# Exploring Synergetic Effects of Dimensionality Reduction and Resampling Tools on Hyperspectral Imagery Data Classification

J. S. Sánchez, V. García, and R. A. Mollineda

Institute of New Imaging Technologies
Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I
Av. Vicent Sos Baynat, s/n
12071 Castellón de la Plana, Spain
{sanchez,jimenezv,mollined}@uji.es

**Abstract.** The present paper addresses the problem of the classification of hyperspectral images with multiple imbalanced classes and very high dimensionality. Class imbalance is handled by resampling the data set, whereas PCA and a supervised filter are applied to reduce the number of spectral bands. This is a preliminary study that pursues to investigate the benefits of combining several techniques to tackle the imbalance and the high dimensionality problems, and also to evaluate the order of application that leads to the best classification performance. Experimental results demonstrate the significance of using together these two preprocessing tools to improve the performance of hyperspectral imagery classification. Although it seems that the most effective order corresponds to first a resampling strategy and then a feature (or extraction) selection algorithm, this is a question that still needs a much more thorough investigation in the future.

## 1 Introduction

Hyperspectral sensors are characterized by a very high spectral resolution that usually results in hundreds of observation channels [27]. Although this allows to address many applications requiring very high discrimination capabilities in the spectral domain [4], the huge amount of data available makes complex the classification of hyperspectral images. In this classification context, another important drawback is that the hyperspectral information is commonly represented by a very large number of features (spectral bands), which are usually highly correlated [27, 31].

A complex situation frequently ignored in hyperspectral imaging refers to the presence of severely skewed class priors. This situation is generally known as the class imbalance problem [13]. A data set is said to be imbalanced when one of the classes (the minority one) is heavily under-represented in comparison to the other (the majority) class. Because of samples of the minority and majority classes usually represent the presence and absence of rare cases respectively, they are also known as positive and negative examples. It has been observed that class imbalance often leads to poor classification performance in many real-world applications, especially for the minority classes.

Most of the approaches to tackle the imbalance problem have been proposed both at the data and algorithmic levels. Data-driven methods consist of balancing the original data set, either by over-sampling the minority class [5,12] and/or by under-sampling [9, 21] the majority class until the classes are approximately equally represented. Belonging to this group, we can also find several algorithms for feature selection [1, 18, 23, 33,35]. At the algorithmic level, solutions include internally biasing the discrimination-based process [7, 8] and assigning distinct costs to the classification errors [24, 25, 39].

Although class imbalance has been extensively studied for binary classification problems in the last decades, very few approaches deal with imbalanced multi-class data sets, as is the case of remote sensing applications. In the particular context of hyperspectral imagery, some proposals are simple adjustments of conventional learning algorithms [3,20,37], whereas others employ ensembles of classifiers [32,36] or feature selection tools [6].

This paper investigates some strategies to select the most relevant features and manage the class imbalance in the classification of hyperspectral imagery acquired by the Airborne Visible/Infrared Imaging Spectrometrer (AVIRIS[1]). The problem is of great importance since these image data present both very high dimensionality and multiple imbalanced classes, what certainly provides additional challenges in the framework of remote sensing classification. In order to face such a problem, this work focuses on the joint use of feature selection/extraction and resampling techniques, and explores the order in which they should be applied to achieve the best classification results.

The rest of the paper is organized as follows. Section 2 describes the methodology proposed to handle both class imbalance and high dimensionality, and also briefly reviews the fundamentals of the classifiers used in this work. Next, Sect. 3 contains a description of the real hyperspectral image database used in this paper and defines the configuration of the experiments carried out. Section 4 provides the results and discusses the most important findings. Finally, Sect. 5 concludes the present study and outlines possible avenues for future research.

## 2 Methodology

This section provides an overview of the method here proposed to classify remote sensing data according to the two issues of interest previously pointed out. In a first stage, the hyperspectral image data set will be preprocessed with the double aim of balancing the skewed classes and reducing the number of features/bands, albeit not necessarily in this order. The second stage will consist of classifying the resulting set after overcoming those two problems. Note that only those algorithms that will be further used in the experiments are described in the present section.

### 2.1 Data Preprocessing

Taking the particular characteristics of hyperspectral data sets into account, most imaging tasks could usually benefit from the application of some preprocessing techniques.

[1] http://aviris.jpl.nasa.gov/

Here we concentrate on a common situtation in which the data set consists of multiple imbalanced classes in a high dimensional representation space.

**Balancing the Classes.** Data level methods for balancing the classes consists of resampling the original data set, either by over-sampling the minority class or by under-sampling the majority class, until the classes are approximately equally represented. Both strategies can be applied in any learning system since they act as a preprocessing phase, thus allowing the system to receive the training instances as if they belonged to a well-balanced data set. By using this strategy, any bias of the learning system towards the majority class due to the skewed class priors will hopefully be eliminated.

The simplest method to increase the size of the minority class corresponds to random over-sampling, that is, a non-heuristic method that balances the class distribution through the random replication of positive examples. Nevertheless, since this method replicates existing examples in the minority class, overfitting is more likely to occur. Chawla et al. [5] proposed an over-sampling technique that generates new synthetic minority samples by interpolating between several preexisting positive examples that lie close together. This method, called SMOTE (Synthetic Minority Over-sampling TEchnique), allows the classifier to build larger decision regions that contain nearby samples from the minority class.

On the other hand, random under-sampling [15, 38] aims at balancing the data set through the random removal of negative examples. Despite its simplicity, it has empirically been shown to be one of the most effective resampling methods. Unlike the random approach, many other proposals are based on a more intelligent selection of the negative examples to be eliminated. For instance, the one-sided selection technique [21] selectively removes only those negative samples that either are redundant or that border the minority class examples (assuming that these bordering cases are noise).

**Dimensionality Reduction.** The reduction in the hyperspectral representation space can be carried out by means of feature selection or extraction techniques [14, 26]. In both approaches, the aim is to reduce the number of bands, without much loss of information. The process of feature selection is to choose a representative subset of features from the original data by assessing its discrimination capabilities according to statistical distance measures among classes (e.g., Bhattacharyya distance, Jeffries-Matusita distance, and the transformed divergence measure). The feature extraction approach addresses the problem of dimensionality reduction by projecting the data from the original feature space onto a low-dimensional subspace, which contains most of the original information.

Probably, one of the most well-known feature extraction methods corresponds to PCA (Principal Component Analysis) [17], which seeks to reduce the dimension of the data by finding a few mutually orthogonal linear combinations of the original variables with the largest variance. It involves a mathematical procedure that transforms a number of (possibly) correlated observed variables into a smaller number of uncorrelated artificial variables called principal components. The principal components extracted in PCA are the eigenvectors of the data coveriance matrix, where the first principal component is the eigenvector with the largest eigenvalue (the one that accounts for a maximal amount of total variance in the observed variables). The remaining components will account for a maximal amount of variance in the observed variables that was not accounted for by the preceding components, and will be uncorrelated with all of the previously extracted components.

In the case of feature selection, the algorithm here used corresponds to a supervised feature filter, namely correlation-based feature selection (CFS) [11]. The central hypothesis of this technique is that good feature sets should contain variables that are highly correlated with the class, yet uncorrelated with each other. Thus irrelevant features should be ignored because they will have low correlation with the class, whereas redundant features should be removed as they will be highly correlated with one or more of the remaining variables. The acceptance of a feature will depend on the extent to which it predicts classes in areas of the input space not already predicted by other features. The implementation of CFS allows the user to choose from three heuristic search strategies: forward selection, backward elimination, and best first (this may start with no features and search forward, or with the full set of features and search backward, or even start at any point and search in both directions).

### 2.2 Classification

We assume that there exists a set of $n$ previously labeled examples (training set), say $X = \{(x_1, \omega_1), (x_2, \omega_2), \ldots, (x_n, \omega_n)\}$, where each element has an attribute vector $x_i$ and a class label $\omega_i$. Two traditional classification techniques for remote sensing data will be used in the present experimental study: a support vector machine and a decision tree.

**Support Vector Machine.** Support vector machine (SVM) models [34] were originally proposed for the classification of linearly separable classes of samples, based on the idea of the empirical risk minimization principle. For any particular set of two-class data, an SVM finds the unique hyperplane (one per class) having the maximum margin $\delta$. The samples that define the hyperplanes are called support vectors. A special characteristic of the SVM is that the solution to a classification problem is represented by the support vectors that determine the hyperplane with the maximum margin.

For nonlinear problems, the SVM maps the input data from the original input space onto a higher dimensional feature space using a kernel function. Formally, an SVM defines a new space where data are linearly separable. For such a purpose, it constructs an optimal hyperplane (or a set of hyperplanes) in a high or infinite dimensional space, which can be used for classification, regression or other tasks. The optimal hyperplane corresponds to a decision surface that maximizes the distance between it and the nearest training samples of any class (this largest distance is the maximum margin). In general, the larger the margin, the lower generalization error of the classifier.

The use of nonlinear kernels provides the SVM with the ability to model complex separation hyperplanes. However, because there is no theoretical tool to predict which kernel will give the best results for a given data set, experimenting with different types (e.g., polynomial, B-spline, radial basis function, sigmoid, tensor product) or using prior knowledge [19] are the only ways to identify the best kernel function.

**Decision Tree.** A decision tree is a classification (or regression) tool [2] that uses a tree-like graph or model of decisions and their consequences. Thus a decision-tree model is built by analyzing training data and the result is then used to classify unseen data. The internal nodes of a tree evaluate the existence or significance of individual attributes. Following a path from the root to the leaves of the tree, a sequence of such tests is performed resulting in a decision about the appropriate class of new objects.

The decision trees are usually constructed in a top-down fashion by choosing the most appropriate attribute each time. An information-theoretic measure (e.g., entropy, Gini impurityindex, information gain, Chi-square) is used to evaluate features, which provides an indication of the ”classification power” of each attribute. Once a feature is chosen, the training data are divided into subsets, corresponding to different values of the selected feature, and the process is repeated for each subset until a large proportion of the instances in each subset belongs to a single class.

Popularity of decision trees comes as a result of flexibility, easy interpretability and simple implementation. Thus many decision-tree algorithms have been developed, being ID3 [29], C4.5 [30], and CART [2] some of the most extensively-used.

## 3 Experimental Set-up

The experiments were carried out on the 92AV3C data set[2], which corresponds to a hyperspectral image ($145 \times 145$ pixels) taken over Northwestern Indiana's Indian Pines by the AVIRIS sensor in June 1992 and employed to recognize different land-cover classes. Although the AVIRIS sensor collects 224 spectral bands, four of these contain only zero values and so they can be removed, leaving 220 non-zero bands. On the other hand, several bands should also be discarded due to the effect of atmospheric absorption and/or noise [16, 22, 26, 27], thus giving a total of 185 bands to be considered in the present study. The ground truth data show that the image has 17 classes, although only 16 classes belonging to different crop types, vegetation, man-made structures or other kinds of land were used (see Table 1). The omitted class contains unlabeled pixels, which presumably correspond to uninteresting regions or were too difficult to label.

In order to increase the statistical significance of the experimental results, several classification performance measures were averaged over 30 different random partitions (2/3 of pixels for training and the rest for testing) of the original data set, preserving the prior class probabilities of each and the statistical independence between the training and test sets of every partition. The training sets were preprocessed by SMOTE and random under-sampling (RUS) to handle the class imbalance, and also by PCA (with a variance of 0.99) and the CFS algorithm for dimensionality reduction. Since it is difficult to decide which classes to resample in a multi-class problem (e.g., class $\sigma_2$ can be deemed as majoritary when compared to class $\sigma_5$, but minoritary with respect to class $\sigma_8$), we divided the biggest class (Soybeans–min, $\sigma_{14}$) into four blocks (each one with 25% of samples). Based on this, the remaining classes were over-sampled to reach 50% and 75% the size of the majority class because they represent non-extreme cases. Similarly, the under-sampling strategy was applied by removing 50% and 75% of samples according to the size of the biggest class.

[2] https://engineering.purdue.edu/ biehl/MultiSpec/hyperspectral.html

**Table 1.** Number of training and test pixels per class, along with the relative percentage of samples belonging to each class

| Class ($\sigma_i$) | Training | Test | % |
|---|---|---|---|
| $\sigma_1$: Stone–steel towers | 63 | 32 | 0.92 |
| $\sigma_2$: Hay–windrowed | 326 | 163 | 4.72 |
| $\sigma_3$: Corn–min | 556 | 278 | 8.05 |
| $\sigma_4$: Soybeans–notill | 645 | 323 | 9.34 |
| $\sigma_5$: Alfalfa | 36 | 18 | 0.52 |
| $\sigma_6$: Soybeans–clean | 409 | 205 | 5.92 |
| $\sigma_7$: Grass/Pasture | 331 | 166 | 4.79 |
| $\sigma_8$: Woods | 863 | 431 | 12.48 |
| $\sigma_9$: Bldg–Grass–Trees—Drives | 253 | 127 | 3.67 |
| $\sigma_{10}$: Grass/pasture–mowed | 17 | 9 | 0.25 |
| $\sigma_{11}$: Corn | 156 | 78 | 2.26 |
| $\sigma_{12}$: Oats | 13 | 7 | 0.19 |
| $\sigma_{13}$: Corn–notill | 956 | 478 | 13.83 |
| $\sigma_{14}$: Soybeans–min | 1645 | 823 | 23.81 |
| $\sigma_{15}$: Grass/Trees | 498 | 249 | 7.21 |
| $\sigma_{16}$: Wheat | 141 | 71 | 2.05 |

The J48 decision tree (an open source Java implementation of the very popular C4.5 algorithm) and a SVM [28] were applied to sets that were preprocessed and also to each original training set (without any preprocessing). Both classifiers were taken from the WEKA toolkit [10] and all hyper-parameters of J48 were set to the default values. The SVM employed a polynomial kernel of degree 1 and was trained with the sequential minimal optimization algorithm [28].

### 3.1 Classification Performance Measures

The decisions made by a classification model over a set of samples can be expresed in the form of a confusion matrix, where each entry $(i, j)$ contains the number of correct/incorrect predictions. Given a problem with $C$ classes, Table 2 corresponds to a $C \times C$ confusion matrix: columns represent the predicted class and rows indicate the actual class. The elements of the diagonal contain the total number of correct predictions in each class, whereas the remaining entries summarise the number of misclassifications.

Several performance measures based on straightforward indices can be easily formulated from the confusion matrix, revealing results on each class. However, very often the evaluation has to be performed using more ellaborated measures in order to reflect the overall effectiveness of a classifier. In this paper, the accuracy, the kappa statistic and the geometric mean were used to assess the classification performance on the 92AV3C hyperspectral image database.

The most traditional metric for measuring the performance of learning systems is the accuracy ($Acc$), which can be defined as the degree of fit (matching) between the predictions and the true classes of data. However, the use of plain accuracy to evaluate

**Table 2.** Confusion matrix for a multi-class problem

| | Predicted class | | | |
|---|---|---|---|---|
| Actual class | $\sigma_1$ | $\sigma_2$ | $\cdots$ | $\sigma_C$ |
| $\sigma_1$ | $r_{1,1}$ | $r_{1,2}$ | $\cdots$ | $r_{1,C}$ |
| $\sigma_2$ | $r_{2,1}$ | $r_{2,2}$ | $\cdots$ | $r_{2,C}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ |
| $\sigma_C$ | $r_{C,1}$ | $r_{C,2}$ | $\cdots$ | $r_{C,C}$ |

the classifiers in imbalanced domains might produce misleading conclusions, since it is strongly biased to favor the majority classes. In order to maximize the individual accuracy on each class while keeping them balanced, Kubat *et al.* [21] proposed to use the geometric mean of accuracies ($Gmean$), which is calculated by multiplying the accuracies of the $C$ classes and taking the $C$'th root of this product. On the other hand, the kappa statistic or kappa coefficient ($\kappa \in [-1, +1]$) measures pairwise agreement among a set of classifications, correcting for expected chance agreement. A value of $+1$ means that there exists a total agreement, whereas $\kappa = -1$ reflects a complete disagreement. A value of $0$ suggests that there is no agreement other than which would be expected by chance. Note that the kappa coefficient is a standard performance measure widely used in remote sensing.

Apart from these global metrics, the average accuracy of each individual class ($Acc_i$) was calculated in order to evaluate the effect of the preprocessing techniques on the majority and minority classes separately. The arithmetic mean of these $C$ individual accuracies, say $Mean$, will be also included in the next section as an additional overall estimate of the performance.

$$Mean = \frac{\sum_{i=1}^{C} Acc_i}{C} \tag{1}$$

## 4 Results and Discussion

Table 3 reports the results, in terms of the four global performance metrics enumerated in the previous section, given by J48 and SVM when classifying the test samples. As can be observed, the use of a dimensionality reduction method (individually or together with some resampling algorithm) produces an important decrease in SVM performance, whereas this effect is much less noticeable with the J48 decision tree. Focusing on the resampling strategies, it is worth pointing out that SMOTE generally excels the random under-sampling approach, except when it is combined with PCA using the SVM classifier.

Among the performance metrics included in Table 3, the $Gmean$ is probably the most useful one since it is calculated from the individual accuracy on each class. This is especially remarkable in the case of SVM: despite the plain accuracy and kappa using the original training set are high enough, the geometric mean is 0, what means that all samples from one or more classes have been misclassified. The only strategies capable of overcoming this problem are SMOTE, CFS+SMOTE and SMOTE+CFS (combining

**Table 3.** Global performance measures obtained with J48 and SVM

| | | J48 | | | | SVM | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $Acc$ | $\kappa$ | $Mean$ | $Gmean$ | $Acc$ | $\kappa$ | $Mean$ | $Gmean$ |
| Original | | 0.739 | 0.703 | 0.710 | 0.692 | 0.817 | 0.791 | 0.694 | 0.000 |
| PCA | | 0.688 | 0.644 | 0.652 | 0.613 | 0.482 | 0.367 | 0.304 | 0.000 |
| CFS | | 0.734 | 0.697 | 0.693 | 0.668 | 0.665 | 0.606 | 0.498 | 0.000 |
| RUS | 50% | 0.708 | 0.671 | 0.707 | 0.688 | 0.807 | 0.781 | 0.710 | 0.000 |
| | 75% | 0.677 | 0.637 | 0.699 | 0.682 | 0.779 | 0.752 | 0.710 | 0.000 |
| SMOTE | 50% | 0.734 | 0.698 | 0.729 | 0.716 | 0.827 | 0.804 | 0.883 | 0.876 |
| | 75% | 0.734 | 0.698 | 0.732 | 0.719 | 0.824 | 0.801 | 0.891 | 0.886 |
| PCA+RUS | 50% | 0.661 | 0.617 | 0.658 | 0.623 | 0.508 | 0.431 | 0.349 | 0.000 |
| | 75% | 0.638 | 0.593 | 0.654 | 0.619 | 0.488 | 0.416 | 0.352 | 0.000 |
| PCA+SMOTE | 50% | 0.671 | 0.627 | 0.692 | 0.673 | 0.474 | 0.388 | 0.547 | 0.333 |
| | 75% | 0.663 | 0.620 | 0.692 | 0.675 | 0.526 | 0.464 | 0.604 | 0.472 |
| RUS+PCA | 50% | 0.661 | 0.618 | 0.657 | 0.622 | 0.505 | 0.428 | 0.347 | 0.000 |
| | 75% | 0.639 | 0.595 | 0.661 | 0.629 | 0.485 | 0.413 | 0.350 | 0.000 |
| SMOTE+PCA | 50% | 0.669 | 0.625 | 0.682 | 0.660 | 0.457 | 0.363 | 0.528 | 0.276 |
| | 75% | 0.661 | 0.618 | 0.683 | 0.664 | 0.509 | 0.444 | 0.587 | 0.459 |
| CFS+RUS | 50% | 0.705 | 0.667 | 0.694 | 0.671 | 0.679 | 0.633 | 0.533 | 0.000 |
| | 75% | 0.672 | 0.632 | 0.691 | 0.672 | 0.661 | 0.618 | 0.563 | 0.000 |
| CFS+SMOTE | 50% | 0.727 | 0.690 | 0.717 | 0.701 | 0.699 | 0.656 | 0.767 | 0.729 |
| | 75% | 0.727 | 0.691 | 0.728 | 0.715 | 0.711 | 0.673 | 0.791 | 0.767 |
| RUS+CFS | 50% | 0.705 | 0.667 | 0.697 | 0.675 | 0.685 | 0.640 | 0.544 | 0.000 |
| | 75% | 0.677 | 0.637 | 0.697 | 0.680 | 0.682 | 0.641 | 0.582 | 0.000 |
| SMOTE+CFS | 50% | 0.735 | 0.699 | 0.730 | 0.716 | 0.764 | 0.731 | 0.826 | 0.808 |
| | 75% | 0.731 | 0.695 | 0.727 | 0.713 | 0.769 | 0.740 | 0.847 | 0.834 |

SMOTE with PCA provides some increase in performance, but still insufficient), thus suggesting that it is preferable to generate synthetic minority samples rather than to remove instances from the majority classes.

In hyperspectral imaging, it is generally accepted that feature selection is better than extraction because of two main reasons [26]. On the one hand, feature extraction would need the whole (or most) of the original data representation to extract the new features, forcing to always deal with the whole initial representation of the data. Besides, since the data are transformed, some crucial and critical information might be compromised and distorted. The present experiments support this assertion since CFS has been consistently superior to PCA in terms of any performance measure, irrespective of the classifier and the strategy (applied alone or combined with resampling) used.

On the other hand, when comparing the results given by SMOTE and SMOTE+CFS, one should be aware of both the classification performance and the computational needs. Although the single application of SMOTE attains better results than combining SMOTE with CFS, it has to be noted that the feature selection algorithm produces a severe reduction in the number of bands, what represents an additional benefit in form of computational savings.

As regards the application order of the two preprocessing tools, it appears that resampling should be performed before any dimensionality reduction, especially in the case of using over-sampling together with feature selection. This suggests that SMOTE and RUS need the whole set of features to lead to the highest performance, meaning that the removal of bands causes a loss of hyperspectral information that is necessary for correctly resampling the data set.

Note that the average number of bands given by CFS is 28 when it has been applied before the resampling algorithms, whereas it selects about 30 and 48 bands when RUS and SMOTE are firstly run, respectively. On the other hand, PCA gives 3 bands in all cases, that is. it produces a dramatic reduction in dimensionality.

**Table 4.** Average classification accuracy on each class with the J48 decision tree ($\sigma_1$, $\sigma_5$, $\sigma_{10}$, $\sigma_{12}$ correspond to the minority classes, and $\sigma_8$, $\sigma_{13}$, $\sigma_{14}$ are the majority classes)

| | | Class | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | $\sigma_1$ | $\sigma_5$ | $\sigma_{10}$ | $\sigma_{12}$ | $\sigma_8$ | $\sigma_{13}$ | $\sigma_{14}$ |
| Original | | 0.866 | 0.643 | 0.590 | 0.496 | 0.916 | 0.645 | 0.749 |
| PCA | | 0.923 | 0.385 | 0.636 | 0.422 | 0.879 | 0.486 | 0.759 |
| CFS | | 0.869 | 0.495 | 0.688 | 0.388 | 0.923 | 0.637 | 0.748 |
| RUS | 50% | 0.893 | 0.648 | 0.625 | 0.415 | 0.883 | 0.595 | 0.609 |
| | 75% | 0.892 | 0.628 | 0.606 | 0.457 | 0.846 | 0.559 | 0.559 |
| SMOTE | 50% | 0.914 | 0.676 | 0.707 | 0.515 | 0.900 | 0.627 | 0.726 |
| | 75% | 0.924 | 0.698 | 0.686 | 0.519 | 0.902 | 0.627 | 0.714 |
| PCA+RUS | 50% | 0.932 | 0.383 | 0.682 | 0.435 | 0.867 | 0.479 | 0.597 |
| | 75% | 0.931 | 0.361 | 0.745 | 0.376 | 0.834 | 0.410 | 0.559 |
| PCA+SMOTE | 50% | 0.951 | 0.596 | 0.750 | 0.618 | 0.814 | 0.427 | 0.722 |
| | 75% | 0.944 | 0.608 | 0.728 | 0.619 | 0.817 | 0.435 | 0.668 |
| RUS+PCA | 50% | 0.932 | 0.372 | 0.654 | 0.430 | 0.874 | 0.470 | 0.597 |
| | 75% | 0.933 | 0.335 | 0.746 | 0.472 | 0.836 | 0.415 | 0.558 |
| SMOTE+PCA | 50% | 0.923 | 0.578 | 0.744 | 0.477 | 0.805 | 0.418 | 0.721 |
| | 75% | 0.926 | 0.559 | 0.769 | 0.487 | 0.793 | 0.428 | 0.668 |
| CFS+RUS | 50% | 0.891 | 0.509 | 0.709 | 0.386 | 0.893 | 0.596 | 0.617 |
| | 75% | 0.912 | 0.511 | 0.702 | 0.479 | 0.851 | 0.546 | 0.569 |
| CFS+SMOTE | 50% | 0.914 | 0.650 | 0.735 | 0.467 | 0.897 | 0.614 | 0.726 |
| | 75% | 0.926 | 0.643 | 0.761 | 0.538 | 0.899 | 0.620 | 0.708 |
| RUS+CFS | 50% | 0.888 | 0.519 | 0.694 | 0.403 | 0.890 | 0.601 | 0.610 |
| | 75% | 0.897 | 0.538 | 0.727 | 0.466 | 0.849 | 0.563 | 0.564 |
| SMOTE+CFS | 50% | 0.924 | 0.648 | 0.740 | 0.521 | 0.899 | 0.628 | 0.730 |
| | 75% | 0.922 | 0.632 | 0.741 | 0.494 | 0.891 | 0.621 | 0.710 |

In order to assess the effect of the preprocessing approaches on each class separately, Tables 4 and 5 report the average classification accuracy achieved on each individual class with J48 and SVM, respectively. For the sake of clarity, only the results corresponding to the smallest classes ($\sigma_1$, $\sigma_5$, $\sigma_{10}$, $\sigma_{12}$) and those of the most represented classes ($\sigma_8$, $\sigma_{13}$, $\sigma_{14}$) were here included. One can see that both resampling techniques improve the accuracy achieved on the minority classes, but in some cases they entail a slight reduction on the performance of the majority classes. It is worth noting that this degradation appears to be less significant when using SMOTE, as already concluded from the global metrics in Table 3.

If we focus on the results of PCA and CFS (without resampling), it is interesting to remark that these lead to a decrease in the performance of most classes, especially when used with the SVM classifier. However, the application of SMOTE before using those algorithms mitigates this effect, suggesting that it is important to balance the classes before reducing the dimensionality of hyperspectral data.

Regarding the classes with 0% of accuracy when classified with SVM (those cases in which the geometric mean was 0 as reported in Table 3), the over-sampling technique allows to overcome this problem and obtain a certain trade-off between the majority and minority classes.

**Table 5.** Average classification accuracy on each class with the SVM ($\sigma_1, \sigma_5, \sigma_{10}, \sigma_{12}$ correspond to the minority classes, and $\sigma_8$, $\sigma_{13}$, $\sigma_{14}$ are the majority classes)

| | | Class | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | $\sigma_1$ | $\sigma_5$ | $\sigma_{10}$ | $\sigma_{12}$ | $\sigma_8$ | $\sigma_{13}$ | $\sigma_{14}$ |
| Original | | 0.936 | 0.000 | 0.314 | 0.000 | 0.970 | 0.754 | 0.857 |
| PCA | | 0.831 | 0.000 | 0.000 | 0.000 | 0.984 | 0.000 | 0.987 |
| CFS | | 0.856 | 0.000 | 0.000 | 0.000 | 0.975 | 0.529 | 0.884 |
| RUS | 50% | 0.936 | 0.000 | 0.376 | 0.000 | 0.968 | 0.786 | 0.695 |
| | 75% | 0.935 | 0.000 | 0.387 | 0.000 | 0.953 | 0.742 | 0.580 |
| SMOTE | 50% | 0.966 | 0.928 | 0.930 | 1.000 | 0.924 | 0.721 | 0.801 |
| | 75% | 0.968 | 0.924 | 0.930 | 1.000 | 0.911 | 0.755 | 0.711 |
| PCA+RUS | 50% | 0.831 | 0.000 | 0.000 | 0.000 | 0.984 | 0.482 | 0.564 |
| | 75% | 0.833 | 0.000 | 0.000 | 0.000 | 0.983 | 0.489 | 0.442 |
| PCA+SMOTE | 50% | 0.935 | 0.648 | 0.949 | 0.843 | 0.850 | 0.016 | 0.822 |
| | 75% | 0.937 | 0.687 | 0.941 | 0.817 | 0.754 | 0.241 | 0.709 |
| RUS+PCA | 50% | 0.832 | 0.000 | 0.000 | 0.000 | 0.984 | 0.486 | 0.553 |
| | 75% | 0.831 | 0.000 | 0.000 | 0.000 | 0.983 | 0.491 | 0.427 |
| SMOTE+PCA | 50% | 0.933 | 0.678 | 0.933 | 0.802 | 0.827 | 0.021 | 0.863 |
| | 75% | 0.935 | 0.702 | 0.929 | 0.767 | 0.730 | 0.257 | 0.696 |
| CFS+RUS | 50% | 0.860 | 0.000 | 0.000 | 0.000 | 0.972 | 0.687 | 0.666 |
| | 75% | 0.858 | 0.000 | 0.000 | 0.000 | 0.925 | 0.570 | 0.580 |
| CFS+SMOTE | 50% | 0.945 | 0.891 | 0.926 | 0.953 | 0.936 | 0.456 | 0.795 |
| | 75% | 0.955 | 0.878 | 0.922 | 0.968 | 0.929 | 0.552 | 0.671 |
| RUS+CFS | 50% | 0.874 | 0.000 | 0.000 | 0.000 | 0.971 | 0.687 | 0.665 |
| | 75% | 0.885 | 0.000 | 0.000 | 0.000 | 0.947 | 0.612 | 0.600 |
| SMOTE+CFS | 50% | 0.959 | 0.898 | 0.926 | 0.989 | 0.931 | 0.626 | 0.783 |
| | 75% | 0.964 | 0.909 | 0.933 | 0.994 | 0.912 | 0.674 | 0.684 |

## 5 Conclusions and Further Extensions

The present paper has focused on classification of hyperspectral imagery with two complex characteristics: high dimensionality and severe skewed class distributions. The experimental study has allowed to draw some preliminary conclusiones: (i) It results more important to balance the classes rather than to reduce the dimensionality, at least in terms of classification performance (accuracy, geometric mean or some other metric); (ii) The best choice seems to be the application of SMOTE followed by a feature selection algorithm; and (iii) The SVM appears to be a more robust classifier than the J48 decision tree, at least for this particular hyperspectral database.

As already pointed out, in hyperspectral imaging, selection selection is commonly better than feature extraction, especially because relevant information might be distorted by means of the transformation. Although the empirical results have corroborated this statement, one should also take into account that PCA obtains a much stronger reduction in the number of bands than CFS and therefore, classification performance could be affected by this fact. Thus future research will be addresed to analyse in depth the relationship between dimensionality reduction and effectiveness when using both feature selection and extraction algorithms. Another direction for future studies would be incorporating an editing/filtering phase to remove possible noisy data before any other process.

## Acknowledgment

This work has partially been supported by the Spanish Ministry of Education and Science under grants CSD2007–00018, AYA2008–05965–0596 and TIN2009–14205, the Fundacio Caixa Castello–Bancaixa under grant P1–1B2009–04, and the Generalitat Valenciana under grant PROMETEO/2010/028.

## References

1. Blagus, R., Lusa, L.: Class prediction for high-dimensional class-imbalanced data. Bioinformatics 11(1), 523–540 (2010)
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth Inc., Monterey, CA (1984)
3. Bruzzone, L., Serpico, S.B.: Classification of imbalanced remote-sensing data by neural networks. Pattern Recogn. Lett. 18(11-13), 1323–1328 (1997)
4. Camps-Valls, G.: Machine learning in remote sensing data processing. In: Proc. IEEE Int'l. Workshop Machine Learning for Signal Processing. pp. 1–6. Grenoble, France (2009)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357 (2002)
6. Chen, X., Fang, T., Huo, H., Li, D.: Semisupervised feature selection for unbalanced sample sets of VHR images. IEEE Geosci. Remote Sens. Lett. 7(4), 781 –785 (2010)
7. Ezawa, K.J., Singh, M., Norton, S.W.: Learning goal oriented bayesian networks for telecommunications risk management. In: Proc. 13th Int'. Conf. Machine Learning. pp. 139–147 (1996)
8. Fawcett, T., Provost, F.: Adaptive fraud detection. Data Min. Knowl. Disc. 1(3), 291–316 (1997)
9. García, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. Evol. Comput. 17(3), 275–306 (2009)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. Newslett. 11, 10–18 (2009)

11. Hall, M.A.: Correlation-based Feature Subset Selection for Machine Learning. Ph.D. thesis, Dept. Computer Science, University of Waikato, Hamilton, New Zealand (1999)
12. Han, H., Wang, W.Y., Mao, B.H.: Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: Proc. Int'l. Conf. Intelligent Computing. pp. 878–887. Hefei, China (2005)
13. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21(9), 1263–1284 (2009)
14. Hsu, P.H., Tseng, Y.H., Gong, P.: Dimension reduction of hyperspectral images for classification applications. Geogr. Inf. Sci. 8(1), 1–8 (2002)
15. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intell. Data Anal. 6(5), 429–449 (2002)
16. Jiménez, L.O., Landgrebe, D.A.: Hyperspectral data analysis and supervised feature reduction via projection pursuit. IEEE Trans. Geosci. Remote Sens. 37(6), 2653–2667 (1999)
17. Jolliffe, I.T.: Principal Component Analysis. Springer-Verlag, New York (2002)
18. Kamal, A.H.M., Zhu, X., Narayanan, R.: Gene selection for microarray expression data with imbalanced sample distributions. In: Proc. Int'l. Joint Conf. Bioinformatics, Systems Biology and Intelligent Computing. pp. 3–9. Shanghai, China (2009)
19. Kecman, V.: Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. MIT Press, Cambridge, MA (2001)
20. Kubat, M., Holte, R.C., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. Mach. Learn. 30(2-3), 195–215 (1998)
21. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. In: Proc. 14th Int'l. Conf. Machine Learning. pp. 179–186. Nashville, USA (1997)
22. Landgrebe, D.A.: Signal Theory Methods in Multispectral Remote Sensing. Wiley, Hoboken, NJ (2003)
23. Lin, L., Ravitz, G., Shyu, M.L., Chen, S.C.: Effective feature space reduction with imbalanced data for semantic concept detection. In: Proc. Int'l. Conf. Sensor Networks, Ubiquitous, and Trustworthy Computing. pp. 262–269. Taichung, Taiwan (2008)
24. Liu, X.Y., Zhou, Z.H.: The influence of class imbalance on cost-sensitive learning: An empirical study. In: Proc. 6th Int'l. Conf. Data Mining. pp. 970–974. Hong Kong (2006)
25. Maloof, M.A.: Learning when data sets are imbalanced and when costs are unequal and unknown. In: Workshop Learning from Imbalanced Data Sets II. Whasington, DC (2003)
26. Martínez-Usó, A., Pla, F., Sotoca, J.M., García-Sevilla, P.: Clustering-based hyperspectral band selection using information measures. IEEE Trans. Geosci. Remote Sens. 45(12), 4158–4171 (2007)
27. Melgani, F., Bruzzone, L.: Classification of hyperspectral remote sensing images with support vector machines. IEEE Trans. Geosci. Remote Sens. 42(8), 1778–1790 (2004)
28. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods. pp. 185–208. MIT Press, Cambridge, MA (1999)
29. Quinlan, J.R.: Induction of decision trees. Mach. Learn. 1, 81–106 (1986)
30. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA (1993)
31. Richards, J.A., Jia, X.: Using suitable neighbors to augment the training set in hyperspectral maximum likelihood classification. IEEE Geosci. Remote Sens. Lett. 5(4), 774–777 (2008)
32. Trebar, M., Steele, N.: Application of distributed SVM architectures in classifying forest data cover types. Comput. Electron. Agr. 63(2), 119–130 (2008)
33. Van Hulse, J., Khoshgoftaar, T.M., Napolitano, A., Wald, R.: Feature selection with high-dimensional imbalanced data. In: IEEE Int'l. Conf. Data Mining Workshops, 2009. pp. 507–514. Miami, USA (2009)
34. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)

35. Wasikowski, M., Chen, X.W.: Combating the small sample class imbalance problem using feature selection. IEEE Trans. Knowl. Data Eng. 22(10), 1388–1400 (2010)
36. Waske, B., Benediktsson, J.A., Sveinsson, J.R.: Classifying remote sensing data with support vector machines and imbalanced training data. In: Proc. 8th Int'l. Workshop Multiple Classifier Systems. pp. 375–384. Reykjavik, Iceland (2009)
37. Williams, D.P., Myers, V., Silvious, M.S.: Mine classification with imbalanced data. IEEE Geosci. Remote Sens. Lett. 6(3), 528 –532 (2009)
38. Zhang, J., Mani, I.: kNN approach to unbalanced data distributions: a case study involving information extraction. In: Proc. Workshop Learning from Imbalanced Datasets. Washington DC (2003)
39. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans. Knowl. Data Eng. 18(1), 63–77 (2006)