Machine Learning Algorithms for Spatio-temporal Data Mining

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Ranga Raju Vatsavai

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor Of Philosophy

December, 2008

Machine Learning Algorithms for Spatio-temporal Data Mining

by Ranga Raju Vatsavai

ABSTRACT

Remote sensing, which provides inexpensive, synoptic-scale data with multi-temporal coverage, has proven to be very useful in land cover mapping, environmental monitoring, forest and crop inventory, urban studies, natural and man made object recognition, etc. Thematic information extracted from remote sensing imagery is also useful in variety of spatiotemporal applications. However, increasing spatial, spectral, and temporal resolutions invalidate several assumptions made by the traditional classification methods. In this thesis we addressed four specific problems, namely, small training samples, multisource data, aggregate classes, and spatial autocorrelation. We developed a novel semi-supervised learning algorithm to address the small training sample problem. A common assumption made in previous works is that the labeled and unlabeled training samples are drawn from the same mixture model. However, in practice we observed that the number of mixture components for labeled and unlabeled training samples differ significantly. Our adaptive semi-supervised algorithm over comes this important limitation by eliminating unlabeled samples from additional components through a matching process. Multisource data classification is addressed through a combination of knowledge-based and semi-supervised approaches. We solved the aggregate class classification problem by relaxing the unimodal assumption. We developed a novel semi-supervised algorithm to address the spatial autocorrelation problem. Experimental evaluation on remote sensing imagery showed the efficacy of our novel methods over conventional approaches. Together, our research delivered significant improvements in thematic information extraction from remote sensing imagery.

Acknowledgements

This thesis would not have been possible without great direction, constant encouragement, and guidance of my thesis advisor, Prof. Shashi Shekhar. I am highly thankful for the flexibility he offered, inputs he provided which greatly enhanced my thinking, and attitude towards research. In addition to the research, he provided a congenial environment to interact with group members, external collaborators, and peers, which greatly helped me in doing a multidisciplinary research and establish long-term collaborations.

I am deeply grateful and owe my sincere gratitude to Prof. Tom Burk, who not only mentored me throughout my stay at Minnesota, but also went beyond and helped me in several difficult situations. I would like to thank my committee members, Prof. Jaideep Srivastava, and Prof. Mohamed F. Mokbel for their feedback on my research and thesis.

There are several people without whom I couldn't have conducted this research in a timely manner, people who have helped me in collecting and processing data – Jamie Smedsmo, Ryan Kirk, Tim Mack, Perry Nacionales, Sonja K. Hansen, and Ty Wilson. I would like to thank Prof. Marvin Bauer, Prof. Paul V. Bolstad, Dr. Mark Hansen, Prof. Sanjay Chawla, Prof. Paul Schrater, and Prof. Joydeep Ghosh who have provided lot of inputs into this research. I would like to thank my colleagues, friends, and staff members at the Remote Sensing Laboratory and Spatial Databases Research Group, who helped me in various ways throughout my stay in Minnesota and beyond. I would like to thank Kim Koffolt for improving the readability of this thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Land management organizations and the public have a need for more current regional land cover information to manage resources and monitor land use change. Remote sensing, which provides inexpensive, synoptic-scale data with multi-temporal coverage, has proven to be very useful in land cover mapping, environmental monitoring, and forest and crop inventory. Thematic information extracted from remote sensing imagery is also useful in variety of spatio-temporal applications.

Image classification, i.e., assigning class labels to pixels, using some discriminant function, is one of the fundamental analysis technique used in remote sensing to generate thematic information. Several classification algorithms have been proposed in the literature for analysis of remote sensing imagery. These algorithms can be broadly grouped into two categories, supervised and unsupervised, based on the kind of training data (labeled or unlabeled) used. In supervised learning, analyst collects samples from the image of interest and then assigns these samples to the predefined thematic classes, with the aid of high resolution aerial photographs, ground visits, existing thematic maps and knowledge. These labeled samples are called training data, which are then fed into supervised classification models. Among supervised classification methods, the maximum likelihood classifier is the most extensively studied and utilized for classification of multi-spectral images. Other classification schemes include, but not limited to, neural networks [1, 2, 3, 4], decision trees [5, 6, 7], support vector machines [8, 9], and graphical models. Unsupervised methods include various clustering algorithms [10] such as ISODATA and k-means, CHAMELEON. In remote sensing, clustering algorithms are generally used for initial processing to understand natural groupings in the data and as an aid in training sample collection for the supervised learning.

Many supervised classification schemes proposed in the literature work well if the land cover classes are spectrally separable, and sufficiently large number of training samples were available. But in reality, the classes under investigation are often spectrally overlapping, and accurate training samples were limited. The *i.i.d.* assumption, that is, samples are independent and identically distributed, poses severe problems with spatial datasets, as the neighboring pixels are often exhibit spatial autocorrelation. As a consequence the classified images often exhibit salt-and-pepper kind of noise. These limitations have prompted us to develop new machine learning algorithms for spatio-temporal data mining. This thesis address these four practical and important problems, namely small training samples, multisource data, aggregate classes, and spatial auto-correlation.

1.1 Small training sample problems

Bayes theory is the backbone behind many supervised statistical classification techniques, including popular algorithms, such as maximum likelihood classification (MLC), and maximum a posteriori classification (MAP). Often it is customary to assume Gaussian or exponential family of distributions from which the training samples were generated, and use the classical maximum likelihood estimation theory to compute the model parameters. The ML estimates have three desirable properties. First, they are asymptotically unbiased, that is, they converge in the mean to the true values; second, they are asymptotically consistent, that is, the estimates converge in probability; and third, the pdf of an ML estimate approaches the Gaussian distribution as the number of classes $(n) \rightarrow \infty$. Unfortunately, all these desired properties are valid, only if we have a large number of training samples, and in practice often one has to deal with small number of training samples.

Several approaches can be also be found in the literature that specifically deal with small sample size problems in supervised learning [11, 12, 13, 14, 15, 16]. These methods are aimed at designing appropriate classifiers, feature selection, and parameter estimation so that classification error rates can be minimized while working with small sample sizes.

Our motivation stems from the fact that although, in reality, we have to deal with a small number of (labeled) training samples, we can obtain a very large number of unlabeled training samples at no cost. Then the central question is how can we augment the ML estimation of parameters in the presence of a large number of unlabeled training samples. We developed semi-supervised algorithms, and experimentally evaluated these techniques on real remote sensing datasets from various study sites in Minnesota.

1.2 Overlapping classes and multisource data classification

The spectral response distribution of classes are dependent on many factors including terrain, slope, aspect, soil type and moisture content, and atmospheric conditions. Thematic classes are often defined on the basis of some of these external factors, and not just the spectral characteristics of the class alone. For example, thematic classes such as upland hardwood and lowland hardwood, might have similar spectral properties, that is, their statistical distributions might be highly overlapping.

In such situations, any classification method that is based on spectral data alone will fail to capture the full essence of the problem. Several recent studies have focused on incorporating ancillary information into the classification process. The most notable approaches are neural networks, expert (knowledge based, rule based) systems, and the maximum likelihood classifiers (MLC) with *a priori* knowledge. Ancillary layers can be directly incorporated into neural network and decision tree learning, as opposed to maximum likelihood classification, since these classifiers do not assume any underlying probability distribution of data. In general neural networks perform as good as MLC or even better in some cases. However, neural network training and the establishment of suitable parameters are difficult in practice and the neural network approach does not offer any significant advantages over conventional classification schemes at the forest type level classification [2]. Knowledge based systems (KBS) offer a flexible framework for incorporating ancillary spatial data into the classification process. However, the main issue associated with KB systems is the development of consistent and accurate rules.

These limitations have led us to investigate two new approaches - a fusion of KBS and MLC for classification of multi-spectral remote sensing imagery utilizing knowledge derived from ancillary spatial databases, and an hierarchical classifier, which exploits relative strengths of individual classifiers from an ensemble. This approach minimizes the limitation of KB by simplifying the rule-base. In this simplified approach, the rule-base is used to stratify the image into homogeneous regions rather than classifying individual pixels. The stratified regions minimize the overlap among the classes and thus provide a robust environment for MLC. We also extended the semi-supervised learning algorithm for multisource data by utilizing the above methods.

1.3 Aggregate Classes

In many practical situations it is not feasible to collect labeled samples for all available classes in a domain. Especially in supervised classification of remotely sensed images it is impossible to collect ground truth information over large geographic regions for all thematic classes. As a result often analysts collect labels for aggregate classes (e.g., Forest, Agriculture, Urban). Aggregate classes violate basic assumption that each class is described by a unimodal statistical distribution. Aggregate classes also tend to increase class covariance structure which further leads to increased overlap with other classes. This overlap degrades classification performance. We developed a novel algorithm which showed improved classification performance.

1.4 Spatial Autocorrelation

Traditional data mining algorithms[17] often make assumptions (e.g. independent, identical distributions) which violate Tobler's first law of Geography: everything is related to everything else but nearby things are more related than distant things[18]. In other words, the values of attributes of nearby spatial objects tend to systematically affect each other. In spatial statistics, an area within statistics devoted to the analysis of spatial data, this is called spatial autocorrelation[19]. Knowledge discovery techniques which ignore spatial autocorrelation typically perform poorly in the presence of spatial data. The simplest way to model spatial dependence is through spatial covariance. Often the spatial dependencies arise due to the inherent characteristics of the phenomena under study, but in particular they arise due to the fact that imaging sensors have better resolution than object size. For example, remote sensing satellites have resolutions ranging from 30 meters (e.g., Enhanced Thematic Mapper of Landsat 7 satellite of NASA) to one meter (e.g., IKONOS satellite from SpaceImaging), while the objects under study (e.g., Urban, Forest, Water) are much bigger than 30 meters. As a result, the per-pixel-based classifiers such as MLC and MAP, which do not take spatial context into account, often produce classified images with *salt and pepper* noise. These classifiers also suffer in terms of classification accuracy.

There are two major approaches for incorporating spatial dependence into classification/prediction problems. They are spatial autoregression models [20], [21], [22], [23], [24], [25] and Markov Random Field models [26], [27], [28], [29], [30], [31], [32]. First we provide a comparison of these two approaches. MRF based classification is widely used in remote sensing. We extended our semi-supervised framework by incorporating spatial neighborhood relationships via MRF model.

1.5 Contributions

In this thesis work, we analyzed and characterized the practical problems that arise in mining spatio-temporal datasets due classical assumptions. We developed efficient algorithms for thematic information extraction from multidimensional remote sensing imagery (raster data) guided by ancillary geo-spatial information (vector data), while accounting for small training sample sizes and spatial autocorrelations.

In the first problem, we developed a semi-supervised classification framework which addresses the small training dataset problem by augmenting the parameter estimation using unlabeled samples. We studied this framework extensively by conducting experiments on different study sites under varying geographic settings. One of the main problems that were not considered in the previous literature is the quality of unlabeled training samples. Though literature survey shows that incorporation of unlabeled training samples improved the overall classification accuracies, our extensive study shows that often classification performance degrades with incorporation of unlabeled samples. One of the main reasons for such degradation in performance can be attributed to the fact that the statical model for unlabeled training data has more components than the number of classes (components). We developed an adaptive semi-supervised learning algorithm which tries to mitigate this problem.

The second problem we addressed is the class overlap problem, where often mismatch occurs between the classes defined by the user and the class distribution supported by the underlying datasets. We developed two new classifiers; one that exploits spectral and spatial knowledge, second one that exploits hierarchical nature of classes. We also extended the semi-supervised approach for multisource data.

We addressed the aggregate class problem by relaxing the unimodal class assumption. Instead we model each aggregate class as a finite Gaussian mixture. The number of components in each finite Gaussian mixture are automatically estimated. Our experimental evaluation showed improved classification accuracy.

In the final problem, we addressed the spatial autocorrelation phenomena which is a norm in the spatio-temporal datasets. We extended our semi-supervised framework to model the spatial neighborhood relationships via Markov Random Field model. Experimental evaluation shows that our new spatial semi-supervised learning algorithm performs not only better than other relevant classifiers, but it also produces thematic maps which is most desired by the domain experts. In addition, we developed an efficient algorithm which is computationally more attractive.

Chapter 2

Supervised Classification

Given a sample set of input-output pairs, the objective of supervised learning is to find a function that learns from the given input-output pairs, and predicts an output for any unseen input (but assumed to be generated from the same distribution), such that the predicted output is is as close as possible to the desired output. The name "supervised" comes from the fact that the input-output example pairs are given by an expert (teacher). Examples of the supervised learning include thematic map generation (classification) from satellite images, tumor or other organs recognition from medical images, recognition of hand written characters from the scanned documents, prediction of stock market indexes, and speech recognition. The input-output pairs, also called training samples, or training dataset, is denoted by (x_i, y_i), where x_i's are often vectors of measurements over the d-dimensional attribute space. For example, in remote sensing image classification, the input attribute space consists of various spectral bands or channels (e.g., blue, green, red, infra-red, thermal, etc.), and the input vectors (x_i's) are reflectance values at the i^{th} location in the image, and the outputs (y_i's) are thematic classes such as forest, urban, water, and Agriculture. Depending on the type of the output attribute, two supervised learning tasks can be distinguished:

- **Classification:** In classification, the input vectors x_i are assigned to few discrete number of classes y_i.
- **Regression:** In regression, also known as function approximation, the input-output pairs are generated from an unknown function of the form $y = f(x)$,

where y is continuous. Typically regression is used in prediction and estimation, for example, stock value prediction, daily temperature prediction, and market share estimation for a particular product. Regression can also be used in inverse estimation, that is, given that we have an observed value of y, we want to determine the corresponding x value.

Classification can be viewed as a special case of regression. In this thesis we specifically consider the problem of multi-spectral remote sensing image classification. Image classification can be formally defined as finding a function $g(x)$ which maps the input patterns x onto output classes y_i (some times y_i's are also denoted as ω_i). The main objective is to assign a label (e.g. Water, Forest, Urban) to each pixel in the classified image, given corresponding feature vector x_j in the input image. We use the following definitions.

Let

X be an image composed of d-dimensional pixels, where d corresponds to the number of features (e.g. blue, green, red, infra-red, and thermal bands in multi-spectral images or soil types, elevation, aspect, ecological zones, and other ancillary geo-spatial data layers).

$y_i, i = 1, \ldots, M$, where M is the total number of classes

$\mathbf{x}$ is a d-dimensional feature (column) vector

$Pr(y_i|x), i = 1, ..., M$ are the conditional probabilities referred to as *a posteriori* probabilities.

Depending on the type of supervised learning method used, the objective of a supervised learning could be finding a function $g(x)$ (also called a discriminant function), that divides the input $d-$dimensional feature space into several regions, where each region corresponds to a thematic class y. One such simple function is given by:

$$x \in y_i \text{ if } p(y_i|x) > p(y_j|x) \ \forall j \neq i. \tag{2.1}$$

That is, the feature vector x belongs to class y_i if $p(y_i|x)$ is the largest.

We now formally define the supervised classification problem in the context of general statistical framework.

2.1 Supervised Statical Classification: Problem Formulation

Let us now formalize the classification problem as following:

Given:

A spatial framework $\mathcal{S}$ consisting of sites s_{ij} ordered as an $l \times p$ matrix, where l is the number of lines and p is the number of pixels, and $\{1 \leq i \leq l, 1 \leq j \leq p\}$.

A $d-$dimensional feature vector at each site s_{ij}, is represented by x_{ij}, where x_{ij} is a continuous random variable.

A set of discrete labels $Y = \{y_1, \ldots, y_k\}$, where k is the total number of distinct labels

Training Dataset $D = (x_i, y_j)$, where i is indexed over S and j is indexed over the number of classes k.

A parametric model (e.g., Gaussian).

Find:

Estimate parameter vector Θ (e.g., $\{\mu_i, \Sigma_i\}$).

Objective:

Maximize complete data *log-likelihood* $L(\Theta)$.

Assumptions:

A1 The size of the labeled training dataset is approximately 10 to 30 times the number of dimensions

A2 Thematic classes are separable

A3 Classes are unimodal

A4 Feature vectors are independent and identically distributed (i.i.d), but features are highly correlated in feature space (that is features are not independent).

Rest of the thesis revolves around this generic classification framework and especially the four assumptions (A1 through A4). Even though the discrimination rule defined above (Eq. 2.1) sounds simple, the class assignment problem is very difficult. There is no single algorithm which will correctly classify any given image. The problem stems from the fact that in many classification situations one or more of the assumptions are violated. In this thesis we address all the four assumptions. We now briefly delve into three popular supervised learning schemes, namely maximum likelihood classification, decision trees, and neural networks.

2.2 Maximum Likelihood Classifier

Maximum likelihood classification is one of the most widely used parametric and supervised classification technique in remote sensing field [33], [34]. The discriminant function defined in eq. 2.1, though simple, needs the $p(y_i|x)$ to be estimated. Assuming that sufficient ground truth (training) data is available for each thematic class, we can estimate the probability distribution $p(x|y_i)$ for a class (y_i) that describes the chance of finding a pixel from that class at the position $\mathbf{x}$. This estimated $p(y_i|x)$ can be related with the desired $p(x|y_i)$ using Bayes' theorem:

$$p(y_i|x) = \frac{p(x|y_i)p(y_i)}{p(x)} \tag{2.2}$$

where $p(y_i)$ is the probability that class y_i occurs in the image, also know as 'a priori' probability, and $p(\mathbf{x})$ is the probability of finding a pixel from any class at location $\mathbf{x}$. Since $p(\mathbf{x})$ is constant, we can omit it from computation and write the discriminant function $g(\mathbf{x})$ by simplifying eq 2.2 and taking logarithm as follows:

$$g_i(\mathbf{x}) = \ln p(\mathbf{x}|y_i) + \ln p(y_i) \tag{2.3}$$

where ln is the natural logarithm. By assuming multivariate normal model for class probability distributions, the discriminant function $g_i(x)$ for the maximum likelihood classification can be written as the following.

$$g_i(\mathbf{x}) = \ln p(\omega_i) - \frac{1}{2}(\mathbf{x} - \mathbf{m}_i)^t \Sigma_i^{-1} (\mathbf{x} - \mathbf{m}_i) \tag{2.4}$$

MLC is an example of a Bayesian classifier. We give a formal treatment of Bayesian classification in the next chapter; more details can also be found in [35, 36].

2.3 Decision Trees

A decision tree (DT) is a supervised classifier that recursively partitions a data set into smaller subdivisions based on a set of simple tests at each internal node in the tree. The leaf nodes represents the class labels y_i. Training data set is used to learn the split conditions at each internal node and to construct a decision tree. For each new sample (i.e., feature vector x), the classification algorithm will search for the region along a path of nodes of the tree to which the feature vector x will be assigned. That is, the classification of a region is determined by a path from the root node to a leaf node. In previous studies decision trees were used for remote sensing classification with accuracies that were comparable to MLC [37] and global land cover classifications [5], [7]. Performance of three different types of decision trees, namely, univariate, multivariate, and hybrid decision trees for remote sensing data classification, against MLC were reported in [6]. Most common decision trees split the feature space into hyper-rectangles with sides parallel to the axis. They work well for discrete attributes and moderate dimensionality. To overcome this limitation, oblique decision trees [38] were proposed to find the optimal hyper-plane (not necessarily axis-parallel) for each node of a decision tree, which showed improved performance for continuous attributes (for example, remote sening images). In this study, we used C4.5, a publicly available univariate decision tree software. For algorithmic details, see Quinlan [39].

2.4 Neural Networks

Artificial neural networks, which are non-parametric classifiers as opposed to Bayesian classifiers, are gaining popularity in remote sensing image classification. This popularity can attributed to several factors: 1) previous studies [1], [40] have shown that their performance is as good as MLC and in many cases even better accuracy, 2) they are non-parametric, so they are capable of classifying multi-source data, where as parametric classifiers have problems with multi-source data, and 3) they have several desirable characteristics like nonlinearity, adaptability and fault tolerance. Their use in remote sensing data analysis has been some what limited till recent years; because of the complexities associated with setting up suitable parameters for network training and the lack of knowledge about internal working of network (especially how they divide the feature space) and lack of comparative studies. The previous "black box" view of neural networks which limited its use is now broken with the insights provided by recent studies [41], [2], [4]. Several recent studies [3], [1], [40] were also focused on comparing statistical and neural network classification of remote sensing data.

The back-propagation algorithm is the most common method of training multi-layer feed-forward neural networks (also know as multi-layer perceptrons). Multi-layer perceptrons (MLP) are simple interconnections of neurons organized into multiple layers, typically consisting of input, output and one or more hidden layers. Hidden layers enables the network to extract higher order statistics [42]. Information processing starts from input end of network and propagates through hidden layers and to the output end of network (thats why it is called as feed-forward networks). Input layers serve as distribution structure for the inputs, hidden neurons intervene between input and output layers, and the output layer constitutes the overall response of the network. This network can be trained in several ways, but back-propagation is a highly popular choice for training MLP networks. The error back-propagation consists of two passes through different layers of the network, a forward and a backward pass. The algorithm works as follows:

1. Initialization. Initialize the network with uniformly distributed random weights, read inputs x and desired outputs, y_j.

2. Presentation of Training Examples. Present network with an epoch of training examples. For each example, repeat steps 3 and 4.

3. Feed-forward computation. In this pass compute the outputs, y_j:

$$y_j = \frac{1}{1 + \exp^{(-v_j)}}, \quad v_j = \Sigma_i w_{ij} y_i \tag{2.5}$$

   Compute the error signal as the difference between computed output ($o_j = y_j$ - for output layer) and the desired output d_j.

4. Backward computation. During this pass synaptic weights are adjusted according to error correction rule,

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j y_i \tag{2.6}$$

where $w_{ij}(t)$ is the weight from the input node i to the hidden node j at time t, η is learning rate, and δ_j is the error term (i.e. local gradients). The δ's are computed as follows:

$$\delta_j = y_j(1-y_j)(d_j - o_j) \;\; \text{for neuron j in output layer}$$

$$\delta_j = y_j(1-y_j)\Sigma_k\delta_k w_{jk} \;\; \text{for neuron j in hidden layer}$$

5. Iteration. Repeat steps 3 and 4 by presenting new epochs of training examples to the network until the network converges, or a predetermined minimum error is achieved.

There are several disadvantage with this plain vanilla back-propagation algorithm, especially it has the danger of stuck in local minima, or oscillation behavior when the error surface has a very narrow minimum area. We have used back-propagation momentum algorithms which avoids the oscillation behavior of vanilla back-propagation incorporating a momentum term in the generalized delta rule as follows:

$$w_{ij}(t+1) = \alpha w_{ij}(t) + \eta\delta_j y_i \tag{2.7}$$

where α is a constant specifying the influence of the momentum. This modification allows faster traversal of flat spots of the error surface with big step size and small step size for rough surfaces, and thus increases learning speed.

In this study we are specifically interested in comparative analysis of MLC, DT, and NN. We used the same training and test dataset for comparing the accuracies of these classifiers.

2.5 Experimental Evaluation

2.5.1 Dataset

The Cloquet study site encompasses Carlton County, Minnesota, which is approximately 20 miles southwest of Duluth, Minnesota. The region is predominantly forested, composed mostly of upland hardwoods and lowland conifers. There is a scattering of agriculture throughout. The topography is relatively flat, with the exception of the eastern portion of the county containing the St. Louis River. Wetlands, both forested and non-forested, are common throughout the area. The largest city in the area is Cloquet, a town of about 10,000. We used a spring Landsat 7 scene, taken May 31, 2000, and clipped to the study region. The final rectified and clipped image size is 1343 lines x 2019 columns x 6 bands. The training and test data set descriptions were given in Table 2.1.

CID y_i	Class Name	Training Samples	Test Samples
1	Hardwood 1	4	23
2	Hardwood 2	9	51
3	Conifer	12	29
4	Agriculture	8	26
5	Urban	7	5
7	Wetlands	17	65
8	Water	3	6

Table 2.1: Training and Test Data Set.

All the three classifiers were trained using the same 60 labeled training dataset and tested with a separate training dataset consisting of 205 labeled plots. Performance of each classifier was measured using the following accuracy measures.

2.5.2 Contingency Table

The relationship between the predicted classes versus ground truth (test data) or reference data is often summarized as table variously known as contingency table, error matrix, or confusion matrix (see Table 2.2).

We can derive several accuracy measures from the contingency table.

Image Label	Reference			
	Class 1	Class 2	...	Class M
Class 1	a_{11}	a_{11}	...	a_{1M}
Class 2	a_{21}	a_{22}	...	a_{2M}
...	...		...	...
Class M	a_{M1}	a_{M2}	...	a_{MM}

Table 2.2: Contingency Table

1. The *overall accuracy (OA)* gives the percentage of the pixels in all reference areas that are correctly classified, that is, the probability that a randomly selected point location is correctly classified in the image.

$$OA = \left(\sum_i a_{ii}\right) \Big/ \left(\sum_{ik} a_{ik}\right) \tag{2.8}$$

2. *Commission Error* (CE_i) *:* The conditional probability that a randomly selected point classified as class i by the classifier is actually the class j in the reference data.

$$CE_i = \left(\sum_j a_{i \neq j}\right) \Bigg/ \left(\sum_k a_{ik}\right) \tag{2.9}$$

3. *Omission Error* (OE_i)*:* The conditional probability that a randomly selected point classified as class i by the reference data is actually the class j in the classified image.

$$OE_i = \left(\sum_j a_{j \neq i}\right) \Bigg/ \left(\sum_k a_{ki}\right) \tag{2.10}$$

4. *User accuracy* (UA_i) for a given class i gives the percentage of the image pixels in all reference areas that classified as class i, which are actually classified as the same in the reference data. The user accuracy is related to the commission error as $UA_i = 100 - CE_i$.

$$UA_i = a_{ii} \Big/ \left(\sum_k a_{ki}\right) \tag{2.11}$$

5. *Producer accuracy* (PA_i) for a given class i gives the percentage of the image pixels in the reference area that are correctly classifed, that is, the conditional probability that a randomly selected point classified as class i by the reference data is also classified as the the class i by the classifier. Producer accuracy is related to omission error as $PA_i = 100 - OE_i$.

$$PA_i = a_{ii} \Big/ \left(\sum_k a_{ik} \right) \tag{2.12}$$

6. *Cohen's kappa coefficient (κ)* is another widely used statistical measure to report classification error. This measure is more robust than simple percent agreement measures defined above, as it takes into account all off-diagonal elements into the estimation of accuracy. Using the entries in the contingency table, the κ is defined as:

$$\kappa = \left(Pr(p_o) - Pr(e)\right) / \left(1 - Pr(e)\right) \tag{2.13}$$

where $pr(o) = \sigma_i a_{ii}$, and $pr(e) = \sigma_i (a_{+j} aj+)$. One interpretation of κ is: ¡ 0.40: poor, 0.40 - 0.59 : fair, 0.60 - 0.74 : good, and ¿ 0.75 : excellent.

2.5.3 Results

Let us now summarize the classification results using the accuracy measures defined in Section 2.5.2. Table 2.3 shows the error matrix for maximum likelihood classification, Table 2.4 shows the error matrix for decision tree classification, and Table 2.5 shows the error matrix for neural network classification.

Figure 2.1 summarizes the producers accuracy (PA) and overall accuracies (OA) of all three classifiers. First seven bars (three bar group representing DT, MLC, and NN classifiers respectively) represents producer's accuracy and last bar represents overall accuracy. It can be seen from this figure, that there are minor differences between individual class accuracies, however the performance of both MLC and NN are very similar.

Image	Reference							
Label	1	2	3	4	5	7	8	PA
1	168.00	26.00	0.00	10.00	2.00	1.00	0.00	81.16
2	22.00	416.00	5.00	1.00	8.00	7.00	0.00	90.63
3	0.00	9.00	207.00	0.00	17.00	28.00	0.00	79.31
4	6.00	0.00	0.00	220.00	4.00	4.00	0.00	94.02
5	0.00	0.00	0.00	0.00	45.00	0.00	0.00	100.00
7	0.00	17.00	1.00	7.00	48.00	512.00	0.00	87.52
8	0.00	0.00	2.00	0.00	16.00	0.00	36.00	66.67
UA	85.71	88.89	96.28	92.44	32.14	92.75	100.00	OA=86.94

Table 2.3: MLC Accuracy (Error Matrix)

Image	Reference							
Label	1	2	3	4	5	7	8	PA
1	176.00	17.00	0.00	14.00	0.00	0.00	0.00	85.00
2	94.00	351.00	7.00	4.00	0.00	3.00	0.00	76.50
3	0.00	23.00	236.00	0.00	0.00	2.00	0.00	90.40
4	24.00	0.00	0.00	208.00	1.00	1.00	0.00	88.90
5	0.00	0.00	0.00	0.00	43.00	2.00	0.00	95.60
7	13.00	28.00	22.00	47.00	79.00	396.00	0.00	67.70
8	0.00	0.00	16.00	0.00	0.00	0.00	38.00	70.40
UA	57.00	84.00	84.00	76.00	35.00	98.00	100.00	OA=78.48

Table 2.4: DT Accuracy (Error Matrix)

2.6 Conclusions

In this chapter we introduced supervised learning and did a comparative analysis of three popular classification schemes, namely, maximum likelihood, decision trees, and neural networks. This study shows that both maximum likelihood and neural networks classifiers performed well and have almost same performance results. However, maximum likelihood classifier, which is a Bayesian classifier, is very simple and have nice properties which will be discussed in the next chapter.

Image	Reference							
Label	1	2	3	4	5	7	8	PA
1	183.00	21.00	0.00	2.00	0.00	1.00	0.00	88.40
2	31.00	421.00	3.00	1.00	0.00	3.00	0.00	91.70
3	0.00	28.00	224.00	0.00	0.00	4.00	5.00	85.80
4	34.00	3.00	0.00	194.00	0.00	3.00	0.00	82.90
5	0.00	0.00	0.00	0.00	45.00	0.00	0.00	100.00
7	1.00	27.00	7.00	27.00	39.00	479.00	5.00	81.90
8	0.00	0.00	0.00	0.00	0.00	0.00	54.00	100.00
UA	73.50	84.20	95.70	86.60	53.60	97.80	84.40	OA=86.72

Table 2.5: NN Accuracy (Error Matrix)

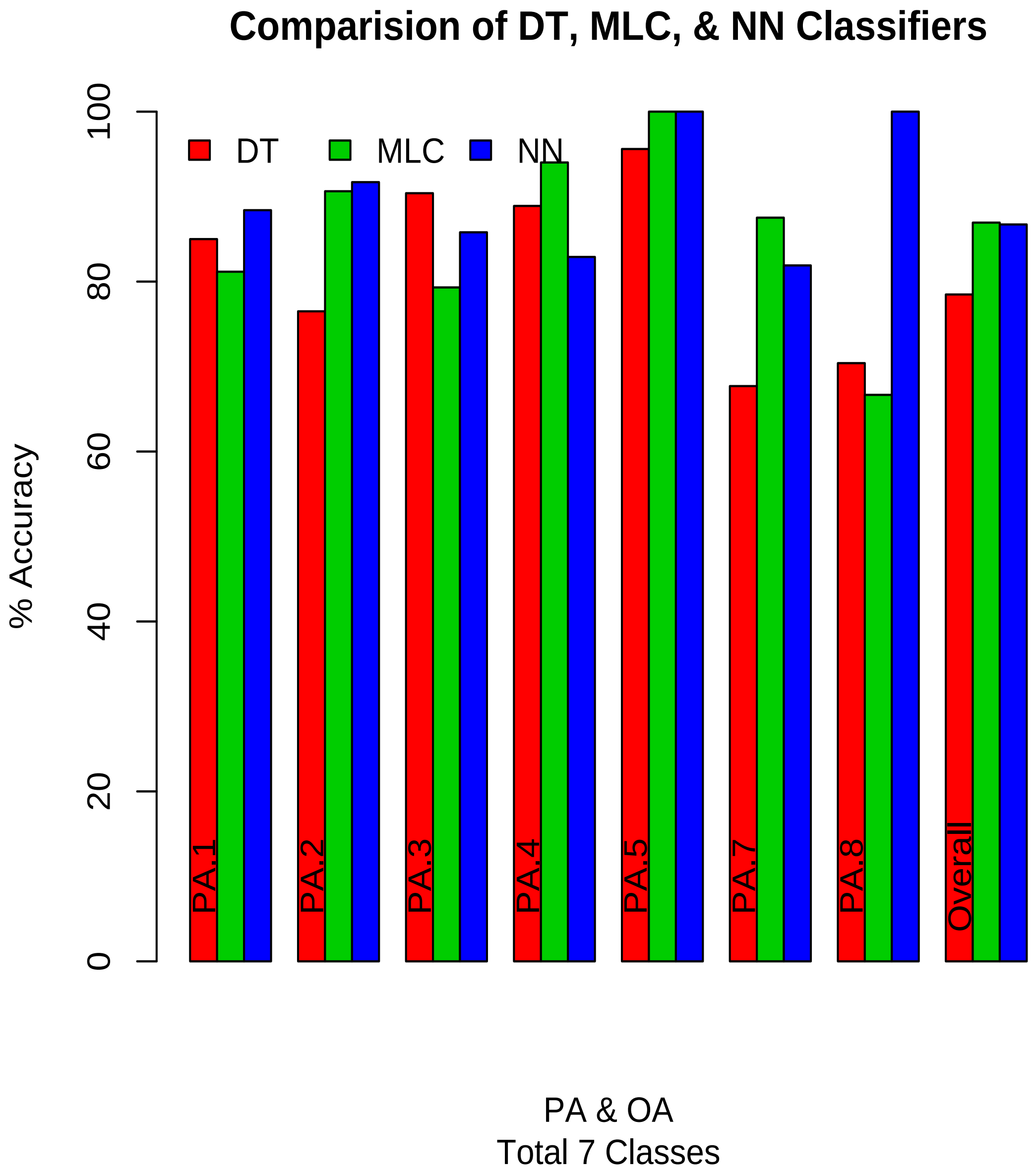


Figure 2.1: Comparison of DT, MLC, and NN Classifiers

Chapter 3

Semi-supervised Learning

3.1 Introduction

New approaches are needed to extract useful patterns from increasingly large remote sensing and geo-spatial databases in order to understand global climatic changes, vegetation dynamics, ocean processes, etc. Remote sensing provides continuous data of large geographic regions. The multi-spectral and multi-temporal digital imagery acquired by using remote sensing has applications in areas such as natural resource monitoring, thematic mapping, flood and fire disaster monitoring, target detection, and urban growth modeling. The new generation of high spatial (and spectral) resolution satellites are acquiring huge amounts of image data. NASA alone collects more than a terabyte of image data per day. There is a great demand for accurate land use and land cover classification derived from remotely sensed data in the applications mentioned above. However, increasing spatial and spectral resolution puts several constraints on supervised classification. The increased spectral resolution requires a large amount of accurate training data. Collecting ground truth data for a large number of samples is very difficult, especially in emergency situations like forest fires, land slides, floods, etc. In this chapter we explore methods that utilize unlabeled samples in supervised learning framework.

3.1.1 Semi-supervised Learning: Problem Formulation

Let us now formalize the semi-supervised learning method.

Given:

A spatial framework $\mathcal{S}$ consisting of sites s_{ij} ordered as an $l \times p$ matrix, where l is the number of lines and p is the number of pixels, and $\{1 \leq i \leq l, 1 \leq j \leq p\}$.

A $d-$dimensional feature vector at each site s_{ij}, is represented by x_{ij}, where x_{ij} is a continuous random variable.

A set of discrete labels $Y = \{y_1, \ldots, y_k\}$, where k is the total number of distinct labels

Training Dataset $D = D_l \cup D_{ul}$, where D_l contains labeled training plots and D_{ul} contains unlabeled training plots.

A parametric model (e.g., Gaussian).

Find:

Estimate parameter vector Θ (e.g., $\{\mu_i, \Sigma_i\}$).

Objective:

Maximize complete data *log-likelihood* $L(\Theta)$.

Assumptions:

A1 The size of the labeled training dataset is less than 10 to 30 times the number of dimensions

A2 Thematic classes are separable

A3 Classes are unimodal

A4 Feature vectors are independent and identically distributed (i.i.d), but features are highly correlated in feature space (that is features are not independent).

A5 D_l and D_u samples are generated by the same GMM.

This problem definition is essentially same as the supervised classification problem definition (2.1), except for the following two modifications: first, the training data set D includes unlabeled training samples, second, the labeled training samples are less than required number to accurately estimate the model parameters. We also made an additional assumption that the labeled and unlabeled samples were generated by the same model (GMM).

3.1.2 Related Work and Our Contributions

Recently, semi-supervised learning techniques that utilize large number of unlabeled training samples in conjunction with small labeled training data are becoming popular in machine learning and data mining [43, 44, 45]. This popularity can be attributed to the fact that several of these studies have reported improved classification and prediction accuracies, and that the unlabeled training samples comes almost for free. The common thread between many of these methods is the Expectation Maximization (EM) algorithm. The EM algorithm, first proposed in [46], has become one of the popular methods for maximum likelihood (ML) based parameter estimation. Key feature of the EM algorithm is that it estimates parameters in the absence of feature values in the input data (also known as incomplete data). Many of the semi-supervised learning methods pose class labels as the missing data and use EM algorithm to improve initial (either guessed or estimated from small labeled samples) parameter estimates.

The semi-supervised learning techniques have not been well explored in the remote sensing and GIS domains. Only notable study is reported in [47] for hyperspectral data analysis. Objectives of this study are to understand the effectiveness of semi-supervised learning with unlabeled samples for multi-spectral remote sensing image classification. In text data mining, often it is assumed that the features (words) are independent [45], which leads to simpler statistical models. Often features (spectral bands) in remote sensing imagery are highly correlated, which leads to the assumption of multivariate normal distributions with general covariance matrices. This assumption increases the number of parameters to be estimated. In this paper we provide a semi-supervised learning method based on expectation maximization (EM) algorithm. As features are highly correlated, we use a Gaussian mixture model (GMM) for describing the training samples and use exploit formulas for estimating all model parameters. We have conducted several experiments to evaluate the usefulness of this method in thematic information extraction from multispectral remote sensing imagery.

3.2 Statistical classification framework

In this section we present a general statistical framework for classification of multispectral remote sensing image data. Each pixel in a remotely sensed image can be thought of as a feature vector (x) and as an instance of a continuous random variable. A continuous random variable is described by a probability density function $(p(\cdot))$. In the classification of a remote sensing image, our objective is to assign a class label (y) to each pixel (x) based on a certain decision criterion. Maximum likelihood classification (MLC) and maximum a posteriori (MAP) classification are two of the most widely used classifiers in remote sensing. Bayesian decision theory plays a central role in statistical pattern classification. MLC and MAP are based on Bayesian decision theory.

3.2.1 Bayesian Classification

In the Bayesian approach, the objective is to find the most probable set of class labels given the data (feature) vector and *a priori* or prior probabilities for each class. Formally, we can state Bayes' formula as:

$$P(y_i|x) = \frac{p(x|y_i)P(y_i)}{p(x)} \tag{3.1}$$

Bayes' formula allows us to compute the posterior probability $(P(y_i|x))$ provided that we know the class conditional probability density $(p(x|y_i))$ and the *a priori* probability distribution $(P(y_i))$. The term $p(x)$ is often called the evidence factor, that is, the probability of finding a feature vector x from any of M classes and is given by:

$$p(x) = \sum_{j=1}^{M} p(x|y_i)P(y_i) \tag{3.2}$$

The evidence $p(x)$ acts as a scale factor that guarantees that the posterior probabilities sum to one; it has no consequence on the decision rule and is thus often omitted from the decision rule. For a two class (y_1, y_2) problem, the Bayes' decision rule is given by:

decide y_1 if $P(y_1|x) > P(y_2|x)$; otherwise decide y_2.

3.2.2 Maximum likelihood and maximum a posteriori classification

The outcome of a Bayesian decision rule is determined by the class conditional densities $p(x|y_i)$ as well as the *a priori* probabilities $P(y_i)$. For $p(x|y_i)$, if we assume a multivariate normal or Gaussian density, then $p(x|y_i)$ is given by

$$p(x|y_i) \;=\; \frac{1}{\sqrt{(2\pi)^{-N}|\Sigma_i|}}e^{\frac{-1}{2}(x-\mu_i)^t|\Sigma_i|^{-1}(x-\mu_i)} \tag{3.3}$$

where μ_i and $|\Sigma_i|$ are the mean vector and covariance matrix of the data for any given class y_i. The $p(x|y_i)$ is also called the *likelihood* of y_j with respect to x. If we have no knowledge about the prior distributions $P(y_i)$, then we can assume that all classes are equally probable, that is, $P(y_1) = P(y_2) = \ldots = P(y_M)$. As a consequence, we can further drop *a priori* term $P(y_i)$ in the computation of the discriminant function $g_i(x)$; the resulting classifier is known as the maximum likelihood classifier (MLC) and the discriminant function is give by

$$g_i(x) \;= -\ln|\Sigma_i| - (x-\mu_i)^t|\Sigma_i|^{-1}(x-\mu_i) \tag{3.4}$$

On the other hand, if we have knowledge about the prior distributions $P(y_i)$, then the resulting classifier is known as the maximum a posterior (MAP), and the discriminant function is given by:

$$g_i(x) \;=\; \ln\; P(y_i) - \frac{1}{2}\ln|\Sigma_i| - \frac{-1}{2}(x-\mu_i)^t|\Sigma_i|^{-1}(x-\mu_i) \tag{3.5}$$

The covariance matrix Σ plays a key role in discriminant analysis. Covariance accounts for the shape (size and orientation) of classes in the feature space. The effectiveness of ML/MAP classification depends on the quality of the estimated parameter vector Θ (e.g., mean vector μ and the covariance matrix Σ for each class) from the training samples.

3.2.3 Parameter estimation

In section 3.2.2 we assumed that the form of class conditional density $p(x|y_i)$ was multivariate normal. This assumption reduces the difficult problem of estimating an unknown density function $p(x|y_i)$ into a simpler parameter (Θ) estimation problem. Here we use a well-known parameter estimation technique, maximum likelihood estimation (MLE), to obtain the parameter vector Θ from the training samples.

Given an M-class problem, let us assume that we have M number of training sample datasets, each organized by a class, $D_1, \ldots, D_M$, where samples in D_j have been drawn independently according to a multivariate normal density function $p(x|y_j)$. Let us denote the corresponding parameter vector as θ_j. The dependence of $p(x|y_j)$ on parameter vector θ_j is denoted explicitly by $p(x|y_j, \theta_j)$ (or sometimes by $p(x|y_j; \theta_j)$). Such samples are known as independent and identically distributed (i.i.d) random variables. That is, the data from one class do not affect the parameter estimation of the other classes. As a result, we can further simplify our notation for class conditional density as $p(D|\theta)$ (or $p(D; \theta)$). First, let us assume that D contains n random samples, $x_1, \ldots, x_N$, drawn independently from the pdf $p(x|\theta)$. Then $p(D|\theta)$ is given by,

$$p(D|\theta) = \prod_{k=1}^{n} p(x_k|\theta). \tag{3.6}$$

The $p(D|\theta)$ in Equation 3.6, which is a function of θ, is also known as the *likelihood* function of θ with respect to the data D (set of training samples for a given class). The *likelihood* function is often represented by the symbol $l(\theta)$ or by $l(\theta|D)$. The MLE of θ is the parameter ($\hat{\theta}$) that maximizes the *likelihood* function $p(D|\theta)$, and is given by

$$\hat{\theta} = arg \max_{\theta} \prod_{k=1}^{n} p(x_k|\theta). \tag{3.7}$$

From standard differential calculus we know that the necessary condition for $\hat{\theta}$ to be maximum is that the gradient of the *likelihood* function with respect to θ should be zero, that is,

$$\frac{\partial \prod_{k=1}^{n} p(x_k|\theta)}{\partial \theta} = 0. \tag{3.8}$$

Often it is mathematically simpler to deal with the *log-likelihood* function, $l(\theta) = \ln p(D|\theta)$. Since the ln function is monotonically increasing, the parameter θ that maximizes the *likelihood* function also maximizes the *log-likelihood* function. Solving Equation

3.8 for a multivariate normal distribution yields parameters μ and Σ such that

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k \tag{3.9}$$

and

$$\hat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\mu})(x_k - \hat{\mu})^t. \tag{3.10}$$

3.2.4 MLC Performance With Small Training Samples

The ML estimates have three very desirable properties. First, they are asymptotically unbiased, that is, they converge in the mean to the true values; second, they are asymptotically consistent, that is, the estimates converge in probability; and third, the pdf of an ML estimate approaches the Gaussian distribution as $n \to \infty$. Unfortunately, all these desired properties are valid for a large number of training samples. Let us now evaluate maximum likelihood classifier with small number of training samples. Figure 3.4 shows the the performance of the MLC as the number of training samples increases. Two trends can be seen from this graph: first as the number of labeled samples increases the overall performance of MLC also increases, second, as the number of samples increases the variance in the performance decreases. As mentioned previously, maximum likelihood estimates required large number of training samples. When the training samples are very few, the estimates are quite different than the estimates obtained from full training dataset. This is reflected in the classification accuracy. Further details about this experiment can be found in the experiments section.

3.3 Semi-supervised learning approach

In the previous section, for simplicity we treated parameter estimation for each class (of M-classes) separately based on M-training datasets. However, in this section we treat all class samples together, by assuming that a given sample comes from a finite mixture of distributions. Both of these examples were shown for univariate and bivariate Gaussian distributions in Figures 3.1 and 3.2 respectively. These distributions were estimated using training (labeled) dataset described in the experiments section. The id 0 represents the GMM fit from the individual components. From the Figure 3.2 it should be clear that the covariance matrix Σ plays an important role (size and shape of the ellipse) in describing the class conditional distributions and discriminant boundaries. Feature independence assumptions (as in text data mining [45]) leads rectangular (or parallelepipeds in multidimensional feature space) approximations which results inaccurate predictions.

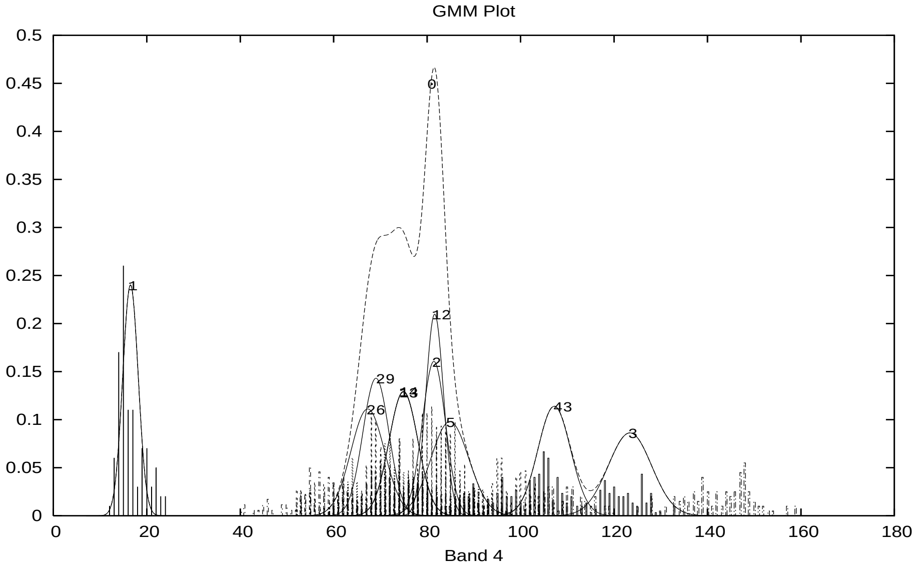


Figure 3.1: Gaussian mixture model (GMM) generated from the training dataset using single feature (spectral band number 4)

We reformulate the likelihood estimation for finite mixture models and describe a parameter estimation technique that is based on expectation maximization algorithm and that also utilizes unlabeled training samples. First let us assume that each sample x_j comes from a super-population D, which is a mixture of a finite number (M) of populations $D_1, \ldots, D_M$ in some proportions $\alpha_1, \ldots, \alpha_M$, respectively, where $\sum_{i=1}^{M} \alpha_i = 1$ and $\alpha_i \geq 0 (i = 1, \ldots, M)$. Compared to our discussion in Section 3.2.1, we can think of α_i as $P(y_i)$. Now we can model the data $D = \{x_i\}_{i=1}^{n}$ as being generated independently from the following mixture density.

$$p(x_i|\Theta) = \sum_{j=1}^{M} \alpha_j p_j(x_i|\theta_j) \tag{3.11}$$

Here $p_j(x_i|\theta_j)$ is the pdf corresponding to the mixture j and parameterized by θ_j,

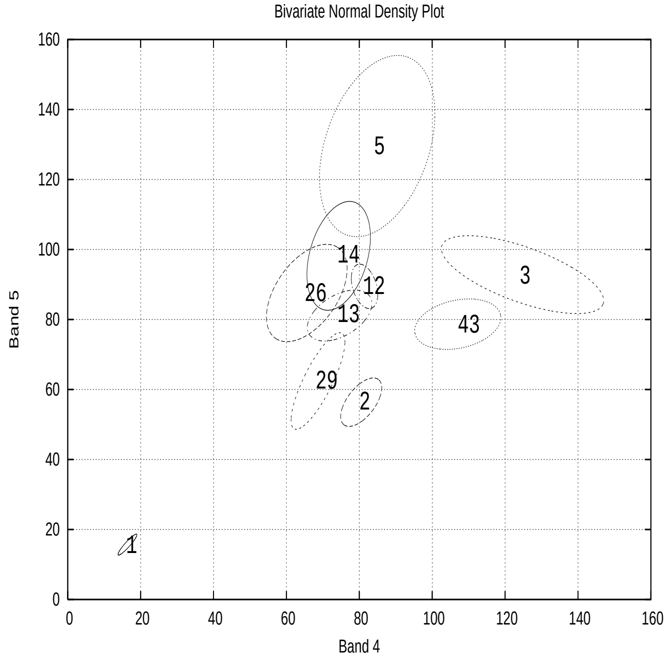


Figure 3.2: Bivariate distribution from the same training dataset using bands 4 and 5.

and $\Theta = (\alpha_1, \ldots, \theta_M, \theta_1, \ldots, \theta_M)$ denotes all unknown parameters associated with the M-component mixture density. For a multivariate normal distribution, θ_j consists of elements of the mean vectors μ_j and the distinct components of the covariance matrix Σ_j. The *log-likelihood* function for this mixture density can be defined as:

$$L(\Theta) = \sum_{i=1}^{n} \ln \left[\sum_{j=1}^{M} \alpha_j p_j(x_i|\theta_j) \right]. \tag{3.12}$$

In general, Equation 3.12 is difficult to optimize because it contains the ln of a sum term. However, this equation greatly simplifies in the presence of unobserved (or incomplete) samples. Let us now pose X as an incomplete dataset, and assume that we have unobserved data $Y = y_i{}_{i=1}^{n}$ such that y_i tells us which component density generated each x_i. Assuming that we know the values of Y, the *log-likelihood* in Equation 3.12 can be simplified as:

$$L(\Theta) = \ln(P(X,Y)|\Theta)) = \sum_{i=1}^{n} \ln(P(x_i|y_i)P(y)) = \sum_{i=1}^{n} \ln(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i})). \tag{3.13}$$

However, in many supervised learning situations, the class labels (y_i)'s are not available. However, assuming the the initial parameters Θ^k can be guessed (as in clustering), or can be estimated (as in semi-supervised learning), we can easily compute $p_j(x_i|\theta_j^k)$ in eq. 3.11. Now, using Bayes' rule, we can compute

$$p(y_i|x_i, \Theta^k) = \frac{\alpha_{y_i}^k p_{y_i}(x_i|\theta_{y_i}^k)}{p(x_i|\Theta^k} = \frac{\alpha_{y_i}^k p_{y_i}(x_i|\theta_{y_i}^k)}{\sum_{j=1}^{M} \alpha_j^k p_j(x_i|\theta_j^k)} \tag{3.14}$$

So, the expectation maximization (EM) algorithm at the first step maximizes the expectation of the *log-likelihood* function, using the current estimate of the parameters and conditioned upon the observed samples. In the second step of the EM algorithm, called maximization, the new estimates of the parameters are computed. The EM algorithm iterates over these two steps until the convergence is reached. These two steps are formalized below:

E-step: At the i^{th} step of the iteration, where $\Theta^{(i-1)}$ is available, compute the expected value of

$$Q(\Theta, \Theta^{(k-1)}) = E\left[\ln p(X, Y|\Theta)|X, \Theta^{(k-1)}\right]. \tag{3.15}$$

This step is called the *expectation step*. In the function $Q(\Theta, \Theta^{(k-1)})$, the first argument Θ corresponds to the parameters that needs to be optimized by maximizing the *log-likelihood*, and the second argument $\Theta^{(k-1)}$ corresponds to the current estimate of the parameters that we used to evaluate the expectation.

M-step: Compute the new estimates of Θ by maximizing the $Q(\Theta, \Theta^{(k-1)})$, that is, find:

$$\theta^{(k)} = arg \max_{\Theta} Q(\Theta, \Theta^{(k-1)}). \tag{3.16}$$

This second step is called the *maximization step*.

These two steps are repeated until convergence is reached. The *log-likelihood* function is guaranteed to increase until a maximum (local or global or saddle point) is reached. For multivariate normal distribution, the expectation $E[.]$ (in Equation 3.16), which is denoted by p_{ij}, is the probability that Gaussian mixture j generated the data point i, and is given by:

$$p_{ij} = \frac{\left|\hat{\Sigma}_j\right|^{-1/2} e^{\left\{-\frac{1}{2}(x_i-\hat{\mu}_j)^t \hat{\Sigma}_j^{-1}(x_i-\hat{\mu}_j)\right\}}}{\sum_{l=1}^{M} \left|\hat{\Sigma}_l\right|^{-1/2} e^{\left\{-\frac{1}{2}(x_i-\hat{\mu}_l)^t \hat{\Sigma}_l^{-1}(x_i-\hat{\mu}_l)\right\}}} \tag{3.17}$$

The new estimates (at the k^{th} iteration) of parameters in terms of the old parameters at the M-step are given by the following equations:

$$\hat{\alpha}_j^k = \frac{1}{n}\sum_{i=1}^{n} p_{ij} \tag{3.18}$$

$$\hat{\mu}_j^k = \frac{\sum_{i=1}^{n} x_i p_{ij}}{\sum_{i=1}^{n} p_{ij}} \tag{3.19}$$

$$\hat{\Sigma}_j^k = \frac{\sum_{i=1}^{n} p_{ij}(x_i - \hat{\mu}_j^k)(x_i - \hat{\mu}_j^k)^t}{\sum_{i=1}^{n} p_{ij}} \tag{3.20}$$

More detailed derivation of these equations can be found in [48]. The semi-supervised algorithm for classification of remotely sensed imagery based on the EM algorithm just described above is given in Table 6.4.

3.4 Experimental Results

The Cloquet study site encompasses Carlton County, Minnesota, which is approximately 20 miles southwest of Duluth, Minnesota. The region is predominantly forested, composed mostly of upland hardwoods and lowland conifers. There is a scattering of agriculture throughout. The topography is relatively flat, with the exception of the eastern portion of the county containing the St. Louis River. Wetlands, both forested and non-forested, are common throughout the area. The largest city in the area is Cloquet, a town of about 10,000. We used a spring Landsat 7 scene, taken May 31, 2000. We designed four different experiments to understand the size and quality of initial labeled samples on the performance of semi-supervised learning, and the impact of unlabeled samples generated from random sampling and informed sampling methods. For all these

Inputs: Training dataset $D = D_l \cup D_{ul}$, where D_l consists of labeled samples and D_{ul} contains unlabeled samples.

Initial Estimates: Build initial classifier (MLC or MAP) from the labeled training samples, D_l. Estimate initial parameter using MLE, to find $\hat{\theta}$ (see Equations 3.9 and 3.10)

Loop: While the complete data *log-likelihood* improves (see Equation 3.15):

> **E-step:** Use current classifier to estimate the class membership of each unlabeled sample, that is, the probability that each Gaussian mixture component generated the given sample point, p_{ij} (see Equation 3.17).
>
> **M-step:** Re-estimate the parameter, $\hat{\theta}$, given the estimated Gaussian mixture component membership of each unlabeled sample (see Equations 3.18, 3.19, 3.20)

Output: A MLC or MAP classifier that takes the given sample (feature vector) and predicts a label.

Table 3.1: Semi-supervised Learning Algorithm

experiments the test dataset was fixed and consisted of 168 plots. Initial labeled and unlabeled samples were varied as explained in each experiment. Each plot, whether it was for training or testing, was a 3×3 window, that is, each plot contributed exactly 9 feature vectors centered on the plot location (x, y) coordinates.

The experimental setup was shown in Figure 3.3. Detailed experimental results are provided as tables in the Appendix. In these tables, 'EM(X)' field , where $X = 0|1$, stand for the weights assigned to the labeled samples. The 'Best' field stands for the best accuracy obtained by either of EM(0) or EM(1). Positive difference indicates how best semi-supervised algorithm performed over conventional Bayesian classifier (MLC) using only labeled samples. Likewise, negative difference indicate lower performance of semi-supervised learning over BC. For discussion purposes we summarized key results as graphs for easy understanding.

Experiment 1. The objective of this experiment was to understand the effect of initial labeled sample quality on semi-supervised learning. The labeled dataset consists of 5 disjoint subsets, each subset consisting of 20 plots at 2 plots per class. We have a fixed unlabeled training dataset consisting of 85 plots. The results are summarized in Table 3.2.

Experiment 2. The objective of this experiment was to understand the impact of the number of initial labeled samples for a fixed set of unlabeled samples on the quality of semi-supervised learning. We combined 2 sets of labeled samples at a time from the previous experiment to form ${}^5C_2 = 10$ labeled datasets, each consisting of $20 + 20 = 40$ plots. Experimental results for the same fixed unlabeled training dataset as in experiment 1 are summarized in Table 3.3.

Next, from 10 datasets in the above experiment we formed 3 different datasets by combining 3 at a time. After elimination of duplicate plots, each of these 3 datasets contains a total of 70 labeled sample plots. The results are summarized in Table 3.4.

Experiment 3. The objective of this experiment was to understand the quality and quantity of unlabeled training samples and their impact on overall performance of semi-supervised learning. For this experiment we devised two sampling schemes, simple random sampling, and informed sampling. For the simple random sampling, we generated 10 datasets, each consisting of multiples of 100 sample plots. No labels were available for these plots. For labeled sample plots we chose two datasets (best [B20] and worst [W20] in terms of accuracy) from the first experiment (see Table 3.2). The results are summarized in Table 3.5.

Experiment 4. From this table it can be seen that random sampling is not good for generating unlabeled training samples. A closer look at the individual class accuracies reveals two main problems. First problem is that the random sampling did not generate sufficient labels for each class; in fact there are very few or no samples at all for smaller (area) classes. The second problem is that some of the samples are not representative of the original labeled samples. These problems led us to believe that the unlabeled samples should be chosen in an informed (or constrained) way. There are several ways to do this. One way is to use some background information such as existing thematic maps from geographical information systems, for example, old land-use land-cover maps, ecological zone maps, or population density maps to manually choose these training plots (no labels though). A more automated way is to predict a rough thematic class map from the classifier trained on initial labeled samples, and then generate random samples guided by these rough predicted classes. Please note that we are only identifying the plot locations and not assigning any labels to them.

We call this second approach as informed sampling and used it to generate about 300 unlabeled sample plots. These plots were then randomly divided into 4 partitions. The first subset consists of 5 independent training sets, each consisting of 30 plots; second subset consists of 5 training datasets, each consisting of 60 unlabeled plots. Third experiment consists of 3 training datasets, each consisting of 110 unlabeled plots and finally the fourth experiment consists of 2 training datasets each consisting of 170 unlabeled plots. For labeled training we used the same two datasets that were used in experiment 3. For each of these labeled training datasets, semi-supervised learning was carried out against each of the unlabeled training datasets from the above 4 partitions and the results are summarized in Tables 3.6 and 3.7.

3.4.1 Discussion

From the first experiment it is clear that maximum likelihood estimates suffer in both quantity and quality of labeled training samples. The plot in Figure 3.4 shows that as the number of training (labeled) samples increases, the conventional maximum likelihood estimates gets better and hence the classification performance of the Maximum likelihood classifier (BC) also improves. It is also interesting to note that the difference between best and worst accuracies gets reduced as the number of samples increase. This is because the noise averages out as the number of samples increases.

The second experiment shows that as the number of labeled samples increases the usefulness of unlabeled samples diminishes (see Figure 3.5). Thus the main benefit of semi-supervised learning occurs when there is only a small number of labeled samples available for training.

In next two experiments we explore the impact of the number unlabeled training samples and how they are generated. Figure 3.6(a) and (b) provides the comparison of randomly generated unlabeled training plots against best and worst cases (labeled training data) taken from the experiment 1. On the other hand Figure 3.7(a) and (b) shows the results against unlabeled training plots generated by informed sampling. From these two experiments it is clear that accuracy increases as the number of unlabeled training samples increase, however pure random samples might degrade performance quite considerably. The main problem we noticed is that random sampling did not generate enough samples for small (geographic area) classes, as a result the corresponding covariance matrices are becoming singular or close to singular, and the mixing coefficients α_i are close to zero. On the other hand equal (or in proportion to class area) number of samples were generated for each class. It can be seen from the figure that the semi-supervised learning using informed sampling generated unlabeled training plots performed consistently well.

3.5 Conclusions

In this chapter we presented a basic semi-supervised learning algorithm. The semi-supervised method presented here uses the classical EM algorithm to augment unlabeled samples to improve initial estimates generated using a small set of training samples. Except for pure randomly generated unlabeled training samples, the semi-supervised learning showed an improved performance in many of the experiments. The overall accuracies varied between -8.67% and $+27.07\%$, and on an average the semi-supervised learning method showed an improvement of 8% in overall accuracy. Given the fact that this is a multi-class (10 classes) classification problem, the accuracies are higher than one would expect from coarse multi-spectral resolution images. This method is very useful in remote sensing data mining, as collection of sufficient training samples for supervised learning is often very difficult and costly. However, the results were not consistent. Basic semi-supervised approach is error prone, especially in the presence of unlabeled training samples from additional classes which were not defined in the labeled training data. Though informed sampling minimized picking up samples from the additional classes, the procedure is still ad hoc. In the next chapter we present a novel semi-supervised algorithm which overcomes this basic limitation.

3.6 Appendix

In this section we provide detailed experimental results.

BC (20L)	CI	CI	BC-EM (0) (20L +85UL)	CI	CI	BC-EM (1) (20L +85UL)	CI	CI	Best	Difference
47.42	44.87	49.97	45.24	42.7	47.78	50.2	47.65	52.75	50.26	+2.84
57.28	54.75	59.8	58.33	55.82	60.85	57.74	55.22	60.26	58.33	+1.05
45.04	42.5	47.58	56.94	54.42	59.47	46.16	43.62	48.71	56.94	+11.9
40.15	37.64	42.65	68.12	65.74	70.5	55.82	53.28	58.36	68.12	*+27.97*
65.08	62.64	67.52	63.62	61.17	66.08	61.24	58.75	63.73	63.62	*-1.46*

Table 3.2: Performance of semi-supervised learning method against small set (20) of labeled training plots and fixed unlabeled plots

BC (40L)	CI	CI	BC-EM (0) (40L +85UL)	CI	CI	BC-EM (1) (40L +85UL)	CI	CI	Best	Difference
65.48	63.05	67.91	66.01	63.58	68.43	68.19	65.81	70.57	68.19	+2.71
52.31	49.76	54.87	59.46	56.95	61.97	57.8	55.28	60.33	59.46	+7.15
55.82	53.28	58.36	61.57	59.09	64.06	68.25	65.87	70.63	68.25	+12.43
69.71	67.36	72.06	63.43	60.97	65.89	64.15	61.7	66.6	64.15	-5.56
58.86	56.35	61.38	63.82	61.37	66.28	48.94	46.39	51.49	63.82	+4.96
55.75	53.22	58.29	60.58	58.09	63.08	57.47	54.95	60	60.58	+4.83
71.83	69.52	74.13	58.02	55.68	60.72	63.16	60.7	65.63	63.16	-8.67
55.03	52.49	57.57	66.8	64.39	69.21	59.66	57.15	62.16	66.8	+11.77
66.87	64.46	69.27	62.57	60.09	65.04	54.83	52.29	57.37	62.57	-4.3
66.87	64.46	69.27	56.81	54.28	59.34	66.2	63.79	68.62	66.2	-0.67

Table 3.3: Performance of semi-supervised learning method against increased number (40) of labeled training plots

BC (70L)	CI	CI	BC-EM (0) (70L +85UL)	CI	CI	BC-EM (1) (70L +85UL)	CI	CI	Best	Difference
72.42	70.13	74.71	64.55	62.11	66.99	61.64	59.64	64.12	64.55	-7.87
70.11	67.77	72.45	59.52	57.02	62.03	63.03	60.56	65.5	63.03	-7.08
74.14	71.9	76.38	59.79	57.28	62.29	65.67	63.25	68.1	65.67	-8.47

Table 3.4: Performance of semi-supervised learning method against increased number (70) of labeled training plots

Unlabeled	BC-EM (W20)	BC-EM (B20)
100	46.76	62.96
200	44.44	49.34
300	49.01	61.77
400	45.11	59.59
500	54.17	42.33

Table 3.5: Performance of semi-supervised learning method against fixed labeled (20) and varying unlabeled labeled training plots (random sampling). B20 stands for the labeled dataset for which semi-supervised performed best (highest positive accuracy) in experiment 1. Likewise W20 stands for that labeled dataset for which semi-supervised accuracy is least amongst five labeled datasets.

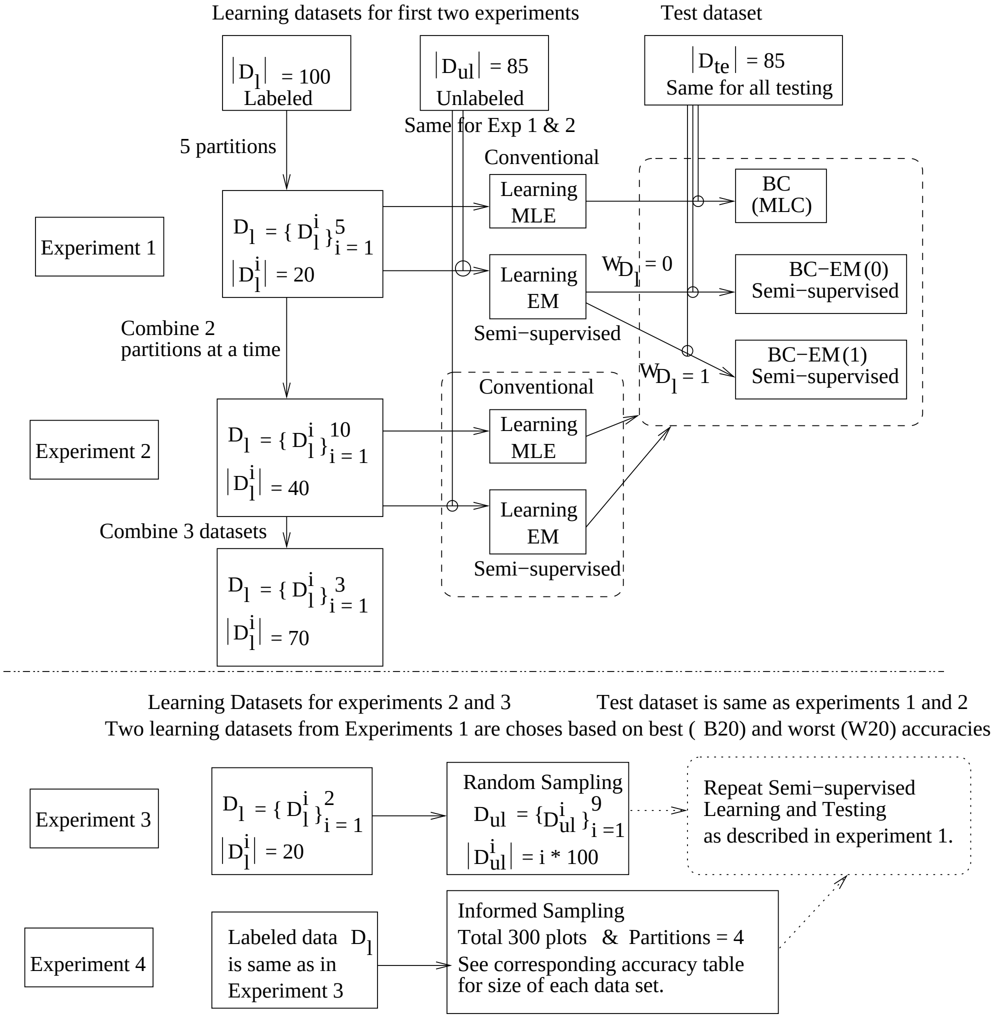


Figure 3.3: Experimental Setup

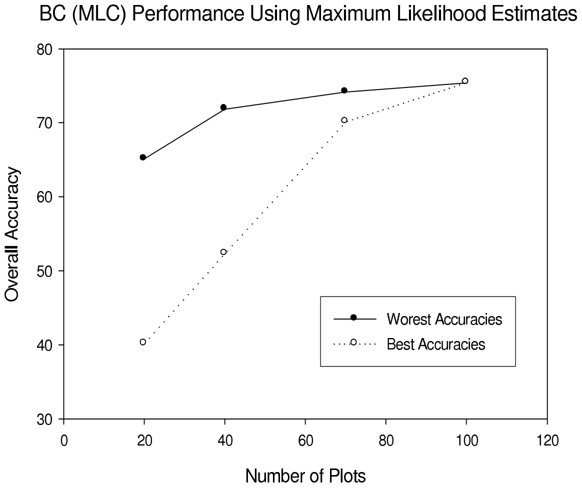


Figure 3.4: Performance of ML (Bayesian) classifier as the number of training samples increase.

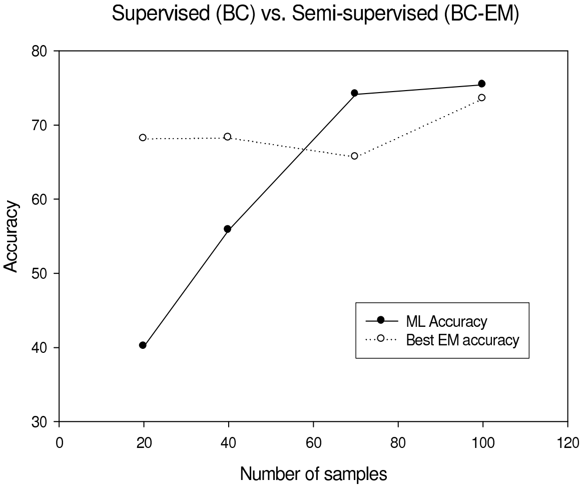
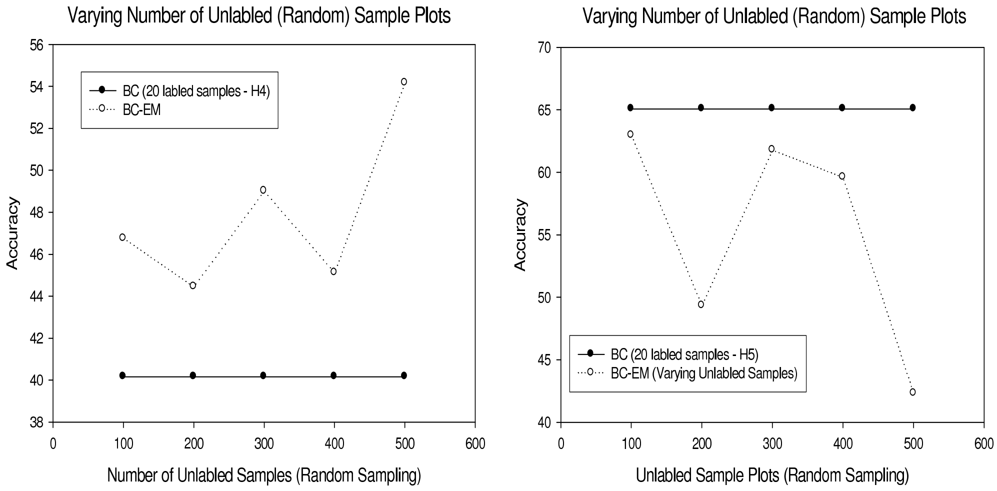


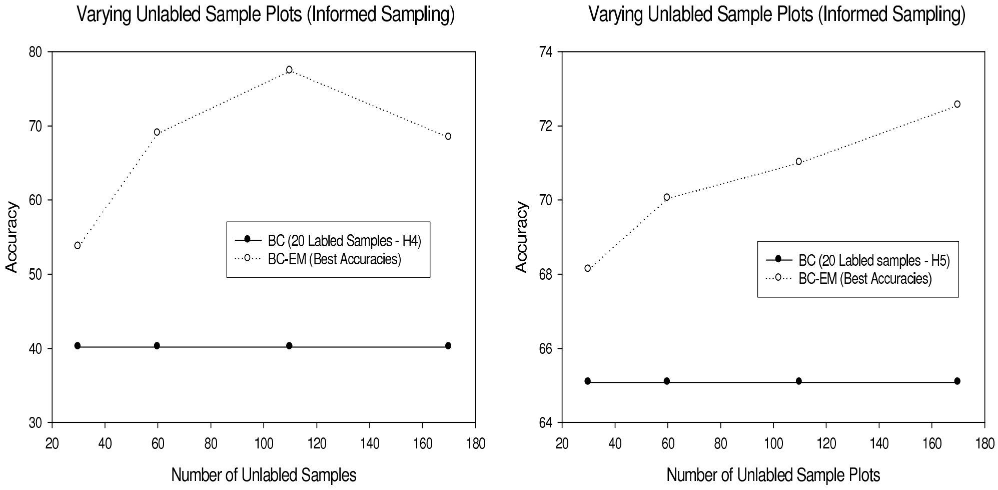
Figure 3.5: Performance of semi-supervised learning as the number of labeled samples increases.



(a) Against W20

(b) Against B20

Figure 3.6: Performance of semi-supervised learning as the number of unlabeled samples increases (random sampling).



(a) Against W20

(b) Against B20

Figure 3.7: Performance of semi-supervised learning as the number of unlabeled samples increases (informed sampling.

Unlabeled	BC-EM(0)	CI	CI	BC-EM(1)	CI	CI	Best
30	31.88	29.50	34.26	47.49	44.94	50.04	47.49
	35.12	32.68	37.56	46.43	43.88	48.98	46.43
	45.04	42.50	47.58	44.78	42.24	47.31	45.04
	43.32	40.79	45.85	53.70	51.16	56.25	53.70
	37.76	35.29	40.24	51.39	48.84	53.94	51.39
60	54.23	51.69	56.78	49.54	46.98	52.09	54.23
	64.88	62.44	67.32	46.69	44.15	49.24	64.88
	61.57	59.09	64.06	53.77	51.22	56.32	61.57
	68.98	66.62	71.35	49.34	46.79	51.89	68.98
	65.34	62.91	67.78	58.80	56.28	61.31	65.34
110	77.38	75.24	79.52	62.76	60.29	65.23	77.38
	72.55	70.27	74.84	56.08	53.55	58.62	72.55
	75.79	73.60	77.99	58.00	55.48	60.52	75.79
170	66.73	64.32	69.14	55.56	53.02	58.09	66.73
	68.39	66.01	70.76	55.59	53.42	58.49	68.39

Table 3.6: Performance of semi-supervised learning method for W20 labeled dataset against varying number of unlabeled training plots (informed sampling)

Unlabeled	BC-EM(0)	CI	CI	BC-EM(1)	CI	CI	Best
30	44.91	42.37	47.45	67.46	65.07	69.86	67.46
	42.46	39.94	44.98	58.13	55.62	60.65	58.13
	44.05	41.51	46.58	68.12	65.74	70.50	68.12
	33.33	30.29	35.74	66.80	64.39	69.21	66.80
	51.65	49.10	54.21	59.66	57.15	62.16	59.66
60	54.23	51.69	56.78	65.87	63.45	68.30	65.87
	64.88	62.44	67.32	70.04	67.70	72.38	70.04
	61.57	59.09	64.06	64.95	62.51	67.39	64.95
	68.98	66.62	71.35	60.58	58.09	63.08	68.98
	65.34	62.91	67.78	64.75	62.31	67.19	65.34
110	69.44	67.09	71.80	71.00	69.05	73.67	71.00
	65.67	63.25	68.10	60.25	57.75	62.75	65.67
	72.75	70.47	75.03	67.92	65.54	70.31	72.75
170	70.77	68.44	73.09	72.55	70.27	74.84	72.55
	71.96	69.66	74.25	62.04	59.56	64.52	71.96

Table 3.7: Performance of semi-supervised learning method for B20 labeled dataset against varying number of unlabeled training plots (informed sampling)

Chapter 4

Adaptive Semi-supervised Learning

4.1 Introduction

In the previous chapter we presented the basic semi-supervised learning algorithm, where we used unlabeled samples in conjunction with few labeled samples for the purpose of improving the parameter estimates. Semi-supervised approaches are becoming popular for several reasons. First, several previous studies have shown the positive value of adding unlabeled data into the supervised classification. However, the experimental results presented in previous chapter shows several limitations of the basic semi-supervised algorithm. Especially we found several instances where adding unlabeled data actually resulted in degradation of classification performance. Several instances can also be found in the literature where addition of unlabeled samples have degraded the classifier performance [47, 49, 45]. Though these studies showed the usefulness of unlabeled samples in improving classification of remote sensing images, these authors also noted several instances where actually the classification performance has degraded. Most notable theoretical study to understand the impact of unlabeled samples can be found in [50]. The basic assumption made in semi-supervised learning algorithms is the labeled and unlabeled samples were generated by the same statistical model (for example, Gaussian Mixture Model). When this assumption is correct, the authors in [47] observed that the covariance matrix estimated with labeled and unlabeled data is smaller than the covariance estimated from the labeled samples alone. Therefore, it can be expected that classification error can be reduced by adding unlabeled samples. More formal analysis of classification performance when the models are incorrect can be found in [50]. Let us now modify the semi-supervised learning problem formulation.

4.1.1 Adaptive Semi-supervised Learning: Problem Formulation

Let us now formalize the adaptive semi-supervised learning method.

Given:

A spatial framework $\mathcal{S}$ consisting of sites s_{ij} ordered as an $l \times p$ matrix, where l is the number of lines and p is the number of pixels, and $\{1 \leq i \leq l, 1 \leq j \leq p\}$.

A $d-$dimensional feature vector at each site s_{ij}, is represented by x_{ij}, where x_{ij} is a continuous random variable.

A set of discrete labels $Y = \{y_1, \ldots, y_k\}$, where k is the total number of distinct labels

Training Dataset $D = D_l \cup D_{ul}$, where D_l contains labeled training plots and D_{ul} contains unlabeled training plots.

A parametric model (e.g., Gaussian).

Find:

Estimate parameter vector Θ (e.g., $\{\mu_i, \Sigma_i\}$).

Objective:

Maximize complete data *log-likelihood* $L(\Theta)$.

Assumptions:

A1 The size of the labeled training dataset is less than 10 to 30 times the number of dimensions

A2 Thematic classes are separable

A3 Classes are unimodal

A4 Feature vectors are independent and identically distributed (i.i.d), but features are highly correlated in feature space (that is features are not independent).

A5 D_l and D_u samples are generated by GMM_l and GMM_u.

The assumption that labeled and unlabeled samples are generated by the same model may hold in certain domains, such as, text classification, however, this assumption is not true in almost all cases in the classification of remote sensing images. Simple reason being that the land use and land cover classification is domain specific. We found that in almost all cases, classification studies differ in the number classes. Literature survey shows that, same image has been used to extract different number of classes by different people. In general we observed that the image contains lot many classes than the analyst interested in. Therefore, at the minimum the statistical distribution models (say, GMM) of labeled samples and unlabeled samples differ in the number of components. Informed sampling presented in previous chapter tries to overcome this problem by reducing the chances of picking of samples from unknown components. However, that was only an ad hoc solution. Though the work in [47] alluded to this problem, no solution was provided. Therefore we relaxed the assumption (A5) in semi-supervised learning to account for the differences in the number of components in D_l and D_u. In this chapter we present a novel semi-supervised algorithm to solve the cases where the labeled and unlabeled models differ in the number of components.

	x	y
C1	50.00	40.00
C2	80.00	50.00
C3	60.00	40.00

Table 4.1: Simulation Parameters (Mean)

	C1		C2		C3	
	x	y	x	y	x	y
x	60.00	50.00	60.00	40.00	8.00	-3.00
y	50.00	70.00	40.00	90.00	-3.00	15.00

Table 4.2: Simulation Parameters (Covariance)

	x	y
C1	49.35	39.20
C2	74.50	42.91

Table 4.3: Estimated Parameters (Mean)

	C1		C2	
	x	y	x	y
x	49.20	36.32	84.33	69.04
y	36.32	54.06	69.04	141.02

Table 4.4: Estimated Parameters (Covariance)

4.1.2 Illustrative Example

First we illustrate the problem just described above with an example. We generated a Gaussian mixture model with three components, using the parameters listed in the Table 4.1 and 4.2. Figure 4.1 (a) shows the probability distribution of the model, solid ellipses represent original distribution and dotted ellipses represent the Gaussians computed from small samples generated from the original distribution (5 samples per class). Figure 4.1 (b) shows the parameters estimated using the semi-supervised learning by using unlabeled samples (150 per class) from all three classes (components). Figure 4.1 (c) shows the parameter estimated using the semi-supervised learning using same unlabeled samples however the for labeled samples we used only first two components (C1 and C2). From this figure we see that the mean of class 1 (black color) is shifted towards class 3 (green color).

We repeated the same experiment by changing the position of class 3 (that is centroid of the class), shape (covariance) of the class. Results were shown in Figure 4.2. In case 2 (Figure 4.2(a)), we shifted the centroid to [60,30] from its original location [60,40], however kept the same size and shape (that is, no change is covariance matrix) of the class. This change in position of class 3 has resulted in the shift of mean of both class 1 and class 2. In case 3 (Figure 4.2(b)) we changed both centroid and covariance matrix. We used the following new parameters: mean [70, 35] and covariance [60, 40; 40, 90]. In this case, both the mean and covariance matrix of class 2 are significantly impacted. In a nutshell, this experiment shows that when labeled and unlabeled samples differ in the number of components, then the unlabeled samples influence the parameters of the new model (both mean and covariance). This significantly impacts the classification performance, typically negatively, for two reasons. Deviation of centroids from their original positions may lead to increase in the overlap between different classes. Likewise increase in the variance of classes also leads to the increase in the overlap among the class distributions. We will show later that this increase in overlap between the class distributions leads to increase in the probability of error. The estimated parameters (case 3) are summarized in the Table 4.3 and Table 4.4.

4.2 Adaptive Semi-supervised Learning Algorithm

As seen in the previous example, presence of unlabeled samples from additional classes, leads to inconsistent estimation of parameters. Therefore simply combining both labeled and unlabeled samples and applying semi-supervised learning is clearly not a good idea. This observation has led us to develop a new semi-supervised which adaptively identifies additional components and eliminates the unlabeled samples from these components. The basic algorithm is shown in Figure 4.3.

We now explain each step.

1. We start with two data sets: labeled (D_l) and unlabeled (D_u)

2. We assume that both D_l and D_u are generated by Gaussian mixture models (GMM).

3. However, we assume that GMMs differ in the number of components, that is, $lk \neq uk$. Therefore, we can't simply pool samples from both data sets and apply regular semi-supervised learning

4. Using maximum likelihood estimation (MLE) technique, we estimate the parameters of GMM_{lk} using D_l. We use expectation maximization (EM) algorithm to estimate the parameters of GMM_{uk}.

5. Output of MLE is $\Theta_{lk} = \{\mu, \Sigma\}_1^{lk}$ and the out of EM is $\Theta_{uk} = \{\pi, \mu, \Sigma\}_1^{uk}$. The MLE is used to estimate parameters of each class and EM is used estimate parameters of each cluster.

6. The objective of matching step is to find class and cluster pairs that meets certain statistical hypothesis testing criteria. There are several options available for this step. We chose multivariate student T^2 test, which we will explain little later.

7. We chose all the samples from the class and cluster pairs which passed the matching criteria. These samples are pooled together to form the training data set $D = \{D_l, D_u\}$.

8. We apply semi-supervised algorithm based on EM (see algorithm presented in previous chapter).

9. Final output of adaptive semi-supervised algorithm is the parameter vector $\Theta = \{\pi, \mu, \Sigma\}_1^{lk}$

4.2.1 Matching

. One of the important step in the adaptive semi-supervised learning algorithm is the matching between the classes (labeled data) and clusters (unlabeled data), in order to see if there are any mixture components (i.e., clusters) for whose there is no corresponding class definitions (labeled samples) exists. This is very typical in the classification of remote sensing images. As the image contains lot many classes than a particular domain user is interested in. For supervised learning, the training samples were carefully chosen by the analyst, keeping in mind the classes he is interested in. However, unlabeled samples are generated by simple random sampling or some kind of stratified random sampling. As a result one can expect to find lot of unlabeled samples which don't have representative classes in the labeled training data set. Our objective is to find those clusters for which there are no corresponding classes in the (labeled) training data set. We pose matching as a statistical significance testing, where we are interested in finding if two process (class and cluster) have the same mean?

Let us assume that two random samples, one from class $(Y_1, \ldots, Y_N)$, and the other from cluster $(Z_1, \ldots, Z_N)$, are generated by independent processes. Then we can pose the null hypothesis to test the true mean of the first process μ_Y against the true mean of the second process μ_Z, as following:

$$H_0 : \mu_Y = \mu_Z \tag{4.1}$$

Assuming that the standard deviations (covariances in multivariate case) are same for both processes, we can now write the test statistic as following:

$$t = \frac{\mu_Y - \mu_Z}{s\sqrt{\frac{1}{N_Y} + \frac{1}{N_Z}}} \tag{4.2}$$

Here s is the pooled standard deviation given by:

$$s = \sqrt{\frac{(N_Y - 1))\sigma_Y^2 + (N_Z - 1))\sigma_Z^2}{(N_Y - 1)(N_Z - 1}} \tag{4.3}$$

with degrees of freedom $\nu = N_Y + N_Z - 2$.

Testing for Null hypothesis is done by computing t statistic using the formulae 4.2 and then perform significance testing at a given level α. Typical values of α are 0.01, 0.05 and 0.10. Null hypothesis is rejected if $|t| \geq t_{\alpha/2;\nu}$.

Possible scenarios as result of matching each cluster against a given class (one at a time) are following:

$Class_i(nomatch)Cluster_j^{uk}$ No unlabeled samples for a class. This may happen where spatial distribution of a class is sparse (e.g, water bodies, linear networks), therefore random sampling may not generate samples from a class. In those cases, an analyst may need to pickup samples manually or with appropriate stratification of image.

$Class_i(match)Cluster_j^{uk}$ In this case more than one cluster may match a given class. This might happen in cases where matched cluster are highly overlapping, or too many small clusters.

$Class_i^{lk}(nomatch)Cluster_j$ A cluster may not match any class. This is the typical case in remote sensing image classification. We eliminate these clusters. Better approach could be to revisit the class definitions to see if a new class can be added the training data.

$Class_i^{lk}(match)Cluster_j$ A cluster may match more than one class. Typically this should not happen if the classes are well defined. This may be a useful case where the classes are highly overlapping.

$Class_i(match)Cluster_j$ A cluster uniquely matches a class.

Classes	s1.m	s1.a	s2.m	s2.a	s3.m	s3.a
1	56.04	48.31	83.57	69.57	86.96	54.59
2	78.21	65.36	74.95	65.36	52.72	65.14
3	43.30	80.84	51.34	77.39	68.58	59.39
4	84.19	94.02	85.04	85.90	75.64	91.03
5	100.00	91.11	100.00	100.00	100.00	100.00
6	6.50	28.38	10.26	35.56	3.59	23.59
7	53.70	90.74	90.74	100.00	68.52	90.74
O. Acc	48.62	58.92	54.42	62.55	47.75	54.85

Table 4.5: MCL vs Adaptive SSL (data sets 1-3)

Classes	s4.m	s4.a	s5.m	s5.a	s6.m	s6.a
1	85.02	80.68	79.23	75.85	76.81	56.04
2	67.10	80.17	79.74	79.96	86.27	92.59
3	52.11	84.67	33.72	72.03	52.11	92.34
4	86.75	92.74	75.64	92.74	83.76	86.32
5	97.78	95.56	100.00	100.00	100.00	97.78
6	84.44	65.81	83.08	61.03	11.11	36.24
7	61.11	92.59	79.63	96.30	66.67	96.30
O. Acc	75.56	78.64	74.20	74.96	55.99	70.03

Table 4.6: MCL vs Adaptive SSL (data sets 4-6)

4.3 Experiments

We applied the adaptive semi-supervised learning algorithm on the Carlton County data set described in the previous chapters. Labeled data set consisted of two plots per class. We collected five hundred unlabeled training samples through the random sampling. Results are summarized in Tables 4.5- 4.7. Overall accuracy of adaptive semi-supervised learning against maximum likelihood classification is summarized in the form of bar plots shown in Figure 4.4. Adaptive semi-supervised learning algorithm provided consistently

Classes	s7.m	s7.a	s8.m	s8.a	s9.m	s9.a	s10.m	s10.a
1	88.89	88.41	92.27	86.47	82.13	84.54	90.82	71.50
2	50.76	65.14	32.68	35.95	77.34	76.03	76.25	93.68
3	92.34	92.72	78.93	94.64	50.96	72.03	46.74	84.29
4	60.68	41.45	42.74	57.69	57.69	42.31	44.44	58.55
5	100.00	100.00	93.33	95.56	100.00	100.00	100.00	100.00
6	40.00	46.50	84.44	71.62	88.89	81.03	50.94	44.10
7	83.33	96.30	83.33	96.30	72.22	94.44	79.63	96.30
O. Acc	60.92	64.50	66.56	67.21	75.72	74.85	62.33	69.92

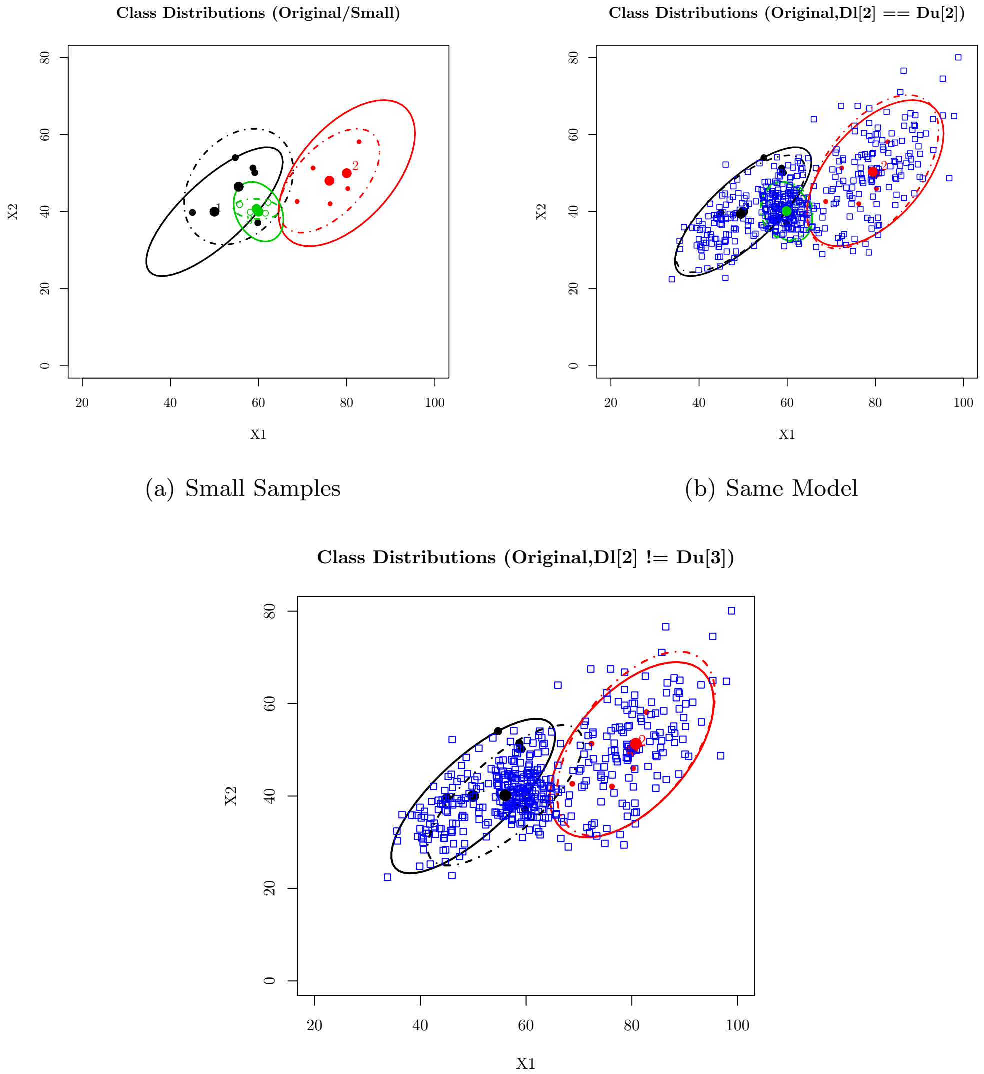
Table 4.7: MCL vs Adaptive SSL (data sets 7-10)

better results than MLC. We observed slightly lower overall accuracy (-0.87%) on only data set. The best overall accuracy is 14% improvement over the ML classifier. Our main observation is that adaptive semi-supervied learning avoids convergence to wrong parameters (which causes shift in means and increase in covariance) by eliminating impact of unlabeled samples from additional components (clusters).

4.4 Conclusions

In this chapter we addressed an important limitation of the expectation maximization based semi-supervised learning algorithms. Semi-supervised algorithms assume that both labeled and unlabeled are generated by the same model. However, in practice we observed that the labeled and unlabeled models differ in the number of components. Though theoretical studies [50] addressed the implications of unlabeled samples on the classification performance, very little attention was given to overcome this important limitation of the semi-supervised algorithms. We developed a novel adaptive semi-supervised learning algorithm, which automatically finds the samples from the additional components of the unlabeled data model. By eliminating samples from the irrelevant components with respect to the labeled data model, our adaptive semi-supervised approach overcomes an important side effect that has direct bearing on the accuracy. Pooling irrelevant unlabeled training samples with labeled training samples leads either shift in the location parameter (means) or size and shape parameter (covariance) or both. This shift in the mean or increase in covariance leads to increase in the overlap of class distributions. Since probability of error is related to the overlap between class conditional probability distributions, increase in this overlap leads to the increase in the probability of error. As can be seen from the experimental results, adaptive semi-supervised learning algorithm overcomes this problem by eliminating samples from additional components using the well-known statistical hypothesis testing. As a result, the adaptive semi-supervised learning algorithm has given consistently better results as opposed to the semi-supervised learning algorithm.

In this chapter we focused only on statistical hypothesis testing for matching classes and clusters. However, statistical hypothesis tests are sensitive to model assumptions and noise. We also assumed that the covariance is same for clusters and classes, which may not be true. We conducted initial experiments with two other statistical measures. One is KL-Divergence and the other is transformed divergence. These measures gives a sense of closeness between two statistical distributions, thus we can combine two distributions which are very close or highly overlapping. The results are encouraging, however further experimentation is need in order to understand the overall solution quality or scenarios where these measures are more desirable than the statistical hypothesis testing.

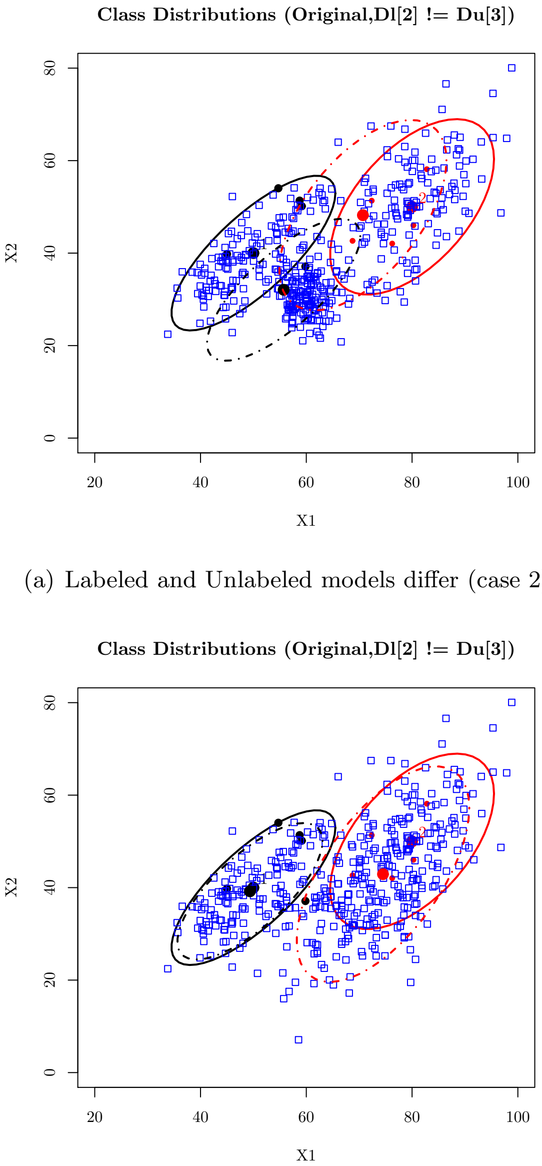


(a) Small Samples

(b) Same Model

(c) Labeled and Unlabeled models differ (case 1)

Figure 4.1: Example showing problem with SSL when models differ



(a) Labeled and Unlabeled models differ (case 2)

(b) Labeled and Unlabeled models differ (case 2)

Figure 4.2: Example showing problem with SSL when models differ (Case 2 and Case 3)

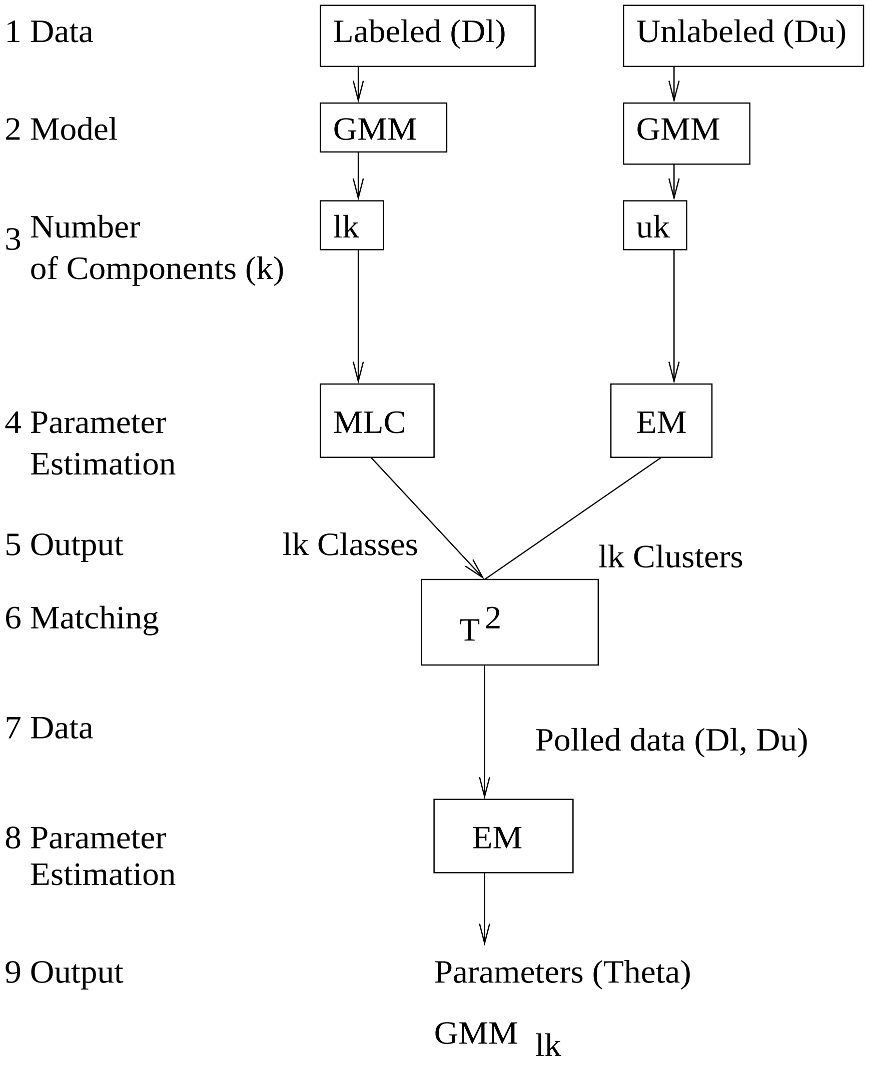


Figure 4.3: Flow of the Adaptive SSL Algorithm

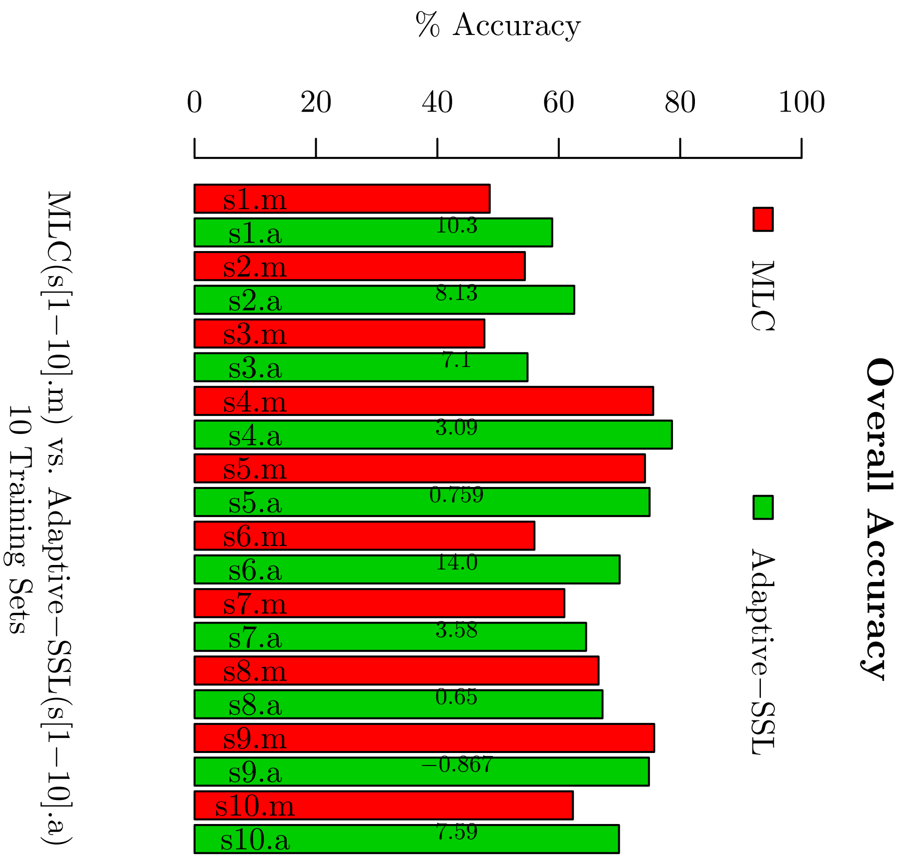


Figure 4.4: Overall accuracy comparison (MLC vs. Adaptive SSL)

Chapter 5

Overlapping Classes: Multisource Data Classification

5.1 Background

MLC is the most widely used method for land cover classification based on multi-spectral remote sensing imagery because of its simplicity and efficiency. However, multi-spectral image classifiers, like MLC, use only the spectral information of the pixel to be classified. However, in practice thematic class definitions often doesn't map well to the spectral classes. For example, upland hardwood and lowland hardwood forest may have similar statistical distribution (if not exactly same). Thus class conditional probability distributions of such classes may be highly overlapping and can't be separated easily without using additional ancillary geospatial data. Fortunately, MLC is very flexible and several researchers have obtained improved classification results by incorporating *a priori* information from ancillary spatial databases into MLC [34], [51]. Another way to incorporate ancillary information is to stratify the image into spectrally homogeneous regions before classification. Stuckens et al. [52] have used pre-classification stratification based on physiography. Training samples were then developed separately for each physiographic stratum. The main purpose of this stratification was to subdivide the large geographic region into physiographically homogeneous zones to avoid confusion between training signatures. However one problem associated with the stratification approach is that there may be artificial contours in the final classified image if appropriate criteria are not adopted for stratification.

An alternative to MLC with *a priori* ancillary information is expert system classification. Expert systems for remote sensing image classification have been considered by several researchers [53], [54]. The main advantage of expert system classification is that it allows the integration of knowledge about the environment into the classification process. The main disadvantage of these systems is the difficulty of developing a complex knowledge base with correct and consistent rules. In the approach used by Skidmore [53], the expert system infers the most probable species class at a location using *a priori* probabilities, computed using Bayes' theorem, for all the items of evidence (e.g. possible thematic class, gradient, aspect). The approach used by Bolstad et al. [54] is based on the concept of the classification model, where a classification model is defined as an automated sequence of operations applied to image and non-image spatial data, which results in a land cover classification. As noted earlier, the major limitation of knowledge based approaches is the complexity of the knowledge base. For example, the KB used by Bolstad et al. [54] contains about 200 rules. For real use of KBs to become practical, more research is needed on continuous learning and automatic population of the KB and consistency checks for KB rules.

In this chapter we present three approaches to improve overlapping class classification problem by using ancillary geospatial data. First let us formalize the multisource data classification.

5.1.1 Multisource Data Classification: Problem Formulation

Given:

A spatial framework $\mathcal{S}$ consisting of sites s_{ij} ordered as an $l \times p$ matrix, where l is the number of lines and p is the number of pixels, and $\{1 \leq i \leq l, 1 \leq j \leq p\}$.

A $d-$dimensional feature vector at each site s_{ij}, is represented by x_{ij}, where x_{ij} is a continuous random variable.

A set of discrete labels $Y = \{y_1, \ldots, y_k\}$, where k is the total number of distinct labels

Training Dataset $D = D_l \cup D_{ul}$, where D_l contains labeled training plots and D_{ul} contains unlabeled training plots.

A parametric model (e.g., Gaussian).

Find:

Estimate parameter vector Θ (e.g., $\{\mu_i, \Sigma_i\}$).

Objective:

Maximize complete data *log-likelihood* $L(\Theta)$.

Assumptions:

A1 The size of the labeled training dataset is less than 10 to 30 times the number of dimensions

A2 Thematic classes are separable

A3 Classes are unimodal

A4 Feature vectors are independent and identically distributed (i.i.d), but features are highly correlated in feature space (that is features are not independent).

A5 D_l and D_u samples are generated by GMM_l and GMM_u.

A6 Input data (x) are a combination of continuous and discrete random variables.

As compared to semi-supervised learning, the multisource data classification problem has two changes. First, we were given ancillary geospatial databases. Second, we relaxed the assumption that x are continuous random variables. In case of multisource data we have to deal with both continuous and discrete random variables.

We now present our first which is based on combining the best features of statistical pattern recognition techniques, such as MLC, and knowledge based systems. We eliminate the main limitation of the KB by i) simplifying the KB to a restricted set of rules, and ii) changing the objective from classification of individual pixels to stratification of the image into spatially homogeneous and disjoint regions.

5.1.2 *Data Sources and Pre-processing*

This research was carried out in the north east portion of the metropolitan area of Minneapolis-St. Paul, Minnesota, U.S.A. The southern part of the area is characterized by high-density urban, industrial and residential build-up, lakes, grass fields (including golf courses), and lawns. The northern part consists of agricultural fields, wetlands, hardwood and conifer forests, lakes, and low-density residential settlements. The satellite imagery used in this study was acquired on May 15, 1998 by the Landsat Thematic Mapper (TM). Several ancillary spatial data sets were used to extract spatial knowledge suitable for stratification of the satellite imagery. These data sets include the 1990 Census, 1995 TIGER road maps, and the National Wetlands Inventory (NWI). Satellite imagery was geometrically corrected and geo-referenced to the UTM projection by collecting about 30 ground control points and fitting a second order polynomial equation with an RMS error less than 7.5 meters (i.e. about a quarter of pixel accuracy). Color infrared aerial photographs and existing thematic maps, along with ground visits, were utilized in the training phase.

5.2 Spectral and Spatial Knowledge-based classification

Our classification system consists of four major modules: spectral knowledge, spatial knowledge, semi-automated learning, and classification, each of which is described below.

Spectral Knowledge: Object extraction from spectral relationships only is almost impossible, nonetheless it is interesting and useful to find simple spectral rules, like: $\forall Pixel(p),\ IF(band1(p) > band2(p) > band3(p))\ THEN\ Output(p) =$ 'WATER'. Even though finding such rules is difficult, the main contribution of spectral knowledge is in finding inherent data structures within the image. Often transformations, like normalized density vegetation index (NDVI) and Tasseled Cap (TC), will yield more insights into the structure of the data. The Tasseled Cap concept involves identifying the existing data structures for a particular sensor and application and changing the viewing perspective such that those data structures can be viewed directly [55]. We have extracted spectral knowledge derived from greenness index channel of the TC transformation for stratifying the TM image. The rules used are summarized in Table 5.1.

Knowledge Base	Class/Region
TM B1 $>$ B2 $>$ B3 $>$ B4	Water
$(TassledCap.Greenness \leq 15)$ && $(Pop.Density\ > 5000)$ || $(Road.Density\ > 0.0145)$	High density developed
$(15 > TassledCap.Greenness \leq 25)$ &&$(1000 \leq Pop.Density\ leq5000)$ || $(0.0078 \leq Road.Density\ \leq 0.0145$	Low density developed

Table 5.1: Spectral and Spatial Knowledge Base.

Spatial Knowledge: The purpose of the spatial knowledge base is to stratify the TM image into homogeneous regions with the following properties:

Let R be any given image.
The purpose of image stratification is to find a
finite set of regions $R_1, R_2, ..., R_q$, such that
$R = \bigcup_{i=1}^{q} R_i, \quad R_i \cap R_j = \emptyset$
and k classes $C_{ik}\ \varepsilon\ R_i$ are spectrally separable
(i.e. inter-class variation is minimum and
intra-class variation is maximum).

Our objective is to find regions in such a way that signature continuity holds within any region R_i and for any class: if $C_{rk} = C_{jk}$, then r = j. But in practice we may not find such regions, so there may be some common classes among the regions. In the training phase we have to collect sufficient samples for overlapping classes to avoid artificial contours in the final classified image. Careful study of the TM image shows that we can find two distinct regions called ‘developed lowlands’ and ‘undeveloped uplands’. The flow chart for extracting spectral and spatial knowledge to derive these two regions is shown in Figure 5.1. The knowledge base is summarized in Table 5.1.

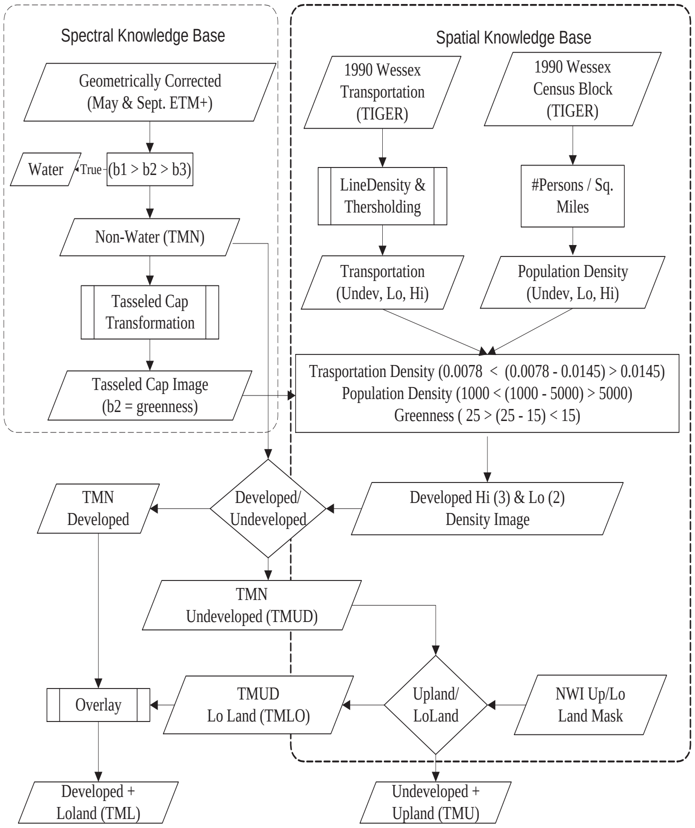

Spectral Knowledge Base
Geometrically Corrected (May & Sept. ETM+)
Water
True
(b1 > b2 > b3)
Non-Water (TMN)
Tasseled Cap Transformation
Tasseled Cap Image (b2 = greenness)
Spatial Knowledge Base
1990 Wessex Transportation (TIGER)
1990 Wessex Census Block (TIGER)
LineDensity & Thersholding
#Persons / Sq. Miles
Transportation (Undev, Lo, Hi)
Population Density (Undev, Lo, Hi)
Trasportation Density (0.0078 < (0.0078 - 0.0145) > 0.0145)
Population Density (1000 < (1000 - 5000) > 5000)
Greenness (25 > (25 - 15) < 15)
TMN Developed
Developed/ Undeveloped
Developed Hi (3) & Lo (2) Density Image
TMN Undeveloped (TMUD)
Overlay
TMUD Lo Land (TMLO)
Upland/ LoLand
NWI Up/Lo Land Mask
Developed + Loland (TML)
Undeveloped + Upland (TMU)

Figure 5.1: Flow chart showing the use of spectral and spatial knowledge base

Semi-automated Learning: Sample plots were collected for the required classes 'to be used as seed points'. A region growing algorithm was applied at each of the seed points to populate polygons with homogeneous characteristics. Approximately 25 aerial photographs and additional ground truth observations were utilized in collecting sample plots. The main criterion used in region growing was obtaining minimum region of N+1 pixels (where N is the number of spectral bands) to 25 pixels within a spectral Euclidean distance of 10 pixels. For a N-dimensional multi-spectral space, we need at least N+1 pixels to avoid a singular covariance matrix. We chose the 25-pixel criterion to check that the sample comes from a homogeneous area. We can't compute this threshold beforehand, so we have iteratively varied the thresholds to reach an optimum limit satisfying the above criteria and eliminated some of the seed points during this process. Once the training samples were collected, training statistics were generated and analyzed both visually and quantitatively to check the between-class separability. Co-spectral, ellipsoidal plots in two-dimensional feature space provide first-hand visual information about between-class separability. An example plot is shown in Figure 5.2. As can be seen from the figure, the ellipses (classes) are fairly separable.

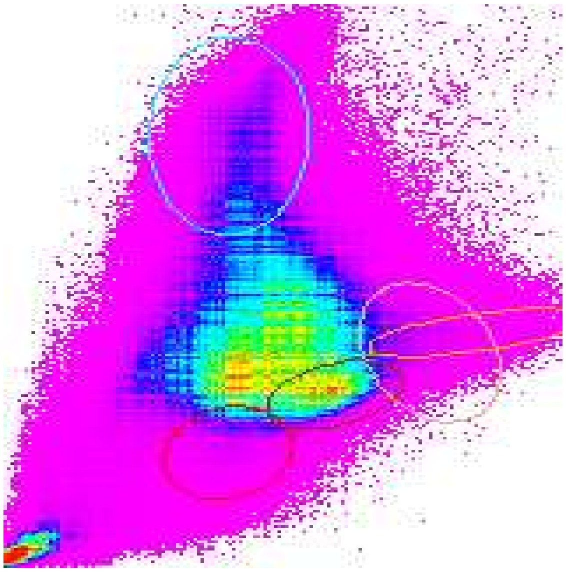


Figure 5.2: Between-class separability in feature space

Transformed divergence was computed using the following equation:

$$TD_{ab} \;=\; 2000[1 - e^{(\frac{-Diverg_{ab}}{8})}], \tag{5.1}$$

where $Diverg_{ab}$ is the divergence between the classes a and b. $Diverg_{ab}$ is computed using the following equation:

$$\begin{aligned} Diverg_{ab} \;&=\; \frac{1}{2}tr[(V_a - V_b)(V_b^{-1} - V_a^{-1})] \\ &\quad + \frac{1}{2}tr[V_a^{-1} + V_b^{-1})(M_a - M_b)(M_a - M_b)^T], \end{aligned}$$

where $tr[\cdot]$ indicates trace of matrix, V_a and V_b are the covariance matrices for any given two classes a and b, and M_a and M_b are the mean vectors for class a and b. The transformed divergence given in equation 1 gives an exponentially decreasing weight to increasing distances between the classes and also scales the divergence values between 0 and 2000. A transformed divergence value of 1900 and above indicates good separation between classes under investigation and a 2000 represents a perfect between-class separation. Training samples with low TD values were carefully studied and either merged or deleted based on ground truth verification. We have achieved, a final average divergence of 1944 for 'developed lowlands' and 1950 for 'undeveloped uplands'. The flow chart for supervised learning is shown in Figure 5.3.

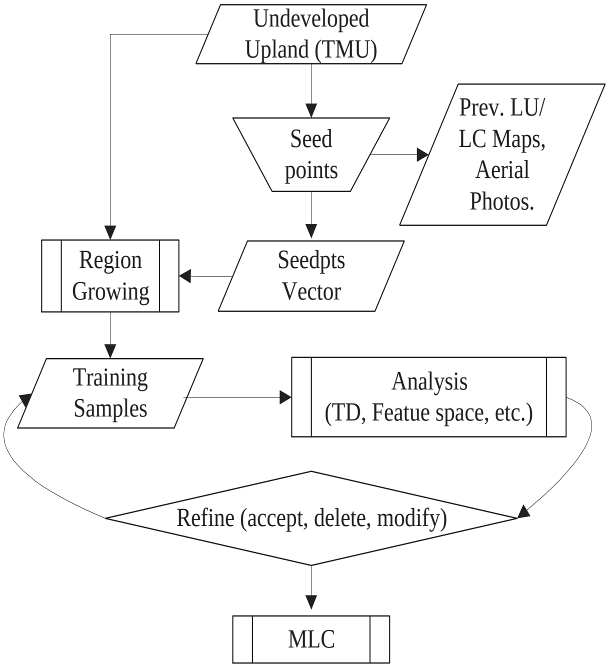

Undeveloped Upland (TMU)
Seed points
Prev. LU/ LC Maps, Aerial Photos.
Region Growing
Seedpts Vector
Training Samples
Analysis (TD, Featue space, etc.)
Refine (accept, delete, modify)
MLC

Figure 5.3: Semi-automated supervised learning scheme

Classification: Classification is performed using MLC with the following discriminant function:

$$g_i(x) = -ln|\Sigma_i| - (x - m_i)^t \Sigma_i^{-1}(x - m_i), \tag{5.2}$$

where m_i and Σ_i are the mean vector and covariance matrix of the training data for class ω_i. Any given pixel vector x is assigned to ω_i if $g_i(x) > g_j(x) \quad \forall j \neq i$. More details on transformed divergence and maximum likelihood classification methods can be found in [56], [35]. All classified regions are merged to obtain a final classified image as shown in Figure 5.4.

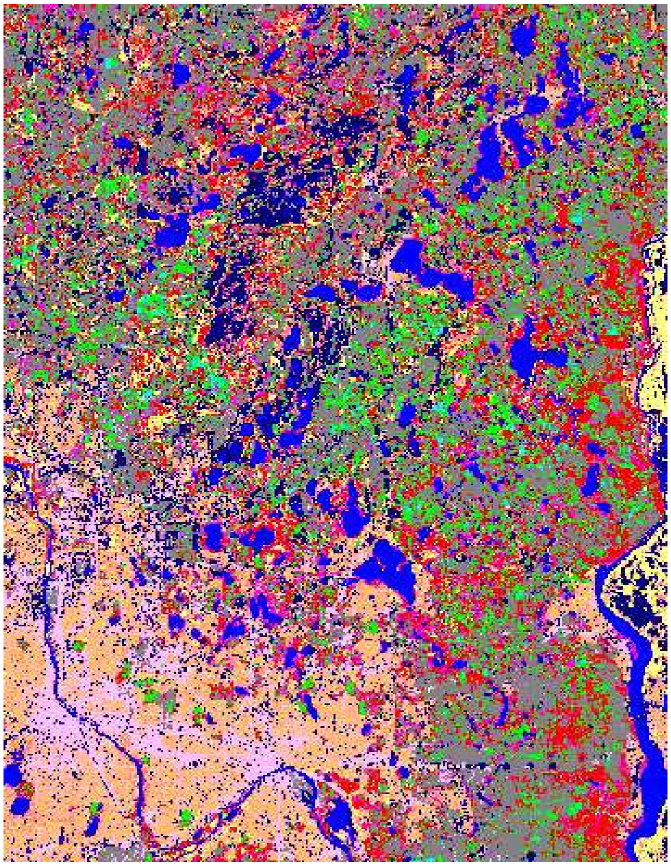


Figure 5.4: Final classified image

Accuracy assessment: We randomly collected about 190 sample plots for accuracy assessment. These sample plots are verified using large scale aerial photographs and field visits. Individual class accuracies are summarized in the Table 5.7. Overall accuracy is about 85%. However, accuracy is low for lowland conifer, wetland, and low density urban because of high spectral overlap among these classes. Additional knowledge base development is needed to improve the accuracy among these classes.

5.2.1 Hybrid Classification System

We now describe our second approach. Our approach is to combine traditional statistical pattern recognition methods and machine learning algorithms like decision trees

Class	Accuracy (%)
Water	100
Bare Soil	100
Crop	85
Upland Conifer	100
Upland Hardwood	97
Lowland Hardwood	63
Lowland Conifer	71
Wetland	71
Low density Urban	100
Hi density Urban	67
Overall Accuracy	85

Table 5.2: Accuracy Estimation

such that the resulting hybrid system yields better predictions (classifications) and minimizes some of the known disadvantages. We already observed that no single classifier provides the best solution, and in fact, it is observed in several previous studies that the classification accuracy of individual classes varies greatly, though the overall classification accuracy is comparable. This very nature of classifier performance on individual classes has led to the design of multiple classifier systems (MCSs). The primary objective of MCSs is to combine the decisions from different classifiers in such a way that the overall accuracy improves. Recent studies by Smits [57], Bruzzone at el. [58] have demonstrated the usefulness of MCSs for remote sensing image classification. Our hybrid classification system is designed on the same guiding principles, however, instead of combining several classifier decisions at the end as in MCSs, our approach utilizes the relative strength of each classifier and applies them in a sequence (stages) such that the final classification accuracy is improved.

A hybrid classification system is defined as follows:

Given: $I = X_1, \ldots, X_M, O_1, \ldots, O_P$ be a set of images, where X_i denote raw images and O_i denote intermediate output generated by f_i,

$S = S_1, \ldots, S_R$ be a set of ancillary geo-spatial data layers,

$T = t_1, \ldots, t_Q$ be a set of training samples,

$L = \omega_1, \ldots, \omega_C$ be a set of labels (classes).

$F = f_1, \ldots, f_N$ be a set of classifiers, where $f_k : \{I_i; S_s; T_t; L_l\} \rightarrow O_k$, and I_i, S_s, T_t, and L_l are subsets of I, S, T, and L respectively. These subsets could be null, for example, if f_i is a clustering algorithm, then S_s; T_t; L_l are all null sets.

Find: Hybrid classifier H, where $H = f_i[f_k[\ldots]]$ is a sequence of classifiers, such that the classification accuracy is maximized.

The main problem in defining H is to find the optimum sequence of classifiers. The performance of each classifier f_i depends on several factors, ranging from model assumptions (e.g., class distribution) to feature sets used. There is no easy way of automatically selecting this sequence, but we used the following guiding principles in deriving this sequence.

We used unsupervised clustering algorithms to overcome the need for accurate training data in the first stage. Several clustering algorithms have proven to be very useful in remote sensing analysis. However, the main problem with the clustering algorithms is mapping these spectral clusters into ground classes. There is no simple way to do this, so the clustering approach is mainly used to find natural clusters in the data with subsequent training samples taken from these clusters. However, we observe that ancillary geo-spatial data can be efficiently used to map these spectral clusters into ground classes. Thus in the second stage, a decision tree is used to guide the classification process using ancillary geo-spatial data. This framework offers greater flexibility because the classification requirement now is reduced to map the entire cluster rather than individual pixels as in the traditional approach. On the other hand, if a supervised classifier like MLC, or a decision tree is chosen at the first stage, then one requires an accurate and sufficiently large learning data set. Direct incorporation of ancillary geo-spatial data S into MLC is difficult (because the lack of a convenient multivariate statistical model) and the true benefits cannot be fully exploited by abstracting the spatial knowledge into *'a priori'* probabilities. Similarly decision trees also require a large learning data set in order to find accurate decision boundaries in feature space. By using clustering in the first stage, we not only avoid the need for large training data, but also reduce the feature space dimensionality (typically from 6 to 1) and the number of data points to be further classified (typically from millions of points to a few thousand polygons). This step also eliminates the need for large training data in the second stage, as the classification task is now reduced to finding a simple set of rules to map the spectral clusters into thematic classes. Each of these classifiers are briefly described below.

5.2.2 C-Means Clustering

Cluster analysis is an often performed data analysis technique in many fields. This has resulted in a multitude of clustering algorithms which can be broadly classified into the following four groups: Hierarchical, Partitional, Density-based, and Grid-based. More details on clustering algorithms and cluster validity measures can be found in recent survey papers [59], [60], [61]. The most often used clustering method in remote sensing is C-Means clustering which belongs to the Partitional clustering group. The main premise of Partitional clustering is to create one cluster for each pattern and iteratively reallocate data points to each cluster until a stopping criterion is met. These methods tend to find clusters of spherical shape. *K-Means* and *K-Medoids* are some other commonly used partitional algorithms. The *C-Means* algorithm is an enhancement over the *K-Means* algorithm and is given as follows:

1. Start with an initial partitioning of the data into k clusters

2. Compute the centroids of the resulting clusters

3. Generate a new partition by assigning each sample to its closest cluster centroid

4. Repeat 2 to 3 until an optimum value of the criterion function is found (e.g., Minimum Square Error)

5. Adjust the number of clusters by merging existing clusters or by removing small and outlier clusters

The algorithm is briefly explained here, but more details can be found in [62]. In *C-Means*, new cluster creation or existing cluster merging are determined by the parameters provided by the user. A cluster is split if it has too many patterns and, usually, a larger variance along the feature with the largest spread. Two clusters are merged if their cluster centers are sufficiently close, again based on a parameter supplied by the user. We have chosen unsupervised classification because it requires a minimal amount of input from the user as opposed to supervised classification, which requires highly accurate training data. Clustering algorithms automatically search for the natural groupings of pixels in the multi-spectral feature space. By applying clustering first, the classification problem is now reduced to finding an f_i which efficiently maps these spectral clusters into thematic classes.

5.2.3 Decision Tree Classification

A decision tree (DT) is a nonlinear classifier that recursively partitions a data set into smaller subdivisions based on a set of simple tests at each internal node in the tree. The leaf nodes represents the class labels ω_i. Training samples are used to learn the split conditions at each internal node and to construct a decision tree. For each new sample (i.e., feature vector x), the classification algorithm will search for the region along a path of nodes of the tree to which the feature vector x will be assigned. That is, the classification of a region is determined by a path from the root node to a leaf node. In previous studies decision trees were used for remote sensing classification with accuracies that were comparable to MLC [37] and global land cover classifications [5], [7]. Performance of three different types of decision trees, namely, univariate, multivariate, and hybrid decision trees for remote sensing data classification, against MLC were reported in [6], and it is found that decision trees produced higher classification accuracies. Most common decision trees split the feature space into hyper-rectangles with sides parallel to the axis. Oblique decision trees [38] find the optimal hyper-plane (not necessarily axis-parallel) for each node of a decision tree. In this study, we used C4.5, a publicly available univariate decision tree software. For algorithmic details, see Quinlan [39].

5.2.4 Maximum Likelihood Classifier (MLC)

The MLC is one of the popular classification schemes, and is often used as the basis for comparison against other classifiers. The MLC is based on Bayes' classification rule, which assigns a feature vector x to the class ω with the highest conditional probability. In practice, it would be very difficult to compute conditional probabilities for each feature vector x. But if sufficient training data are available for each thematic class, then it is possible to estimate a probability distribution for each class that describes the chance of finding a pixel from class ω_i, i.e., $p(x|\omega_i)$ say, at the position x. In order to compute the probability distributions for classes, we have to make certain assumptions about the underlying distribution of the population (e.g., multi-variate normal or Gaussian). Classifiers which make assumptions about the underlying parameterized probability density functions are called *parametric* classifiers. This assumption simplifies the computation of parameters like mean, standard deviation and cross-covariance matrix from the training data. For a multi-variate normal distribution, the discriminant function can be written as

$$g_i(x) = \ln p(\omega_i) - \frac{1}{2} \mid \Sigma_i \mid - \frac{1}{2}(x - m_i)^t \Sigma_i^{-1}(x - m_i) \tag{5.3}$$

where m_i and Σ_i are the mean vector and covariance matrix of the data in class ω_i. Then the MLC assigns any given feature vector x to a class ω_i, if $g_i(x) > g_j(x)\ \forall_{j \neq i}$. However, one disadvantage of MLC (as with any parametric classifier) is the strict assumption of unimodal normal distribution; as a result, the analyst has to train for each of the sub-classes which correspond to a mode in the distribution. In an extreme case, this problem may also arise due to *signature extension.* Signature extension refers to the concept of assuming that a defined class or object in one spatial region can be extended to another region without requiring changes in the model, such as additional ground truth or investigation by the user [63]. However, due to several extraneous factors, signature extension may not hold, and we may have to train the classifier accordingly. Also parametric classifiers like MLC cannot incorporate ancillary geo-spatial databases (like soil, elevation, slope, aspect) into the classification process because the probability distribution (e.g., Gaussian) does not hold with these ancillary data sets. The parametric classifiers will also perform poorly if the underlying distribution do not conform to the assumed distribution (for example, any given class may not be normally distributed, or the distribution may contain more than one mode). It is well known that several spatial relationships exist between the images and geo-spatial databases, as the spectral distribution is a function of objects and the physical environment (e.g., slope, aspect, soil type). Several recent studies have aimed at incorporating *a priori* probabilities computed from these ancillary data into MLC. Studies by [34], [51] show that accuracy can be improved by incorporating *a priori* probabilities computed from ancillary geo-spatial data sets. *A priori* probability can be thought of as a scaling factor which shifts (scales down or scales up) the decision boundaries in proportion to the expected class volumes (frequency) in N-dimensional feature space. Though these approaches may marginally increase accuracy, they have inherent limitations. For example, since *a priori* probabilities are a global phenomenon, it cannot accurately model local or regional spatial relationships. We have trained our MLC with *a priori* probabilities derived from the ancillary data sets as suggested in [51].

5.3 Methodology and Results

We now describe major components of the proposed approach.

5.3.1 Data Sources and Pre-processing

This research was carried out in the north east portion of the metropolitan area of Minneapolis-St. Paul, Minnesota, U.S.A. The southern part of the area is characterized by high-density urban, industrial and residential build-up, lakes, grass fields (including golf courses), and lawns. The northern part consists of agricultural fields, wetlands, hardwood and conifer forests, lakes, and low-density residential settlements. The satellite imagery used in this study was acquired on May 15, 1998 by the Landsat Thematic Mapper (TM). Several ancillary geo-spatial data sets were used to aid classification process. These data sets include the 1990 Census, 1995 TIGER road maps, and the National Wetlands Inventory (NWI). A road density layer was derived from TIGER road map using the `LINEDENSITY` function in ARC/INFO software. This function calculates the density of lines in a circular neighborhood around each pixel. We also generated an upland/lowland layer using the NWI dataset. The terrain is relatively flat, so a DEM was not used in this study, however, it may be an important input for separating upland and lowland classes in highly varying terrain conditions. From the Census dataset, we derived a population density layer based on the number of persons per square mile. Satellite imagery was geometrically corrected and geo-referenced to the UTM projection by collecting about 30 ground control points and fitting a second order polynomial equation with an RMS error less than 7.5 meters (i.e., about a quarter of a pixel accuracy). Color infrared aerial photographs and existing thematic maps, along with ground visits, were utilized in collecting training data. A normalized difference vegetation index (NDVI) was computed from the TM image which enhances the discrimination power between vegetation and non-vegetation and is used in decision tree training along with ancillary data sets.

Clustering: Using the C-Means clustering algorithm we obtained 20 spectral clusters. Small patches (with size < 5 pixels) were further eliminated using the "clump analysis" module in Imagine software. The cluster image was converted into a polygonal vector layer for further processing in the ARC/INFO system [64]. This process resulted in about 172564 polygons. Each of these polygons was classified into one of ten thematic classes (see Table 5.3) using the decision tree classifier.

Decision Tree Construction: Approximately 180 training samples were collected and for each sample the attributes from each of the geo-spatial data layers described above were extracted. C4.5 decision tree software was used to build the decision tree. For the hybrid system, the feature set consists of the clustered image and ancillary geo-spatial data sets. We trained the decision tree classifier on this feature set with 95% accuracy. The final pruned tree had 29 leaves with a maximum depth of 9. On the other hand, the direct training on a feature set consisting of the original TM image, NDVI image, and ancillary data had resulted in a tree of depth 16. These trees were then used to classify the respective data sets into ten thematic classes.

ML Classification: To train the MLC, we applied a region growing algorithm at each sample location to populate the polygons (maximum of 36 pixels) with homogeneous characteristics. Training samples were further purified based on transformed divergence analysis. Using these training polygons, training statistics (m_i, Σ_i) were generated. Ancillary data sets were used to compute *a priori* probabilities using the technique suggested in [51].

5.3.2 Analysis of Results

In this section we present a comparative analysis of our new approach with MLC and DT classifiers. The classified images are shown in Figure 5.5. Table 5.3 summarizes the classification accuracy of the MLC, DT, and hybrid classification system methods.

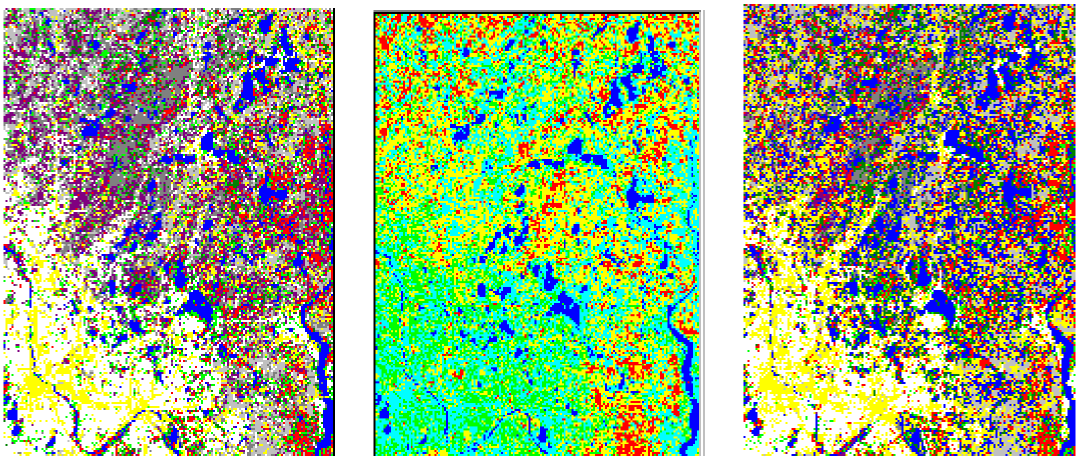


Figure 5.5: Classified images using MLC, C-Means, and Hybrid techniques.

The final supervised MLC classification yielded about 78% accuracy, the decision tree classification gave about 85% and the hybrid system gave about 88%. The performance gain over MLC can be attributed to the fact that the upland and lowland classes are highly overlapping and decision trees were learned to separate these classes using ancillary data sets. Similarly urban classification is also improved with the use of road density and urban density data sets. Performance of decision trees and hybrid system were comparable in terms of classification accuracies, however, hybrid system is computationally more attractive than decision trees because of the reduction in feature set size and number of points to be classified (due to initial clustering of the original TM image). Thus the hybrid system can be used to mine large data sets (for global and regional applications), where spectral overlapping is common and signature extension does not hold. Although direct decision tree classification yields better accuracy than MLC, the training requirements are substantially higher and the resulting trees are very big and thus demand more computational resources. For example, the study in [5] resulted in a tree with 105 terminal nodes (leaves) and a visual pruning resulted in 57 final nodes for global land cover classification at 8KM resolution. These requirements will further grow with the MODIS data set as the spatial resolution is improved to 250 meters.

Class	MLC (%)	DT (%)	Hybrid (%)
Water	87	90	90
Bare Soil	81	87	85
Crop (and Grass)	80	86	86
Upland Conifer	68	75	80
Upland Hardwood	80	85	89
Lowland Hardwood	78	84	85
Lowland Conifer	78	84	85
Wetland	70	76	82
Low density Urban	72	81	80
Hi density Urban	83	90	95
Overall Accuracy	78	85	88

Table 5.3: Comparison of Classification Accuracy (MLC, DT, and Hybrid Systems)

5.4 Hybrid Semi-supervised Learning Scheme

In this section we present our third approach. A common task in analyzing remote sensing imagery is supervised classification, where the objective is to construct a classifier based on few labeled training samples and then to assign a label (e.g., forest, water, urban) to each pixel (vector, whose elements are spectral measurements) in the entire image. The commonly used maximum likelihood classifier (MLC) has two well known limitations. First, it works well if the land cover classes are spectrally separable. In reality, the classes under investigation are often spectrally overlapping as the reflectance recorded by remote sensing satellites for many of these thematic classes is dependent on several extraneous factors like terrain, soil type, moisture content, acquisition time, atmospheric conditions, etc. The usefulness of ancillary data for improving classification accuracy is well known, but there is no convenient multivariate statistical tool for modeling this multi-source data (i.e., images and ancillary geo-spatial data together). Previous studies [34], [51] have focused on incorporating ancillary information into the MLC (typically via *a priori* term).

Second, MLC uses maximum likelihood estimation (MLE) technique for estimating class probability distribution parameters which requires large amounts of accurate training data. Collecting ground truth data for large number of samples is very difficult. Apart from time and cost considerations, in many emergency situations like forest fires, land slides, floods, it is impossible to collect accurate training samples. As a result, supervised learning is often carried out with small number of training samples, which leads to large variance in parameter estimates and thus higher classification error rates. Several approaches can be also be found in the literature that specifically deal with small sample size problems in supervised learning [12, 13, 16, 15, 11]. These methods are aimed at designing appropriate classifiers, feature selection, and parameter estimation so that classification error rates can be minimized while working with small sample sizes. However, only recently attempts have been made to incorporate unlabeled samples in supervised learning, which gave raise to new breed of techniques, collectively known as semi-supervised learning methods. Well-known studies in this area include, but not limited to [47, 43, 44, 45, 65]. The common thread between many of these methods is the Expectation Maximization (EM) [46] algorithm. Many of the semi-supervised learning methods pose class labels as the missing data and use the EM algorithm to improve initial (either guessed or estimated from small labeled samples) parameter estimates. Though previous studies [47, 66] showed that adding unlabeled training samples improves overall classification accuracy, little attention was given to extending semi-supervised learning for multisource data classification.

We now describe a new hybrid semi-supervised learning method based on a mixture of discrete and continuous distributions. In typical semi-supervised approach, the population is assumed to be generated by a mixture of multivariate normal distributions for continuous attributes (e.g., remote sensing images), or mixture of multinomial distributions for categorical attributes (e.g., text documents, ancillary geospatial data such as soil types, upland and lowlands).

5.4.1 Semi-supervised Learning for Multisource Data Classification

As explained previously, multisource data is a mixture of both continuous and discrete distributions. Let us now divide our attribute into two partitions: one consisting of all continuous variables and the other consisting of all discrete variables. We can now rewrite our mixture model given in Eq. 3.11 as following:

$$p(x_i|\Theta) = \sum_{j=1}^{M} \alpha_j \prod_{l=1}^{2} p_{jl}(x_{il}|\theta_{jl}) \tag{5.4}$$

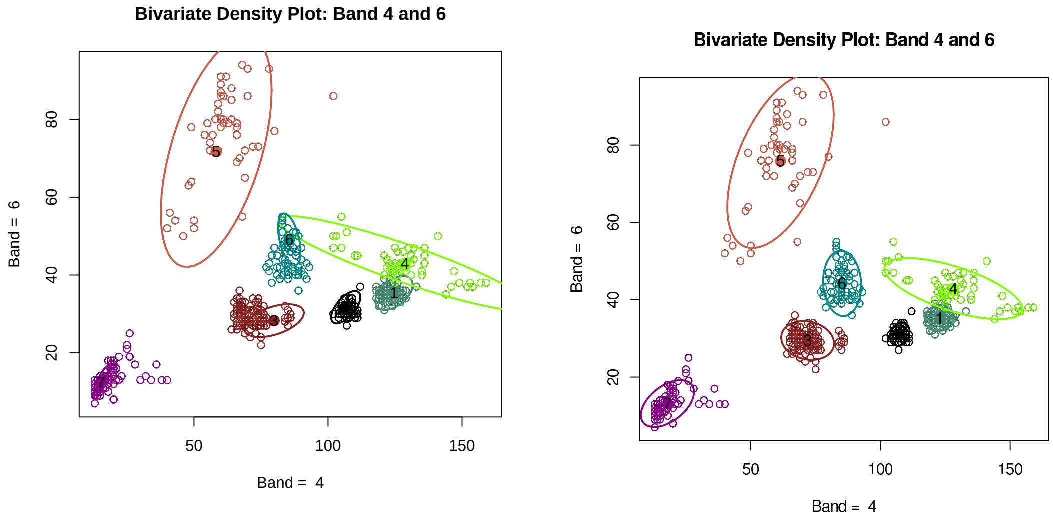
where θ_{jl} consists of the parameters of the distribution p_{jl} for the partition l. To reduce the complexity we used a knowledge based approach to stratify the geographic region into three broad categories, viz., uplands, lowlands and developed area. The main objective for this stratification is to split the geographic region into different spatial units where each spatial unit contains classes that are easily discriminable. So the discrete variables partition consists of a single attribute with three possible values. Finally we used expectation maximization algorithm to estimate the model parameters, where our model consists of six continuous attributes (corresponding to six channels in the ETM image) and a categorical attribute (generated using a knowledge based classification algorithm). We have conducted several experiments to evaluate the usefulness of our method in thematic classification of multisource geospatial datasets. We now briefly present the results in the following section.

5.5 Experimental Results

We used a spring Landsat 7 scene, taken on May 31, 2000 over Cloquet town located in Carlton County, Minnesota. We designed two different experiments to validate our hypothesis that adding ancillary geospatial datasets and unlabeled training samples improve the classification performance.

We have used the following ancillary information: normalized density vegetation index (NDVI) and Tasseled Cap (images), transportation data (lines), National Wetland Inventory (NWI) data (polygons) and population data (polygons/attributes). We used a knowledge based approach [67] to generate a stratified image consisting of upland, lowland, and developed regions. This stratified image is used as a categorical attribute in our multisource classification experiment.

The labeled training data consists of 14 plots (2 plots per class), and unlabeled training data consists of 50 plots. For both of these experiments the test dataset was fixed and consisted of 205 plots. We trained three classifiers: MLC, Semi-supervised



(a) ML

(b) SSL(EM)

Figure 5.6: Parameters Estimated using (a) ML, (b) SSL(EM).

C-ID	Class	MLC	SSL	SSL-MS
1	Hardwood.1	56.04	48.31	36.23
2	Hardwood.2	78.21	65.36	61.87
3	Conifer	43.30	80.84	86.59
4	Agriculture	84.19	94.02	98.72
5	Urban	100.00	91.11	97.78
6	Wetlands	6.50	28.38	66.50
7	Water	53.70	90.74	96.30
O	Overall	48.62	58.92	70.51

Figure 5.7: Class and Overall Accuracy

Classifier (SSL), and Multisource Semi-supervised classifier (SSL-MS). The estimates obtained by maximum likelihood and semi-supervised approaches (using expectation maximization) are summarized (in the form of bivariate density plots) in Figure 5.6. The individual class accuracy and overall classification accuracies were summarized in Figure 5.7. This figure (table) shows the great potential of our proposed classification scheme in small sample and multisource data classification problems. The plain semi-supervised learning method improved classification accuracy by 10% and on the other hand semi-supervsied learning scheme on multisource data has resulted in improvement of about 22% over maximum likelihood classification.

5.6 Conclusions

In this study we presented a semi-supervised learning scheme for multisource data classification. This new scheme addresses two major limitations of the most widely used maximum likelihood classifier: small training samples and multisource data. Finite mixture modeling offers great flexibility in modeling multisource data. Initial experimental results showed an improvement of more than 20% as compared to MLC with training data of just 2 plots for class. Processing ancillary data to come up with meaningful stratified units is still an open problem. Further research is needed to automatically discover the stratified units from ancillary data for a given classification task. More experiments are needed to see the performance of the proposed algorithm in different geographic settings.

Chapter 6

Aggregate Classes

6.1 Introduction

Remote sensing, which provides inexpensive, synoptic-scale data with multi-temporal coverage, has proven to be very useful in land cover mapping, environmental monitoring, forest and crop inventory, urban studies, natural and man made object recognition, etc. Thematic information extracted from remote sensing imagery is also useful in a variety spatio-temporal applications. For example, land management organizations and the public have a need for more current regional land cover information to manage resources and monitor land use changes. Likewise, intelligence agencies, such as, National Geospatial Intelligence Agency (NGA), and Department of Homeland Security (DHS), utilizes pattern recognition and data mining techniques to classify both natural and man made objects from large volumes of high resolution imagery.

Image classification, i.e., assigning class labels to pixels, using some discriminant function, is one of the fundamental analysis technique used in remote sensing to generate thematic information. Image classification can be formally defined as finding a function $g(x)$ which maps the input patterns x onto output classes y_i (some times y_i's are also denoted as ω_i or c_i). The main objective is to assign a label (e.g. Water, Forest, Urban) to each pixel in the image to be classified, given corresponding feature vector x_j's in the input image. Depending on the type of supervised learning method used, the objective of a supervised learning could be finding a function $g(x)$ (also called a discriminant function), that divides the input $d-$dimensional feature space into several regions, where each region corresponds to a thematic class y. One such simple function is given in 2.1.

That is, the feature vector x belongs to class y_i if $p(y_i|x)$ is the largest. Even though it sounds simple, this assignment problem is very difficult. There is no single algorithm which will correctly classify any given image. Multi-spectral image classification is still an open problem.

One of the main problem in image classification stems from the fact that spectral classes are often overlapping and may not directly correspond to information classes. One of the main reasons for this mismatch between spectral classes (image supported) and information classes (analyst given) is due to the fact that it impossible to collect labels for all spectrally separable classes. For accurate estimation of parameters of the statistical model, one approximately needs $(10 - 30) \times d$ labeled samples per class [68]. Collecting ground truth (labels) data over large geographic regions is costly, time consuming, and poses several other practical problems. It is also estimated that the cost of collecting a single training sample in remote sensing is approximately \$500. Given these practical limitations, often the analyst groups the relevant classes and collects labels only for those aggregate (grouped) classes. Typical examples are forest, agriculture, urban, and other land-use classes. Usually, the analyst given forest class may contains samples from all types of forests, such as hardwoods, conifer, etc., each of which can be described by a unique statistical distribution. These distributions are clearly identifiable in many image classification problems (depending on the spectral resolution), though in some cases these finer classes may be highly overlapping (in low spectral resolution images).

Aggregate classes poses two problems, first it violates the common assumption that each class is unimodal, secondly the estimated parameters could be wrong. Let us now relax this assumption and revise our problem definition as following.

6.1.1 Aggregate Class Classification: Problem Formulation

Given:

A spatial framework $\mathcal{S}$ consisting of sites s_{ij} ordered as an $l \times p$ matrix, where l is the number of lines and p is the number of pixels, and $\{1 \leq i \leq l, 1 \leq j \leq p\}$.

A $d-$dimensional feature vector at each site s_{ij}, is represented by x_{ij}, where x_{ij} is a continuous random variable.

A set of discrete labels $Y = \{y_1, \ldots, y_k\}$, where k is the total number of distinct labels

Training Dataset $D = D_l \cup D_{ul}$, where D_l contains labeled training plots and D_{ul} contains unlabeled training plots.

A parametric model (e.g., Gaussian).

Find:

Estimate parameter vector Θ (e.g., $\{\mu_i, \Sigma_i\}$).

Objective:

Maximize complete data *log-likelihood* $L(\Theta)$.

Assumptions:

A1 The size of the labeled training dataset is less than 10 to 30 times the number of dimensions

A2 Thematic classes are separable

A3 Classes are not unimodal

A4 Feature vectors are independent and identically distributed (i.i.d), but features are highly correlated in feature space (that is features are not independent).

A5 D_l and D_u samples are generated by GMM_l and GMM_u.

If we ignore assumption (A3), then we are combining distinctly identifiable classes. Therefore, the estimated covariance matrix is large, as it accounts for both inter-class covariance and intra-class covariances (of component classes). It is very important to estimate covariance matrix accurately because even for a fixed means, it can be shown that increase in variance of any one class (keeping means fixed), leads to the increase in probability of error. Figure 6.1 shows what happens when the individual spectral classes are combined into one aggregate class. Sample image with two plots collected over an agriculture land is shown in Figure 6.1(a). Though both plots are collected over agriculture land, we can see two distinct histograms in Figure 6.1(b) each corresponding to finer classes (different crops in this case). Bivariate Normal density plots corresponding to the two sub-classes are shown in Figure 6.1(d). This plots shows that these final classes are clearly identifiable and separable in the future space. On the other hand, Figure 6.1(e) shows bivariate Normal density plot of the *agriculture* class (aggregated), which clearly shows what the problem might be. The increase in the size of ellipsoid implies that the *agriculture* class now overlaps with other classes in the feature space, and this increase in overlap between classes leads to increase in the probability of error, as shown in Figure 6.2.

Figure 6.2 shows the relationship between the overlap and the probability of error. Here, E_{12} denotes the error region where feature vectors x_i are classified as class 1 (c_1) where as x_i actually belongs to the class 2 (c_2). Figure 6.2(b) shows the increased probability of error due increase in variance of c_1. Though, the mean is same in both cases, the increase in variance of one class (c_1) has resulted in increase of probability of error, $p_E = E_{12} + E_{21}$. This observation motivated us to develop a new classification scheme which relaxes the common assumption that the class has to be a unimodal distribution. Instead, we assume that each class is a finite mixture model.

6.1.2 Contributions

In this work we clearly identified an important practical problem in supervised classification of remotely sensed imagery. We developed a novel learning algorithm which takes user defined aggregate classes and automatically discovers sub-classes within each aggregate class. The resulting classifier showed not only improvement in overall classification accuracy but also recognized finer classes which analyst is always interested in, but sufficient ground truth data cannot be collected for the finer classes. Then we presented a expectation maximization based semi-supervised learning scheme with new update equations for classifying the newly discovered sub-classes.

6.2 Statistical Classification Framework

In the classification of a remote sensing images, our objective is to assign a class label (y) to each pixel (x) based on certain decision criterion. Maximum likelihood classification (MLC) and maximum a posteriori (MAP) classification are two of the most widely used classifiers in remote sensing. Bayesian decision theory plays a central role in statistical pattern classification. Both ML and MAP classifiers are based on Bayesian decision theory.

6.2.1 Bayesian Classification

In the Bayesian approach, the objective is to find the most probable set of class labels given the data (feature) vector and *a priori* probabilities for each class. Formally, we can state Bayes' formula as:

$$P(y_i|x) = \frac{p(x|y_i)P(y_i)}{p(x)} \tag{6.1}$$

The unknown conditional probabilities $p(y_i|x)$ can be estimated if sufficient training samples for each of the classes are available. Assuming the training samples were generated by a multivariate normal or Gaussian density, we can write the decision rule for maximum a posterior (MAP) classifier as following:

$$g_i(x) \;=\; \ln\; P(y_i) - \frac{1}{2}\ln|\Sigma_i| - \frac{-1}{2}(x-\mu_i)^t|\Sigma_i|^{-1}(x-\mu_i) \tag{6.2}$$

If we don't have *a priori* knowledge about the classes then we can drop $P(y_i)$ term in the above equation and the resulting decision rule is known as maximum likelihood classification (MLC). The covariance matrix Σ plays a key role in discriminant analysis. Covariance accounts for the shape (size and orientation) of classes in the feature space. The effectiveness of ML/MAP classification depends on the quality of the estimated parameter vector Θ (i.e., mean vector μ and the covariance matrix Σ for each class) from the training samples. Using a well-known parameter estimation technique, maximum likelihood estimation (MLE), we can obtain the parameters μ and Σ as following:

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k; \quad \hat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}(x_k-\hat{\mu})(x_k-\hat{\mu})^t. \tag{6.3}$$

6.2.2 Limitations of MLC and MAP

One of the classical assumptions in supervised (statistical) classification is that the classes are unimodal. We now test the impact of violation of this constraint through a simulated example. We generated bivariate Normal samples (150 per class) using the parameters given in the Table 6.3 and 6.1. Maximum likelihood estimates from the simulated samples were also summarized along with the original parameters.

	C_1		C_2		C_3	
Features	X1	X2	X1	X2	X1	X2
X_1	30.00	25.00	60.00	40.00	60.00	50.00
X_2	25.00	40.00	40.00	90.00	50.00	70.00
Estimated Parameters (MLE)						
X_1	27.03	19.10	57.66	43.43	56.81	40.60
X_2	19.10	31.58	43.43	100.03	40.60	55.84

Table 6.1: Simulation Parameters (Covariance)

C_{23}		X1	X2
Mean		64.66	44.89
Covariance	X1	280.49	121.80
	X2	121.80	106.26

Table 6.2: ML Estimates of Aggregated Class ($C_{23} = C_2 + C_3$)

	Given		No of	Estimated (MLE)	
Classes	X1	X2	Samples	X1	X2
C_1	55.00	25.00	150.00	54.93	24.50
C_2	80.00	50.00	150.00	79.58	50.23
C_3	50.00	40.00	150.00	49.74	39.55

Table 6.3: Simulation Parameters (Means)

Let us now assume that classes 2 and 3 are sub-classes of an aggregate class C_{23}, i.e., analyst gave a single label to all the samples generated from the classes C_2 and C_3. The new estimates of the aggregate class C_3 are given in the Table 6.2. For understanding the distribution (and interaction) of original classes (C_1, C_2, C_3) and aggregate classes (C_1, C_{23}), we have generated the bivariate density plots which are shown in Figure 6.3.

The overlap between classes (C_1, C_2, C_3) is almost negligible (see Figure 6.3(a)). However, aggregation of classes C_2, C_3 into C_{23} has greatly increased its overlap with class C_1 (see Figure 6.3(b)). We have seen previously that this overlap directly accounts for the probability of error, p_E. One can expect to improve the classification accuracy, if somehow the original classes (C_2, C_3) which gave raise to aggregate class C_{23} can be automatically discovered. Also, there is a great need for finer class discovery from remotely sensed images, and precisely this is what our proposed algorithm tries to accomplish.

6.3 Learning To Discover Sub-classes

Basic idea behind the proposed algorithm is very simple. Instead of assuming that each class is a unimodal multivariate Gaussian, we assume that the samples from each class are generated by finite Gaussian mixture. There are two subproblems associated with this assumption. First, we don't know how many components (sub-classes) are there in this finite mixture model. Second, we don't have labels for any of the component (sub-class) so that we can employ regular MLE technique to estimate the parameters of each component. Finally, we need to identify (classify) these sub-classes with minimal additional training efforts. Solution to each of these subproblems were given in the following subsections.

6.3.1 Estimating Finite Mixture Parameters

Let us now solve the first problem by assuming that we know the number of components but not the labels for any the component class. Now assume that the training dataset D_j is generated by a finite Gaussian mixture model consisting of M components (as opposed to unimodal Gaussian per class). If the labels for each of these components were known, then problem simply reduces to usual parameter estimation problem and we could have used MLE. Since the labels for sub-classes were not known, we have to reformulate the likelihood estimation for the finite mixture model. We now describe a parameter estimation technique that is based on expectation maximization algorithm. Let us assume that each sample x_j comes from a super-population D, which is a mixture of a finite number (M) of sub-classes, $D_1, \ldots, D_M$, in some proportions $\alpha_1, \ldots, \alpha_M$, respectively, where $\sum_{i=1}^{M} \alpha_i = 1$ and $\alpha_i \geq 0 (i = 1, \ldots, M)$. Compared to our discussion in the subsection 6.2.1, we can think of α_i as $P(y_i)$. Now we can model the data $D = \{x_i\}_{i=1}^{n}$ as being generated independently from the following mixture density.

$$p(x_i|\Theta) = \sum_{j=1}^{M} \alpha_j p_j(x_i|\theta_j) \tag{6.4}$$

$$L(\Theta) = \sum_{i=1}^{n} \ln \left[\sum_{j=1}^{M} \alpha_j p_j(x_i|\theta_j) \right] . \tag{6.5}$$

Here $p_j(x_i|\theta_j)$ is the pdf corresponding to the mixture j and parameterized by θ_j, and $\Theta = (\alpha_1, \ldots, \alpha_M, \theta_1, \ldots, \theta_M)$ denotes all unknown parameters associated with the M-component mixture density. The *log-likelihood* function for this mixture density is given in 6.5. In general, Equation 6.5 is difficult to optimize because it contains the ln of a sum term. However, this equation greatly simplifies in the presence of unobserved (or incomplete) samples. We now simply proceed to the expectation maximization algorithm, interested reader can find detailed derivation of parameters for GMM in [48]. The expectation maximization (EM) algorithm at the first step maximizes the expectation of the *log-likelihood* function, using the current estimate of the parameters and conditioned upon the observed samples. In the second step of the EM algorithm, called maximization, the new estimates of the parameters are computed. The EM algorithm iterates over these two steps until the convergence is reached. For multivariate normal distribution, the expectation $E[.]$, which is denoted by p_{ij}, is the probability that Gaussian mixture j generated the data point i, and is given by:

$$p_{ij} = \frac{\left|\hat{\Sigma}_j\right|^{-1/2} e^{\left\{-\frac{1}{2}(x_i-\hat{\mu}_j)^t \hat{\Sigma}_j^{-1} (x_i-\hat{\mu}_j)\right\}}}{\sum_{l=1}^{M} \left|\hat{\Sigma}_l\right|^{-1/2} e^{\left\{-\frac{1}{2}(x_i-\hat{\mu}_l)^t \hat{\Sigma}_l^{-1} (x_i-\hat{\mu}_l)\right\}}} \tag{6.6}$$

The new estimates (at the k^{th} iteration) of parameters in terms of the old parameters at the M-step are given by the following equations:

$$\hat{\alpha}_j^k = \frac{1}{n} \sum_{i=1}^{n} p_{ij} \tag{6.7}$$

$$\hat{\mu}_j^k = \frac{\sum_{i=1}^{n} x_i p_{ij}}{\sum_{i=1}^{n} p_{ij}} \tag{6.8}$$

$$\hat{\Sigma}_j^k = \frac{\sum_{i=1}^{n} p_{ij}(x_i - \hat{\mu}_j^k)(x_i - \hat{\mu}_j^k)^t}{\sum_{i=1}^{n} p_{ij}} \tag{6.9}$$

We can now put together these individual pieces into the following algorithm (Table 6.4) which computes the parameters for each component in the finite Gaussian mixture model that generated our training data D (without any labels).

Inputs: D_j, training dataset (no labels for sub-classes) for any aggregate class y_j; M, the number of sub-classes in the corresponding aggregate class.

Initial Estimates: Do clustering by K-Means, and estimate initial parameter using MLE, to find $\hat{\theta}$ (see Equations 6.3)

Loop: While the complete data *log-likelihood* improves:

> **E-step:** Use current classifier to estimate the class membership of each unlabeled sample, i.e., the probability that each Gaussian mixture component generated the given sample point, p_{ij} (see Equation 6.6).
>
> **M-step:** Re-estimate the parameter, $\hat{\theta}$, given the estimated Gaussian mixture component membership of each unlabeled sample (see Equations 6.7, 6.8, 6.9)

Output: Parameter vector Θ.

Table 6.4: Algorithm for Computing Parameter of Finite Gaussian Mixture Model Over Unlabeled Training Data

6.3.2 Estimating the Number of Components

We now address our second problem, i.e., we don't know how many components (sub-classes) are there in each (aggregate) class. As with estimating model parameter for finite Gaussian mixture model, we assume that the training dataset D is generated by a finite Gaussian mixture model, but we don't know either the number of components or the labels for any of the sub-component. In previous section, we devised an algorithm to find parameters by assuming a M-component finite Gaussian mixture model. In general, we can estimate parameters for any arbitrary M-component model, as long as there are sufficient number of samples available for each component and the covariance matrix does not become singular. Then the question remains, which M-component model is better? This question is addressed in the area of model selection, where the objective is to chose a model that maximizes a cost function. There are several cost functions available in the literature, most commonly used measures are Akaike's information criterion (AIC), Bayesian information criteria (BIC), and minimum description length (MDL). The common criteria behind these models is to penalize the models with additional parameters, so BIC and AIC based model selection criteria follows the principal of parsimony. In this study we considered BIC as a model selection criteria, which is also takes the same form as MDL. We also chose BIC, as it is defined in terms of maximized log-likelihood which any way we are computing in our parameter estimation procedure defined in the previous section. BIC can be defined as

$$BIC = MDL = -2 \log L(\Theta) + m \log(N) \tag{6.10}$$

where N is the number of samples and m is the number of parameters. We now describe our BIC based model selection criteria to determine the number of number components in each aggregate class (see Table 6.5).

Inputs: Samples from any aggregate class (D_l), Maximum number of Sub-classes (y_l^{max})

Parameter Estimates: Apply parameter estimation algorithm (Table 6.4) recursively by posing D_l as finite Gaussian mixture model with number of components varying from 2 to y_l^{max}

Compute BIC: Compute BIC (eq. 6.10) for each of the model (using parameters computed in the previous step)

Output: Select the model (parameter vector Θ), for which BIC is maximum

Table 6.5: Algorithm for Computing Parameters Including the Number of Components

Repeat the algorithm for each (aggregate) class in the original classification problem. At the end of each iteration we have parameters for each sub-class within an aggregate class. We can now apply MLC/MAP in two ways. First, we modified MLC/MAP to output both aggregate classes (original analyst given classes) and as well sub-classes which were discovered automatically using the procedure just described. We can combine the finer classes (predicted) into the corresponding aggregate class in order to find the aggregate class classification accuracy.

6.3.3 Semi-supervised Learning (SSL)

One natural question that needs to be answered in order for the proposed method to be useful in real world applications is, 'What these sub-classes are?' This question is easy if we have sufficient ground truth about these sub-classes. However, we developed this entire classification framework based on the premise that it is very difficult to obtain sufficient ground truth information for each identifiable class. If good number of ground truth data is not available, is it possible to identify these sub-classes if small number of ground truth information is available? This is where semi-supervised approaches are useful. Though we may not have sufficient ground truth available, we do have large number of unlabeled samples (pixels - feature vectors extracted from the image). In semi-supervised learning, we can exploit these unlabeled training samples to improve the parameter obtained from the small number of labeled training samples. The parameter estimation is similar to the algorithm described in Section 6.3.1. However, adding unlabeled training samples may not necessarily improve the classification performance [50]. However, our previous work showed that semi-supervised learning can produce better results with careful selection of unlabeled training samples and proper weighting of labeled and unlabeled training samples [66]. In typical semi-supervised learning small number of labeled samples were used to obtain initial estimates, and unlabeled training samples were used to iteratively improve these initial estimates using the update equations given in Section 6.3.1 (Eq. 6.7- 6.9 and algorithm 6.4. We now provide a new set of update equations using which one can emphasize (or deemphasize) the importance of unlabeled samples in the semi-supervised learning. The new update equations are given by:

$$\hat{\alpha}_j^k = \frac{(\lambda_l m_j + \sum_{i=1}^{n} \lambda_{ul} p_{ij})}{(\lambda_l m + \lambda_{ul} n)} \tag{6.11}$$

$$\hat{\mu}_j^k = \frac{(\sum_{i=1}^{m_j} \lambda_l y_{ij} + \sum_{i=1}^{n} \lambda_{ul} x_i p_{ij})}{(\lambda_l m_j + \sum_{i=1}^{n} \lambda_{ul} p_{ij})} \tag{6.12}$$

$$\hat{\Sigma}_j^k = \frac{\left\{ \begin{array}{c} \sum_{i=1}^{m_j} \lambda_l (y_{ij} - \hat{\mu}_j^k)(y_{ij} - \hat{\mu}_j^k)^t + \\ \sum_{i=1}^{n} p_{ij} \lambda_{ul} (x_i - \hat{\mu}_j^k)(x_i - \hat{\mu}_j^k)^t \end{array} \right\}}{(\lambda_l m_i + \sum_{i=1}^{n} \lambda_{ul} p_{ij})} \tag{6.13}$$

We made simple modifications to the algorithm 6.4 to use these new update equations. Using this semi-supervised approach we can identify the sub-classes with minimal additional training requirements. We now present the experimental results showing the usefulness of the proposed approach.

6.4 Experimental Results

We have conducted several experiments using simulated and as well as the real dataset. Classification accuracy results were summarized in the form of contingency table. Accuracy measures included are producers accuracy (P.Acc), users accuracy (U.Acc) and overall accuracy (O.A), whose definitions can be found in citecong-91.

Dataset 1: The objective of first experiment on simulated data is to see performance of proposed method in aggregate class classification and as well as finer class classification. We used the parameters listed in Table 6.3 to generate two distinct datasets. First dataset consisted of 150 samples at 50 samples per class, and second dataset consisted of 450 samples at 150 samples per class. We used first dataset for training and the second dataset for testing. We conducted following three experiments.

G.Truth	C_1	C_2	C_3	P.Acc
C_1	141.00	4.00	5.00	94.00
C_2	1.00	147.00	2.00	98.00
C_3	1.00	1.00	148.00	98.67
U.Acc	98.60	96.71	95.48	(OA) 96.89

Table 6.6: Accuracy (All Classes)

G.Truth	C_1	C_3	P.Acc
C_1	281.00	19.00	93.67
C_3	2.00	148.00	98.67
U.Acc	99.29	88.62	(OA) 95.33

Table 6.7: Accuracy ($C_1, C_2 \rightarrow C_1$)

G.Truth	C_1	C_2	C_3	P.Acc
C_1	132.00	11.00	7.00	88.00
C_2	1.00	147.00	2.00	98.00
C_3	1.00	1.00	148.00	98.67
U.Acc	98.51	92.45	94.27	(OA) 94.89

Table 6.8: Accuracy ($C_1 \rightarrow C_1, C_2$)

G.Truth	C_1	C_3	P.Acc
C_1	291.00	9.00	97.00
C_3	2.00	148.00	98.67
U.Acc	99.32	94.27	(OA) 97.56

Table 6.9: Accuracy ($C_1 \rightarrow C_1, C_2 \rightarrow C_1$)

Experiment 1: MAP classification was carried out using all three classes, whose distribution is shown in Figure 6.4(a). Test accuracy of MAP is shown in Table 6.6.

Experiment 2: Classes C_1, C_2 were combined into one aggregate class and class C_3 remained untouched. Resulting new class distributions were shown in Figure 6.4(b). Test accuracy of MAP using aggregated class is shown in Table 6.7.

Experiment 3: Our new algorithm was applied on the dataset generated in Experiment 2. We tested both aggregate classification performance and as well as the finer class performance. Newly discovered classes were shown in Figure 6.4(c). Test accuracy of MAP using newly discovered classes is shown in Table 6.8, and the corresponding aggregated class accuracy is shown in Table 6.9.

Dataset 2: For the next set of experiments we used a spring Landsat 7 scene, taken on May 31, 2000 over the Cloquet town located in Carlton County of Minnesota state. The training dataset consisted of sixty plots and four aggregate classes, namely, Forest(1), Agriculture(2), Urban(3), and Wetlands(4). We have an independent test dataset consisting of 205 plots. Feature vectors were extracted from the Landsat image (6-dimensional) by placing a 3×3 window at each of these plots.

Experiment 4: Maximum likelihood classification is carried out using the conventional approach and as well as the proposed approach using Datset 2. The results were summarized in the following contingency tables (or error matrices).

Table 6.10 gives MLC accuracy using the standard approach and Table 6.11 provides the classification accuracy obtained by our proposed method. We regrouped the sub-classes into the corresponding aggregate classes for testing the accuracy using same test dataset (consisting of four aggregate classes).

Experiment 5: For this experiment, we collected labels at the rate of two plots per sub-class. That is, of all the training plots available in an aggregate class we randomly labeled only two plots per sub-class. For collecting unlabeled training samples, we restricted random sampling to only those areas (pixels) where the aggregate model predicted class labels with high confidence (probability grater than twice 1/number of classes). We used same test dataset, but all the test samples in aggregate class were relabeled with sub-class labels. Table 6.12 shows sub-class classification accuracy for 'forest' class using only labeled training data and Table 6.13 shows accuracy for the same 'forest' sub-classes but with using unlabeled training samples and semi-supervised

G. Truth	1	2	3	4	P. Acc.
Forest(1)	1475.00	9.00	28.00	0.00	97.55
Agriculture(2)	90.00	142.00	2.00	0.00	60.68
Urban(3)	0.00	0.00	45.00	0.00	100.00
Wetlands(4)	18.00	0.00	2.00	34.00	62.96
Users Acc.	93.18	94.04	58.44	100.00	91.92

Table 6.10: Accuracy (Aggregate Classes)

GT	1	2	3	4	P. Acc.
Forest(1)	1448.00	13.00	51.00	0.00	95.77
Ag.(2)	14.00	214.00	6.00	0.00	91.45
Urban(3)	0.00	0.00	45.00	0.00	100.00
Wet.(4)	3.00	0.00	13.00	38.00	70.37
U. Acc.	98.84	94.27	39.13	100.00	94.58

Table 6.11: Accuracy (Each aggregate class is modeled as a GMM)

classification algorithm.

G. Truth	HW	CF	L/Wet	P.Acc
HW	662.00	3.00	1.00	99.40
CF	20.00	201.00	40.00	77.01
L/Wet	56.00	19.00	321.00	81.06
U.Acc	89.70	90.13	88.67	(OA) 89.49

Table 6.12: MLC Accuracy (Forest Sub-classes: HW - hardwood, CF - Conifer, and L/Wet - lowland/wetland forests)

6.4.1 Analysis

Accuracy assessment on simulated data shows interesting results. Bivariate density plot shown in Figure 6.4(a) and as well as hight test accuracy (Table 6.6) shows that the three classes were clearly separable. Aggregation of classes $C_1, C_2 \rightarrow C_1$ has increased the overlap between the aggregate class C_1 and C_3 (see Figure 6.4(b)), and this overlap has resulted in more classification error (Table 6.7) as compared to finer class classification error. Our proposed algorithm on this aggregate data has discovered two sub-classes in the aggregate class C_1 (see Figure 6.4(c)) and the corresponding BIC value (for number of clusters = 2) is maximum (Figure 6.4(d)). The test accuracy of MAP classifier trained

GT	HW	CF	L/Wet	P.Acc
HW	642.00	14.00	10.00	96.40
CF	9.00	245.00	7.00	93.87
L/Wet	11.00	2.00	383.00	96.72
U.Acc	96.98	93.87	95.75	(OA) 95.99

Table 6.13: SSL Accuracy (Forest Sub-classes)

on newly discovered classes is given in the Table 6.8 and the corresponding aggregated class accuracy is shown in Table 6.9. First, comparison of this accuracy table with original classification accuracy (Table 6.6) reveals that the sub-classes discovered closely corresponds to the original classes. Second, the aggregate (i.e., predicted sub-classes were merged) classification accuracy with our new scheme is higher than the MAP on original aggregate class classification (compare with Table 6.7). This study revels that our new algorithm not only discovered sub-classes that are close to the original fine classes (without providing any labeled training data) but also improved classification accuracy of original aggregate classes.

Let us now compare the classification error matrices (Table 6.10 and Table 6.11) obtained on the remote sensing classification dataset. From these two tables, we can see that our new procedure improved overall classification accuracy (OA) for the same training dataset without any additional (sub-class related training) information. In addition the new procedure automatically discovered sub-classes within each aggregate class. In this (aggregate class) training dataset, our new procedure discovered four additional component in the forest class, two additional components in the agriculture class, and two additional components in the urban class. Figure 6.5 shows BIC values and corresponding bivariate density plots for four (max BIC) forest sub-classes.

Our preliminary investigation into the newly discovered sub-classes reveled very interesting information. The four sub-classes discovered in aggregate forest class roughly corresponds to the following information classes: hardwoods (HW), conifer (CF) and two lowland/wetland forest classes (L/Wet). We used semi-supervised learning (SSL) to classify these sub-classes (Experiment 5). We assigned two plots for each of the sub-classes (HW and CF) and three plots for combined lowland/wetland forests. Table 6.12

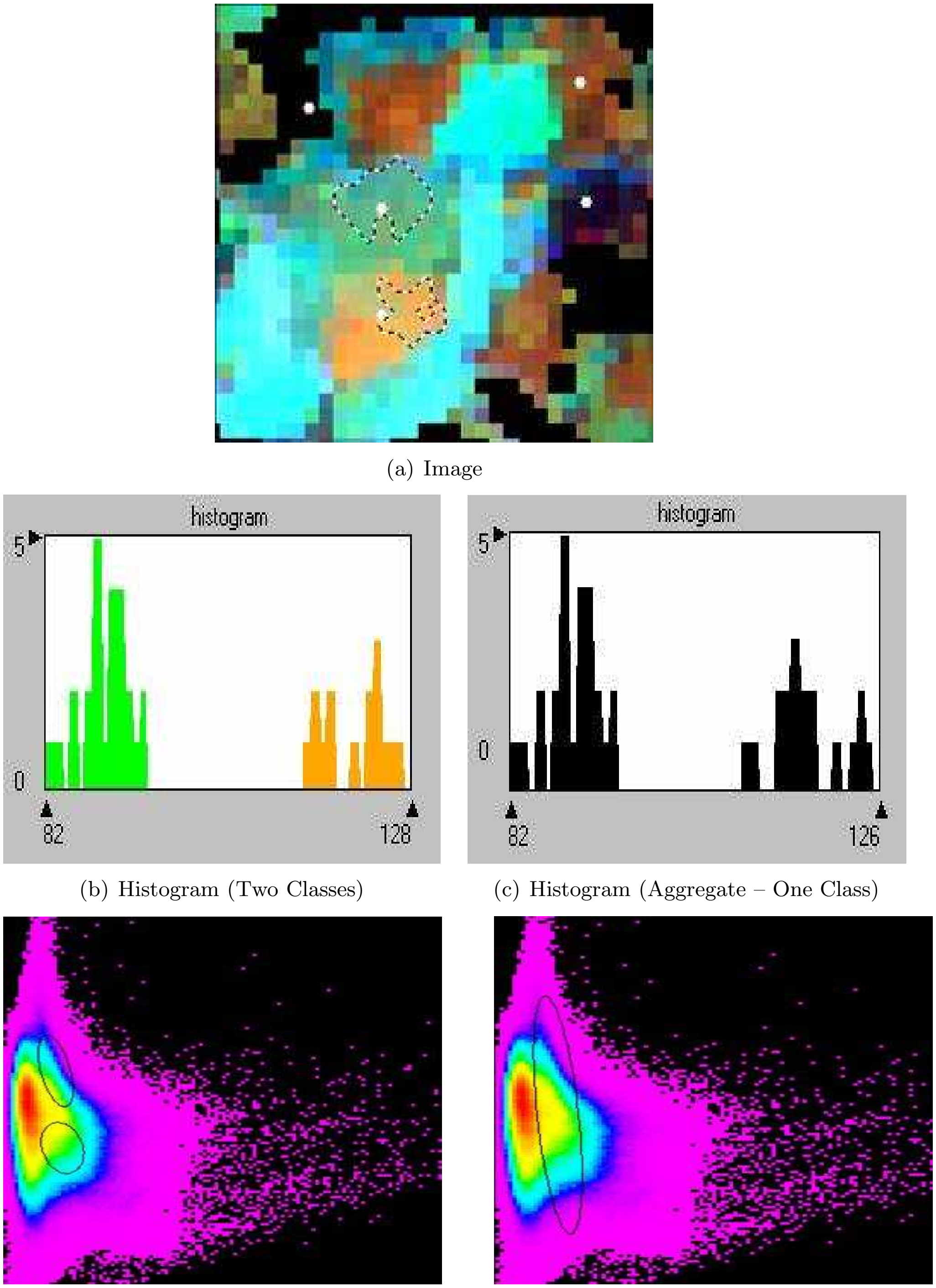
and 6.13 show the maximum likelihood (MLC) and semi-supervised (SSL) classification accuracies respectively. As can be seen from these two tables, it is evident that using very few labeled samples (2 plots/sub-class) and semi-supervised learning, it is possible to accurately identify fairly large number of classes from remote sensing images as opposed to few aggregate classes. Semi-supervised learning improved overall classification accuracy by almost 6.5%.

6.5 Related Work

Supervised methods are extensively used in remote sensing imagery classification [69, 56]. Finite mixture model parameter estimation [70] for unlabeled samples is also investigated extensively under various disguises. For example, semi-supervised learning approaches are very close to the approach we presented in Section 6.3. Well-known studies in this area include, but not limited to [43, 44, 45, 65, 47]. Model selection approaches have also been extensively studied [71, 72] and used in finding the number of clusters [73]. However, most of the classification approaches are based on the basic assumption that each class is described by a unique (unimodal) distribution. Even when mixture models were employed, basic assumption is that each class is described a single component in the mixture. We relaxed this basic assumption by modeling each class a mixture and automatically estimated the mixture components and parameters. We also showed that mixture assumption leads better discrimination of classes even for aggregate class classification (for same training dataset) as inter-class (probability distribution) overlap is reduced.

6.6 Conclusions

We identified an important practical classification problem that requires knowledge discovery approaches for automatically discovering the sub-classes from the aggregate classes. We developed a new classification scheme that automatically discovers the sub-classes from the user given aggregate classes, without any additional labeled training data for sub-classes. In addition, the procedure showed improvement in the classification of aggregate classes as well (for same training dataset). This improvement can be attributed to the fact that the aggregate classes tend to increase the overlap between class distributions. Our preliminary investigation also showed a strong correspondence between sub-classes and true information classes. As can be seen from the experiments, it is also possible to accurately identify fairly large number of classes from remote sensing images with little additional ground truth efforts (as little as 2 plots/sub-class) and semi-supervised classification approach presented in this paper. There is a great demand for such additional information in many real world applications. Further experimentation is needed to test the robustness of the proposed method. We are also investigating the semantic relationships between various information classes that are common in this domain which might help to automatically label these sub-classes.



(a) Image

(b) Histogram (Two Classes)

(c) Histogram (Aggregate – One Class)

(d) Density (Fine – Two Classes)

(e) Density (Aggregate – One Class)

Figure 6.1: Finer Classes vs. Aggregate Classes (a) Sample Image (training plots for two different crops), (b-c) Histograms (Fine vs. Aggregate) and (d-e) Corresponding Bivariate Density plots.

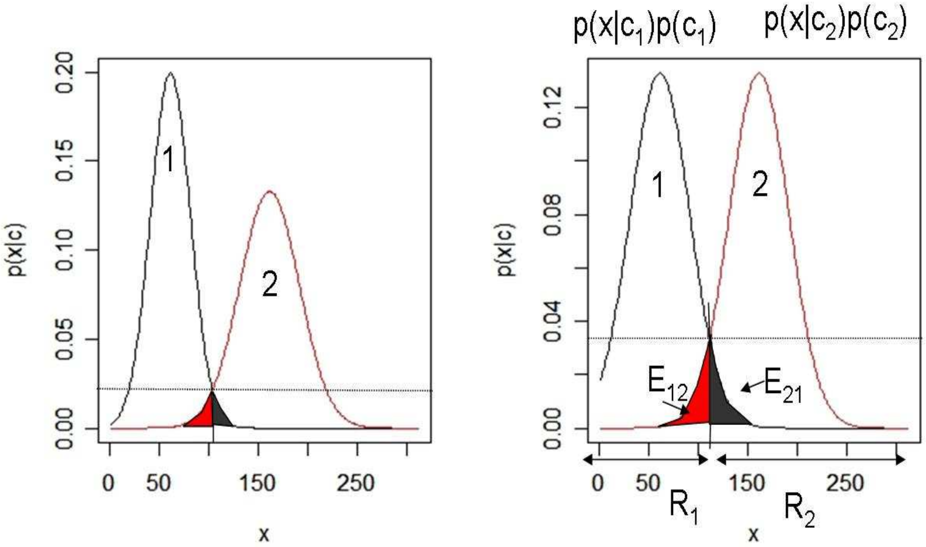
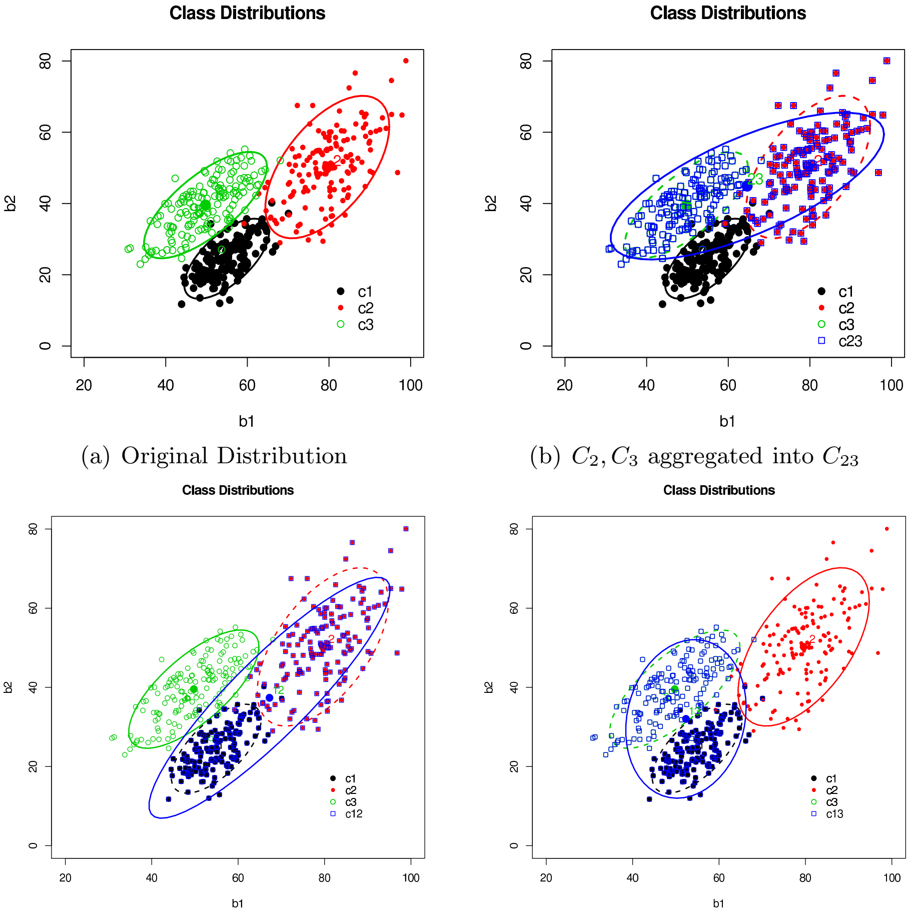


Figure 6.2: Relationship between class overlap and probability of error



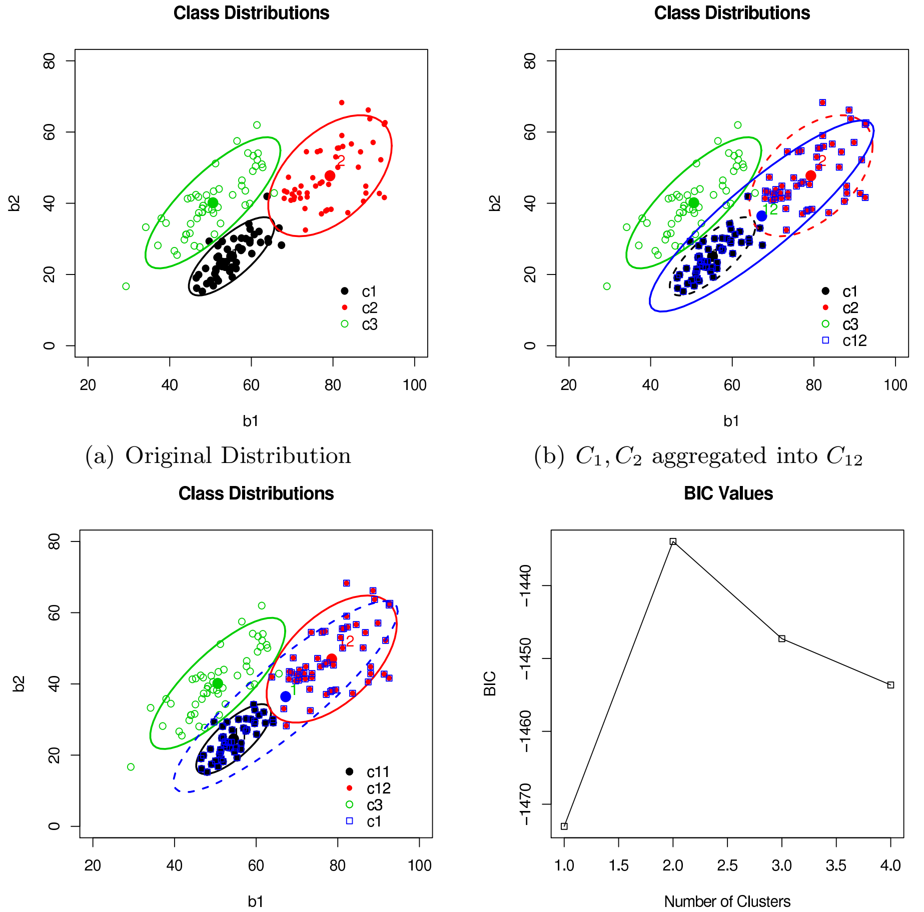
(a) Original Distribution

(b) C_2, C_3 aggregated into C_{23}

(c) C_1, C_2 aggregated into C_{12}

(d) C_1, C_3 aggregated into C_{13}

Figure 6.3: Interaction Between Finer and Aggregate Classes



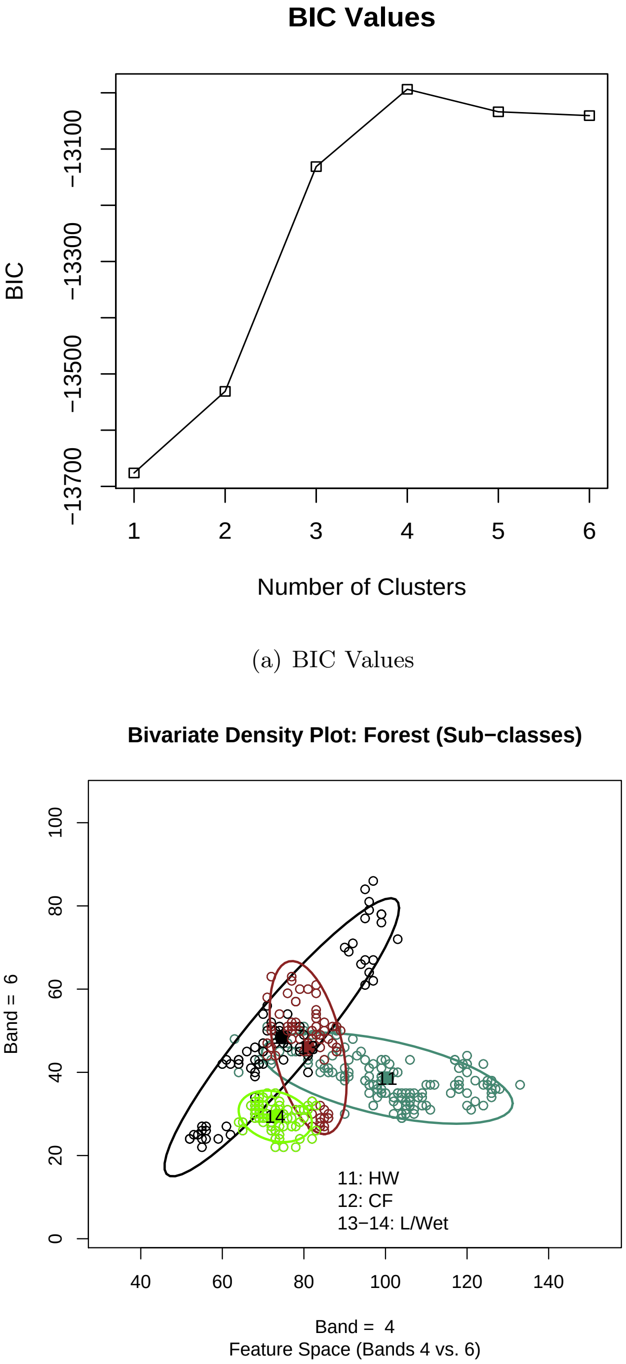
(a) Original Distribution

(b) C_1, C_2 aggregated into C_{12}

(c) Sub-classes C_{11}, C_{12} discovered from C_{12}

(d) BIC Values For Different Number of Clusters

Figure 6.4: Interaction Between Finer, Aggregate, and Newly Discovered Classes



(a) BIC Values

(b) Bivariate Density Plots

Figure 6.5: Forest Sub-classes

Chapter 7

Interpolation

7.1 Introduction

Remote Sensing and in situ data collection are two broad methods of acquiring earth science data. Remote sensing provides continuous data of large geographic regions. Multispectral, multitemporal, multistage, and multisensor digital image data obtained from remote sensing has proven to be invaluable in natural resource management. The U.S. Forest Service (USFS) conducts periodic inventories to determine the extent, condition, volume, growth and depletions of timber on the nation's forest-land. Conventionally, forest inventory data have been collected primarily by means of field surveys that consists of numerous random plot locations where the attributes of interest (volume, growth, number of trees) are measured on a per unit area basis. Exhaustive in situ data collection is very costly, time consuming and often infeasible due to adverse field conditions. Forest inventory analysis (FIA) also utilizes remote sensing imagery to obtain initial land-use classes. Ground plots are then used to adjust remote sensing samples and obtain other estimates that can not be made from remote sensing samples. The primary goal of FIA is to provide up-to-date information on forest resources, forming a basis for realistic forest policies and programs. The FIA system has evolved over the last 60 years, adapting to the advances in forest inventory technology and to the changing information needs of the public. The most significant changes in the FIA program in several decades is the recent shift from periodic (10 year cycle) surveys to an annual system of field data collection [74], adaptation of a nationwide common sampling design, a uniform data structure for FIA inventories [75] and Internet access to the forest inventory analysis database (FIADB).

Most of the databases (scientific, statistical, spatial), populated either from field studies or experimental measurements, are based on statistical sampling techniques. FIADB is one example, others include rainfall, temperature, soil, and climatic data. Estimates from these point measurements can be generalized in a limited way over continuous surfaces such as counties, states, or regions. Geostatistical techniques are often used to interpolate the unknown values at each location in a region of interest (ROI). Over the years remote sensing analysis and inventory analysis took separate paths, and very little research has been done to integrate both data sources to extend database queries to small ROI. This chapter presents a generic framework to integrate satellite imagery with FIADB to extend queries over any arbitrary ROI.

Problem Definition: The problem of estimating values at any area (point, ROI, etc.) can be formulated as follows:

Given: We started with a set of sample plot locations (see fig. 7.1) and attribute database (FIADB) that associates each plot location with observations of the attributes of interest. The FIADB for the study area contains 2582 locations where the attributes of interest were observed. Only a small random sample of all possible locations are contained in the FIADB. The GISDB (satellite image) contains 24259794 cells that cover the entire study area. Each cell in the GISDB contains spectral data.

Find: Generate estimates for any arbitrary ROI (see fig. 7.2)

Constraint: Minimize RMS_{error}.

The research contributions discussed in this chapter are described below.

Integration: We have integrated remote sensing and field observations (FIADB) using a kNN algorithm to generate a field plot image database, which provides a framework

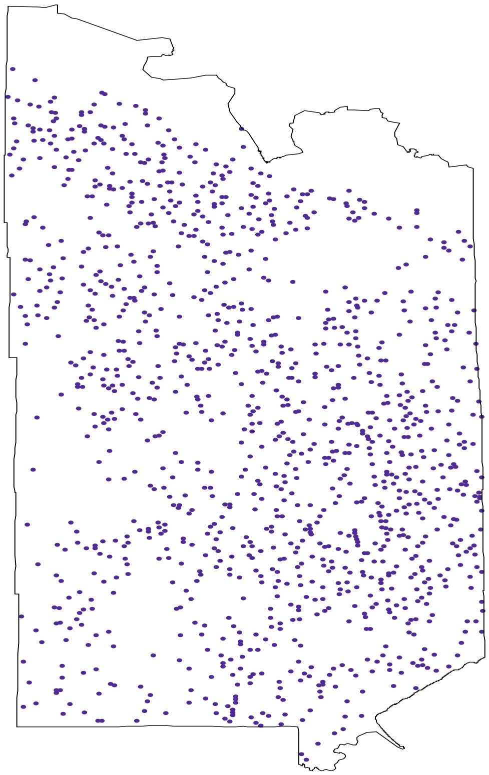


Figure 7.1: Ground sample plots locations (FIADB) in the study area.

to generalize estimates (queries) for any arbitrary ROI.

Query generalization: Database queries are extended from discrete sampling space to a continuous space using geostatistics and a field plot image database.

Internet access: The whole system is integrated into a Web enabled browsing and spatial analysis system (WEBSAS), which provides easy access to the FIADB, Remote Sensing and Spatial Databases. It allows browsing and displaying maps, generating spatial queries in a browser environment and generating summary statistics for the end user.

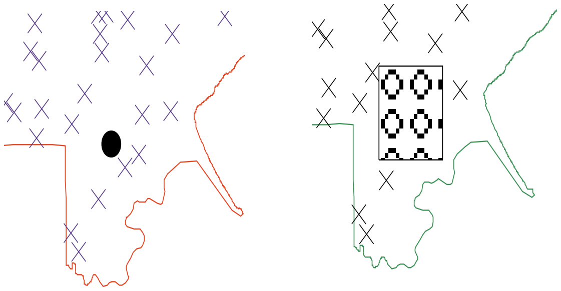


Figure 7.2: Queries involving unknown samples

7.2 System Architecture

The World Wide Web has become a popular vehicle for information distribution and client/server applications. The increasing public need for online access to earth science data products have prompted several researchers to design and build web based systems to support online access, query and analysis of these scientific data products. Dobinson et. al. [76] have described a web based system for data discovery and access earth science data products based on user defined spatial and temporal constraints, Ruixin et. al. [77] have described a user-friendly client/server web tool for online search, analysis and ordering of earth science data products, and Vatsavai et.al. [78] have described a "load balancing client/server" web based spatial analysis system (WEBSAS) for natural resource analysis and mapping. Continued demand for online access and query capabilities for FIADB have prompted us to integrate the techniques presented in the following sections into the WEBSAS. Here we present only overall system architecture for completeness of the discussion, but more details can be found in [78]. WEBSAS was built on standard 3-tier client/server architecture.

Tier 1: The Client. In general, the client is any standard Web browser. The front-end system provides easy to use graphical user interface using standard HTML tags and Java scripts. The browser constructs the Universal Resource Locator (URL) from user input, opens an HTTP connection with the server, and renders the results retrieved from the server. The WEBSAS client also consists of several applets which locally perform the rendering of geographic elements and certain geospatial analysis tasks based on the load balancing criterion.

Tier 2: The Application Server. The application server is the core component of WEBSAS. It is build around MapServer [79], loosely coupling several geospatial analysis and database access components. The URL is parsed and the appropriate geospatial analysis operation is initiated or reformulated into an appropriate query for the geospatial database engine. The results (optionally with appropriate processing and rendering applets) will be restructured and returned to the client in a format understood by the browser.

Tier 3: The Geospatial Database Access System. This layer is based on an open architecture which provides hooks to several standard raster and vector file formats, and relational database systems. The overall system architecture of WEBSAS is shown in fig. 7.3

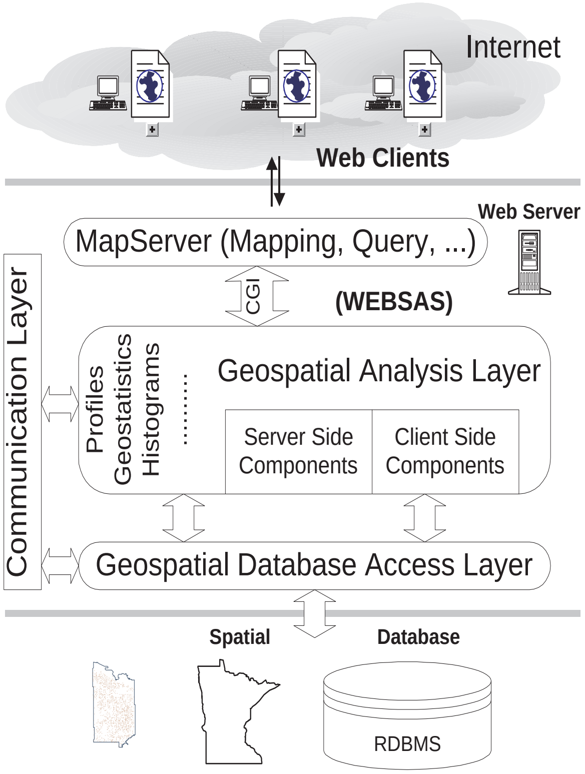


Figure 7.3: System Architecture

7.2.1 Data Sources

In this section we briefly discuss preprocessing and data characteristics of the remote sensing imagery and FIADB.

Satellite image data: Multitemporal (September 25, 1987; March 3, 1988; June 7, 1998) Landsat images (row 26, 27 and path 27) covering Saint Louis county in Minnesota, have been geometrically corrected and re-projected to Universal Transverse Mercator (UTM) projection. The images were stacked together to form a 18 channel (3 * 6 channels - blue, green, red, near infra-red(IR), mid IR, mid IR) image. A false color composite image is shown in fig. 7.4.

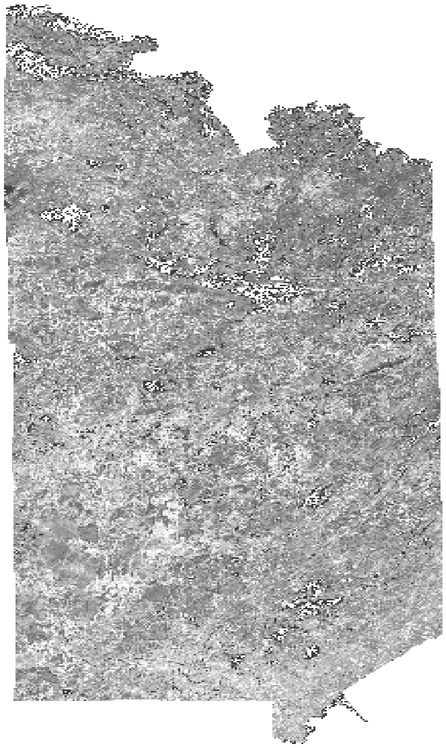


Figure 7.4: Landsat TM image

FIADB: The inventory database for this study area is part of the FIA Aspen-Birch Unit and included 2582 forested field plots. Each plot is approximately 1 acre in area and consists a 10-point clusters. These plots were pre-processed to eliminate the plots in non-reserved forest land-use class. The resulting sampling intensity was approximately

0.0009 plot/hectare [80]. The FIADB was organized into several hierarchical tables in Oracle RDBMS. We have utilized the following four tables, whose description is give below.

Table	Important attributes
COUNTY	State and County Codes, Unit name, ..
PLOT	Plot id, Cycle, Lat, Lon, Elevation, Expansion, Growth, Volume factors, ..
COND	Condition proportion, land class, reserved status, owner, forest type, density, stand size, slope, aspect, ..
TREE	Tree status, Species code, diameter height of diameter, tree class, damaging agents, trees per acre, net cubic volume flag, ..

7.3 Geostatistics

Estimating the value of an attribute at an un-sampled location is relatively easy, but getting accurate estimates is a challenging task. Until the early 90s, techniques such as Thiessen polygons, arithmetic average and inverse distance based interpolation were commonly used to estimate unknown values. These estimates are often inaccurate because they don't take into account nonuniform variability over space and time. Geostatistics [81], which is based on the theory of regionalized variables [82], is increasingly used because it takes into account the spatial dependence (or spatial correlation) of neighboring observations for estimating attribute values at un-sampled locations. Several studies in ecology, forestry, geology, biology and epidemiology have shown that geostatistics provides better estimates than conventional methods. Ordinary kriging and several other variants are the most widely-used interpolation methods in geostatistics. In general, kriging predicts missing values at un-sampled locations by taking a weighted linear average of available samples. Kriging attempts to minimize the the expected error by inferring the variance from an empirical model of spatial dependence with distance and direction; the unique model is known as a semi-variogram. The variogram is given by

$$\gamma(h) = \frac{1}{2}E[\{Z(x) - Z(x+h)\}^2], \tag{7.1}$$

where $Z(x)$ denotes the continuous variable (attribute) at x, and h denotes separation, i.e. the distance between two plots. Practically, the variogram is estimated by a series of observations $z(x_i)$

$$\hat{\gamma}(h) = \frac{1}{2N(h)} \sum_{i=1}^{n(h)} (z(x_i) - z(x_i + h))^2, \tag{7.2}$$

where $N(h)$ denotes the number of pairs of points in the distance interval h. This estimated variogram is fitted to a suitable model $\gamma(h)$, often using nonlinear regression. This experimental variogram provides weights for the ordinary kriging predictor. The ordinary kriging predictor can be written as a weighted average of n observations

$$\hat{z}(x_0) = \sum_{i=1}^{n} \lambda_i \cdot z(x_i), \tag{7.3}$$

where λ_i denotes the weight of the i^{th} observation. Detailed treatment of kriging and its variants can be found in [83].

In several studies, geostatics were successfully employed as an interpolation technique for generating estimates at un-sampled locations. Goovaerts [84] has extensively studied the application of geostatistics in soil sciences. Nalder et. al. [85] have compared several kriging variants and gradient-pulse-inverse distance squared (GIDS) techniques for spatial interpolation of climatic normals. Collins et. al. [86] have conducted a comparative study of several spatial interpolation techniques for meteorological data. Most of these studies have employed domain specific systems (e.g. GIS or Spatial Statistical packages) or specifically written software. Very little work has done in the database community to handle interpolated data.

Recently, Stephne Grumbach et. al. [87] have studied the management of interpolated data in a relational database framework. Their approach was to separate the complexities of data organization and the operation of interpolation functions on the data. One nice feature of their model is that queries can be expressed in standard relational query language, such as SQL. Instead of integrating interpolation operators into SQL, we have separated separated the geostatistical analysis and database access for the following reasons:

Portability Incorporation of geostatistical functions (predicates) into SQL is implementation specific; by separating processing logic and integrating it into the application, we can achieve greater portability. Also, for reasons to be explained in next section, it is necessary to separate them.

Visualization Often the results are graphical in nature, such as plots or images, and can't be handled directly by SQL.

Limitations: Interpolation-based techniques may look simple and straightforward, but they do have several limitations. First, the phenomena being interpolated must be truly continuous. Secondly, the sampling must be done at a sufficient density to reflect the continuous nature of the phenomena. In reality, these conditions may hold or may not hold. Kriging assumes that the variogram is accurate over the entire study area. So the accuracy of Kriging is directly related to the accuracy of the variogram. Also, field observations are considered to be an accurate representation of an area, but the real world is dynamic. Tree species distribution is affected by several factors, such as environmental conditions, the direct and indirect influence of humans, and land-use practices. These factors can't be effectively taken into consideration without the aid of some additional data sources and knowledge. Remote sensing offers potential for improving estimates, because it provides multi-spectral, high spatial and temporal coverage of broad geographic regions. In the next section, we present a framework for integrating remote sensing imagery and field data to accurately estimate the attributes under study for any arbitrary region of interest.

7.4 kNN based integration

Forest monitoring systems utilize classification techniques for propagating field plot information through the landscape, local estimation and wall-to-wall mapping of forest variables (e.g. basal area, volume, forest type). The design-based regression estimator is a well known technique for estimating land-use patterns at regional scales (e.g. national, state) using remote sensing and field data [88], [89]. However, the design-based regression estimator techniques can't be extended to local scales (counties, PLSS) due to the small sample sizes in small areas. kNN-based classification of remote sensing imagery using field plot information for estimation of forest variables at local scales was first described by [90]. Recently John Holmgren et. al. [91] have used a weighted kNN method for estimating stem volume and basal area at the local scale (compartments) by combining satellite image data with measurements from forest inventory plots. Hector et. al. [80] used a kNN-based approach for forest cover type mapping at local scale by combining Landsat TM imagery and FIA field plot information, and for wall-to-wall mapping of forest-stand density and volume [92]. The main objectives of these studies were to produce wall-to-wall mapping of a few forest variables and accurate estimates of these variables (or attributes) at local scales.

To the best of our knowledge based on a literature survey, there have been no previous studies that combined remote sensing imagery and field (sample plot) information in a database environment. Our study was also motivated by the fact that wall-to-wall mapping approaches don't capture the complex query results that can be generated from a database, such as acres by site index and forest cover type.

In this section, we present a kNN based framework for integrating satellite remote sensing imagery and field information (FIADB) to generate a FIA plot-id image. Basic kNN based integration is shown in algorithm 1.

Algorithm 1: Generate FIA plot image using kNN.

```
1. Extract FIA plot-id and coordinates from FIADB.
      plot-id[], x[], y[] ←
         SELECT p.plot, p.lat, p.lon
         FROM Plot p
         WHERE p.countycd = '137' and ....

2. Convert geographic coordinates (latitude, longitude)
   into image (UTM projected) coordinates.
      img_x[], img_y[] ← geo_to_utm(x[], y[])

3. Extract spectral reflectance values (DN) at a 3x3 window
```

centered on each FIA plot location from Landsat images
and compute mean values to generate spectral signatures.
    signature[][] ← mean(id[], DN[][])

4. For each pixel(x, y), generate the pixel vector.
    pixel[] ← DN_i where i = 1,2,..,Number of channels

5. Compute the Euclidean-distance between pixel[] and
each spectral signature (signature[][]).
    distance[plot-id[]] ← euc_dist(pixel[], plot-id[])

6. Assign output pixel(x, y) (opixel) the closest FIA plot-id.
    opixel(x, y) ← min(distance[])

7. Repeat 3 until all pixels have been scanned.

At each FIA plot location, spectral signatures were extracted from the Landsat (fig 7.5) image. Then the distance between each pixel signature in the input image (pixel[]) and each FIA plot signature (signature[]) in multi-dimensional spectral space (fig: 7.6) is computed. The Euclidean distance in vector form is give as

$$d(\mathbf{x}, \mathbf{m_i})^2 = (\mathbf{x} - \mathbf{m_i}) \cdot (\mathbf{x} - \mathbf{m_i}) i = 1, ...M \tag{7.4}$$

where $\mathbf{x}$ is the position of unknow sample, and $\mathbf{m_i}, i = 1, ...M$ are the mean signatures of M classes (i.e. M number of FIA plots).

Sample signatures were shown in the following table.

Plot	**Signature (Mean of 3x3 window)**
...	....
...	....
957	62 26 29 63 91 29 165 38 23
958	58 23 22 52 49 17 100 39 22

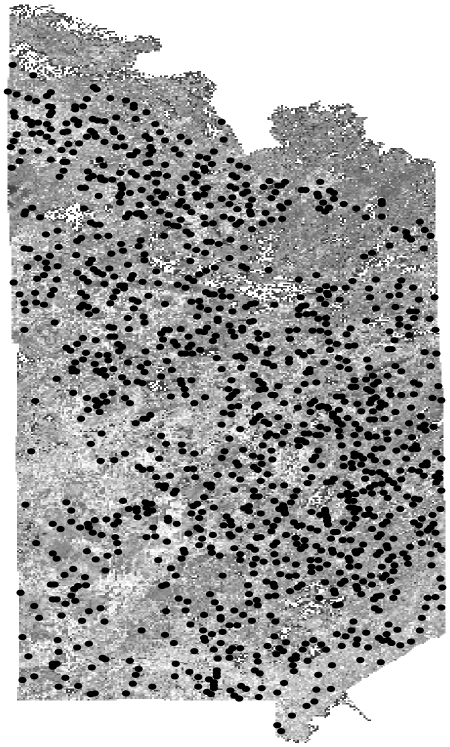


Figure 7.5: Signature Extraction from Landsat image

```
...     ................................
...     ................................
965     51 20 15 32 24 9 106 42 .... 20
```

The pixel is then assigned the closest FIA plot-id. The resulting plot-id image is shown in fig. 7.7. This image is then used in database queries for estimating variables (attributes) at local scales. Example queries can be found in the next section.

7.5 Generalizing Queries

In this section we present example queries for estimating attributes at unknown sample points, rectangular ROI, and arbitrary ROI (e.g. a census block).

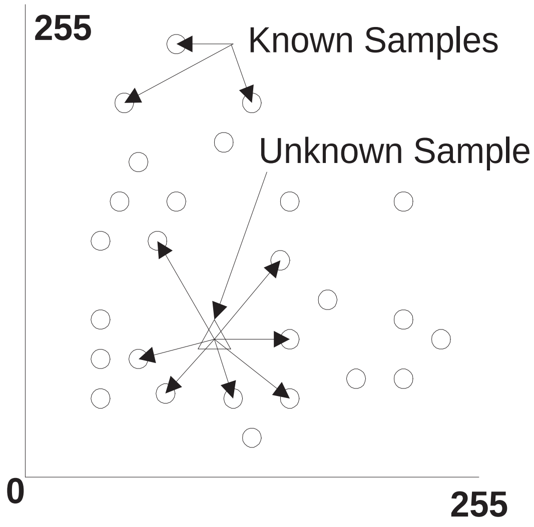


Figure 7.6: kNN Assignment

7.5.1 Generic Database Query

The following is a generic database query for estimating timber area by forest type in a given county. We can generalize the query up to the county level directly with SQL, because FIA associates with each plot the county code, and provides the current plot expansion factor (*expcurr*). Area estimates are summed across a condition or plot level variable such as ownership and land class. Estimates of numbers of trees, volume, and biomass are summed across tree level variables such as species or tree class. Area estimates are calculated by multiplying *expcurr* by the proportion of the plot in a condition (*condprop*). These conditions are expressed in WHERE clause (lines 24 to 34). Refer to FIADB database description and user's manual [75] for a detailed explanation of the algorithms used to compute various estimates.

The major limitation with these algorithms is that we can't find estimates at an un-sampled location in the county or any region (for e.g. a census block in a county) or any user defined arbitrary region of interest. However the plot-id image database generated using the k-NN algorithm can be used to generalize the queries over any arbitrary ROI. Example queries 2 to 4 show how to estimate attributes at any arbitrary point, rectangular ROI or polygon, respectively. The conditions (line 25-34) in the

Figure 7.7: Plot-id image generated kNN algorithm

WHERE clause may slightly vary for each of the queries in section 5.

```
SELECT forest_type, sum(area) Acres_timberland
FROM
(SELECT
    DECODE(c.fortypcd,
    101 , ’101 Jack pine’,
    102 , ’102 Red pine’,
    103 , ’103 Eastern white pine’,
    121 , ’121 Balsam fir’,
    122 , ’122 White spruce’,
    125 , ’125 Black spruce’,
    126 , ’126 Tamarack’,
    127 , ’127 Northern white-cedar’,
    381 , ’381 Scotch pine’,
500 , '500 Oak-hickory',
700 , '700 Elm-ash-soft maple',
800 , '800 Maple-basswood',
901 , '901 Aspen',
902 , '902 Paper birch',
904 , '904 Balsam poplar',
999 , '999 Non stocked',
'000 Missing') forest_type,
p.expcurr area
FROM r_plot p, r_cond c
WHERE
    c.plot = p.plot and
    c.countycd = p.countycd and
    c.unitcd = p.unitcd and
    c.cycle = p.cycle and
    c.statecd = p.statecd and
    p.cycle = 5 and
    p.statecd = 27 and
    p.unitcd = 1 and
    c.landclcd = 1 and
    c.siteclcd in (1,2,3,4,5,6) and
    reservcd > 0)
group by forest_type
```

7.5.2 Point Query

In this section we present a point query involving an unknown sample location (fig. 7.2(a)). In a point query, the user selects a point location involving unknown sample. This location is encoded into the URL and the request is sent to the server. The server extracts the point coordinates from the URL and reads the plot-id from the plot-id image file. This plot-id is substituted for plot (line 05), and the appropriate FIA algorithm is applied to estimate the required attributes, and the results will be sent back to the client. The following simplified SQL statement computes the area by forest type.

```
01 SELECT c.fortypcd, p.expcurr
```

```
02 FROM fia.plot p, fia.cond c
03 WHERE
04     (lines 25-34 from query in 5.1)
05     and c.plot = plot-id
06 GROUP BY c.fortypcd
```

7.5.3 Region Query

In this section we present a query to obtain estimates of area by forest cover type and size classes for any region (a rectangular window) selected by the user. The server extracts the bounds of the region and the rectangular window is read from plot-id image and a plot-id histogram (frequency) is generated. The unique plot-ids are substituted for plot-id-list (line 05) in the following SQL statement, and the resulting data is used to estimate the area using FIA algorithms and the plot-id frequency.

```
01 SELECT c.fortypcd,
02     sum( (stdszcd = 1) * p.expcurr) Sawtimber,
03     sum( (stdszcd = 2) * p.expcurr) Poletimber,
04     sum( (stdszcd = 3) * p.expcurr) Seedling_Sapling,
05     sum( (stdszcd = 4) * p.expcurr) Nonstocked,
05     sum(p.expcurr) Total_acres_timberland,
02 FROM fia.plot p, fia.cond c
03 WHERE
04     (lines 25-34 from query in 5.1)
05     and c.plot in (plot-id-list)
06 GROUP BY c.fortypcd
```

7.5.4 Polygon Query

In general polygon queries are more involved to compute. The user will display a polygonal map (e.g. public land survey system (PLSS)) and select a unit (polygon) for which FIA estimates are needed. The server extracts all the FIA plot-ids from the plot-id images (using point in polygon algorithm). After that, the estimation procedure is same as in region query described above.

```
01 SELECT c.fortypcd, p.expcurr,
          t.tpacurr, t.netcfv
FROM fia.plot p, fia.cond, c fia.tree t
WHERE
     (lines 25-34 from query in 5.1)
     and c.plot in (plot-id-list)
GROUP BY c.fortypcd
```

7.6 Analysis

Spatial interpolation and kNN based integration for extending queries can be implemented in two broad ways. Interpolated data can either be stored explicitly (full materialization) in the database, or it can be computed from the sample data on-the-fly by applying any interpolation function. Alternatively, computationally intensive steps may be done in advance to produce intermediate results which are then used during query processing to determine generalized attributes. Full materialization on all attributes will result in a huge database, but we can get a good query response time. The amount of data generated is given by the following equation.

$$Size_i = Lines \ * \ Pixels \ * \ Size(a_i). \tag{7.5}$$

For example, interpolation on volume attribute results in an image of 110 MB (3626 pixels * 7601 lines * 4byte/pixel). Also the number of lines and pixels depend on the resolution of interpolation (30 meters in this case). Finer resolution (half the resolution will result in a 4 times bigger image) will greatly effect the data size and as well the compute time. On the other hand, if the interpolation is done at query time, the computational costs are high, but the data storage requirements are minimum. Finally, if we pre-compute certain compute intense tasks (like kNN), and attribute estimation is done at query time, we can get optimum performance from the system. The empirical relationship between query response time and storage requirements for these three options are shown in the figure 7.8.

The kNN based integration framework presented in this chapter is very generic. A simple lookup table substitution will result into a wall-to-wall mapping procedures described in [80]. Also, the accuracy of estimates obtained by kNN approach are better than spatial interpolation based estimates. In another wall-to-wall forest cover mapping

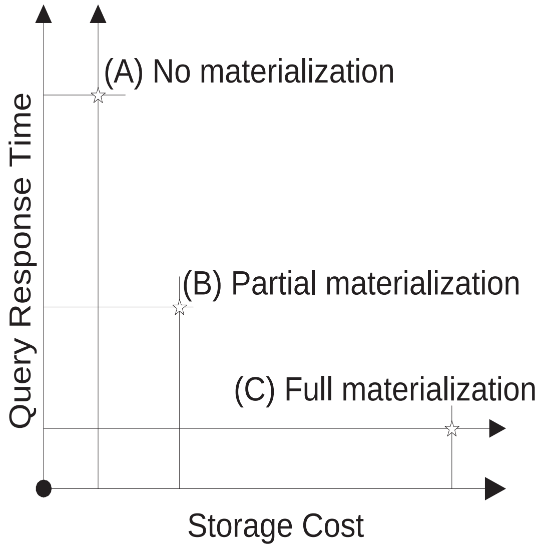


Figure 7.8: Performance of materialization choices

for this study area using kNN approach, it is found that the cross-validation estimator of overall accuracy is 0.47 and the bootstrap 0.632+ estimator of overall accuracy is 0.52. For more details on accuracy procedures and estimates refer to [80]. Also kNN is one of several possible approaches for integration of RS imagery with FIADB. The limitations of kNN method are well know when the data set contains noise. Also kNN favors spherical clusters. If the class distribution in spectral feature space is not spherical, then accuracy of plot assignment will be degraded. The methodology presented in this chapter is very generic. One can replace the kNN method with neural networks or any appropriate classification algorithm. The choice of algorithm depends on the kind of data we are dealing with and the application in hand. We have chosen kNN because of its wide-spread use in this application domain. The queries presented here are representative of several possible queries. More detailed description of several queries on FIADB can be found in [75]. All these queries can be extended to any arbitray ROI using the framework presented in this chapter without any modifications to the system. We only need to incorporate these new queries in a template file.

7.7 Conclusions

We have presented a kNN based algorithm to integrate remote sensing imagery and inventory database to extend queries involving unknown samples. We have presented several example queries that arise from practical situations. This methodology is integrated into a WebGIS, which provides browsing, mapping, analysis and query capabilities for Internet users [93]. The integrated approach provides better estimates than interpolated data using geostatistics. However, estimates are as accurate as the accuracy of finding the closest FIA plot at an unknown sample location. Because of the time lag between the sample data collection (inventory) and remote sensing acquisition, data at some of the plots may have changed (e.g. timber harvesting, damage due to insects etc). Research is needed in automatically detecting those changes and eliminating the changed plots from training data. Also, we have used a 3x3 window at each location for generating mean spectral plots, this approach eliminates noise (often arise due to mis-registration), but better approaches like region growing algorithms may help to improve the overall accuracy of the estimates. We are also interested in investigating other classification algorithms, like neural networks, for integration of heterogeneous and disparate databases. Selective materialization (e.g. plot-id image using compute intense kNN), integration of disparate data sources, and the load-balancing nature of WEBSAS have resulted in an efficient system for extending queries against inventory data to any arbitrary regions of interest.

Chapter 8

Spatial Semi-supervised Learning

8.1 Introduction

Widespread use of spatial databases [94], an important subclass of multimedia databases, is leading to an increasing interest in mining interesting and useful but implicit spatial patterns[95, 96, 97, 98]. Traditional data mining algorithms[17] often make assumptions (e.g. independent, identical distributions) which violate Tobler's first law of Geography: everything is related to everything else but nearby things are more related than distant things[18]. In other words, the values of attributes of nearby spatial objects tend to systematically affect each other. In spatial statistics, an area within statistics devoted to the analysis of spatial data, this is called spatial autocorrelation[19]. Knowledge discovery techniques which ignore spatial autocorrelation typically perform poorly in the presence of spatial data. The simplest way to model spatial dependence is through spatial covariance. Often the spatial dependencies arise due to the inherent characteristics of the phenomena under study, but in particular they arise due to the fact that imaging sensors have better resolution than object size. For example, remote sensing satellites have resolutions ranging from 30 meters (e.g., Enhanced Thematic Mapper of Landsat 7 satellite of NASA) to one meter (e.g., IKONOS satellite from SpaceImaging), while the objects under study (e.g., Urban, Forest, Water) are much bigger than 30 meters. As a result, the per-pixel-based classifiers, which do not take spatial context into account, often produce classified images with *salt and pepper* noise. These classifiers also suffer in terms of classification accuracy.

There are two major approaches for incorporating spatial dependence into classification/prediction problems. They are spatial autoregression models [20], [21], [22], [23], [24], [25] and Markov Random Field models [26], [27], [28], [29], [30], [31], [32]. Here we want to make a note on the terms spatial dependence and spatial context. These words originated in two different communities. Natural resource analysts and statisticians use spatial dependence to refer to spatial autocorrelation and the image processing community uses spatial context to mean the same. We use spatial context, spatial dependence, and spatial autocorrelation interchangeably to relate to readers of both communities. We also use classification and prediction interchangeably. Natural resource scientists, ecologists and economists have incorporated spatial dependence in spatial data analysis by incorporating spatial autocorrelation into logistic regression models (called SAR). Spatial Autoregressive Regression (SAR) model states that the class label of a location is partially dependent on the class labels of nearby locations and partially dependent on the feature values. SAR tends to provide better models than logistic regression in terms of achieving higher confidence (R^2). Similarly Markov Random Fields (MRFs) is a popular model for incorporating spatial context into image segmentation and land-use classification problems. Over the last decade, several researchers [31], [29], [32] have exploited spatial context in classification using Markov Random Fields to obtain higher accuracies over their counterparts (i.e. non-contextual classifiers). MRFs provide a uniform framework for integrating spatial context and deriving the probability distribution of interacting objects.

There is little literature comparing alternative models for capturing spatial context, hampering the exchange of ideas across communities. For example, solution procedures [23] for SAR tend to be computationally expensive just like the earlier stochastic relaxation [28] approaches for MRF despite optimizations such as sparse-matrix techniques [24], [25]. Recently new solution procedures, e.g. graph cuts [26], have been proposed for MRF. An understanding of the relationship between MRF and SAR will facilitate the development of new solution procedures for SAR. It will also likely lead to cross fertilization of other advances across the two communities.

We compare the SAR and MRF models in this paper using a common probabilistic framework. SAR makes more restrictive assumptions about the probability distributions of feature values as well as the class boundaries. We show that SAR assumes the conditional probability of feature value given a class label belongs to an exponential family, e.g. Gaussian, Binomial, etc. In contrast MRF models can work with many other probability distributions. SAR also assumes the linear separability of classes in a transformed feature space resulting from the local smoothing of feature values based on autocorrelation parameters. MRF can be used with non-linear class boundaries. Readers familier with classification models which ignore spatial context may find the following analogy helpful. The relationship between SAR and MRF is similar to the relationship between logistic regression and Bayesian classification.

Let us revise our problem definition by relaxing the i.i.d. assumption.

8.1.1 Spatial Semi-supervsied Learning: Problem Formulation

Given:

A spatial framework $\mathcal{S}$ consisting of sites s_{ij} ordered as an $l \times p$ matrix, where l is the number of lines and p is the number of pixels, and $\{1 \leq i \leq l, 1 \leq j \leq p\}$.

A $d-$dimensional feature vector at each site s_{ij}, is represented by x_{ij}, where x_{ij} is a continuous random variable.

A set of discrete labels $Y = \{y_1, \dots, y_k\}$, where k is the total number of distinct labels

Training Dataset $D = D_l \cup D_{ul}$, where D_l contains labeled training plots and D_{ul} contains unlabeled training plots.

A parametric model (e.g., Gaussian).

Find:

Estimate parameter vector Θ (e.g., $\{\mu_i, \Sigma_i\}$).

Objective:

Maximize complete data *log-likelihood* $L(\Theta)$.

Assumptions:

A1 The size of the labeled training dataset is less than 10 to 30 times the number of dimensions

A2 Thematic classes are separable

A3 Classes are not unimodal

A4 Feature vectors are correlated.

A5 D_l and D_u samples are generated by GMM_l and GMM_u.

Before getting into the spatial semi-supervsied algorithm, we look at two competing spatial classification methods and make theoretical comparisons between them.

8.2 Classification Without Spatial Dependence

In this section we briefly review two major statistical techniques that have been commonly used in the classification problem. These are logistic regression modeling and Bayesian classifiers. These models do not consider spatial dependence. Readers familiar with these two models will find it easier to understand the comparison between SAR and MRF.

8.2.1 Logistic Regression Modeling

Given an $n-$vector $\mathbf{y}$ of observations and an $n \times m$ matrix $\underline{X}$ of explanatory data, classical linear regression models the relationship between y and $\underline{X}$ as

$$\mathbf{y} = \mathbf{X}\beta + \epsilon.$$

Here $\mathbf{X} = [1, \underline{X}]$ and $\beta = (\beta_0, \ldots, \beta_m)^t$. The standard assumption on the error vector ϵ is that each component is generated from an independent, identical, zero-mean and normal distribution, i.e., $\epsilon_i = N(0, \sigma^2)$.

When the dependent variable is binary, as is the case in the “bird-nest” example, the model is transformed via the logistic function and the dependent variable is interpreted as the probability of finding a nest at a given location. Thus, $Pr(l|y) = \frac{e^y}{1+e^y}$. This transformed model is referred to as **logistic** regression [20].

The fundamental limitation of classical regression modeling is that it assumes that the sample observations are independently generated. This may not be true in the case of spatial data. As we have shown in our example application, the explanatory and the independent variables show a moderate to high degree of spatial autocorrelation. The inappropriateness of the independence assumption shows up in the residual errors, the ϵ_i's. When the samples are spatially related, the residual errors reveal a systematic variation over space, i.e., they exhibit high spatial autocorrelation. This is a clear indication that the model was unable to capture the spatial relationships existing in the data. Thus the model may be a poor fit to the geospatial data. Incidentally the notion of spatial autocorrelation is similar to that of time autocorrelation in time series analysis but is more difficult to model because of the multi-dimensional nature of space. A statistic that quantifies spatial autocorrelation is introduced in the spatial autoregression model (SAR). Inaddition, a logistic regression based classifier is equivalent to a perceptron [99], [100], [42] which can only separate linearly separable classes.

8.2.2 Bayesian Classification

Bayesian classifiers use Bayes' rule to compute the probability of the class labels given the data:

$$Pr(l_i|X) = \frac{Pr(X|l_i)Pr(l_i)}{Pr(X)} \tag{8.1}$$

In the case of the location prediction problem, where a single class label is predicted for each location, a decision step can assign the most-likely class chosen by Bayes' rule to be the class for a given location. This solution is often referred to as the maximum a posteriori estimate(MAP).

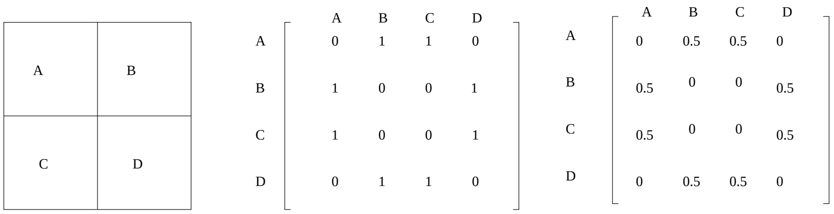
Given a learning data set, $Pr(l_i)$ can be computed as a ratio of the number of locations s_j with $f_L(s_j) = l_i$ to the total number of locations in S. $Pr(X|l_i)$ also can be estimated directly from the data using the histograms or a kernel density estimate over the counts of locations s_j in S for different values X of features and different class labels l_i. This estimation requires a large training set if the domains of features f_{X_k} allow a large number of distinct values. A possible approach is that when the joint-probability distribution is too complicated to be directly estimated, then a sufficiently large number of samples from the conditional probability distributions can be used to estimate the *statistics* of the full joint probability distribution [1] . $Pr(X)$ need not be estimated separately. It can be derived from estimates of $Pr(X|l_i)$ and $Pr(l_i)$. Alternatively it may be left as unknown, since for any given dataset, $Pr(X)$ is a constant that does not affect the assignment of class labels.

	Classifier	
Criteria	Logistic Regression	Bayesian
Input	$f_{x_1}, \ldots, f_{x_k}, f_l$	$f_{x_1}, \ldots, f_{x_k}, f_l$
Intermediate Result	β	$Pr(l_i),\ Pr(X|l_i)$ using kernel esti.
Output	$Pr(l_i|X)$ based on β	$Pr(l_i|X)$ based on $Pr(l_i)$ and $Pr(X|l_i)$
Decision	Select most likely class for a given feature value	Select most likely class for a given feature value
Assumptions		
- $Pr(X|l_i)$	Exponential Family	-
- class boundaries	linearly separable in feature space	-
- autocorrelation in class labels	none	none

Table 8.1: Comparison of Logistic Regression and Bayesian Classification

Table 8.1 summarizes key properties of logistic regression based classifiers and Bayesian classifiers. Both models are applicable to the location prediction problem if spatial autocorrelation is insignificant. However, they differ in many areas. Logistic regression assumes that the $Pr(X + l_i)$ distribution belongs to an exponential family (e.g., Binomial, normal) whereas Bayesian classifiers can work with arbitrary distribution. Logistic regression finds a linear classifier specified by β and is most effective when classes are not linearly separable in feature space, since it allows non-linear interaction among features in estimating $Pr(X|l_i)$. Logistic regression can be used with a relatively small training set since it estimates only $(k + 1)$ parameters, i.e. β. Bayesian classifiers usually need a larger training set to estimate $Pr(X|l_i)$ due to the potentially large size of feature space. In many domains, parametric probability distributions (e.g., normal [31], Beta) are used with a Bayesian classifiers if large training datasets are not available.

[1] While this approach is very flexible and the workhorse of Bayesian statistics, it is a computationally expensive process. Furthermore, at least for non-statisticians, it is a non-trivial task to decide what "priors" to choose and what analytic expressions to use for the conditional probability distributions.



$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 1 & 0 \\ B & 1 & 0 & 0 & 1 \\ C & 1 & 0 & 0 & 1 \\ D & 0 & 1 & 1 & 0 \end{array}$$

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 0.5 & 0.5 & 0 \\ B & 0.5 & 0 & 0 & 0.5 \\ C & 0.5 & 0 & 0 & 0.5 \\ D & 0 & 0.5 & 0.5 & 0 \end{array}$$

(a) Spatial Framework

(b) Neighbor relationship

(c) Contiguity Matrix

Figure 8.1: A spatial framework and its four-neighborhood contiguity matrix

8.3 Modeling Spatial Dependencies

Modeling of spatial dependency (often called context) during the classification process has improved overall classification accuracy in several previous studies. Spatial context can be defined by the correlations between spatially adjacent pixels in a small neighborhood. The spatial relationship among locations in a spatial framework is often modeled via a contiguity matrix. A simple contiguity matrix may represent the neighborhood relationship defined using adjacency, Euclidean distance, etc. Example definitions of neighborhood using adjacency include four-neighborhood and eight-neighborhood. Given a gridded spatial framework, the four-neighborhood assumes that a pair of locations influence each other if they share an edge. The eight-neighborhood assumes that a pair of locations influence each other if they share either an edge or a vertex.

Figure 8.1(a) shows a gridded spatial framework with four locations, namely A, B, C, and D. A binary matrix representation of a four-neighborhood relationship is shown in Figure 8.1(b). The row normalized representation of this matrix is called a contiguity matrix, as shown in Figure 8.1(c). Other contiguity matrices can be designed to model neighborhood relationship based on distance. The essential idea is to specify the pairs of locations that influence each other along with the relative intensity of interaction. More general models of spatial relationships using cliques and hypergraphs are available in the literature [32].

8.3.1 Spatial Autoregression Model(SAR)

We now show how spatial dependencies are modeled in the framework of regression analysis. In spatial regression, the spatial dependencies of the error term, or, the dependent variable, are directly modeled in the regression equation[20]. If the dependent values y_i' are related to each other, i.e., $y_i = f(y_j)\ i \neq j$, then the regression equation can be modified as

$$\mathbf{y} = \rho W\mathbf{y} + \mathbf{X}\beta + \epsilon. \tag{8.2}$$

Here W is the neighborhood relationship contiguity matrix and ρ is a parameter that reflects the strength of spatial dependencies between the elements of the dependent variable. After the correction term $\rho W\mathbf{y}$ is introduced, the components of the residual error vector ϵ are then assumed to be generated from independent and identical standard normal distributions. As in the case of classical regression, the SAR equation has to be transformed via the logistic function for binary dependent variables.

We refer to this equation as the ***Spatial Autoregressive Model (SAR).*** Notice that when $\rho = 0$, this equation collapses to the classical regression model. The benefits of modeling spatial autocorrelation are many: The residual error will have much lower spatial autocorrelation, i.e., systematic variation. With the proper choice of W, the residual error should, at least theoretically, have no systematic variation. If the spatial autocorrelation coefficient is statistically significant, then SAR will quantify the presence of spatial autocorrelation. It will indicate the extent to which variations in the dependent variable (y) are explained by the average of neighboring observation values. Finally, the model will have a better fit, i.e., a higher R-squared statistic. We compare SAR with linear regression for predicting nest location in Section 4.

A mixed model extends the general linear model by allowing a more flexible specification of the covariance matrix of ϵ. The mixed model can be written as

$$y = X\beta + X\gamma + \epsilon \tag{8.3}$$

where γ is the vector of random-effects parameters. The name *mixed model* comes from the fact that the model contains both fixed-effects parameters, β, and random-effects parameters, γ. The spatial autoregressive model (SAR) can be extended to a mixed model that allows for explanatory variables from neighboring observations [22]. The new model (MSAR) is given by

$$y = \alpha W y + X\beta + W X \gamma + \epsilon. \tag{8.4}$$

The marginal impact of the explanatory variables from the neighboring observations on the dependent variable y can be encoded as a $k * 1$ parameter vector γ.

Solution Procedures

The estimates of ρ and β can be derived using maximum likelihood theory or Bayesian statistics. We have carried out preliminary experiments using the spatial econometrics matlab package[2] , which implements a Bayesian approach using sampling-based Markov Chain Monte Carlo (MCMC) methods[23]. Without any optimization, likelihood-based estimation would require $O(n^3)$ operations. Recently [24], [25], and [22] have proposed several efficient techniques to solve SAR. The techniques studied include divide and conquer, and sparse matrix algorithms. Improved performance is obtained by using LU decompositions to compute the log-determinant over a grid of values for the parameter α by restricting it to $[0, 1]$.

8.3.2 Markov Random Field Classifiers

A set of random variables whose interdependency relationship is represented by a undirected graph (i.e., a symmetric neighborhood matrix) is called a Markov Random Field [30]. The Markov property specifies that a variable depends only on the neighbors and is independent of all other variables. The location prediction problem can be modeled in this framework by assuming that the class label, $f_L(s_i)$, of different locations, s_i, constitute an MRF. In other words, random variable $f_L(s_i)$ is independent of $f_L(s_j)$ if $W(s_i, s_j) = 0$.

The Bayesian rule can be used to predict $f_L(s_i)$ from feature value vector X and neighborhood class label vector L_M as follows:

$$Pr(l(s_i)|X, L\backslash l(s_i)) = \frac{Pr(X(s_i)|l(s_i), L\backslash l(s_i))Pr(l(s_i)|L\backslash l(s_i))}{Pr(X(s_i))} \tag{8.5}$$

[2] We would like to thank James Lesage (http://www.spatial-econometrics.com/) for making the matlab toolbox available on the web.

The solution procedure can estimate $Pr(l(s_i)|L\backslash l(s_i))$ from the training data by examining the ratios of the frequencies of class labels to the total number of locations in the spatial framework. $Pr(X(s_i)|l(s_i), L\backslash l(s_i))$ can be estimated using kernel functions from the observed values in the training dataset. For reliable estimates, even larger training datasets are needed relative to those needed for the Bayesian classifiers without spatial context, since we are estimating a more complex distribution. An assumption on $Pr(X(s_i)|l(s_i), L\backslash l(s_i))$ may be useful if large enough training data set is not available. A common assumption is the uniformity of influence from all neighbors of a location. Another common assumption is the independence between X and L_N, hypothesizing that all interaction between neighbors is captured via the interaction in the class label variable. Many domains also use specific parametric probability distribution forms, leading to simpler solution procedures. In addition, it is frequently easier to work with the Gibbs distribution specialized by the locally defined MRF through the Hammersley-Clifford theorem [101].

Solution Procedures

Solution procedures for the MRF Bayesian classifier include stochastic relaxation [28], iterated conditional modes [102], dynamic programming [103], highest confidence first [27] and Graph cut [26]. We have used the graph cut method and provided its description in Appendix I.

8.4 Modeling Spatial Dependencies using Markov Random Fields

Modeling of spatial dependency (often called context) during the classification process has improved the overall classification accuracy in several previous studies. Spatial context can be defined by the correlations between spatially adjacent pixels in a small neighborhood. The classification problem with MRF is formulated as follows.

A Markov Random Field is used to model the spatial context in $Pr(l_i)$. For a Markov Random Field L, the conditional distribution of a point in the field given all

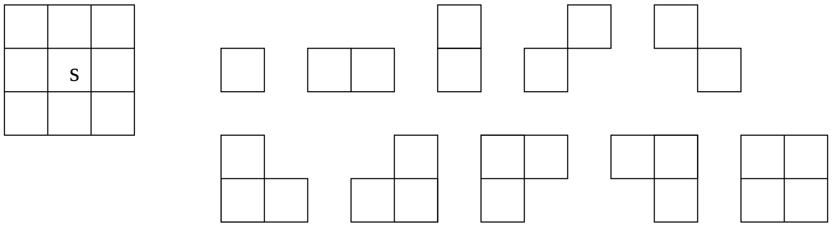


Figure 8.2: Second order cliques

other points is only dependent on its neighbors; given as

$$Pr(l(i,j)|l(k,l)); \quad \{k,l\} \neq \{i,j\} \; = \; Pr(l(i,j)|l(k,l)); \quad k,l \in N) \tag{8.6}$$

where N is the local neighborhood of pixel at (i,j). Now the problem is how to incorporate this MRF locality property into the MAP solution given in eq 5.3. Gibbs Random Fields (GRF) provide an easy way of incorporating this neighborhood information. GRFs are defined in terms of the joint distribution of random variables, which are easier to compute, as opposed to the conditional distribution given by MRFs. A Gibbs distribution for a given clique is defined as:

$$Pr(l) \; = \; \frac{1}{z} e^{-\frac{1}{T}\Sigma_C V_c(l)} \tag{8.7}$$

where $V_c(l)$ is the potential associated with clique c, and C is the set of all cliques. A clique is defined as a subset of points in X such that if a and b are any two points contained in clique c, then a and b are neighbors. Simply, cliques are specific groups of neighbors. A clique can be a single pixel, or two or more pixels such that each pixel is an immediate neighbor of the other [32]. Second order cliques are shown in Figure 8.2.

According to the Hammersley-Clifford theorem [101], there is a one-to-one correspondence between MRFs and GRFs. Therefore, if $Pr(l)$ is formulated as a Gibbs distribution, L should have the properties of a Markov Random Field. For a first order neighborhood system, the prior distribution is given by

$$Pr(L) \; = \; \frac{1}{z} e^{-\frac{1}{T}\Sigma_C V_c(L)} \; = \; \frac{1}{z} e^{-\beta\frac{1}{T}\Sigma_C t_c(L)} \tag{8.8}$$

where $t_c(L)$ is a step function. One possible function is

$$t_c(L) \;=\; \begin{cases} -1 & \text{if} \;\; l(i,j) \;=\; l(k,l). \\ 0 & \text{if} \;\; l(i,j) \;\neq\; l(k,l). \end{cases}$$

The parameter β is a weight which emphasizes the homogeneity. If $\beta = 0$, then this formulation reduces to a noncontextual classification and as β increases, more homogeneous regions are favoured. By combining ML and MRF, we get a more generic framework to model spatial dependencies. For multi-spectral image classification, ML-MRF is given by

$$\hat{l}_{MAP} = \sum_{N \in X} U_{data}(x_s) + \beta \sum_{C} t_c(l). \tag{8.9}$$

8.4.1 Solution Procedures

The minimization of eq. 8.9 is a compute intense task. Some of the solutions found in the literature to solve this problem are stochastic relaxation [28], iterated conditional modes [102], dynamic programming [103], and highest confidence first [27]. Computational demands of stochastic relaxation methods are very high. Iterated conditional mode is an improvement over stochastic relaxation, and converges to a local minimum of the energy function. Instead of minimizing equation 6 as a whole, ICM obtains an initial estimate of $U_{data}(x_s)$ using the conventional maximum likelihood classifier, which does not consider neighborhood information and merely chooses l_i to maximize $Pr(l_i|x)$ at each pixel. ICM is then applied for a fixed number of cycles, or, until it converges, to produce the final class label. The algorithm works as follows.

1. Compute initial class label l at each pixel using the non-contextual energy function $\sum_{N \in X} U_{data}(x_s)$.
2. For all pixels (i,j), update $l(i,j)$ by the class l that minimizes Eq. 5.3.
3. Repeat (2) for a fixed number of times (usually 5 or 6 iterations are sufficient).

For supervised training, assuming the classes are from multivariate normal distribution, the energy function is given by

$$U_{data}(x_s) \;=\; ln|\Sigma_i| + (x - m_i)^t \Sigma_i^{-1} (x - m_i). \tag{8.10}$$

Σ_i and m_i are the covariance matrix and mean vector for class l_i respectively.

We have experimented with a graph cut based energy minimization based on a recently proposed [26] method. The appendix 1 provides more details of this approach.

8.4.2 Comparison of SAR and MRF Using a Probabilistic Framework

We use a simple probabilistic framework to compare SAR and MRF in this section. We will assume that classes $L = l_1, l_2, \ldots, l_M$ are discrete and the class label estimate $\hat{f}_L(s_i)$ for location s_i is a random variable. We also assume that feature values (X) are constant since there is no specified generative model. Model parameters for SAR are assumed to be constant, i.e. β is a constant vector and ρ is a constant number. Finally we assume that the spatial framework is a regular grid.

We first note that the basic SAR model can be rewritten as follows:

$$y = X\beta + \rho W y + \epsilon$$

$$(I - \rho W)y = X\beta + \epsilon$$

$$y = (I - \rho W)^{-1}X\beta + (I - \rho W)^{-1}\epsilon = (QX)\beta + Q\epsilon \tag{8.11}$$

where $Q = (I - \rho W)^{-1}$ and $\beta,\ \rho$ are constants (because we are modeling a particular problem). The effect of transforming feature vector X to QX can viewed as a spatial smoothing operation. The SAR model is similar to the linear logistic model in the transformed feature space. In other words SAR model assumes linear separability of classes in transformed feature space.

Figure 8.3 shows two datasets with a *salt and pepper* spatial distribution of the feature values. There are two classes, l_1 and l_2, defined on this feature. Feature values close to 2 map to class l_2 and feature values close to 1 or 3 will map to l_1. These classes are not linearly separable in the original feature space. Spatial smoothing can eliminate the *salt and pepper* spatial pattern in the feature values to transform the distribution of the feature values. In the top part of Figure 8.3, there are few values of 3 and smoothing revises them close to 1 since most neighbors have values of 1. SAR can perform well with this dataset since classes are linearly separable in the transformed space. However, the bottom part of Figure 8.3 show a different spatial dataset where local smoothing

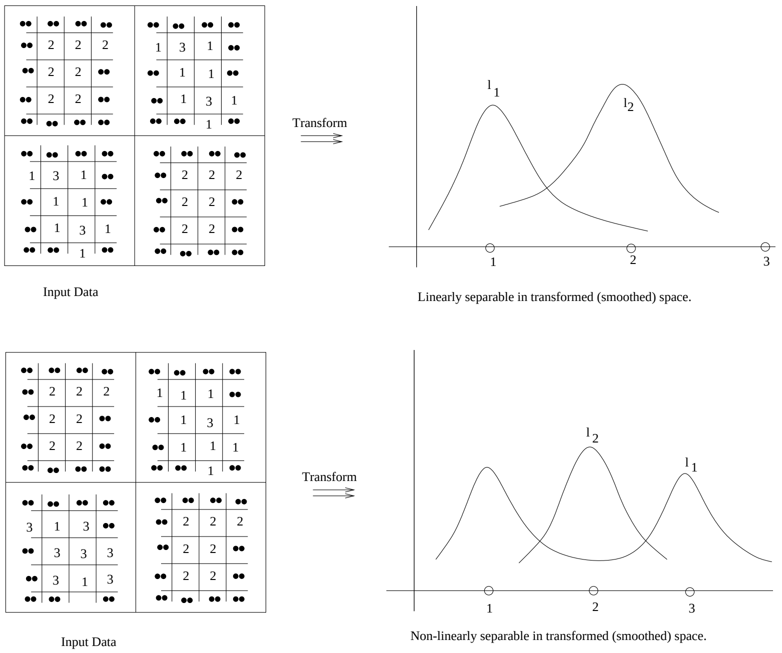

Transform
l_1
l_2
1
2
3
Input Data
Linearly separable in transformed (smoothed) space.
Transform
l_2
l_1
1
2
3
Input Data
Non-linearly separable in transformed (smoothed) space.

Figure 8.3: Spatial datasets with *salt and pepper* spatial patterns

does not make the classes linearly separable. Linear classifiers cannot separate these classes even in the transformed feature space.

Although MRF and SAR classification have different formulations, they share a common goal, estimating the posterior probability distribution: $p(l_i|X)$. However, the posterior for the two models is computed differently with different assumptions. For MRF the posterior is computed using Bayes' rule. On the other hand, in logistic regression, the posterior distribution is directly fit to the data. For logistic regression, the probability of the set of labels L is given by:

$$Pr(L|X) = \prod_{i=1}^{N} p(l_i|X) \tag{8.12}$$

One important difference between logistic regression and MRF is that logistic regression assumes no dependence on neighboring classes. Given the logistic model, the probability that the binary label takes its first value l_1 at a location s_i is:

$$Pr(l_i|X) = \frac{1}{1 + \exp(-Q_i X \beta)} \tag{8.13}$$

where the dependence on the neighboring labels exerts itself through the W matrix, and subscript i denotes the i^{th} row of the matrix Q. Here we have used the fact that y can be rewritten as in equation 8.11.

To find the local relationship between the MRF formulation and the logistic regression formulation, at point s_i

$$\begin{aligned} Pr((l_i = 1)|X, L) &= \frac{Pr(X|l_i = 1, L\backslash l_i)Pr(l_i = 1, L\backslash l_i)}{Pr(X|l_i = 1, L\backslash l_i)Pr(l_i = 1, L\backslash l_i) + Pr(X|l_i = 0, L\backslash l_i)Pr(l_i = 0, L\backslash l_i)} \qquad (8.14) \\ &= \frac{1}{1 + \exp(-Q_i X\beta)} \end{aligned}$$

which implies

$$Q_i X\beta = \ln(\frac{Pr(X|l_i = 1, L\backslash l_i)Pr(l_i = 1, L\backslash l_i)}{Pr(X|l_i = 0, L\backslash l_i)Pr(l_i = 0, L\backslash l_i)}) \qquad (8.15)$$

This last equation shows that the spatial dependence is introduced by the W term through Q_i. More importantly, it also shows that in fitting β we are trying to simultaneously fit the relative importance of the features and the relative frequency $(\frac{Pr(l_i=1,L\backslash l_i)}{Pr(l_i=0,L\backslash l_i)})$ of the labels. In contrast, in the MRF formulation, we explicitly *model* the relative frequencies in the class prior term. Finally, the relationship shows that we are making distributional assumptions about the class conditional distributions in logistic regression. Logistic regression and logistic SAR models belong to a more general exponential family. The exponential family is given by

$$Pr(x|l) = e^{A(\theta_l)+B(x,\pi)+\theta_l^T x} \qquad (8.16)$$

This exponential family includes many of the common distributions as special cases such as Gaussian, Binomial, Bernoulli, Poisson etc. The parameters θ_l and π control the form of the distribution. Equation 8.15 implies that the class conditional distributions are from the exponential family. Moreover the distributions $Pr(X|l_i = 1, L\backslash l_i)$ and $Pr(X|l_i = 0, L\backslash l_i)$ are matched in all moments higher than the mean (e.g. covariance, skew, kurtosis, etc.), such that in the difference $ln(Pr(X|l_i = 1, L\backslash l_i)) - ln(Pr(X|l_i = 0, L\backslash l_i))$, the higher order terms cancel out leaving the linear term $(\theta_l^T x)$ in equation 8.16 on the left hand-side of the equation 8.15.

8.5 Spatial Semi-supervised Learning

We now present spatial semi-supervised learning algorithm which is an extension of semi-spervised algorithm via the Markov Random Fields (MRF).

Related Work and Our Contributions: Supervised methods are extensively used in remote sensing imagery classification [69, 56]. Several approaches can be also be found in the literature that specifically deal with small sample size problems in supervised learning [11, 12, 13, 14, 15, 16]. These methods are aimed at designing appropriate classifiers, feature selection, and parameter estimation so that classification error rates can be minimized while working with small sample sizes. However, only recently that attempts have been made to incorporate unlabeled samples in supervised learning, which gave raise to new class of techniques, collectively known as semi-supervised learning methods. Well-known studies in this area include, but not limited to [43, 44, 45, 65]. The semi-supervised learning techniques have not been well explored in the remote sensing and GIS domains. Only notable study is reported in [47] for hyperspectral data analysis. The common thread between many of these methods is the Expectation Maximization (EM) [46] algorithm. Many of the semi-supervised learning methods pose class labels as the missing data and use EM algorithm to improve initial (either guessed or estimated from small labeled samples) parameter estimates. These algorithms also assume that samples are independent and identically distributed.

8.6 Spatial SSL Algorithm

There are two major approaches for modeling spatial dependencies (context, neighborhood relationships, spatial autocorrelation) in prediction/classification problems, namely, spatial autoregressive models (SAR) and Markov random fields (MRF). These two models were compared in Shekhar et al. [104]. Knowledge discovery techniques which ignore spatial autocorrelation typically perform badly on the spatial datasets. Over the last decade, several researchers [31], [29], [32] have exploited spatial context in classification using Markov Random Fields to obtain higher accuracies over their counterparts (i.e., non-contextual classifiers). MRFs provide a uniform framework for integrating spatial context and deriving the probability distribution of interacting objects. In this paper we extended the semi-supervised learning algorithm (Chapter 3) to model spatial context via the MAP-MRF model. MRF exploits spatial context through the prior probability $p(y_i)$ term in the Bayesian formulation (Section 6.2). For a Markov Random Field y, the conditional distribution of a point in the field given all other points is only dependent on its neighbors; given as

$$p(y(i,j)|y(k,l);\quad k,l \neq i,j \;=\; p(y(i,j)|y(k,l);\quad k,l \in s). \tag{8.17}$$

where s is the local neighborhood of pixel at (i,j). Now the problem is how to incorporate this MRF locality property into the MAP solution given in eq. 5.3 Gibs Random Fields (GRF) provide an easy way of incorporating this neighborhood information. GRFs are defined in terms of a joint distribution of random variables, which is easier to compute, as opposed to the conditional distribution given by MRFs. Gibs distribution for a given clique is defined as:

$$p(y) \;=\; \frac{1}{z}e^{-\frac{1}{T}\Sigma_C V_c(y)} \;=\; \frac{1}{z}e^{-\beta\frac{1}{T}\Sigma_C t_c(y)} \tag{8.18}$$

where $V_c(y)$ is the potential associated with clique c, and C is the set of all cliques, and $t_c(y)$ is a step function. The parameter β is a weight which emphasizes the homogeneity in the classified image. If $\beta = 0$, then this formulation reduces to a non-contextual classification and as β increases, more homogeneous regions are favored. By combining MAP and MRF, we get a more generic framework to model spatial dependencies. Now the **MAP-MRF** classification can be defined as

$$\hat{y}_{MAP} = \sum_{s \in X} U_{data}(x_s) + \beta \sum_{C} t_c(y). \tag{8.19}$$

where $U_{data}(x_s)$ is the non-contextual energy. According to the Hammersley-Clifford theorem [101], there is a one-to-one correspondence between MRFs and GRFs. Therefore, if $p(y)$ is formulated as a Gibbs distribution, y should have the properties of a Markov random field. Since, MRF models spatial context in the *a priori* term, we optimize a *penalized log-likelihood* [105] instead of the *log-likelihood* function. The *penalized log-likelihood* can be written as

$$\begin{aligned} ln(P(X,Y|\Theta)) = &- \sum_{C} V_C(y,\beta) - lnC(\beta) \\ &+ \sum_{i} \sum_{j} Y_{ij} \ln p_j(x_i|\Theta_i) \end{aligned} \tag{8.20}$$

Then the E-step for a given Θ^k, reduces to computing

$$\begin{aligned} Q(\Theta,\Theta^k) = &\sum_{i} \sum_{j} E(Y_{ij}|x,\theta^k) ln p_j(x_i|\theta_i) \\ &- \sum E(Vc(Y,\beta)|x,\theta^k) - lnC(\beta) \end{aligned} \tag{8.21}$$

However, exact computation of the quantities $E(Vc(Y,\beta)|x,\theta^k)$ and $E(Y_{ij}|x,\theta^k)$ in the eq. 8.21 are impossible [106]. Also the maximization of eq. 8.21 with respect to β is also very difficult, because of computing $z = C(\beta)$ is intractable except for very simple neighborhood models. Several approximate solutions for this problem in un-supervised learning can be found in [106, 107]. We extended the approximate solution provided in [106] for semi-supervised learning and showed its usefulness in improving land cover and land use predictions from remote sensing imagery. The E-step is divided into two parts: first, we compute complete data *log-likelihood* for all data points, second, for the given neighborhood, we iteratively optimize contextual energy using iterative conditional modes (ICM) [108] algorithm. Since the estimation of β is difficult [106], we assume that it is given *a priori*, and proceed with M-step as described in the semi-supervised learning algorithm. Basic spatial semi-supervised learning scheme is summarized in Table 8.2. We used the following update equations.

The new equations are given by:

$$\hat{\alpha}_j^k = \frac{(\lambda_l m_j + \sum_{i=1}^{n} \lambda_{ul} p_{ij})}{(\lambda_l m + \lambda_{ul} n)} \tag{8.22}$$

$$\hat{\mu}_j^k = \frac{(\sum_{i=1}^{m_j} \lambda_l y_{ij} + \sum_{i=1}^{n} \lambda_{ul} x_i p_{ij})}{(\lambda_l m_j + \sum_{i=1}^{n} \lambda_{ul} p_{ij})} \tag{8.23}$$

$$\hat{\Sigma}_j^k = \frac{\left\{ \begin{array}{c} \sum_{i=1}^{m_j} \lambda_l (y_{ij} - \hat{\mu}_j^k)(y_{ij} - \hat{\mu}_j^k)^t + \\ \sum_{i=1}^{n} p_{ij} \lambda_{ul} (x_i - \hat{\mu}_j^k)(x_i - \hat{\mu}_j^k)^t \end{array} \right\}}{(\lambda_l m_i + \sum_{i=1}^{n} \lambda_{ul} p_{ij})} \tag{8.24}$$

Inputs: Training data set $D = D^l \cup D^u$, where D^l consists of labeled samples and D^u contains unlabeled samples, s a neighborhood model, and β a homogeneity weight.

Initial Estimates: Build initial classifier (MLC or MAP) from the labeled training samples, D^l. Estimate initial parameter using MLE, to find $\hat{\theta}$ (see Equations 3.9 and 3.10)

Loop: While the complete data *log-likelihood* improves (see Equation 8.21):

E-step: Use current classifier to estimate the class membership of each unlabeled sample, that is, the probability that each Gaussian mixture component generated the given sample point, p_{ij} (see Equation 3.17).

ICM-step: Optimize contextual energy given by eq. 8.19 using ICM [108] algorithm.

M-step: Re-estimate the parameter, $\hat{\theta}$, given the estimated Gaussian mixture component membership of each unlabeled sample (see Equations 8.22, 8.23, 8.24)

Output: An MAP-MRF classifier, that takes the given sample (feature vector), a neighborhood model and predicts a class label.

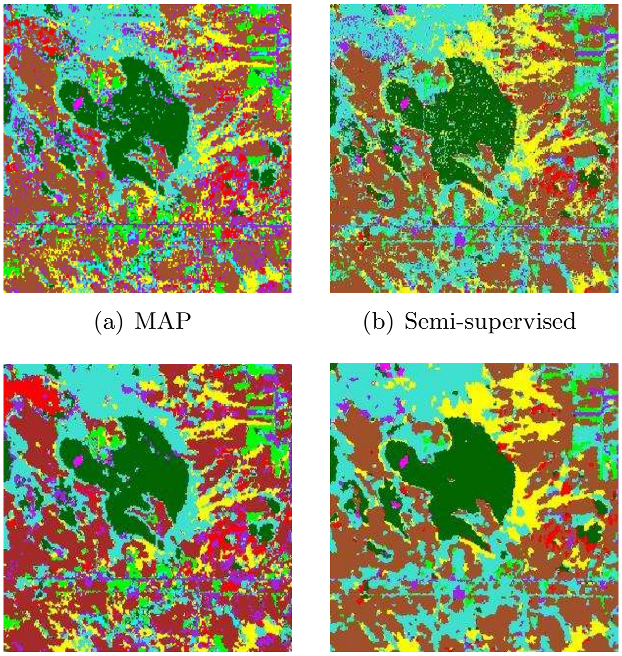
Table 8.2: Spatial Semi-supervised Learning Algorithm

8.7 Experimental Results

We used a spring Landsat 7 scene, taken on May 31, 2000 over the Cloquet town located in Carlton County of Minnesota state. Semi-supervised experiments on this dataset can be found in previous chapters. We now describe only one experiment that is to measure the performance of spatial semi-supervised algorithm.

Experiment Setup In this experiment we applied all four classifiers, namely, MAP, Semi-supervised, MAP-MRF, and Spatial Semi-supervised, on one of the randomly selected training set from the SSL experiment (see previous chapters). The results were summarized in the Figure 8.4.

The overall classification accuracy of the spatial semi-supervised algorithm is about 72%, as compared to the BC(60%), MAP-MRF(65%), and semi-supervised(68%) classifiers on the test dataset. Figure 8.4 shows the classified images using all four methods for a small area from the north west corner of the full image. From these figures it is clearly evident that the output generated by the spatial semi-supervised learning algorithm is not only accurate but is also more preferable (less salt and pepper noise) in various application domains.



(a) MAP

(b) Semi-supervised

(c) MRF

(d) Contextual Semi-supervised

Figure 8.4: Small portion from the classified *NW* corner of the Carleton image. (a) Bayesian (MAP), (b) Semi-supervised (EM-MAP), (c) MRF (MAP-MRF) and (d) Contextual Semi-supervised (EM-MAP-MRF)

8.8 Conclusions

In this chapter first we compared two popular spatial classification methods. Later we extend semi-supervised learning method with Markov Random Field model. Form the experiments (Figure 8.4(b)), it can be seen that though semi-supervised learning is more accurate than the base MAP classifier, the classified image contains lot of 'salt and pepper' noise. It should also be noted from the Figures 8.4(c) and (d) that modeling context is classification not only improves the overall accuracy but also eliminates the 'salt and pepper' noise. The output of contextual semi-supervised classification is more desirable from several other GIS applications point of view (e.g., less number of smaller polygons).

Further research is needed for incorporating additional GIS layers like population density, upland and lowland maps, digital elevation models, and soil maps into the spatial semi-supervised learning. We also identified two issues with spatial semi-supervised learning, namely, performance and convergence. In all our experiments, the contextual semi-supervised learning converged, however, formal theoretical proof of convergence is need. A close look at the spatial semi-supervised algorithm, reveals that the contextual energy is optimized at each iteration of the EM algorithm, which is clearly not desirable from the computational complexity point of view. We need smarter algorithms to speedup the convergence and as well reduce the need to optimize contextual energy at each iteration. Further research is also needed to develop other approximate solutions, such as, linear programming and graph min-cut algorithms for spatial semi-supervised algorithm presented in this paper.

Chapter 9

Concluding Remarks

Remote sensing, which provides inexpensive, synoptic-scale data with multi-temporal coverage, has proven to be very useful in land cover mapping, environmental monitoring, forest and crop inventory, urban studies, natural and man made object recognition, etc. Thematic information extracted from remote sensing imagery is also useful in a variety spatio-temporal applications. For example, land management organizations and the public have a need for more current regional land cover information to manage resources and monitor land use changes. Likewise, intelligence agencies, such as, National Geospatial Intelligence Agency (NGA), and Department of Homeland Security (DHS), utilizes pattern recognition and data mining techniques to classify both natural and man made objects from large volumes of high resolution imagery.

Image classification, i.e., assigning class labels to pixels, using some discriminant function, is one of the fundamental analysis technique used in remote sensing to generate thematic information. Many supervised classification schemes proposed in the literature work well if the land cover classes are spectrally separable, and sufficiently large number of training samples were available. But in reality, the classes under investigation are often spectrally overlapping, and accurate training samples were limited. The *i.i.d.* assumption, that is, samples are independent and identically distributed, poses severe problems with spatial datasets, as the neighboring pixels are often exhibit spatial autocorrelation. As a consequence the classified images often exhibit salt-and-pepper kind of noise. These limitations have prompted us to develop new machine learning algorithms for spatio-temporal data mining. This thesis addressed these three practical and important problems, namely overlapping classes, small training samples, and autocorrelated training samples.

One of the important solution proposed in the literature to address small training samples problem is semi-supervised learning. Most semi-supervised learning algorithms assume that the underlying model which generated both the labeled and unlabeled samples is same. However, in practice, we observed that owing to the nature of how labeled and unlabeled samples were collected, the underlying statistical model (e.g., GMMs) differ in the number of components. Though this problem was mentioned in previous studies [47, 45], and theoretical implications addressed in [50], practical solutions were missing. We developed a novel adaptive semi-supervised learning algorithm, which automatically finds the samples from the additional components of the unlabeled data model. By eliminating samples from the irrelevant components with respect to the labeled data model, our adaptive semi-supervised overcomes an important side effect that has direct bearing on the accuracy. Pooling irrelevant unlabeled training samples with labeled training samples leads either shift in the location parameter (means) or size and shape parameter (covariance) or both. This shift in the mean or increase in covariance leads to increase in the overlap of class distributions. Since probability of error is related to the overlap between class conditional probability distributions, increase in this overlap leads to the increase in the probability of error. Our experimental studies showed that the adaptive semi-supervised learning algorithm overcomes this problem by eliminating samples based on the well-known statistical hypothesis testing. As a result, the adaptive semi-supervised learning algorithm has given consistently better results as opposed to the semi-supervised learning algorithm.

A second problem that we address in this thesis is that of spectral overlap between various thematic classes. The spectral response distribution of classes are dependent on many factors including terrain, slope, aspect, soil type and moisture content, and atmospheric conditions. Thematic classes are often defined on the basis of some of these external factors, and not just the spectral characteristics of the class alone. For example, thematic classes such as upland hardwood and lowland hardwood, might have similar spectral properties, that is, their statistical distributions might be highly overlapping. Simple incorporation of these external features into classification process might not yield the desired results.

The limitations of existing methods has led us to investigate two new approaches - a fusion of KBS and MLC for classification of multi-spectral remote sensing imagery utilizing knowledge derived from ancillary spatial databases, and an hierarchical classifier, which exploits relative strengths of individual classifiers from an ensemble. We found that this approach minimizes the limitation of KB by simplifying the rule-base. In this simplified approach, the rule-base is used to stratify the image into homogeneous regions rather than classifying individual pixels. The stratified regions minimize the overlap among the classes and thus provide a robust environment for MLC. Later we extended the semi-supervised learning algorithm by modeling both continuous attributes (images) and discrete attributes (ancillary geospatial data) through a mixture of continuous and discrete distributions. This approach offered a highly flexible statistical tool to model multi-source geospatial data and the experimental resulted significant improvement in the overall accuracy.

Another related problem occurs due to the aggregate classes which are common due cost considerations of collecting training data for a large number of classes. Aggregate classes violates one of the basic assumptions that each class is described by a unimodal statistical distribution. We developed a novel classification scheme which relaxed this assumption and models each class as a GMM. Our solution then automatically recognizes these sub-classes without any additional training data. Combined with semi-supervised approach this novel algorithm allows one to classify large number of classes (which is essential to extract better information) with minimal additional training efforts.

We addressed spatial autocorrelation problem through the development of a spatial semi-supervised algorithm. This approach showed improved performance over both semi-supervised approach and spatial classification approach.

9.1 Future Directions

During the course of this research we have come across several problems where our methods can be either directly applied or customized to solve domain dependent issues. We now briefly list some of the interesting problems that we are planning to address in the future.

- With respect to adaptive semi-supervised learning we were focused mostly on the statistical hypothesis testing for matching classes and clusters. However, statistical hypothesis tests are sensitive to model assumptions and noise. We also assumed that the covariance is same for clusters and classes, which may not be true. We conducted initial experiments with two other statistical measures. One is KL-Divergence and the other is transformed divergence. These measures gives a sense of closeness between two statistical distributions, thus we can combine two distributions which are very or highly overlapping. The results are encouraging, however further experimentation is need in order to understand the overall solution quality or scenarios where these measures are more desirable than the statistical hypothesis testing.

- With respect to multi-source geospatial data classification we found that finite mixture modeling offers great flexibility to readily incorporate ancillary data. However, processing ancillary data to come up with meaningful stratified units is still an open problem. Further research is needed to automatically discover the stratified units from ancillary data for a given classification task. More experiments are needed to see the performance of our algorithm in different geographic settings.

- With respect to spatial semi-supervised learning, there is a great opportunity to scale this algorithm by utilizing modern computing platforms. It is also interesting to work on a solution to automatically find the smoothing parameter. Proving convergence of the algorithm is also a challenging task and our future research addresses these problems in more depth.

References

[1] A. Benediktsson, P.H. Swain, and O.K. Ersoy. Neural network approaches versus statistical methods in classificaion of multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 28(4):550, 1990.

[2] A.K. Skidmore, B.J. Turner, W. Brinkhof, and E. Knowles. Performance of a neural network: Mapping forest using gis and remorely sensed data. *Photogrammetric Engineering & Remote Sensing*, 63(5):501–514, May 1997.

[3] L. Bruzzone, C. Consese, F. Masellit, and F. Roli. Multisource classification of complex rural areas by statistical and neural-network approaches. *Photogrammetric Engineering & Remote Sensing*, 63(5):523–533, May 1997.

[4] J.D. Paola and R.A. Schowengerdt. The effect of neural-network structure on a multispectral land-use/land-cover classification. *Photogrammetric Engineering & Remote Sensing*, 63(5):535–544, May 1997.

[5] R. S. DeFries, M. Hansen, and J. R. G. Townshend. Global land cover classifications at 8 KM spatial resolution: the use of training data derived from Landsat imagery in decision tree classifiers. *International Journal of Remote Sensing*, 16:3141–3168, 1998.

[6] Mark A. Friedl and Carla E. Brodley. Decision Tree Classification of Land Cover from Remotely Sensed Data. *Remote Sensing of Environment*, 61:399–409, 1997.

[7] Mark A. Friedl, Carla E. Brodley, and Alan H. Strahler. Maximizing Land Cover Classification Accuracies Produced by Decision Trees at Continental to Global Scales. *IEEE Transactions on Geoscience and Remote Sensing*, 37:969–977, 1999.

[8] B. Scholkopf. *Support Vector Machines.* Oldenbourg Verlag, 1997.

[9] G.M. Foody and A. Mathur. A relative evaluation of multiclass image classification by support vector machines. 42(6):1335–1343, June 2004.

[10] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, NJ-07632, 1988.

[11] R. Duin. Classifiers in almost empty spaces. In *Proc. 15th Int. Conference on Pattern Recognition (Barcelona, Spain, Sep.3-7), vol. 2, IEEE Computer Society Press*, pages 1–7., 2000.

[12] K. Fukunaga and Raymond R. Hayes. Effects of sample size in classifier design. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(3):252–264, 1989.

[13] Sarunas J. Raudys and Anil K. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(3):252–264, 1991.

[14] Sarunas Raudys. On dimensionality, sample size, and classification error of nonparametric linear classification algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):667–671, 1997.

[15] S. Tadjudin and David A. Landgrebe. Covariance estimation with limited training samples. *IEEE Trans. Geosciences and Remote Sensing.*, 37(4):2113–2118, 1999.

[16] M. Skurichina and R. Duin. Stabilizing classifiers for very small sample sizes. In *Proc. 10th Int. Conference on Pattern Recognition, IEEE Computer Society Press*, pages 891–896, 1996.

[17] R. Agrawal. Tutorial on database mining. In *Thirteenth ACM Symposium on Principles of Databases Systems*, pages 75–76, Minneapolis, MN, 1994.

[18] W.R. Tobler. *Cellular Geography, Philosophy in Geography.* Gale and Olsson, Eds., Dordrecht, Reidel, 1979.

[19] N.A. Cressie. *Statistics for Spatial Data (Revised Edition).* Wiley, New York, 1993.

[20] L Anselin. *Spatial Econometrics: methods and models.* Kluwer, Dordrecht, Netherlands, 1988.

[21] J. LeSage. Regression Analysis of Spatial data. *The Journal of Regional Analysis and Policy (Publisher: Mid-Continent Regional Science Association and UNL College of Business Administration)*, 27(2):83–94, 1997.

[22] J. P. LeSage and R.K. Pace. Spatial dependence in data mining. In *Geographic Data Mining and Knowledge Discovery.* Taylor and Francis, forthcoming, 2001.

[23] J.P. LeSage. Bayesian estimation of spatial autoregressive models. *International Regional Science Review*, (20):113–129, 1997.

[24] R. Pace and R. Barry. Quick Computation of Regressions with a Spatially Autoregressive Dependent Variable. *Geographic Analysis*, 1997.

[25] R. Pace and R. Barry. Sparse spatial autoregressions. *Statistics and Probability Letters (Publisher: Elsevier Science)*, (33):291–297, 1997.

[26] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts . *International Conference on Computer Vision*, September 1999.

[27] P.B. Chou, P.R. Cooper, M. J. Swain, C.M. Brown, and L.E. Wixson. Probabilistic network inference for cooperative high and low levell vision. In *In Markov Random Field, Theory and Applicaitons.* Academic Press, New York, 1993.

[28] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741, 1984.

[29] Yonhong Jhung and Philip H. Swain. Bayesian Contextual Classification Based on Modified M-Estimates and Markov Random Fields. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 34(1):67–75, 1996.

[30] S. Li. Markov Random Field Modeling. *Computer Vision (Publisher: Springer Verlag*, 1995.

[31] A. H. Solberg, Torfinn Taxt, and Anil K. Jain. A Markov Random Field Model for Classification of Multisource Satellite Imagery. *IEEE Transaction on Geoscience and Remote Sensing*, 34(1):100–113, 1996.

[32] C. E. Warrender and M. F. Augusteijn. Fusion of image classifications using Bayesian techniques with Markov rand fields. *International Journal of Remote Sensing*, 20(10):1987–2002, 1999.

[33] M. Hixson, D. Scholz, and N. Funs. Evaluation of several schemes for classification of remotely sensed data. *Photogrammetric Engineering & Remote Sensing*, 46:1547–1553, 1980.

[34] A.H. Strahler. The use of prior probabilities in maximum likelihood classification of remote sensing data. *Remote Sensing of Environment*, 10:135–163, 1980.

[35] John A. Richards and Xiuping Jia. *Remote Sensing Digital Image Analysis*. Springer, New York, 1999.

[36] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, New York, 2000.

[37] M. Hansen, R. Dubayah, and R. DeFries. Classification trees: an alternative to traditional land cover classifiers. *International Journal of Remote Sensing*, 17:1075–1081, 1996.

[38] Sreerama K. Murthy, Simon Kasif, and Steven Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

[39] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Diego, CA, 1993.

[40] J.D. Paola and R.A. Schowengerdt. A detailed comparison of backpropagation neural network and maximum likelihood classifiers for urban land use classification. *IEEE Transactions on Geoscience and Remote Sensing*, 33(4):981–996, 1995.

[41] I. Kanellopoulos and G. G. Wilkinson. Strategies and best practice for neural network image classification. *International Journal of Remote Sensing*, 18(4):711–725, 1997.

[42] Simon Haykin. *Neural Networks - A comprehensive foundation.* Macmillan, 866 Third Avenue, New York, NY-10022, 1994.

[43] T. Mitchell. The role of unlabeled data in supervised learning. In *Proceedings of the Sixth International Colloquium on Cognitive Science, San Sebastian, Spain.*, 1999.

[44] Sally Goldman and Yan Zhou. Enhancing supervised learning with unlabeled data. In *Proc. 17th International Conf. on Machine Learning*, pages 327–334. Morgan Kaufmann, San Francisco, CA, 2000.

[45] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[46] A.P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[47] B. Shahshahani and D. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon. *IEEE Trans. on Geoscience and Remote Sensing*, 32(5), 1994.

[48] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report, University of Berkeley, ICSI-TR-97-021, 1997., 1997.

[49] Shumeet Baluja. Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. In *NIPS*, pages 854–860, 1998.

[50] F.G. Cozman and I. Cohen. Unlabeled data can degrade classification performance of generative classifiers. In *Fifteenth International Florida Artificial Intelligence Society Conference*, pages 327–331, 2002.

[51] F. Maselli, C. Conese, L. Petkov, and R. Resti. Inclusion of prior probabilities derived from a non-parametric process into the maximum likelihood classifier. *Photogrammetric Engineering & Remote Sensing*, 58(2):201–207, 1992.

[52] J. Stuckens, P. R. Coppin, and M. E. Bauer. Integrating contextual information with per-pixel classification for improved land cover classification. *Remote Sensing of Environment*, 71(3):282–296, March 2000.

[53] A.K. Skidmore. An expert system classifies eucalypt forest types using thematic mapper data and a digital terrain model. *Photogrammetric Engineering & Remote Sensing*, 55(10):1449–1464, October 1989.

[54] P.V. Bolstad and T.M. Lillesand. Rule-based classification models: Flexible integration of satellite imagery and thematic spatial data. *Photogrammetric Engineering & Remote Sensing*, 58(7):965–971, July 1992.

[55] E.P. Crist and R. J. Kauth. The tasseled cap de-mystified. *Photogrammetric Engineering & Remote Sensing*, 52(1):81–86, January 1986.

[56] John R. Jensen. *Introductory Digital Image Processing, A Remote Sensing Perspective*. Prentice Hall, Upper Saddle River, NJ-07458, 1996.

[57] Paul C. Smits. Combining supervised remote sensing image classifiers based on individual class performances. In *2nd International Workshop on Multiple Classifier Systems*, pages 269–278, 69121 Heidelberg, Germany, 2001. LNCS 2096, Springer Verlag.

[58] Lorenzo Bruzzone and Roberto Cossu. A robust multiple classifier system for a partially unsupervised updating of land-cover maps. In *2nd International Workshop on Multiple Classifier Systems*, pages 259–268, 69121 Heidelberg, Germany, 2001. LNCS 2096, Springer Verlag.

[59] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[60] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering algorithms and validity measures. In *SSDBM 2001*, pages 2–22. IEEE Press, 2001.

[61] J. Han, M. Kamber, and A. K. H. Tung. Spatial Clustering Methods in Data Mining: A Survey. In *In book: Geographic Data Mining and Knowledge Discovery.* Taylor and Francis, 2001.

[62] M.J. Sabins. Convergence and Consistency of Fuzzy c-Means and ISODATA Algorithms. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 9:661–668, 1987.

[63] Robert F. Cromp and William J. Campbell. Data mining of multi-dimensional remotely sensed images. In *Proceedings of the 2nd International Conference on Information and Knowledge Management*, pages 471–480, New York, NY, USA, 1993. ACM Press.

[64] Environmental Systems Research Institute. Arc/info and arc view software. http://www.esri.com/.

[65] F.G. Cozman, I. Cohen, and M.C. Cirelo. Semi-supervised learning of mixture models. In *Twentieth International Conference on Machine Learning (ICML)*, 2003.

[66] Ranga Raju Vatsavai, Shashi Shekhar, and Thomas E. Burk. A semi-supervised learning method for remote sensing data mining. In *ICTAI*, pages 207–211, 2005.

[67] Ranga R. Vatsavai, Thomas E. Burk, Paul V. Bolstad, Marvin E. Bauer, Sonja K. Hansen, Tim Mack, Jamie Smedsmo, and Shashi Shekhar. Multi-spectral image classification using spectral and spatial knowledge. In *CISST*, 2001.

[68] P. M. Mather. *Computer processing of remotely-sensed images: an introduction.* John Wiley & Sons, Inc., New York, NY, USA, 2004.

[69] John A. Richards and Xiuping Jia. *Remote Sensing Digital Image Analysis.* Springer, New York, 1999.

[70] Mclachlan. *Mixture Models: Inference and Applications to Clustering.* CRC, New York, 1987.

[71] Xuelei and XU Lei. Investigation on several model selection criteria for determining the number of clusters. *Neural Information Processing - Letters and Reviews*, 4(1):139–148, 2004.

[72] Maja Miloslavsky and Mark J. van der Laan. Fitting of mixtures with unspecified number of components using cross validation distance estimate. *Comput. Stat. Data Anal.*, 41(3-4):413–428, 2003.

[73] M.A.T. Figueiredo and A.K. Jain. Unsupervised selection and estimation of finite mixture models. *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, 2:87–90, 2000.

[74] Paul C. Van Deusen, Stephen P. Prisley, and Alan A. Lucier. Adopting an annual inventory system: User perspectives. *Journal of Forestry*, 97(12):11–15, December 1999.

[75] Larry Bednar, Gary J. Brand, Thomas Frieswyk, Joseph F Glover, Patrick D. Miles, and S. W. Woudenberg. *The Forest Inventory and Analysis Database: Database Description and User's Manual*. USDA Forest Service, 1992 Folwell Avenue, St. Paul, MN 55108, 1999.

[76] E. R. Dobinson and R. G. Raskin. Remote access tool for earth science data. *Proceedings of the Ninth International Conference on Scientific and Statistical Database Management*, 1997.

[77] Ruxin Yang, C. Wang, M. Kafatos, X. S. Wang, and T. A. El-Ghazawi. Remote data access via the siesip distributed information system. *Proceedings of the Eleventh International Conference on Scientific and Statistical Database Management*, 1999.

[78] R. R. Vatsavai, T. E. Burk, B. T. Wilson, and S. Shekhar. A web-based browsing and spatial analysis system for regional natural resource analysis and mapping. *Proceedings of the eighth ACM Symposium on Advances in geographic information systems, ACMGIS 2000*, pages 95–101, 2000.

[79] MapServer. Mapserver home page. http://mapserver.gis.umn.edu.

[80] Hector Franco-Lopez, Alan R. Ek, and M. E. Bauer. Forest cover type mapping using k-nearest neighbors method. Under publication.

[81] P. Goovaerts. *Geostatistics for Natural Resource Evaluation*. Oxford University Press, New-York, 1997.

[82] G. Matheron. Principles of geostatistics. *Economic Geology*, 58:1246–1266, 1963.

[83] N. A. C. Cressie. *Statistics for Spatial Data*. Wiley, New-York, 1991.

[84] Pierre Goovaerts. Regional estimation of soil properties from local observations. *In proceedings of soil quality workshop, Alberta Agriculture, Food and Rural Development*, pages 51–58, 1998.

[85] I. A. Nalder and Ross W. Wein. Spatial interpolation of climatic normals: test of new method in the canadian boreal forest. *Agricultural and Forest Meteorology*, 92:211–225, 1998.

[86] F. C. Collins and P. V. Bolstad. A comparision of spatial interpolation techniques in temperature estimation. *In proceedings of the third International Conference on Integrating GIS and Environmental Modeling*, 1996.

[87] S. Grumbach, P. Rigaux, and L. Segoufin. Manipulating interpolated data is easier than you thought. *In proceedings of the 26th VLDB Conference, Cairo*, 2000.

[88] J. D. Allen. A look at the remote sensing applications program of the national agricultural statistics service. *Journal of Official Statistics*, 6:393–409, 1990.

[89] L. Ambrosio and L. I. Martinez. Land cover estimation in small areas using ground survey and remote sensing. *Remote Sensing of Environment*, 6:393–409, 2000.

[90] E. Tomppo. Satellite image-based national forest inventory of finland. *International Archives of Photogrammetry and Remote Sensing*, 28(7-1):2333–51, 1991.

[91] J. Homgren, S. Joyce, M. Nilsson, and H. Olsson. Estimating stem volume and basal area in forest compartments by combining satellite image data with field data. *Scand. J. Forest Resources*, 15:103–111, 2000.

[92] Hector Franco-Lopez, Alan R. Ek, and M. E. Bauer. Estimation and mapping of forest stand density and volume using the k-nearest neighbors method. Under publication.

[93] kNN Application. Extend field data to landscapes. http://terrasip.gis.umn.edu/projects/egis/knn.

[94] S. Shekhar, S. Chawla, S. Ravada, A.Fetterer, X.Liu, and C.T. Lu. Spatial databases: Accomplishments and Research Needs. *IEEE Transactions on Knowledge and Data Engineering*, 11(1), Jan-Feb 1999.

[95] K. Koperski, J. Adhikary, and J. Han. Spatial data mining: Progress and challenges. In *Workshop on Research Issues on Data Mining and Knowledge Discovery(DMKD'96)*, pages 1–10, Montreal, Canada, 1996.

[96] D. Mark. Geographical information science: Critical issues in an emerging cross-disciplinary research domain. In *NSF Workshop*, Feburary 1999.

[97] C. Greenman. Turning a map into a cake layer of information. *New York Times*, January 20th (http://www.nytimes.com/library/tech/00/01/ circuits/arctiles/20giss.html) 2000.

[98] John F. Roddick and Myra Spiliopoulou. A bibliography of temporal, spatial and spatio-temporal data mining research. *ACM Special Interest Group on Knowledge Discovery in Data Mining(SIGKDD) Explorations*, 1999.

[99] D.W. Hosmer and S. Lemeshow. *Applied Logistic Regression.* John Wiley & Sons, 1989.

[100] W.S. Sarle. Neural Networks and Statistical Models. In *Proceeding of 9th Annual SAS user group conference.* SAS Institue, 1994.

[101] J.E. Besag. Spatial Interaction and Statistical Analysis of Latice Systems. *Journal of Royal Statistical Society, Ser. B (Publisher: Blackwell Publishers)*, 36:192–236, 1974.

[102] J. Besag. On the statistical analysis of dirty pictures. *J. Royal Statistical Soc.*, (48):259–302, 1986.

[103] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, (9):39–55, 1987.

[104] S. Shekhar, P. Schrater, R. Vatsavai, W. Wu, and S. Chawla. Spatial contextual classification and prediction models for mining geospatial data. *IEEE Transaction on Multimedia*, 4(2):174–188, 2002.

[105] Peter J. Green. On use of the em algorithm for penalized likelihood estimation. *Journal of the Royal Statistical Society, Series B*, 52(3):443–452, 1990.

[106] W. Qian and D.M. Titterington. Estimation of parameters in hidden markov models. *Philosophical Transactions of the Royal Statistical Society, Series A*, 337:407–428, 1991.

[107] W. Qian and D.M. Titterington. Stochastic relaxations and em algorithms for markov random fields. *Journal of Statistical Computation and Simulation*, 41, 1991.

[108] J. Besag. On the statistical analysis of dirty pictures. *Journal of Royal Statistical Society*, 48(3):259–302, 1986.