Advanced Materials Research Vols. 255-260 (2011) pp 2082-2086
Online available since 2011/May/31 at www.scientific.net

doi:10.4028/www.scientific.net/AMR.255-260.2082

Colonial Competitive Algorithm Assisted Least Squares Support Vector Machines

Yuan Ren[a] and Guangchen Bai[b]

School of Jet Propulsion, Beijing University of Aeronautics and Astronautics, Beijing, China

[a]renyuan116@sina.com, [b]dlxbgc@buaa.edu.cn

Key words: colonial competitive algorithm, parameters tuning, machine learning, optimization

Abstract. The use of least squares support vector machine (LSSVM), a novel machine learning method, for classification and function approximation has increased over the past few years especially due to its high generalization performance. However, LSSVM is plagued by the drawback that the hyper-parameters, which largely determine the quality of LSSVM models, have to be defined by the user, and this increases the difficulty of applying LSSVM and limits its use on academic and industrial platforms. In this paper we present a novel method of automatically tuning the hyper-parameters of LSSVM based on colonial competitive algorithm (CCA), a newly developed evolutionary algorithm inspired by imperialistic competition mechanism. To show the efficacy of the CCA assisted LSSVM methodology, we have tested it on several benchmark examples. The study suggests that proposed paradigm can be a competitive and powerful tool for classification and function approximation.

Introduction

As a machine learning method, support vector machine (SVM) originally introduced by Vapnik within the area of statistical theory and structural risk minimization has emerged as a powerful tool for data analysis [1]. While classical neural networks approaches, such as multilayer perceptrons (MLP) and radial basis function networks (RBNN) suffer from problems like the existence of local minima and the choice of the number of hidden neurons, the regression function of SVM is related to a convex quadratic programming (QP) problem whose solution is global and in general unique. The formulation of QP problem in primal space results into inequality constraints, which derive from the ε-insensitive loss function adopted by SVM.

In the late 1990s, a modified version of SVM, i.e. the least squares support vector machine (LSSVM), was developed by Suykens and Vandewalle [2]. The formulation of LSSVM considers equality constraints and a sum squared error cost function. This reformulation facilitates solution generation by Karush-Kuhn-Tucker system which takes a similar form as the linear system that one solves in every iteration step by interior point methods for standard SVMs. This linear system can be efficiently solved by iterative methods such as conjugate gradient.

LSSVM inherits excellent generalization ability and small sample learning ability from SVM, and meanwhile overcomes its drawback of a high computational cost for training. Though LSSVM has been widely used for many applications such as regression and pattern recognition, it is plagued by the drawback that two hyper-parameters, i.e. the regularization parameter and the kernel parameter, have to be defined by the user. Numerous experiments have proved that the choice of the two parameters can dramatically affect the quality of the obtained LSSVM model. A good choice of them makes LSSVM an ideal function approximater or classifier, while a poor choice can result in very large prediction errors. As a result, the main issue for practitioners trying to apply LSSVM is how to set the two parameters to ensure good generalization performance for a given training data set.

The problem of having to choosing the hyper-parameters inevitably increases the difficulty of applying LSSVM and limits its use on academic and industrial platforms. For this reason, it is necessary to develop an automated and effective method of tuning these parameters. In this paper, we present such a method which is based on colonial competitive algorithm (CCA), having the purpose of finding the optimal setting of regularization parameter and kernel parameter. CCA, which forms the basis of the proposed parameter tuning method, is a newly developed evolutionary algorithm

inspired by imperialistic competition mechanism [3]. The excellent capabilities of CCA such as higher convergence rate and better global optimum achievement have made it outperform its classic counterparts in preliminary investigations.

The validity of the CCA assisted LSSVM (hereafter referred to as CCA-LSSVM) is verified by using several benchmark test examples, and the simulation results demonstrate the efficacy of proposed paradigm in comparison with MLP, RBNN, as well as LSSVM whose hyper-parameters are tuned by other technique.

Least Squares Support Vector Machine

Let $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_l,y_l)\}$ be a training set with input data $\boldsymbol{x}_k\in R^d$ and corresponding output data $y_k\in R$. Learning from the training data can be viewed as approximating a multivariate function that represents the relation between the input data and output data. In the general case, the input data are mapped into a high dimensional space called feature space by a nonlinear mapping $\boldsymbol{\varphi}$, and then a linear model can be constructed in the feature space:

$$f(\boldsymbol{x})=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})+b \tag{1}$$

where $\boldsymbol{\omega}$ is a m-dimensional coefficient vector and b is a bias term.

LSSVM for Function Estimation. In LSSVM for function estimation, one defines the following optimization problem:

$$\min_{\boldsymbol{\omega},b,\boldsymbol{e}} J(\boldsymbol{\omega},b,\boldsymbol{e})=\frac{1}{2}\|\boldsymbol{\omega}\|^2+\frac{\gamma}{2}\sum_{k=1}^{l}e_k^2 \tag{2}$$

s.t.

$$y_k=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_k)+b+e_k \quad (k=1,2,\ldots,l) \tag{3}$$

where $\boldsymbol{e}=[e_1,e_2,\ldots,e_l]^{\mathrm{T}}$, $e_k\in R$ denotes the error vector, and γ is the regularization parameter. Using the optimization theory, one can define the Lagrange function for this problem as follows:

$$L(\boldsymbol{\omega},b,\boldsymbol{e};\boldsymbol{\alpha})=J(\boldsymbol{\omega},b,\boldsymbol{e})-\sum_{k=1}^{l}\alpha_k(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_k)+b+e_k-y_k) \tag{4}$$

where $\alpha_k(k=1,\ldots,l)$ are Lagrange multipliers. The conditions for optimality lead to a set of linear equations:

$$\begin{bmatrix}0 & \mathbf{1}^{\mathrm{T}}\\ \mathbf{1} & \boldsymbol{\Omega}+\gamma^{-1}\boldsymbol{I}\end{bmatrix}\begin{bmatrix}b\\ \boldsymbol{\alpha}\end{bmatrix}=\begin{bmatrix}0\\ \boldsymbol{y}\end{bmatrix} \tag{5}$$

where $\boldsymbol{y}=[y_1,y_2,\ldots,y_l]^{\mathrm{T}}$, $\mathbf{1}=[1,1,\ldots,1]^{\mathrm{T}}_{1\times l}$, $\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\ldots,\alpha_l]^{\mathrm{T}}$ and $\boldsymbol{\Omega}_{mn}=\boldsymbol{\varphi}(\boldsymbol{x}_m)^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_n)$ $(m,n=1,\ldots,l)$. According to Mercer's condition, there exists kernel function $K(\boldsymbol{x}_m,\boldsymbol{x}_n)=\boldsymbol{\varphi}(\boldsymbol{x}_m)^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_n)$. The resulting LSSVM model for function estimation becomes

$$f(\boldsymbol{x})=\sum_{k=1}^{l}\alpha_k K(\boldsymbol{x},\boldsymbol{x}_k)+b \tag{6}$$

where α_k, b are the solutions to the linear system Eq. 5.

LSSVM for Classification. Considering $y_k(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_k)+b)=1-e_k$, $y_k\in\{+1,-1\}$ for classification, similar to function estimation, the solution lead to a set of linear equations:

$$\begin{bmatrix}0 & \boldsymbol{y}^{\mathrm{T}}\\ \boldsymbol{y} & \boldsymbol{\Omega}+\gamma^{-1}\boldsymbol{I}\end{bmatrix}\begin{bmatrix}b\\ \boldsymbol{\alpha}\end{bmatrix}=\begin{bmatrix}0\\ \mathbf{1}\end{bmatrix} \tag{7}$$

where $\boldsymbol{\Omega}_{mn} = y_m y_n \boldsymbol{\varphi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{x}_n)$. The resulting LSSVM model for classification is

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{k=1}^{l} \alpha_k y_k K(\boldsymbol{x}, \boldsymbol{x}_k) + b \right) \tag{8}$$

where α_k, b are the solutions to the linear system Eq. 7. Two kinds of kernel function, i.e. Gaussian kernel and wavelet kernel [4], are considered in this work, and their analytical expressions are

$$K(\boldsymbol{x}, \boldsymbol{x}_k) = \exp(-\| \boldsymbol{x} - \boldsymbol{x}_k \|^2 / (2\sigma^2)) \tag{9}$$

$$K(\boldsymbol{x}, \boldsymbol{x}_k) = \prod_{i=1}^{d} \left(\cos(1.75(\boldsymbol{x}^{(i)} - \boldsymbol{x}_k^{(i)}) / a) \exp(-\| \boldsymbol{x}^{(i)} - \boldsymbol{x}_k^{(i)} \|^2 / (2a^2)) \right) \tag{10}$$

where $\boldsymbol{x}_k^{(i)}$ denotes the ith component of the kth training sample. The kernel parameter to be tuned in Eq. 9 is the kernel width σ, while that in Eq. 10 is the wavelet dilation coefficient a.

Basic Principle of CCA

CCA is a novel evolutionary algorithm for optimization using developmental process of human history as source of inspiration. Similar to other evolutionary algorithms that start with initial populations, CCA begins with initial empires in which each individual is called a country. There are two types of countries, colony and imperialist state that collectively form an empire. Imperialistic competitions among the empires form the basis of CCA. During this competition, weak empires collapse and powerful ones take possession of their colonies. Finally, imperialistic competitions will converge to a state in which there exists only one empire and its colonies are in the same position and have the same fitness value as the imperialist state.

First, initialize a number of countries which represent possible combinations of (γ, σ) or (γ, a), and select those with better fitness values to be the imperialist states, while the rest form the colonies of these imperialists. As is shown in Fig. 1(a), the stronger empires have greater number of colonies while weaker ones have less.

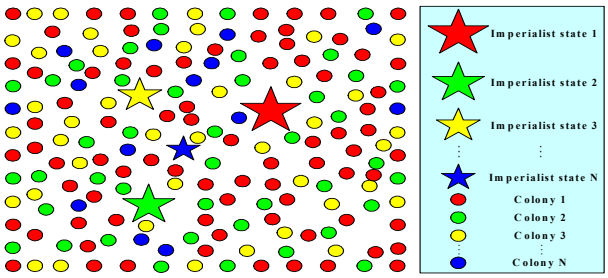
Then, after forming initial empires, the colonies in each of them start moving toward their relevant imperialist state, and the moving model is shown in Fig. 1(b). While moving toward the imperialist, a colony might reach a position with better fitness value than that of imperialist. In this case, the imperialist and the colony change their positions. Additionally, a fraction of colonies will try to improve their power by moving randomly to different positions. Such sudden changes in colony positions are referred to as revolution.

In the imperialistic competition process, all empires try to take the possession of colonies of other empires. The imperialistic competition gradually brings about a decrease in the power of weaker empires and an increase in the power of stronger ones, which is modeled by just picking some (usually one) of the weakest colonies of the weakest empires and making a competition among all empires to posses this colony, as shown in Fig. 1(c). An empire collapses when it loses all of its colonies. After a while all the empires except the most powerful one will collapse and all colonies will be under the control of this unique empire, which means CCA converges to the best solution.

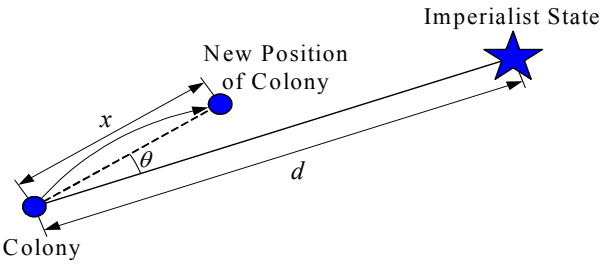
CCA for (γ, σ) or (γ, a) Tuning

As mentioned in the introduction, the generalization performance of LSSVM heavily depends on the proper setting of the regularization parameter and the kernel parameter. A CCA-based method for automatically tuning them is developed by authors. To apply CCA to (γ, σ/a) tuning, one must define the fitness function to be optimized. To obtain overall generalization performance of the model, leave-one-out, z-fold-cross-validation and training-validation approach are commonly used in the training of machine learning techniques [5]. In this work, the mean squared error derived from leave-one-out approach is chosen as the fitness function, and it is used to check the prediction quality of a LSSVM model based on a specific country, i.e. a specific (γ, σ) or (γ, a) sample. The CCA assisted LSSVM process is depicted in Fig. 2, where the dashed area is the flowchart of CCA. The

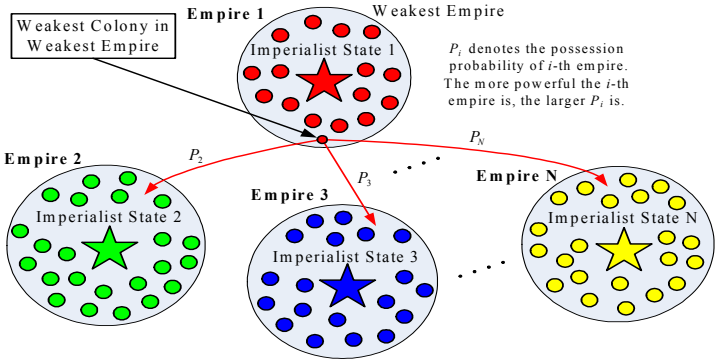
objective of CCA is to find the optimal values of (γ, σ) or (γ, a) that yield LSSVM the lowest leave-one-out mean squared error.



(a) The initial empires



(b) Motion of a colony toward imperialist state



(c) The modeled imperialistic competition

Fig. 1 The rationale of CCA

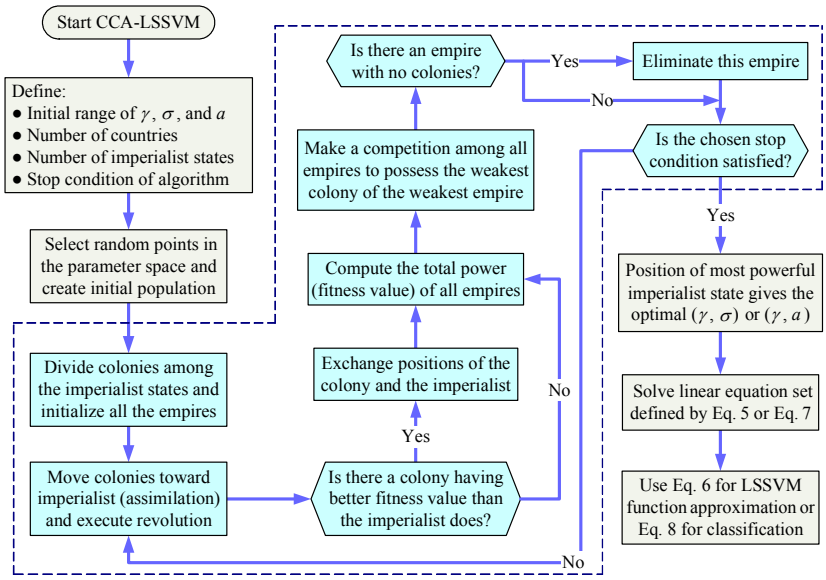


Fig. 2 CCA assisted LSSVM for function estimation and classification

Classification on Benchmark Problem. The UCI binary classification benchmark repository is used: Bupa liver disorders (bld) and Pima Indians diabetes (pid) with input dimension equal to 6 and 8, and the total number of patterns 345 and 768. Two thirds of the data are extracted as the training set, and the rest as the test set. The initial ranges of γ, σ and a are [0.1, 10000], [0.1, 100] and [0.1, 100], respectively. The numbers of countries and imperialist states in the initial population of CCA are 20 and 5. Stop condition is that only one empire is left, and it and its colonies are in the same position.

Simulating annealing (SA) representing an alternative method of tuning hyper-parameters is used to provide a comparison for results obtained using CCA, and the maximum number of fitness function

evaluations of SA is set equal to that actually performed by CCA. The classification results of CCA/SA-LSSVM are shown in Table 1, where those of MLP and RBNN are also presented.

Table 1 Classification results on the datasets of bld and pid

Data set	Number of instances		Classification error rate on test set (%)					
	Training	Testing	CCA-LSSVM (Gaussian)	CCA-LSSVM (Wavelet)	SA-LSSVM (Gaussian)	SA-LSSVM (Wavelet)	MLP	RBNN
bld	230	115	19.13	17.39	24.35	23.48	29.57	26.96
pid	512	256	21.48	20.70	25.39	24.22	29.69	26.95

Approximation of Single-Variable Function. The function to be approximated is defined as

$$f(x)=\begin{cases}-2.186x-12.864 & -10\le x<-2\\ 4.246x & -2\le x<0\\ 10\mathrm{e}^{-0.05x-0.5}\cdot\sin((0.03x+0.7)x) & 0\le x\le 10\end{cases}. \tag{11}$$

The approximation results of CCA/SA-LSSVM using both Gaussian and wavelet kernels, MLP and RBNN on this example are reported in Table 2.

Table 2 Approximation results on the single-variable function example

Number of instances		Normalized root of mean-square-error on test set					
Training	Testing	CCA-LSSVM (Gaussian)	CCA-LSSVM (Wavelet)	SA-LSSVM (Gaussian)	SA-LSSVM (Wavelet)	MLP	RBNN
100	100	0.0067	0.0022	0.0085	0.0031	0.0215	0.0119

Approximation of Two-Variable Function. This experiment is to approximate

$$f(x_1,x_2)=(x_1^2-x_2^2)\sin(0.5x_1)\quad -10\le x_1\le 10,\ -10\le x_2\le 10. \tag{12}$$

The approximation results for this example are reported in Table 3.

Table 3 Approximation results on the two-variable function example

Number of instances		Normalized root of mean-square-error on test set					
Training	Testing	CCA-LSSVM (Gaussian)	CCA-LSSVM (Wavelet)	SA-LSSVM (Gaussian)	SA-LSSVM (Wavelet)	MLP	RBNN
81	1600	0.0274	0.0223	0.0329	0.0297	0.1948	0.0459

Conclusion

In this paper, we develop an automatic and effective method for tuning the regularization parameter and kernel parameter of LSSVM, which takes full advantage of the excellent global optimum achievement of CCA. Results on several benchmark examples reveal that the proposed CCA assisted LSSVM is a competitive and powerful tool for classification and function approximation.

References

[1] V. Vapnik. The nature of statistical learning theory. New York: Springer-Verlag, 1995.
[2] J. Suykens and J. Vandewalle. Least squares support vector machine classifiers. Neural Processing Letters, 1999, 9(3), 293–300.
[3] R. Rajabioun, E. A. Gargari and C. Lucas. Colonial competitive algorithm as a tool for Nash equilibrium point achievement. LNCS, 2008, Volume 5073: 680–695.
[4] L. Zhang, W. Zhou and L. Jiao. Wavelet support vector machine. IEEE Trans Syst Man Cybern Part B Cybern, 2004, 34(1), 34–39.
[5] K. Ito, R. Nakano. Optimizing support vector regression hyperparameters based on cross-validation. Proceedings of the International Joint Conference on Neural Networks, 2003, Volume 3: 2077–2082.

Advances in Civil Engineering
10.4028/www.scientific.net/AMR.255-260

Colonial Competitive Algorithm Assisted Least Squares Support Vector Machines
10.4028/www.scientific.net/AMR.255-260.2082