

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

A Novel Optimization Algorithm Integrating Immunity Clone and Differential Evolution for Parameter Selection of SVM

^{1,2,3}Li Jun, ^{1,2}Ding Lixin and ³Zou Qilin

¹State Key Laboratory of Software Engineering, Wuhan University, Wuhan, 430072, China

²School of Computer, Wuhan University, Wuhan, China

³College of Computer Science and Technology, Wuhan University of Science and Technology, WUST, Wuhan, China

Abstract: The classification accuracy of Support Vector Machine (SVM) depends on parameters strongly. In nature, parameter selection is a search optimization process. A Differential Evolution (DE) algorithm is a real-coding optimal algorithm based on swarm evolution. It has powerful global searching ability. But it gets into premature convergence easily. So a novel hybrid optimization algorithm based on Immunity Clone (IC) and differential evolution is proposed for parameter selection of SVM. In this algorithm, clonal selection and receptor edit mechanism are inserted into the differential evolution process. Thirteen experimental results on UCI datasets distinctly show that compared with default parameters SVM classifier, the differential evolution algorithm, the proposed algorithm has higher classification accuracy.

Key words: Support vector machines, differential evolution algorithm, immunity clone algorithm, clone selection, parameter optimization

INTRODUCTION

Support Vector Machine (SVM) has been successfully applied in many fields such as classification and regression (Vapnik, 2000). In the research work to improve the learning ability and generalization performance of SVM, an important question is how to optimize the parameters of the support vector machine SVM to improve its classification accuracy.

In the related research, the grid search proposed in the literature (Hsu *et al.*, 2003) is an alternative search method that is simple and easy to use, but its search ability is poor. Genetic algorithm proposed in (Zheng and Jiao, 2004) and particle swarm optimization algorithm proposed in (Shao *et al.*, 2006) are for optimization selection of SVM parameters respectively, although these intelligent methods reduce the reliance on initial value selection, yet the algorithm principles and the thought are more complex and they are easy to fall into local optimum. Differential evolution algorithm is a kind of real number encoding optimization algorithm based on population evolution (Storn and Price, 1997; Qin *et al.*, 2009). The algorithm can extract different information from the current population and guide the further search. The principles are relatively simple and its algorithm control parameters is relatively less, while the global search ability is far stronger. But it is easy to fall into local optimum; the

phenomenon of premature convergence also exists. In order to overcome defects resulted from using the DE algorithm to solve the problem global optimization of SVM parameters, there introduces clonal selection and receptor editing mechanism (Cheng *et al.*, 2012; De Castro and von Zuben, 2000) in the differential evolution process, advances the diversity of population distribution and enhance the capability of global optimization of population

The remainder of this study is organized as follows. The basic principle of the Support Vector Machine (SVM) is given in section 2. Section 3 introduces the differential evolution algorithm. Section 4 presents the immune clone algorithm. Section 5 elaborates the proposed SVM parameter selection method based on immune clone and differential evolution algorithm. Section 6 gives the experimental result of comparison of the proposed method with the two other methods. Section 7 summaries this study.

SUPPORT VECTOR MACHINE

SVM separates the classes with a decision surface that maximizes the margin between the classes. The surface is often called the optimal hyperplane and the data points closest to the hyperplane are called support vectors.

Provided sample pairs as $\{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\} \in (X \times Y)^l$, among them, $x_i \in X \subset R^n$ is the input vector, $y_i = \{-1, 1\}$ is the corresponding output values for x_i , $i = 1, 2, \dots, l$.

When the sample points approximately linear classification, the problem boils down to the following optimization problem:

$$\begin{aligned} \min & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \zeta_i \\ \text{s.t. } & y_i((w \cdot x_i) + b) + \zeta_i \geq 1 (\zeta_i \geq 0, i=1, 2, \dots, l) \end{aligned} \quad (1)$$

Among this, $\|w\|^2 [0, 1]$ is called structural risk which is on behalf of the complexity of the model and make the function smoother, thereby can improve the generalization ability:

$$C \sum_{i=1}^l \zeta_i$$

is known as the empirical risk which on behalf of the error of the model; and C is known as punishment parameter.

When the sample points are shown to exhibit non-linear relationship, the original sample set will be mapped to the high dimensional feature space through a nonlinear mapping $\varphi(x)$, then make the linear classification in the high-dimensional feature space. The inner product operation on the high dimensional feature space can be defined as kernel function: $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$, only need to use the function operate the variables in the original low dimension.

Using the duality theory in the optimization theory can get the final decision function:

$$f(x) = \text{sgn} \left(\sum_{i=1}^l y_i \alpha_i K(x_i, x) + b^* \right) \quad (2)$$

Several kernel functions help the SVM obtain the optimal solution. The most frequently used such kernel functions are the polynomial, sigmoid and radial basis kernel function (RBF). That is:

$$K(x, y) = e^{-\frac{\|x-y\|^2}{\sigma^2}}$$

or

$$K(x, y) = e^{-\delta \|x-y\|^2} \quad (3)$$

Generally the RBF is applied most frequently, because it can classify high-dimensional data, unlike the linear kernel function. Additionally, the RBF has fewer parameters to set than a polynomial kernel. Therefore, this study applies

an RBF kernel function in the SVM to obtain optimal solution. The major parameters that affect the performance of support vector machine are punished parameter C and RBF kernel parameter δ . They must be set appropriately.

DIFFERENTIAL EVOLUTION ALGORITHM

Differential evolution algorithm can be used to solve the global optimization problems which have N continuous variables. In the population initialization phase, individuals in a population with an average random initial space in the search space. In the evolutionary stage, primarily through mutation, crossover and selection process until the stop condition is met. Three control parameters of original differential evolution algorithm, they are respectively, population size NP , scaling factor F and the probability of crossover CR .

The differential mutation: In the differential evolution algorithm, mutation operator is one of the most important operators. $DE/rand/2$ is used in this study:

$$V_i = X_{r_1} + F(X_{r_2} - X_{r_3}) + F(X_{r_4} - X_{r_5}) \quad (4)$$

Among this, X_{best} is the best individual in the current population, X_i is a Father individual, $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$ are five individuals randomly selected from population, V_i is a variation vector, $X_{r_2} - X_{r_3}$ is a difference vector and $F \in [0, 1+)$ is a zoom factor which used for the difference vector to zoom in, thus to control the search step length.

Crossover operation

Differential evolution algorithm use discrete crossover operator, including the Binomial Crossover and Exponential Crossover. Crossover operator let Variable vector V_i generated by mutation operator and father individual vector X_i do the discrete crossover and acquire the try vector U_i . Binomial Crossover operator can also be expressed as:

$$U_i(j) = \begin{cases} V_i(j) & \text{if } (\text{rndreal}, [0,1] < CR \text{ or } j = j_{\text{rand}}) \\ X_i(j) & \text{otherwise} \end{cases} \quad (5)$$

Among this, $j = 1, \dots, D$; j_{rand} is a random integer in the range $[1, D]$ which ensures that at least one dimension of the attempt vector U_i from the variation vector V_i , thus to avoid the same as the parent individual vector X_i . In each mutation operator of differential evolution algorithm can be combined with index of crossover operator.

Select operation: Through mutation operator and crossover operator, differential evolution algorithm generates subgroup, there adopt the one-to-one selection operator to compare individual to corresponding parent

individual, then the excellent individual can be saved to the next group. Optimization for minimizing the selection operator can be described as:

$$X_i = \begin{cases} U_i & \text{if}(f(U_i) \leq f(X_i)) \\ X_i & \text{otherwise} \end{cases} \quad (6)$$

Among this, $f(X_i)$ is the adaptive value for X_i .

The selection operation of DE algorithm can increase the convergence rate of the algorithm, but the difference among individuals getting smaller and smaller, resulted that population easily converge at one point and the last population in the solution space can't be re-search, so it is easy to fall into local optimum and appear premature convergence phenomenon. In order to guarantee the diversity of individuals, the study introduces the immune clone algorithm.

IMMUNITY CLONE ALGORIRHM

Immunity clone can be dated back to the research in the biology immune system area. There are clone, hypermutation, combination of antibody and antigen, generating of memory cell and other process in the clonal selection mechanism. This mechanism can not only ensure the convergence rate, but also could maintain the diversity of antibodies.

The general algorithm is given as follows:

- **Step 1:** Initialize candidate possible solutions according to the problem to be solved
- **Step 2:** Calculate echo individual Affinities and extract the n antigens with high individual affinity
- **Step 3:** Clone these n-best individuals extracted to generate a temporary clone population C, the number of clones of antibody-antigen affinity proportional the number of cloning:

$$N_c = \sum_{i=1}^n \text{round}(\frac{\beta * n}{i} + b) \quad (7)$$

($\beta \in (0,1)$, β is the coefficient cloning, n is population size, b is a constant coefficient, a minimum number of cloning, Assume that $b = 2$)

- **Step 4:** Hypermutate individuals in the temporary clone population C to obtain population V, the degree of variation is inversely proportional to individual affinity (fitness) the formula as follows:

$$x_{id}^{g+1} = x_{id}^g + \alpha * \eta * x_{id}^g * \text{rand}(0,1) - \alpha * \eta * x_{id}^g * \text{rand}(0,1) \quad (8)$$

$$\alpha = \begin{cases} 1, \text{rand}(0,1) \leq P_m \\ 0, \text{else} \end{cases}$$

Rand (0, 1) is a random number between 0 and 1, $P_m = 0.5$:

$$\eta(t) = 1 - r^{[1 - \frac{g}{GEN}]^b} \quad (9)$$

b is a constant which represents the index variation of the control system of space, $r \in (0,1)$, represents the base variation of the control system of space, In the early evolution, when r takes a smaller value, $\eta(t) \sim 1$, the variation of range become bigger; rather late in the evolution, g is close to GEN, local search is limited in the small range

- **Step 5:** Receptor editing (Ge *et al.*, 2008), extract individuals which have the best degree after antigen stimulation in the mutated population and the remaining will be replaced by these individuals receptor editing formula as follows:

$$X_{id}^g = X_{id}^g + \frac{X_{id}^{max} - X_{id}^{min}}{m} + U_{r+1} \quad (10)$$

$X_{id}^{max}, X_{id}^{min}$ is the upper and lower bounds of the location, m is the number of normal which represents individual position correct scaling factor, U_{r+1} is the Logistic chaotic sequence and U_{r+1} sequence is as follows:

$$U_{r+1} = u * U_r * (1 - U_r), r = 0, 1, 2, \dots, 0 < U_0 < 1 \quad (11)$$

u is the state control parameters, when u equals 4, initial value of U_0 is $U_0 \in (0.25, 0.5, 0.75, 1)$, the system is fully represented on the chaotic state. U_{r+1} will traverse from 0 to 1

- **Step 6:** Replace the original antibody population with the antibodies extracted, individuals with low affinity to be eliminated

Cloning of biology immune mechanism system selectively generates the most reactive with the antigen antibody clone variation; thereby generate antibody diversity in the sample. Under this guiding ideology, the immune algorithm avoids falling into local extremum to the maximum extent possible and has a fast convergence to the global optimum.

DE-CLONE-SVM

This research proposes a novel hybrid optimization algorithm based on immunity clone and differential evolution, denoted as de-clone-svm model, to obtain the optimal parameter settings for SVM kernel parameters which result in a better classification accuracy rate.

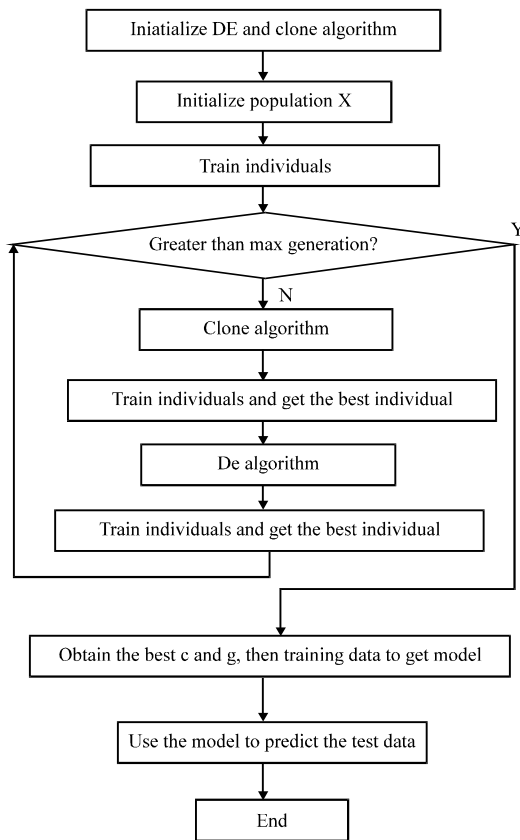


Fig. 1: The flow chart of the de-clone-svm model

So the research uses the SVM classification accuracy as fitness function.

In optimization algorithm based on differential evolution (denoted as de-svm model), the difference among individuals getting smaller and smaller, resulted that population easily converge at one point. But clone, mutation, selection, suppression and replacement can be executed on each individual with Clone Algorithm, then these individual cross-validation accuracy obtained by svm training determine whether an individual into a local solution, if individual trapped in local solutions with high fitness clones from particle to be updated its suppression and replacement, thus escape from the local extremum interval.

Figure 1 shows the flow chart of the developed de-clone-svm model.

The outline of the proposed algorithm lists as follows:

- **Step 1:** Initialize population size NP, population X and algorithm parameters
- **Step 2:** Train individuals in the population X with svm to get each individual cross-validation accuracy,

then find the best cross-validation accuracy BestAccuracy and its corresponding c and g, so Bestc = c, Bestg = g

- **Step 3:** Determine whether the current accuracy and number of iterations reaches the end of the condition, if reached, go to step 14, otherwise the next step into the next generation iterative process
- **Step 4:** Extract P_m (such as P_m = X*1/4) optimal individuals from the population X
- **Step 5:** Clone P_m individuals to generate temporary populations C
- **Step 6:** Hypermutate individuals in the population C to obtain populations V, the degree of variation is inversely proportional to individual fitness (individual cross-validation accuracy)
- **Step 7:** If g(the current algebra) divides n(a fixed constant, such as n = 10), perform a receptor editing on inactive individuals in the population V, in order to improve their fitness (i.e., cross-validation accuracy), to obtain population S. Otherwise go to the next step
- **Step 8:** Train individuals in the population X with svm to get each individual cross-validation accuracy. then find the best cross-validation accuracy temp_BestAccuracy and its corresponding temp_c and temp_g, If BestAccuracy < temp_BestAccuracy, then BestAccuracy = temp_BestAccuracy, Bestc = temp_c, Bestg = temp_g
- **Step 9:** Extract NP optimal individuals from the population S to form population X
- **Step 10:** Execute differential mutation in the population X to obtain population Y
- **Step 11:** Execute crossover operation in the population Y to obtain populations Z
- **Step 12:** Train individuals in the population Z with svm to get each individual cross-validation accuracy, then find the best cross-validation accuracy temp_BestAccuracy and its corresponding temp_c and temp_g, If BestAccuracy < temp_BestAccuracy, then BestAccuracy = temp_BestAccuracy, Bestc = temp_c, Bestg = temp_g
- **Step 13:** Execute choose operation in the population Z, the offspring population in Z and X in the parent population individuals compete to select the fitness of individuals for outstanding replace X. Go to step 3
- **Step 14:** Utilize Bestc and Bestg to train the training data set to obtain learning model, use the model to predict the test data set to get the prediction accuracy of the test data set

EXPERIMENT RESULTS

The platform adopted to develop the DE-Clone-SVM approach is Dell desktop with the following features: Intel Pentium Cores duo CPU E7200 2.53 GHz main frequency, 2GMB RAM, a Windows XP sp3 operating system. Our implementation was carried out on the Matlab2010a development environment by extending the LIBSVM which is originally designed by Chang and Lin (2001).

In order to measure the performance of the developed DE-Clone-SVM approach, the following datasets in UC Irvine Machine Learning Repository[11] are used: balance, diabets, glass, haberman, iris, kr_V_kp, liver, parkinsons, sonar_all_data, splice, wdbc, wine, zoo (Hettich *et al.*, 1998). Table 1 presents the properties of these datasets.

To guarantee valid results for making predictions regarding new data, the dataset is further randomly partitioned into training sets and independent test sets via a k-fold cross validation. Each of the k subsets acts as an independent holdout test set for the model trained with the remaining k-1 subsets. The advantages of cross validation are that all of the test sets were independent and the reliability of the results could be improved. This study used k = 10. Each of the 10 subset is used as test data sets in turn, so the program runs 10 times. The final classification accuracy is expressed in the form “mean±standard deviation”.

The searching range of parameters C and δ of SVM is between 10^{-7} and 10^7 . In the de-clone-algorithm, the parameters are initially set as follows. Iterations GEN is 200, population size NP is 40, scaling factor F is 0.8, crossover probability CR is 0.5; Clone proportion P_m is 0.5, index of spatial variation b is 2, base of spatial variation r is 0.6, individual position correct scaling factor m is 20.

To verify the excellence of DE-Clone-SVM for parameters optimization, we use the approach for comparing the performance of default parameter SVM classifier algorithms and DE algorithm. The experimental result is shown in Table 2.

We can distinctly see that in most of the datasets, the mean test accuracy of the de-clone-svm algorithm is better than the original svm and the de-svm and the standard deviation of the de-clone-svm algorithm is smaller.

On the classification accuracy, the de-clone-svm algorithm is better than svm in 12 datasets with only one a little less than it. Compared to the de-svm algorithm, de-clone-svm is better in 10 datasets, with one more or less the same and two less than it.

On the standard deviation, de-clone-svm has less variance than svm in 12 datasets, with only one a little

Table 1: Datasets from the UCI repository

Dataset	No. of instances	No. of features	No. of classes
balance	625	4	3
diabets	768	8	2
glass	214	9	6
haberman	306	3	2
Iris	150	4	3
kr_V_kp	3196	36	2
liver	327	6	2
parkinsons	195	22	2
sonar_all_data	208	60	2
splice	1000	60	2
wdbc	569	30	2
wine	178	12	3
zoo	101	16	7

Table 2: Experimental results

	svm (%)	de-svm (%)	de-clone-svm (%)
balance	98.71±2.12	99.05±2.01	99.68±0.67
diabets	76.32±3.34	77.27±2.47	80.91±1.38
glass	45.24±9.05	54.09±7.86	54.54±10.71
haberman	68.00±8.63	83.87±4.56	83.87±0.00
iris	92.00±13.62	100.00±0.00	96.25±3.23
kr_V_kp	99.40±0.54	98.78±0.93	99.38±0.00
liver	66.88±11.04	52.43±6.24	76.97±6.42
parkinsons	82.63±11.91	85.00±4.71	90.50±7.25
sonar_all_data	84.00±9.37	93.34±2.46	91.43±2.01
splice	88.10±1.52	82.68±1.34	91.09±0.00
wdbc	87.68±8.89	88.24±3.51	94.92±0.55
wine	87.06±14.36	93.89±9.95	94.71±5.85
zoo	93.00±8.23	90.91±0.00	100.00±0.00

higher than it. De-svm-clone has less variance than de-svm in 8 datasets, with one almost the same and 4 a little higher than it. The smaller the standard deviation, the smaller the rise and fall of the solution will be. It means that the solution of de-clone-svm is more stable than others.

The obtained results clearly confirm the superiority of the DE-clone-SVM algorithm compared to default parameters SVM classifier and the de-svm algorithm.

CONCLUSION

In this study, aiming at the shortcomings of differential evolution based parameters selection for support vector machine, combined with the clonal selection algorithm, a novel hybrid optimization algorithm based on immune clone and differential evolution is proposed. With clonal selection and receptor editing, the algorithm improves the diversity of population and avoids premature convergence. Simulation results show that the proposed algorithm greatly enhances classification accuracy of SVM.

AKNOWLEDGMENT

This research was supported by National Natural Science Foundation of China under Grant No. 61273303.

REFERENCES

- Chang, C.C. and C.J. Lin, 2001. LIBSVM: A library for support vector machines. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- Cheng, L., Y. Ding, K. Hao and Y. Hua, 2012. An ensemble kernel classifier with immune clonal selection algorithm for automatic discriminant of primary open-angle glaucoma. *Neurocomputing*, 83: 1-11.
- De Castro, L.N. and F.J. von Zuben, 2000. The clonal selection algorithm with engineering applications. *Proceedings of the Conference on Genetic and Evolutionary Computation, Workshop on Artificial Immune Systems and their Applications*, July 2000, Las Vegas, USA., pp: 36-37.
- Ge, H.W., L. Sun, Y.C. Liang and F. Qian, 2008. An effective PSO and AIS-based hybrid intelligent algorithm for job-shop scheduling. *IEEE Trans. Man Cybernetics*, 38: 358-368.
- Hettich, S., C.L. Blake and C.J. Merz, 1998. UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, CA.
- Hsu, C.W., C.C. Chang and C.J. Lin, 2003. A practical guide to support vector classification. Technical Report, University of National Taiwan, Department of Computer Science and Information Engineering, July, 2003. <http://www.csie.ntu.edu.tw/~cjlin/studys/guide/guide.pdf>
- Qin, A.K., V.L. Huang and P.N. Suganthan, 2009. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evolut. Comput.*, 13: 398-417.
- Shao, X.G., H.Z. Yang and G. Chen, 2006. Parameters selection and application of support vector machines based on particle swarm optimization algorithm. *Control Theory Appl.*, 23: 740-743.
- Storn, R. and K. Price, 1997. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.*, 11: 341-359.
- Vapnik, V.N., 2000. *The Nature of Statistical Learning Theory*. Tsinghua University Press, Beijing, China.
- Zheng, C. and L. Jiao, 2004. Automatic parameters selection for SVM based on GA. *Proceedings of the 5th World Congress on Intelligent Control and Automation*, Vol. 2, June 15-19, 2004, Piscataway, NJ., pp: 1869-1872.