Nesting Differential Evolution to Optimize the Parameters of Support Vector Machine for Gender Classification of Facial Images

Pin Liao[1, a], Sensen Wang[2, b], Xin Zhang[1, c], Kunlun Li[1, d], Mingyan Wang[2,e]

[1]College of Science and Technology, Nanchang University, Nanchang, China

[2]Information Engineering School, Nanchang University, Nanchang, China

[a]pinliao@ncu.edu.cn, [b]wsensen@hotmail.com, [c]zhangxin77@gmail.com, [d]likunlunnihao@126.com , [e]mingyanw@ncu.edu.cn

Keywords: Support Vector Machine; Parameter Optimization; Differential Evolution; Nesting Optimization Method; Gender Classification of Facial Images

Abstract. Support Vector Machine (SVM) is an influential and fashionable statistical learning technique for binary classification and regression. The generalization performance of SVM highly depends on proper tuning of the penalty parameter and kernel function parameter(s). An original technique is proposed to quickly search the optimal SVM parameters separately by nesting two differential evolution (DE) algorithms, which can avoid repetitious costly computation and then shrink the computation cost by orders of magnitude compared to the existing approaches which tune all the parameters concurrently. The experimental results on gender classification of facial images illustrate that the proposed technique can efficiently construct an SVM classifier with significant generalization capabilities.

Introduction

Support Vector Machine (SVM) [1, 2] is one of the best-known and significant supervised statistical learning methods for the solution of classification and regression problems with strong theoretical foundations based on the principle of structural risk minimization.

However, improper selecting of SVM parameters usually leads to very poor generalization capabilities. Searching the optimal SVM parameters is decisive for achieving exceptional performance, which however is still a very hard problem. The parameters of SVM to be optimized consist of the penalty parameter and the kernel function parameter(s). The trade-off between empirical error minimization and hypothesis space complexity minimization is balanced by the penalty parameter. And the kernel function parameter(s) can be decisive to mapping input data points into some high-dimensional feature space, if the kernel function is non-linear.

In this study, a novel method is presented to optimize the penalty parameter and the kernel function parameter(s) individually based on nesting two differential evolution algorithms (NDE), unlike the traditional methods which search all the parameters simultaneously. Therefore, by keeping off expensive repetitious computation, NDE can decrease the computing cost by orders of magnitude.

Support Vector Machine for Classification

Training SVM requires solving the following quadratic programming (QP) optimization problem.

$$\begin{aligned} &\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - \sum_{i=1}^{N}\alpha_i \\ &s.t. \qquad y^T\alpha = 0 \\ &\qquad\qquad 0 \le \alpha_i \le C, i = 1,\ldots,\mathrm{N} \end{aligned} \tag{1}$$

where α_i are Lagrange multipliers, N is the size of the given training set

$$S = \left\{ \left(x_i, y_i \right), i = 1, \ldots, \mathrm{N}, x_i \in \Re^D, y_i \in \left\{ -1, 1 \right\} \right\} \tag{2}$$

where D is the dimension of data points x_i.

The entries of Q_{ij} of the symmetric and positive semidefinite matrix Q are defined by

$$Q_{ij} = y_i y_j K(x_i, x_j), \quad \mathrm{i, j} = 1, 2, \ldots, \mathrm{N}, \tag{3}$$

where $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ denotes a kernel function.

Linear kernel, radial basis function (RBF) kernel, polynomial kernel, and sigmoid kernel are the usually-used kernel functions. And RBF kernel can be described as

$$K(x, y) = \exp\left(-\gamma \left(x - y \right)^T \left(x - y \right) \right), \gamma > 0. \tag{4}$$

SVM is trained using the given training set, and then the values of all Lagrange multipliers α_i are calculated by solving the QP problem (1). As a result, a new data point x can be classified with the following function:

$$g(x) = sign\left(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + \beta \right) \tag{5}$$

where the bias term is denoted by β, calculated through the SVM training. Specifically β can be a hidden part of the kernel function and be unneeded, when the kernel function is RBF.

Due to the significant generalization capabilities and the simplicity with only one parameter γ, RBF is adopted in this study as the kernel function.

Differential Evolution (DE)

As one of the most powerful evolutionary algorithms first introduced by Storn and Price [3], the differential evolution (DE) algorithm is a very competitive population-based stochastic search method with desirable convergence attributes and has been successfully employed in many applications. DE is a simple, straightforward and efficient scheme for global optimization over continuous spaces with a very few number of control parameters, which particularly does well in dealing with non-differentiable, nonlinear and multimodal optimization problem.

Inspired by natural evolution, DE iteratively evolves a population of trial solutions to seek the global optimum point in a continuous search space. The initial population contains individuals randomized uniformly within the search range, and is supposed to cover the whole seeking space as much as possible. Then by performing mutation, crossover and selection operators, the population is gradually evolved. New individuals are generated by using mutation and crossover; and then selection operator decides which of the new individuals should be retained for the next generation. The evolutionary operators are performed in the order: first mutation, second crossover, and finally selection.

1. Mutation

For each individual (called target vector) in the population x_i^t, *i= 1, 2, 3, . . ., NP* (population size), a mutant vector is generated by adding the weighted difference between two individuals to a third individual

$$v_i^{t+1} = x_{r1}^t + F \cdot \left(x_{r2}^t - x_{r3}^t \right) \\ i \neq r1 \neq r2 \neq r3 \tag{6}$$

where *r1, r2, r3* are randomly chosen from *{1, 2, . . . , NP},* and the scalar number F = 0.8 in this work.

2. Crossover

The parent individual is mixed with the mutated vector to produce a trial vector u_{ij}^{t+1} as below.

$$u_{\mathrm{ij}}^{t+1} = \begin{cases} v_{\mathrm{ij}}^{t+1} & \text{if } \left(\mathrm{rand_j}(0,1) \leq \mathrm{CR} \right) \text{ or } \left(\mathrm{j} = \mathrm{j}_{rand} \right) \\ \mathrm{x}_{\mathrm{ij}}^{t} & otherwise \end{cases} \qquad j = 1, 2, \ldots, D \tag{7}$$

where rand(0, 1) represents a uniformly distributed random variable within the range [0, 1]; the crossover rate CR is a user-defined constant within the range [0, 1] and is set as 0.95 for this study;

and j_{rand} is a randomly chosen integer from {1, 2, . . . , D}, as D is the dimension number of the vectors.

3. Selection

The fitness values of the trial vector and the target vector are compared, and then the winner is chosen for the next generation, as showed in Eq. (8).

$$x_i^{t+1} = \begin{cases} u_i^{t+1} & \text{if } f(u_i^{t+1}) \le f(x_i^t) \\ x_i^t & \text{otherwise} \end{cases} \tag{8}$$

Parameter Optimization Based on NDE

A convex quadratic programming (QP) problem needs to be solved during SVM training, which can be decomposed into a series of sub-problems by the sequential minimal optimization (SMO) [4] algorithm. SMO employs just a two-sampled working set to reduce the training cost significantly. Further more, LIBSVM [5], a modified SMO algorithm, can lead to a fast convergence by exploiting second order information. In [6], we proposed a parallel implementation of LIBSVM for multi-core and multiprocessor systems which can reduce the training time dramatically. Thereby, the parallel implementation of LIBSVM is adopted to train SVM in this study.

Traditional methods optimize the penalty factor C and the kernel parameter γ simultaneously, and then the kernel matrix must be updated for evaluating each solution, as the updating of the kernel matrix is the most computation expensive part of SMO. NDE optimizes the parameters separately by nesting two DE algorithms, while the inner loop DE optimizes C and the outer loop DE optimizes γ. As a result, in the inner loop with a fixed γ, the kernel matrix is unchanged and can be reused for all the inner iterations, as each of the iterations corresponds to an individual C in a population. Therefore, NDE can decrease the computing cost by orders of magnitude by keeping off expensive repetitious computation.

The algorithm of NDE is given as follows:

```
1    Initialize a $\gamma$ population with each individual uniformly distributed in the range.
2    Calculate the fitness of each $\gamma$ in the population.
**Loop for each $\gamma$:**
     2.1    Initialize a $C$ population with each individual uniformly distributed in the range.
     2.2    Calculate the fitness of each $C$ in the population with the fixed $\gamma$ (i.e. the average test
            rates of a k-fold cross validation corresponding to the {$C$, $\gamma$}).
     2.3    Set the fitness of the $\gamma$ as the greatest fitness among the $C$ population.
     2.4    Go to Step 2 if the best fitness is good enough.
     2.5    Using the mutation, crossover and selection operators to improve a new C population
            by. Go to Step 2.2.
3    Go to Step 5 if the best fitness is good enough.
4    Using the mutation, crossover and selection operators to improve a new $\gamma$ population. Go to
     Step 2.
5    Attain the optimal {$C$, $\gamma$} corresponding to the optimum fitness (i.e. the average test rate of a
     k-fold cross validation).
```

Experimental Results

With the purpose of assessing the effectiveness of the proposed NDE approach, a lot of experiments for gender classification of facial images are conducted. A training set with over 60,000 samples is produced based on over 20,000 facial images collected from the Internet, including about 40,000 synthetic samples derived from the original images with slight geometric transforms of translation, scaling, rotation and mirror-reflection. The testing set contains 4404 original facial images independent of the training samples. Gabor filters are used for feature extraction and the feature dimension is 5856.

The fitness values are computed using the 10-fold cross-validation technique on the training data set. After that, an SVM classifier is trained on the whole training set with the optimal parameters {C, γ} corresponding to the top fitness, and then is used to classify the testing set.

The range of C is experientially set as [2^{-3}, 2^{10}], and the range of γ [2^{-13}, 2^{-1}]. Table 1 illustrates that NDE is superior to the grid search and the traditional DE which search the penalty parameter and the kernel parameter(s) concurrently. NDE is able to explore much more solutions than the traditional techniques in the same amount of time by avoiding costly repetitious computation, and hence can obtain a more optimum solution corresponding to a higher classification rate.

Table 1. Optimal parameters and classification rates corresponding to different optimization methods

Method	C	γ	Accuracy (%)
Grid Search	3.48220225	0.00170029	96.9119
DE	7.72841890	0.00118574	97. 2525
NDE	5.26491264	0.00137800	97. 3887

Conclusion

In this study, a new method based on nesting two DE algorithms, NDE, is developed and successfully implemented to the parameters optimization of SVM. Different from the traditional methods, NDE optimizes the parameters separately and can decrease the computing cost by orders of magnitude by avoiding expensive repetitious computation. The experimental results indicate that NDE is very efficient and effective to optimize the parameters for SVM and is capable to generate favorable results.

Acknowledgement

This research is partially sponsored by the Jiangxi Province Education Department science and technology project (Project No. GJJ13086), and Nanchang City Key Laboratory of Pattern Recognition.

References

[1] Corinna Cortes and Vladimir Vapnik, Support-vector networks. Machine Learning (Historical Archive) 20(3) (1995) 273-297.

[2] V. N. Vapnik, The Nature of Statistical Learning Theory, 2nd ed., New York: Springer-Verlag, 1999.

[3] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (4) (1997) 341–359.

[4] John C. Platt. Fast training of support vector machines using sequential minimal optimization, in B. Schölkopf, C. Burges, and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning. MIT Press, (1999) 185–208.

[5] C. C. Chang, C. J. Lin. LIBSVM: A Library for Support Vector Machines, Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[6] Pin Liao, et al. Parallel Training of Support Vector Machine for Multi-core and Multiprocessor Systems, International Academic Conference on the Information Science and Communication Engineering (ISCE2014), (2014) 265-270.