Graph-Based Semi-Supervised Classification on Very High Resolution Remote Sensing Images

Yupeng Yan, Manu Sethi, Anand Rangarajan, Ranga Raju Vatsavai and Sanjay Ranka

Dept. of CISE, University of Florida, Gainesville, FL, USA

Abstract

Classification of very high resolution (VHR) remote sensing imagery is a rapidly emerging discipline but faces several challenges owing to the huge scale of the pixel data involved, indiscernibility in the traditionally used features to represent various regions, and the lack of available ground truth data. This paper provides a framework which elegantly overcomes these hurdles by providing a novel semi-supervised learning approach which employs multiscale superpixel tessellation representations of VHR imagery. Superpixels are homogeneous and irregularly shaped regions which form the backbone of our approach and are used to derive novel features by learning a decision tree. Our semi-supervised learning approach works on a superpixel graph and seamlessly combines the large margin capability of a support vector machine (SVM) with a graph based Laplacian label propagation approach to obtain a novel objective function. Further we also provide a self-contained and easily parallelizable linear iterative optimization approach based on the principle of majorization-minimization. We evaluate this approach on four different geographic settings with varying neighborhood types and draw comparisons with the popular and widely used Gaussian Multiple Instance Learning algorithm. Our results showcase several advantages in accuracy and efficiency, which coupled with the ease of model building and inherently parallelizable optimization make our framework a great choice for deployment in large scale applications like global human settlement mapping and population distribution, and change detection.

Keywords: remote sensing; superpixel segmentation; ultrametric contour maps; majorization-minimization; support vector machine; graph Laplacian; semi-supervised learning; Gaussian multiple instance learning; label propagation; surrogate function.

1 Introduction

Remote sensing technologies and machine learning algorithms intended to characterize, categorize and classify land cover have received great attention during the past two decades. Due to a surge in the launch of satellites by private companies like Digital Global (*e.g.* WorldView-3 in August 2014), images with sub-meter resolution are available easily, thereby making fine-grained classification possible across several applications like urban settlement mapping, biomass monitoring, space exploration, *etc.* New avenues for automatic classification of world-wide natural regions (*e.g.* forest, sea, terrains) and man-made structures (*e.g.* residential and commercial buildings) have also been extensively provided by such very high resolution (VHR) imagery. A few national laboratories or international agencies such as Oak Ridge National Laboratory (ORNL) and European Commission Joint Research Center (JRC) have also made great efforts to generate informal and

Figure 1: An example of the high resolution aerial images obtained from Google Earth Pro that we use in our experiments for terrain classification. The above image contains more than 1 million pixels and represents an area of about 1 sq km on the ground.

formal settlement mapping at global scale by developing various new approaches as mentioned in [23].

Routine application of machine learning algorithms for classification have to be re-examined because of certain characteristics in remote sensing applications. First, remote sensing data live on a spatiotemporal grid, so the independent and identically distributed (i.i.d) assumption does not hold as in the traditional sample-based machine learning approach. For this reason, establishing an image representation which takes the gridded nature of the data into account is necessary and important. Hence we adopt a segmentation-driven representation to leverage the spatial remote sensing grid. Secondly, as the queries in remote sensing become more complicated, newer applications require us to classify entire homogeneous regions into a single category instead of just performing pixel-level classification. This is quite different from merely labeling individual samples as in standard machine learning schemes. Thirdly, these applications produce very large volume (terabytes to petabytes) and velocity (gigabytes to terabytes per day) of data, so we are required to develop low complexity and multi-scale algorithms that can be effectively implemented on modern architectures with deep memory hierarchies—another vital challenge for existing machine learning techniques. Fourthly, experts are responsible for labeling whole regions rather than individual pixels in many cases, and we need a semi-supervised approach which does not separate training and testing regimes since the entire image is available during training. Thus, expert interaction and demarcation of training and test set are very different in remote sensing. In a nutshell, we will leverage a segmentation-based image representation and then perform a graph-based semi-supervised classification in our remote sensing application. Figure 1 shows an example image that we use in this work.

There is one fundamental point of departure which deserves more attention—the use of super-pixel tessellation representations of remote sensing images. A review of previous literature shows that either a pixel-based classification approach is used without any consideration of the local region homogeneity, or pixels are aggregated in rectangular patches to be further classified without con-

sidering the natural shape as well as the local homogeneity. In sharp contrast to these approaches, we begin with a superpixel tessellation representation of remote sensing images, by adequately capturing the common characteristic of coherent local structures. In fact, superpixels are local homogeneous groupings of pixels and the superpixel tessellation ensures that the image domain can be covered by non-overlapping superpixels. Since region homogeneity is a function of scale, we adopt a pyramid of superpixel tessellations with coarser scales using local groupings of superpixels from finer scales. This results in a scale space of superpixel tessellations—a segmentation-driven image representation that is fundamental to our framework and which has, by and large, remained unexploited by other competing remote sensing classification approaches.

We now briefly delve into the specifics of the superpixel tessellation method to be deployed. Essentially, the popular ultrametric contour map (UCM) approach for image segmentation is adapted to obtain superpixel tessellations. Here we summarize UCM and highlight its use as a feature extractor since we have not seen this aspect in any other recent work. UCM begins by combining scale-space oriented image gradients at each pixel. A graph is constructed by joining any pair of pixels which exhibit good evidence for a line segment connecting them in the actual image. The top eigenvectors of the graph Laplacian are computed and rearranged in image space thereby adapting recent techniques in graph partitioning and spectral clustering. Next, gradients are computed on the eigenvector images and combined with the original image gradients to generate a contour descriptor at each pixel. An oriented watershed algorithm is executed on the gradient image to produce the lowest level superpixel tessellation, and these superpixels are merged based on grouping heuristics to obtain the final ultrametric contour map with superpixel containment across levels ([3]).

Once this image representation is obtained, we can focus on superpixel classification. As mentioned earlier, we prefer to classify superpixels at a suitably chosen level instead of individual pixels of rectangular regions. A semi-supervised machine learning approach is most suitable in remote sensing as well as in this context of superpixel tessellation representations. The remote sensing expert is tasked with labeling $O(1)$ superpixels and the subsequent machine learning algorithm labels all the remaining superpixels. Assuming the expert labels are in place, we now extract features at each superpixel to achieve the kind of discrimination required in this application—the separation of remote sensing image data into urban, slum, forest and other categories. To this end, we gather some useful information (*e.g.* the density and type of superpixels) which stems from our chosen UCM-based image representation, and integrate it into distinctive semantic features. While the aggregated information in these features is similar to that in [34], the process of generating these features is significantly different. In this work we learn these features using decision trees so as to obtain a binary feature vector of the size of the number of classes. Because our features are learnt and do not use fixed parameters, it leads to better generalization capabilities unlike in [34]. These novel features are concatenated with more standard features (*e.g.* multi-level color histograms) to generate the complete feature vector to be used in our semi-supervised superpixel classification algorithm which we describe below.

Our semi-supervised classification algorithm is an extension of the work in [34], and combines the SVM and graph Laplacian regularization into a single objective function. SVM is a sample-based method and does not leverage the advantages offered by the gridded nature of the spatiotemporal data. This is overcome in [34] by using a two-stage pipeline in which an SVM is learnt from the sparsely available data and the rudimentary classification obtained as a result of this is further refined using a graph Laplacian objective function formed by connecting spatially adjacent superpixels. However, we believe that this pipeline approach of first performing SVM and then smoothing using the Laplacian can be eliminated by regularizing the SVM's hinge loss function with a graph Laplacian-based regularization term in the same objective function. This is a more robust approach,

in our opinion, since this minimizes the tendency to overfit when an SVM is trained with just a sparsely available ground truth data. Further, this combines the maximum margin capabilities of the SVM with the underlying gridded nature of the data. Finally, in this paper we also provide a method based on majorization-minimization to optimize our objective function.

We utilize the above approach for labeling regions into different land use and land cover (LULC) classes (*e.g.* urban, slums, forests, sea) in VHR aerial images. The rest of the paper is organized as follows. In Section 2, we briefly introduce the related work of classification on remote sensing image datasets by using semi-supervised approaches. Our approach is described in detail in Section 3 and the experimental results are reported in Section 4, followed by the conclusions in Section 5.

2 Previous work

There are four major steps involved in current remote sensing classification frameworks: collecting ground-truth data for a few sample locations, extracting features from the image, building a classification model, and predicting labels for all pixels in the entire image. Many existing approaches depend on the spectral features (*e.g.* RGB and thermal infrared) and derived features (*e.g.* histogram of oriented gradients, scale invariant feature transform, vegetation indices and textons) that are extracted at each pixel. Reviews of these pixel-based or single instance learning (SIL) based techniques can be found in [44, 16]. Most of the pixel-based classification models (*e.g.* Bayesian classifier, logistic regression, and neural networks) only establish the correlations in feature space but completely ignore the spatial and structural information of those features. Thus approaches of [31, 38] using Markov random fields(MRF) which incorporate spatial location and contextual information were proposed to improve the performance of the traditional classifiers. The utility of both spectral and spatial information was also proven to be effective in [12, 26] with a kernel-based setting wherein SVM was used for classifying high resolution images. Since the spatial correlations and feature correlations are modeled simultaneously, they are also known as spatial classification schemes which bring about much smoother class distributions in the final classification. Note that, essentially these spatial classification methods are still single instance learners, so another way to overcome the single instance limitation is to exploit additional features beyond spectral features. For example, [41, 16, 49] showed the improvement of SIL methods by adding extended features (*e.g.* texture, edge density, morphological features), and [49, 20] introduced the optimal way of linearly combining the multiple features to obtain low-dimensional representation or constructing an SVM ensemble to combine the features in multiple levels.

However, although these studies showed that the classification accuracy of SIL methods could be improved by integrating spatial contextual information or adding extended features, the underlying image complexity and interpixel relationships are still not fully exploited. Object-based classification schemes were proposed based on the idea of grouping the pixels into coherent regions by investigating the spatial and spectral features. Comparative reviews of [5, 47] revealed the superiority of the object-based approach. One can either use these objects to build a meta classifier on the features which describe the whole object instead of just a particular pixel, or simply aggregate and pool all features for pixels (belonging to the same object) into a single feature vector and then apply any single instance learning algorithm for classification. Nonetheless, we should note that all these object-based approaches fail to fully consider the important structural and spatial properties in the aggregation process. To overcome these shortcomings, multiple instance learning (MIL) methods have been developed among which the seminal work of [11, 28, 1, 21] is the most notable. In general, MIL methods have better performance than SIL schemes which can be also seen in the applications of remote sensing image classification. For example, in [39] MIL schemes

were explored for simultaneously inferring local target labels and global target decision boundaries; an MIL framework was proposed in [6] for target spectra learning in hyperspectral imagery; an efficient Gaussian Multiple Instance Learning (GMIL) algorithm was developed in [42, 43] in order to overcome the high computational cost of Citation-KNN ([45, 21]). Even though these approaches are often computationally expensive and leveraging them for global scale problems is hard, our work of utilizing irregular patches or homogeneous superpixels and adopting parallelizable machine learning techniques is able to meet the scale requirements of target applications, and it can also bypass the need for determining an appropriate size of patch or grid which always impacts the performance in terms of computation and accuracy.

Concomitantly, we summarize the evolution of graph-based semi-supervised learning (SSL) methodologies. Earlier work on SSL mainly focused on optimization ([10, 17]), multi-view learning ([32]) and transductive inference ([22, 36]). Since then, it became standard to use graphs in SSL ([15, 37]), where the node labels are assigned by using a weighted combination of its neighbors. Different principles are also used to design objective functions for label propagation. For example, in [50] a framework of harmonic energy minimization was proposed over a quadratic objective function based on a weighted graph formulated in terms of a Gaussian random field model; [4] used a regression method to determine the weighted combination—first computing the graph Laplacian and then using a regression objective to estimate a weighted combination of the principal eigenvectors for predicting the unlabeled nodes; [48] even addressed the multiple-label problem by simultaneously exploiting the inherent correlations and consistency over the graph.

In this paper, we extend the work of [33, 34] by employing a decision tree to generate semantic binary features, and combine the SVM hinge loss function with a graph Laplacian based regularization in order to implement the semi-supervised classification. Our combined approach—henceforth, referred as GLSVM—is better capable of exploiting the semantic correlation of the inter-class and intra-class feature attributes and also significantly reduce the complexity of the pipeline framework presented in [33, 34]. We will describe the details in the following sections.

3 Approach

In this paper, an efficient image representation and semi-supervised learning approach is described to analyze large scale remote sensing imagery. The goal of this work is to label all pixels in an image when only a small fraction of labeled ground truth data is available from the expert. Typically, the candidate classes are forestry, slums, urban areas and others. Our framework also supports a wide range of earth science applications that concern classification and predictions.

Our framework which is an efficient modification of the work in [34], mainly includes the following stages:

1. **Tessellation of the data into superpixels**: This step converts the data into irregular (but coherent) patches called superpixels. Superpixels correspond to coherent patches or areas in 2D. These coherent superpixels reduce the data complexity since processing is moved to the superpixel level from the pixel level. Superpixels also have a huge advantage over partitioning the image into regular patches because regular patches ignore the local variability of the underlying data w.r.t. the grid.

2. **Generating multi-pixel features for each superpixel**: At this step we generate features which are effective in discriminating between terrains present in the spatiotemporal image data. We exploit intensity, geometry, scale of tessellation to arrive at these features. Other useful side information is also picked up to produce additional semantic binary features.

3. **Building a superpixel graph**: This step constructs a superpixel graph with nodes corresponding to the superpixels and edges connecting the neighborhood nodes.

4. **Label propagation algorithm**: Labels are only available for a small number of pixels. This information is used to derive the labels for a subset of superpixels. In order to do so, we construct a novel classifier which incorporates the maximum margin property of the SVM along with the smoothness property of the graph Laplacian in the same objective function. Our algorithm utilizes these partial labels and the features of all superpixels to predict a labeling field on the superpixels.

In the following subsections we will describe these stages separately. Figure 1 is used as a running example for illustration. The size of this image is roughly one million pixels.

3.1 Superpixel formation

Since the advent of normalized cuts in [35] and graph-cuts in [13, 7], there has been considerable interest in segmenting an image into sets of superpixels. There are several techniques available in the computer vision literature. The ultrametric contour map (UCM) proposed by [3] is a popular method and sets the groundwork for our approach and therefore, we believe that it warrants a concise description in this paper for the sake of completeness.

UCM uses local and global cues to produce a hierarchy of tessellations at different scales ranging from fine to coarse. These tessellations respect the containment property, that is every fine scale tessellation is contained within the next higher (or coarse) scale tessellation. We use UCM in two different ways: (i) The first usage is more traditional and direct in the sense that a fine scale tessellation is obtained which is used for classifying (or labeling) each superpixel (as against each individual pixel). (ii) The second usage is more subtle because the tessellations obtained at the coarser scales are used as features for classifying the superpixels at the finer scale selected in (i). For example, for the target application, the tessellation at a coarser scale mostly picks up prominent boundaries thus increasing the probability of detecting urban regions. Furthermore, the density of superpixels is greater in slum regions than in other regions like urban, forests *etc.* thereby making it a suitable feature for slum detection. Therefore, these cues aid in generating novel and discriminating features for superpixels at the selected finer scale of classification. In Figure 2, we show the superpixels obtained for the image in Figure 1 corresponding to fine and coarse scales.

UCM employs the gPb contour detector ([29, 3]) to obtain a probability map of the existence of boundaries at various orientations over all pixels in the image. The soft boundary map obtained from this process does not produce closed contours. Therefore, the oriented watershed transform is used to obtain closed segments from the boundary map. These segments are hierarchically merged using a greedy algorithm to obtain an ultrametric contour map. Below we summarize the steps used in this superpixel estimation process using the gPb framework: (i) feature extraction using oriented scale-space gradients, (ii) graph construction, (iii) eigenvector computation, (iv) scale-space gradient computation on the eigenvector image, (v) combination of local and global information and (vi) oriented watershed transform to produce non-uniform tessellations, and (vii) region merging to obtain the tessellation hierarchy. While this sequence is somewhat of a simplification, the major steps have been highlighted. Note that the UCM approach obtains local and global contour information by combining information from the original image and weighted graph eigenvector "images". This perceptual grouping property is mainly responsible for obtaining good superpixel tessellations. We now detail the individual steps in the overall sequence:

Step 1: Multiscale local feature extraction: Gradient maps are obtained at multiple orientations and multiple scales for each of the feature (brightness, color, and the texton map) channels obtained

(a) Superpixels obtained at a finer scale.

(b) Superpixels obtained at a coarser scale.

Figure 2: Superpixels obtained at different scales for Figure 1. Different scales capture different properties. Coarser scales mostly pick up strong boundaries while the finer scale can be leveraged to obtain the varying density of superpixels contained in coarser superpixels. Coarser superpixels contain higher density of finer superpixels near slum regions but lower density in urban and forest regions, which is useful side information that can be exploited to encode binary features as shown in Figure 4.

from the given input image. To achieve this, a circular disc is placed at each pixel and the image discontinuity is measured by splitting the disc into two halves at various angles and by taking the difference of feature histograms on both sides of the disc. Varying the size of the disc yields a multiscale oriented gradient signal $G(\mathbf{x}, \theta)$ for every feature channel. This results in a set of *local* features which are linearly combined to obtain a multiscale oriented signal:

$$I_{\text{local}}(\mathbf{x}, \theta) = \sum_s \sum_i w_{i,s} G_{i,s}(\mathbf{x}, \theta) \tag{1}$$

where $\{w_{i,s}\}$ is a set of weights that depend on the channels and scales.

Step 2: Weighted graph construction: While the above step only considers local information for deciding the boundary, this step brings in the global information pertaining to most salient curves existing in the image. For this purpose, a weighted graph is constructed using the local filter responses above. Following the intervening contour cue ([14]) strategy, pixels within a certain distance of each other are linked by a weighted edge using the relation

$$W(\mathbf{x}, \mathbf{y}) = \exp\left\{-\alpha \max_{\mathbf{z}(\mathbf{x},\mathbf{y})} \max_{\theta} I_{\text{local}}(\mathbf{z}(\mathbf{x}, \mathbf{y}), \theta)\right\} \tag{2}$$

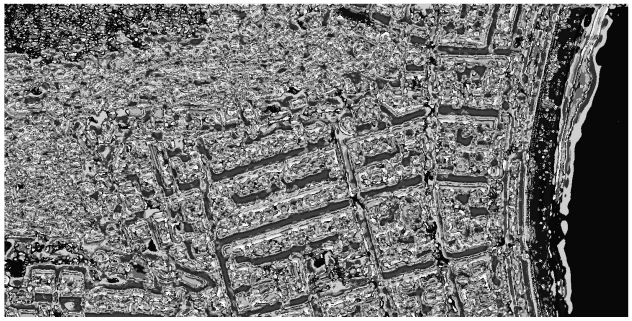
where α is a constant and $\mathbf{z}(\mathbf{x}, \mathbf{y})$ is any point lying on the line segment connecting $\mathbf{x}$ and $\mathbf{y}$.

Step 3: Eigenvector computation from the weighted graph $W(\mathbf{x}, \mathbf{y})$: Following the standard strategy of spectral clustering, the top K eigenvectors corresponding to the K smallest eigenvalues of the weighted graph are computed. Since these eigenvectors are in location space, the result is a set $\{e_k(\mathbf{x})\}$ (usually rescaled using the eigenvalues of the weighted graph).

Step 4: Spectral information obtained from the top K eigenvectors: Since gradient information computed from the scaled eigenvectors can be expected to contain complementary spectral information ([3]), a set of gradient operations in different orientations are computed to obtain

$$I_{\text{spectral}}(\mathbf{x}, \theta) = \sum_k \nabla_\theta (e_k(\mathbf{x})) \tag{3}$$

(a) Corners (magenta points) detected by Harris corner detector.



(b) Texton map generated by filter bank, using different gray levels to show 32 kinds of clusters.

Figure 3: Examples of two features used on finer-scale superpixels.

Step 5: Combination of local and spectral information: The information in equations (1) and (3) is linearly combined to obtain the final, global contour probability measure.

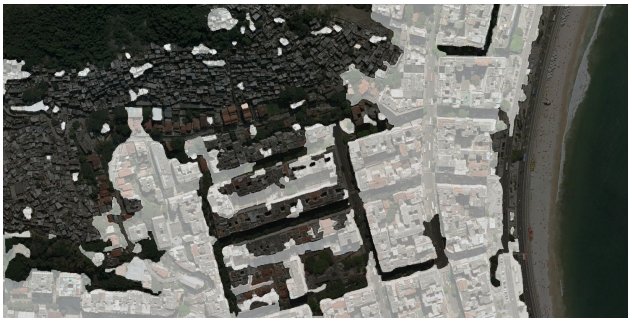
Step 6: Oriented watershed transform applied to the global contour measure: Since the global contour probability map may not always produce closed curves and therefore may not divide the image into regions, another operation is required to extract closed contours. UCM employs the Oriented Watershed Transform (OWT) ([3]) to construct closed contours. Here, the orientation that maximizes the response of the contour detection approach (gPb) is used to construct a set of regions. Further, real valued weights are associated with each possible segmentation by averaging the gPb values available at different orientations along the boundary.

Step 7: Hierarchical construction: Average strength of the common boundary is computed between two adjacent regions using the weights on the region boundaries. This quantifies the dissimilarity between two adjacent regions and is used to construct an ultrametric contour map (UCM) by gradually combining adjacent regions with the weakest dissimilarity to the strongest dissimilarity. Thus a hierarchical tree of closed regions is generated. By thresholding the UCM at a specific threshold, a set of resulting closed contours obtained can be seen either as a segmentation or as the output of the super-pixellization. Further it can be seen that the uncertainty of a segmentation can be represented: at low thresholds, the image can be oversegmented respecting even the least probable boundaries and as the threshold is increased only very strong boundaries survive. This has the benefit of introducing a trade off between the extreme ends of the segmentation.

The resulting tessellation for the image in Figure 1 is given in Figure 2. It shows that areas of significant variation require smaller patch sizes while areas with less variations are captured by larger patch sizes. We believe that this variability is one of the major strengths of the proposed approach.

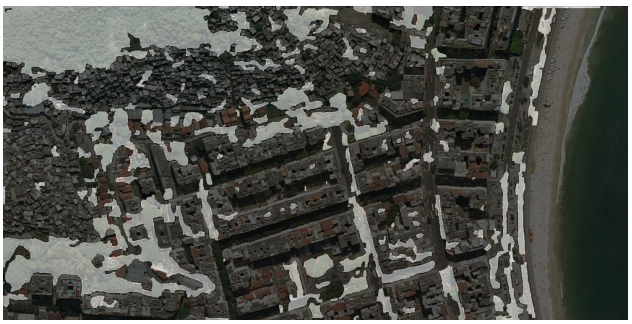
3.2 Superpixel Descriptor

Each superpixel at a finer level is described using four kinds of features—intensity histograms, textons, corner density and a binary feature derived from the coarser levels. For the intensity histograms, we quantize the grayscale intensities into 64 bins and obtain a 64 dimensional feature vector for each superpixel. For the textons, we use a regular texton filter bank ([40]) and cluster the responses of each pixel to 32 centers (Figure 3b). These 32 centers act as words of a texton vocabulary which are used to compute a 32 dimensional histogram describing the frequency of texton features in each superpixel.



(a) Binary descriptor to mark urban-like area.

(b) Binary descriptor to mark slum-like area.



(c) Binary descriptor to mark forest-like area.

(d) Binary descriptor to mark sea-like area.

Figure 4: Region overlayed with the binary descriptors, where the white parts correspond to 1 in each binary descriptor and signify a higher likelihood of being in the same semantic group.

We use the Harris corner detector ([19]) to obtain a density measure of the corners per unit area for each superpixel. As Figure 3a shows, corners can play an important role in discriminating between regions with human buildings and regions with water or vegetation. To further exploit the difference between various classes, we utilize much of the side information to derive effective semantic binary features. For example, the histograms of each color channel help to reveal the unique color properties of some classes, like green describes the forest and blue describes the sea. Similarly, the containment density of superpixels—the density of superpixels at a finer level contained within the superpixel at a coarser level—can also be used to distinguish between urban and slum dwellings, based on the visual observation that slums have a higher superpixel density relative to forest and urban settlements at the same level of the superpixel pyramid. Another key factor for distinguishing between different kind of human settlements (slum versus urban) is the size of superpixels found at coarse scales. As UCM inherently involves a combination of dense features like textons and color histograms and only keeps prominent boundaries at coarser scale, the task of determining urban regions is greatly simplified with the output of much larger superpixels. This is because urban regions have well defined boundaries which get picked up at the coarser level of the UCM hierarchy, while the weaker boundaries mostly representing slum regions get filtered out. Therefore, at the coarser levels of the UCM hierarchy regions like slums get agglomerated into big chunks while the urban regions are retained as they are. Therefore, this side information can be used to learn binary feature vectors discriminating various classes. Unlike the traditional features such as dense SIFT, HOG, *etc.* which can be used to detect these regions but suffer from the problem of determining the appropriate scale, this side information is also greatly helpful to simplify the task of discriminating the urban regions from the other classes.

We collect all these useful clues (*i.e.* side information attributes) and percolate them down the UCM hierarchy to each finer scale superpixel. All the finer superpixels contained in the larger superpixels get the same value as that of their larger parent superpixel. A decision tree is then constructed to pick up the essential attributes for dividing the superpixels into a set of groups with least possible overlap. The output of each superpixel is a binary vector of size equal to the number of groups. The occurrence of one represents the membership of a group at each superpixel and the occurrence of zero denotes the absence of that group at a particular superpixel. We constrain the number of groups to equal the number of classes (different terrains to be classified), so semantically we obtain a series of binary descriptors of slum, urban, forest, *etc.* respectively (as illustrated in Figure 4). These descriptors can be seen as a semantic complement to the four kinds of finer level features mentioned above and used in [46], since we integrate the suitable side information into the representation of a superpixel. Further, average RGB values corresponding to each superpixel are also used. Finally, we concatenate all these kinds of features to form a long feature vector to describe each superpixel of the finer tessellation. Other features like HOG ([8]) and dense SIFT ([27]) can also be added to the above framework if needed.

3.3 Semi-supervised Learning and Classification

Owing to the large size of the underlying datasets but given only a limited number of experts to label the data, it is not practical to obtain ground truth labels except at a small number of pixels. Therefore, an efficient semi-supervised learning approach has to be employed to deal with this scalability problem.

In this work, we achieve semi-supervised learning through GLSVM which follows the similar principle as in [30], so that we can fully utilize the ground truth data available from only a small number of superpixels as labeled by experts and the derived features of all the superpixels. Majorization-minimization is used to train our classifier and this iterative method is very fast

(a) Spatial graph connecting adjacent superpixels.

(b) Final result obtained after GLSVM. The output labels are smoothed during the model training.

Figure 5: Graph and resulting labels for image in Figure 1. The colors red, gray, green, yellow and blue correspond to slum, urban, forest, sand and sea regions respectively.

w.r.t. reasonable stopping conditions. By applying a spatial graph in the classifier optimization, our classification can also avoid artifacts such that neighboring regions which belong to the same class may get labeled incorrectly due to the independence assumption in conventional sample-based classifiers like SVM or k-nearest neighbors (KNN). We relegate the details of our objective function comprising the SVM hinge loss term and ℓ_2 graph Laplacian regularizer along with the iterative optimization algorithm to the Appendix.

The image in Figure 5a below shows the spatial graph we are using and Figure 5 shows the resulting labels obtained from our GLSVM algorithm based on majorization-minimization.

4 Experiments

We conduct extensive experiments on VHR imagery to evaluate the accuracy and efficiency of our classification approach. In this section we describe the data used for our experiments and the parameters used in our classification algorithm.

The image data are collected from three different geographic settings. The first five images are obtained from Rio, Brazil by using Google Earth Pro. All of these images are around one million pixels corresponding to 1 square kilometer of area. The Rio images contain two major types of settlement—formal (*e.g.* high-rise apartments and commercial complexes) and informal (*e.g.* favelas). Another two images are obtained from Madison and Milwaukee suburbs respectively from Wisconsin, USA and the last image is obtained from Sterling Heights (Detroit metropolis), Michigan, USA. All these images correspond to about 4 square kilometers with the same resolution of 1 meter as in the Rio images. The Madison image represents two distinct categories of commercial complexes and suburban residential communities, while the Milwaukee image consists of downtown and residential neighborhoods. Sterling Heights is the second largest suburb of metropolitan Detroit and the fourth largest city in the state of Michigan. The chosen subregion consists of commercial and residential areas. In addition to the major settlements, these images also contain a set of diverse categories such as forests and isolated trees (mixed with houses), grass fields and lawns, barren lands and rock outcrops, water bodies and sandy areas along the shore. Classification results of the first five Rio images are shown in Figure 5 and Figure 6. The Madison, Milwaukee, and Detroit images are shown in Figure 7.

Before the superpixel tessellation is obtained using the method given in [2], we convert our color images to grayscale and then perform Gaussian smoothing on the grayscale version of the

color images. The support of the Gaussian filter is set to 10 pixels wide and the standard deviation is chosen to be 15. Note that this heavy smoothed image data is only utilized for computing UCM ([2]), since the high resolution detail provided by the images can lead to the generation of an enormous number of superpixels which are not needed for classification purposes. Thus, it is an important step to blur the images for reducing the number of superpixels. Once the UCM hierarchy is obtained, the original images are used at every other stage of our framework.

We extract grayscale intensity histograms, RGB values averaged over each superpixel at multiple levels of the UCM hierarchy, corner density and textons for each superpixel. As explained in Section 3.2, binary descriptors are extracted from the useful side information in the UCM hierarchy and encoded into several semantic groups, which can be intuitively seen as urban-like, slum-like, forest-like, *etc.* regions.

The intensity histograms are obtained by quantizing the intensities into 64 bins. The texton features are generated by clustering and pooling into 32-dimensional vectors. The average RGB values comprise a 3-dimensional vector of average color values. The corner density feature obtained is a scalar which is multiplied by a factor of 100. The number of semantic binary features, which integrate the auxiliary attributes like coarse-scale superpixel size and multilevel superpixel containment density, is equal to the number of different neighborhood categories. All these weighted features were concatenated together to obtain a long feature vector to describe each superpixel.

To train the GLSVM classifier, we follow the same experimental setting as in [34]. The ground truth labels are provided to only about 1% (see Table 1) of the superpixels at the finest level, and this is further reduced for the Milwaukee (0.52%) and Detroit (0.26%) images. The constant value of τ is set to 2 and the stopping threshold δ of iterative majorization-minimization is set to 0.001. For all the images, we compute the misclassification error (see Table 1) by taking the weighted average of each misclassified superpixel where the weight for each one is the ratio of its covering area to the total image area.

The total time required for the overall processing (including UCM) is about 20 minutes on a sequential machine.

We compare our proposed framework against the recent GMIL algorithm in [42, 43] which showed good improvement over standard per-pixel based classification schemes. The accuracy estimates for these three images are summarized in Table 2. Results from our approach are comparable to GMIL, however, it should be noted that GMIL is computationally expensive and also requires more ground truth training data compared to our technique. Similar to the analysis in [34], our semi-supervised method preserves the advantages of (i) generating irregular coherent and hierarchical segments instead of rectangular blocks, (ii) requires only a fraction of ground-truth superpixels, (iii) has lower computation cost and better classification map.

We also compare our proposed framework with the scalable machine learning scheme in [34]. Our single-step GLSVM result shows competitive performance when compared with the pipeline-based classification approach in [34] which contains two separate stages of graph label initialization and label refinement. Further, unlike the approach in [34], we do not manually threshold the slum feature and the urban feature. Instead we automatically generate the binary descriptors by using all suitable side information attributes through a decision tree. This makes our features more generic and not limited to the slum or urban terrain classes. Further, the robustness and flexibility of our features overcomes the challenges faced in similar-looking categories that are difficult to distinguish or several special categories for which we have no prior information. Additionally, unlike the label initialization which only utilizes a very small proportion of feature data with ground-truth in model training, our semi-supervised GLSVM approach makes full use of the features of unlabeled superpixels and is also able to rapidly achieve convergence. All these advantages demonstrate that our framework is capable of being deployed at global scale and in other applications.

Figure 6: Results for 4 more images from Rio, Brazil. The left column shows the actual image and the right column corresponds to our GLSVM classification results. The colors red, gray, green, blue, and orange correspond to slum, urban, forest, sea, sand, and farm regions respectively.

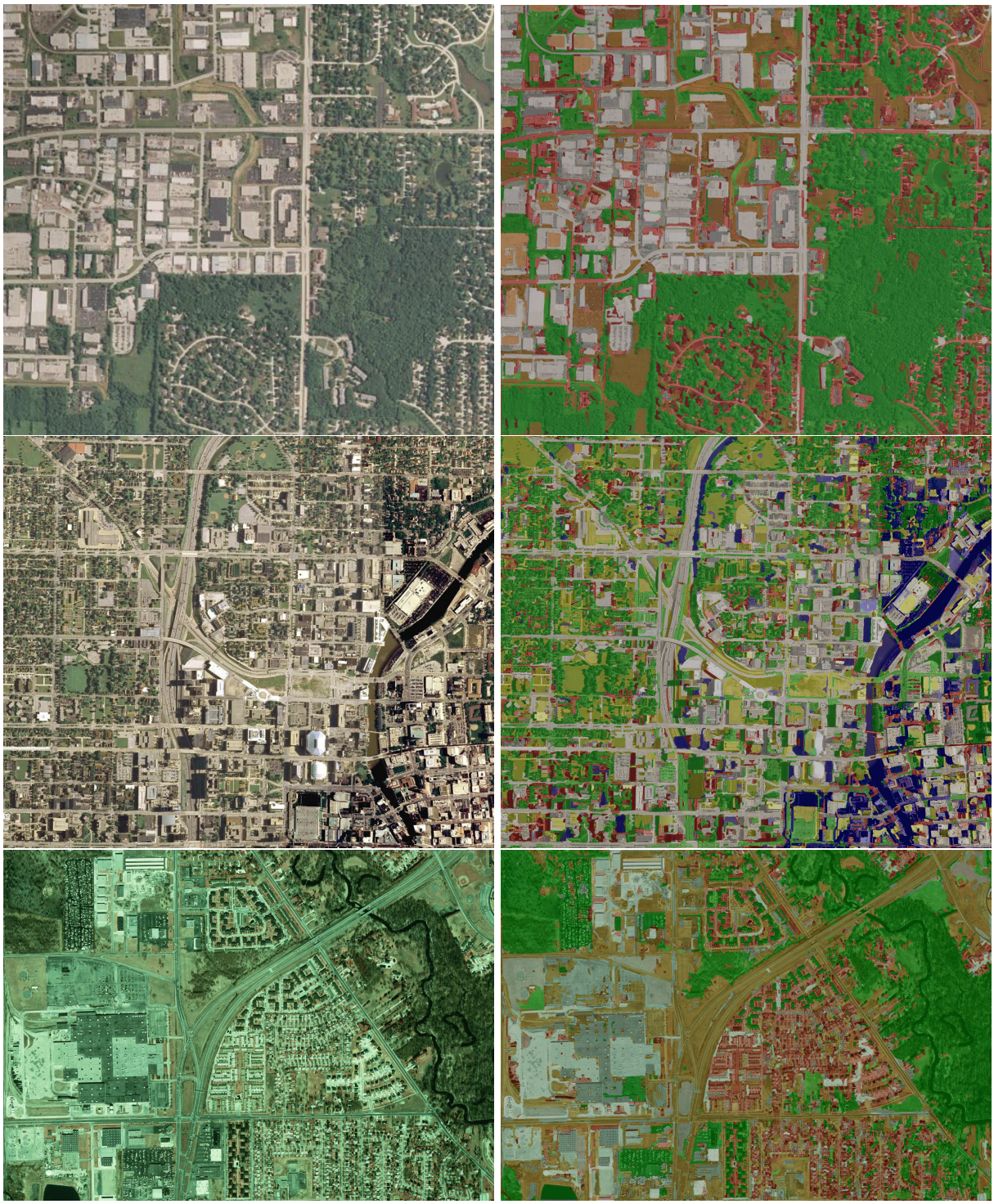


Figure 7: Results for images from Madison (top), Milwaukee (middle), and Detroit (bottom). The left column shows the actual images and the right column corresponds to our GLSVM classification results. The colors red, gray, dark green, light green, orange, and blue correspond to residential areas, commercial areas, forests, grass, road, and water.

Image	Labeled Data (%)	Misclassification Error (%)
Rio-1	1.21	13.76
Rio-2	1.49	10.83
Rio-3	1.14	11.58
Rio-4	1.38	13.19
Rio-5	1.30	9.27
Madison	1.01	5.68
Milwaukee	0.52	8.28
Detroit	0.26	8.49

Table 1: Quantitative results for the images in Figure 1, Figure 6 and Figure 7.

Image	Error Grids or Superpixels (%)	
	GMIL	GLSVM
Madison	6.8	6.87
Milwaukee	17.2	9.16
Detroit	15.39	5.49

Table 2: Results obtained with GMIL and GLSVM for the first Rio image in Figure 6 and the 3 images belonging to Madison, Milwaukee, and Detroit as shown in Figure 7.

5 Conclusions

In this paper, we have developed a novel and scalable machine learning framework for classifying different terrain types in VHR images. Accurate identification is critical for many applications, including global scale high-resolution population databases, national security, human health, and energy. To meet these challenges, we have also derived features which can effectively discriminate between different categories. This approach combines a superpixel image tessellation representation with an efficient and non-pipeline semi-supervised classification based on a majorization-minimization approach. The superpixel representation naturally coheres with local boundary information and is a major reason for obtaining good classification. Experimental evaluation on three different geographic settings shows good classification performance. Our future work will focus on extending this approach to temporal data as well as on integrating the feature extraction, selection and classification into a single scheme to improve the correlation between superpixels and model training.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pages 561–568, 2002.

[2] A. Angelova and S. Zhu. Efficient object detection and segmentation for fine-grained recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 811–818. IEEE, 2013.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011.

[4] M. Belkin and P. Niyogi. Semi-supervised learning on Riemannian manifolds. *Machine learning*, 56(1-3):209–239, 2004.

[5] T. Blaschke, G. J. Hay, M. Kelly, S. Lang, P. Hofmann, E. Addink, R. Q. Feitosa, F. van der Meer, H. van der Werff, F. van Coillie, et al. Geographic object-based image analysis – towards a new paradigm. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87:180–191, 2014.

[6] J. Bolton and P. Gader. Application of multiple-instance learning for hyperspectral image analysis. *Geoscience and Remote Sensing Letters, IEEE*, 8(5):889–893, 2011.

[7] Y. Boykov and G. Funka-Lea. Graph cuts and efficient ND image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Conference on*, volume 1, pages 886–893. IEEE, 2005.

[9] J. de Leeuw and K. Lange. Sharp quadratic majorization in one dimension. *Computational Statistics & Data Analysis*, 53(7):2471–2484, 2009.

[10] A. Demiriz and K. P. Bennett. Optimization approaches to semi-supervised learning. In *Complementarity: Applications, Algorithms and Extensions*, pages 121–141. Springer, 2001.

[11] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71, 1997.

[12] M. Fauvel, J. Chanussot, and J. A. Benediktsson. A spatial–spectral kernel-based approach for the classification of remote-sensing images. *Pattern Recognition*, 45(1):381–392, 2012.

[13] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[14] C. C. Fowlkes and J. Malik. How much does globalization help segmentation? *UC Berkeley, Tech. Rep.*, CSD-04-1340, 2004.

[15] A. B. Goldberg, X. Zhu, and S. J. Wright. Dissimilarity in graph-based semi-supervised classification. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 155–162, 2007.

[16] J. Graesser, A. Cheriyadat, R. R. Vatsavai, V. Chandola, J. Long, and E. Bright. Image based characterization of formal and informal neighborhoods in an urban landscape. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(4):1164–1176, 2012.

[17] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, pages 529–536, 2004.

[18] P. J. Groenen, G. Nalbantov, and J. C. Bioch. SVM-maj: a majorization approach to linear support vector machines with different hinge errors. *Advances in Data Analysis and Classification*, 2(1):17–43, 2008.

[19] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50. Citeseer, 1988.

[20] X. Huang and L. Zhang. An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(1):257–272, 2013.

[21] L. Jiang, Z. Cai, D. Wang, and H. Zhang. Bayesian Citation-KNN with distance weighting. *International Journal of Machine Learning and Cybernetics*, 5(2):193–199, 2014.

[22] T. Joachims et al. Transductive learning via spectral graph partitioning. volume 3, pages 290–297, 2003.

[23] P. T. Kemper, N. Mudau, and M. Pesaresi. Towards a country-wide mapping & monitoring of formal and informal settlements in south africa. *Science and Policy Report JRC92657, Joint Research Centre, European Commission*, 2015.

[24] K. Lange. The EM algorithm. In *Optimization*, pages 221–236. Springer, 2013.

[25] K. Lange. Penalty and barrier methods. In *Optimization*, pages 313–339. Springer, 2013.

[26] C.-H. Li, B.-C. Kuo, C.-T. Lin, and C.-S. Huang. A spatial–contextual support vector machine for remotely sensed image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(3):784–799, 2012.

[27] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[28] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. *Advances in Neural Information Processing Systems*, pages 570–576, 1998.

[29] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(5):530–549, 2004.

[30] S. Melacci and M. Belkin. Laplacian support vector machines trained in the primal. *The Journal of Machine Learning Research*, 12:1149–1184, 2011.

[31] G. Moser, S. B. Serpico, and J. A. Benediktsson. Land-cover mapping by Markov modeling of spatial–contextual information in very-high-resolution remote sensing images. *Proceedings of the IEEE*, 101(3):631–651, 2013.

[32] I. Muslea, S. Minton, and C. A. Knoblock. Active + semi-supervised learning = robust multi-view learning. In *ICML*, volume 2, pages 435–442, 2002.

[33] M. Sethi, Y. Yan, A. Rangarajan, R. R. Vatsavai, and S. Ranka. An efficient computational framework for labeling large scale spatiotemporal remote sensing datasets. *Contemporary Computing (IC3), 2014 Seventh International Conference on*, pages 635–640, 2014.

[34] M. Sethi, Y. Yan, A. Rangarajan, R. R. Vatsavai, and S. Ranka. Scalable machine learning approaches for neighborhood classification using very high resolution remote sensing imagery. *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2069–2078, 2015.

[35] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[36] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML*, pages 824–831. ACM, 2005.

[37] A. Subramanya, S. Petrov, and F. Pereira. Efficient graph-based semi-supervised learning of structured tagging models. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 167–176, 2010.

[38] S. Sun, P. Zhong, H. Xiao, and R. Wang. Spatial contextual classification of remote sensing images using a gaussian process. *Remote Sensing Letters*, 7(2):131–140, 2016.

[39] P. Torrione, C. Ratto, and L. M. Collins. Multiple instance and context dependent learning in hyperspectral data. *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 2009. WHISPERS'09. First Workshop on*, pages 1–4, 2009.

[40] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81, 2005.

[41] R. R. Vatsavai. High-resolution urban image classification using extended features. *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 869–876, 2011.

[42] R. R. Vatsavai. Gaussian multiple instance learning approach for mapping the slums of the world using very high resolution imagery. *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1419–1426, 2013.

[43] R. R. Vatsavai. A scalable complex pattern mining framework for global settlement mapping. *Big Data (BigData Congress), 2015 IEEE International Congress on*, pages 514–521, 2015.

[44] R. R. Vatsavai, E. Bright, C. Varun, B. Budhendra, A. Cheriyadat, and J. Grasser. Machine learning approaches for high-resolution urban land cover classification: a comparative study. In *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications*, page 11. ACM, 2011.

[45] J. Wang and J.-D. Zucker. Solving multiple-instance problem: A lazy learning approach. *International Conf. on Machine Learning (ICML)*, 2000.

[46] Y. Yan, X. Tian, L. Yang, Y. Lu, and H. Li. Semantic-spatial matching for image classification. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[47] N. Zerrouki and D. Bouchaffra. Pixel-based or object-based: Which approach is more appropriate for remote sensing image classification? *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 864–869, 2014.

[48] Z.-J. Zha, T. Mei, J. Wang, Z. Wang, and X.-S. Hua. Graph-based semi-supervised learning with multiple labels. *Journal of Visual Communication and Image Representation*, 20(2):97–103, 2009.

[49] L. Zhang, L. Zhang, D. Tao, and X. Huang. On combining multiple features for hyperspectral remote-sensing image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(3):879–893, 2012.

[50] X. Zhu, Z. Ghahramani, J. Lafferty, et al. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.

Appendix

Here we provide a detailed mathematical description of our semi-supervised majorization-minimization (SSMM) learning method outlined in Section 3.3. Before we provide the description, we note that our method runs on the finer layer of superpixels extracted from the UCM hierarchy. The coarser layers of UCM were only used to obtain the features for each superpixel of this finer layer. Therefore, all the superpixels referred to in this description only belong to the finer layer.

Let $\mathbf{s}_L$ denote the set of superpixels for which the ground truth labels are available, and similarly let $\mathbf{s}_U$ denote the set of superpixels which are unlabeled. Then our goal is to obtain the label for each superpixel in the set $\mathbf{s}_U$. Because there are $K > 2$ classes we run our SSMM algorithm K times (once for each class) in a one-versus-rest manner. Let $k \in \{1, 2, \ldots, K\}$ index the classes, and let i index each superpixel. Then the standard linear SVM hinge loss function for k^{th} class at the i^{th} superpixel, and $i \in \mathbf{s}_L$ is defined as

$$f_H^{(i)}\left(\mathbf{w}_k, b_k\right) = \max\left[0, 1 - y_{ik}\left(\mathbf{w}_k^T \mathbf{x}^{(i)} + b_k\right)\right] \tag{4}$$

where $\mathbf{x}^{(i)}$ denotes the feature vector describing the i^{th} superpixel, $\mathbf{w}_k$ and b_k are the weight vector and bias to be learnt in the algorithmic execution corresponding to the k^{th} class, and $y_{ik} = 1$ indicates that $\mathbf{x}^{(i)}$ belongs to the k^{th} class and -1 otherwise. We note in passing that the above SVM hinge loss is only defined for the superpixels belonging to the set $\mathbf{s}_L$.

In order to perform graph Laplacian smoothing akin to [34, 2], we construct a graph connecting adjacent superpixels in the spatial domain (and not in the feature domain). For every superpixel j adjacent to the i^{th} superpixel, the edge weight is set as $W_{ij} = \exp\left(-\frac{\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2}{2\tau^2}\right)$. For every other superpixel not adjacent to the superpixel i, $W_{ij} = 0$. Now similar to the work of [2], we define the penalty for the difference between two neighboring superpixels as

$$f_S\left(\mathbf{w}_k, b_k\right) = \sum_{i,j \in \mathbf{s}_L \cup \mathbf{s}_U} W_{ij} \left| \frac{\mathbf{w}_k^T \mathbf{x}^{(i)} + b_k}{\sqrt{D_{ii}}} - \frac{\mathbf{w}_k^T \mathbf{x}^{(j)} + b_k}{\sqrt{D_{jj}}} \right|^2 \tag{5}$$

where $D_{ii} = \sum_{j \in \mathbf{s}_L \cup \mathbf{s}_U} W_{ij}$. Once again, we note in passing that the above penalty function is defined for both the sets $\mathbf{s}_L$ and $\mathbf{s}_U$ of superpixels.

Therefore, in order to combine the capabilities of both the SVM and the graph Laplacian regularization, we combine the above two terms, f_H and f_S, with the quadratic term $\mathbf{w}_k^T \mathbf{w}_k$, and minimize the following objective function:

$$L\left(\mathbf{w}_k, b_k\right) = \frac{1}{2}\mathbf{w}_k^T \mathbf{w}_k + \lambda_H \sum_{i \in \mathbf{s}_L} f_H^{(i)}\left(\mathbf{w}_k, b_k\right) + \lambda_S f_S\left(\mathbf{w}_k, b_k\right) \tag{6}$$

where λ_H and λ_S are positive parameters to control the trade-off among the maximum margin based regularization term, hinge loss and the graph Laplacian smoothing penalty.

The gradient descent optimization method cannot be directly used because the first term, f_H is not differentiable when $y_{ik}\left(\mathbf{w}_k^T \mathbf{x}^{(i)} + b_k\right) = 1$. While most strategies deflect the optimization of SVM objective functions to the dual formulation, we instead follow the approach given in [25, 18] wherein a sharp quadratic surrogate function ([18, 9]) is used to majorize this hinge loss. The majorized version of (4) can be written as

$$g_H^{(i)}\left(\mathbf{w}_k, b_k, z\right) = \frac{1}{4z_{ik}}\left[1 - y_{ik}\left(\mathbf{w}_k^T \mathbf{x}^{(i)} + b_k\right) + z_{ik}\right]^2 \tag{7}$$

where z_{ik} is a hidden latent variable that is always non-negative. It has been shown in [9, 25] that this is, in fact, the best quadratic majorizer of (4). Taking the derivative of (7) w.r.t z_{ik} and setting it to zero to get the minimum, *i.e.* $\frac{\partial g_H^{(i)}(\mathbf{w}_k, b_k, z_{ik})}{\partial z_{ik}} = 0$, we obtain

$$z_{ik} = \left| 1 - y_{ik} \left(\mathbf{w}_k^T \mathbf{x}^{(i)} + b_k \right) \right|. \tag{8}$$

Note that (7) has a singularity at $z_{ik} = 0$. So we simply fix this problem by bounding z_{ik} from below by a small positive constant ϵ such that

$$z_{ik} = \max \left[\epsilon, \left| 1 - y_{ik} \left(\mathbf{w}_k^T \mathbf{x}^{(i)} + b_k \right) \right| \right]. \tag{9}$$

Letting z_{ik}^* denote the value of z_{ik} when (7) reaches its minimum, the new loss function can be rewritten as

$$L_M \left(\mathbf{w}_k, b_k, \{ z_{ik}^* \} \right) = \frac{1}{2} \mathbf{w}_k^T \mathbf{w}_k + \lambda_H \sum_{i \in \mathbf{s}_L} g_H^{(i)} \left(\mathbf{w}_k, b_k, z_{ik}^* \right) + \lambda_S f_S \left(\mathbf{w}_k, b_k \right). \tag{10}$$

Now the hinge loss has been converted to a sequence of weighted least-squares problems. Let $\widetilde{\mathbf{w}}_k = \left(b_k, \mathbf{w}_k^T \right)^T$ and $\widetilde{\mathbf{x}}^{(i)} = \left(1, \left(\mathbf{x}^{(i)} \right)^T \right)^T$, so that $\mathbf{w}_k^T \mathbf{x}^{(i)} + b_k$ and $\mathbf{w}_k^T \mathbf{w}_k$ can be compactly written as $\widetilde{\mathbf{w}}_k^T \widetilde{\mathbf{x}}^{(i)}$ and $(\mathbf{R}\widetilde{\mathbf{w}}_k)^T (\mathbf{R}\widetilde{\mathbf{w}}_k)$, where $\mathbf{R}$ is a $N \times N$ diagonal matrix with 1 on its diagonal except the first element being 0. Plugging $\widetilde{\mathbf{w}}_k$ and $\mathbf{R}$ back into (10) and taking the gradient w.r.t. $\widetilde{\mathbf{w}}_k$, we obtain

$$\begin{aligned} \frac{\partial L_M}{\partial \widetilde{\mathbf{w}}_k} = \mathbf{R}\widetilde{\mathbf{w}}_k - \lambda_H \sum_{i \in \mathbf{s}_L} \frac{1}{2 z_{ik}^*} \left(1 - y_{ik} \widetilde{\mathbf{w}}_k^T \widetilde{\mathbf{x}}^{(i)} + z_{ik}^* \right)^T y_{ik} \widetilde{\mathbf{x}}^{(i)} \\ + 2\lambda_S \sum_{i,j \in \mathbf{s}_L \cup \mathbf{s}_U} W_{ij} \mathbf{C}^{(ij)} \widetilde{\mathbf{w}}_k \end{aligned} \tag{11}$$

where $\mathbf{C}^{(ij)} = \left(\frac{\widetilde{\mathbf{x}}^{(i)}}{\sqrt{D_{ii}}} - \frac{\widetilde{\mathbf{x}}^{(j)}}{\sqrt{D_{jj}}} \right) \left(\frac{\widetilde{\mathbf{x}}^{(i)}}{\sqrt{D_{ii}}} - \frac{\widetilde{\mathbf{x}}^{(j)}}{\sqrt{D_{jj}}} \right)^T$ is a symmetric matrix as can be seen from the expression in (11) above. After substituting each minimizer z_{ik}^*, (11) can be set to zero to solve for $\widetilde{\mathbf{w}}_k$ at every iteration. Assuming $x_p^{(i)}$ is the value at the p^{th} dimension of the vector $\widetilde{\mathbf{x}}^{(i)}$, we get a set of linear equations in the form of $\mathbf{A}\widetilde{\mathbf{w}}_k = \mathbf{d}$, where the elements in matrix $\mathbf{A}$ and vector $\mathbf{d}$ are defined as

$$\mathbf{A}_{pq} \equiv \mathbf{R}_{pq} + \lambda_H \sum_{i \in \mathbf{s}_L} \frac{1}{2 z_{ik}^*} x_p^{(i)} x_q^{(i)} + 2\lambda_S \sum_{i,j \in \mathbf{s}_L \cup \mathbf{s}_U} W_{ij} \mathbf{C}_{pq}^{(ij)}. \tag{12}$$

and

$$\mathbf{d}_p \equiv \sum_{i \in \mathbf{s}_L} \frac{1 + z_{ik}^*}{2 z_{ik}^*} y_{ik} x_p^{(i)}. \tag{13}$$

Thus, a sequence of updates can be obtained by alternating between weighted least-squares solutions for $\widetilde{\mathbf{w}}_k$ and all z_{ik}^*, until the ℓ_2 norm of the difference of $\widetilde{\mathbf{w}}_k$ between two sequential iterations is less than a small constant δ or the maximum number of iterations is reached. This whole process of alternating updates bears sharp resemblance to the EM algorithm ([24]). The step in (7) is similar to the E (expectation) step of the EM algorithm where the missing data (z_{ik} in our case) is filled in by using a surrogate function. And the step in (11) is akin to the M (maximization) step of the EM algorithm wherein a much simpler surrogate function is optimized. And similar to EM, our method therefore incorporates all the advantages of the EM algorithm in that it is

numerically stable, and avoids wildly overshoots or undershoots of the maximum likelihood of the data along its current direction of search for the optimum parameters ([24]): this is due to the fact that line search parameters need not be estimated at each step.

After repeating this one-versus-rest method for all K classes, finally the i^{th} superpixel in $\mathbf{s}_U$ is assigned to the category which corresponds to the maximum score, *i.e.*

$$l_i = \arg \max_{k=1,2,\ldots,K} \left\{ \widetilde{\mathbf{w}}_k^T \widetilde{\mathbf{x}}^{(i)} \right\} \tag{14}$$