EVALUATION OF FACTORS AFFECTING SUPPORT VECTOR MACHINES FOR HYPERSPECTRAL CLASSIFICATION

Pakorn Watanachaturaporn
Pramod K. Varshney
Electrical Engineering and Computer Science Department, Syracuse University, Syracuse, NY 13244
{pwatanac, varshney}@ecs.syr.edu

Manoj K. Arora
Department of Civil Engineering, I I T Roorkee, Roorkee, 247 667, India
manojfce@iitr.ernet.in

ABSTRACT
Remote sensing data are attractive for deriving land cover information through image classification. A number of parametric and non-parametric classifiers such as the maximum likelihood classifier (MLC) and the artificial neural network (ANN) have been developed and tested successfully on multispectral data. However, the existing classifiers have shown marked limitations in the classification of hyperspectral images obtained from sensors such as AVIRIS, HYMAP, HYDICE and MODIS. Recently, Support Vector Machine (SVM) has been proposed as an alternative for classification of both multi and hyperspectral data. SVM is a machine learning algorithm that employs an optimizer to identify an optimal separating hyperplane to discriminate two classes of interest. The results from a few studies on the use of SVM for remote sensing image classification are promising and encouraging. However, there are several issues that need to be considered and investigated before SVM becomes operational in remote sensing applications. This paper presents some results from an ongoing study on SVM based classification of remote sensing data. The aim is to investigate the effect of some factors on the accuracy of SVM classification. The factors considered are selection of multiclass method, choice of the optimizer and the type of kernel function. The results show that among different multiclass methods, optimizers and kernel functions, classification performed with Directed Acyclic Graph multiclass method using the RBF kernel function produced the highest accuracy of 97% when Lagrangian SVM is used as the optimizer.

INTRODUCTION
Land cover is one of the crucial elements for scientific research and real-life earth science applications (Campbell, 1996; Sellers et al., 1995). For many years, global, national, and regional managers and planners have recognized the importance of land cover for a variety of development of activities as it has also been used as a fundamental variable in several fields such as agriculture, environment, forestry, geology, and hydrology. Due to the large scale proliferation of remote sensing data, they have become attractive sources of land cover information. Several classification algorithms have been developed and successfully implemented to produce land cover classification from multispectral data. However, these algorithms fail to deliver high accuracies for classifying hyperspectral images (Campbell, 1996; Jensen, 1996; Landgrebe, 2002; Richards & Jia, 1999). Classification algorithms may be grouped into parametric and non-parametric techniques. For parametric classifiers, such as the maximum likelihood classifier (MLC), the data are assumed to follow a statistical distribution to estimate accurate parameters, which in turn depends on the selection of appropriate training sample size. In case of hyperspectral data acquired in several dimensions, these classifiers suffer from the problem of the curse of dimensionality, also referred to as the Hughes phenomenon (Hughes, 1968; Landgrebe, 2002). As a result, non-parametric classifiers such as neural networks and decision trees have often been implemented to classify hyperspectral data. In fact, neural network classifiers, particularly the one based on the back propagation algorithm, have been viewed as a substitute for the most widely used MLC. Though neural networks have an advantage as they can work with complex data sets, they are slow during the training phase. Moreover, they work as a black box in which the internal architecture is hidden. Recently, Support Vector Machines (SVMs), a machine learning algorithm, have been proposed that can overcome the limitations of the other non-parametric classifiers.
SVMs have originated from statistical learning theory pioneered by Boser et al. (1992) in the middle of the 1990’s. Statistical learning theory establishes bounds on the error rate of a learning machine on unseen data, called the generalization error rate. These bounds are a function of the sum of the training error rate and the terms that measure classifier complexity. To minimize the bounds on the generalization error rate, both the sum of the training error rate and the classifier complexity must be minimized. Vapnik (1998) shows that the bounds on the generalization error rate can be minimized by explicitly maximizing the margin of separation. Consequently, a better classification performance on unseen data can be expected, and thus high generalization can be achieved. Moreover, since the margin of separation is not dependent on the dimensionality of the data, accurate classification from high dimensional data is possible.

SVMs aim to maximize the margin between two classes of interest and place a linear separating hyperplane between them. Moreover, SVMs can adapt themselves to become a nonlinear classifier by simply mapping data into a higher dimensional feature space that spreads the data out. This is equivalent to a nonlinear classifier in the original input space. Because SVMs can adequately classify data in a higher dimensional feature space with a limited number of training data sets, it overcomes the Hughes Phenomenon.

This paper is organized in the following manner. We first briefly discuss the theory of SVMs for a binary classification problem. We then discuss three popular methods to construct multiclass SVM based classifiers from a number of binary SVM based classifiers. After that we describe optimizers that can be used to find support vectors. Since there are many issues related to the construction of a SVM classifier, we then discuss the parameters that affect the classification performance of an SVM. Afterwards, we present the results of our experiments on the classification of a hyperspectral image acquired from the AVIRIS sensor.

**SUPPORT VECTOR MACHINES (SVMs)**

In this section, a brief introduction on how to construct an SVM is presented. More details can be found in many publications (e.g., Boser et al., 1992; Cortes & Vapnik, 1995; Gualtieri et al., 1999; Vapnik, 1998).

![Figure 1](image.png)

Figure 1. Illustration of an SVM construction in a two dimensional feature space. Dashed lines pass through the support vectors defined by data in circles forming the boundary of classes.

Consider a binary classification problem, where the given dataset is partitioned into two classes with a linear hyperplane separating them (Figure 1). Assume that the training dataset consists of $k$ training samples represented by $(x_1, y_1), \ldots, (x_k, y_k)$, where $x_i \in \mathbb{R}^N$ is an $N$-dimensional data vector with each sample belonging to either of the two classes labeled as $y_i \in \{-1, +1\}$. The goal of SVMs is to find a linear decision function defined by $f(x) = w \cdot x + b$, where $w \in \mathbb{R}^N$ determines the orientation of a discriminating hyperplane, and $b \in \mathbb{R}$ is a bias. The hyperplanes for the two classes are, therefore, represented by $y_i(w \cdot x + b) \geq 1$. Sometimes, due to the noise or mixture of classes introduced during the selection of training data, variables $\xi_i > 0$, called slack variables, are used to account for the effects of misclassification. The hyperplanes for the two classes then become $y_i(w \cdot x + b) \geq 1 - \xi_i$. The optimal hyperplane (i.e., $f(x) = 0$) is located where the margin between two classes of interest is maximized and the error is minimized. This can be achieved by solving the following constrained optimization problem,
Minimize \[ \frac{1}{2} \|w \|^2 + C \sum_{i=1}^{k} \xi_i, \] (1)

Subject to: \[ y_i (w \cdot x + b) \geq 1 - \xi_i, \quad \text{for } i = 1, 2, ..., k. \]

The constant \( 0 < C < \infty \), called the penalty value or \( C \) value, is a regularization parameter. It defines the trade-off between the number of misclassifications in the training data and the maximization of the margin. In practice, the penalty value is selected by trial and error.

The constrained optimization problem in (1) is solved by the method of Lagrange multipliers. The equivalent optimization problem becomes,

\[ \text{Maximize} \sum_{i=1}^{k} \alpha_i - \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j), \]

Subject to: \( \sum_{i=1}^{k} \alpha_i y_i = 0 \) and \( 0 \leq \alpha_i \leq C, \quad \text{for } i = 1, 2, ..., k. \) (2)

In (2), \( \alpha_i \geq 0 \) are the Lagrange multipliers. The solution of the optimization problem given in (2) is obtained in terms of the Lagrange multipliers \( \alpha_i \). According to the Karush-Kuhn-Tucker (KKT) optimality condition (Fletcher, 1987), some of the multipliers will be zero. The multipliers that have nonzero values are called the support vectors. The result from the optimizer, called an optimal solution, is the set \( \alpha^* = (\alpha_1^*, ..., \alpha_k^*) \). The value of \( w \) and \( b \) are calculated from \( w^* = \sum_{i=1}^{k} y_i \alpha_i^* x_i \) and \( b^* = \frac{1}{2} [w^* \cdot x_i^* + w^* \cdot x_j^*] \), where \( x_i^* \) and \( x_j^* \) are the support vectors of class labels \( +1 \) and \( -1 \) respectively. The decision rule is then applied to classify the dataset into two classes viz. \( +1 \) and \( -1 \),

\[ f(x) = \text{sign} \left( \sum_{\text{support vector}} y_i \alpha_i^* (x_i \cdot x) + b^* \right) \] (3)

where \( \text{sign}(\bullet) \) is the signum function. It returns \( +1 \) if the element is greater than or equal to zero and \( -1 \) if it is less than zero.

There are instances where a linear hyperplane cannot separate classes without misclassification; however, those classes can be separated by a nonlinear separating hyperplane. In this case, data may be mapped to a higher dimensional space with a nonlinear transformation function. In the higher dimensional space, data are spread out, and a linear separating hyperplane may be found. This concept is based on Cover’s theorem on the separability of patterns (Cover, 1965). Figure 2 illustrates that two classes in the input space may not be separated by a linear separating hyperplane. However, when the two classes are mapped by a nonlinear transformation function, a linear separating hyperplane can be found in the higher dimensional feature space.

Let a nonlinear transformation function \( \phi \) map the data into a higher dimensional space. Suppose there exists a function \( K \), called a kernel function, such that,

\[ K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \] (4)

A kernel function is substituted for the dot product of the transformed vectors, and the explicit form of the transformation function \( \phi \) is not necessarily known. Further, the use of the kernel function is less computationally intensive. The formulation of the kernel function from the dot product is a special case of Mercer’s theorem (Mercer, 1909; Schölkopf & Smola, 2002). The optimization problem then becomes,
Maximize $\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$, 

Subject to: $\sum_{i=1}^{m} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$, for $i=1,2,\ldots,k$.

The decision function becomes,

$$f(x) = \text{sign} \left( \sum_{\text{support vector}} y_i \alpha_i K(x, x_i) + b^o \right)$$

Examples of some well-known kernel functions are provided in Table 1.

<table>
<thead>
<tr>
<th>Kernel Function</th>
<th>Definition</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$x \cdot x_i$</td>
<td></td>
</tr>
<tr>
<td>Polynomial with degree $d$</td>
<td>$(x \cdot x_i + 1)^d$</td>
<td>$d$ is a positive integer</td>
</tr>
<tr>
<td>Radial Basis Function</td>
<td>$\exp \left( -\frac{</td>
<td></td>
</tr>
<tr>
<td>Sigmoid</td>
<td>$\tanh (\kappa (x \cdot x_i) + \Theta)$</td>
<td>$\kappa$ and $\Theta$ are user defined values</td>
</tr>
</tbody>
</table>

**MULTICLASS CLASSIFICATION**

SVMs were originally developed to perform binary classification. However, classification of data into more than two classes, called multiclass classification, is more practical in remote sensing applications such as land use land cover classification. A number of methods have been proposed to create multiclass SVMs from binary counterparts. Details of various multiclass classification methods with their merits and demerits can be found in Pal and Watanachaturaporn (2004). In this section, we will explain the three most popular methods.

**One against the rest classification**

This method is also known as *winner-take-all* classification. For an $M$ class classification, $M$ binary SVM classifiers are created. Each classifier is trained to discriminate one class from the remaining $M-1$ classes. During the testing or application phase, data are classified by computing the margin from the linear separating hyperplane; i.e., the output from the equation (3) or (6) without the signum function. Data are assigned to the class labels of the SVM classifier that produces the maximal output.

**Pairwise classification**

In this method, SVM classifiers for all possible pairs of classes are created. For an $M$ class classification, we create $M(M-1)/2$ binary classifiers. Each binary classifier is trained to classify two classes of interest. During the testing or application phase, the output from each binary classifier in the form of a class label is obtained. The class label that occurs the most is assigned to that data. We adopt a tie-breaking strategy in case of a tie. A common tie-breaking strategy is to randomly select one of the class labels that are tied.

**Classification based on Directed Acyclic Graph (DAG)**

This method is based on the *Decision Directed Acyclic Graph* structure that has a tree-like structure. Similar to the pairwise classification method, we create $M(M-1)/2$ binary classifiers for an $M$ class classification. Each binary classifier is trained to distinguish two classes and forms a node in the graph structure. Nodes are organized in the form of a triangle with the single root node at the top and increasing subsequently in an increment of one node in each level until the last level that will have $M$ nodes (see Figure 3).

The DAG evaluates an input data starting at the root node (top node) and moves to the next level based on the output values. The binary classifier in the next level then evaluates the input data. The path traversed by data is called the *evaluation path*. The DAG method eliminates one class out from the list at each level. At the root node, all classes are in the list. Each node discriminates between the first class and the last class in the list. Each level gives the result in one class out of the two classes; the class that is not in favor of that level is eliminated from the list. The procedure is terminated when only one class remains in the list. Although, here the number of binary classifiers
equals the number of classifiers required by the pairwise classification method, inputs are evaluated only $M - 1$ times resulting in faster classification.

**Figure 3.** The Directed Acyclic Graph SVM.

### OPTIMIZATION METHODS

One of the key processing steps in the development of SVM algorithms is to employ an optimizer to find the support vectors. A variety of optimization methods may be used. Typically, the conventional SVMs have used an optimizer based on quadratic programming (QP) or linear programming (LP) methods to solve the optimization problem. A number of optimization packages are available such as CPLEX (CPLEX Optimization Inc, 1992), MINOS (Murtagh & Saunders, 1987), and LOQO (Vanderbei, 1997). The major disadvantage of QP algorithms is the storage requirement of the kernel data matrix in the memory. For large datasets such as hyperspectral data, the size of the kernel data matrix becomes large that requires huge memory, which may not always be available. Alternatives that work on a smaller or a subset of the whole dataset are **chunking** and **decomposition** methods. These methods are based on the assumption that the number of support vectors is quite small in comparison to the total number of training samples. Instead of sequentially updating the Lagrange multipliers, these methods update many parameters in each iteration unlike other methods that update one parameter at a time. In the chunking method, QP is used to optimize an initial arbitrary subset of data. The support vectors found by the optimizer from this subset are kept while other data points are discarded. A new subset is then selected based on these support vectors and the additional data. The process is then repeated until the margin is maximized. However, the chunking method fails when the training dataset is too large or most of the training data points become support vectors. The decomposition method works on the fixed size working dataset rather than on an arbitrary size as in the chunking method. A QP optimizer updates the Lagrange multipliers on the fixed size working dataset only. The other Lagrange multipliers that are not in the working set are kept fixed. The **Sequential Minimal Optimization** (SMO) algorithm (Platt, 1999) is a special case of the decomposition method when the size of the working dataset is fixed at two such that an analytical solution is derived. This discards the use of QP and LP optimizers. This method needs more number of iterations but requires a small number of operations thus resulting in an increase in computational speed for very large data sets.

There are a number of other alternative optimization methods too. For example, Mangasarian and Musicant (2000) proposed the Lagrangian SVM (LSVM) that reformulates the constrained optimization problem as an unconstrained optimization problem. The problem is solved through an optimizer based on the system of linear equalities. The LSVM can work with linear and nonlinear kernels. For the linear kernel, the LSVM uses the Sherman-Morrison-Woodbury formula to increase the computation speed.

Thus, the application of SVMs to any classification problem requires the proper selection of design parameters related to several issues discussed above. Some of these considered here are: determination of appropriate penalty value to regulate the error term, choice of an appropriate kernel and its parameters, selection of a suitable multiclass method, and decision on the type of optimizer to be used.
EXPERIMENTAL DATA

In this experiment, a hyperspectral image acquired from AVIRIS available at (ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/92AV3C) has been used. One of the bands of this image is shown in Figure 4. The image was acquired on June 12, 1992 over the northern part of Indiana. Two-thirds of the scene is covered with agricultural land while one-third is forest and others. Two major dual lane highways, a smaller road, a rail line, low density housing, and other building structures can also be seen in the scene. A field surveyed map consisting of sixteen classes and one unclassified class is also available (Figure 5) and has been used as reference data (ground data) for training and testing data collection and accuracy assessment. The availability of reference data makes this hyperspectral image an excellent source for conducting experimental studies, and, therefore, this image has been used in many earlier studies (e.g., Gualtieri et al., 1999; Tadjudin & Landgrebe, 1998). The size of the 224 bands image is $145 \times 145$ pixels. Four of the bands do not contain any data. In our experiments, similar to the other studies, twenty water absorption bands numbered [104 – 108], [150 – 163], and 220 were removed from the original image. In addition, fifteen noisy bands [1 – 3], 103, [109 – 112], [148 – 149], [164 – 165], and [217 – 219], as observed from visual inspection, were also discarded. A number of training and testing pixels for each class were randomly selected and are given in Table 2.

![Figure 4. A single band AVIRIS image (band 100 [1.3075-1.3172 µm])](image1)

![Figure 5. Reference data](image2)

Table 2. The number of training and testing pixels from the AVIRIS image

<table>
<thead>
<tr>
<th>Class</th>
<th>Number of training pixels</th>
<th>Number of testing pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfalfa</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Corn-notill</td>
<td>117</td>
<td>117</td>
</tr>
<tr>
<td>Corn-min</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>Corn</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Grass/pasture</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>Grass/trees</td>
<td>58</td>
<td>58</td>
</tr>
<tr>
<td>Grass/pasture-mowed</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Hay-windrowed</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>Oats</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Soy-notill</td>
<td>115</td>
<td>116</td>
</tr>
<tr>
<td>Soy-mintill</td>
<td>146</td>
<td>145</td>
</tr>
<tr>
<td>Soy-clean</td>
<td>45</td>
<td>46</td>
</tr>
<tr>
<td>Wheat</td>
<td>25</td>
<td>24</td>
</tr>
<tr>
<td>Woods</td>
<td>104</td>
<td>104</td>
</tr>
<tr>
<td>Bldg-grass-trees-drives</td>
<td>40</td>
<td>41</td>
</tr>
<tr>
<td>Stone-steel towers</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>916</strong></td>
<td><strong>915</strong></td>
</tr>
</tbody>
</table>

SVM BASED CLASSIFICATION

In this section, we show the use of the SVM based classifier to produce classification from the AVIRIS image. The effect of a number of factors on the accuracy of classification produced by the SVM based classifier has also
been investigated. These factors are: selection of multiclass method, choice of optimizer, and type of the kernel function. In all experiments, we assess the accuracy using the overall accuracy obtained from an error matrix (Congalton & Green, 1999).

**Multiclass Classification Experiment**

Three multiclass classification methods have been employed to investigate the effect of each on classification accuracy. These three methods are: one against the rest, pairwise classification, and directed acyclic graph (DAG). A number of classifications using the Lagrangian Support Vector Machine and the linear kernel with different C values have been produced. The accuracy values and the training time spent for each classification are shown in Figure 6 and Figure 7 respectively.

![Multiclass SVM using Linear Kernel](image1)

**Figure 6.** Overall accuracy of multiclass approaches applied on the hyperspectral image

![Multiclass SVM using Linear Kernel](image2)

**Figure 7.** Training time spent to classify the hyperspectral image

It can be seen from Figure 6 that the accuracy obtained from the pairwise and DAG methods are significantly higher than the one against the rest method. Both pairwise and DAG methods show a similar trend with accuracy reaching 95%, even for a small value of C whereas for one against the rest method the maximum achievable accuracy is 80%. In all the methods, as the value of C is increased, the accuracy increases and then drops after reaching a maximum value. This shows that there is an optimum C value for the classification of a dataset.

The training time for the pairwise classification and DAG methods are alike since they result into creation of 120 binary classifiers whereas the one against the rest method requires only 16 binary classifiers. Even though they create more binary classifiers than the one against the rest method, they take less time since the training data of each binary classifier are much less than the other method.

**Choice of Optimizer**

The selection of an optimizer is important to implement an SVM based classifier. Though there are many optimizers that can be used, we have considered only the widely used chunking-decomposition (CD) based optimizer (Chang & Lin, 2002) and the Lagrangian Support Vector Machine (LSVM) method (Mangasarian &
Musicant, 2000). We use the pairwise approach and the linear kernel with different C values for the experiments in this section. The accuracy results, the training time, and the testing time of both optimizers are shown in Figure 8, Figure 9, and Figure 10 respectively.

Figure 8. Overall accuracy when using different optimizers

Figure 9. Training time when using different optimizers

Figure 10. Testing time when using different optimizers

It is shown that the LSVM performs slightly better than the CD optimizer for C values up to 1. After that the classifications performed by both optimizers attain the same accuracy with the best accuracy of 95% occurring when C value is between 1 and 1000. For larger C values, the accuracy of the CD optimizer remains at the same level while it drops for the LSVM optimizer. In terms of training and testing time, both optimizers take less time and are close to each other up to a C value of 5, where the accuracy reaches its maximum. After that the LSVM takes longer time than the CD optimizer and does not improve the overall accuracy.
It appears that there is no significant difference between the classification accuracy obtained by the use of two optimizers. Either of these can be used to construct an SVM based classifier when applying to hyperspectral data.

**Effect of Kernel Function**

The choice of the kernel function also plays an important role in SVM based classification. Four type of kernels, the linear kernel, the polynomial kernel, the RBF kernel, and the sigmoid kernel have been investigated. Five polynomial kernels with degrees varying from 2 to 7 have been used. Therefore, we investigate the effect of 9 kernel functions on the accuracy of classification implemented by using the LSVM optimizer and the pairwise multiclass method for different C values. The variation in classification accuracy is depicted in Figure 11 to Figure 13.

![Figure 11. SVM performance using the linear, RBF, and Sigmoid kernels](image1)

![Figure 12. SVM performance using the polynomial kernels of degrees 2 to 4](image2)

![Figure 13. SVM performance using the polynomial kernels of degrees 5 to 7](image3)
Except the polynomial kernels of degree 6 and 7, all other kernels achieved a maximum accuracy of more than 90%. The maximum accuracy occurs at different C values for each kernel. For example, the linear kernel can obtain a maximum accuracy of 95% at a certain C value; however, it drops dramatically with any further increase in the C value. For the RBF and Sigmoid kernels, the maximum accuracy occurs at very high values of C. For the polynomial kernel, an accuracy of 92% is obtained with a C value as low as $10^{-5}$ for the polynomial kernel of degree 5. The polynomials of lower degree obtained higher than 92% accuracy with the higher C values. Therefore, a trade-off between the C value and the degree of polynomial may be maintained to get maximum classification accuracy for the hyperspectral data by using searching methods (see Watanachaturaporn & Arora, 2004). Since an increase in C value is directly proportional to the training time required for an optimizer (as observed in the previous section), any kernel function, which can produce the highest accuracy at lower C values is preferred to perform efficient classification. This is important especially with hyperspectral data.

CONCLUSIONS

In this paper, we have evaluated the application of SVMs for the classification of hyperspectral data. We have also investigated the effect of three major factors on the accuracy of SVM based classification. It is clear from the results that, with an appropriate selection of the multiclass method, optimizer, and the kernel function, an accuracy of the order of 97% can be achieved from SVM classification. We have shown that the multiclass method used for the classification contributes greatly to provide the best classification accuracy, and there is an optimum C value for the classification of a certain dataset. Although the two optimizers yield the same performance, LSVM takes a longer time for higher values of C. We have also demonstrated that the SVM classification performed with the RBF kernel and polynomial kernel of degree 2 performed the best with an accuracy of 97% whereas the best accuracy achieved by other kernels ranged between 70% and 96%.

ACKNOWLEDGEMENT

The work was supported by NASA under grant number NAG5 – 11227

REFERENCES


