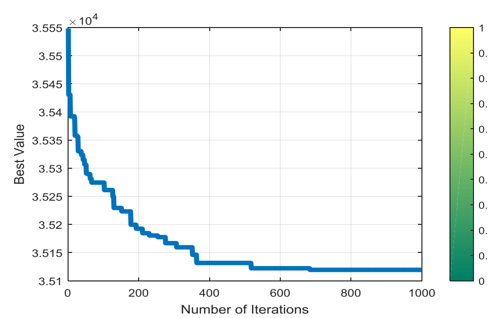


(c) Accuracy of Distrust Prediction

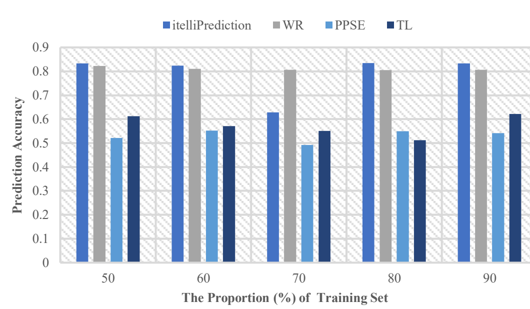
Figure 4. The Prediction Process and The Prediction Result of WikiElec

Fig.4(a) shows the process of how we find the optimal solution via HPSO algorithm. We have a total iteration of 100 times. Fig. 4(b-c) shows the accuracy of trust prediction and distrust prediction of different predictors. In the trust prediction experiment, overall, our framework is better than others. In the distrust prediction experiment, our method always achieves the highest accuracy, followed by TL, PPSE and WR. This because the latent features can be captured the common structural patterns among online users, despite the different distributions in users' relation. Thus, the new representation matrix ***G'***, can indicate the users' relationship properly.

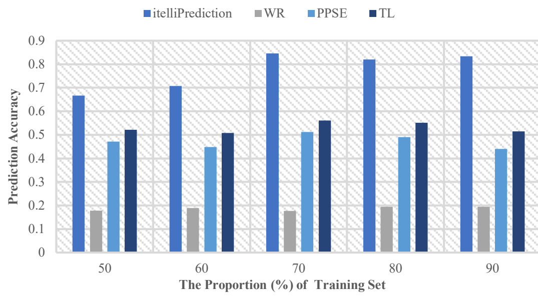
The next experiment, we apply itelliPrediction on Slashdot. The results are demonstrated in Fig. 5. The parameters are as follows: ***NIDN***=20, ***Pc***=0.87, ***Pm***=0.02, ***MAXGEN***=1000, ***α***=0.4, ***β***=0.6. The first observation is that all method performs the best in trust relation prediction. It indicates that with the network density greatly influenced the results. The gradient method put forward in this paper has better accuracy and stability, especially in distrust prediction.



(a) Process of Finding the Optimal Solution



(b) Accuracy of Trust Prediction



(c) Accuracy of Distrust Prediction

Figure 5. The Prediction Process and the Prediction Result of Slashdot.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a framework, itelliPrediction, which combined hybrid particle swarm optimization algorithm and matrix factorization for online user relation prediction. First we analyze the datasets and introduced low-rank matrix factorization. Next, we extended the method of matrix factorization, so it can be applied to factorize matrix containing both positive values and negative values. Then we conduct experiment on WikiElec. The result shows good in trust prediction and distrust prediction. Finally, extensive experiments are conducted to evaluate the scalability and the stability of the proposed framework.

This work leaves few directions for future work. One is to use different intelligence algorithm e.g., Genetic Algorithm, Artificial Immune Algorithm and Ant Colony algorithm for positive and negative matrix factorization. The other is predicting user links and their symbols by extending social balance theory and social status theory, so we can calculate how important each user is in the relation network. In the future, we will further study user relationship in a social network combining with multimodal data, to see if it can be applied it into various domains, e.g. recommendation system, traffic control and logistics distribution.

7. REFERENCES

[1] Rennie, Jasson D. M., and N. Srebro. "Fast maximum margin matrix factorization for collaborative prediction." International Conference on Machine Learning ACM, 2005:713-719.

[2] Jian-Feng, Cai, et al. "Cine cone beam CT reconstruction using low-rank matrix factorization: algorithm and a proof-of-principle study" IEEE Transactions on Medical Imaging 33.8(2014):1581 - 1591.

[3] Sainath, Tara N., et al. "Low-rank matrix factorization for Deep Neural Network training with high-dimensional output targets." International Conference on Acoustics, Speech, & Signal Processing, Icassp IEEE, 2013:6655-6659.

[4] Koren, Yehuda. Factorization meets the neighborhood: a multifaceted collaborative filtering model. [C] Acm Sigkdd International Conference on Knowledge Discovery & Data Mining. 2008:426-434.

[5] Tang, Jiliang, Gao, Huiji, Liu, Huan. mTrust: discerning multi-faceted trust in a connected world. [C] Proceedings of the fifth ACM international conference on Web search and data mining. ACM, 2012:93-102.

[6] Xu W, Liu X, Gong Y. Document clustering based on non-negative matrix factorization. [C] in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR'03. 2003:267-273).

[7] Huang, Xiaodi, et al. "Enhanced clustering of biomedical documents using ensemble non-negative matrix factorization." Information Sciences 181.11(2013):2293-2302.

[8] Tang, Jiliang, X. Wang, and H. Liu. "Integrating social media data for community detection." Proceedings of the 2011 international conference on Modeling and Mining Ubiquitous Social Media Springer-Verlag, 2011:1-20.

[9] Tang, Jiliang, et al. "Exploiting homophily effect for trust prediction." Proceedings of the sixth ACM international conference on Web search and data mining ACM, 2013:53-62.

[10] Berry, Michael W., et al. "Algorithms and applications for approximate nonnegative matrix factorization." Computational Statistics & Data Analysis 52.1(2007):155-173.

[11] S Zhou, Z Yu, C Wang. An improved non-negative matrix factorization algorithm based on genetic algorithm, 2014 International Conference on Computer Science and Electronic Technology (ICCSET 2014).

[12] Merino, S., J. Martínez, and F. Guzmán. "Metadomotic optimization using genetic algorithms." Applied Mathematics & Computation 267(2015):170-178.

[13] Ding, Chris, Tao Li, and Michael I. Jordan. "Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding." Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on. IEEE, 2008.

[14] Liben-Nowell, David, and Jon Kleinberg. "The link prediction problem for social networks." Journal of the American society for information science and technology 58.7 (2007): 1019-1031.

[15] Cui, Laizhong, et al. "Exploring A Trust Based Recommendation Approach for Videos in Online Social Network." Journal of Signal Processing Systems (2016): 1-13.

[16] Zhang, Zhijun, and Hong Liu. "Social recommendation model combining trust propagation and sequential behaviors." Applied Intelligence 43.3 (2015): 695-706.

[17] Agag, Gomaa, and Ahmed A. El-Masry. "Understanding consumer intention to participate in online travel community and effects on consumer intention to purchase travel online and WOM: An integration of innovation diffusion theory and TAM with trust." Computers in Human Behavior 60 (2016): 97-111.

[18] Matsutani, Kanji, et al. "Combining activity-evaluation information with NMF for trust-link prediction in social media." Big Data (Big Data), 2015 IEEE International Conference on. IEEE, 2015.

[19] Agrawal, Priyanka, Vikas K. Garg, and Ramasuri Narayanam. "Link Label Prediction in Signed Social Networks." IJCAI. 2013.

[20] Ostadrahimi, Leila, Miguel A. Mariño, and Abbas Afshar. "Multi-reservoir operation rules: multi-swarm PSO-based optimization approach." Water resources management 26.2 (2012): 407-427.

[21] Papadopoulos, Konstantinos A., et al. "Particle swarm optimization of antenna arrays with efficiency constraints." Progress in Electromagnetics Research M 17 (2011): 237-251.

[22] Spiliotis, M. "A Particle Swarm Optimization (PSO) heuristic for water distribution system analysis." Water Util J 8 (2014): 47-56.

[23] DuBois, Thomas, Jennifer Golbeck, and Aravind Srinivasan. "Predicting trust and distrust in social networks." Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on. IEEE, 2011.

[24] Ye, Jihang, et al. "Predicting positive and negative links in signed social networks by transfer learning." Proceedings of the 22nd international conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2013.

[25] Menon A K, Elkan C. Link Prediction via Matrix Factorization[C] European Conference on Machine Learning and Knowledge Discovery in Databases. Springer-Verlag, 2011:437-452.

[26] Kuter, Ugur, and J. Golbeck. "SUNNY: a new algorithm for trust inference in social networks using probabilistic confidence models." AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada 2007:1377-1382.

[27] Qian, Yi, and Sibel Adali. "Extended structural balance theory for modeling trust in social networks." Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on. IEEE, 2013.

[28] Hsieh, Cho-Jui, Kai-Yang Chiang, and Inderjit S. Dhillon. "Low rank modeling of signed networks." Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012.

[29] Forsati R, Mahdavi M, Shamsfard M, et al. Matrix factorization with explicit trust and distrust side information for improved social recommendation[J]. ACM Transactions on Information Systems (TOIS), 2014, 32(4): 17.