Computers, Materials & Continua CMC, vol.63, no.2, pp.755-768, 2020

Improvement of Stochastic Competitive Learning for Social Network

Wenzheng Li[1] and Yijun Gu[1, *]

Abstract: As an unsupervised learning method, stochastic competitive learning is commonly used for community detection in social network analysis. Compared with the traditional community detection algorithms, it has the advantage of realizing the time-series community detection by simulating the community formation process. In order to improve the accuracy and solve the problem that several parameters in stochastic competitive learning need to be pre-set, the author improves the algorithms and realizes improved stochastic competitive learning by particle position initialization, parameter optimization and particle domination ability self-adaptive. The experiment result shows that each improved method improves the accuracy of the algorithm, and the F1 score of the improved algorithm is 9.07% higher than that of original algorithm.

Keywords: Stochastic competitive learning, particle swarm optimization, algorithm improvement.

1 Introduction

The theories and approaches to the structure of complex networks is rapidly developing with the popularity of social networks. It is especially important to extract effective information in network data, accurately obtain the hidden information of individuals or communities in the network, and complete the accurate detection of social network communities. Classic community detection methods include Louvain algorithm [Blondel, Guillaume, Lambiotte et al. (2008)], GN algorithm [Newman and Girvan (2004)], DSCEN algorithm [McAuley and Leskovec (2014)], and spectral clustering [Zhao, Yuan, Nie et al. (2018)] et al. The traditional community detection algorithm has realized the community detection by analyzing the complex network structure from different aspects. However, with the emergence of application requirements such as public opinion communication and community dynamic evolution analysis [Pan, Deng and Dong (2011)], the traditional static community detection algorithm has been difficult to cope with.

In order to implement a time-series community detection algorithm, competitive learning algorithm is applied to it. Competitive learning is one of the main achievements of unsupervised learning, and it has many applications such as clustering [Liu, Pang and

[1] School of Information Technology and Cyber Security, People's Public Security University of China, Beijing, 100038, China.

[*] Corresponding Author: Yijun Gu. Email: guyijun@ppsuc.edu.cn; sdrzlwz@126.com.

Received: 17 July 2019; Accepted: 06 August 2019.

Lloyd (2008)] and pattern recognition [Bacciu and Starita (2008)]. In 2008, an Isotropic multi-particle competition mechanism was proposed [Quiles, Zhao, Alonso et al. (2008)]. The basis of stochastic competitive learning is the particle competition mechanism. In 2012, the particle competition mechanism was first applied to clustering problems and the clustering function was improved by introducing nonlinear dynamics and mathematical models [Silva and Zhao (2012)]. Later, Silva et al. [Silva and Zhao (2013)] proposed a community detection algorithm based on proposed stochastic competitive learning algorithm, successfully combining dynamic models with community detection, which realized time-series community detection.

Because social networks are different from other complex network structures, such as complete graphs and random graphs, social networks often have different communities [Fu, Gu and Zhao (2017)] and vary in size [Liu (2008)]. Therefore, in the actual application process, the stochastic competitive learning algorithm has the following two problems. On the one hand, the initialization of particle positions in stochastic competitive learning is randomly selected. Although the final community detection result caused by particle moving is only affected by the network structure, if the initial location of the particles is concentrated in few communities, the result of community detection would be difficult to complete convergence within a limited time, which leads to a reduction in the accuracy of the algorithm. On the other hand, in the stochastic competitive learning, there is the ratio of preferential walk in random-preferential walk named parameter λ, the particle energy change parameter Δ, and the particle control capability value that are previously artificially set. The artificial setting of the parameters will reduce the practical performance of the algorithm, and it is difficult to ensure the accuracy of the algorithm.

To solve above problems, firstly, this paper constructs the particle initialization algorithm of stochastic competitive learning based on Jaccard similarity. Then, the particle swarm optimization algorithm is used to optimize the parameters of stochastic competitive learning. Finally, the particle domination ability is improved by the self-adaptive method. The experiment is based on the open source SNAP community data. The experiment results show that the above three methods improve the accuracy of stochastic competitive learning. The improved stochastic competitive learning algorithm improves the F1 score by 9.07% compared with the original algorithm.

The content of this paper is arranged as follows. Section 2.1 introduces the stochastic competitive learning. Section 2.2 proposes the improvement methods from three aspects with the features of social network. Section 3.1 introduces the source and reliability of data set. In Section 3.2, the three methods mentioned above are respectively tested, and the experimental results are analyzed. Section 3.3 combined the above three methods to achieve the improved stochastic competitive learning, and verified the effect of the improved algorithm with experiments.

2 Related works and improve methods

2.1 Introduction to stochastic competitive learning

Stochastic competitive learning is a competitive dynamics system composed of multiple particles. The unsupervised learning method is used to realize the dynamic community

partitioning function. Given a network G=(V, E), V is the set of nodes, and E is the set of edges. In the process of stochastic competitive learning, a set of particles K={1, 2, 3, ..., k} is randomly placed at the nodes of the network. The goal of each particle is to dominate new nodes while defending the nodes that they have already dominated. As the limited number of nodes in the network, competition between particles would naturally occur. Nodes are equivalent to resources in the whole competition process. When particles visit any node, they will enhance their domination level over that node, and weaken the domination level of other particles at the same node. The frequency that particles visit nodes can be represented by a matrix as Eq. (1), and the domination level can be expressed as a domination element in the domination level matrix as shown in Eq. (2). Since each time a particle visits a node, the particle's domination level value for that node is increased by one. Therefore, the domination level of the particle to the node is numerically equal to the number of times the particle visits the current node, so Eqs. (1) and (2) are equivalent in the calculation result [Silva and Zhao (2016)].

$$\mathrm{N}_i(t) \triangleq [\mathrm{N}_i^{(1)}(t), \mathrm{N}_i^{(2)}(t), ..., \mathrm{N}_i^{(k)}(t)]^T \tag{1}$$

$$\overline{\mathrm{N}}_i(t) \triangleq [\overline{\mathrm{N}}_i^{(1)}(t), \overline{\mathrm{N}}_i^{(2)}(t), ..., \overline{\mathrm{N}}_i^{(k)}(t)]^T \tag{2}$$

In the equation, $\mathrm{N}_i(t)$ records the total times each particle in the network reaches node i until time t, and $\overline{\mathrm{N}}_i(t)$ represents the domination level value of each particle to node i at time t. Node i belongs to the community represented by the particle with the highest domination level value. The Belong(k)(t) function defines the set of nodes belonging to particle k at time t, as shown in Eq. (3).

$$\mathrm{Belong}^{(k)}(t) = \{u \mid u \in V, \max(\overline{\mathrm{N}}_u(t)) = \overline{\mathrm{N}}_u^{(k)}(t)\} \tag{3}$$

In addition, in order to enhance the domination level of the particle on nodes which already dominated by the particle, and further to promote the formation of obvious boundaries of communities. Active stage and silent stage were introduced into the algorithm. When the particle in the network is active it would be guided to walk with a specific walk rule. This rule is the result of the combination of the random walk and the preferential walk adjusted by parameter λ, which named random-preferential walks. In addition, in order to prevent the particles from traveling for a long time in other particle-dominated communities that would disturb the domination situation, the energy value of each particle was set. When the active particle visits the node dominated by other particles, the energy value decreases, otherwise increases. If the energy value is less than zero, the particle enters the silent state from the active state. The silent state of the particle will automatically jump to the node with the highest level of particle domination, and then initialize the energy value to re-enter the active state and continue to walk. Set the vector E(k)(t) to represent the energy value of the particle k at time t. The update rule of energy is as shown in Eq. (4) [Silva and Zhao (2016)].

$$\mathrm{E}^{(k)}(t) \triangleq \begin{cases} \min(\omega_{\max}\mathrm{E}^{(k)}(t-1)+\Delta), owner(k,t) \\ \max(\omega_{\min}\mathrm{E}^{(k)}(t-1)-\Delta), \neg owner(k,t) \end{cases} \tag{4}$$

In the equation, $\omega_{\max}$ and $\omega_{\min}$ respectively represent the maximum and minimum values of the energy of the particle, and the function *owner* can determine whether the *k* particle is at the position of the node under its domination at time *t*.

After multiple random-preferential walks of the particles, the domination area of each particle tends to be stable, thereby reaching a convergence state, and completing the work of community detection. Since the stochastic competitive learning algorithm is essentially a dynamic system, the algorithm can implement dynamic community evolution calculations that cannot be implemented by other community detection algorithms.

2.2 Improve methods for stochastic competitive learning

2.2.1 Initialization of particle position

Different from network structures such as *random graphs* and *complete graphs*, social networks usually consist of many communities [Fu, Gu and Zhao (2017)]. In the stochastic competitive learning, the initialization of the particle position is randomly selected. This method of initialization may cause the starting location of particle concentrated in few communities, which would cause a long time of walking until convergence. In practical applications, it is necessary to limit the time of particle walk. If the total time of the walk is determined, the accuracy of the community detection result is often reduced because the stochastic competitive learning process does not reach the convergence state. In order to make the position selected by the particles in the initialization as scattered as possible, Jaccard similarity is selected as the method for particle position initialization. The particle position initialization method is as shown in Algorithm 1.

Algorithm 1: Particle Location Initialization

Input: Network G=(V,E), K particles are supposed to be filled in the network.

Output: Initialized particle position list

(a) The degree centrality values of nodes in the network G are arranged in descending order, and the node with the largest value of the centrality is the location of the first particle.

(b) If the number of particles whose initial positions have been determined is equal to K, then Step e is performed, otherwise Step c is performed

(c) Calculate the Jaccard similarity between the nodes where the currently determined particle is located and the other node not occupied by any particle, and then perform Step d.

(d) There is a node n_i, and the average value of the Jaccard similarity value between this node and the node where other particles are located is the smallest, then the node n_i is determined as the position of the next particle. Hence, the initial position of the next particle in the network is the node n_i, the number of particles is increased by one. Return b.

(e) Complete the particle position initialization and output the initial position of particles.

Algorithm 1 judges the degree of closeness of correlation between nodes by Jaccard similarity, and selects the scattered nodes as the initial position of the particles. In terms of time complexity, it is assumed that the total number of nodes in the network G is N. Therefore, the times of the Jaccard similarity calculation needs to be performed each time the algorithm is executed is the product of the current number of particles and the number of other nodes in the network. Therefore, the times of the calculation of similarities in the

algorithm is (N-1)*(1+2+...+K-1). Since the number of particles in the stochastic competitive learning is much smaller than N, the K is taken as a constant, then the time complexity of the algorithm is Ω=O(N). If the algorithm directly searches the network for the set composed of K nodes with the smallest Jaccard similarity value, it is necessary to perform the Jaccard similarity calculation C_N^K times. The time complexity of this method is Ω_1=O(N^K) which is much larger than that of Algorithm 1. Therefore, in order to control the time complexity within a reasonable degree, Algorithm 1 is selected as the particle position initialization method.

2.2.2 Parameter optimization based on particle swarm optimization algorithm

In 1995, Kennedy and Eberhart proposed a search algorithm based on group collaboration, namely Particle Swarm Optimization (PSO) [Kennedy and Eberhart (2002)]. The algorithm needs to set the position vector $X_i=\{x_{i1,...,}x_{id}\}$ and velocity vector $V_i=\{v_{i1,...,}v_{id}\}$ for each particle used for searching, and d is the dimension of the search space. PSO needs to record the historical best position p_{id} of its particle and the historical best position p_{gd} of the group each time the search operation is performed, and adjust the particle position X_i in combination with the velocity vector V_i [Li, Zhang and Chen (2013)]. Each time a search operation is performed, the position vector and velocity vector of the particle change as shown in Eqs. (5) and (6).

$$v_{id}^{t+1} = wv^t v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) \tag{5}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{6}$$

In the equation, c1 and c2 are learning factors. In the normal condition, c_1=c_2=2, and r_1 and r_2 are random numbers between [0, 1]. w is the inertia weight, which was proposed by Shi and Eberhart to balance the ratio of global search and local search in particle swarm optimization and improve the performance of the algorithm [Li, Zhang and Chen (2013)].

In Stochastic Competitive Learning, the ratio of preferential walk in random-preferential walk named parameter λ. If the parameter λ is much too large, the movement of the particles will be excessively restricted to the dominated area. If the value of λ is much too small, the particles will not have a tendency to dominate the communities that have been dominated by themselves, which would lead to the situation that current particle-dominated community being easily dominated by other particles. The particle energy change parameter Δ determines the ability of the particle to move outside the dominated area. The larger the parameter, the greater the tendency of the community dominated by the particle to expand outward.

How to make the parameter selection of the stochastic competitive learning suitable for the current network is a problem that needs to be discussed in this section. With the help of the particle swarm optimization algorithm, the parameters λ and Δ can be dynamically adjusted with different network to solve the above problems. In addition, parameter optimization also eliminates the problem of parameter selection during stochastic competitive learning. Therefore, the dynamic adjustment of parameters can improve the accuracy of the algorithm, and enhance the practical effect of stochastic competitive

learning. The stochastic competitive learning parameter optimization algorithm based on PSO algorithm is shown as Algorithm 2.

Algorithm 2: Parameter Optimization of Stochastic Competitive Learning Based on PSO

Input: Network G=(V, E)

Output: Community Detection Results

(a) The ratio of preferential walk in random-preferential walk named parameter λ, the particle energy change parameter Δ, and put these two parameters into PSO method searching for optimal parameter.

(b) Iteratively calculate the particle position in the PSO algorithm. Then, update each particle position and moving speed by comparing the current position x_{id} with p_{id} and p_{gd}.

(c) Perform Step b multiple times until the maximum number of iterations is reached and the optimal parameters are output.

(d) Taking the optimal parameters as the values of λ and Δ in the network G, and bringing the parameters into original stochastic competitive learning algorithm to output the result of community detection.

(e) The algorithm finishes.

When searching for optimal parameters by the method of PSO, the range of values of parameters λ and Δ can be determined according to the verification analysis in reference [Shi and Eberhart (1998)]. It can be got from reference [Silva and Zhao (2016)] that the values of the parameters λ and Δ are respectively [0.2, 0.8], [0.1, 0.4]. Therefore, the changes of parameter λ when λ was brought into PSO for parameter optimization is as shown in the Eq. (7).

$$\lambda_i^{t+1} = \lambda_i^t + w^t v_{i\lambda}^t + c_1 r_1 (p_{i\lambda}^t - \lambda_i^t) + c_2 r_2 (p_{g\lambda}^t - \lambda_i^t), \lambda \in [0.2, 0.8] \tag{7}$$

In Eq. (7), λ_i^{t+1} is the location of the particle optimization parameter λ for the next iteration. The changes of parameter Δ when Δ was brought into PSO for parameter optimization is shown in Eq. (8).

$$\Delta_i^{t+1} = \Delta_i^t + w^t v_{i\Delta}^t + c_1 r_1 (p_{i\Delta}^t - \Delta_i^t) + c_2 r_2 (p_{g\Delta}^t - \Delta_i^t), \Delta \in [0.1, 0.4] \tag{8}$$

Similar to the Eq. (7), Δ_i^{t+1} is the location of the particle optimization parameter Δ for the next iteration.

2.2.3 Particle domination ability self-adaptive

There are often differences in the size of different communities in a social network [Liu (2008)]. In stochastic competitive learning, each particle has a domination ability value that is constant for each particle, so each particle has the same domination ability in the process of competing. This situation would result in a result that the size of the community dominated by each particle tends to be the same. However, due to the differences in size between different communities, the results of dynamic community detection are often not ideal. As shown in Fig. 1, particles that control the community on the left are likely to walk in the marginal area of the right community. At the same time, a large number of nodes dominated by the particles of the community on the right would

cause the particles have not enough walking in their edge area. Therefore, the edge area of the right community will be encroached by the left community.

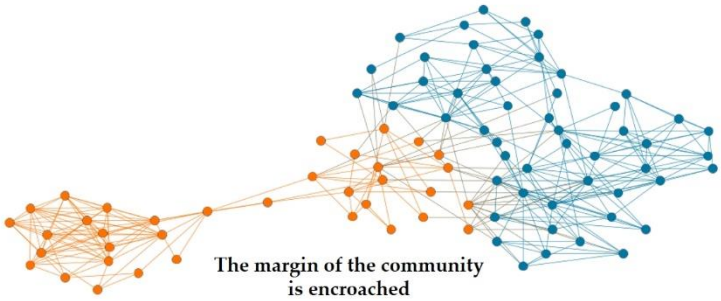

Figure 1: Schematic diagram of community detection when the particle domination ability value is constant

To solve the above problem, the domination ability value needs to be self-adaptive. The domination ability parameter $\varepsilon=1$ in the original algorithm. Suppose that the particle k moves to the node i at time $t+1$, and the domination ability of the particle k to the node i before the time t is the element corresponding to k, i in the control matrix, that is $\overline{\mathrm{N}}_i^{(k)}$.The change of $\overline{\mathrm{N}}_i^{(k)}$ is shown in Eq. (9).

$$\overline{\mathrm{N}}_i^{(k)}(t+1)=\overline{\mathrm{N}}_i^{(k)}(t)+1 \tag{9}$$

The algorithm after particle domination ability self-adaptive no longer directly uses the times of the particle visits the node as the domination ability of the particle to the node. Rather, the percentage of the community size dominated by the particle in the network is taken as the domination ability value for that node each time the particle visits the node. Assuming that the particle k moves to the node i at time $t+1$, the change of the corresponding domination element of the node i and the particle k in the control matrix is as shown in the Eq. (10). In the equation, $|\mathrm{Belong}^{(k)}(t)|$ and $|V|$ represent the size of $\mathrm{Belong}^{(k)}(t)$ and $|V|$ respectively.

$$\overline{\mathrm{N}}_i^{(k)}(t+1)=\overline{\mathrm{N}}_i^{(k)}(t)+\frac{|\mathrm{Belong}^{(k)}(t)|}{|V|} \tag{10}$$

The particle domination ability after self-adaptive is proportional to the current domination size of the particle. Hence, each particle tends to maintain its own domination size during random-preferential walks. The improved algorithm could solve the situation that the small-size community shown in Fig. 1 would encroach on the marginal nodes of large-size community. The results after the improvement are shown in Fig. 2.

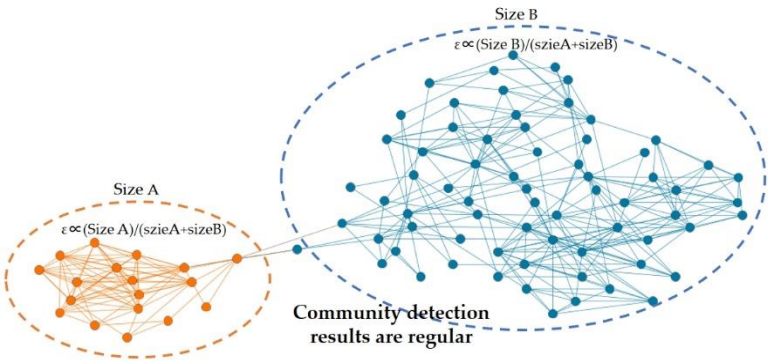

Figure 2: Schematic diagram of community detection when particle domination ability value is self-adaptive

It can be seen from Fig. 2 that the self-adaptive domination ability of particles can prevent small-size communities from encroaching on large-size communities' edge nodes. In social networks, this improvement is necessary for stochastic competitive learning algorithms because the size of different communities often varies.

3 Results and discussion

3.1 Data analysis

The data set used in this experiment is an open source social network dataset provided by the SNAP project team [SNAP (2012)]. The reason for choosing this data is that the SNAP data set has been widely used, so the reliability of the data can be guaranteed. The data set is introduced as shown in Tab. 1. In terms of evaluation methods, this experiment mainly uses F1 score and Balanced Error Rate (BER) as the algorithm accuracy evaluation index. An F1 score reaches its best value at 1 and worst at 0. On the contrary, a BER score reaches its best value at 0 and worst at 1.

Table 1: Introduction of data in the experiment

Number	Name	Number of edges	Number of nodes	Number of connected components
1	0	2519	333	5
2	348	3192	224	1
3	3437	4813	534	2
4	356963	495	126	2
5	428333	929	65	1
6	612473	934	76	1
7	778446	2174	187	1
8	779715	2564	183	1
9	1367531	6681	239	1
10	1548841	2159	179	1

11	2363991	3430	214	2
12	2589521	2414	172	1
13	5506012	549	64	1
14	5747502	539	87	2
15	7424642	391	34	1
16	7670202	1318	98	1
17	7861312	3443	220	1
18	7888452	934	120	2
19	9298152	1647	182	1
20	9524062	2204	143	1
21	11548841	2159	179	1
22	12029971	1936	143	1
23	12771872	1290	130	2
24	13275962	2446	143	2
25	14060856	1764	186	1
26	14120253	759	141	2
27	14528221	1118	114	1
28	14618160	427	75	2
29	14836915	4682	196	1
30	15747502	549	64	1

As shown in Tab. 1, the “Number” column is the number of the data set, the “Name” column is the name of the different data set, and the next few columns are different attributes of the corresponding data set.

3.2 Several methods of algorithm improvement experiment

On the basis of the original stochastic competitive learning, the experiments of the improved algorithm proposed in the above sections are respectively carried out. The comparison experiment is based on the value of parameters proposed in reference [Silva and Zhao (2016)]. The ratio of preferential walk in random-preferential walk named parameter λ and the particle energy change parameter Δ are set as 0.5, 0.3 respectively. The particle domination capability value ε is set as 1. The F1 score was used as an evaluation index in each group of comparative experiments. The accuracy of the original algorithm after thirty sets of data sets was evaluated using the F1 score, with an average of 0.576.

In the improved experiment of particle position initialization, the average of F1 score was 0.598. The results of the two algorithms before the improvement and that after the improvement were shown in Fig. 3.

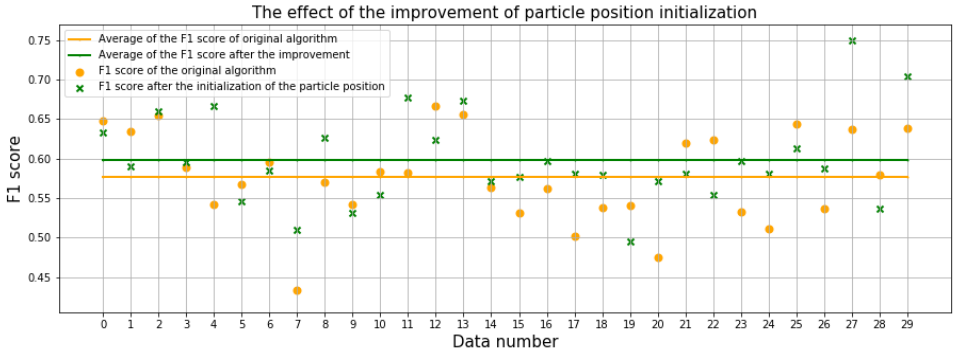

Figure 3: Schematic diagram of the effect of the improvement of particle position initialization

In Fig. 3, the F1 score after the initialization of the particle position is the green cross, and the average results of the algorithm after the particle position initialization is represented by the green and x-axis horizontal lines. The average value of the particle position after initialization is 3.84% higher than that before the improvement. Hence, this improvement is helpful to improve the accuracy of the results of community detection in stochastic competitive learning. In the improved experiment based on particle swarm optimization, the average of F1 score was 0.616. The results of the algorithms before the improvement based on particle swarm optimization and the results after that were shown in Fig. 4.

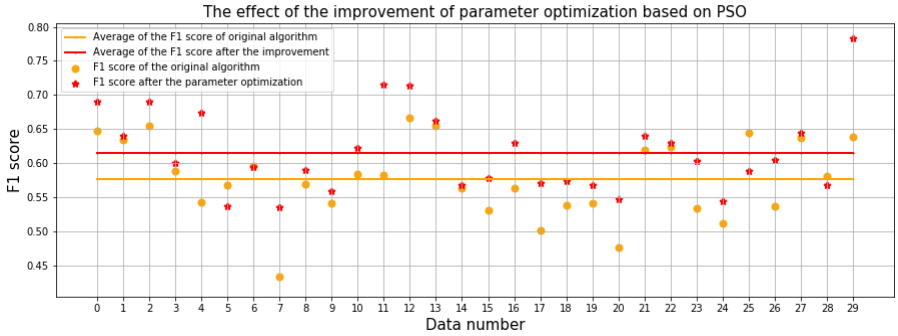

Figure 4: Schematic diagram of the effect of the improvement of parameter optimization

In Fig. 4, the F1 score after the parameter optimization is the red "*", and the average results of the algorithm after the improvement of PSO is represented by the red and x-axis horizontal lines. The average value of the F1 score after the improvement of parameter optimization is 6.84% higher than that before the improvement. Hence, this improvement is helpful to improve the accuracy of the results of community detection in stochastic competitive learning. In the improved experiment based on particle domination ability self-adaptive, the average of F1 score was 0.614. The results of the algorithms before particle domination ability self-adaptive and the results after that were shown in Fig. 5.

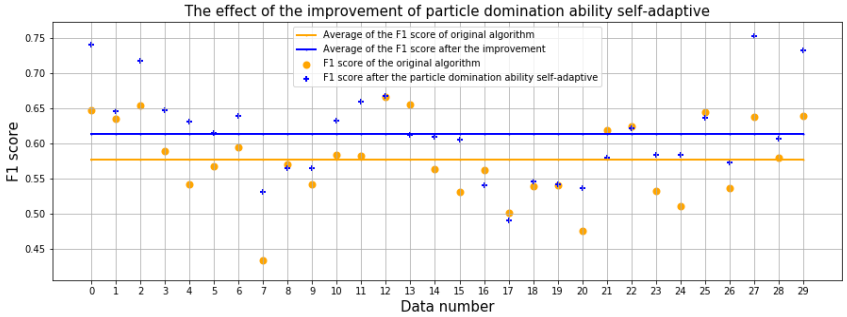

Figure 5: Schematic diagram of the effect of the improvement of particle domination ability self-adaptive

In Fig. 5, the F1 score after the particle domination ability self-adaptive is the blue "+", and the average results of the algorithm after particle domination ability self-adaptive is represented by the blue and x-axis horizontal lines. The average value of the improved algorithm is 6.56% higher than that before the improvement. Hence, this improvement is helpful to improve the accuracy of the results of community detection in stochastic competitive learning.

3.3 Community detection experiments based on improved stochastic competitive learning

Based on the features of social networks, the improved stochastic competitive learning can be obtained by combining the above three improvement methods. The improved stochastic competitive learning needs to use the particle swarm optimization algorithm to search the optimal parameters λ and Δ firstly. The position of particle initialization is then determined according to Algorithm 1. Finally, in the particle walk of stochastic competitive learning, each particle adopts a self-adaptive domination ability value. The framework of the improved algorithm was shown in Fig. 6.

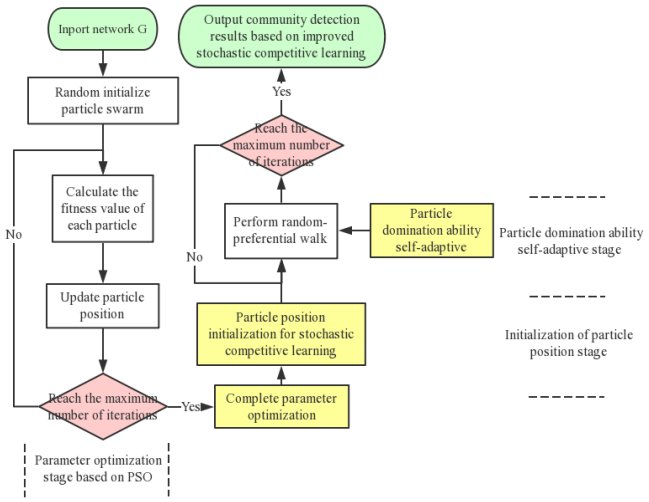

Figure 6: Framework of improved stochastic competitive learning

The obtained social network data is brought into the improved stochastic competitive learning to obtain comparison results as shown in Tab. 2. The "Number" column represents the number of different data sets, the "Name" represents the name of different data sets, and the "Average" row represents the average of the scores of different evaluation index. The effect of verifying the algorithm by averaging the results of the thirty sets of data sets could avoid the contingency of the experimental results as much as possible.

Table 2: The effect of improved stochastic competitive learning

Number	Name	F1 score	BER score
1	0	0.6802	0.6324
2	348	0.5992	0.6346
3	3437	0.7158	0.5857
4	356963	0.6377	0.5738
5	5506012	0.7450	0.4658
6	428333	0.6875	0.4588
7	612473	0.6004	0.6142
8	778446	0.6458	0.5830
9	779715	0.6099	0.6260
10	1367531	0.5614	0.5221
11	1548841	0.6067	0.5925
12	2363991	0.6016	0.5564
13	2589521	0.5939	0.5411
14	5747502	0.7450	0.4658
15	7424642	0.6742	0.4751
16	7670202	0.5885	0.6064
17	7861312	0.5635	0.5979
18	7888452	0.5747	0.6018
19	9298152	0.6388	0.5953
20	9524062	0.5705	0.5734
21	11548841	0.6251	0.5290
22	12029971	0.6589	0.5395
23	12771872	0.6120	0.5806
24	13275962	0.6043	0.5671
25	14060856	0.5386	0.5993
26	14120253	0.6567	0.5683
27	14528221	0.6385	0.6057
28	14618160	0.5817	0.5739
29	14836915	0.6285	0.5598
30	15747502	0.6646	0.4732
	Average	0.6283	0.5633

It can be got from Tab. 2 that the F1 score of the thirty sets of data sets using the improved stochastic competitive learning for community detection is 0.628, which is 9.07% higher than the original algorithm result.

4 Conclusion

This study introduces the existing stochastic competitive learning and analyzes its shortcomings in social network at the beginning of the paper. Then, it points out that this algorithm can realize time series community detection which is unique feature. Therefore, it is meaningful to improve this algorithm.

Based on the features and data of social networks, three improved methods were proposed in this study. Firstly, the particle position initialization method in stochastic competitive learning is set, so that stochastic competitive learning can reach the convergence state as soon as possible. Secondly, based on the particle swarm optimization algorithm, two parameters in stochastic competitive learning are optimized. Finally, the self-adaptation of the particle control ability value is realized, and the phenomenon that the community detection result is not ideal due to the out of bounds of the particle is prevented.

In the experimental, the author tests the three improved methods with the open source social network data provided by SNAP, and finds that each improvement method improves the accuracy of community detection. Finally, the three improvement methods are combined into improvement. The stochastic competitive learning algorithm is compared with the original stochastic competitive learning algorithm. The F1 score was used for evaluation. The experimental results show that the improved stochastic competitive learning algorithm is 9.07% more accurate than the original algorithm. On the one hand, the artificial designation of parameters' value is not necessary in the improved stochastic competitive learning proposed in this study and enhances the practical significance. On the other hand, the accuracy is greatly improved compared with the original algorithm.

Acknowledgement: This research was funded by National Natural Science Foundation of China (Grant No. 2017YFC0820100). The authors sincerely acknowledge the reviewers for their suggestions which helped in improving the quality of the paper.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Bacciu, D.; Starita, A. (2008): Competitive repetition suppression (CoRe) clustering: a biologically inspired learning model with application to robust clustering. *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1922-1941.

Blondel, V. D.; Guillaume, J. L.; Lambiotte, R.; Lefebvre, E. (2008): Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, vol. 5, no. 10.

Fu, S.; Gu, Y.; Zhang, D. (2017): Overlapping community detection algorithm based on improved MCMC method. *Netinfo Security*, vol. 17, no. 9, pp. 138-142.

Kennedy, J; Eberhart, R. (2002): Particle swarm optimization. *ICNN95-International Conference on Neural Networks*, pp. 760-766.

Li, Q.; Zhang, C.; Chen, P. (2013): Improved ant colony optimization algorithm based on particle swarm optimization. *Control & Decision*, vol. 28, no. 6, pp. 873-872.

Liu, D.; Pang, Z.; Lloyd, S. R. (2008): A neural network method for detection of obstructive sleep apnea and narcolepsy based on pupil size and EEG. *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 308-318.

Liu, Y. (2008): *Research on Social Network Structure (Ph.D. Thesis)*. Zhejiang University, Hangzhou.

McAuley, J.; Leskovec, J. (2014): Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 1, pp. 1-28.

Moon, S.; Lee, J. G.; Kang, M.; Choy, M.; Lee, J. W. (2015): Parallel community detection on large graphs with MapReduce and GraphChi. *Data & Knowledge Engineering*, vol. 104, no. 5, pp. 17-31.

Pan, X.; Deng, G.; Dong, B. (2011): Construction and analysis of the opinion formation model based on the social network. *Operations Research and Management Science*, vol. 20, no. 2, pp. 176-179.

Quiles, M. G.; Zhao, L.; Alonso, R. L.; Romero, R. A. (2008): Particle competition for complex network community detection. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 18, no. 3.

Shi, Y.; Eberhart, R. (1998): A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*, pp. 69-73.

Silva, T. C.; Zhao, L. (2012): Stochastic competitive learning in complex networks. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 385-398.

Silva, T. C.; Zhao, L. (2013): Uncovering overlapping cluster structures via stochastic competitive learning. *Information Sciences*, vol. 247, no. 6, pp. 40-61.

Silva, T. C.; Zhao, L. (2016): *Machine Learning in Complex Networks*. Switzerland: Springer.

SNAP. (2019): *Stanford Network Analysis Project. Stanford Large Network Dataset Collection*. http://snap.stanford.edu/data/index.html.

Zhao, Y.; Yuan, Y.; Nie, F.; Wang, Q. (2018): Spectral clustering based on iterative optimization for large-scale and high-dimensional data. *Neurocomputing*, vol. 318, pp. 227-235.

Appendix

The code and data involved in this paper are here:

https://github.com/youguqiaomu/scl_social_network.