Annalisa Appice, Pietro Guccione, Donato Malerba,

A novel spectral-spatial co-training algorithm for the transductive classification of hyperspectral imagery data, Pattern Recognition, Volume 63, 2017, Pages 229-245, ISSN 0031-3203,

The published version is available on

https://doi.org/10.1016/j.patcog.2016.10.010

A Novel Spectral-Spatial Co-Training Algorithm for the Transductive Classification of Hyperspectral Imagery Data

Annalisa Appice[a,*], Pietro Guccione[b], Donato Malerba[a]

[a] *Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, via Orabona, 4 - 70125 Bari - Italy, Consorzio Interuniversitario Nazionale per l'Informatica - CINI, Centro Interdipartimentale di Logica e Applicazioni - CILA*
[b] *Dipartimento di Ingegneria Elettrica ed Informazione, Politecnico di Bari, via Orabona, 4 - 70125 Bari - Italy*

Abstract

The automatic classification of hyperspectral data is made complex by several factors, such as the high cost of true sample labeling coupled with the high number of spectral bands, as well as the spatial correlation of the spectral signature. In this paper, a transductive collective classifier is proposed for dealing with all these factors in hyperspectral image classification. The transductive inference paradigm allows us to reduce the inference error for the given set of unlabeled data, as sparsely labeled pixels are learned by accounting for both labeled and unlabeled information. The collective inference paradigm allows us to manage the spatial correlation between spectral responses of neighboring pixels, as interacting pixels are labeled simultaneously. In particular, the innovative contribution of this study includes: (1) the design of an application-specific co-training schema to use both spectral information and spatial information, iteratively extracted at the object (set of pixels) level via collective inference; (2) the formulation of a spatial-aware example selection schema that accounts for the spatial correlation of predicted labels to augment training sets during iterative learning and (3) the investigation of a diversity class criterion that allows us to speed-up co-training classification. Experimental results validate the accuracy and efficiency of the proposed spectral-spatial, collective, co-training strategy.

Keywords: Hyperspectral imagery classification, Transductive learning, Collective Inference, Co-training, Spectral-spatial data

1. Introduction

Hyperspectral Image (HSI) collected by imaging spectrometers has captured increasingly rich spectral information. Advances in hyperspectral imaging technology allow nowadays the simultaneous measurement of hundreds of spectral bands for each image pixel. This high spectral resolution increases the possibility

*Corresponding author (Tel: +39 (0)805443262 Fax: +39(0)805443269)
Email addresses: annalisa.appice@uniba.it (Annalisa Appice), guccione@poliba.it (Pietro Guccione), donato.malerba@uniba.it (Donato Malerba)

of more accurately discriminating materials of interest in the spectral domain. This benefits the theoretical research on hyperspectral data classification in various real-world applications, such as environmental mapping, crop analysis, plant and mineral exploration, as well as biological and chemical detection [43].

Hyperspectral data classification is the process used to produce thematic maps from remote sensed images. A thematic map represents the Earth's surface objects (e.g. soil, vegetation, roof, road, buildings). Its construction implies that themes or categories, selected for the map, are distinguished in the remote sensed image based on their surface reflectance in optic and Near InfraRed (NIR) wavelengths. Classification involves clustering the pixels (imagery data examples) of an image into a set of known classes (themes) such that pixels in the same class have similar properties. In this scenario, every pixel is expressed with a vector model that represents the spectral signature as a vector of numeric features (namely *spectral features*). A spectral feature represents the spectral reflectance at a specific band. Additionally, every pixel is associated with a specific position of a uniform grid, which describes the spatial arrangement of the sensed scene. Finally, it is assigned with a certain (possibly unknown) class label. From a methodological viewpoint, the automatic classification of hyperspectral data is not a trivial task [16]. It is made complex by two factors: (1) the high cost of true sample labeling coupled with the high number of spectral channels and (2) the spatial correlation of pixels due to the arrangement of the topographic objects.

The human-supervised effort needed to collect only few labeled imagery pixels, properly distributed among the classes, makes the definition of the proper training set for learning an imagery classifier still an open challenge [14]. On the other hand, the low number of collected ground truth labels, compared to the high number of spectral features, is not always sufficient for a reliable estimate of the classifier parameters. In fact, if the number of samples (training set) is too low compared to the number of variates, overfitting the training data may be a problem, i.e. we can learn a model that exactly fits the training data without accounting for a wider generalization [10]. This behavior, which is known as Hughes's phenomenon [23], may cause a reduction in classification accuracy.

Recent studies have focused on the use of unlabeled samples, in order to overcome the problem of small size labeled samples in high dimensionality data classification [48]. Two main approaches have been proposed: the semi-supervised approach and the transductive approach [56]. Both settings jointly exploit labeled and unlabeled samples [46]. The semi-supervised approach is a type of inductive learning, since it learns a general hypothesis that can be used to make predictions on any possible example (also outside the unlabeled sampled set considered during the learning phase). The transductive approach requires less - it is only interested in reducing the inference error for the given set of unlabeled data, without trying to improve the overall quality of the learned hypothesis. Therefore, as pointed out by Vapnik [57], the idea of transduction (labeling a test set) appears inherently easier than (semi-supervised) induction (learning a general hypothesis) and it is likely to become much more popular in the future. In the last few years, the machine learning scientific community has shown a growing interest in the definition of a variety of semi-supervised and transductive

classification algorithms (e.g. [17, 26, 27, 50, 56]). Following this mainstream of research, several semi-supervised and transductive algorithms (e.g. [1, 6, 7, 20, 36, 58]) have been specifically defined to cope with limited labeled data in hyperspectral image classification problems. They are application-specific algorithms that, in addition to utilizing both labeled and unlabeled pixels for classification, account for the spatial correlation of the pixels during the learning process.

The spatial correlation of the spectral signature refers to the relation (or dependence) between pixels, due to their close locations. Intuitively, spatial correlation is a property of random features taking values, at pairs of locations a certain distance apart, that are more similar (positive autocorrelation) or less similar (negative autocorrelation) than expected for pairs of observations at randomly selected locations [30]. In particular, positive spatial correlation occurs when the values of a given property are highly uniform among objects in close proximity, i.e., in the same neighborhood. In the case of hyperspectral images of geographical areas, spatial correlation exists in the positive form as there is a *slowly progressive* spatial variation in the spectral signature [39]. This means that by picturing the spatial variation of the observed features in a map, we may observe regions where the distribution of values is smoothly continuous, with some boundaries possibly marked by sharp discontinuities. These discontinuities are due to the bounds of the topographic objects. An emerging trend into hyperspectral analysis is to accommodate spatial correlation into the classification process as, in this way, classification accuracy can be gained [43, 16].

Recent research in data mining has explored the use of collective inference to exploit data correlation when learning predictive models. According to Jensen et al. [25], as well as Getoor and Taskar [19], collective inference refers to the combined classification of a set of correlated instances. This means that, contrary to traditional algorithms, which make predictions for data instances individually, regardless of the relationships or correlations among instances, collective inference approaches predict the labels of related instances simultaneously, using similarities that appear among groups of correlated data. Recently, collective classification has been investigated in combination with semi-supervised and transductive learning [60, 49, 37]. On the other hand, various studies in hyperspectral imagery classification [16, 20] have initiated the investigation of collective inference as a means to explicitly account for the spatial variation of the imagery data.

Co-training, originally introduced by Blum and Mitchell [4], is an important paradigm of both semi-supervised and transductive learning [35, 9] that offers an unique opportunity to explicitly deal with the spatial correlation of the spectral signature, by presumably reducing the labeling uncertainty that may exist when only spectral information is used. It is usually applied to data sets whose features are separated into two disjoint sets, which are regarded as two independent data profiles. At each iteration of the co-training process, two learners are trained independently from the two profiles and are required to label some unlabeled examples for each other to augment the training set. The iterative process continues until it reaches some stopping criterion. The success of co-training lies in utilizing the unlabeled samples and the information from the other profile. At present, the co-training strategy has already been considered

to solve hyperspectral image classification issues [22, 61], due to the following twofold reasons: (1) co-training can exploit the limited labeled data with a wealth of unlabeled data to improve the performance and (2) besides the spectral features, to be used as a profile, we can generate another kind of discriminative feature (especially spatial information) as an additional profile to help learning. This use of co-training in hyperspectral image classification makes sense as we can suppose that spectral and spatial features are conditionally independent (i.e. the spatial correlation on a group of neighbor pixels is not a function of their spectral signatures). The still open problem of the co-training strategy is the selection of reliably predicted labels for augmenting the training set. Several criteria have been defined for general co-training [11], as well as for hyperspectral co-training [22, 61]. But, to the best of our knowledge, none exploits the spatial correlation of the spectral signature as a way to describe the spatial correlation of the label.

Motivated by the interest in improving the accuracy of spatial information classification when learning from few labeled data in a high dimensional spectral space, a new transductive algorithm, called S^2CoTraC (Spectral-Spatial Co-training for Transductive Hyperspectral Classification) is proposed in this paper. This algorithm is synthesized by following the main stream of our recent research on collective inference and transductive learning in hyperspectral image classification [20, 1]. It performs the iterative construction of *various* spatial features over pixel objects (spatial neighborhoods) via a *collective iterative convergence* algorithm, in order to deal with spatial information. It applies a *transductive learning* approach with a *co-training* schema, in order to make accurate predictions of the unknown labels of a sparsely labeled image.

The novelty of this study, in particular with respect to our previous works [20, 1], includes the formulation of an application-specific co-training schema to manage spectral and (collective-based) spatial information. The effectiveness of the proposed algorithm is assessed via an empirical study on several hyperspectral data sets corresponding to various contexts. This study contributes to proving that the proposed formulation of a collective-based co-training classifier is more accurate than the collective turbo-code described in [20], as well as the collective ensemble described in [1]. In addition, the presented algorithm outperforms the state-of-the-art co-training classifiers, which are defined for hyperspectral classification [22, 61], but do not account for collective inference to deal with spatial information. In general, our algorithm gains in accuracy compared to various classifiers defined in the hyperspectral image analysis literature. Another novel contribution of this study is the consideration of an example selection schema that accounts for the spatial correlation of imagery labels, in order to select new training examples for the iterative learning process. This spatial-aware schema advances our previous research [20, 1] that disregarded the spatial arrangement of the predicted labels by basing the estimate of the reliability of a label on either the posterior probability of the label predicted with a logistic classifier [20] or the diversity of the labels predicted by an ensemble [1]. Finally, in this study, we start to pay attention to the efficiency issue and propose a diversity class criterion that can be used in combination with co-training, in order to speed-up the learning process.

The paper is organized as follows. The next section reports relevant related works. Section 3 introduces basic concepts, while Section 4 illustrates the algorithm. Section 5 describes the data sets, the experimental setup and reports the results. Finally, in Section 6 some conclusions are drawn and future work is outlined.

2. Related works

Over the last two decades, several supervised machine learning algorithms have been applied to hyperspectral image classification. Spectral information is processed, in order to train a classifier with the labeled data samples. The quality of these pixelwise classification algorithms is strongly related to the quality and number of training samples under the influence of Hughes's phenomenon. In this context, Support Vector Machines (SVMs) have been widely used to deal with Hughes's phenomenon by addressing large feature spaces and producing solutions from sparsely labeled data [38]. Recently, Multinomial Logistic Regression (MLR) [31, 32] has been shown to provide an alternative approach to deal with ill-posed problems. In alternative, dimensionality reduction techniques have been adopted, in order to mitigate Hughes's phenomenon as the dimensionality of the multispectral data is high [51]. Finally, multiple classifier systems, e.g. classifier ensembles, have proved successful in several hyperspectral image classification applications [8]. However, these algorithms neither account for the unlabeled data nor account for the spatial correlation of data.

A new learning trend has recently emerged in hyperspectral imagery analysis. It exploits semi-supervised or transductive learning and uses unlabeled data to increase the number of labeled samples, reduce the impact of the overfitting and alleviate Hughes's phenomenon. Several general-purpose algorithms have been defined in the machine learning field, in order to classify using unlabeled data (e.g. generative algorithms [17], transductive inference of Support Vector Machines [26, 50], spectral graph partitioning algorithms [27] and self-labeling algrithms [56]). On the other hand, various application-specific, semi-supervised and transductive algorithms have been formulated, in order to specifically classify hyperspectral data. For example, Bruzzone et al. [6], as well as Maulik and Chakraborty [36] described a transductive SVM algorithm for hyperspectral image classification. Ratle et al. [44] proposed a semi-supervised hyperspectral image classification algorithm based on neural networks. The algorithm consists of adding a flexible embedding regularizer to the loss function used for training neural networks. Although these algorithms incorporate unlabeled samples in the training phase, they still neglect the spatial correlation of data.

A wide plethora of spectral-spatial classifiers has recently been formulated in the hyperspectral imaging literature. They are commonly considered as hybrid algorithms, which combine spectral pixel-based information and object (neighborhood)-based spatial information. In particular, they learn the imagery classifier by accounting for the spatial correlation of the spectral signature and/or the class label. For example, several algorithms, based on Markov Random Fields, have proved quite successful in hyperspectral imaging [32, 31, 55, 33]. They take into account the continuity (in a probabilistic sense) of neighboring labels. In

other words, they exploit the likely fact that, in a hyperspectral image, two neighboring pixels may have the same label as a consequence of the spatial correlation of the class labels. On the other hand, Plaza et al. [43], Bovolo et al. [5], as well as Fauvel et al. [15] decided to account for the spatial correlation of the pixel spectra. They represented each pixel with a linear combination of the spectra of its neighboring pixels. This is done by defining several spectral-spatial kernels, which model the inter-pixel relations as the mean of the pixel spectra from a pixel's neighborhood system. Spatial information is directly included in the training process as a new constraint for the optimization problem. Tarabalka et al. [54] investigated the use of a watershed transformation, in order to determine a segmentation map of the image. They defined a two-stepped classification process, according to which the pixelwise SVM classification is followed by majority voting within the watershed regions. Shackelford and Davis [47] described a fuzzy pixel-based classifier that accounts for both spectral and spatial information to discriminate between spectrally similar road and building urban land cover classes. After pixel-based classification, they used a technique that utilizes both spectral and spatial heterogeneity, in order to segment the image and facilitate further object-based classification. Bernardini et al. [3] combined the results of automatic segmentation with the land cover information derived from pixel classification by means of the Winner Takes All algorithm. These algorithms deal with spatial correlation of data, but in the supervised setting.

Finally, there are studies which exploit hyperspectral spatial information in active learning, transductive and/or semi-supervised learning algorithms. Li et al. [33] adopted the loopy belief propagation, in order to estimate the posterior marginal distributions from the spectral and spatial information in the hyperspectral data. They enlarged the training set with new samples obtained via an active learning strategy. This strategy is based on the conditional marginals of the unlabeled samples, which encode the spatial information embedded in the posterior probabilities. Camps-Valls [7] presented a semi-supervised graph-based algorithm, designed to exploit both spectral and spatial information in the images through composite kernels. Wang et al. [58] proposed a spectral-spatial label propagation for the semi-supervised classification of hyperspectral imagery. Tan et al. [52] presented a semi-supervised support vector machine (SVM) with a segmentation-based ensemble. The algorithm utilizes spatial information extracted by a segmentation algorithm for unlabeled sample selection, while the classification is refined through a spectral-spatial feature ensemble technique. Guccione et al. [20] described a semi-supervised algorithm that constructs collective spatial features of the imagery data by computing the frequency of a class in the neighborhoods of a pixel. It constructs neighborhoods with growing size. Both spectral and spatial features are considered, in order to build, iteratively, two classifiers by running the MLR algorithm. The posterior probabilities of the two classifiers are combined using an ensemble decision, in order to determine the joint class prediction and decide the exit strategy from the iterative loop. Appice et al. [1] described a transductive algorithm to integrate the spectral information and the label spatial correlation through an ensemble system. The algorithm constructs two types of spatial profiles of the imagery data. One spatial profile is populated with

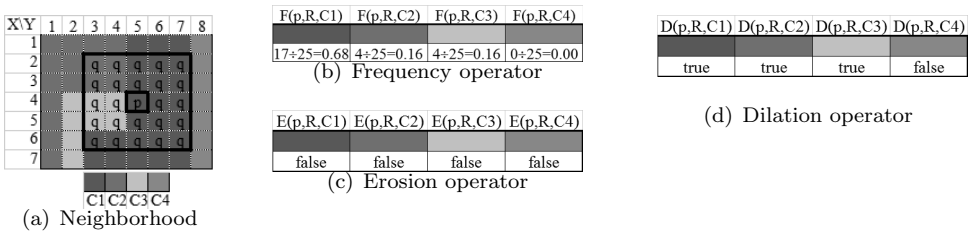
collective spatial features that describe the frequency of a class in the neighborhoods of a pixel. The other spatial profile is populated with collective spatial features that describe the morphology of a class in the neighborhoods of a pixel. Spatial features are iteratively updated, as new classes can be computed by the ensemble system during the iterative learning. The unlabeled examples which are equally classified by the majority of classifiers in the ensemble are used to expand the labeled set. The idea of constructing various spatial profiles of the imagery data was also discussed in [22], where Huang and He described a learning schema to iteratively combine a spectral profile with two spatial profiles of data through a transductive co-training system. In this study, collective inference is neglected, while the spatial profiles are extracted through the calculation of the Gray Level Co-occurrence Matrices (GLCM) and the Markov Random Field (MRF) of the spectral information, respectively. In each round of co-training, an unlabeled pixel is labeled for a classifier if the other two classifiers agree on the labeling. More recently, Zhang et al. [61] investigated the use of the spectral features and the 2-D Gabor features extracted from spectral domains as two distinct profiles for a new transductive co-training system of SVMs. In this case, the co-training process rates the classes according to the classification accuracy of the training data, and then adopts a different selection criterion according to the rating of the class. In particular, a class receives a good rating if the majority of the training data of this class are correctly predicted. Otherwise, it receives a poor rating. According to this characterization, the confidence criterion is used to select samples of good classes, while the probability pattern clustering is used to select samples of poor classes.

We note that the algorithms described in [20, 1, 22, 61] use few learning components (e.g. collective inference, various spatial profiles and co-training) that are also considered in this study. In any case, there are substantial differences that contribute to defining the novelty of this study (see Table 1). In particular, these algorithms, with the exception of the algorithms described in [20] and [1], extract spatial information based on the spectral information. So, they construct spatial features, which do not change during the learning phase. On the other hand, the algorithms described in [20] and [1] use collective inference to extract spatial information based on spectral-predicted labels. However, the algorithm described in [20] prevents the possibility of constructing and managing various spatial (-collective) profiles of the imagery data. In addition, both algorithms do not resort to the co-training paradigm, in order to deal with multiple data profiles. Finally, none of these algorithms accounts for the spatial correlation of data, when determining the reliability of predicted labels which can be selected, in order to augment the training set.

Table 1: A comparative analysis of the characteristics of the algorithms presented in this study, as well as in [20, 1, 22, 61].

	S^2CoTraC	irmc[20]	S^2Tec[1]	TriTraining[22]	mcogpc[61]
learning schema	co-training	ensemble	ensemble	co-training	co-training
spectral profile	yes	yes	yes	yes	yes
spatial profiles	2	1	2	2	1
collective inference	yes	yes	yes	no	no
example selection schema	spatial	aspatial	aspatial	aspatial	aspatial

3. Preliminary Concepts



(a) Neighborhood (b) Frequency operator (c) Erosion operator (d) Dilation operator

Figure 1: Pixel neighborhoods and spatial features: 1(a) the neighborhood constructed for the pixel p with radius R= 2 and a square shape; 1(b) the spatial features constructed for the pixel p over the neighborhood $\mathcal{N}(p, R = 2)$ with the frequency operator; 1(c) the spatial features constructed for the pixel p over the neighborhood $\mathcal{N}(p, R = 2)$ with the erosion operator; 1(d) the spatial features constructed for the pixel p over the neighborhood $\mathcal{N}(p, R = 2)$ with the dilation operator.

Let $\mathcal{D}$ be a hyperspectral imagery data set. It is a set of pixels (examples). Each pixel is a region of around a few square meters of the Earth's surface and a function of the sensor spatial resolution. It is associated to the spatial coordinates XY in the image, it is characterized by an m-dimensional vector of spectral features $\mathbf{S} = S_1, S_2, \ldots, S_m$ (descriptive space) and it can, in principle, be labeled according to an unknown target function, whose range is a finite set of k distinct labels, i.e. $C = \{C_1, C_2, \ldots, C_k\}$. As pixels are, in general, equally-space distributed over a regular grid, a hyperspectral data set is represented as a matrix. Thus, the spatial coordinate X is associated with the row index, while the spatial coordinate Y is associated with the column index of the matrix. Every spectral feature S_i is numeric and expresses how much the radiation is reflected, on average, at the i-th band of the considered spectral profile, from the resolution cell of the considered pixel. Every class C_i represents a distinct theme (i.e. type of Earth's surface object). A spatial neighborhood is a set of pixels q surrounding p in the imagery matrix. In the imagery analysis literature, spatial neighborhoods frequently have a square shape [43, 20], although alternative shapes like a circle or a cross can be also considered. Let R be a positive, integer-valued radius, the *square-shaped* spatial neighborhood $\mathcal{N}(p, R)$ of pixel p (see Figure 1(a)) is defined as $\mathcal{N}(p, R) = \bigcup_{I=-R}^{+R} \bigcup_{J=-R}^{+R} \{q(x + I, y + J) | q \text{ is a pixel of } \mathcal{D}\}$. The construction of spatial neighborhoods, coupled with every pixel of a hyperspectral imagery data set, define the actual *spatial structure* of the data set. By accounting for this spatial structure, *spatial features* can be constructed, in order to synthesize the information on spatial variation of data over imagery pixels. A vector of spatial features represents one *spatial data profile* for the image. In this study, we resort to the theory of collective inference [25, 19] and construct spatial features that express the label of a pixel depending on the labels of all the related neighbors of the pixel. The construction of these collective spatial features can be done only after the unlabeled part of the image has been preliminarily classified.

In hyperspectral imaging, spatial features are, frequently, constructed through the application of the

frequency-based operator [20] and/or the *morphology-based* (erosion, dilation, opening and closing) *operators* [53]. Given a pixel p (with $p \in \mathcal{D}$) and a square-shaped neighborhood $\mathcal{N}(p, R)$ (with radius R) coupled with p, the frequency operator can construct k real-valued spatial features to describe p, one feature for each class label C_i (with $i = 1, \ldots, k$), so that: $F(p, R, C_i) = \frac{|\{q \in \mathcal{N}(p,R) | q = C_i\}|}{|\mathcal{N}(p,R)|}$. The frequency operator allows us to build features that describe the relative abundance of the class labels over the neighborhood around every target pixel of the image. In this way, constructed features are able to quantify possible changes in the distribution of labels (see Figure 1(b)). On the other hand, the morphological operators allow us to construct $4 \cdot k$ Boolean-valued spatio-relational features, four for each class label C_i (with $i = 1, \ldots, k$) and for each morphological operator (erosion, dilation, opening and closing). The morphological operators are non-linear operations related to the shape of the objects in an image [2]. Let us consider the spatial neighborhoods as structuring elements. For a given class label, the erosion is defined as $E(p, R, C_i) = true$ if $\forall q \in \mathcal{N}(p, R)$ label$(q) = C_i$; $false$ otherwise. The dilation is defined as $D(p, R, C_i) = true$ if $\exists q \in \mathcal{N}(p, R)$ label$(q) = C_i$; $false$ otherwise. Intuitively, the erosion (see Figure 1(c)) erodes all the pixels that cannot contain the structuring element; the dilation (see Figure 1(d)), instead, preserves pixels if at least one of its neighbor is included in the structuring element [43]. By combining erosion and dilation, *opening* and *closing* operators are defined. The opening is the erosion followed by the dilation with the specified structuring elements. The idea of dilating the eroded image is to recover most structures of the original image, i.e. structures that were not removed by the erosion and are bigger than the structuring element. On the other hand, the closing is the dilation followed by the erosion. With opening or closing it is possible to obtain objects of the image which are larger or smaller than the structuring element [16].

4. The algorithm

The algorithm S^{2}CoTraC inputs the spectral signature and the spatial position of pixels of a sparsely labeled image, performs a transductive classification process and outputs the completely labeled image. The classification process is done by accounting for the initial spectral profile of pixels, as well as for various spatial profiles, which are constructed through collective inferences. These profiles are used to learn multiple classifiers via a co-training strategy. An ensemble consensus pattern of these multiple classifiers is, finally, used to predict unknown labels. In the following, we first formulate the learning problem (Subsection 4.1), then we describe the data profiles that we use in this study (Subsection 4.2), the algorithm that yields the spectral-spatial co-training inference within imagery data (Subsection 4.3) and the example selection criterion (Subsection 4.4). Finally, we analyze the time complexity of the algorithm (Subsection 4.5).

4.1. The transductive classification problem

Let $\mathcal{D}$ be a hyperspectral imagery data set whose pixels are sparsely labeled according to an unknown target function C, while they are all described by the spectral feature vector model $\mathbf{S}$ (details in Section 3).

The transductive classification problem inputs a labeled set $\mathcal{L} \subset \mathcal{D}$, by considering its information along the descriptive space $\mathbf{S}$ and the target space C, as well as the projection of the unlabeled set $\mathcal{U} = \mathcal{D} - \mathcal{L}$ on the descriptive space $\mathbf{S}$. It outputs predictions of the class values of examples in the unlabeled set $\mathcal{U}$, which are as accurate as possible. The learner receives full information (including labels) on $\mathcal{L}$ and partial information (without labels) on $\mathcal{U}$ and is required to predict the class values only of the examples in $\mathcal{U}$.

4.2. Imagery Data Profiles

The vector of spectral features is input as the hyperspectral imagery data set. It populates spectral profile $\mathbf{S}$. The vector of spatial features, constructed through the application of the frequency operator (see Section 3), defines frequency-based spatial profile $\mathbf{F}$. The vector of spatial features, constructed through the application of the four morphological operators (see Section 3), defines morphology-based spatial profile $\mathbf{M}$. Frequency and morphological features are constructed through coupling imagery pixels with (square-shaped) neighborhoods (details in Section 3). For each pixel, a set of neighborhoods with growing sides ($R \in RSet$) is constructed to deal with the fact that the image reflectivity (and consequently its labeling) is, usually, modeled as non-stationary in the spatial domain [24]. This idea of using a range of sizes follows the point of view of [43, 20], which showed how a range of texture structuring element sizes must be used, in order to capture as much as possible the shape and size of the spatial structures present in the image.

We note that both these spatial profiles account for the spatial correlation of labels collectively predicted, but they synthesize different information. The frequency operator represents the class distribution for a specific texture defined by the shape and the size of a given neighborhood. The morphological operators, instead, yield Boolean information concerning the structure and the density of edges separating land cover types present in the given neighborhood. In particular, morphological operators, computing, respectively, the erosion and the dilation of a class label either destroy or enhance this kind of information. In short, the relative frequency qualitatively describes the label structure making a sort of spatial average (*low-pass* filtering), while the morphology qualitatively follows the edges (*high-pass* filtering). Hence, both profiles can be considered in some way independent views of the data. On the other hand, spectral and spatial profiles are conditionally independent too, as the spatial correlation on a group of neighbor pixels is not a function of the particular spectral signature.

A final remark concerns the fact that, while spectral features do not change during learning, spatial features, which are constructed through collective inference during transduction, can be updated every time predicted labels are changed under the influence of the collective inference.

4.3. Co-training strategy

A top-level description of the spectral-spatial co-training strategy is reported in Algorithm 1. The algorithm comprises an initialization phase, an iterative co-training phase and a labeling phase. Three

Algorithm 1 Spectral-Spatial Co-Training

Require: $\mathcal{D}$ (imagery data pixels as they are split into $\mathcal{L}$ (labeled set) and $\mathcal{U}$ (unlabeled set)), XY (spatial coordinates of imagery pixels of $\mathcal{D}$), $\mathbf{S}$ (vector of spectral features), $radiusSet$ (set of radius values used to construct pixel neighborhoods), $minTransfer$ (minimum number of pixels transferred from the labeled set to the unlabeled set)

Ensure: $\mathcal{U}$: labeled $\mathcal{U}$

{**initialization phase**}
$\mathcal{LS} \leftarrow \mathcal{LF} \leftarrow \mathcal{LM} \leftarrow \mathcal{L}$; $\mathcal{US} \leftarrow \mathcal{UF} \leftarrow \mathcal{UM} \leftarrow \mathcal{U}$;
$cS \leftarrow$ classifier$(\mathcal{LS}, \mathbf{S})$;
$\mathcal{US} \leftarrow$label$(cS, \mathbf{S}, \mathcal{US})$;
$NMap \leftarrow$neighborhood$(\mathcal{D}, XY, radiusSet)$;
$varMap \leftarrow$ localSpectralVariation$(\mathcal{D}, \mathbf{S}, NMap)$;
$\mathbf{F} \leftarrow$frequencyFeatures$(\mathcal{LS} \cup \mathcal{US}, NMap)$; $\mathbf{M} \leftarrow$morphologicalFeatures$(\mathcal{LS} \cup \mathcal{US}, NMap)$;
$cF \leftarrow$classifier$(\mathcal{LF}, \mathbf{F})$; $cM \leftarrow$classifier$(\mathcal{LM}, \mathbf{M})$
{**co-training loop phase**}
repeat
  $enSF \leftarrow$ ensemble(cS, cF); $enSM \leftarrow$ ensemble(cS, cM); $enFM \leftarrow$ ensemble(cF, cM);
  $[\mathcal{LS}, \mathcal{US}] \leftarrow$classify$(\mathcal{LS}, \mathcal{US}, enFM, \{\mathbf{F}, \mathbf{M}\}, varMap)$;
  $[\mathcal{LF}, \mathcal{UF}] \leftarrow$ classify$(\mathcal{LF}, \mathcal{UF}, enSM, \{\mathbf{S}, \mathbf{M}\}, varMap)$;
  $[\mathcal{LM}, \mathcal{UM}] \leftarrow$ classify$(\mathcal{LM}, \mathcal{UM}, enSF, \{\mathbf{S}, \mathbf{F}\}, varMap)$;
  $\mathbf{F} \leftarrow$frequencyFeatures$(\mathcal{LS} \cup \mathcal{US}, NMap)$; $\mathbf{M} \leftarrow$morphologicalFeatures$(\mathcal{LS} \cup \mathcal{US}, NMap)$;
  $cS \leftarrow$classifier$(\mathcal{LS}, \mathbf{F})$; $cF \leftarrow$classifier$(\mathcal{LF}, \mathbf{F})$; $cM \leftarrow$classifier$(\mathcal{LM}, \mathbf{M})$;
until $((\mathcal{U} = \oslash)$ OR (number of pixels transferred from $\mathcal{US}$ to $\mathcal{LS}$ is less than $minTransfer))$;
{**assigning final labels to pixels of** $\mathcal{U}$}
for $(p \in \mathcal{U})$ **do**
  $pLabel \leftarrow$ majority$(cS, cF, cM, p))$; assign $pLabel$ to p in $\mathcal{U}$;
end for

classifiers are learned through the initialization phase and the iterative co-training phase. These classifiers are then adopted, in the labeling phase, in order to assign each originally unlabeled pixel of the hyperspectral imagery data set to a consensually predicted label. The three classifiers are learned from the spectral profile $\mathbf{S}$ (cS), the frequency-based spatial profile $\mathbf{F}$ (cF) and the morphology-based spatial profile $\mathbf{M}$ (cM) of the imagery data, respectively. In the initialization phase, these three classifiers are learned from the originally labeled part of the input data. In the iterative co-training phase, every classifier is learned from the labeled part of the imagery data, appropriately augmented with unlabeled pixels reliably predicted by an ensemble

of the left-out classifiers. A description of the three phases is reported in the following.

Initialization phase. It consists of six steps (Alg. 1 lines 2-8).

1. Three labeled data sets $\mathcal{L}S$, $\mathcal{L}F$ and $\mathcal{L}M$, as well as three unlabeled data sets $\mathcal{U}S$, $\mathcal{U}F$ and $\mathcal{U}M$ are created (Alg. 1, line 2). They represent labeled data set $\mathcal{L}$ and unlabeled data set $\mathcal{U}$, as they are spanned on spectral profile $\mathbf{S}$ ($\mathcal{L}S$ and $\mathcal{U}S$), spatial frequency-based profile $\mathbf{F}$ ($\mathcal{L}F$ and $\mathcal{U}F$) and spatial morphology-based profile $\mathbf{M}$ ($\mathcal{L}M$ and $\mathcal{U}M$), respectively. These data sets will be modified during the co-training phase.
2. Spectral classifier cS is learned from labeled set $\mathcal{L}S$. This classifier is used to determine initial labels for imagery pixels in unlabeled set $\mathcal{U}S$ (Alg. 1, lines 3-4).
3. The spatial neighborhood structure of imagery data set $\mathcal{D}$ is constructed (Alg. 1 line 5). For each pixel of $\mathcal{D}$, a set of square-shaped spatial neighborhoods is built and associated to the pixel. Each neighborhood is constructed with a specified radius. The set of radius values ($radiusSet$) is a user-defined parameter.
4. The local spatial variation of spectral signatures is calculated for every pixel of $\mathcal{D}$ (Alg. 1, line 6). These values will define local reliability thresholds for the selection criterion of the co-training phase (details in Section 4.4).
5. Spatial features are constructed (Alg. 1, line 7). For this phase, we consider the ground truths for the pixels of the labeled part and labels predicted by the spectral classifier (cS) for the pixels of the unlabeled part. Constructed spatial features populate both frequency-based profile $\mathbf{F}$ and morphology-based profile $\mathbf{M}$, according to the description reported in Section 4.2. These multiple perspectives participate in learning through the co-training system.
6. Two spatial classifiers, cF and cM, are learned from $\mathcal{L}F$ and $\mathcal{L}M$, respectively (Alg. 1, line 8).

Iterative phase. It is produced by the main loop and consists of three steps (Alg. 1, lines 10-17):

1. For each pair of classifiers, their ensemble system is constructed (Alg. 1, line 11) and used to augment the labeled part of the left-out target classifier (Alg. 1, lines 12-14). This is done in accordance with the co-training philosophy, although this philosophy is *generalized* here, in order to deal with multiple classifiers. In particular, the reference ensemble composed of cF and cM is used to determine the labels of $\mathcal{U}S$ (Alg. 1, line 12). The reference ensemble composed of cS and cM is used to determine the labels of $\mathcal{U}F$ (Alg. 1, line 13). The reference ensemble composed of cS and cF is used to determine the labels of $\mathcal{U}M$ (Alg. 1, line 14). Each ensemble works as follows. For every pixel of the unlabeled part of the target classifier, a label is predicted by performing the classification with each classifier in the reference ensemble and choosing the label predicted with the highest posterior probability. The optimal posterior probabilities are determined by using the Platt's method [42]. Subsequently, the

reliability of the labels consensually predicted by the reference ensemble is evaluated (see details in Section 4.4). Finally, pixels associated with reliable labels are transferred from the unlabeled part to the labeled part of the target classifier. A filtering criterion, called diversity class criterion (DCC), can be applied, in order to "smartly" sample the reliable examples, which are actually transferred from the unlabeled part to the labeled part of the target classifier. This criterion aims at reducing the size of the labeled parts maintained for the considered profiles, in order to scale-up the computation time spent learning new classifiers from the augmented labeled sets. By applying the diversity class criterion, a pixel is actually transferred only if its label is reliably predicted by the reference ensemble and this ensemble label is different from the original label predicted by the corresponding target classifier.[1]

2. The spatial features of both $\mathbf{F}$ and $\mathbf{M}$ are updated according to the new labels injected into $\mathcal{L}S$ (Alg. 1, line 15).
3. Classifiers cS, cF and cM are re-learned from augmented $\mathcal{L}S$, $\mathcal{L}F$ and $\mathcal{L}M$, respectively (Alg. 1, line 16).

This iterative inference stops (Alg. 1, line 17) when the unlabeled set of the spectral profile ($\mathcal{U}S$) is empty or the number of pixels transferred from the unlabeled set of the spectral profile ($\mathcal{U}S$) to the labeled set of the spectral profile ($\mathcal{L}S$) is less than a certain threshold ($MinTransfer$). By default, this threshold is equal to 10 pixels.[2] The iterative inference procedure is guaranteed to converge as eventually one of the stopping criteria will be satisfied. If each iteration transfers more than $MinTransfer$ pixels from $\mathcal{U}S$ to $\mathcal{L}S$ (Alg. 1, line 12), then $\mathcal{U}S = \oslash$; and the first condition is satisfied. Otherwise, if the number of pixels, transferred from $\mathcal{U}S$ to $\mathcal{L}S$, at the present iteration, is less than $MinTransfer$, the second stopping condition is satisfied.

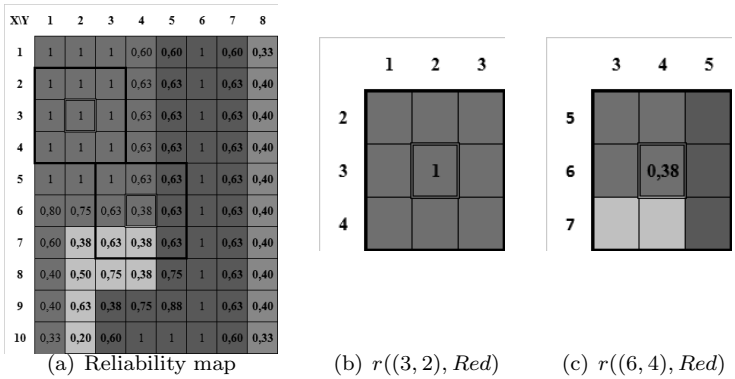
Labeling phase. It assigns each originally unlabeled pixel of the hyperspectral image to the class consensually predicted by the majority of classifiers (spectral, as well as spatial) learned in the last iteration of the co-training phase (Alg. 1, lines 19-21). Whenever, for a given pixel, three distinct labels are predicted by the considered classifiers, the label predicted with the highest posterior probability is assigned to the pixel. The optimal posterior probability of a predicted label is determined by using the Platt's method [42].

4.4. Spatial example selection

An unlabeled pixel is selected for the augmentation of a training set based on the estimate of the reliability of its predicted label. The reliability of a label is here estimated by measuring the local spatial variation of

[1] The trade-off between the efficiency and the accuracy of the classification due to the diversity class criterion will be empirically investigated in Section 5.2.

[2] The influence of $MinTransfer = 10$ on the accuracy of the classification will be empirically investigated in Section 5.3.



(a) Reliability map (b) $r((3,2), Red)$ (c) $r((6,4), Red)$

Figure 2: Reliability map for the ground truth labels: 2(b) reliability of the red label assigned to the pixel (3,2) (i.e. $r((3,2), Red) = 1 - \frac{0}{8}$ over $\mathcal{N}((3,2),1)$), which falls in the internal part of the red colored region and 2(c) reliability of the red label assigned to the pixel (6,4) (i.e. $r((6,4), Red) = 1 - \frac{5}{8}$ over $\mathcal{N}((6,4),1)$), which falls on the boundary between the red colored region, the blue colored region and the yellow region. The reliability is measured over neighborhoods constructed with radius equal to 1.

the label itself. The lower the variation, the higher the reliability.[3]As a measure of the local spatial variation of a label, we measure how common the label is over a neighborhood. For this calculation, we focus the search over the smallest neighborhoods of every pixel, which were already used for the construction of the spatial feature spaces. Formally, let p be a pixel of the unlabeled part, C be the label predicted for p, $\mathcal{N}(p, R_{min})$ be the neighborhood of p constructed according to the theory reported in Section 3 and with radius $R_{min} = \min_{R \in RadiusSet} R$, the reliability $r(p, C)$ is computed as follows:

$$r(p, C) = 1 - \frac{|\{q \in \mathcal{N}(p, R_{min}), q \neq p | label(q) \neq C\}|}{|\mathcal{N}(p, R_{min})| - 1}, \tag{1}$$

where $label(q)$ is either the label already assigned to q in the labeled part or the label predicted for q in the unlabeled part. This reliability measure takes values in the range $[0, 1]$. 0 suggests an outlier classification, while 1 suggests an inlier classification coherently with the property of positive spatial correlation.

In this study, we use a threshold-based strategy, in order to identify labels which are reliable enough for the augmentation operation. A pixel whose predicted label reliability is greater than a threshold is transferred from the unlabeled part to the labeled one. The selection of the threshold value is the real challenge of this criterion. The main difficulty lies in the fact that the reliability measure of an inlier class exhibits discontinuous values when calculated for pixels falling on the boundary between distinct regions

[3]The idea of a spatial criterion to estimate the reliability of a predicted class has been investigated in a few active learning algorithms recently defined in the hyperspectral classification literature [40, 41]. In the active learning algorithms, predicted class reliability is estimated, in order to select examples which are *inappropriately* predicted by the learned hypothesis. In this study, the scope is different. In fact, we intend to select labels which are *appropriately* predicted by the learned hypotheses and use them to augment the labeled sets managed through the co-training paradigm.

(see Figure 2). Accordingly, the threshold should be locally estimated, rather then globally selected. To estimate this local threshold, we take into account the fact that the label is expected to be a function of the spectral signature. This is a common hypothesis that already motivates any step of supervised learning performed with the use of the spectral feature space. Based upon this hypothesis, we can reasonably expect that the local spatial variation of the labels is consistent with the local spatial variation of the spectra. Therefore, we compute the local spatial dispersion of the spectra though the imagery data and use these local estimates, in order to define local thresholds for the decision on the label reliability. Operatively, for each pixel p of the imagery data, we compute the local spatial dispersion of the spectral signature at p as $disp(p,\mathbf{S}) = \sqrt{\frac{\sum_{q\in\mathcal{N}(p,R_{min}),q\neq p} d^2(p,q,\mathbf{S})}{|\mathcal{N}(p,R_{min})|-1}}$, where $d(p,q,\mathbf{S})$ is the Euclidean spectral distance computed between p and q as both are spanned on features of the spectral space. Since spectral features may have different ranges, they are all scaled between 0 and 1 for the computation of this local spatial variation measure. Finally, the local reliability threshold at pixel p is determined as follows:

$$rThr(p) = (1 - disp(p,\mathbf{S}))|_{scaled}, \tag{2}$$

where the $1 - disp(p,\mathbf{S})$ is scaled into $[\min_{p\in\mathcal{U}_{current}} r(p, label(p)) \ , \max_{p\in\mathcal{U}_{current}} r(p, label(p))]$.

Final remarks concern the fact that the presented example selection schema resorts to spectral and label information synthesized on a neighborhood object. From this point of view, this example selection procedure contributes to characterizing this algorithm as a hybrid classification approach (i.e. a fusion of pixel-based and object-based classification).[4] Various studies in hyperspectral image analysis, in particular for urban land cover classification (e.g. [47, 3]), have already claimed that hybrid approaches can gain classification accuracy by dealing with spectrum discontinuity along theme boundaries. Therefore, based upon these studies, we expect that this spatial example selection schema would contribute to gaining accuracy, in particular along the boundaries, with respect to an aspatial schema (see Section 5.3.2 on "Example selection criterion" for an empirical investigation of this aspect).

4.5. Learning complexity

For this analysis, we consider that: (1) $N^{\mathcal{L}}$ denotes the number of labeled pixels ($N^{\mathcal{L}} = |\mathcal{L}|$), while $N^{\mathcal{U}}$ denotes the number of unlabeled pixels ($N^{\mathcal{U}} = |\mathcal{U}|$) of $\mathcal{D}$, so that $N = N^{\mathcal{L}} + N^{\mathcal{U}}$; (2) r is the number of neighborhoods constructed for each pixel of $\mathcal{D}$; (3) R_{max} (R_{min}) is the radius of the largest (smallest) neighborhood, so that $(2R_{max}+1)^2 = 4R_{max}^2 + 4R_{max} + 1$ ($(2R_{min}+1)^2 = 4R_{min}^2 + 4R_{min} + 1$) is the

[4]The pixel-based classification is performed with spectral information, while the object-based classification is performed with the neighborhood-based spatial information. However, differently from our previous work in [20, 1], where the spatial information is synthesized for populating spatial profiles only, here it is also accounted for determining reliably predicted examples, in order to reduce the number of pixels misclassified along the boundaries between various thematic objects.

maximum (minimum) number of pixels grouped per neighborhood; (4) $nIter$ is the number of iterations performed in the co-training phase; (5) k is the number of class labels; (6) m is the number of spectral features, kr is the number of frequency-profile features, $4kr$ is the number of morphology-profile features, while $M = \max\{m, 4kr\}$; (7) $\Lambda(|Data|, |FeatureSpace|)$ denotes the cost of learning a supervised classifier[5] from a training set $Data$, as it is spanned on a feature space $FeatureSpace$. Based on these premises, the computational complexity of S^{2}CoTraC is computed by summing up the cost of the initialization phase, the co-training phase and the labeling phase.

Initialization phase. The time cost of creating a copy of both the labeled set and the unlabeled set for each profile is $3N$, that is, $\mathrm{O}(N)$. The time cost of constructing the neighborhood structure is $(4R^2_{max} + 4R_{max} + 1)N$, that is, $\mathrm{O}(R^2_{max}N)$. The time cost of measuring the local spatial dispersion of spectral signatures is $m(4R^2_{min} + 4R_{min})N$, that is $\mathrm{O}(mR^2_{min}N)$. The time cost of learning the spectral classifier from the initial labeled set $\mathcal{L}$ is $\Lambda(N^{\mathcal{L}}, m)$, while the time cost of learning the spatial frequency-profile classifier is $\Lambda(N^{\mathcal{L}}, kr)$ and the time cost of learning the spatial morphology-profile classifier is $\Lambda(N^{\mathcal{L}}, 4kr)$. Hence, the total cost of initializing the three classifiers is $\mathrm{O}(\Lambda(N^{\mathcal{L}}, M))$. The time cost of constructing the spatial features by using both the frequency operator and the morphological operators is $5kr(4R^2_{max} + 4R_{max} + 1)N$, that is, $\mathrm{O}(krR^2_{max}N)$. Therefore, the time complexity of the initialization phase is $\mathrm{O}(N + R^2_{max}N + mR^2_{min}N + \Lambda(N^{\mathcal{L}}, M) + krR^2_{max}N)$, that is, $\mathrm{O}(mR^2_{min}N + \Lambda(N^{\mathcal{L}}, m) + (kr)R^2_{max}N)$, with $kr > 1$.

Co-training phase. The co-training phase is performed by considering each pair of reference classifiers on one side and the left-out target classifier on the other side of the co-training system. Co-training is iterated per $nIter$ number of times. The time cost of constructing the reference ensemble from a pair of selected classifiers and using it to label pixels of the unlabeled part of the target classifier is $2N^{\mathcal{U}}$, at worst. The time cost of computing the local reliability of the predicted labels over the smallest neighborhoods is $(4R^2_{min}+4R_{min})N^{\mathcal{U}}$, at worst, while the time cost of transferring pixels from the unlabeled part to the labeled part is $N^{\mathcal{U}}$, at worst. Therefore, the time complexity spent, in order to predict labels and move pixels from the labeled part to the unlabeled part of a target classifier, is $2N^{\mathcal{U}} + (4R^2_{min} + 4R_{min})N^{\mathcal{U}} + N^{\mathcal{U}}$, that is $\mathrm{O}(R^2_{min}N^{\mathcal{U}})$, at worst. This cost is multiplied by three, that is, the number of combinations of reference ensemble classifiers vs target classifier to be evaluated. The time complexity of updating the spatial features is $\mathrm{O}(krR^2_{max}N)$, while the time cost of updating spectral and spatial classifiers is $\mathrm{O}(\Lambda(N^{\mathcal{L}}, M))$. Hence, the total cost of performing the co-training phase is $nIter(3R^2_{min}N^{\mathcal{U}}+krR^2_{max}N+\Lambda(N^{\mathcal{L}}, M)\,)$, that is, $\mathrm{O}(nIter\ krR^2_{max}N+nIter\Lambda(N^{\mathcal{L}}, M))$ by considering that $R_{min} < R_{max}$, $N^{\mathcal{U}} < N$ and $kr > 3$.

Labeling phase. The labeling phase uses an ensemble of the three classifiers learned in the last iteration of the co-training phase, in order to classify initially unlabeled pixels of the imagery data. The time cost of

[5]This cost depends on the algorithm selected as the base classifier.

this phase is $3N^{\mathcal{U}}$, that is $\mathrm{O}(N^{\mathcal{U}})$.

Total complexity. The total time cost of the algorithm is:

$$\underbrace{mR^2_{min}N + \Lambda(N^{\mathcal{L}}, m) + (kr)R^2_{max}N}_{inizialization} + \underbrace{nIter\ krR^2_{max}N + nIter\Lambda(N^{\mathcal{L}}, M)}_{co-training} + \underbrace{N^{\mathcal{U}}}_{labeling},$$

that is, $\mathrm{O}(mR^2_{min}N + nIter(krR^2_{max}N + \Lambda(N^{\mathcal{L}}, M)))$

5. Experimental evaluation and discussion

S^{2}CoTraC, whose implementation is publicly available,[6] is written in Java. It integrates the inductive Support Vector Machine (SVM) [7] [13] algorithm as a base classifier of the transductive co-training system. This choice is motivated by several studies reported in the literature (e.g. [43, 16]), which show that inductive SVMs are applied to hyperspectral image classification with great success. In fact, they outperform several other inductive classifiers. The empirical results are organized as follows. We start (Subsection 5.2) by comparing the accuracy of the presented transductive algorithm to that of inductive and transductive competitor hyperspectral classifiers. We proceed (see Subsection 5.3) by evaluating the sensitivity of the accuracy of the presented algorithm along the parameter configuration. Then we analyze the accuracy of various semi-supervised and transductive algorithms described in the machine learning literature (see Subsection 5.4). Finally, we report a brief discussion of additional recent evaluation results reported in the hyperspectral image classification literature (see Subsection 5.5). We perform this study, in order to seek answers to the following questions: (1) Is the defined transductive schema more accurate than the base inductive learner, the transductive approaches that do not use the collective inference, as well as the transductive approaches that do not use the co-training paradigm (see the comparative analysis in Section 5.2)? (2) How does the performance of the classification change by varying the number of performed iterations (see Section 5.3)? (3) Is the classification robust enough to a change in the size of the initial labeled set and the size of the spatial neighborhoods (see the sensitivity study in Section 5.3)? (4) How do the individual components (spatial profiles, spatial selection schema) of the co-training schema affect its overall accuracy (see the sensitivity study in Section 5.3)? (5) How does the schema's accuracy compare to the state-of-the-art general-purpose semi-supervised/transductive classifiers defined in the machine learning literature (see the semi-supervised/transductive literature study in Section 5.4), as well as to the state-of-the-art application-specific classifiers defined in the hyperspectral imaging literature (see the hyperspectral literature study in Section 5.5)?

[6] `http://www.di.uniba.it/~appice/software/S2COTRAC/`

[7] We use the Java implementation of SVM included in the WEKA toolkit [59]

5.1. Hyperspectral image data sets

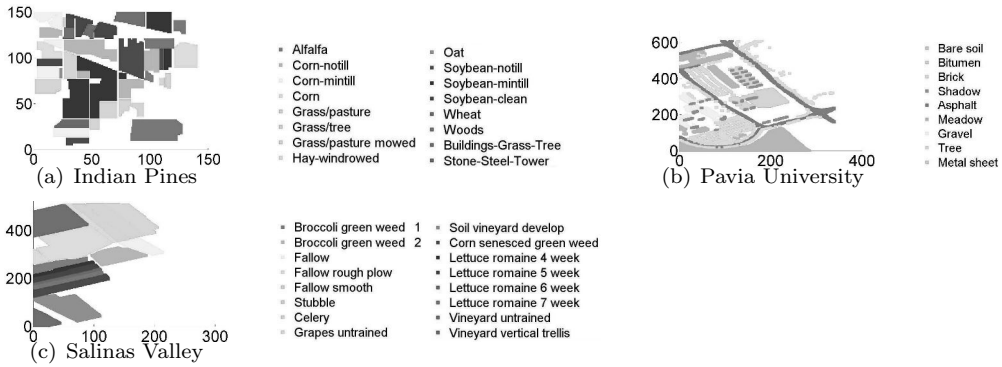
Three real data sets, namely Indian Pines, Pavia University and Salinas Valley,[8] are used for the comparative analysis reported in this experimental study. These data sets are selected for the following reasons: (1) They contain rich spectral information (100-200 bands) and a reasonable number of classes (9-16 classes).(2) They correspond to different scenarios. (3) Ground truths are available for these data.[9] Additionally, they are still considered in the majority of recent, relevant works on hyperspectral image classification (e.g. [33, 31, 32, 55, 20, 1, 34, 12, 21]). In particular, the Indian Pines data set is also used in the sensitivity study, where we evaluate the performances of the presented algorithm along the parameter configuration. This data set is selected as it is a very challenging image, due to the significant presence of mixed pixels in all the available classes and also to the unbalanced number of available labeled pixels per class.

AVIRIS Indian Pines (10249 pixels) [29]. It was obtained by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor over the Indian Pines region in Northwestern Indiana in 1992. The image contains 220 spectral bands, but 20 spectral bands have been removed due to the noise and water absorption phenomena. The spatial resolution is of 20 m and the spatial size is of 145 $\times$ 145 pixels, which are classified into 16 mutually exclusive classes distributed as follows: Alfalfa (0.45%), Corn-notill (13.93%), Corn-mintill (8.10%), Corn (2.31%), Grass/pasture (4.71%), Grass/tree (7.12%), Grass/pasture-mintill (0.27%), Hay-windrowed (4.66%), Oat (0.20%), Soybean-notill (9.48%), Soybean-mintill (23.95%), Soybean-clean (5.79%), Wheat (2.0%), Woods (12.34%), Buildings-Grass-Trees-Drives (3.77%) and Stone-Steel-Towers (0.91%). The map of the class ground truth is shown in Figure 3(a). As reported in [43], this data set represents a very challenging land-cover classification scenario, in which the primary crops of the area (mainly corn and soybeans) were very early in their growth cycle, with only about 5% of canopy cover. Discriminating among the crops under these circumstances can be a very difficult task. This scenario is also made more complex by the imbalanced number of available labeled pixels per class.

ROSIS Pavia University (42776 pixels) [18]. It was obtained by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign over the Engineering School at the University of Pavia in 2003. Water absorption bands were removed and the original 115 bands were reduced to 103 bands. It has a spatial resolution of 1.3 m. The image has a spatial size of 610 $\times$ 340 pixels, which are classified into 9 classes distributed as follows: Asphalt (15.5%), Meadow (43.6%), Gravel (4.91%), Tree (7.16%), Metal sheet (3.14%), Bare soil (11.76%), Bitumen (3.11%), Brick (8.61%) and Shadow (2.21%) . The map of the class ground truth is shown in Figure 3(b).

[8] `http://www.grss-ieee.org/community/technical-committees/data-fusion/`

[9] Although the data acquisition can be a relatively easy process, the generation of a reliable ground truth is a very expensive process.



(a) Indian Pines (b) Pavia University (c) Salinas Valley

Figure 3: Indian Pines, Pavia University and Salinas Valley: class ground truth (3(a)-3(c).

AVIRIS Salinas Valley (54129 pixels) [28]. It was collected by AVIRIS over Salinas Valley, Southern California, in 1998. It has a spatial resolution of 3.7 m. The area contains a spatial size of 512 × 217 pixels and 206 spectral bands. The 20 water absorption bands are discarded. Pixels are classified into 16 classes distributed as follows: Broccoli green weeds 1 (3.71%), Broccoli green weeds 2 (6.88%), Fallow (3.65%), Fallow rough plow (2.58%), Fallow smooth (4.95 %), Stubble (7.31%), Celery (6.61%), Grapes Untrained (20.82%), Soil vineyard develop (11.46%), Corn senesced green weeds (6.06%), Lettuce romaine 4 weeks (1.97%), Lettuce romaine 5 weeks (3.56%), Lettuce romaine 6 weeks (1.69%), Lettuce romaine 7 weeks (1.98%), Vineyard untrained (3.34%) and Vineyard vertical trellis (13.43%). The map of the class ground truth is shown in Figure 3(c).

To perform the empirical evaluation, these data sets are divided into labeled data sets and unlabeled data sets. For each data set, we consider a subset of ground truths to populate the labeled set. The labeled pixels are randomly selected from the available ground truth of the image, by using stratified random sampling without replacement. In this way, for each class, the number of pixels randomly sampled for the labeled set is proportional to the number of pixels labeled with the selected class in the ground truth map. The remaining pixels are used as the unlabeled part of the learning process. In this study, various partitioning trials between labeled and unlabeled sets are generated; the classification process is evaluated on these trials. This setting is usual in hyperspectral image classification [43].

5.2. Comparative analysis

For this study, we consider Indian Pines, Pavia University and Salinas Valley.

5.2.1. Experimental set-up

We compare S^2CoTraC to the inductive SVM, to the Spatio-Spectral Transductive Ensemble Classifier (S^2Tec) [1], to the Iterative Relational Multinomial Classifier (irmc) [20], to the Spatio-Spectral TriTraining

Classifier (TriTraining) [22],[10] as well as to the Modified Co-Training With Gaussian Process Classifier (mcogpc) [61] (see Table 1, Section 2). Our algorithm has been run without the diversity class criterion (S^2CoTraC), as well as with the diversity class criterion (S^2CoTraC-DCC). S^2CoTraC, S^2Tec and irmc have been run with the size of neighborhoods growing from 5 to 10 and 15. Remaining competitors have been run with the optimal parameter setup suggested by the authors in the corresponding works.

For this study, S^2CoTraC, inductive SVM, S^2Tec, irmc and mcogpc have been evaluated by using 5% of the ground truth as the labeled set of the transductive learning phase. Five random partitioning trials between labeled and unlabeled sets have been generated; metrics have been averaged on these trials and standard deviation has been computed. As the implementation of these algorithms has been made available by the authors, they have been evaluated by using the same division between labeled samples and unlabeled samples for each running trial. For TriTraining, we have considered the evaluation results reported in [22]. These results have been collected by using subsets of both Indian Pines and Pavia University. In particular, in the experiment on Indian Pines, from the 16 different classes (see Figure 3(a)), 7 were discarded since the authors judged that an insufficient number of training samples was available. The remaining 9 classes (Corn-notill, Corn-mintill, Grass/pasture, Grass/tree, Hay-windrowed, Soybean-notill, Soybean-mintill, Soybean-clean and Woods) have been used. In the experiment on Pavia University, from the 9 different classes (see Figure 3(b)), 3 classes were discarded. The remaining 6 classes (Asphalt, Meadows, Gravel, Trees, Bricks and Shadow) have been projected on the 200 $\times$ 200 pixel bottom scene of the ROSIS image for Pavia University and used for the evaluation. For both these reduced data sets, accuracy of TriTraining has been averaged on five trials generated for three settings using 2%, 5% and 10% of the ground truth.[11]

We evaluate the accuracy performances of the compared algorithms in terms of overall accuracy (OA), average accuracy (AA) and Cohen's kappa coefficient (κ) [45]. These metrics are defined in terms of elements x_{ij} of the error matrix associated to the classified imagery pixels. Each element x_{ij} denotes the number of imagery pixels with ground truth c_j, which are labeled with class c_i. C is the number of distinct classes in the image. Let us consider $x = \sum_h x_{hh}$, (i.e. the number of pixels, which are correctly labeled - true positive), $x_{h+} = \sum_l x_{hl}$, (i.e. the number of pixels which are labeled with class h - true positive+false positive of class h), $x_{+h} = \sum_l x_{lh}$ (i.e. the number of pixels of class h - true positive + false negative of class h) and $N = \sum_i \sum_j x_{ij}$ (i.e. the number of pixels). Then, $OA = \frac{x}{N}$ $AA = \frac{1}{C} \sum_h \frac{x_{hh}}{X_{h+}}$ $\kappa = \frac{N x_i - \sum_i x_{h+} x_{+h}}{N^2 - \sum_i x_{h+} x_{+h}}$

Metrics OA, AA and κ are selected as they are, usually, considered by the hyperspectral image classifi-

[10] A version of TriTraining with the construction of collective features is evaluated in the sensitivity study reported in Section 5.3 (see the results achieved with the learning schema Aspatial in Table 5).

[11] We note that results reported in [22] have been achieved by starting from labeled samples, generated with the same stratified sampling strategy used here, but which may be different from those considered for evaluating S^2CoTraC and S^2CoTraC-DCC.

Table 2: Overall Accuracy (OA), Average Accuracy (AA) and κ: S^2CoTraC, S^2CoTraC (DCC), SVM, S^2Tec, irmc and mcogpc. Accuracy metrics are collected on five trials produced by considering 5% of ground-truth data as the labeled set. The mean (± standard deviation) of each metric is computed on the unlabeled set of these trials. The highest accuracy is in bold.

algorithm	OA	AA	κ	OA	AA	κ
	Indian Pines			Pavia University		
S^2CoTraC	**.964±.008**	**.905±.040**	**.959±.009**	**.994±.003**	**.987±.004**	**.992±.004**
S^2CoTraC-DCC	.961± .009	.898±.033	.956± .010	.985 ±.009	.958± .008	.980±.012
SVM	.739±.011	.626±.012	.700±.013	.931±.002	.912±.004	.909±.003
S^2Tec	.936±.011	.874±.031	.927±.012	.988±.009	.982±.003	.984±.001
irmc	.873±.031	.821±.075	.856±.035	.867±.008	.837±.007	.814±.010
mcogpc	.828±.007	.692±.027	.803±.008	.947±.003	.906±.009	.929±.004
	Salinas Valley					
S^2CoTraC	**.994±.004**	**.997±.001**	**.994±.005**			
S^2CoTraC-DCC	.993± .004	.996±.001	.993± .0048			
SVM	.925±.002	.957±.002	.916±.003			
S^2Tec	.970±.006	.984±.003	.967±.006			
irmc	.955±.018	.952±.016	.950±.020			
mcogpc	.957±.001	.976±.001	.952±.001			

Table 3: Overall Accuracy (OA) and κ: S^2CoTraC, S^2CoTraC-DCC and TriTraining. Accuracy metrics are collected on five trials produced by considering 2%, 5% and 10% of ground-truth data as the labeled set, as reported in [22]. The mean (± standard deviation) of each metric is computed on the unlabeled set of these trials. The accuracy metrics of TriTraining are reported in [22]. The Average Accuracy (AA) of TriTraining is not considered in this study, as it is not reported in [22].

imagery data	%	S^2CoTraC		S^2CoTraC-DCC		TriTraining	
		OA	κ	OA	κ	OA	κ
Indian Pines	2%	.911±.027	.896±.031	.909±.023	.893±.027	.822	.812
9 classes	5%	.974±.011	.970±.013	.972±.014	.967±.016	.887	.886
145 × 145 pixels	10%	.984±.008	.981±.010	.985±.007	.981±.008	.931	.930
Pavia University	2%	.970±.008	.953±.012	.967±.008	.948±.013	.948	.909
6 classes	5%	.989±.006	.983±.010	.984±.008	.975±.013	.960	.937
200 × 200 pixels	10%	.996 ±.006	.994±.001	.996±.0008	.993±.001	.968	.950

cation community, as well as by the machine learning community. For each accuracy metric (OA, AA and κ), the higher the metric, the more accurate the classifier.

5.2.2. Results and discussion

The accuracy metrics for S^2CoTraC, S^2CoTraC-DCC, inductive SVM, S^2CoTec, irmc and mcogpc are reported in Table 2, while the metrics for S^2CoTraC and TriTraining are reported in Table 3. We recall that the metrics collected in Table 3 refer, respectively for the Indian Pine and Pavia University data sets, to 9 and 6 classes, contrary to 16 and 9 classes collected in Table 2. The simpler classification scenario should lead to an increase of OA. As expected, the OA increases for Indian Pines (from .964±.008 to .974±.011 for S^2CoTraC and from .961±.009 to .972±.014 for ^{2}CoTraC-DCC), but decreases for Pavia University (from .994±.003 to .989±.006 for S^2CoTraC and from .993±.004 to .984±.008 for S^2CoTraC-DCC). However, the difference of scene must be taken into account. The problems formulated with 16 and 9 classes for Indian

Pines are both spanned across the entire scene (see 145×145 pixels in Figure 3(a)). On the contrary, the problem formulated with 6 classes for Pavia University is spanned across the 200×200 pixel bottom scene of the ROSIS image of Pavia University (the ROSIS image covers 610×340 pixels as reported in Figure 3(b)). This means that there are pixels of the original Pavia University scene that may belong to the selected class set, but are neglected as they are outside the selected scene. This makes unfair the comparison between results produced for these two formulations of the Pavia University classification problem.

The analysis of the metrics collected in Tables 2 and 3 deserves several considerations.

Firstly, the construction of collective spatial features (S^2CoTraC, S^2CoTraC-DCC and S^2TeC), which change during iterative learning, gains accuracy with respect to the inductive SVM learner, that neglects the spatial information, as well as to the transductive learners (mcogpc and TriTraining), that construct spectral spatial features, which do not change during iterative learning. The accuracy metrics confirm the intuition reported in [22], that classification accuracy can be improved by considering various spatial profiles of the imagery data. In fact, the accuracy of the algorithms (S^2CoTraC, S^2CoTraC-DCCand S^2TeC), which integrates two different spatial profiles, is always better than the accuracy of their competitors (irmc and mcogpc), which learn imagery data along a single spatial profile. The sensitivity of the accuracy of S^2CoTraC and S^2CoTraC-DCC along the number of constructed spatial profiles is analyzed in Section 5.3.

Secondly, the co-training strategy described in this paper (S^2CoTraC and S^2CoTraC-DCC) generally performs better than the ensemble strategy (S^2Tec and irmc) and the co-training strategy (TriTraining and mcogpc) described in the literature. We note that, by following the literature, the ensemble strategy of the competitors is combined with the construction of collective spatial features, while the co-training strategy of competitors is combined with the construction of spectral spatial features. Therefore, this comparative analysis actually shows that the accuracy of the co-training strategy can be improved by the use of collective inference and the co-training strategy outperforms the ensemble strategy when both use collective inference. The co-training algorithms compared in this study adopt different example selection criteria. The sensitivity of the accuracy of co-training with collective inference along the example selection criterion (spatial vs aspatial) is analyzed in Section 5.3.

Finally, the spatial example selection criterion (S^2CoTraC and S^2CoTraC-DCC) described in this paper generally performs better than the aspatial criteria described in the literature (S^2Tec, irmc, mcogpc, TriTraining). On the other hand, the use of the diversity class criterion in combination with the spatial example selection criterion slightly diminishes the classification accuracy (S^2CoTraC-DCC is slightly less accurate than S^2CoTraC), although S^2CoTraC-DCC is still more accurate than the competitors except for S^2Tec in Pavia University. As the diversity class criterion is introduced to speed-up the iterative learning process, we also evaluate the performance of S^2CoTraC, S^2CoTraC-DCC and S^2Tec[12] in terms of a trade-off between

[12]They are the most accurate algorithms in this study. Their learning times can be safely compared as they are implemented

Table 4: Learning time (in secs) and total number of pixels transferred from the unlabeled sets to the labeled sets during the iterative process and used to learn the final classifiers: S^2CoTraC, S^2CoTraC-DCC and S^2Tec. The number of transferred pixels is reported per profile (F - frequency, M - morphology, S - spectral signature). The metrics are collected and averaged on five trials produced by considering 5% of ground-truth data as the labeled set. The lowest metrics are in bold.

algorithm	learning time	pixels (F)	pixels (M)	pixels(S)	learning time	pixels (F)	pixels (M)	pixels(S)
	Indian Pines				Pavia University			
S^2CoTraC	18051.20	8589	8694	8689	186375.00	39977	40034	40036
S^2CoTraC-DCC	**3621.21**	**230**	**579**	**2795**	**42539.01**	**4851**	**5112**	**7870**
S^2Tec	28555.25	9729	9729	9729	186740.277	40627	40627	40627
	Salinas Valley							
S^2CoTraC	204266.50	50632	50795	50817				
S^2CoTraC-DCC	**70019.61**	**2347**	**4838**	**13220**				
S^2Tec	231226.00	51423	51423	51423				

accuracy and efficiency. Table 4 collects the learning time, as well as the number of examples transferred from the unlabeled sets to the labeled sets per profile. We observe that S^2CoTraC-DCC spends less time than the competitors (S^2CoTraC and S^2CoTeC) to complete the iterative learning process. This depends on the fact that S^2CoTraC-DCC transfers a lower number of examples than S^2CoTraC and S^2TeC during the iterative learning, so that it deals with smaller training sets to learn spectral and spatial classifiers. In any case, the high accuracy achieved by S^2CoTraC-DCC confirms that it is still able to sample examples that are those that actually contribute to increasing the accuracy of the classifiers.

5.3. Sensitivity analysis

For this analysis, we consider the Indian Pines data set.

5.3.1. Experimental set-up

We perform a sensitivity analysis of the performance of the proposed algorithm along the size of the initial labeled set, the size of the spatial neighborhoods, the example selection schema, the number of spatial profiles, as well as the number of iterations. We analyze the accuracy (OA, AA and κ) of classification, the number of examples transferred from the unlabeled sets to the the labeled sets per profile, the learning time and the number of iterations performed to complete the task. As five partitioning trials between labeled and unlabeled sets are generated for each data set, both the mean and the standard deviation of the considered metrics are computed on these running trials.

5.3.2. Results and discussion

Size of original labeled sets. We vary the percentage of pixels which are labeled in the image between 3%, 5% (baseline) and 10%, while we run co-training by constructing neighborhoods with sizes growing from 5 to

in a Java environment that resorts to the same libraries to represent data, compute profiles and learn SVM classifiers.

Table 5: Sensitivity study (Indian Pines): the accuracy metrics are collected from five trials; the average (± standard deviation) of the measures is computed on these trials.

labeled %	OA	AA	κ	Selection	OA	AA	κ
3%	.955± .011	.872±.017	.949±.012	Aspatial [22]	.952±.009	.888±.026	.945±.011
5%	.961± .009	.898±.033	.956± .010	Spatial	.964± .008	.905±.040	.959± .009
10%	.980± .007	.927±.030	.977±.009	Spatial+DCC	.961± .009	.898±.033	.956± .010
Neighborhood	OA	AA	κ	Learning/Profile schema	OA	AA	κ
5	.942 ±.020	.887±.040	.934±.228	Self-SVM(**S**)	.804 ±.010	.621 ±.034	.774 ±.011
5,10	.955± .022	.895±.041	.949±.025	SVM(**S**)+SVM(**FM**)	.922 ±.015	.797 ±.036	.911±.017
5,10,15	.961± .009	.898±.033	.956± .010	**S**+**F**Co-Training	.948±.009	.862±.034	.941±.011
5,10,15,20	.965± .013	.893±.960	.960±.015	**S**+**M**Co-Training	.950±.009	.865±.036	.943±.010
5,10,15,20,25	.962±.012	.886±.037	.956±.014	**S**+**F**+**M**Co-Training	.961± .009	.898±.033	.956± .010

10 and 15. We use both spectral, spatial-frequency and spatial-morphology profiles and apply the diversity class criterion. The computed metrics are reported in Tables 5 (rows 2-4, columns 2-4) and 6 (rows 2-4) The results show that the classification process gains in accuracy and efficiency by augmenting the number of pixels in the originally labeled set. The learning process is accelerated since starting from a larger labeled set allows the classification process to diminish the number of examples transferred from the unlabeled sets to the labeled sets during the iterations.

Size of neighborhoods. We construct neighborhoods with sizes: 5, 5-10, 5-10-15 (baseline), 5-10-15-20 and 5-10-15-20-25, while we run co-training with the initial labeled set sampled with the labeling percentage equal to 5%. We use both spectral, spatial-frequency and spatial-morphology profiles and apply the diversity class criterion. The computed metrics are reported in Tables 5 (rows 6-10, columns 2-4) and 6 (rows 6-10). The results show that the classifier gains accuracy by augmenting the number of neighborhoods and enlarging their size. In fact, in this way, we increase the chances of building spatial features that better fit the spatial variation of classes, even when classes vary over space with different density and granularity.[13] In any case, the classification accuracy does not change greatly when the neighborhood size is greater than 15. The learning process is completed in six iterations on average, independently of both the number and the size of the neighborhoods used to construct the spatial features, while the learning time spent to complete the task increases with the size of the neighborhoods.

Example selection criterion. We compare the performance of the spatial example selection criterion (Spatial), the spatial example selection criterion combined with the diversity class criterion (Spatial+DCC) and the aspatial example selection criterion described in [22] .[14] We construct neighborhoods with size growing

[13] Determining the optimal size and number of neighborhoods is an open issue whose investigation is out of the scope of this paper.

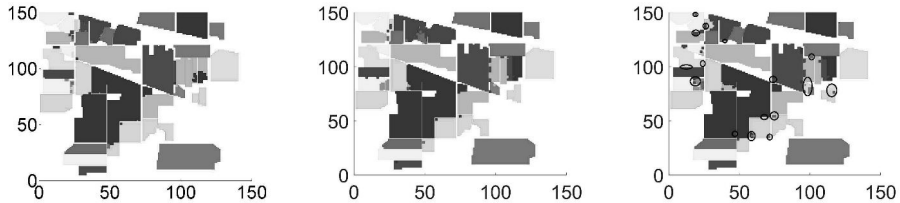
[14] The Aspatial criterion considers a predicted label reliable for a target classifier if it is equally predicted by both classifiers included in the reference ensemble.

Table 6: Sensitivity study (Indian Pines): Total number of examples transferred from the unlabeled sets to the labeled sets during the iterative process and used to learn the final classifiers, number of performed iterations and learning times (in secs). The number of transferred examples is reported per profile (F - frequency, M - morphology, S - spectral signature). The metrics are collected from five trials; the average (± standard deviation) of the measures is computed on these trials.

labeled %	examples (F)	examples (M)	examples(S)	nIter	time
3%	390.6± 78.09	975.6±36.38	3459.20±190.38	6.2±.788	4332.554±564.244
5%	230.6± 78.11	579.2±92.63	2795.4±197.46	6.6±1.166	3621.21±1184.63
10%	62.4± 15.20	257.4±86.37	1946.8±49.02	5.6±.489	3278.83±365.74
labeled %	examples (F)	examples (M)	examples(S)	nIter	time
5	229.4±86.56	1340.0±219.14	2678.2±205.55	5.8±.979	2420.12±414.10
5,10	197.0± 62.35	739.0±183.06	2770.0±181.47	5.8±.399	2581.65±410.09
5,10,15	230.6± 78.11	579.2±92.63	2795.4±197.46	6.6±1.166	3621.21±1184.63
5,10,15,20	.234.8± 105.87	478.2±95.31	2834.2±145.32	5.8 ±1.166	3911.88±1020.75
5,10,15,20,25	226.2±73.83	414.2±102.15	2853.4±196.07	6.4±1.199	6415.632 ±1808.412
labeled %	examples (F)	examples (M)	examples(S)	nIter	time
Aspatial [22]	9669.6±43.45	9654.4±45.37	9694.6±17.16	6.6±.800	25257.99±1906.26
Spatial	8589.0± 65.64	8694.8±54.11	8689.4± 64.96	5.8±.748	18051.2± 297.75
Spatial+DCC	230.6± 78.11	579.2±92.63	2795.4±197.46	6.6±1.166	3621.21±1184.63
labeled %	examples (F)	examples (M)	examples(S)	nIter	time
Self-SVM(**S**)	0.0 ±0.00	0.0 ±0.00	7466.0 ±587.27	9.8±.399	5519.05±1522.25
SVM(**S**)+SVM(**FM**)	0.0 ±0.00	0.0 ±0.00	0.0 ±0.00	.0±.0	236.17±4.67
S+**F**Co-Training	126.8±57.05	0.0±0.00	2806.4±232.81	6.2±.074	3510.24 ±458.83
S+**M**Co-Training	0.0±0.00	592.2±358.18	2723.2±236.96	7.2±1.59	4179.866±1222.77
S+**F**+**M**Co-Training	230.6± 78.11	579.2±92.63	2795.4±197.46	6.6±1.166	3621.21±1184.63

from 5 to 10 and 15, while we run co-training with the initial labeled set sampled with the labeling percentage equal to 5%. We use both spectral, spatial-frequency and spatial-morphology profiles of the imagery data. The computed metrics are reported in Tables 5 (rows 2-4, columns 6-8) and 6 (rows 12-14) . The results show that dealing with the spatial information in the selection schema improves the classification accuracy and speeds-up the learning process, as a lower number of examples is moved from the unlabeled sets to the labeled sets during the iterative learning process. The classification maps reported in Figures 4(a)-4(c) highlight that accounting for the spatial information, in order to select reliable labels, reduces the number of pixels misclassified along the boundaries between various thematic objects (see Figure 4(c) vs Figures 4(a)-4(b)) -This confirms our considerations reported in Section 4.4. We also note that the use of a diversity class criterion scales-up the computation without greatly changing the classification map produced (see Figure 4(a) vs Figure 4(b)).

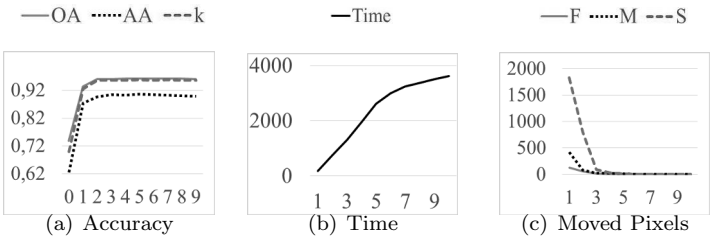
Spectral and/or spatial profiles. We compare the performances of the following learning schemes: iterative learning performed with the spectral profile only (Self-SVM(**S**)); two-stepped learning composed by one SVM learned from the spectral profile followed by one SVM learned from the spatial (frequency and morphology) profile (SVM(**S**)+SVM(**FM**)); iterative learning performed with the co-training strategy, one spectral profile and one spatial profile (frequency (**S**+**F**Co-Training) or morphology (**S**+**M**Co-Training)); and iterative



(a) Spatial selection (b) Spatial + DCC (c) Aspatial selection

Figure 4: Indian Pines: classification maps produced, while running S^{2}CoTraC with 4(a) the spatial selection criterion (described in Section 4.4), 4(b) the spatial selection criterion and the diversity class criterion (described in Section 4.3), and 4(c) the aspatial selection criterion (described in [22]). A circle highlights every wrongly labeled region in Figure 4(c).

learning performed with the co-training strategy, one spectral profile and two spatial profiles (**S**+**F**+**M**Co-Training). We note that **S**+**F**+**M**Co-Training corresponds to the learning schema of the algorithm proposed in this study (baseline). We construct neighborhoods with size growing from 5 to 10 and 15, while we consider the initial labeled set sampled with the labeling percentage equal to 5%. We use the spatial example selection criterion and combine it with the class diversity criterion when the co-training strategy is applied. The computed metrics are reported in Tables 5 (rows 6-10, columns 6-8) and 6 (rows 16-20). The results show that the iterative learning performed along the spectral and spatial profiles is more accurate than the iterative learning performed along the spectral profile only (**S**+**F**+**M**Co-Training, **S**+**F**Co-Training and **S**+**M**Co-Training outperform Self-SVM(**S**)). This confirms the considerations reported in [32, 31, 55, 33, 5, 15, 54, 7, 58, 20, 1], which inspire the emerging trend of considering spatial information, in addition to spectral information in imagery data. We also observe that, interestingly, the iterative learning converges faster when spatial profiles of the data are processed. The number of iterations, as well as the number of transferred examples diminish when the classification is based on both spectral data and collective spatial data. This means that accounting for spatial collective information actually contributes to determining correct classes by avoiding a computation burden. At the same time, by focusing this analysis on the number of spatial profiles, the results (rows 8-10, columns 6-8, Table 5; rows 18-20 Table 6) show that the classification accuracy produced with one spectral profile and "two" spatial profiles (**S**+**F**+**M**Co-Training) is higher than the classification accuracy produced with one spectral profile and one spatial profile (**S**+**F**Co-Training and **S**+**M**Co-Training). This confirms the considerations reported in [22], which inspire us to consider "various" (and possibly independent) spatial profiles of data. On the other hand, by analyzing the contribution of the iterative learning performed in combination with the collective inference, the results (rows 7-10, columns 6-8, Table 5; rows 17-20 Table 6) show that the use of iterative learning really improves the classification accuracy. In fact, SVM(**S**)+SVM(**FM**) is outperformed by **S**+**F**Co-Training, **S**+**M**Co-Training and **S**+ **F**+**M**Co-Training. This confirms the results of previous studies [25, 19] in collective inference, which have assessed the effectiveness of iterative learning to account for the correlation of labels.



(a) Accuracy (b) Time (c) Moved Pixels

Figure 5: Sensitivity study: the accuracy (Y axis, Figure 5(a)), the number of examples moved from the unlabeled set to the labeled set per profile (Y axis, Figure 5(c)) and the learning time (in secs, Y axis, Figure 5(b)) are plotted along the the number of performed iterations (X axis). S^2CoTraC is run by considering the labeled sets generated by sampling 5% of ground-truth pixels and by constructing the spatial features over the spatial neighborhoods with size growing from 5 to 10 and 15.

Number of Iterations. This analysis is performed by applying the co-training strategy, constructing neighborhoods with size growing from 5 to 10 and 15, while considering the initial labeled set sampled with the labeling percentage equal to 5%. We use the spatial example selection criterion and combine it with the class diversity criterion. We analyze changes in the performance of the learning process along the number of iterations. The accuracy metrics, the computation time (in secs) and the number of transferred examples per profile are the plots in Figures 5(a)- 5(c). These plots show that accuracy is gained as new iterations are performed. This is a further confirmation of the effectiveness of the iterative learning approach. In any case, we can also observe that the highest increase of accuracy is obtained in the initial iterations of the learning process, which are also those showing the highest number of transferred examples per profile. Consequently, stopping the iterative learning when the number of transferred examples is less than 10 ($MinTransfer$) saves a computation burden that would not change the classification accuracy greatly.

5.4. Semi-supervised and Transductive Literature

Several general-purpose semi-supervised and transductive classifiers have been proposed in the machine learning literature. We consider the Fast Linear transductive SVM (SVMLin) [50],[15] the Spectral Graph Transducer (SGT) [27],[16] and the self-labeled algorithms described in [56].[17] The self-labeled algorithms differ in various characteristics. In particular, the mechanism to enlarge the labeled set can be: incremental (the most reliable examples are added step-by-step from the unlabeled set to the labeled set), batch (the algorithm decides whether each unlabeled example meets the addition criterion before adding any of them to the labeled set; it does not assign a definitive label to each unlabeled example and can reprioritize the hypothesis learned from the labeled set) and amending (any example that meets the specific criterion is added or removed from the labeled set). The classifier can be: single (one classifier is used to enlarge

[15]The implementation of SVMLin is available at `http://vikas.sindhwani.org/svmlin.html`.
[16]The of SGT is available at `http://sgt.joachims.org/`.
[17]The implementation of considered self-labeled algorithms is available at `http://sci2s.ugr.es/SelfLabeled`.

the labeled set) or multiple (multiple classifiers are considered). Apart from the number of classifiers, the learning can be: single (one learning algorithm is considered) or multiple (multiple learning algorithms are used). Finally, by referring to the way the input feature space is taken into consideration, the learning can be: single view (all features are considered at once) and multi-view (the feature set is split in two or more redundant, conditionally-independent subsets). The self-labeled algorithms, considered for this study, include: SelfTraining (single-view, single learning, single classifier, incremental); CoTraining, Rasco and RelRasco (multi-view, single learning, multiple classifiers); DemocraticCo (single view, multiple learning, multiple classifiers, incremental); CoBagging and TriTraining (single view, single learning, multiple classifiers, incremental); ADECoForest, DETriTraining and CLCC (single view, single learning, multiple classifiers, augmenting); CoForest (single view, single learning, multiple classifiers, batch); APSSC (single-view, single learning, single classifier, batch); as well as SETRED and SNNRCE (single-view, single learning, single classifier, augmenting). SelfTraining, CoTraining, Co-Bagging, TriTraining, DETriTraining, Rasco and RelRasco are designed with SVM, k-NN, C4.5 or Naive Bayes as base classifiers. These self-labeled algorithms are run with the configuration parameters reported in [56].

For all these competitors, we analyze the accuracy (OA, AA and κ) of classification performed on the five partitioning trials between labeled and unlabeled sets generated for each data set. All algorithms have been evaluated by using the same division between labeled samples and unlabeled samples for each running trial. We evaluate the accuracy of all competitors described above in the classification of Indian Pines data set, while we consider the outstanding competitors in the classification of Pavia University and Salinas Valley too. The accuracy of these competitors is compared to that of S^2CoTraC-DCC.[18]

For collected accuracy metrics reported in Table 7, we make the following considerations. The accuracy-based ranking of all algorithms, that is performed by considering Indian Pines data set, reveals that the outstanding competitors are TriTraining(SVM), Co-Training(SVM) and Co-Bagging(SVM). This ranking, independently of the base learner, is consistent with the comparative analysis illustrated in [56]. In fact, this previous analysis, performed for the self-labeled competitors with 55 standard classification data sets, showed TriTraining(C45), Democratic-Co, Co-Bagging(C45) and Co-Training(SVM) as outstanding transductive algorithms. These results confirm that the highest accuracy can be achieved when SVMs are applied as base learners. This supports the considerations formulated by Plaza et al, [43], as well as Fauvel et al. [16], which pointed out that inductive SVMs can be applied to hyperspectral image classification with great success, by outperforming several other inductive classifiers. Finally, the comparative study highlights that no general-purpose competitor outperforms the application-specific algorithm S^2CoTraC-DCC. Based on these results, we can conclude that the transductive process performed without accounting for the spatial information is ineffective to deal with all the challenges of the hyperspectral scenario and motivates the

[18] S^2CoTraC can be characterized as an incremental, multi-classifier, single learning and multi-view self-labeled algorithm.

Table 7: Accuracy metrics (± standard deviation) of state-of-the-art semi-supervised and transductive classifiers defined in machine learning literature. Metrics are collected on five trials produced by considering 5% of ground-truth data as the labeled set. The top-three competitors which achieve the highest accuracy in Indian Pines have been underlined.

system	OA	AA	κ	system	OA	AA	κ
Indian Pines							
S^{2}CoTraC-DCC	**.961**± .009	**.898**±.033	**.956**± .010	SVMLin	.605±.009	.517±.010	.553±.011
SGT	.604±.012	.537±.024	.541±.018	ADECoForest	.638 ± .007	.496 ± .010	.582 ± .007
APSSC	.499 ± .013	.576 ± .014	.437 ± .013	CLCC	.522 ± .011	.325 ± .021	.430 ± .016
$\underline{CoBagging(SVM)}$	.697±.018	.610±.008	.657±.020	CoBagging(NN)	.594±.007	.510±.026	.537±.007
CoBagging(C45)	.475±.011	.381±.005	.405±.012	CoBagging(NB)	.446±.032	.366±.019	.379±.028
CoForest	.674 ± .003	.546 ± .016	.624 ± .003	$\underline{CoTraining(SVM)}$	.708±.016	.625±.008	.670±.018
CoTraining(NN)	.640±.010	.590±.013	.589±.011	CoTraining(NB)	.507±.095	.446±.092	.447±.103
CoTraining(C45)	.544±.016	.445±.032	.480±.017	DemocraticCo	.544 ± .016	.446 ± .033	.480 ± .018
DETriTraining(SVM)	.638 ± .015	.509 ± .006	.586 ± .015	DETriTraining(NN)	.621 ± .012	.496 ± .017	.563 ± .014
DETriTraining(C45)	.569 ± .009	.421 ± .013	.501 ± .010	DETriTraining(NB)	.485 ± .023	.388 ± .013	.417 ± .024
Rasco(SVM)	.351±.027	.268±.018	.245±.025	Rasco(NN)	.187 ± .008	.121 ± .005	.073 ± .008
Rasco(C45)	.205 ± .009	.129 ± .006	.093 ± .011	Rasco(NB)	.215 ± .114	.203 ± .073	.153 ± .081
RelRasco(SVM)	.367 ± .040	.268 ± .033	.263 ± .040	RelRasco(NN)	.192 ± .008	.129 ± .009	.079 ± .008
RelRasco(C45)	.202 ± .010	.127 ± .005	.090 ± .011	RelRasco(NB)	.297 ± .131	.257 ± .064	.217 ± .096
SelfTraining(SVM)	.662 ± .017	.639 ± .027	.618 ± .018	SelfTraining(NN)	.659 ± .013	.622 ± .036	.610 ± .016
SelfTraining(C45)	.535 ± .017	.433 ± .031	.470 ± .020	SelfTraining(NB)	.443 ± .030	.375 ± .032	.373 ± .030
SETRED	.656 ± .006	.581 ± .022	.602 ± .007	SNNRCE	.658 ± .011	.597 ± .014	.608 ± .012
$\underline{TriTraining(SVM)}$	.709± .019	.628± .010	.671± .021	TriTraining(NN)	.640 ± .010	.589 ± .015	.588 ± .011
TriTraining(C45)	.572 ± .010	.456 ± .028	.510 ± .011	TriTraining(NB)	.458 ± .040	.395 ± .018	.392 ± .036
Pavia University							
S^{2}CoTraC-DCC	.985 ±.009	.958± .008	.980±.012	CoBagging(SVM)	.906±.004	.882±.008	.876±.006
Co-Training(SVM)	.908±.005	.882±.009	.878±.007	TriTraining(SVM)	.908±.005	.884±.008	.878±.007
Salinas University							
S^{2}CoTraC-DCC	.993± .004	.996±.001	.993± .0048	CoBagging(SVM)	.916±.003	.950±.002	.907±.003
CoTraining(SVM)	.914±.003	.950±.003	.905±.005	TriTraining(SVM)	.916±.005	.949±.003	.906±.006

design of application-specific transductive algorithms in this scenario.

5.5. Hyperspectral Image Processing Literature

Several (transductive) spatio-spectral algorithms have been evaluated by considering Indian Pines, Pavia University and/or Salinas Valley data sets. In this study, we consider the most recent (and competitive) results [33, 31, 32, 55, 54, 20, 1] produced in in hyperspectral imaging analysis for these data scenarios. The application-specific classifiers evaluated include: lorsalmll, that resorts to a multilevel logistic prior that encodes the spatial information and uses active learning [31]; mpmlbp, that considers spectral and spatial information, by using loopy belief propagation and active learning [33]; mlrsubmll, that integrates spectral and spatial information in a multinomial logistic regression (MLR) algorithm and uses a multilevel logistic Markov-Gibbs with a Markov random field prior to synthesize the spatial information [32]; svmmrf, that firstly applies a probabilistic support vector machine spectral-based classification of the hyperspectral image and then refines the classification obtained by using spatial contextual information through a Markov

random field regularization [55]; a spatial-aware SVM that learns SVMs after extending the spectral feature space with a spatial-aware morphological profile [43, 33];[19] Watershed, that uses watershed segmentation, in order to define information on spatial structures and perform spectral-based SVM classification, followed by majority voting within the watershed regions [54, 33]; irmc, that implements two MLR classifiers, which are fed with spectral features and spatial features, respectively, and work iteratively, so that every classifier exploits the decision of the other [20]; S^2Tec that learns SVM classifiers and integrates the spectral information and the label spatial correlation through an ensemble system.

For Indian Pines, results have been produced in the literature with 5% (irmc), 6% (svmmrf) and 10% (lorsalmll, mpmlbp, mlrsubmll and SVMMRC) of the pixels labeled according to the available ground truths. For Pavia University, the results have been produced in the literature with 5% (irmc) and 9% (spatial SVM, lorsalmll, mpmlbp, mlrsubmll, SVMMRC and Watershed) of the pixels labeled according to the available ground truths. For Salinas Valley, the results have been produced in the literature with 5% (irmc) of the pixels labeled according to the available ground truths. In all these data sets, the remaining pixels have been unlabeled according to the proper transductive setting. It is noteworthy that the accuracy performance reported in [33, 31, 32, 55] is achieved by starting from labeled samples which are different from those considered for this study. Thus the comparison in these cases is not properly safe. However, the low standard deviation of the accuracy metrics computed for S^2CoTraC on several trials supports the theory that it would perform equally well if it were run with the labeled sets used in [33, 31, 32, 55].

Accuracy results are reported in Table 8. The accuracy metrics of all the competitors in this study are collected with the percentage of the pixels labeled for the learning phase greater than or equal to 5%. In any case, the accuracy of the competitors is, almost always, outperformed by the accuracy of S^2CoTraC-DCC run with only 5% of the pixels labeled. The only exceptions are observed when loraslmll, mpmlbp, mlrsubmll and svmmrf are used to classify Indian Pines data. In any case, the best performance of these competitors is achieved only in association to metric AA and differences in AA performance are smaller when S^2CoTraC-DCC is run with 10% of the pixels labeled. In fact, both OA and κ achieved by lorsalmll, mpmlbp and mlrsubmll (with 10% of the pixels labeled), as well as by svmmrf (with the 6% of the pixels labeled) are lower than OA and κ achieved by S^2CoTraC-DCC (with both 5% and 10% of the pixels labeled). In general, if we look at the class-by-class precision results (see Figure 6), we observe that the gain in the average accuracy is due to the ability of these competitors to achieve higher accuracy than S^2CoTraC-DCC, when classifying the pixels belonging to the minority classes "Grass/pasture-mintill" (0.27% of imagery pixels) and "Oat" (0.20% of imagery pixels). By considering that lorsalmll and mpmlbp operate with the active learning strategy, while mlrsubmll and svmmrf apply the Markov random field theory, this analysis suggests that extensions of the proposed algorithm may include either active learning or Markov random fields, in

[19] This profile includes opening and closing features, as well as spatial information like size, orientation and local contrast.

Table 8: Accuracy metrics (± standard deviation) of state-of-the-art, spatio-spectral, hyperspectral classifiers.

algorithm	ref	label%	OA	AA	κ
Indian Pines					
S^{2}CoTraC-DCC	Table 2	5%	.961±.009	.898±.033	.956±.010
S^{2}CoTraC-DCC	Table 5	10%	.980±.007	.927±.030	.977±.009
lorsalmll	[31, 33]	10%	.927	.951	.916
mpmlbp	[33]	10%	.947	.962	.939
mlrsubmll	[32]	10%	.936	.939	.926
svmmrf	[55]	6%	.920	.958	.909
irmc	Table 2,[20]	5%	.873±.031	.821±.075	.856±.035
S^{2}Tec	Table 2,[1]	5%	.936±.011	.874±.031	.927±.012
Pavia University					
S^{2}CoTraC-DCC	Table 2	5%	.985±.009	.958±.008	.980±.012
spatial SVM	[43] [33]	9%	.852	.907	.808
lorsalmll	[31, 33]	9%	.855	.925	.818
mpmlbp	[33]	9%	.857	.922	.820
mlrsubmll	[32]	9%	.941	.935	.922
svmmrf	[55]	9%	.976	.945	.959
Watershed	[54] [33]	9%	.854	.913	.813
irmc	Table 2,[20]	5%	.867±.008	.837±.007	.814±.010
S^{2}Tec	Table 2,[1]	5%	.988±.009	.982±.003	.984±.001
Salinas Valley					
S^{2}CoTraC-DCC	Table 2	5%	.993±.004	.996±.001	.994±.005
irmc	Table 2,[20]	5%	.955±.018	.952±.016	.950±.020
S^{2}Tec	Table 2,[1]	5%	.970±.006	.984±.003	.967±.006

Figure 6: Indian Pines dataset (10% of labeled data): class-by-class accuracy achieved by S^{2}CoTraC-DCC, lorsalmll and mpmlbp (as reported in [33]), mlrsubmll (as reported in [32]) and svmmrf (as reported in [55]).

order to improve the accuracy in the classification of minority classes. In any case, these competitors are outperformed by (or perform similarly to) S^{2}CoTraC-DCC when considering all the remaining classes.

6. Conclusion

In this paper, we describe a novel, hybrid, transductive hyperspectral image classification algorithm to cope with a limited number of labeled pixels in the high dimensional spectral space. The algorithm iteratively constructs various spatial features over spatial neighborhoods via a collective algorithm. It uses a co-training system of spectral and spatial classifiers to determine the pixel labels and applies transductive

learning to make accurate predictions. Spatial features model the continuity of neighboring labels. They exploit the likely fact that two neighboring pixels may have the same label (label spatial correlation).

Collective inference, transductive learning and co-training have already been explored in the literature. The novel contribution of this study is that it combines these three strategies in a single learning algorithm. This algorithm, which represents one of the main contributions of this work, proves effective for the challenging problem of hyperspectral classification. The effectiveness of the proposed algorithm is assessed via an empirical study on several hyperspectral data sets. This study contributes to proving that the proposed formulation of a collective-based co-training classifier is more accurate than the collective-based turbo code, as well as the ensemble described in our previous works [20, 1]. On the other hand, the presented algorithm is more accurate than co-training classifiers [22, 61], which are already defined for hyperspectral classification, but ignore collective inference to deal with spatial information. Finally, the described algorithm gains in accuracy compared to various state-of-the-art classifiers defined in both the semi-supervised/transductive learning literature and the hyperspectral image analysis literature.

Another novel contribution of this study is the consideration of an example selection schema that accounts for the spatial correlation of imagery labels and spectral signatures, in order to select new training examples for the iterative learning process. This advances our previous hybrid algorithms described in [20, 1], which used the spatial correlation of the imagery labels during collective inference, but neglected the spatial information during the example selection phase. The empirical investigation has proved that this spatial selection schema contributes to improving the accuracy, to speeding-up the learning process and to reducing the number of pixels misclassified along the boundaries between various thematic objects. A final contribution of this study is the investigation of a diversity class criterion that, used in combination with co-training, can speed-up the learning process.

We note that this study is relevant for the machine learning community, as it contributes to proving that by combining transductive learning and collective classification it is possible to improve significantly the accuracy of the classifier in comparison to a supervised/non collective setting. These improvements are shown to be relevant in an applicative context (remote sensing) that has recently gained importance. This study is also significant for the hyperspectral image classification community, as it describes an algorithm that deals with spectral and spatial information by gaining accuracy with respect to state-of-the-art algorithms.

Some directions for further work are still to be explored. The selection of the initial labeled set is still an open problem, which is unexplored in this study. The active learning can be explored, in order to initially select and intelligently augment the labeled set during the iterative process. This can be done by trying to improve the classification accuracy of sparsely populated classes. On the other hand, the Markov random field theory can be investigated in the synthesis of the spatial information. Additionally, it would be interesting to study ways to adaptively determine both the size and the shape of neighborhoods. Finally, big data technologies may be investigated, in order to apply the presented solution to big imagery data.

Acknowledgments

The authors would like to acknowledge the support of the European Commission through the project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944). The authors wish to thank Qiang Song for providing the code of mcogpc, as well as Lynn Rudd for her help in reading the manuscript.

References

[1] A. Appice, P. Guccione, D. Malerba, Transductive hyperspectral image classification: toward integrating spectral and relational features via an iterative ensemble system, Machine Learning Journal 103(3) (2016) 343–3.

[2] J. Benediktsson, M. Pesaresi, K. Amason, Classification and feature extraction for remote sensing images from urban areas based on morphological transformations, IEEE Transactions on Geoscience and Remote Sensing 41 (9) (2003) 1940–1949.

[3] A. Bernardini, E. Frontoni, E. Malinverni, A. Mancini, A. Tassetti, P. Zingaretti, Pixel, object and hybrid classification comparisons, Journal of Spatial Science 55 (1) (2010) 43–54.

[4] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proc. of the 11th Workshop on Computational Learning Theory, COLT 1998, 1998.

[5] F. Bovolo, L. Bruzzone, M. Marconcini, A novel context-sensitive SVM for classification of remote sensing images, in: Proc. of the 2006 IEEE International Conference on Geoscience and Remote Sensing Symposium, IGARSS 2006, 2006, pp. 2498–2501.

[6] L. Bruzzone, M. Chi, M. Marconcini, A novel transductive SVM for semisupervised classification of remote-sensing images, IEEE Transactions on Geoscience and Remote Sensing 44 (11) (2006) 3363–3373.

[7] G. Camps-Valls, T. Bandos Marsheva, D. Zhou, Semi-supervised graph-based hyperspectral image classification, IEEE Transactions on Geoscience and Remote Sensing 45 (10) (2007) 3044–3054.

[8] X. Ceamanos, B. Waske, J. Benediktsson, J. Chanussot, J. Sveinsson, Ensemble strategies for classifying hyperspectral remote sensing data, in: J. Benediktsson, J. Kittler, F. Roli (eds.), Multiple Classifier Systems, vol. 5519 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 62–71.

[9] M. Ceci, A. Appice, H. L. Viktor, D. Malerba, E. Paquet, H. Guo, Transductive relational classification in the co-training paradigm, in: P. Perner (ed.), Proc. of the 8th International Conference Machine Learning and Data Mining in Pattern Recognition, MLDM 2012, vol. 7376 of Lecture Notes in Computer Science, Springer, 2012, pp. 11–25.

[10] C.-I. Chang, Hyperspectral Data Exploitation: Theory and Applications, Wiley, 2007.

[11] W. Cheetham, J. Price, Measures of solution accuracy in case-based reasoning systems, in: P. Funk, P. A. González-Calero (eds.), Proc. of the 7th European Conference on Advances in Case-Based Reasoning, ECCBR 2004, vol. 3155 of Lecture Notes in Computer Science, Springer, 2004, pp. 106–118.

[12] J. Chen, J. Xia, P. Du, J. Chanussot, Combining rotation forest and multiscale segmentation for the classification of hyperspectral data, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing PP (99) (2016) 1–14.

[13] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[14] B. Demir, C. Persello, L. Bruzzone, Batch-mode active-learning methods for the interactive classification of remote sensing images, IEEE Transactions on Geoscience and Remote Sensing 49 (3) (2011) 1014–1031.

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

[16] M. Fauvel, Y. Tarabalka, J. Benediktsson, J. Chanussot, J. Tilton, Advances in spectral-spatial classification of hyperspectral images, Proc. of the IEEE 101 (3) (2013) 652–675.

[17] A. Fujino, N. Ueda, K. Saito, Semisupervised learning for a hybrid generative/discriminative classifier based on the maximum entropy principle, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (3) (2008) 424–437.

[18] P. Gamba, ROSIS Pavia University 2003 data set, 2003.

[19] L. Getoor, B. Taskar, Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning), The MIT Press, 2007.

[20] P. Guccione, L. Mascolo, A. Appice, Iterative hyperspectral image classification using spectral-spatial relational features, IEEE Transactions on Geoscience and Remote Sensing 53 (7) (2015) 3615–3627.

[21] R. Hang, Q. Liu, H. Song, Y. Sun, Matrix-based discriminant subspace ensemble for hyperspectral image spatial-spectral feature fusion, IEEE Transactions on Geoscience and Remote Sensing 54 (2) (2016) 783–794.

[22] R. Huang, W. He, Using tri-training to exploit spectral and spatial information for hyperspectral data classification, in: Proc. of the 2012 International Conference on Computer Vision in Remote Sensing, CVRS 2012, 2012, pp. 30–33.

[23] G. Hughes, On the mean accuracy of statistical pattern recognizers, IEEE Transactions on Information Theory 14 (1) (1968) 55–63.

[24] E. H. Isaaks, M. R. Srivastava, An Introduction to Applied Geostatistics, Oxford University Press, USA, 1990.

[25] D. Jensen, J. Neville, B. Gallagher, Why collective inference improves relational classification, in: Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2004, ACM, 2004, pp. 593–598.

[26] T. Joachims, Transductive inference for text classification using support vector machines, in: I. Bratko, S. Dzeroski (eds.), Proc. of the 16th International Conference on Machine Learning, (ICML 1999), Morgan Kaufmann, 1999, pp. 200–209.

[27] T. Joachims, Transductive learning via spectral graph partitioning, in: T. Fawcett, N. Mishra (eds.), Proc. of the 20th International Conference on Machine Learning, ICML 2003, AAAI Press, 2003, pp. 290–297.

[28] L. Johnson, AVIRIS Hyperpsectral Radiance Data from: f981009t01r07, 1998.

[29] D. Landgrebe, AVIRIS NW Indiana's Indian Pines 1992 data set, 1992.

[30] P. Legendre, Spatial autocorrelation: Trouble or new paradigm?, Ecology 74 (6) (1993) 1659–1673.

[31] J. Li, J. Bioucas-Dias, A. Plaza, Hyperspectral image segmentation using a new bayesian approach with active learning, IEEE Transactions on Geoscience and Remote Sensing 49 (10) (2011) 3947–3960.

[32] J. Li, J. Bioucas-Dias, A. Plaza, Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields, IEEE Transactions on Geoscience and Remote Sensing 50 (3) (2012) 809–823.

[33] J. Li, J. Bioucas-Dias, A. Plaza, Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning, IEEE Transactions on Geoscience and Remote Sensing 51 (2) (2013) 844–856.

[34] Q. Lv, X. Niu, Y. Dou, J. Xu, Y. Lei, Classification of hyperspectral remote sensing image using hierarchical local-receptive-field-based extreme learning machine, IEEE Geoscience and Remote Sensing Letters 13 (3) (2016) 434–438.

[35] D. Malerba, M. Ceci, A. Appice, A relational approach to probabilistic classification in a transductive setting, Engineering Applications of Artificial Intelligence 22 (1) (2009) 109–116.

[36] U. Maulik, D. Chakraborty, Learning with transductive SVM for semisupervised pixel classification of remote sensing imagery, ISPRS Journal of Photogrammetry and Remote Sensing 77 (0) (2013) 66 – 78.

[37] L. McDowell, D. W. Aha, Semi-supervised collective classification via hybrid label regularization, in: Proc. of the 29th International Conference on Machine Learning, ICML 2012, Omnipress, 2012.

[38] F. Melgani, L. Bruzzone, Classification of hyperspectral remote sensing images with support vector machines, IEEE Transactions on Geoscience and Remote Sensing 42 (8) (2004) 1778–1790.

[39] L. Miao, Z. Shuying, B. Zhang, L. Shanshan, W. Changshan, A review of remote sensing image classification techniques: the role of spatio-contextual information, European Journal of Remote Sensing 47 (2014) 389–411.

[40] E. Pasolli, F. Melgani, D. Tuia, F. Pacifici, W. J. Emery, SVM active learning approach for image classification using spatial information, IEEE Transactions on Geoscience and Remote Sensing 52 (4) (2014) 2217–2233.

[41] E. Pasolli, H. L. Yang, M. M. Crawford, Active-metric learning for classification of remotely sensed hyperspectral images, IEEE Transactions on Geoscience and Remote Sensing 54 (4) (2016) 1925–1939.

[42] J. C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, MIT Press, 1999, pp. 61–74.

[43] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, G. Trianni, Recent advances in techniques for hyperspectral image processing, Remote Sensing of Environment 113(1) (2009) 110 – 122.

[44] F. Ratle, G. Camps-Valls, J. Weston, Semisupervised neural networks for efficient hyperspectral image classification, IEEE Transactions on Geoscience and Remote Sensing 48 (5) (2010) 2271–2282.

[45] J. A. Richards, Remote Sensing Digital Image Analysis: An Introduction, 2nd ed., Springer-Verlag New York, Inc., 1993.

[46] M. Seeger, Learning with labeled and unlabeled data, Tech. rep. (2001).

[47] A. K. Shackelford, C. H. Davis, A combined fuzzy pixel-based and object-based approach for classification of high-resolution multispectral data over urban areas, IEEE Transactions on Geoscience and Remote Sensing 41 (10) (2003) 2354–2363.

[48] B. Shahshahani, D. Landgrebe, The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon, IEEE Transactions on Geoscience and Remote Sensing 32 (5) (1994) 1087–1095.

[49] X. Shi, Y. Li, P. Yu, Collective prediction with latent graphs, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM 2011, ACM, 2011, pp. 1127–1136.

[50] V. Sindhwani, S. S. Keerthi, Large scale semi-supervised linear SVMs, in: E. N. Efthimiadis, S. T. Dumais, D. Hawking, K. Järvelin (eds.), Proc. of the 29th Annual International Conference on Research and Development in Information Retrieval, SIGIR 2006, ACM, 2006, pp. 477–484.

[51] S. D. Stearns, B. E. Wilson, J. R. Peterson, Dimensionality reduction by optimal band selection for pixel classification of hyperspectral imagery, in: Proc. of SPIE Applications of Digital Image Processing, vol. 2028, 1993, pp. 118–127.

[52] K. Tan, E. Li, Q. Du, P. Du, An efficient semi-supervised classification approach for hyperspectral imagery, ISPRS Journal of Photogrammetry and Remote Sensing 97 (0) (2014) 36 – 45.

[53] K. Tan, E. Li, Q. Du, P. Du, Hyperspectral image classification using band selection and morphological profiles, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 7 (1) (2014) 40–48.

[54] Y. Tarabalka, J. Chanussot, J. Benediktsson, Segmentation and classification of hyperspectral images using watershed transformation, Pattern Recognition 43 (7) (2010) 2367 – 2379.

[55] Y. Tarabalka, M. Fauvel, J. Chanussot, J. Benediktsson, SVM- and MRF-based method for accurate classification of hyperspectral images, IEEE Geoscience and Remote Sensing Letters 7 (4) (2010) 736–740.

[56] I. Triguero, S. García, F. Herrera, Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study, Knowledge and Information Systems 42 (2) (2015) 245–284.

[57] V. N. Vapnik, The Nature of Statistical Learning Theory, New York, NY, USA, 1995.

[58] L. Wang, S. Hao, Q. Wang, Y. Wang, Semi-supervised classification for hyperspectral imagery based on spatial-spectral label propagation, ISPRS Journal of Photogrammetry and Remote Sensing 97 (0) (2014) 123 – 137.

[59] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed., Morgan Kaufmann, 2005.

[60] R. Xiang, J. Neville, Pseudolikelihood em for within-network relational learning, in: Proc. of the 8th IEEE International Conference on Data Mining, ICDM 2008, IEEE, 2008, pp. 1103–1108.

[61] X. Zhang, Q. Song, R. Liu, W. Wang, L. Jiao, Modified co-training with spectral and spatial views for semisupervised hyperspectral image classification, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 7 (6) (2014) 2044–2055.

*Author Biography

Annalisa Appice is an Assistant Professor in the Department of Informatics, University of Bari Aldo Moro. Her research activity mainly concerns data mining and machine learning. She has published more than 120 papers in international journals and conference proceedings. She is involved in many European and national projects on data mining. She has served in the program committee of more than 30 international conferences and workshops.

Pietro Guccione is an Assistant Professor with the Department of Electric and Information Engineering, Polytechnic University of Bari. %, where he teaches signal theory and digital communication systems. He has authored nearly 80 papers in the field of synthetic aperture radar, signal processing, and multivariate statistical analysis. He has contributed and coordinated several research projects in the field of remote sensing, working with ESA, the Italian Space Agency (ASI), and aerospace companies.

Donato Malerba is a Full Professor at the Department of Computer Science of the University of Bari Aldo Moro. His research activity mainly concerns data mining, machine learning and big data. He has published more than 200 papers in international journals and conference proceedings. He received the IBM Faculty Award for the year 2004. He is responsible for a research unit of several European and national projects.