Two Sequential Stages Classifier for Multispectral Data

Refaat M Mohamed and Aly A Farag

Computer Vision and Image Processing Laboratory
University of Louisville, Louisville, KY 40292.

Email: {refaat, farag}@cvip.uofl.edu
www.cvip.uofl.edu

Abstract- In this paper, we present an approach for the classification of remote sensing multispectral data, which consists of two sequential stages. The first stage exploits the capabilities of the Support Vector Machines (SVM) approach for density estimation and uses it in a Bayes classification setup. In a typical image, the class of a pixel is highly dependent on the classes of its neighbor pixels. The second stage of our classifier applies for this dependency of the class. We incorporate this dependency using stochastic modeling of the context as a Markov Random Field (MRF). The MRF is modeled using Besag model and implemented using the Iterative Conditional Modes (ICM) algorithm. Results show that the stochastic modeling approach enhances the results and provides reasonable smoothness in the classified image.

1. Introduction

The site labeling problems involve classification of each site (a pixel, an edge, or a region) into a certain number of classes based on an observed value at each site. Bayes classifier constitutes the basic setup for a large category of classifiers. To implement the bayes classifier, there are two issues to be addressed [1]. The first issue is the estimation of the class conditional probability for each class and the second issue is the a priori probability of each class.

Support Vector Machines (SVM) were developed to solve the classification problem, but recently have been extended to regression problems [2]. SVM have been shown to perform well for density estimation where the probability distribution function of the feature vector $x$ can be inferred from a random sample $D$. SVM represent the data sample $D$ by a few number of support vectors and the associated kernels [3].

For solving the site labeling problems, the basic implementation of Bayes classifier treats each site in the scene independently, i.e. it does not exploit the contextual information. Whereas contextual information plays an important role because the true label of a site should be compatible with the labels of the neighboring sites. Markov Random Fields (MRFs) are appropriate models for the context because they can be used to specify this spatial dependency or spatial distribution [4].

In this paper, 1) we illustrate the application of SVM for density estimation in different dimensions. 2) As an application, in seven-dimensional space we estimate the probability density function of the feature $x$ of multispectral Landsat data. This estimator is used in a Bayes classification setting, which represent the first stage of the proposed classifier. 3) In the second stage, we incorporate contextual dependency by a Markov Random Field (MRF) using ICM algorithm, to enhance those results from the Bayes classifier in the first stage.

2. Support Vector Machines for Density Estimation

Support Vector Machines (SVM) have been developed by Vapnik [5] and are gaining popularity due to many attractive features and promising empirical performance. The formulation embodies the Structural Risk Minimization (SRM) principle, which has been shown to be superior to traditional Empirical Risk Minimization (ERM) principle employed in conventional learning algorithms (e.g. neural networks). SRM minimizes an upper bound on the generalization error, as opposed to ERM, which minimizes the error on the training data. It is this difference, which makes SVM more attractive in statistical learning applications. In this section, we present a brief outlines of the SVM approach for density estimation. Details of the algorithm can be found in [7].
2.1 The Density Estimation Problem

The probability density function, \( p(x) \), of the random vector \( x \) is an nonnegative quantity which is defined as:

\[
F(x) = \int_{-\infty}^{x} p(x') \, dx'
\]

where \( F(x) \) is the cumulative distribution function. Hence, by definition, in order to estimate the probability density, we need to obtain a solution of the integral equation:

\[
\int_{-\infty}^{x} p(x', \alpha) \, dx' = F(x)
\]

on a given set of densities \( p(x, \alpha) \), where the integration in (2) is a vector integration, and \( \alpha \) is the set of parameters to be determined.

The estimation problem in (2) can be regarded as solving the linear operator equation:

\[
A \, p(x) = F(x)
\]

where the operator \( A \) is a one-to-one mapping for the elements \( p(x) \) of the Hilbert space \( E_1 \) into elements \( F(x) \) of the Hilbert space \( E_2 \). This regression problem is solved in the image space (right hand side of (3)) and this solution can be used to describe the solution in the pre-image space (before the operator \( A \) is applied). Unfortunately, the distribution function \( p(x) \) is usually unknown.

However, if we have a random sample from the distribution, \( D = \{x_1, x_2, ..., x_N\} \), a reasonable approximation for \( F(x) \) can be obtained by:

\[
F_N(x) = \frac{1}{N} \sum_{k=1}^{N} \mathbb{I}(x-x_k)
\]

where \( \mathbb{I}(a) \) is the indicator function. We define the following error measure in order to characterize the estimate in (4):

\[
\varepsilon_k = \lambda \sigma_k = \lambda \sqrt{\frac{1}{N} F_N(x_k)(1-F_N(x_k))}
\]

where \( \lambda \) is usually chosen to be 1. Therefore we can construct the estimation (3) with the triplets:

\[
(x_1, F_N(x_1), \varepsilon_1), ..., (x_N, F_N(x_N), \varepsilon_N)
\]

The SVM is used to find a solution for the regression problem (3) using these data triplets.

2.2 Density Estimation as an ill-posed problem

The problem of solving the operator equation (3) is well-posed if a solution exits which is unique and stable. The problem of density estimation is known to be ill-posed because it violates the stability condition; a small change in the cumulative distribution function \( F(x) \) can cause large changes in the derivative, the density function \( p(x) \).

One solution for the ill-posed problems is to introduce a semi-continuous, and positive functional \( \Lambda(p(x)) \leq c ; c > 0 \) in a Reproducing Kernel Hilbert Space (RKHS). Also, we define \( p(x) \) as a trade-off between \( \Lambda(p(x)) \) and \( \| Ap(x) - F_N(x) \| \). An example for such method is that by Phillips [6]:

\[
\min \Lambda(p(x))
\]

such that:

\[
\| Ap(x) - F_N(x) \|_{E_2} < \varepsilon_N, \varepsilon_N > 0, \varepsilon_N \to 0
\]

2.3 Support Vector Machines

In support vector machines, we look for a solution of the density estimation problem in the form:

\[
p(x) = \sum_{k=1}^{N} \beta_k K(x, x_k)
\]

where \( K(x, x_k) \) is the kernel that defines a RKHS. To use Philips’ method (7) to solve the density estimation (3) in a RKHS, we minimize (7) subject to the constraint:

\[
\max_k \left| F_N(x) - \int_{-\infty}^{x} p(t) \, dt \right| = \varepsilon_k
\]

Taking into account the properties of RKHS, we can write the regularization functional in (7) as:

\[
\Lambda(p(x)) = \langle p(x), p(x) \rangle_H
\]

\[
= \sum_{k=1}^{N} \beta_k K(x, x_k), \sum_{j=1}^{N} \beta_j K(x, x_j) \rangle_H
\]

\[
= \sum_{k=1}^{N} \beta_k \sum_{j=1}^{N} \beta_j (K(x_k, x_j), K(x, x_j))_H
\]

\[
= \sum_{k=1}^{N} \sum_{j=1}^{N} \beta_k \beta_j K(x_k, x_j)
\]

Therefore, to solve the estimation problem in (3) we minimize the functional:
subject to the constraints (10), and:

\[ \beta_k \geq 0, \quad \sum_{j=1}^{N} \beta_j = 1 \]  

(13)

where \( 1 \leq k \leq N \). The constraints in (13) are imposed to obtain the solution in the form of a mixture of densities.

The only remaining issue for the SVM description is the kernel choice. To obtain a solution in the form of a mixture of densities, we choose a nonnegative kernel which satisfies the following conditions:

\[ K_\gamma(x, x_j) = a(\gamma)K_\gamma \left( \frac{x-x_j}{\gamma} \right); \quad a(\gamma) \int K_\gamma \left( \frac{x-x_j}{\gamma} \right) dx = 1; \]

and \( K(0) = 1 \)  

(14)

A listing for the implementation steps for SVM as a density estimator can be found in [7]. We apply the above algorithm on data of different dimensions as will be illustrated in section 4.

3. Incorporating Contextual Information using MRF

The brightness level at a point in an image is highly dependent on the brightness levels of neighboring points unless the image is simply random noise. This fact raises the necessity to use contextual or dependency information in segmentation/classification of a scene image. In this section, we explain a model of this dependency, called Markov Random Field (MRF).

A typical outline for the problem of modeling a segmentation problem of multispectral image using MRF is as follows. The observed image (assumed to be a random process), \( G \), is modeled as a composite of two random processes, a high level process \( G^h \) and a low level process \( G^l \), that is, \( G = (G^h, G^l) \). Each of the three processes is a random field defined on the same lattice \( S \). The high level process (the labeling or coloring process) \( G^h \) is used to characterize the spatial clustering of pixels into regions. The low level process (pixel process) \( G^l \) describes the statistical dependence of pixel gray level values in each region. The processes \( G^h \) and \( G^l \) are discrete parameter random fields with state spaces defined as follows:

\[ \Xi^h = \{g^h : g^h \in [c_1, c_R] \} \] and  
\[ \Xi^l = \{g^l : g^l \in [0, q-1] \} \]  

(15)

where \( R \) is the number of regions (with colors or labels \( c_1, c_2, ..., c_R \)) and \( q \) is the number of possible gray levels in a particular region (e.g., 256). An instance of the observed image \( g \) can be described as follows: Consider a region of type \( c_k \), the gray level value at pixel \( s \in S \) of the observed image \( g \) equals that of region type \( c_k \), that is:

\[ g = g^l \quad \text{if} \quad g^h = q_k, \quad s \in S, \quad k \in [1, R] \]  

(16)

The Maximization of the Aposteriori Probabilities (MAP) segmentation involves the determination of \( g^h \) that maximizes \( P(G^h = g^h | G = g) \) with respect to \( g^h \). By Bayes’ theorem,

\[ P(G^h = g^h | G = g) = \frac{P(G = g | G^h = g^h)P(G^h = g^h)}{P(G = g)} \]  

(17)

Since the denominator of (17) does not affect the optimization, the MAP segmentation can be obtained, equivalently, by maximizing the numerator or its natural logarithm. That is, we need to find \( \hat{g}^h \) which maximizes:

\[ \Gamma(G, G^h) = \ln P(G = g | G^h = g^h) + \ln P(G^h = g^h) \]  

(18)

The first term of (18) is the likelihood due to the low level process and the second term is due to the high level process. Based on the models of the high level and low level processes, the MAP estimate can be obtained. In this paper, the low level process is modeled using the SVM method. Whereas, a first order Markov model is used for the high level process. The Iterated Conditional Modes (ICM) algorithm will be used to carry out the optimization in the MAP algorithm.

The ICM is a relaxation algorithm to find a local maximum (mode) [4]. The algorithm assumes that the classes of all neighbors of a pixel (vector) \( x \) are known. The high level process is assumed to be formed of \( R \)-independent processes. We will model each of the \( R \) processes by a first order Markov Random Field. One form of such model is that presented by Besag [4],

\[ P(c_j / b) = \frac{1}{Z} \exp(-\lambda m) \]  

(19)
where \( b \) is the neighbor set, \( m \) is the portion of these neighbors belonging to class \( c_i \). The term \( Z \) is a normalization factor. The parameter \( \lambda \) is the clique potential. The value of \( \lambda \) controls the dependency (e.g. \( \lambda = 0 \) corresponds to no dependence on the neighbors, i.e. Bayes classifier). The larger the value of \( |\lambda| \), the stronger is the dependence on the neighbors, and the more homogeneous results.

If the classes of the neighbors are known, then \( x \) can be classified to the class that maximizes:

\[
P(c_i | x) \propto P(x | c_i) P(c_i | b)
\]

(20)

where, \( P(c_i | b) \) is computed from (19) and the class conditional probability \( P(x | c_i) \) (the low level process) is to be estimated using SVM.

Initially, the classes of the neighbors are not known. Hence, the Bayes classifier (as stated in section 2) is used to give an initial estimate.

This procedure is applied to all the pixels of the image to constitute a single step of the ICM algorithm. The procedure is then iteratively applied until a satisfactory probability of error is attained (A listing of the algorithm steps can be found in [1]).

4. Experimental Results

We carry out several experiments to illustrate the applicability of the various components of the proposed classification set up. First, we illustrate the performance of the SVM as a density estimator using synthetic data in different dimensions. Then, we use the SVM density estimator in a Bayes classifier setup for multispectral data classification. Finally, we apply MRF modeling to incorporate contextual information in the segmentation problem.

4.1 Density Estimation for One- and Two-Dimensional Cases

In this part of the experiments, we use the SVM method to estimate the probability density functions using random samples from a unimodal normal, a bimodal function, and a 2-D normal density functions. To implement the SVM, a Gaussian-like kernel is used:

\[
K(x, x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-0.5\left(\frac{x-x_i}{\sigma}\right)^2\right)
\]

(21)

Fig 1 shows the results of simulating a standard normal density function using SVM. Fig 2 shows a bimodal density function (a triangular part plus a uniform part, see [8] pp. 171). We can see that SVM tried well to approximate the function compared to the results in [8] pp. 171). Fig 3 shows a 2-D normal density with zero-means and unit variances in both dimensions. With a random sample \( D \) of size 100, the SVM estimate captures the shape of the real density function, except the variance of the estimate is more than one in both directions.

4.2 Bayes Classifier Setup

In this section we present the classification results of using the first stage of the proposed setup. Here, we use the SVM as a density estimator in the design of a Bayesian classifier. The used kernel is a 7-D Gaussian-like function similar to that in (14).

The experiments were carried out using real data acquired from the Landsat Thematic Mapper (TM) for two different data sets as will be discussed below.

A. Golden Gate Bay Area

The first part of the experiments is carried out using real multispectral data for the Golden Gate Bay area
of the city of San Francisco, California, USA. A 700x700 image scene of the Golden Gate Bay area is cropped from a 7-band Landsat data set, Fig. 4. Five classes are defined on this image: Trees, Streets (light urban), Water, Buildings (dense urban) and Earth. The available ground truth data for this data set includes about 1000 points per class, which is approximately 1% of the image scene.

The results for this experiment are shown in Fig. 4 and the classification results are summarized in Table (1). The results show that the SVM based classifier achieves excellent, over 95% classification accuracy for Water, Trees and Streets classes. And it performs well, over 91% for Building and Earth classes. The average classification accuracy of the classifier is about 5% higher than other algorithms used before (see [1]).

### B. Agricultural Area

The second part of the experiments was carried out using real multispectral data for an agricultural area, Fig 5. Nine classes are defined within this data set: Background, Corn, Soybean, Wheat, Oats, Alfalfa, Clover, Hay/Grassland, and Unknown. The ground truth data for that area is available. About 1% of ground truth data associated with each class is used for training. The results of this experiment are shown in Fig. 5 and the classification results are summarized in Table 2.

The results illustrate that the SVM based classifier performs well for some classes (e.g., the Unknown and Corn classes), good for some (Wheat, Soybean, Alfalfa and Oats), and fair on other classes (Background, Clover and Hay/Grass). But the main distinguishable feature of the SVM based classifier is that it can distinguish classes (Oats, Alfalfa, Clover, and Hay/Grassland), which are hard for other classifiers (based on other density estimators). The average of classification accuracy is about 5% better than the other methods used [1].

### 4.3 Contextual Information Incorporation

In this part of the experiments we illustrate the effect of incorporating contextual information on the classification results. We model the high level process as a MRF using Besag model (19). The essence of Besag model is that it uses only one parameter $\lambda$. In this paper we use an ad hoc value of 5 for $\lambda$ that maximizes the classification accuracy. The classification results, Table 1 and Table 2, show that using contextual information, the classification results are highly enhanced for individual classes as well as the overall accuracy.

Fig. 4 and Fig. 5, show the effect on the MRF modeling on the smoothness of the classified images. Using Bayes classifier stage only, the classified images are noisy. This is because each pixel is classified independently of its neighbors. But, after applying MRF modeling, the classified images are smooth and the edges of classes are clearer (this is more apparent from Fig. 5).

### Table 1 Classification Results for the golden-Bay area

<table>
<thead>
<tr>
<th>Class</th>
<th>Bayes Classifier</th>
<th>After RMF modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>97.5</td>
<td>99</td>
</tr>
<tr>
<td>Corn</td>
<td>95.6</td>
<td>97</td>
</tr>
<tr>
<td>Soybean</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Wheat</td>
<td>91.8</td>
<td>92</td>
</tr>
<tr>
<td>Oats</td>
<td>91.3</td>
<td>98</td>
</tr>
<tr>
<td>Average</td>
<td>95.7</td>
<td>97.6</td>
</tr>
</tbody>
</table>

### Table 2 Classification Results for agricultural Area

<table>
<thead>
<tr>
<th>Class</th>
<th>Bayes Classifier</th>
<th>After RMF modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>50.4</td>
<td>51.08</td>
</tr>
<tr>
<td>Corn</td>
<td>91.5</td>
<td>96.29</td>
</tr>
<tr>
<td>Soybean</td>
<td>77.9</td>
<td>88.39</td>
</tr>
<tr>
<td>Wheat</td>
<td>84.2</td>
<td>93.03</td>
</tr>
<tr>
<td>Oats</td>
<td>72.5</td>
<td>74.25</td>
</tr>
<tr>
<td>Alfalfa</td>
<td>76.9</td>
<td>93.85</td>
</tr>
<tr>
<td>Clover</td>
<td>69.8</td>
<td>84.81</td>
</tr>
<tr>
<td>Hay/Grass</td>
<td>66.2</td>
<td>82.39</td>
</tr>
<tr>
<td>Unknown</td>
<td>95.7</td>
<td>96.97</td>
</tr>
<tr>
<td>Average</td>
<td>76.1</td>
<td>82</td>
</tr>
</tbody>
</table>

### 5. Conclusion and Future Work

This paper presented a two stage classifier for multispectral data classification. The first stage uses the SVM as an estimator for the class conditional probabilities in a Bayes setup. Then, we model the context as a MRF in the second stage. The results show that the SVM has the capabilities to simulate well different densities in different dimensions. Also, modeling the context as a MRF quite enhances the classification results. For future work, we try some algorithms for enhancing the SVM performance and also to automatically choose the parameters for it (e.g. kernel function, kernel parameters, etc.). For the MRF, we will try to find a way for automatically choosing the parameters of the model and also to use another model rather than Besag model for the sake...
of comparison. Also, one of our current work is that we try to unsupervisely estimate the number of classes in an image.

References


