AUTOMATIC SELECTION OF INFORMATIVE SAMPLES FOR SVM-BASED CLASSIFICATION OF HYPERSPECTRAL DATA USING LIMITED TRAINING SETS

Antonio Plaza and Javier Plaza

Department of Technology of Computers and Communications
University of Extremadura, Avda. de la Universidad s/n, E-10071 Cáceres, SPAIN
E-mail: {aplaza, jplaza}@unex.es

ABSTRACT

In this work, we focus on how to select the most highly informative samples for effectively training support vector machine (SVM) classifiers in remotely sensed hyperspectral data classification. This issue is investigated by comparing different unsupervised algorithms which account for the spectral purity of training samples in the process of selecting those samples for classification purposes. Sample sets obtained using these algorithms are used to train an SVM architecture implemented using different kernels, with the ultimate goal of exploring the suitability of the aforementioned algorithms to reduce the number of training samples required by these architectures in the context of hyperspectral image classification. Experimental results are provided using the full version of a hyperspectral data set collected by NASA's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) over the Indian Pines region in Northwestern Indiana.

Index Terms— Machine learning, hyperspectral image classification, support vector machines (SVMs), automatic selection of training samples

1. INTRODUCTION

Hyperspectral imaging is concerned with the measurement, analysis, and interpretation of spectra acquired from a given scene (or specific object) at a short, medium or long distance by an airborne or satellite sensor [1]. The concept of hyperspectral imaging originated at NASA's Jet Propulsion Laboratory in California with the development of the Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS), able to cover the wavelength region from 0.4 to 2.5 μm using more than two hundred spectral channels, at nominal spectral resolution of 10 nm [2]. As a result, each pixel vector collected by a hyperspectral instrument can be seen as a *spectral signature* or *fingerprint* of the underlying materials within the pixel.

Supervised classification is one of the most commonly undertaken analyses of remotely sensed hyperspectral data [3]. The output of a supervised classification is effectively a thematic map that provides a snapshot representation of the spatial distribution of a particular theme of interest such as land cover. Recent research has indicated the considerable potential of SVM-based approaches for the supervised classification of remotely sensed hyperspectral data [4, 5, 6]. Comparative studies have shown that classification by a SVM can be more accurate than techniques such as neural networks, decision trees and probabilistic classifiers such as maximum likelihood classification [7]. SVMs were designed for binary classification but various methods exist to extend the binary approach to multiclass classification, such as the *one versus the rest* and the *one versus one* stategies.

The SVM was first investigated as a binary classifier. Given a training set $S = \{(\Phi(\mathbf{x})_i, y_i) \mid i \in [1, n]\}$ projected into a Hilbert space $\mathcal{H}$ by some mapping Φ, the SVM separates the data by an optimal hyperplane H_p that maximizes the margin. Allowing some training errors, H_p is found by jointly maximizing the margin $\|\mathbf{w}\|$ and minimizing the sum of errors $\sum_{i=1}^{n} \xi_i$. Through the use of a kernel function, k, it is possible to compute implicitly the inner product in $\mathcal{H}$ in the original space: $\langle \Phi(\mathbf{x})_i, \Phi(\mathbf{x})_j \rangle_{\mathcal{H}} = k(\mathbf{x}_i, \mathbf{x}_j)$. SVM used with a kernel function is a non-linear classifier, where the non-linear ability is included in the kernel. The use of different kernel functions leads to different SVM configurations. The most used kernels are the polynomial kernel, $k_{poly}(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + \theta)^d$ and the Gaussian kernel, $k_{gauss}(\mathbf{x}, \mathbf{z}) = \exp\left(-\gamma \|\mathbf{x} - \mathbf{z}\|^2\right)$. When some *a-priori* are known, it is possible to include them into the kernel, to improve the classification. A spectral angle distance (SAD)-based kernel was recently introduced as follows: $k_{SAD}(\mathbf{x}, \mathbf{z}) = \exp\left(-\gamma \alpha(\mathbf{x}, \mathbf{z})^2\right)$, where $\alpha(\mathbf{x}, \mathbf{z}) = \arccos\left(\frac{\langle \mathbf{x}, \mathbf{z} \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{z}\|}\right)$.

The SVM classification process is based on the notion of fitting an optimal separating hyperplane between classes by focusing on the training samples that lie at the edge of the class distributions, the support vectors. All of the other training samples are effectively discarded as they do not contribute to the estimation of hyperplane location. As a result, the intelligent selection of training samples has the potential to improve the performance of SVMs (in particular, when limited training samples are available) since only the most informative training samples are used by the SVM architecture [8].

978-1-4244-8907-7/10/$26.00 ©2010 IEEE

In this paper, we address the issue of how to select the most highly informative samples for SVM classification. The remainder of the paper is organized as follows. Section 2 describes two automatic algorithms for intelligent selection of training samples which are aimed at reducing the number of training samples required to produce an accurate classification result. Section 3 provides experimental results with the full AVIRIS Indian Pines data set. Finally, section 4 concludes the paper with some remarks.

2. TRAINING SAMPLE SELECTION ALGORITHMS

Traditionally, training samples have been selected randomly from available labeled samples. However, it has been shown in previous work that not only the size but also the composition of the training set has a significant impact on the final classification results. In particular, a challenging aspect in the design of SVM classifiers for hyperspectral imagery is to reduce the need for large training sets. In this work, we hypothesize that effective supervised classifiers demands intelligent training sample selection algorithms able to seek for the most informative training samples, thus optimizing the compromise between estimation accuracy (to be maximized) and ground-truth knowledge (to be minimized). Since training samples can be either pure or mixed in nature, we develop two unsupervised algorithms which account for the spectral purity of training samples in the process of selecting those samples for training purposes. Training sets obtained using these algorithms are compared to randomly selected sets in the classification experiments conducted in section 3.

2.1. Selection of pure training samples

In order to extract the purest pixel vectors in the data set as training samples, we use a modified version of the pixel purity index (PPI) algorithm available commercially from Research Systems ENVI [9]. This algorithm automatically searches for the purest spectral signatures which are assumed to be the vertices of a convex hull. The algorithm proceeds by generating a large number of random, N-dimensional unit vectors called 'skewers' through the dataset. Every data point is projected onto each skewer, and the data points that correspond to extrema in the direction of a skewer are identified and placed on a list. As more skewers are generated, the list grows, and the number of times a given pixel is placed on this list is also tallied. The pixels with the highest tallies are considered the final endmembers.

The inputs to the algorithm are a hyperspectral data cube $\boldsymbol{f}$ with N spectral bands, the number of training samples to be extracted, *t*, and the number of random skewers to be generated during the process, *k*. The output of the algorithm is a set of *t* training samples $\{\mathbf{t}_i\}_{i=1}^{t}$. The algorithm can be summarized by the following steps:

1. *Skewer generation*. Produce a set of *k* randomly generated unit vectors $\{\textbf{skewer}_j\}_{j=1}^{k}$.

2. *Extreme projections*. For each $\textbf{skewer}_j$, all sample pixel vectors $\boldsymbol{f}$(x,y) in the original data set, where (x, y) denotes the spatial coordinates, are projected onto $\textbf{skewer}_j$ via dot products of $|\boldsymbol{f}(x, y) \cdot \textbf{skewer}_j|$ to find sample vectors at its extreme (maximum and minimum) projections, thus forming an extrema set for $\textbf{skewer}_j$ which is denoted by $S_{extrema}(\textbf{skewer}_j)$. Despite the fact that a different $\textbf{skewer}_j$ would generate a different extrema set $S_{extrema}(\textbf{skewer}_j)$, it is very likely that some sample vectors may appear in more than one extrema set. In order to deal with this situation, we define an indicator function of a set *S*, denoted by $I_S(\mathbf{x})$, to denote membership of a vector element $\mathbf{x}$ to that particular set as follows:

$$I_S(\mathbf{x}) = \begin{Bmatrix} 1 \text{ if } \mathbf{x} \in S \\ 0 \text{ if } \mathbf{x} \notin S \end{Bmatrix} \quad (1)$$

3. *Calculation of PPI scores*. Using the indicator function above, we calculate the PPI score associated to each sample pixel vector $\boldsymbol{f}(x, y)$ in the input scene (i.e., the number of times that given pixel has been selected as extreme in step (2) using the following equation:

$$\text{PPI}(x, y) = \sum_{j=1}^{k} I_{S_{extrema}(\textbf{skewer}_j)}(\boldsymbol{f}(x, y)) \quad (2)$$

4. *Selection of training samples*. Find the t pixel vectors with the highest scores of PPI(x, y) and use them to form a training set $\{\mathbf{t}_i\}_{i=1}^{t}$.

2.2. Selection of border training samples

As an alternative to the algorithm developed in the previous subsection, we describe below an automatic algorithm that iteratively seeks for border training samples. The separation of a training set into border and non-border patterns was first explored by Foody [10], who expressed *borderness* as the difference between the two smallest distances measured for each training pattern. Here, membership was indicated by the Mahalanobis distance, which provides a measure of the typicality of a pattern to a certain class. A border-training pattern is expected to be almost as close to its actual class of membership as it is to any other class. Therefore, the difference in the Mahalanobis distances between the two most likely classes of membership would be small for a border pattern. This focus on the vicinity of the hyperplanes that can optimally separate the classes is similar to aspects of Lee and Landgrebe's decision boundary feature extraction [11]. Here, we develop an automatic algorithm inspired by the concept proposed by Foody, but further adapted to a mixed pixel interpretation scenario. The concept implemented by this algorithm can actually be viewed as the opposite to that used by

convex geometry-based endmember extraction methods such as PPI or Winter's N-FINDR algorithm [12]. Since the PPI does not produce a set of spectrally distinct endmembers but a set of pure pixels which are actually endmember candidates that may contain several instances of the same endmember, we use the N-FINDR for initialization of our algorithm (allowing N-FINDR estimate how many endmembers are in the input hyperspectral data). The border training sample selection algorithm can be summarized as follows:

1. Label the set of spectral endmembers $\{\mathbf{e}_i\}_{i=1}^{p}$ produced by the N-FINDR algorithm as class *core* patterns.

2. Apply a spectral screening algorithm to identify the sample pixel vectors within a small spectral angle θ from any of the p core classes above, denoted by $\{\mathbf{r}_j\}_{j=1}^{q}$ with $q \geq p$.

3. Associate each signature of the set $\{\mathbf{r}_j\}_{j=1}^{q}$ to one of the available pure classes $\{\mathbf{e}_i\}_{i=1}^{p}$ by computing $\mathbf{r}_j^{(i)} = \text{argmin}_j\{\text{SAD}(\mathbf{r}_j, \mathbf{e}_i)\}$ for all $j = 1, 2, \cdots, q$, where the notation of $\mathbf{r}_j^{(i)}$ indicates that the SAD between $\mathbf{r}_j$ and $\mathbf{e}_i$ is the minimum, i.e., $\mathbf{e}_i$ is the most spectrally similar endmember to $\mathbf{r}_j$.

4. Let $\mathbf{r}_{j,k}^{(i)} \subseteq \{\mathbf{r}_j\}_{j=1}^{q}$ be the k-th sample associated with class $\mathbf{e}_i$, and let $|\mathbf{r}_{j,k}^{(i)}|$ be the cardinality of the set $\{\mathbf{r}_{j,k}^{(i)}\}$, composed of the samples in $\{\mathbf{r}_j\}_{j=1}^{q}$ associated with $\mathbf{e}_i$.

5. For each sample pixel vector $\boldsymbol{f}(x, y)$, compute the Mahalanobis distance from each pure class $\mathbf{e}_i$ as follows: $\text{MD}(\boldsymbol{f}(x, y), \mathbf{e}_i) = (\boldsymbol{f}(x, y) - \mu_i)^T \mathbf{K}_i^{-1} (\boldsymbol{f}(x, y) - \mu_i)$, where $\mathbf{K}_i$ is the sample covariance matrix of the class given by $\mathbf{e}_i$, and μ_i is the mean for that class, given by $\mu_i = \frac{1}{|\mathbf{r}_{j,k}^{(i)}|} \sum_{k=1}^{|\mathbf{r}_{j,k}^{(i)}|} \mathbf{r}_{j,k}^{(i)}$.

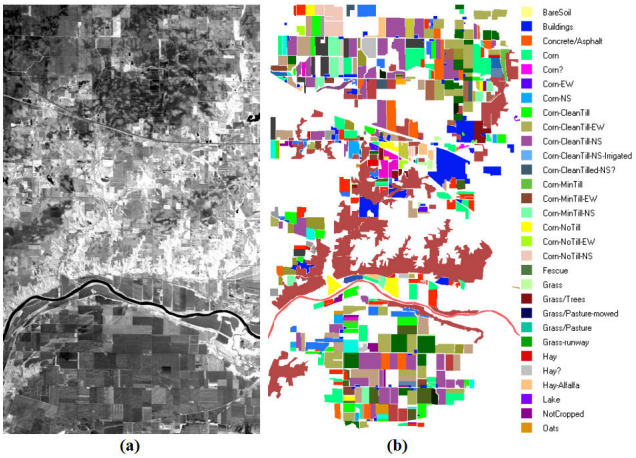
6. Compute a borderness score $\text{BS}(x, y)$ for each pixel vector $\boldsymbol{f}(x, y)$ by finding the two most likely classes of membership for the pixel, say, $\mathbf{e}_i$ and $\mathbf{e}_j$ with $1 \leq i \leq p$ and $1 \leq j \leq p$, and then calculating:

$$\text{BS}(x, y) = |\text{MD}(\boldsymbol{f}(x, y), \mathbf{e}_i) - \text{MD}(\boldsymbol{f}(x, y), \mathbf{e}_j)|. \quad (3)$$

7. Find the t pixel vectors with the lowest scores of $\text{BS}(x, y)$ and use them to form a training set $\{\mathbf{t}_i\}_{i=1}^{t}$.

3. EXPERIMENTS

The hyperspectral scene used in experiments was gathered by AVIRIS over the Indian Pines test site in Northwestern Indiana, a mixed agricultural/forested area, early in the growing season, and consists of 1939×677 pixels and 204 spectral bands in the wavelength range 0.4–2.5 μm (523 MB in size). We emphasize that the image that we used in experiments is different to the commonly used one with dimensions



(a) (b)

Fig. 1. (a) Spectral band at 587 nm of a portion of the full 1939×677 pixels Indian Pines scene used in experiments. (b) Ground-truth map.

145×145 pixels in size, available from `http://cobweb.ecn.purdue.edu/~biehl/MultiSpec/`. Our reason to use this scene is that we anticipated a more challenging classification problem in the larger scene due to the presence of a higher number of classes, but the presence of large homogeneous classes with well-defined spectral properties allowed us to increase the overall accuracies with regards to those observed in the 145×145 pixels scene resulting from the accurate modelling of such classes. A total of 20 bands were removed from the scene prior to analysis due to low SNR in those bands. For illustrative purposes, Fig. 1(a) shows the spectral band at 587 nm of the original scene and Fig. 1(b) shows the corresponding ground-truth map, displayed in the form of a class assignment for each labeled pixel.

Three types of kernels were used in experiments: polynomial (k_{poly}), Gaussian (k_{gauss}), and SAD-based (k_{SAD}). Small training sets, composed of 1%, 2%, 4%, 6%, 8%, and 10% of the ground-truth pixels available per class, were extracted using the two training sample selection algorithms described in section 2, and also using a random selection procedure. The SVMs were trained with each of these training subsets and then evaluated with the remaining test set. Each experiment was repeated five times, and the mean accuracy values were reported. In all cases, spatial post-processing based on class majority voting in a sliding window was applied to increase classification accuracies.

Table 1 summarizes the overall classification results obtained using the three considered kernels and training sample selection algorithms. From Table 1, it can be seen that SVMs generalize quite well: with few training pixels per class, high classification accuracy is reached by all kernels. It should be noted that the classification results reported in Table 1 for a portion of the 1939×677 pixel scene are higher than those

Table 1. Overall classification accuracies (percentage) after applying polynomial, Gaussian and SAD-based kernels to a portion of the 1939×677 pixel Indian Pines scene, using different training algorithms (spatial post-processing is also applied).

Training set size (%)		1%	2%	4%	6%	8%	10%
k_{poly}	Random training samples	82.33	82.94	83.21	83.82	85.34	86.12
	Pure training samples	81.23	82.06	82.80	83.00	84.03	84.45
	Border training samples	83.44	84.23	84.45	84.96	86.27	87.44
k_{gauss}	Random training samples	87.94	88.23	88.78	88.96	89.45	89.48
	Pure training samples	86.53	87.02	87.64	87.93	88.12	88.26
	Border training samples	89.45	90.25	91.24	92.08	92.93	93.04
k_{SAD}	Random training samples	85.90	86.22	86.49	87.03	87.56	88.09
	Pure training samples	85.12	85.67	86.08	86.45	86.97	87.13
	Border training samples	86.05	86.93	87.57	88.12	89.30	90.12

reported in the literature for a smaller 145×145 pixel scene. This is mainly due to the presence of larger homogeneous classes in the considered portion. These large classes are easy to classify and, combined with spatial post-processing, increase the overall accuracy significantly. In all cases, the overall classification accuracies decreased when random and pure samples were used for the training sate. This confirms the fact that kernel-based methods in general and SVMs in particular are less affected by the Hughes phenomenon. It is also clear from Table 1 that the classification accuracy is generally correlated with the training set size. However, when border training samples were used, higher classification accuracies were achieved with less training samples. The above results indicate the importance of including mixed pixels at the border of class boundaries in the training set, as these border patterns are most efficient to determine the hyperplane between two classes. Finally, it can be seen in Table 1 that the best classification scores were generally achieved for the k_{gauss} kernel. On the other hand, the k_{SAD} kernel gives slightly degraded classification results. Finally, the k_{poly} kernel needs more training samples than the two other kernels to perform appropriately, as can be seen from the relatively poor results obtained by this kernel for limited training samples.

4. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we have focused on the issue of how to select the most highly informative samples for effectively training support vector machine (SVM) classifiers in remotely sensed hyperspectral data classification. Specifically, *border* training samples are shown to be more useful than *core* training samples in order to increase the classification accuracies that can be achieved using SVM architectures with the full version (1939×677 pixels) of an AVIRIS hyperspectral image collected over the Indian Pines region in NW Indiana. Further experiments should be conducted with additional scenes and training algorithms to fully substantiate these remarks.

5. REFERENCES

[1] A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, "Imaging spectrometry for earth remote sensing," *Science*, vol. 228, pp. 1147–1153, 1985.

[2] R. O. Green, "Imaging spectroscopy and the airborne visible-infrared imaging spectrometer (AVIRIS)," *Remote Sensing of Environment*, vol. 65, pp. 227–248, 1998.

[3] P. Gamba, F. Dell'Acqua, A. Ferrari, J. A. Palmason, and J. A. Benediktsson, "Exploiting spectral and spatial information in hyperspectral urban data with high resolution," *IEEE Geoscience and Remote Sensing Letters*, vol. 1, pp. 322–326, 2004.

[4] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, pp. 93–97, 2006.

[5] M. Fauvel, J.A. Benediktsson, J. Chanussot, and J.R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geoscience and Remote Sensing*, vol. 46, pp. 3804–3814, 2008.

[6] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive svm for the semisupervised classification of remote sensing images," *IEEE Trans. Geoscience and Remote Sensing*, vol. 44, pp. 3363–3373, 2006.

[7] Y. Tarabalka, J.A. Benediktsson, and J. Chanussot, "Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques," *IEEE Trans. Geoscience and Remote Sensing*, vol. 47, pp. 2973–2987, 2009.

[8] J. Plaza, A. Plaza, and C. Barra, "Multi-channel morphological profiles for classification of hyperspectral image data using support vector machines," *Sensors*, vol. 9, no. 1, 2009.

[9] J. W. Boardman, "Automating spectral unmixing of aviris data using convex geometry concepts," in *Summaries of Airborne Earth Science Workshop*, R. O. Green, Ed., 1993, vol. 93-26 of *JPL Publication*, pp. 111–114.

[10] G. M. Foody and A. Mathur, "Toward intelligent training of supervised image classifications: directing training data acquisition for svm classification," *Remote Sensing of Environment*, vol. 93, pp. 107–117, 2004.

[11] C. Lee and D. A. Landgrebe, "Decision boundary feature extraction for neural networks," *IEEE Trans. Neural Networks*, vol. 8, pp. 75–83, 1997.

[12] M. E. Winter, "Algorithm for fast autonomous spectral endmember determination in hyperspectral data," in *Imaging Spectrometry V*, M. R. Descour and S. S. Shen, Eds., 1999, vol. 3753 of *Proceedings of SPIE*, pp. 266–275.