Finding high-influence microblog users with an improved PSO algorithm

Biao Zhang

School of Mechanical Engineering and Automation,
Wuhan Textile University,
Wuhan, 430074, China
E-mail: zhangbiao007@sina.com

Shuai Zhong, Kunmei Wen*, Ruixuan Li and Xiwu Gu

School of Computer Science and Technology,
Huazhong University of Science and Technology,
Wuhan, 430074, China
E-mail: zhongshuai463@gmail.com
E-mail: kmwen@hust.edu.cn
E-mail: rxli@hust.edu.cn
E-mail: guxiwu@hust.edu.cn
*Corresponding author

Abstract: Particle swarm optimisation (PSO) is a stochastic optimisation algorithm based on swarm intelligence. The algorithm applies the concept of social interaction to find optimal solution. Sina Weibo is one of the most popular Chinese microblog platforms. Microblog users participate in network interaction by publishing tweets and retweets. The influences of microblog users are determined by the users' behaviours, which exactly match the five principles of swarm intelligence. Therefore, we propose an improved PSO algorithm to find the microblog users with the maximum influence. Microblog users' retweeting behaviours can be described as a variable of the user influence space, which contains user experiences and surrounding network. The variable is defined as the velocity change in our method. By iteratively calculating based on users' behaviour, the maximum influence will be obtained. The experiments validate that our method can effectively identify the high-influence microblog users.

Keywords: particle swarm optimisation; PSO; social network; influence; Sina Weibo.

Reference to this paper should be made as follows: Zhang, B., Zhong, S., Wen, K., Li, R. and Gu, X. (2013) 'Finding high-influence microblog users with an improved PSO algorithm', *Int. J. Modelling, Identification and Control*, Vol. 18, No. 4, pp.349–356.

Biographical notes: Biao Zhang received his PhD in School of Mechanical Science and Engineering at Huazhong University of Science and Technology in 2008. He is currently a Lecturer in the School of Mechanical Engineering and Automation at Wuhan Textile University. His current research interests include swarm intelligence and emergent computation, management decision theory, and creative design.

Shuai Zhong is a graduate student of School of Computer Science and Technology at Huazhong University of Science and Technology. He received his BS degree in Huazhong University of Science and Technology. His current research interests focus on social network and swarm intelligence.

Kunmei Wen is a Lecturer in the School of Computer Science and Technology at Huazhong University of Science and Technology. She received her PhD and MS in Computer Science from Huazhong University of Science and Technology. Her current research focuses on social network, artificial intelligence and information retrieval.

Ruixuan Li is a Professor at the School of Computer Science and Technology, Huazhong University of Science and Technology. He received her PhD in Computer Science from Huazhong University of Science and Technology. His current research focuses on artificial intelligence and distributed computing.

Xiwu Gu is a Lecturer at the School of Computer Science and Technology, Huazhong University of Science and Technology. He received her PhD in Computer Science from Huazhong University of Science and Technology. His current research focuses on artificial intelligence and social network.

1 Introduction

Particle swarm optimisation (PSO) is a stochastic optimisation algorithm based on swarm intelligence. The algorithm applies the concept of social interaction to find optimal solution.

Sina Weibo is one of the most popular Chinese microblog platforms. Microblog users participate in network interaction by publishing tweets and retweets. The influences of microblog users are determined by the users' behaviours. The research about information and influence propagation in social networks has been particularly active for a number of years in some fields, for example, sociology, communication, marketing, political science and physics. However, most of the researches are based on complex network.

The influence of microblog users is determined by users' behaviours, which exactly match the five principles of swarm intelligence. Therefore, we propose an improved PSO algorithm to calculate the influence of users. Microblog users' behaviours of retweeting can be described as a variable of the user influence space, which contains user experiences and surrounding network. The variable is defined as the velocity change in the proposed method. By iteratively calculating based on users' behaviour, users' influence will be obtained.

This paper is organised as follows. In Section 2, the basic PSO is introduced. We give a brief introduction about user influence on social networks in Section 3. In Section 4, the similarity between microblog users and swarm intelligence, and the link between user behaviour and influence are discovered. In Section 5, we propose an improved PSO algorithm. Experimental results are presented in Section 6. Finally, in Section 7 conclusions and future works are given.

2 Particle swarm optimisation

PSO (Valle, 2008) is a computational intelligence-based technique that is not largely affected by the size and non-linearity of the problem. A number of papers have been published in the past few years that focus on this issue. Moreover, PSO has some advantages comparing with other similar optimisation techniques such as GA, namely the following.

1 PSO is easier to implement. There are fewer parameters to adjust.

2 In PSO, every particle remembers its own previous best value as well as the neighbourhood's; therefore, it has a more effective memory capability than the GA.

3 PSO is more efficient in maintaining the diversity of the swarm (Engelbrecht, 2006; Zhao and Wang, 2011) (more similar to the ideal social interaction in a community), since all the particles use the information related to the most successful particle in order to improve themselves.

PSO is based on two fundamental disciplines: social science and computer science. In addition, PSO uses the swarm intelligence concept, which is the property of a system, whereby the collective behaviours of unsophisticated agents that are interacting locally with their environment create coherent global functional patterns. Therefore, the cornerstones of PSO can be described as follows.

1 Social concepts (Eberhart et al., 2001): It is known that 'human intelligence results from social interaction'. Evaluation, comparison, and imitation of others, as well as learning from experience allow humans to adapt to the environment and determine optimal patterns of behaviour, attitudes, and suchlike. In addition, a second fundamental social concept indicates that 'culture and cognition are inseparable consequences of human sociality'. Culture is generated when individuals become more similar due to mutual social learning. The sweep of culture allows individuals to move towards more adaptive patterns of behaviour.

2 Swarm intelligence principles (Eberhart et al., 2001; Kennedy and Eberhart, 1995; Eberhart and Kennedy, 1995; Millonas, 1994): swarm intelligence can be described by considering five fundamental principles.

- Proximity principle: the population should be able to carry out simple space and time computations.
- Quality principle: the population should be able to respond to quality factors in the environment.
- Diverse response principle: the population should not commit its activity along excessively narrow channels.
- Stability principle: the population should not change its mode of behaviour when the environment changes.
- Adaptability principle: the population should be able to change its behaviour mode when it is worth the computational price.

3 Computational characteristics (Eberhart et al., 2001; Dong et al., 2011; Chen et al., 2011): Swarm intelligence provides a useful paradigm for implementing adaptive systems. In particular, PSO is an extension, and a potentially important incarnation of cellular automata (CA). The particle swarm can be

conceptualised as cells in CA, whose states change in many dimensions simultaneously. Both PSO and CA share the following computational attributes.

- individual particles (cells) are updated in parallel
- each new value depends only on the previous value of the particle (cell) and its neighbours
- all updates are performed according to the same rules.

In the real number space, each individual possible solution can be modelled as a particle that moves through the problem hyperspace. The position of each particle (Kennedy, 1997; Wang et al., 2011) is determined by the vector $x_i \in R^n$ and its movement by the velocity of the particle $v_i \in R^n$, as shown in formula (1):

$$x_i(t) = x_i(t-1) + v_i(t) \tag{1}$$

The information available for each individual is based on its own experience (the decisions that it has made so far and the success of each decision) and the knowledge of the performance of other individuals in its neighbourhood. Since the relative importance of these two factors can vary from one decision to another, it is reasonable to apply random weights to each part, and therefore the velocity will be determined by

$$\begin{aligned} v_i(t) = \omega \cdot v_i(t-1) + \varphi_1 \cdot rand_1 \cdot \left(p_i - x_i(t-1)\right) \\ +\varphi_2 \cdot rand_2 \cdot \left(p_g - x_i(t-1)\right) \end{aligned} \tag{2}$$

where φ_1, φ_2 and ω are three positive numbers, $rand_1$ and $rand_2$ are two random numbers with uniform distribution in the range of [0.0, 1.0].

The velocity update equation in formula (2) has three major components (Boeringer and Werner, 2004; Huynh and Dunnigan, 2012).

1. The first component is sometimes referred to as 'inertia', 'momentum', or 'habit'. It models the tendency of the particle to continue in the same direction it has been travelling.
2. The second component is a linear attraction towards the best position ever found by the given particle: (whose corresponding fitness value is called the particle's best: P_{best}), scaled by a random weight φ_1 $rand_1$. This component is referred to as 'memory', 'self-knowledge', 'nostalgia', or 'remembrance'.
3. The third component of the velocity update equation is a linear attraction towards the best position found by any particle: P_g (whose corresponding fitness value is called global best: g_{best}), scaled by another random weight φ_2 $rand_2$. This component is referred to as 'cooperation', 'social knowledge', 'group knowledge', or 'shared information'.

3 User influence

The study of information and influence propagation in social networks has been particularly active for a number of years in some fields, such as sociology, communication, marketing, political science and physics. Earlier work focused on the effects that scale-free networks and the affinity of their members for certain topics had on the propagation of information (Wu et al., 2004). Others discussed the presence of key influential (Domingos and Richardson, 2001; Goyal et al., 2010; Weng et al., 2010) in a social network, defined as those who are responsible for the overall information dissemination in the network.

Huberman et al. (2009) studied the social interactions on Twitter to reveal that the driving process for usage is a sparse hidden network underlying the friends and followers, while most of the links represent meaningless interactions. Jansen et al. (2009) have examined Twitter as a mechanism for word-of-mouth advertising. They considered particular brands and products and examined the structure of the postings and the change in sentiments.

There are some earlier studies focused on social influence and propagation. Aral et al. (2009) have distinguished the effects of homophiles from influence as motivators for propagation. As to the study of influence within Twitter, Cha et al. (2010) have performed a comparison of three different measures of influence – indegree, retweets and user mentions. They discovered that while retweets and mentions correlated well with each other, the indegree of users did not correlate well with the other two measures. Based on this, they hypothesised that the number of followers may not be a good measure of influence. On the other hand, Weng et al. (2010) have proposed a topic-sensitive PageRank measure for influence in Twitter. The measure is based on the fact that they observed high reciprocity among follower relationships in their dataset, which they attributed to homophile.

Recently, Romero et al. (2010) introduced a novel influence measure that takes into account the passivity of the audience in the social network. They developed an iterative algorithm to compute influence in the style of the HITS algorithm and empirically demonstrated that the number of followers is a poor measure of influence.

4 Microblog and swarm intelligence

4.1 Microblog user behaviours

In general, the behaviours of microblog users are divided into the following categories: sending original tweets, retweeting and commenting on someone's tweets. Replies can only be seen by the one who has also made comment or has been replied. Therefore, commenting has limitation on propagation of information and influence.

The behaviours of microblog users are affected by two factors: personal and social.

User's cognitive level and personal values are directly reflected in the user's behaviours. Therefore, the impact from user's own behaviour is stable.

User's social relations also have impact on user's behaviours. In Weibo, users have two kinds of relationships: *followers*, with a capability of browsing all users' tweets, and *friends*, with a capability of browsing the tweets from users. Both information from *friends* and the feedback of retweeting from *followers* can affect user's behaviours.

4.2 Microblog user behaviours and swarm intelligence

Swarm intelligence has five basic principles. In fact, the characteristics of behaviour of microblog user are corresponding to these principles:

- Microblog users can compute simple space and time.
- Microblog users can response to the change of the environment.
- Different users have different behaviours, thus these discrepancies have been ensuring the diversity of environment.
- The deviations of users' cognitive level and personal values are directly related to the user's behaviours. User's behaviours are stable if the change of users' cognitive level and personal values is slow.
- When users are changing, users' behaviours would be changing gradually. Meanwhile, user is affected by his followers and friends. As a result, user's behaviours would be changed via the user's own changes and the changes of his social mates.

Considering these characteristics above, we believe that microblog user's behaviour can be described as one kind of swarm intelligence.

4.3 Microblog user behaviour and user influence

There are several important factors determining the users' behaviours.

Firstly, it is obvious that users have a certain number of *followers*. When one user has few *followers*, apparently, it is hard to propagate his own tweets to whole social networks. The reason is that the fewer *followers* who can directly browse the user's tweets, the fewer *retweeters* who can retweet the user's tweets and the fewer users who can browse the original user's tweets. Therefore, users should have sufficient number of *followers* to propagate his tweets and influence. However, some researchers have discovered that the *followers* of users did not correlate well with user influence. Sometimes more followers do not necessarily mean more influence.

In addition, retweeting is significant to users' influence. Users retweet some tweets in which they are interested. The more tweets being retweeted, the more influence value the tweets have. Therefore, user who has many forwarded tweets can be considered as an influential user.

Moreover, user's influence also has close relationship with the influence of the user's *followers* and *friends*. A *follower* who has higher influence can do much better in spreading this user's tweets than a *follower* who has lower influence. Meanwhile, an influential *friend* provides better performance than a low-influential friend does. Hence, a *follower/friend* who has a higher influence on tweeter plays a significant role in increasing the user's influence.

5 An improved PSO algorithm for evaluating user influence

Considering the similarity between user behaviours and swarm intelligence, we proposed an improved PSO algorithm to calculate the users' influences.

5.1 Influence function

Given two users u_1 and u_2, when u_1 has retweeted u_2 or u_1 has been retweeted by u_2, the incremental of u_1's influence Δ_{12} is defined as:

$$\Delta_{12} = \varphi_{12} \cdot Influ_2 \tag{3}$$

where φ_{12} is the impact factor of behaviours between u_1 and u_2, $Influ_2$ is u_2's influence.

In the user set $U = \{u_1, u_2, u_3, \ldots, u_n\}$, user u_1's global influence is calculated as follows:

$$Influ_1(t+1) = Influ_1(t) + \sum\nolimits_{u_i \in U} \Delta_{1i} \tag{4}$$

Combine (1) and (2), we get user's influence equation:

$$Influ_1(t+1) = \theta_1 I_U \tag{5}$$

where $\theta_1 = [\varphi_1, \varphi_{12}, \varphi_{13}, \ldots, \varphi_{1n}]$ is the vector of impact factor of behaviours between user u_1 and other user in user set U. $I_U = [Influ_1, Influ_2, Influ_3, \ldots, Influ_n]$ is user influence vector.

5.2 Define of velocity and position

Given users set $U = \{u_1, u_2, u_3, \ldots, u_n\}$ as a n-dimension user space, each dimension represents a user. User u_1's position in each dimension is determined by the impact factor with other user represented by other $n - 1$ dimension. We get the position in the whole space.

When user u_1 have made some behaviour like tweeting or retweeting, the impact factors between u_1 and other users have been changed. X_1 would be changed too. Let this variation of position be velocity.

In basic PSO algorithm, the velocity update equation has three major components: 'habit', 'self-knowledge' and 'social-knowledge'. The velocity V_1 can also be divided into three components.

Suppose that some behaviour happened between users u_1 and u_2 at some time, both users would expect these behaviours will happen again in future because of the memorability of users. Based on these expect, users u_1 and u_2 would keep attention on each other in future. These attention will reduce with time elapsing. So does the impact factor.

Moreover, the user's behaviour are associated with the impact of the individual and society. Microblog users receive information from their *friends*. Among this information, user chooses some information which the user considers valuable to retweet. At the same time, user's tweets are retweeted by his *followers*. From his *followers'* retweeting, user got some feedbacks about which kind of information would be accepted and which would not.

User's 'self-knowledge' is derived from the feedbacks of behaviours of retweeting by his *followers*. User will improve his behaviours according to these feedbacks.

Since user's retweeting his *friends'* tweets is a choice behaviour, it is obviously that the more influence the social network has on the user, the more the user has retweeted other's tweets. Thus user's 'social-knowledge' can be represented by the feature of behaviours that user's retweeting on his *friends'* tweets.

Refer to the velocity update equation in basic PSO algorithm, the velocity update equation to calculate user influence is shown as:

$$v(t+1)_1 = \omega \cdot v(t)_1 + \varphi_p \theta_{1beRt} + \varphi_g \theta_{1Rt} \tag{6}$$

where these equation also has three components:

The first component is $\omega \cdot v(t)_1$. This component represents 'habit'. This ω is inertia constant and $v(t)_1$ is previous velocity of user u_1.

The second component is $\varphi_p \theta_{1beRt}$. This component represents 'self-knowledge'. This φ_p is personal constant. To calculate θ_{1beRt}, first we calculate *be-retweeted ratio*:

$$\alpha_{1i} = \frac{\sum retweets_{1i}}{\sum tweets_1} \tag{7}$$

where the denominator is the number of tweets that user u_1 has sent in a certain period. The numerator represents the number of tweets that user u_i has retweeted the tweets of u_1 in the same period. Expanding to the n-dimension user space, we get θ_{1beRt} finally:

$$\theta_{1beRt} = [\alpha_{11}, \alpha_{12}, \alpha_{13}, .., \alpha_{1n}] \tag{8}$$

The third component is $\varphi_g \theta_{1Rt}$. This component represents 'social-knowledge'. This φ_g is social constant. To calculate θ_{1Rt}, we calculate *retweeting ratio*:

$$\beta_{1i} = \frac{\sum retweets_{i1}}{\sum tweets_1} \tag{9}$$

where the denominator is the number of tweets that user u_i has sent in a certain period. The numerator represents the number of tweets that user u_1 has retweeted the tweets of u_i in the same period. Expanding to the n-dimension user space, we get:

$$\theta_{1Rt} = [\beta_{11}, \beta_{21}, \beta_{31}, .., \beta_{n1}] \tag{10}$$

Therefore, the velocity update equation and position update equation are shown as:

$$\begin{aligned} v(t+1)_1 &= \omega \cdot v(t)_1 + \varphi_p \theta_{1beRt} + \varphi_g \theta_{1Rt} \\ x(t+1)_1 &= x(t)_1 + v(t+1)_1 \end{aligned} \tag{11}$$

5.3 Differences between basic PSO and our algorithm

Basic PSO algorithm utilises a 'population' of particles that fly through the problem hyperspace with given velocities. Though each iteration, the velocities of the individual particles are stochastically adjusted according to the historical best position for the particle itself and the neighbourhood best position. Both the particle best and the neighbourhood best are derived according to a user defined fitness function. The movement of each particle naturally evolves to an optimal or near-optimal solution.

However, in our proposed algorithm, we use influence function as fitness function to calculate user's influence. Since there is no optimal or near-optimal solution for user's influence, we do not need the historical best position for the particle itself and the neighbourhood best position. In addition, we do not need the random number because user's behaviours are diversity.

6 Experiment and result analysis

To test the performance of our algorithm, we use the dataset that are obtained from Sina Weibo. Sina Weibo is a Chinese microblog (weibo) website. Similar to a hybrid of Twitter and Facebook, it is one of the most popular sites in China. Over 30% of internet users are also the users of Sina Weibo, with a similar market penetration that Twitter established in the USA.

Sina Weibo provides a search API for extracting tweets and information of users. To obtain the dataset, we continuously queried the Sina Weibo Search API for a period of 720 hours starting on 1 April 2011.

There are more than 560 K users, 36 M tweets and 1.6 M relationships. Because of the limitation of Sina Weibo API, the obtained information about users and tweets is incomplete.

1,151 users are chosen to test our algorithm. There are 16,781 relationships among these users and 36,166 tweets in our querying period.

We first give some statistics on our dataset. In Figure 1 and Figure 2, it is shown that the retweet ratio is not high.

The parameters are assigned by empirical value: ω is 0.8, φ_p and φ_g both are 0.5. The velocities of all users are initialised to the value 0. The position is determined by the number of user's followers, described by 1/*follower Num*.

Most users have few followers (1 to 5 followers), most users' influence is low after the completion of the first iteration. It is shown in Figure 3.

In Figure 4 and Figure 5, some users' influence has increased because of the users' behaviours. Those bottom users in Figure 5 have few followers or little tweets or retweets.

We have chosen four typical users to find out what happened in their influence in this period. It is shown in Figure 6.

In the first 10 days, the influence curve goes down, because the initial value of position is higher than their real values. However, the influence curve representing user's influence became normal and valid since 10th day. From 10th day, these users have shown their different influences.

According to the whole experimental results it shown that the algorithm becomes effective since 7th or 10th day, because our algorithm does not have sufficient precise data to calculate user influence before the time for the distortion of initial value. However, when the algorithm has obtained sufficient data via iterations of earlier days, it can do much better in calculating user influence.

Figure 1 The number of tweets and retweets

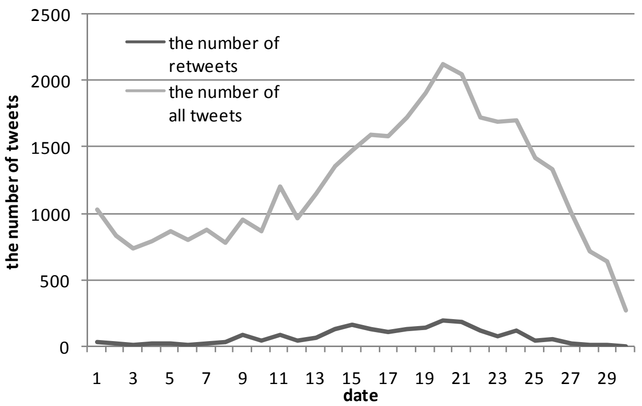


Figure 2 The ratio of retweets

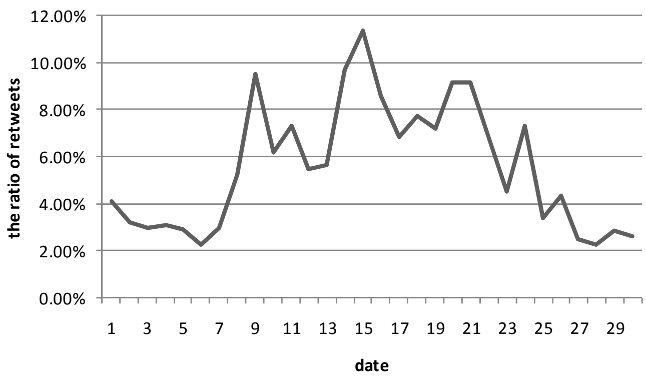


Figure 3 User influence after first iteration

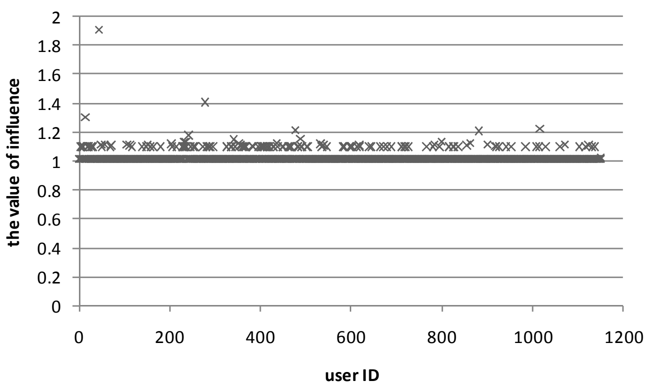
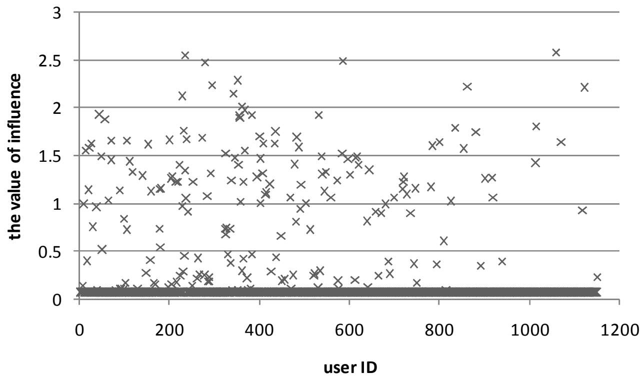


Figure 4 User influence after 15th iteration



We ranked all users according to the values of influence. It is shown in Figure 7. We found that all users whose rank is behind 265 have the minimum value of influence, which is 0.1443.

Figure 5 User influence after 30th iteration

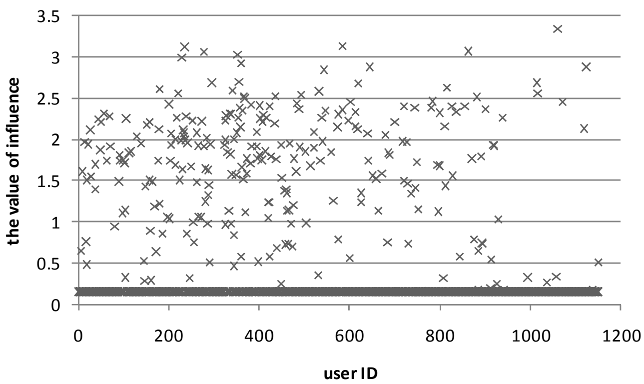


Figure 6 Chosen users' influence in the month

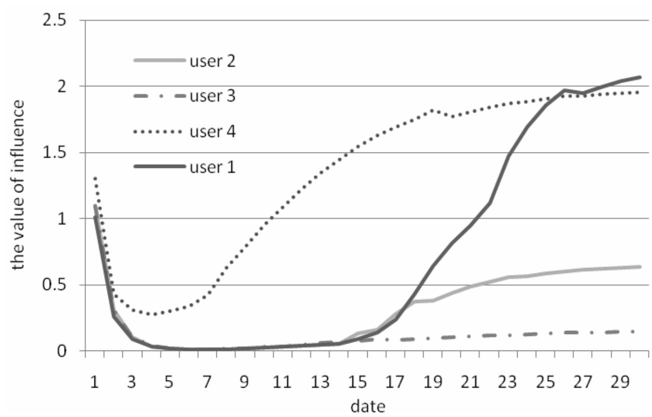


Table 1 Max and min influence

Max	3.3293
Min	0.1443

Figure 8 shows the amount of ranked users' tweets. It is shown that the amount of tweets is not a sufficient condition which can guarantee a high influence user.

Figure 7 Ranked users influence

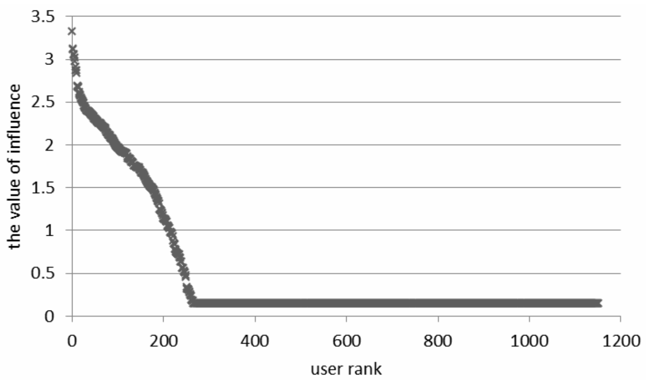


Figure 8 Number of tweets that users have sent

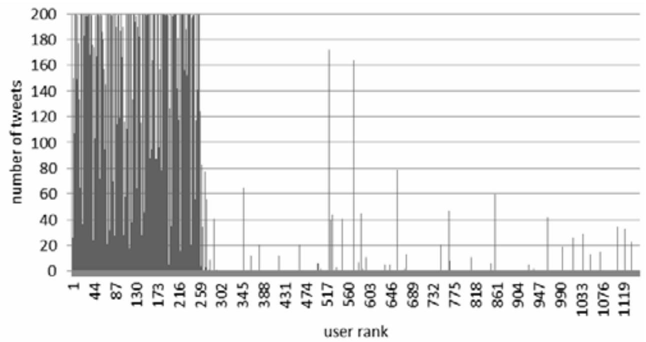
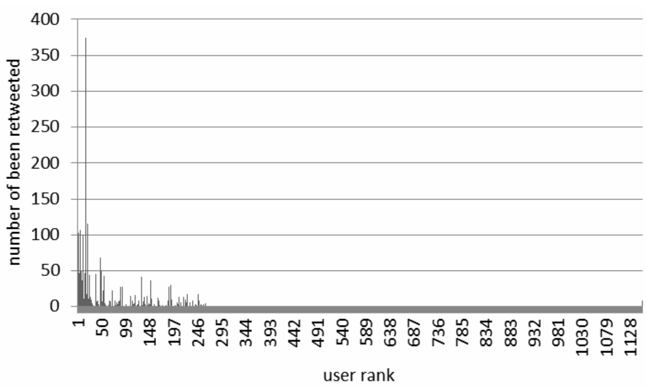


Figure 9 Number of user been retweeted



The number of retweets can reflect user influence. The more user has been retweeted, the higher influence user has. Thus, in Figure 9, we counted the number of each user been retweeted. It is clear that the users that been more retweeted have higher influence.

These experiment results are shown that our improved PSO algorithm is effective in calculating user influence.

7 Conclusions and future works

PSO is a stochastic-based search technique that has its roots in artificial life and social psychology, as well as in engineering and computer science. It utilises a 'population' called particles, which flows through the problem hyperspace with given velocities; in each iteration, velocities are stochastically adjusted considering the historical best position for the particle itself and the neighbourhood best position (both of them defined according to a predefined fitness function). Then, the movement of each particle naturally evolves to an optimal or near-optimal solution.

Evaluating user's influence has become important in social networks. There have some factors to determine user's influence like the number of user's followers or the number of user being retweeted. The important thing is that the characteristics of users' behaviour have many similarities to swarm intelligence.

In this paper, we propose an improved PSO algorithm to evaluate users' influence. This algorithm takes personal-knowledge and social-knowledge into account. The experiment results show that the algorithm can be valid in calculating user influence.

However, there are some issues for future work. The selection of parameters and other factors may affect user influence.

Acknowledgements

This paper is supported by National Natural Science Foundation of China under Grant Nos. 61173170, 60873225 and 70771043, CCF Opening Project of Chinese Information Processing, Central University basic research special funding (Innovation Fund of Huazhong University of Science and Technology under Grant No. 2011TS135 and 2010MS068).

References

Aral, S., Muchnik, L. and Sundararajan, A. (2009) 'Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks', *Proceedings of the National Academy of Sciences*, Vol. 106, No. 51, pp.21544–21549.

Boeringer, D. and Werner, D. (2004) 'Particle swarm optimization versus genetic algorithms for phased array synthesis', *IEEE Trans. Antennas Propagat., Mar.*, Vol. 52, No. 3, pp.771–779.

Cha, M., Haddadi, H., Benevenuto, F. et al. (2010) 'Measuring user influence in Twitter: the million follower fallacy', *4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, pp.10–17.

Chen, Y.J., Cui, Z.H., Yin, J. et al. (2011) 'Global minimum structure optimization of Lennard-Jones clusters by hybrid PSO', *International Journal of Modelling, Identification and Control (IJMIC)*, Vol. 14, No. 4, pp.303–309.

Domingos, P. and Richardson, M. (2001) 'Mining the network value of customers', *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.57–66.

Dong, Z.S., Guo, Y.N. and Zeng, J.C. (2011) 'Recurrent hidden Markov models using particle swarm optimization', *International Journal of Modelling, Identification and Control (IJMIC)*, Vol. 14, No. 4, pp.258–264.

Eberhart, R. and Kennedy, J. (1995) 'A new optimizer using particle swarm theory', *Proc. 6th Int. Symp. Micro Machine and Human Science (MHS)*, October, pp.39–43.

Eberhart, R., Shi, Y. and Kennedy, J. (2001) *Swarm Intelligence*, Morgan Kaufmann, San Mateo, CA.

Engelbrecht, A.P. (2006) 'Particle swarm optimization: where does it belong', *Proceedings of the IEEE Swarm Intell. Symp.*, May, pp.48–54.

Goyal, A., Bonchi, F. and Lakshmanan, L.V.S. (2010) 'Learning influence probabilities in social networks', *ACM International Conference on Web Search and Data Mining*, pp.241–250.

Huberman, B.A., Romero, D.M. and Wu, F. (2009) 'Social networks that matter: Twitter under the microscope', *First Monday*, January, Vol. 14, Nos. 1.

Huynh, D.C. and Dunnigan, M.W. (2012) 'Advanced particle swarm optimization algorithms for parameter estimation of a single-phase induction machine', *International Journal of Modelling, Identification and Control (IJMIC)*, Vol. 15, No. 4, pp.227–240.

Jansen, B., Zhang, M., Sobel, K. et al. (2009) 'Twitter power: Tweets as electronic word of mouth', *Journal of the American Society for Information Science and Technology*, Vol. 60, No. 11, pp.2169–2188.

Kennedy, J. (1997) 'The particle swarm: social adaptation of knowledge', *Proc. IEEE Int. Conf. Evol. Comput.*, April, pp.303–308.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', *Proc. IEEE Int. Conf. Neural Netw. (ICNN)*, November, Vol. 4, pp.1942–1948.

Millonas, M.M. (1994) 'Swarms, phase transitions, and collective intelligence', in Langton, C.G. (Ed.) *Artificial Life III, Proc. Santa Fe Institute Studies in the Sciences of Complexity*, Addison-Wesley, New York, Vol. 17, pp.417–445.

Romero, D.M., Galuba, W., Asur, S. et al. (2010) 'Influence and passivity in social media', *Proceedings of the 20th international conference companion on World Wide Web*, pp.113–114.

Valle, Y.D. (2008) 'Particle swarm optimization: basic concepts, variants and applications in power systems', *IEEE Transaction on Evolutionary Computation*, April, Vol. 12, No. 2, pp.171–194.

Wang, W.J., Wang, H. and Rahnamayan, S. (2011) 'Improving comprehensive learning particle swarm optimizer using generalized opposition-based learning', *International Journal of Modelling, Identification and Control (IJMIC)*, Vol. 14, No. 4, pp.310–316.

Weng, J.S., Lim, E.P., Jiang, J. et al. (2010) 'TwitterRank: finding topic-sensitive influential twitterers', *ACM International Conference on Web Search and Data Mining*, pp.261–270.

Wu, F., Huberman, B.A., Adamic, L. et al. (2004) 'Information flow in social groups', *Physica A*, Vol. 337, Nos. 1/2, pp.327–335.

Zhao, C.X. and Wang, R.C. (2011) 'Particle swarm optimization based on Monte Carlo localization for mobile sensor network', *International Journal of Modelling, Identification and Control (IJMIC)*, Vol. 14, No. 4, pp.242–249.