STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LI**, Number 1, 2006

NONLINEAR EVOLUTIONARY SUPPORT VECTOR MACHINES. APPLICATION TO CLASSIFICATION

RUXANDRA STOEAN, D. DUMITRESCU, AND CATALIN STOEAN

Abstract. Support vector machines are a modern and very efficient learning heuristic. However, their internal engine relies on not very easy or common mathematical concepts. The paper presents a newly developed simpler design of the engine, built through the means of evolutionary computation, in the context of nonlinear support vector machines. Experiments are carried on fictitious 2-dimensional points data sets and demonstrate once again the promise of the new approach.

Keywords:*support vector machines, nonlinear hyperplane, evolutionary algorithms, polynomial classifier, radial decision function, 2-dimensional points data sets*

1. Introduction

Support vector machines (SVMs) are a type of learning machines [9]. According to [6], "support vector machines are a system for efficiently training linear learning machines in kernel-induced feature spaces, while respecting the insights of generalization theory and exploiting optimization theory".

As all learning machines, SVMs act in two stages. In the training stage, the correspondence between every input vector and given output is internally discovered and learnt. In the test step, prediction of the output for previously unknown input vectors is performed according to what has been learnt.

SVMs have been successfully applied to a wide range of pattern recognition problems. Interest in present paper is shown however only in what concerns classification.

Received by the editors: December 1, 2005.

2000 *Mathematics Subject Classification.* 68T05, 68T20, 92D10.

1998 *CR Categories and Descriptors.* I.2.6 [**Learning**]: Subtopic – *Concept Learning*; I.2.8 [**Problem Solving, Control Methods, and Search**]: Subtopic – *Heuristic methods*; J.3 [**Life and Medical Sciences**]: Subtopic – *Biology and Genetics*.

The task for SVMs here is then to achieve an optimal separation of data into classes. By resorting to evolutionary algorithms, authors propose a simpler alternative to the standard approach of SVMs to solving this optimization problem.

Initially, SVMs were developed for linearly separable data and were later improved to handle nonseparable cases as well. Consequently, the new technique, called evolutionary support vector machines (ESVMs), has followed the same steps. Linear separating hyperplanes for separable and nonseparable data were detected through evolutionary computation in [11, 12]. The last and most difficult step, i.e. the discovery of the optimal nonlinear hyperplane to deal with nonseparable data, is treated in present paper. The data sets that experiments are conducted on are fictitious 2-dimensional points sets.

The paper is structured as follows. Section 2 presents an overview of support vector machines for classification. Section 3 outlines the idea and structure of evolutionary support vector machines; values for parameters of both the evolutionary algorithm and the support vector machine are appointed and conducted experimental results are illustrated. Finally, some conclusions are reached and ideas for future work are discussed.

2. Principles of support vector machines

In standard manner, support vector machines deal with binary classification problems. They have, however, been extended to handle multi-class categorization. The new evolutionary support vector machines are built according to the classical binary situation; in the future, ESVMs will also be broadened to cover the multi-class circumstances. Consequently, in what follows, the concepts within SVMs will be explained on binary labelled data [3, 10].

Suppose the training data is of the following form:

$$\{(x_i, y_i)\}_{i=1,2,\ldots,m} \tag{1}$$

where every $x_i \in R^n$ represents an input vector and each y_i an output (label).

Let us first suppose that the two subsets of input vectors labelled with $+1$ and -1, respectively, are linearly separable. The positive and negative training vectors are then separated by the hyperplane:

$$\langle w, x \rangle - b = 0, \tag{2}$$

where $w \in R^n$ is the normal to the hyperplane, $b \in R$ and $\frac{|b|}{\|w\|}$ is the distance from the origin to the hyperplane.

Accordingly, two data subsets are linearly separable iff there exist $w \in R^n$ and $b \in R$ such that:

$$\begin{cases} \langle w, x_i \rangle - b > 0, & y_i = 1, \\ \langle w, x_i \rangle - b < 0, & y_i = -1, i = 1, 2, ..., m. \end{cases} \tag{3}$$

According to [1], two data subsets are linearly separable iff there exist $w \in R^n$ and $b \in R$ such that:

$$\begin{cases} \langle w, x_i \rangle - b > 1, & y_i = 1, \\ \langle w, x_i \rangle - b < -1, & y_i = -1, i = 1, 2, ..., m. \end{cases} \tag{4}$$

Consequently, the separating hyperplane lies in the middle of the parallel supporting hyperplanes of the two classes.

Following the structural risk minimization principle [13, 14, 15], i.e. one gets that, in order to generalize well, the support vector machine must provide a hyperplane that separates the training data with as few errors as possible and, at the same time, with a maximal margin of separation. One subsequently obtains the optimization problem (P_1):

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \frac{\|w\|^2}{2}, \\ \text{subject to } y_i(\langle w, x_i \rangle - b) \geq 1, i = 1, 2, ..., m. \end{cases} \tag{5}$$

where $\frac{2}{\|w\|}$ is the value of the margin.

Given a training data set that is nonseparable, it is obviously not possible to build a separating hyperplane without any classification errors. However, construction of an optimal hyperplane that minimizes misclassification would be of interest [8]. Previous ideas can be extended to handle this new situation by relaxing the constraints in (4). This can be achieved by bringing in some positive variables, called *slack* variables [4]. The introducing of these new variables relies on the fact that any training data point has a deviation from its supporting hyperplane, i.e. from the ideal condition of data separability, of $\frac{\pm \xi_i}{\|w\|}$. This affects the separation condition, which then becomes [4]:

$$y_i(\langle w, x_i \rangle - b) \geq 1 - \xi_i, i = 1, 2, ..., m \tag{6}$$

where $\xi_i \geq 0$. Thus, for a training data point to be erroneously classified, its corresponding ξ_i must exceed unity.

Simultaneously with (6), sum of misclassifications must be minimized. As a consequence, (P1) changes to (P2):

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \frac{\|w\|^2}{2} + C\sum_{i=1}^{m} \xi_i, \\ \text{subject to } y_i(\langle w, x_i \rangle - b) \geq 1 - \xi_i, \\ \xi_i \geq 0, i = 1, 2, ..., m. \end{cases} \tag{7}$$

where C corresponds to assigning higher penalties for errors.

The concepts can be extended even further to the construction of a nonlinear separating hyperplane for nonseparable data. Based on [5], the training data can be nonlinearly mapped into a high enough dimensional space and linearly separated there.

Suppose an input vector is mapped into some Euclidean space, H, through a mapping $\Phi : R^n \mapsto H$. It can be easily seen that within (P_2) vectors in R^n appear only as part of dot products. Vectors in H should appear as part of dot products in its formulation, as well. Therefore, the equation of the separating hyperplane in H becomes:

$$\langle \Phi(w), \Phi(x_i) \rangle - b = 0 \tag{8}$$

where $\Phi(w)$ is the normal to the hyperplane.

The squared norm

$$\|w\|^2 = \langle w, w \rangle \tag{9}$$

changes to

$$\langle \Phi(w), \Phi(w) \rangle. \tag{10}$$

Now, if there were a kernel function K such that:

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle \tag{11}$$

where $x, y \in R^n$, one would use K in the training algorithm and would never need to explicitly even know what Φ is.

At this moment, the question is what kernel functions meet (11). The answer is given by Mercer's theorem from functional analysis [2]. The problem is that it may not be easy to check whether Mercer's condition is satisfied in every case of a new kernel. There are, however, a couple of classical kernels that had been demonstrated to meet Mercer's condition [2]:

- Polynomial classifier of degree p: $K(x, y) = \langle x, y \rangle^p$
- Radial basis function classifier: $K(x, y) = e^{\frac{\|x-y\|^2}{\sigma}}$

Consequently, the optimization problem (P_2) will now change to (P3):

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \frac{K(w,w)}{2} + C\sum_{i=1}^{m}\xi_i, \\ \text{subject to } y_i(K(w,x_i) - b) \geq 1 - \xi_i, \\ \xi_i \geq 0, i = 1, 2, ..., m. \end{cases} \tag{12}$$

3. Design of nonlinear evolutionary support vector machines

The optimization problem in support vector machines is standardly solved using concepts of convexity and resorting to an extension of the well-known method of Lagrange multipliers. The mathematics of the method can be found to be difficult.

Authors have brought the ESVM alternative to this technique, which is very easy to understand and apply. Standard evolutionary algorithms are used in this respect.

In present work, nonlinear ESVMs are outlined only. Linear ESVMs for separable and nonseparable data are particular situations of proposed algorithm; however, they can be found in [11, 12], respectively.

The way in which components of the evolutionary algorithm are considered with respect to nonlinear support vector machines is outlined. Experimental results are reached and illustration of the separation is given for different fictitious 2D points data sets appointed in order to obtain three nonlinear separating hyperplanes, *i.e* odd or even polynomial and radial classifiers.

3.1. Components of the evolutionary algorithm.

Components regard representation, initialization of the population, expression of the fitness function, the selection and variation operators.

Representation

A chromosome has the following structure of w, b and ξ:

$$c = (w_1, ..., w_n, b, \xi_1, ..., \xi_m) \tag{13}$$

Proposed evolutionary algorithm thus includes the training errors in the structure of the chromosome. In the end of the algorithm, the training points that are correctly placed will have the corresponding $\xi_i s$ less than unity, while those erroneously placed will have their $\xi_i s$ exceed it.

Initial population

Chromosomes are randomly generated following a uniform distribution, such that $w_i \in [-1, 1], i = 1, 2, ..., n$, $b \in [-1, 1]$ and $\xi_j \in [0, 1], j = 1, 2, ..., m$.

Fitness evaluation

The expression of the fitness function is considered as follows:

(14)

$$f(c) = f(w_1, ..., w_n, b, \xi_1, ..., \xi_m) = K(w,w) + C\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{m}[t(y_i(K(w,x_i) - b) - 1 + \xi_i)]^2,$$

where

$$t(a) = \begin{cases} a, & a < 0, \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

One is led to minimize(f(c), c).

Genetic operators

Tournament selection is used. Intermediate crossover and mutation with normal perturbation are considered. Mutation is restricted only for errors, preventing the ξ_is from taking negative values.

Stop condition

The algorithm stops after a predefined number of generations. In the end, it obtains the equation of the hyperplane, i.e. w and b. Errors on training set also result from the algorithm, i.e. those corresponding to $\xi_i > 1$, $i = 1, 2, ..., m$

3.2. **Experimental results.** Three fictitious 2-dimensional points data sets were built in order to allow construction of an even polynomial classifier, an odd one and a radial decision function. Illustration of their configurations and of the obtained nonlinear hyperplanes is given in Figures 1, 2 and 3.

Values for the specific parameters of every chosen kernel are given in turn in Table 1. The parameters of the support vector machine and of the evolutionary algorithm that are common to all kernels had the same values in all three cases and are outlined in Table 2. Some abbreviations are used therein, i.e. *ps* stands for population size, *ng* for number of generations, *cp* for crossover probability, *mp* for mutation probability, *ms* for mutation strength.

Degree odd polynomial	15
Degree even polynomial	14
Sigma	500

Table 1. Values for parameters of the chosen kernels

C	ps	ng	cp	mp	mp - $\xi_i s$	ms	ms - $\xi_i s$
1	200	1000	0.3	0.5	0.5	0.1	0.1

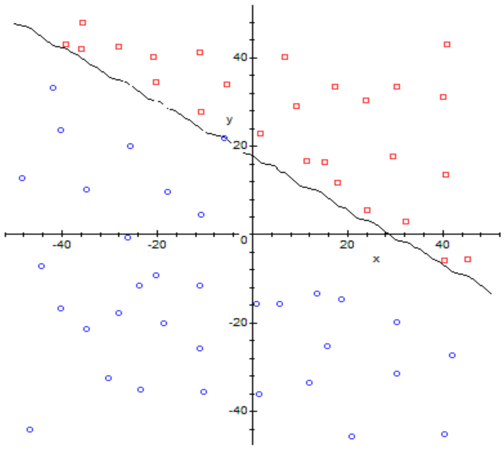

Figure 1. An odd polynomial classifier

Table 2. Values for parameters of the support vector machine and of the evolutionary algorithm

4. Conclusions and future work

From the discussion above, there arise various advantages in using evolutionary support vector machines instead of the classical architecture:

1. The evolutionary approach is much easier for both the developer and the end user.

2. The evolutionary solving of the optimization problem leads to the obtaining of w and b directly, while in the classical approach the equation of the optimal hyperplane is determined after Lagrange multipliers are found.

3. Moreover, in the case of the classical technique, when using kernels in which Φ cannot be explicitly obtained, it is not possible to determine w and b at all - it can only predict the class for a test vector.

4. The evolutionary method can also provide which training data cannot be correctly classified, as errors are included in the structure of the chromosomes; the evolutionary support vector machines self-determine their training error.

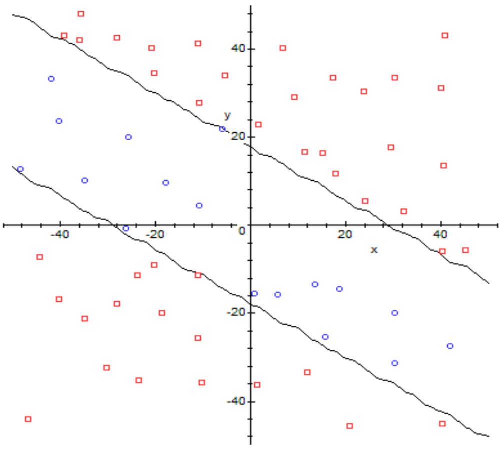

Figure 2. An even polynomial classifier

Future work envisages application and validation of proposed ESVMs to real-world problems. Also, the design of evolutionary multi-class support vector machines, based on classical SVM approaches to classification with more than two categories, is desired.

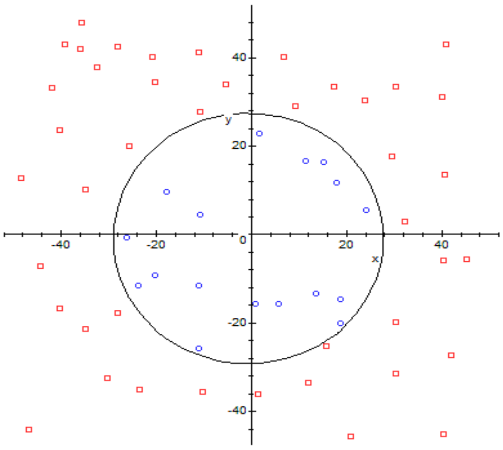

Figure 3. A radial polynomial classifier

References

[1] R.A. Bosch, J.A. Smith, *Separating Hyperplanes and the Authorship of the Disputed Federalist Papers*, American Mathematical Monthly, Volume 105, Number 7, pp. 601-608, 1998

[2] B. E. Boser, I. M. Guyon and V. Vapnik, *A Training Algorithm for Optimal Margin Classifiers*, In D. Haussler, editor, Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, pp. 11-152, Pittsburgh, PA, ACM Press, 1992

[3] C.J.C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery 2, 121-167, 1998

[4] C. Cortes, V. Vapnik, *Support Vector Networks*, Machine Learning, 20:273-297, 1995

[5] T. M. Cover, *Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition*, IEEE Transactions on Electronic Computers, vol. EC-14, pp. 326-334, 1965

[6] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000

[7] D. Dumitrescu, B. Lazzerini, L.C. Jain, A. Dumitrescu, *Evolutionary Computation*, CRC Press, Boca Raton, Florida, 2000

[8] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, New Jersey, 1999

[9] R. Lothian, *Introduction to Support Vector Machines*, Talk, 2003

[10] B. Scholkopf, *Support Vector Learning*, Dissertation, Berlin, 1997

[11] R. Stoean, D. Dumitrescu, *Linear Evolutionary Support Vector Machines for Separable Training Data*, Annals of the University of Craiova, Seria Matematica-Informatica, submitted for publication, 2005

[12] R. Stoean, D. Dumitrescu, *Evolutionary Support Vector Machines - a New Learning Paradigm. The Linear Non-separable Case*, Proceedings of the Symposium ”Colocviul Academic Clujean de Informatica”, accepted for publication, 2005

[13] V. Vapnik, *Inductive Principles of Statistics and Learning Theory*, In Smolensky, Mozer and Rumelhart (Eds.), Mathematical Perspectives on Neural Networks, Lawrence Erlbaum, Mahwah, NJ, 1995

[14] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, New York, 1995

[15] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998

Faculty of Mathematics and Computer Science, Department of Computer Science, University of Craiova, Craiova Romania

E-mail address: ruxandra.stoean@inf.ucv.ro

Faculty of Mathematics and Computer Science, Department of Computer Science, Babes-Bolyai University, Cluj - Napoca Romania

E-mail address: ddumitr@cs.ubbcluj.ro

Faculty of Mathematics and Computer Science, Department of Computer Science, University of Craiova, Craiova Romania

E-mail address: catalin.stoean@inf.ucv.ro