Fuzzy Support Vector Machines Based on Convex Hulls

Hongbing Liu¹,², Shengwu Xiong¹, Qiong Chen¹

1 School of Computer Science and Technology
Wuhan University of Technology
Wuhan 430070, P.R. China

2 Department of Computer Science
Xinyang Normal University
Xinyang 464000, P.R. China

Abstract—Fast Fuzzy Support Vector Machines (FFSVMs) based on the convex hulls are proposed in this paper. Firstly, the convex hull of each class data is generated by using the quick hull algorithm, and the data points lying inside the convex hull are not important to form FSVMs and then discarded. Secondly, the reduced training set consisting of the convex points is used to train the FFSVMs. Thirdly, the benchmark two-class problems and multi-class problems datasets are used to test the effectiveness and validness of FFSVMs. The experiment results indicate that FFSVMs not only reduce the training set but also achieve the same or better performance compared with the traditional FSVMs.

Keywords—support vector machines; fuzzy support vector machines; convex hulls; fast fuzzy support vector machines

I. INTRODUCTION

Support vector machines (SVMs) are new machine learning methods, evolving from the statistical learning theory. They embody the principle of the structural risk minimization. Owe to their higher generalization ability and better classification precision, SVMs can solve the overfitting problem effectively and can be applied to a number of issues [1], such as classification and regression fields. At present, SVMs have already been applied successively to the problems ranging from hand-written character recognition, face detection, speech recognition to medicine diagnosis [1, 2].

Support vector machines (SVMs) are new machine learning methods, evolving from the statistical learning theory. They embody the principle of the structural risk minimization. Owe to their higher generalization ability and better classification precision, SVMs can solve the overfitting problem effectively and can be applied to a number of issues [1], such as classification and regression fields. At present, SVMs have already been applied successively to the problems ranging from hand-written character recognition, face detection, speech recognition to medicine diagnosis [1, 2].

SVMs for pattern classification are based on two-class classification problems. Unclassifiable regions exist when SVMs are extended to multi-class problems. In SVMs, three questions must be paid attention to. One is how to extend two-class problems to multi-class problems. There are many methods to solve this problem, such as one-against-one, one-against-all and DAGSVMs [3,4]. In order to reduce unclassifiable regions, Inoue and Abe presented Fuzzy Support Vector Machines (FSVMs) [5]. They defined the decision functions according to the membership functions in the directions orthogonal to the hyperplane. The other is how to solve the overfitting problem, which is caused by treating every data point equally during training. Han-Pang Huang and Yi-Hung Liu presented the other FSVMs [6]. The performance of SVMs has been enhanced through assigning each training data a membership degree. The third problem is the time costs of training scale \( O(\ell^2) \), where \( \ell \) is the total size of training data. So far, many algorithms such as Chunking [7], SMO [8], SVMTorch [9], and LIBSVM [10] have been proposed to reduce the training time with time complexity \( TO(\ell^2+\tau) \), where \( T \) is the number of iterations and \( \tau \) is the size of working set. However, their training time complexity is still strongly related to the number of training patterns. The other method of decreasing the costs of training is reducing the number of training data. Although reduced set strategy has been successfully applied to solve SVMs pattern recognition problems [11], the selection of the suitable parameters is the bottleneck to limit their applications. In [12], the authors proposed a heuristic method for accelerating SVM regression training (HSVM for short), in which all the training data are first divided into several groups using some clustering methods, and then for each group, some training vectors are discarded based on the measurement of similarity among examples. In order to obtain the proper learning machines, the program must be run many times while tuning the parameter of the clustering, such as the number of clusters, the objective values, etc.

The aim of this paper is to seek the reduced training set by using the convex hull algorithm which can form the convex points. For the convex hulls, all the data points lying inside the convex polygon or polyhedron formed by the convex points are not important to form FSVMs and then discarded. In the other hand, it is impossible for the data points lying inside the convex hull of the class to be support vectors, which support the classification hyperplane. The convex points are selected to form the reduced training set, which are used to construct the FFSVMs.

II. FUZZY SUPPORT VECTOR MACHINES

The training set \( S \)

\[ S = \{(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}, y_i \in \{-1,1\}, i = 1, 2, ..., \ell\} \]

can be linearly separated by a maximum margin classifier named the following hyperplane.

\[ w^T x + b = 0 \]
where $w$ is a vector, $b$ is a scalar, they can be obtained by the constrained optimization problem \[1\]. FSVMs proposed in \[6\] solved the overfitting problem by introducing the membership degrees $u_i$ for every data, which are defined as

$$u_i = 1 - \frac{\|x_i - x^*\|}{\max_j \|x_j - x^*\|}$$

where $x^*$ is the center of class. The $w$ and $b$ of FSVMs are determined by the constrained optimization problem

$$\min W(\alpha) = \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{n} \alpha_i$$

s.t. $\sum_{i=1}^{n} \alpha_i y_i = 0$, $0 \leq \alpha_i \leq \mu C$ for $i = 1, 2, ..., \ell$

Some classification problems are non-linearly separable. Namely, in low dimensional space, they are not linearly separable, but they can be separated in higher dimensional space. Such as 0-1 task, these four points can’t be separated linearly in two-dimensional space, but they can be separated linearly in three-dimensional space. The key to the success of kernel functions lies in the special types of mapping which obeys Mercer’s theory and offers an implicit mapping into feature space. That means we needn’t to know and calculate the formula of the mapping. The decision function in the higher dimensional space is

$$g(x) = \text{sign} \left( \sum_{x \in S} \alpha_i y_i K(x_i, x) + b \right)$$

where $\alpha_i$ is determined by the problem

$$\min W(\alpha) = \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{n} \alpha_i$$

s.t. $\sum_{i=1}^{n} \alpha_i y_i = 0$, $0 \leq \alpha_i \leq \mu C$ for $i = 1, 2, ..., \ell$

For multi-class problem, one-against-one FSVMs are adopted in this paper. There are $\frac{n(n-1)}{2}$ two-class FSVMs for $n$-class problem and their decision functions are

$$g(x) = \text{sign}(f_\iota(x))$$

where $f_\iota(x) = \sum_{i \in \mathcal{A}_\iota} \alpha_i y_i K(x_i, x) + b$

There are many decision strategies, such as one-against-all method in \[1\], DDAGSVMs \[7\]. In the decision phases in \[4\], the membership function of $x$ to class $i$ is defined as

$$m_i(x) = \min \left( 1, \min_{j \in \mathcal{A}_i} f_j(x) \right)$$

and an unknown datum $x$ is classified into the class

$$\arg \min_{i \in \mathcal{A}} m_i(x)$$

III. FUZZY SUPPORT VECTOR MACHINES BASED ON CONVEX HULLS

The convex hull of a set of points is the smallest convex set that contains the points. It is a fundamental construction for mathematics and computational geometry. Convex hull has been proven to be a very useful shape in dealing with many computational problems, such as automated feature recognition from 2D as well as 3D. The convex hull of a set of points $S$ in a plane is the smallest convex polygon that encloses $S$. In other words, the convex hull of a set of points $S$ in a plane is the enclosing polygon with the smallest area and perimeter.

In SVMs, the support vectors are important to form classifiers and lie in the edge of the class body. The main point of FFSVMs is to preserve the more important data lying on the convex hull and discard the inside less important ones. The idea comes from the fact that the different training data have different roles in training. The points close to the hyperplane are decisive to support the hyperplane, such as the support vectors, the other points are less important and may bring negative effects to SVMs. The training set shown in figure 1 is the linearly separable case, and no points violate the classification hyperplane. Both the two hyperplanes are supported by three support vectors, namely the proposed

![Figure 1. Hyperplane on the convex hull](image)
method achieves the same support vectors and classification hyperplane as those of the traditional FSVMs. The red polygons are the convex hulls of the two class data, and the convex data points are selected to train FFSVMs. The data points marked “o” are the support vectors of FFSVMs and FSVMs.

As mentioned above, the reduced training set including the convex points is used to train FFSVMs. So we can obtain the following steps: firstly, the convex points of each class data are searched by using the quick hull algorithm [13]. Secondly, the above FFSVMs are formed on the reduced training set containing the convex points.

The complexity analysis of quick hull is an open problem. By algorithm, the performance of quick hull algorithm is similar to the performance of the randomized incremental algorithms. This is significantly better than quick hulls worst case complexity which is bounded by m iterations that partition m points into the maximum number of facets for m vertices or \(O(m^2)\), where m is the number of the class data which are ready to construct the convex hull. Rcalling the complexity of SVMs, we find it is related to the total number of training data, but the class data for the convex hull are the part of the whole training data. From the analysis of the FFSVMs, we can see the complexity of the FFSVMs is \(O(m^2)\), where \(m < i\) is the maximal value of the numbers of class data, \(t << m\) is the number of the reduced training set.

IV. COMPARISON OF FFSVMs AND FSVMs

To verify the performance of FFSVMs, experiments were performed using the two-class datasets designed for testing the velocity of FFSVMs in the paper and some multi-class problem datasets in machine learning databases, such as thyroid, iris, and car. As for the selection of parameter, we select the constant C to make the maximal accuracy of FSVMs. In all the tables, poly i denotes the polynomial kernel when order \(d = i\), and RBF j denotes the RBF kernel when width \(\sigma = j\).

The two-class datasets are used to compare the FFSVMs with the traditional FSVMs. The positive training data satisfy the normal distribution with the mean [1 1] and the covariance matrix \(
\begin{pmatrix}
1 & -0.05 \\
-0.05 & 1 \\
\end{pmatrix}
\)
, and the negative ones satisfy the normal distribution with the mean [4 4] and the same covariance matrix. In order to verify the influence of the amounts of the training data on the speed of learning machines, the incremental numbers of the training data are generated, such as 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000 training data consisting of the same numbers of the positive data and negative ones. Figure 2 shows the relationships between the scale of the training set and the running time for FFSVMs and FSVMs, the running time of FFSVMs includes the preprocessing time of seeking the convex hull and the training time on the reduced training set. With the increasing of the number of the training data, the training time of FSVMs increases too, and the running time of FFSVMs is relatively stable, and focuses on the interval [0.015625, 0.03125] because FFSVMs select the convex points to train the learning machines. The performances of the two kinds of FSVMs are listed in table I, such as the size of the training set (N), the numbers of the training data (Ntr) who participate the training process actually and the support vectors (Nsv), and the training accuracies (Tr(%)). From the first row to the eighth one, we can see the FFSVMs can achieve the same training accuracies as that of the FSVMs, the FFSVMs have the less training accuracies that than that of FSVMs for the training data listed in the last two low are the non-linearly separable case. We can classify them by tuning the kernel function, such as the polynomial function and RBF. During the training process, the convex data points in FFSVMs are regarded as the very important ones and assigned the same penalty value C, the membership function of FSVMs is defined through the distance between the training data and their class center, namely, the FFSVMs are the typical SVMs on the reduced training set for two-class problem.

The multi-class problem datasets, such as thyroid, iris, and car, are used to verify the extension ability of FSVMs. The dataset thyroid includes 215 5-dimensional inputs and 3-class outputs, the dataset iris consists of 150 4-dimensional inputs and 3-class target values, and the dataset car is composed of 1728 6-dimensional inputs and 4-class target values. The reduced proportion of the dataset thyroid and iris is nearly 50%, and the FFSVMs can still achieve the same extension ability as the FSVMs shown in the table II. For the dataset car, the FFSVMs is faster than the FSVMs, the running time is 1.8906 and 63.875 for RBF respectively, and have the lower training accuracies than the traditional FSVMs. We can improve the performance by adding the data near to the convex points during training process.

---

Table I: The Comparison of FFSVMs and FSVMs with the Same Parameters C=5000, Dot Kernel

<table>
<thead>
<tr>
<th>N</th>
<th>Ntr</th>
<th>Nsv</th>
<th>Tr(%)</th>
<th>FFSVMs</th>
<th>FSVMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>19</td>
<td>3</td>
<td>100</td>
<td>200</td>
<td>3</td>
</tr>
<tr>
<td>400</td>
<td>20</td>
<td>3</td>
<td>100</td>
<td>400</td>
<td>3</td>
</tr>
<tr>
<td>600</td>
<td>22</td>
<td>5</td>
<td>99.667</td>
<td>600</td>
<td>5</td>
</tr>
<tr>
<td>800</td>
<td>22</td>
<td>3</td>
<td>100</td>
<td>800</td>
<td>3</td>
</tr>
<tr>
<td>1000</td>
<td>18</td>
<td>7</td>
<td>99.9</td>
<td>1000</td>
<td>7</td>
</tr>
<tr>
<td>1200</td>
<td>18</td>
<td>3</td>
<td>100</td>
<td>1200</td>
<td>3</td>
</tr>
<tr>
<td>1400</td>
<td>25</td>
<td>3</td>
<td>100</td>
<td>1400</td>
<td>3</td>
</tr>
<tr>
<td>1600</td>
<td>24</td>
<td>99.938</td>
<td>1600</td>
<td>4</td>
<td>99.938</td>
</tr>
<tr>
<td>1800</td>
<td>22</td>
<td>3</td>
<td>100</td>
<td>1800</td>
<td>3</td>
</tr>
<tr>
<td>2000</td>
<td>22</td>
<td>7</td>
<td>99.9</td>
<td>2000</td>
<td>7</td>
</tr>
</tbody>
</table>

Table II: The Comparison of FFSVMs and FSVMs on Multi-Class Problems

<table>
<thead>
<tr>
<th>datasets</th>
<th>kernel</th>
<th>Classifiers</th>
<th>Ntr</th>
<th>Tr(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>thyroid</td>
<td>Dot</td>
<td>FFSVMs</td>
<td>119</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Poly4</td>
<td>FFSVMs</td>
<td>215</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>RBF5</td>
<td>FFSVMs</td>
<td>119</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>FFSVMs</td>
<td>FSVMs</td>
<td>215</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>FFSVMs</td>
<td>FSVMs</td>
<td>75</td>
<td>98.667</td>
</tr>
<tr>
<td></td>
<td>FSVMs</td>
<td>FSVMs</td>
<td>150</td>
<td>98.667</td>
</tr>
<tr>
<td></td>
<td>Poly2</td>
<td>FFSVMs</td>
<td>75</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>FSVMs</td>
<td>FSVMs</td>
<td>150</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>RBF5</td>
<td>FFSVMs</td>
<td>75</td>
<td>98.667</td>
</tr>
<tr>
<td></td>
<td>FSVMs</td>
<td>FSVMs</td>
<td>150</td>
<td>98</td>
</tr>
<tr>
<td>iris</td>
<td>Dot</td>
<td>FFSVMs</td>
<td>313</td>
<td>79.34</td>
</tr>
<tr>
<td></td>
<td>Poly2</td>
<td>FFSVMs</td>
<td>1128</td>
<td>88.079</td>
</tr>
<tr>
<td></td>
<td>RBF5</td>
<td>FFSVMs</td>
<td>313</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>FSVMs</td>
<td>FSVMs</td>
<td>1728</td>
<td>100</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

In this paper, we present FFSVMs based on the convex hull. By computer simulations using two-class datasets and multi-class benchmark data sets in machine learning field, we demonstrate the superiority of FFSVMs. The training speed of FFSVMs can be accelerated avoiding the parameter selection during the preprocessing. The FFSVMs can achieve the same or acceptable classification accuracies rapidly. For higher dimensional spaces, the process of seeking the convex points will increase the time consumption because a great deal of facets must be found according to their visible properties. In the other hand, how to add the data points near the convex points and improve the performance of FFSVMs are the further researches.

ACKNOWLEDGMENT

This work was in part supported by the 973 Program (Grant No. 2004CCA02500), NSFC (Grant No. 60572015, 40701153) and Natural Science Foundation of the Education Department of Henan Province.

REFERENCES