Contributions of Machine Learning to Remote Sensing Data Analysis

Paul Scheunders[a], Devis Tuia[b], Gabriele Moser[c]

[a]*Vision Lab, Department of Physics, University of Antwerp, 2610 Wilrijk, Belgium (e-mail: paul.scheunders@uantwerpen.be).*
[b]*MultiModal Remote Sensing, Department of Geography, University of Zurich, 8057 Zurich, Switzerland (e-mail: devis.tuia@geo.uzh.ch).*
[c]*DITEN Department, University of Genoa, via Opera Pia 11a, 16145 Genoa, Italy, (e-mail: gabriele.moser@unige.it)*

Abstract

This chapter describes the state of the art on the development and application of machine learning methodologies in the remote sensing domain. In an introductory section, we describe the specific remote sensing analysis problems that are typically handled by machine learning. The remaining of the chapter is subdivided in a number of sections, dealing with groups of machine learning strategies. The following strategies are elaborated on: kernel methods, neural network methods, manifold learning methods, structured output methods, ensemble learning methods and sparse learning methods. This specific choice is based on the frequency with which these strategies appeared in the recent remote sensing literature. Each subsection contains a short description of the specific machine learning paradigm and an extensive description of the recent state of the art, with a conceptual description of the new methodologies. We end with some insights in future developments.

Keywords: Ensemble Learning, Hyperspectral Image Classification, Kernel methods, Machine Learning, Manifold Learning, Neural Networks, Sparse Learning, Structured Output.

1. Introduction

1.1. Machine learning

This chapter describes how machine learning methods are used for the analysis of remote sensing data. Machine learning is a subfield of computer science in which algorithms are developed to learn from and make predictions on data. Rather than following strict program instructions, machine learning algorithms build models from example inputs to make data-driven decisions (Fukunaga, 1990; Vapnik, 1998; Bishop, 2006).

The machine learning paradigm covers a huge research area with very diverse methodologies. It is impossible to make one unique taxonomy that covers all relevant methodologies. Most of the methods described in this chapter are covered by the following rough taxonomy. The most widespread task is supervised learning, in which an algorithm learns a model that maps inputs to outputs, from a selected number of inputs and their desired outputs. Specifically, a number of training samples (a data sample and its label, i.e. desired output) are provided. When the outputs are discrete variables, the labels indicate predefined classes, and the methods are referred to as classification methods. When the outputs are continuous variables, the methods are referred to as regression methods.
The second task is unsupervised learning. Here, the algorithm tries to find structure in the data without supervision. Clustering is the task of grouping the inputs and can be regarded as unsupervised classification.
Besides the two main tasks, a variety of hybrid tasks exist: algorithms that make jointly use of labeled and unlabeled training samples, known as semisupervised learning algorithms (Bruzzone et al., 2006), or active learning (Crawford et al., 2013), where a supervised classification algorithm is able to query the user for the labels of certain samples, preferably those with low classification accuracy, to be included in the training set for reclassification.

1.2. Machine learning for remote sensing data analysis

Machine learning is an important and frequently applied tool for the interpretation and analysis of remote sensing data. A search on the SCI-Expanded database of the ISI Web-Of-Science learns that over the period 2004-2015 about 60.000 papers were published in the domain of remote sensing, of which 10.000 dealing with classification and 3.000 with regression. More than half of these papers deal with the particular problem of hyperspectral image

classification. In Fig. 1, the paper counts per year are depicted. The picture shows a linear increase of papers dealing with classification, while the increase of hyperspectral-related papers is even higher in the last 5 years.

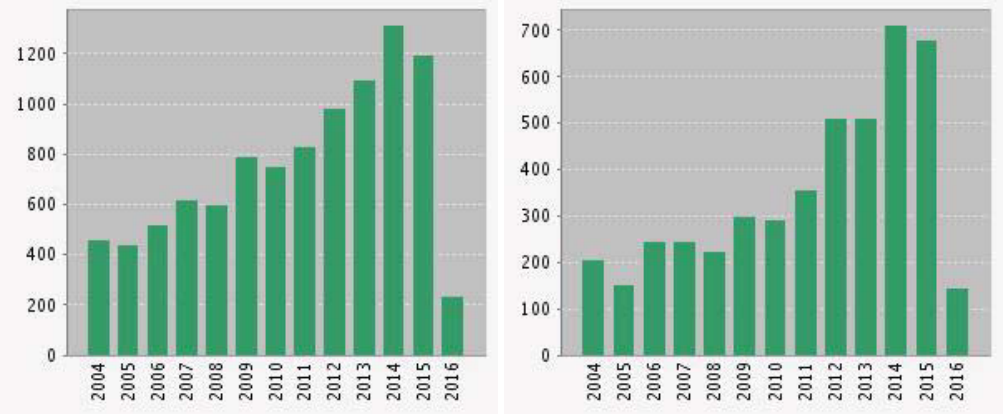


Figure 1: Paper counts per year by searching the SCI-Expanded database of the ISI Web-Of-Science with the following topics: Left - (remote sensing) and (classification); Right - (hyperspectral) and (classification). Search done in April 2016.

There are a number of typical remote sensing analysis tasks for which machine learning can be of great use (Bioucas-Dias et al., 2013). In the following, we give a short description of a (non exhaustive) number of frequently occuring tasks in the recent literature. Dedicated chapters in this book series are devoted to several of these analysis tasks. In this chapter, these tasks will pop up as examples for which specific machine learning methods have been developed.

- By far the most prominent task is *land cover mapping*. Here, each pixel is to be assigned to one of a number of predefined land cover classes (e.g. water, sand, vegetation, ...). Mostly, this is done using the supervised approach, where labeled training samples are obtained e.g. by ground campaigns. Land cover maps can be obtained from any type of remote sensing data (Camps-Valls et al., 2014).

- Satellite sensor platforms fly over the same regions with varying revisiting periods, which allows to study changes of the earth surface. In particular, multitemporal *change detection* is an important research domain (Bruzzone and Bovolo, 2013). Applications can vary from seasonal vegetation mapping to disaster monitoring of floods, tsunami's etc. The problem of change detection can be treated in a supervised or unsupervised manner. One particular problem that has to be tackled is that light reflectance of materials heavily depends on acquisition conditions and external factors, which may vary over time. To tackle such specific problems, researchers have been considering strategies based either on physical models or on the machine learning field of *domain adaptation* (Tuia et al., 2016).

- Another important task is the *estimation of land physical parameters* from the data. Here, a forward model describes a measurement in function of a number of parameters (e.g. moisture) and measurement conditions (e.g. viewing direction). Model inversion then tries to estimate the physical parameters from the measurements. Besides a long tradition of inversion methods based on look up tables using spectra generated by physical models, supervised methods are nowadays being used to estimate these parameters using a training set of input (spectra)-output (parameter values) data samples. In this case, the output labels are continuous variables, which requires regression methods (Verrelst et al., 2012).

- A particular property of many remote sensing data is the high dimensionality. A pixel may contain many measurements as in hyperspectral images, or may be represented by a large number of features, e.g. contextual features from the pixel's neighborhood. Special attention is given to the particular problem of *feature selection and extraction*, or *dimensionality reduction* (Benediktsson et al., 2003). Natural images are in general spatially smooth, remote sensing images make no exception. The research area of *spectral-spatial classification* makes advantage of this property to enhance the performance of classification algorithms (Fauvel et al., 2013).

- Some sensors generate very high spatial resolution image data, in the optical or in the microwave range (SAR). The high resolution allows to perform specific analysis tasks, such as *object and scene detection* (Inglada, 2007), with applications in e.g. urban monitoring (Gamba et al., 2007).

- The ever increasing number of remote sensing products allows to combine information from different sensors to obtain more information than can be obtained from individual sensors. One example is the combination of information from a high spatial resolution sensor with a high spectral resolution sensor. Such combinations can improve classification performances (Fauvel et al., 2006; Gomez-Chova et al., 2015). The information can be fused at the image level (*image fusion*) or at the level of the classifier (*decision fusion*). Another emerging task due to the ever increasing amount of remote sensing data is the *retrieval of images* from large databases (Datcu et al., 1998; Schroder et al., 1998).

- One particular analysis problem is the detection of targets smaller than the pixel size. In *target detection*, one searches for particular materials with a size smaller than the pixel size (Nasrabadi, 2014). If the spectral signature of the target is not known, one refers to anomaly detection (Manolakis et al., 2014). A very active field of research is *spectral unmixing*, where one wants to discover the fractional abundance of materials within a pixel (Bioucas-Dias et al., 2012). These materials can be mixed because of the limited spatial resolution of a pixel, or can be intimately mixed at microscales. Finally *subpixel mapping* aims at obtaining land cover classification maps at the subpixel level, e.g. by combining unmixing with a spatial arrangement of the obtained abundances within a pixel (Atkinson et al., 2008). *Superresolution* methods on the other hand try to improve the spatial resolution of the images before analysis (Gu et al., 2008).

1.3. The chapter structure

The content of this chapter is largely based on the very recent state of the art on the development and application of machine learning methodologies in the domain of remote sensing. The focus is on techniques with a relevant and significant contribution to the state of the art during the last 12-18 months. To illustrate the very broad range of topics covered by the recent literature, figure 2 displays a word cloud generated by the titles of all relevant papers on machine learning for remote sensing in the period January 2015 - April 2016.

Figure 2: wordcloud based on the titles of papers of machine learning for remote sensing in the period January 2015 - April 2016 (main words such as hyperspectral and classification were removed).

Based on our analysis of the main topics emerging, we organized the rest of the chapters in six groups of methodologies:

1. *kernel methods*, where models are based on data similarity and projections into higher dimensional spaces;
2. *neural network methods*, where the input-outputs relations are learned by backpropagating errors through a series of operations learned from data;

3. *manifold learning methods*, where the underlying nonlinear structure of the data is taken into account by the model;
4. *structured output learning methods*, where prior knowledge about the problem to be solved is encoded as structural relationships among the outputs (e.g. in the spatial domain);
5. *ensemble learning methods*, where, instead of using a single method for prediction, many imperfect models are run and then their average is used as a solution able to generalize better;
6. *sparse learning methods*, where models reduce the dependency on specific training data or features by selecting subsets that are relevant to the problem.

In the following, a section is devoted to each of these groups. Each section contains a short introduction with basic notions and methods, a review of the recent state of the art, with a conceptual description of some more involved methods.

2. Kernel methods

One of the biggest success stories of machine learning in remote sensing is certainly kernel methods (Camps-Valls and Bruzzone, 2009). Learning with kernels allows to cast the image classification and retrieval process in a sound mathematical framework rooted on two main statistical principles: *similarity functions*, i.e. the kernels themselves – as ways to approach distances in high dimensional spaces – and *regularization*, as a way to avoid overfitting to enhance the generalization capabilities of methods.

Basically, a kernel method is a method where the decision about class membership (or continuous values or cluster membership) is made using only the similarity between samples. Practically speaking, this means that in all the operations involved in method, the input data appear only in distance computations casted as *dot products*. Why is this important? Because kernel methods procedures that apply to linear algorithms, which are generally simple and fast, and make them nonlinear by considering the data structures occurring in some higher dimensional feature space, where a complex and nonlinear structure observed in the inputs becomes, de facto, linear. Let us take the example of Fig. 3, depicting the use of a linear classifier in a kernel framework: in the original space, the data cannot be separated by a linear classifier (Fig. 3a), but when using the right projection (Fig. 3b) the training data can be projected into a space, where a linear model can be run efficiently and lead to a simple linear solution in the higher dimensional space (Fig. 3c), which is also a nonlinear complex decision function in the original space (Fig. 3d).

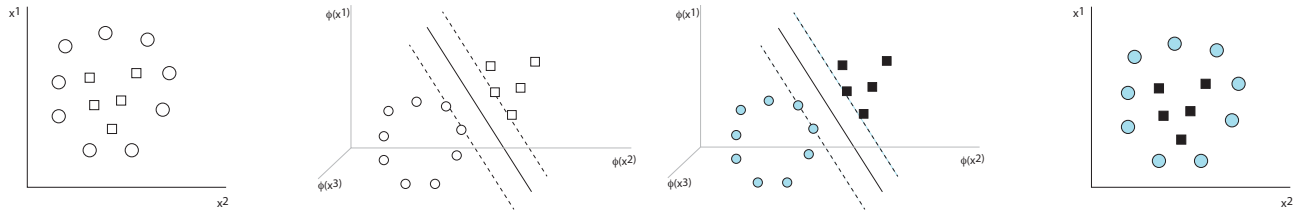


Figure 3: Intuition of the kernel trick: nonlinear separable data (a) are projected in a higher dimensional space (b) where linear separation is possible (c). Back in the original space, classification is nonlinear.

2.1. An intuition via the Support Vector Machine

The classifier in Fig. 3 is the well-known *Support Vector Machine* (SVM, first proposed by Boser et al. (1992)). SVM is probably the most used classifier in recent remote sensing literature (Melgani and Bruzzone, 2004; Camps-Valls and Bruzzone, 2005; Mountrakis et al., 2011; Camps-Valls et al., 2014). We start by reviewing its functioning, in order to better understand the role of the kernel and of the regularization.

As stated above, SVM is a linear binary classifier, which finds a decision function of the type $f(\mathbf{x}) = \mathbf{w}\mathbf{x}+b$, where b is a bias term, $\mathbf{x}$ are the data to be predicted in $\mathcal{R}^d$ and $\mathbf{w}$ is a vector of model weights, with $(d \times 1)$ values. It finds

the best separating hyperplane between two classes as the one maximizing the distance between the closest samples of each class :

$$\min_{\mathbf{w},\xi_i,b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \right\}, \tag{1}$$

under the following constraints:

$$y_i(\langle \boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}\rangle + b) \geq 1 - \xi_i \qquad \forall i = 1,\ldots,n \tag{2}$$

$$\xi_i \geq 0 \qquad \forall i = 1,\ldots,n \tag{3}$$

The distance being maximized is called the *margin* and corresponds to the left hand side term in Eq. (1). SVM maximizes the width of the margin, while ensuring perfect classification (this is enforced by the constraint in Eq. (2)). However, in order to avoid being too strict on the latter point, it also allows some errors to be committed: the right term of Eq. (1) sums a ξ_i positive penalization (Eq. (3)) every time the classifiers makes a mistake on training samples, thus allowing errors, while keeping them to a minimum. The two-terms of Eq. (2) show the inner structure of the SVM method: as explained above, the left part is the margin-maximization part, which limits too large coefficients in the final weights $\mathbf{w}$. This corresponds to finding the weights that depend as little as possible to the training data. Such a squared-norm of coefficients is also often called a *regularizer* (Vapnik, 1998). Minimizing this term only leads to finding the simplest function possible coherent with the training data and therefore avoids overfitting the training data (and in turn performing poorly on new, unseen data). The second term, in the right hand side of Eq. (1) counteracts the former, by penalizing all classification mistakes done in the training area. This term, alos called a *loss function* avoids choosing a function that would be too simplistic and that, even if minimizing the role of training data, would lead to several samples being missclassified. The minimization in Eq. (1) bering the best of both worlds: being correct on the training data, while also keeping the model as simple as possible.

The solution of SVM is found by applying Lagrangian multiplying and equating partial derivatives of the model to 0. After some mathematical operations (the interested reader is invited to consult the books by Vapnik (1998); Schölkopf and Smola (2002)), the dual formulation of the SVM becomes:

$$\max_{\alpha} \left\{ \sum_{i=1}^{n}\alpha_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle \right\} \tag{4}$$

subject to:

$$0 \leq \alpha_i \leq C \qquad \forall i = 1,\ldots,n \tag{5}$$

$$\sum_{i=1}^{n} y_i\alpha_i = 0 \qquad \forall i = 1,\ldots,n \tag{6}$$

Once the α coeficients have been found, the dual final solution for a test sample $\mathbf{x}_*$ is :

$$y_* = \operatorname{sgn}\Big(f(\mathbf{x}_*)\Big) = \operatorname{sgn}\left(\sum_{i=1}^{n} y_i\alpha_i\langle \mathbf{x}_i, \mathbf{x}_*\rangle + b\right) \tag{7}$$

where $\operatorname{sgn}(\cdot)$ is the *sign(·)* function, returning whether the function $f(\mathbf{x})$ is positive or negative.

So far, no kernels were employed. However, one can notice that both in the problem dual formulation (Eq. (4)) and in the final decision function (Eq. (9)) the input patterns appear only as dot products, the first computing similarities between all couples of training samples and the latter computing similarities between the test sample being considered and all the training samples.

When considering nonlinear problems, using a linear classifier is not possible. We need to find a nonlinear solution or cast our linear classifier in a projected, higher dimensional Hilbert space $\mathcal{H}$ where linear classification can be achieved. This is true by the Cover theorem (Cover, 1965) stating that the probability of linear separation increases with the number of dimensions. In fact, even if the solution will be linear in $\mathcal{H}$, it will be nonlinear in the original

space $\mathcal{X}$. The "only" thing we need to define is a mapping function to that space $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{H}, \mathbf{x} \mapsto \phi(\mathbf{x})$. Of course finding $\boldsymbol{\phi}$ is no easy task: there is an infinity of possibilities, and one does not want to try them all. Moreover, they can be infinite dimensional, which makes the computation of the actual coordinates $\phi(\mathbf{x})$ very complex.

This is where kernel functions come in. Kernel functions are measures of similarity between samples which correspond to the dot product of the samples in a higher dimensional Hilbert space. In short, kernels take as inputs the samples in their original input space and return their similarity in the projected one $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that $(\mathbf{x}, \mathbf{x}') \mapsto K(\mathbf{x}, \mathbf{x}')$. We can define them as:

$$K(\mathbf{x}, \mathbf{x}') = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') \rangle_{\mathcal{H}}. \tag{8}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes a dot product in the Hilbert space. By using this function, we can substitute the dot product of the SVM with the kernel and use exactly the same algorithm to perform nonlinear classification. The decision function in Eq. (4) becomes:

$$y_* = \operatorname{sgn}\Big(f(\phi(\mathbf{x}_*))\Big) = \operatorname{sgn}\left(\sum_{i=1}^{n} y_i \alpha_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_*) \rangle + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{n} y_i \alpha_i K(\mathbf{x_i}, \mathbf{x}_*) + \mathbf{b}\right). \tag{9}$$

As such, the original dot product is a valid kernel function projecting the original data into the input space itself.

2.2. Interest of kernel methods in remote sensing

The interest of using kernel methods in remote sensing is to obtain a nonlinear solution by only slightly modifying existing algorithms: if an algorithm can be expressed in a way such that the input data only appear as dot products, it can be kernelized. Beyond classification, recent efforts in this direction include kernelization of :

- feature extractor methods as the principal component analysis (Fauvel et al., 2009) or the minimum noise fraction (Nielsen, 2011);
- clustering methods as the k-means (Volpi et al., 2012a);
- target detection methods such as the RX-detector, the adaptive subspace detector of the matched filter (Kwon and Nasrabadi, 2007);
- regression methods, including kernel ridge regression (Camps-Valls et al., 2011), support vector regression (Camps-Valls et al., 2008b; Tuia et al., 2011), Gaussian processing (Verrelst et al., 2011).

But the interest of kernel methods goes beyond the simple exercise of kernelization: beyond a solid mathematical framework, kernel methods also provide a very flexible framework for data combination: kernel functions can also be *derived* from other kernel functions. New kernels can be built by taking advantage of a set of rules which define the validity of some operations *between* kernels (Schölkopf and Smola, 2002). For valid kernels (i.e. symmetric, positive semidefinite kernels), the following combinations lead to other valid kernels:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-d(\mathbf{x}, \mathbf{x}')) \tag{10}$$
$$K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}') + K_2(\mathbf{x}, \mathbf{x}') \tag{11}$$
$$K(\mathbf{x}, \mathbf{x}') = \mu K(\mathbf{x}, \mathbf{x}'),\ \mu > 0 \tag{12}$$
$$K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}') \cdot K_2(\mathbf{x}, \mathbf{x}') \tag{13}$$

This means that the sum of two kernels, their element-wise product or any exponentiation of a negative distance is a kernel, and thus opens wide opportunities to exploit specific characteristics of remote sensing data within the kernel framework.

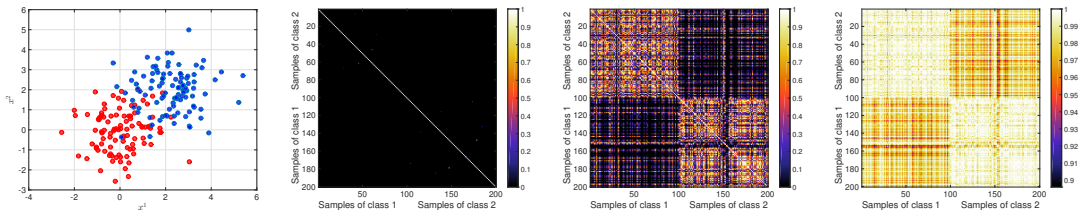
The first property (Eq. (10)) leads to the popular Gaussian kernel:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-2\gamma^2 \|\mathbf{x} - \mathbf{x}'\|_2). \tag{14}$$

Table 1: summary of PCA, PLS and their corresponding kernel versions (from Izquierdo-Verdiguier et al. (2014)). Tilde symbols are for centered matrices.

	PCA	PLS	KPCA	KPLS
Max. problem	$\mathbf{u}^\top \mathbf{C}_{xx}\mathbf{u}$	$\mathbf{u}^\top \mathbf{C}_{xy}\mathbf{v}$	$\boldsymbol{\alpha}^\top \tilde{\mathbf{K}}_x \tilde{\mathbf{K}}_x \boldsymbol{\alpha}$	$\boldsymbol{\alpha}^\top \tilde{\mathbf{K}}_x \tilde{\mathbf{Y}}\mathbf{v}$
Constraints	$\mathbf{u}^\top\mathbf{u} = 1$	$\mathbf{u}^\top\mathbf{u} = 1, \mathbf{v}^\top\mathbf{v} = 1$	$\boldsymbol{\alpha}^\top \tilde{\mathbf{K}}_x \boldsymbol{\alpha} = 1$	$\boldsymbol{\alpha}^\top \tilde{\mathbf{K}}_x \boldsymbol{\alpha} = 1, \mathbf{v}^\top\mathbf{v} = 1$
Max # features	rank($\tilde{\mathbf{X}}$)	rank($\mathbf{C}_{xy}$)	rank($\tilde{\mathbf{K}}_x$)	rank($\tilde{\mathbf{K}}_x \tilde{\mathbf{Y}}$)

This kernel is easy to interpret, as it yields maximum similarity between identical samples and decreases together with Euclidean distance, at a rate depending from the bandwidth parameter γ. The Gaussian kernel has similarity bounded between $[0, 1]$. Eq. (14) computes the similarity between two samples, $\mathbf{x}$ and $\mathbf{x}'$, in a Hilbert space of dimension as high as the number of training samples. If we compute these scalar values for each couple of training samples, we obtain a kernel matrix $\mathbf{K}$ containing all the pairwise similarities among the training samples. Some examples of Gaussian kernel matrices are provided in Fig. 4 for the 2-classes example in Fig. 4a: if the bandwidth is chosen too narrow, as in Fig. 4b, each trainig sample will be similar only to itself and similarity with other samples will be practically 0. In this case, a kernel method can learn hardly anything. When the bandwidth is too large (Fig. 4d), every sample is strongly similar to all the others: in that extreme case learning is very difficult too, as the relationships between samples of the same class (diagonal blocks of the matrix) look like those with samples of the other class (off-diagonal block). The kernel matrix shown in Fig. 4c is the one we are aiming at: in this case, samples of the same class look alike ($K(\mathbf{x}, \mathbf{x}') \sim 1$), while samples of different classes have low similarity ($K(\mathbf{x}, \mathbf{x}') \sim 0$).



2D data (a) $\gamma = 0.1$ (b) $\gamma = 0.8$ (c) $\gamma = 3$ (d)

Figure 4: Intuition of the Gaussian kernel matrix. Given a 2-classes problem (a), a kernel can have a (b) too short; (c) correct or; (d) too large bandwith.

2.3. Trends in kernel methods

In this final section, we review some recent trends in the use of kernel methods in remote sensing and provide pointers to literature.

2.3.1. Kernels for feature extraction

Feature extraction is the unavoidable preprocessing, when working in high dimensional spaces such as hyperspectral image cubes, Fisher vectors or spaces enhanced by spatial filters. Extracting filters that are informative given a criterion of interest, is of primary importance for the success of image processing pipelines. Nonlinearizing these methods through kernelization is therefore very appealing, since assumptions of linearity sometimes lead to poor description of the data. As mentioned above, traditional dimensionality reduction methods such as PCA have been kernelized and tested in remote sensing as nonlinear feature extractors (Fauvel et al., 2009). Extraction of features representing the entropy of the input space has been proposed in (Gomez-Chova et al., 2012).

More recently, research has been pushed further with feature extractors encoding other desirable properties of the projected space: class-discrimination can be enforced using feature extractors such as Partial least squares (PLS), whose efficiency in remote sensing problems has been shown in Izquierdo-Verdiguier et al. (2014). Table 1 summarizes the formulations of the linear and kernel versions of these methods.

Transforms that work over several data sources and make them more similar (or *multi view methods*) have been explored, where Canonical correlation analysis is kernelized and applied to the processing of hyperspectral (Volpi et al., 2014) and multi-sensor (Volpi et al., 2015) data. Finally, a transform effective in both ways (therefore discriminative and making data sources more similar) has been recently proposed in Tuia and Camps-Valls (2016).

2.3.2. Kernels composition and the way to multiple kernels

Combining data from different sources is one of the most appealing strategies when dealing with remote sensing data: as the images are geo-referenced, it is relatively easy to observe a single region with a variety of sensors. Moreover, the possibility of adding any kind of signal filtering as additional features opens a wide range of possibilities in terms of deformation of the kernel function to account for the signal specificities of each source. This idea, previously referred to as the *composite kernels* framework (Camps-Valls et al., 2006) directly stems from Eqs. (11) and (12): any function issued by a convex combination of other valid kernels is a valid kernel. Otherwise said:

$$K(\mathbf{x}_i, \mathbf{x}_j) = dK_1(\mathbf{x}_i, \mathbf{x}_j) + (1-d)K_2(\mathbf{x}_i, \mathbf{x}_j), \tag{15}$$

where $0 < d < 1$ is a scalar weight parameter tuning the importance of each kernel in the combination. This idea was exploited for spatio-spectral classification (Tuia et al., 2010; Li et al., in press; Huang and Zhang, 2013; Huo et al., 2015), multitemporal classification (Camps-Valls et al., 2008a), decision fusion (Thoonen et al., 2012) and semisupervised kernel deformation, where a Gaussian kernel was combined with a structural unsupervised kernel providing the probability that two samples end up in the same cluster (Tuia and Camps-Valls, 2011). Finally, spatial neighborhood relationships in the feature space were studied by Liu et al. (2013); Gurram and Kwon (2013), who in turn proposed to compute a kernel corresponding to an average in the feature space (a *mean map kernel (Gómez-Chova et al., 2010))* and combine it with the kernel accounting for single inputs only. In Moser and Serpico (2013), the Marovianity of a Markov random field is casted as a kernel deformation and a composite kernel. Composite kernels provide an intuitive way of tradeoff the importance of different feature sets and have therefore driven a large amount of research. However, they are not suitable for cases involving more than a few data sources, since the tuning of a set of d weights heuristically would become computationally expensive.

To answer to this problem, research has turned to *Multiple kernel learning* (MKL), which is basically a way to solve the composite kernels problems efficiently, when several sources are considered. In this case, the convex combination is given by:

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) \quad &= \quad \sum_{m=1}^{M} d_m K_m(\mathbf{x}_i, \mathbf{x}_j), \\ s.t. \quad & \quad d_m \geq 0 \\ & \quad \sum_{m=1}^{M} d_m = 1. \end{aligned} \tag{16}$$

where $\mathbf{d}$ has become a vector, with one entry per data source. MKL aims at optimizing the convex linear combination of kernels, i.e. the d_m weights, while training the classifier (Rakotomamonjy et al., 2008).

Multiple kernel learning is receiving a lot of attention in recent remote sensing literature, going from multisource image classification (Tuia et al., 2010; Wang et al., in press), to unmixing (Gu et al., 2013; Liu et al., 2015b), SAR segmentation (Gu et al., 2016), feature selection (Gu et al., 2012), or as a way to combine spatio-spectral indices (Cusano et al., 2015). In Wang et al. (2015b), authors aimed to obtain the kernel by repeated nonlinear mappings. Multiple kernel learning was considered in combination with other classification frameworks: for example with active learning for intelligent training samples selection (Zhang et al., 2015d) or in domain adaptation (Sun et al., 2013).

2.3.3. Structured outputs with kernel methods

As discussed above, including spatial information is one of the major benefits for remote sensing image classification. Beyond spatial filters – that can then be used to build the kernel –, one can also decide to enforce spatial consistency in the outputs space. Without entering in details of structured prediction (the interested reader can go to Section 5 of this Chapter), recent research has considered the use of kernels to make Conditional random fields (CRF)

models more accurate via the use of *contrast-sensitive priors*. A contrast sensitive prior is a pairwise prior (i.e. a measure estimating the cost of attributing two neighboring samples into two classes) that, besides estimating the cost as a function of the output classes considered, also accounts for similarity between samples in the input domain. In other words, if a simple Potts penalization looks like:

$$V'(y_i, y_j) = [1 - \delta(y_i, y_j)], \tag{17}$$

where $\delta(y_i, y_j)$ is a function returning 1 if pixels i and j are classified in the same class and 0 otherwise. This means that a cost of 1 is given if two neighbors are assigned to different classes and that no penalization will be applied if the neighbors are assigned to the same class. Since this has been shown to promote oversmoothing of the final maps, contrast-sensitive penalizations have been proposed to take into account the similarity between the input (the $\mathbf{x}$ vectors) too:

$$V'(y_i, y_j|\mathcal{X}) = [1 - \delta(y_i, y_j)]K(\mathbf{x}_i, \mathbf{x}_j). \tag{18}$$

This means that two samples assigned to the same class will never be penalized, while two samples assigned to different classes will be penalized according to their degree of similarity estimated by the kernel. In other words, this penalization allows to attribute different classes if two regions really are not alike (probably since they belong to different objects) and avoid oversmoothing of the CRF. This type of penalization has been used recently in Schindler (2012); Tuia et al. (2016); Volpi and Ferrari (2015b).

Another recent development in kernel methods connected to structured outputs is to encode relations in the outputs that are learned by a so-called *structured support vector machine* (SSVM). In this case, the SSVM learns an SVM model with a loss depending on the structure of the outputs too, which has been explored as a tree-shaped loss corresponding to class-similarities (Tuia et al., 2011) or as a set of pairwise weights in a CRF, as in Volpi and Ferrari (2015a).

2.3.4. Sparse representation and subspace learning

Sparse representation-based classifiers (more details in Section 7) are valid challengers to discriminative models such as the support vector machine and some of the methods in the literature have also been kernelized: this is somehow logic, since the subspaces where the data live do not need to be linear and kernel methods are a valid way to describe complex manifolds (nonlinear, intersecting, multimanifold) using linear algebra. Classifiers based on collaborative representation (Li et al., 2015c; Liu et al., 2016b) and subspace-regularized graphs (de Morsier et al., in press) are few recent examples of methods for sparse representation made nonlinear through kernels.

2.3.5. Active learning

Active learning (Tuia et al., 2011; Crawford et al., 2013) is an area of research that has emerged in the last years. With active learning, one aims at increasing the number of labeled samples to use to train a model. But unlike random and stratified sampling, which select all labeled samples before training, active learning uses the uncertainty of a trained model as a driving criterion for selection of new labeled samples. This way, machine and human interact, the former asking for the labels of the samples the model is being the most uncertain about, and the latter looking into these samples and providing the correct label to be added to the training set. The success of active learning methods in remote sensing can be explained by the advent of very high resolution imagery, since photointerpretation has become a more affordable option to gather the additional labels in an online fashion (even though studies deploying active learning on field campaigns are found in the literature, see e.g. Demir et al. (2014)).

But regardless of the final aim, kernel methods have been extensively used in active learning. First, because SVM is probably the most used algorithm in active learning routines, mainly since the uncertainty of the classifier can be easily obtained as the value of the decision function itself (Eq. (9), without the *sign(·)* operator). As only support vectors are accounted for in the decision function and have $|f(\mathbf{x})| = 1$, when an unlabeled new sample scores a decision function $|f(\mathbf{x}) \leq 1|$, it automatically falls within the margin, and therefore in the uncertainty zone of the classifier (Tuia et al., 2009). Therefore, these samples are those selected for labeling. Secondly, because kernels can be used to account for diversity between samples: when selecting new samples for labeling, one wants to select samples that are different to each other, in order to make the labeling effort the most useful for the classifier: in Demir et al. (2011), authors use kernel k-means to select samples belonging to different clusters of unlabeled samples in the feature space, while in Volpi et al. (2012b) a similar logic is exploited to sample new examples different from each other, and also from those selected at previous iterations.

2.3.6. Domain adaptation

A last area, where kernels are being extensively accounted for is *domain adaptation* (Tuia et al., 2016). In domain adaptation, one seeks to transfer models from one acquisition to another, in order to exploit existing labels to process unlabeled (or scarcely labeled) new acquisitions. First approaches to domain adaptation with kernels were based on the adaptation of an SVM (Bruzzone and Marconcini, 2010) or accounting for the manifold structure (see Section 4.1.4 on manifold regularization and Kim and Crawford (2010); Gómez-Chova et al. (2008)). Later on, many methods relying on feature extraction strategies have been proposed: among them one can find solutions based on KPCA (Matasci et al., 2015a), centered mean kernels (Gómez-Chova et al., 2010), manifold alignment (Tuia and Camps-Valls, 2016) or other domain invariant kernel projectors as the Transfer Component Analysis (Matasci et al., 2015b). Another recent interesting use of kernels in domain adaptation lies in the selection of features that are invariant in their feature space to the domain shift (Persello and Bruzzone, 2016). Another new research direction is the direct encoding of invariances of interest in a kernel classifier: in Izquierdo-Verdiguier et al. (2013), authors encode invariances to spatial scaling, objects rotation and shadowing by using virtual support vectors, i.e. by applying the distortion for which we want to become invariant to the SVM support vectors and training a new model with the real and virtual new training samples. In all cases, kernels provided flexible solutions to minimize the distance between domains and ease the transfer of classifiers to new scenes. Finally, multi-task methods have also been considered, where each sub-classification problem is seen as a separate (but similar) task and a cost based on all the tasks is optimized simultaneously (Leiva-Murillo et al., 2013).

3. Neural Networks

In this section, we present the family of methods known as Neural Networks (NN, Haykin (1999)). We will provide an introduction on feed forward neural networks and then move to Convolutional Neural Networks (CNN, LeCun et al. (1998)) as new learning paradigms for feature learning and prediction.

3.1. Neural networks: principles

Neural networks are models that learn from data (as the kernel methods presented above). They relate inputs (in the case of remote sensing, the value of the pixels in the bands, or spatial filters, etc.) to outputs (the function to be estimated, which can be continuous (in the case of biophysical parameters estimation (Fang and Liang, 2003)) or discrete (in the case of land cover classification (Benediktsson et al., 2005)). They do so by constructing a network of connected simple computational units called *neurons*.

3.1.1. Neurons

Neurons are basic processing units combining information from some inputs into a single output via a function $f(\mathbf{x})$:

$$f(\mathbf{x}) = g(A). \tag{19}$$

This means that a neuron takes information form a data vector $\mathbf{x}$ (for example a value for each band) and re-combines it into a single number $f(\mathbf{x})$ according to its mathematical architecture. Neurons are made of three basic components (Fig. 5):

1. A set of weights connecting all the elements of the neuron in a forward way, w^j, with j being the input to the neuron being considered. Often an additional weight is used, which corresponds to the bias of the neuron. In the following, we omit it for clarity.
2. The integration operator A. A is generally a linear combination of the inputs and the weights:

$$A = w^\top x \tag{20}$$

3. A nonlinearity $g(\cdot)$ limiting the amplitude of the neuron, and also permitting learning more complex functions. Typical nonlinearities are the sigmoid (or logistic) function

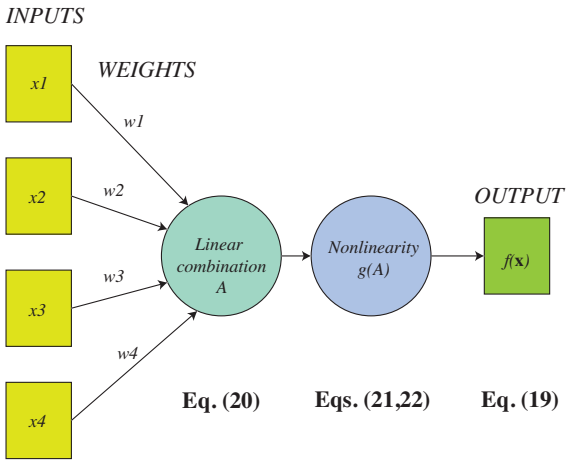


Figure 5: Sketch of a neuron combining a set of inputs $x^{(i)}$ into an output using the relation of Eqs. (19) to (22).

$$g(A) = \frac{1}{[1 + \exp(-A)]}, \tag{21}$$

which limits the output in the range [0,1], and the rectified linear unit (ReLU, Nair and Hinton (2010)) function

$$g(A) = max(0, A), \tag{22}$$

which is a non-saturating nonlinearity that changes all negative outputs to 0.

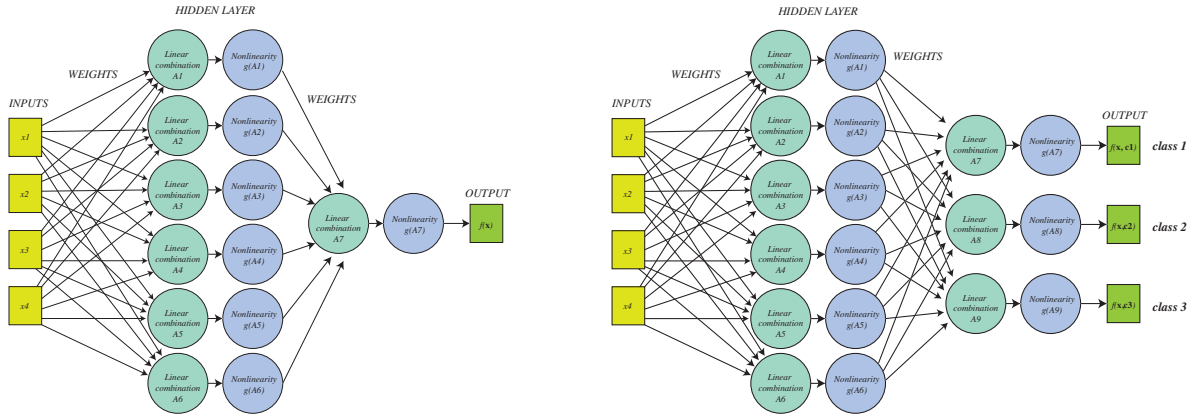
If a single neuron corresponds to the whole model being used, it is called a *perceptron* (Haykin, 1999).

3.1.2. Networks of neurons : the example of multilayer perceptron

Let's assume we know how to learn the parameters $\mathbf{w}$ (we discuss how to learn them in the next Section). Neurons are the building block of a neural network, where sets of neurons are used to learn different properties of the data. A first thing that can be done is to connect several neurons in sequence: this is called a *Multilayer Perceptron* (MLP, Haykin (1999)) and is represented in Fig. 6 for both regression (left side) and classification (right side).

An MLP is a *feedforward neural network*, where feedforward means that the information follows a sequential flow from the inputs, all the way down to the predicted output (without loops, as can be seen in the arrows of Fig. 6). Moreover, the network of neurons is organized in *layers*: the inputs layer, hidden layers and a final combination layer leading to the final output(s). If the input and combination layers are easily interpreted, the hidden layer has the role of learning the specific properties about the data: since each neuron in the hidden layers receives a different share of the input information (the weights going into each neuron are different), each neuron specializes on a different aspect of the training data and – by the minimization of a loss – extracts progressively meaningful information about the data. The specific aspect of each neuron is defined during the training phase, which boils down to learning the values of the weights of the neuron. Another thing worth noticing is that the neurons of different layers are generally fully connected (for example, in the network in Fig. 6, each input provides information to all neurons in the hidden layer, while the neuron of a specific layer are not connected to each other (they 'ignore' each other's existence until they are combined in the next layer).

Why are hidden layers required? When using a simple network as the single-layer perceptron depicted in Fig. 5, the neuron only learns a linear combination of the input data. If the data relationships are more complex than that, one would need a model of greater capacity (i.e. able to learn more complex functions): to do so, hidden layers can be used to learn different structures in the data (each neuron in the hidden layer learns something different) and recombine them nonlinearly (each neuron has its own nonlinearity). Nonlinearities are important, since on one hand they allow to learn more complex functions, but also they give a reason to chaining the layers, since learning a set of linear combinations without nonlinearities would boil down to learning a single linear combination of weights. This



a) MLP for regression

b) MLP for classification (3 classes)

Figure 6: Sketch of a Multilayer Perceptron with one hidden layer composed of 6 neurons. On the left, an example for regression (single output): there are 24 + 6 weights to be learned. On the right, an example for classification (three classes, three outputs, corresponding to one score per class): there are 24 + 18 = 42 weights to be learned.

is what the MLP does. Adding more hidden layers is also possible (we then start to talk about *deep networks*), which results in learning more abstract properties of the data (the nonlinear combinations are then combined again), but at the price of increasing drastically the number of weights to be learned.

3.1.3. Backpropagation

It is now time to discuss how an MLP learns the weights $\mathbf{w}$. Intuitively, MLP learns by confronting some known outputs to those predicted by the current network. Knowing the current error, it then goes back and changes the network weights responsible for the largest share of the current error by computing gradients. Then the new network sees again the training data, predicts the output and repeats the procedure of weights update until the error becomes small enough. We call this procedure *backpropagation* (Rumelhart et al., 1986a,b).

From this intuitive description, one can see that the space of possible solution is very large and complex (non-convex), as each weights combination and nonlinearity give rise to a function. One had to find a computationally efficient and practical algorithm to the weights learning problem, which in the end corresponds to decide what the hidden units should represent (Hinton, 1989). Backpropagation was an answer to this call. As can be seen from the intuition above, it is a supervised algorithm to learn the best weights $\mathbf{w}$, given some input-output pairs $\{\mathbf{x}_i, y_i\}_{i=1}^n$. It is based on a *cost function*, comparing predictions over the training samples, y_i^* to the true values, y_i. A typically used cost in classification is the softmax cost:

$$L_i = -\log\left(\frac{\exp[f(\mathbf{x}_i, y_i)]}{\sum_{cl}\exp[f(\mathbf{x}_i, cl)]}\right). \tag{23}$$

The $f(\mathbf{x}_i, j)$ function is the nonlinearity function that has as many entries as there are classes (see the right-most green squares in Fig. 6b), for example the sigmoid function, but also sums to 1. For class j, its entry is:

$$f(\mathbf{x}_i, j) = \frac{\exp[g(A, j)]}{\sum_{cl}\exp[g(A, cl)]}, \tag{24}$$

where $g(A, j)$ is the output of the last layer for class j. In other words, the softmax loss function of Eq. (23) returns the smallest loss when all the mass (the probability for all the classes) is attributed to the correct class for the training sample y_i.

The loss should be averaged over the whole training set. To allow efficient learning (and simpler implementations), sequential backpropagation (or *stochastic backpropagation*) computes the update of the weights after each training sample has been considered, thus only approximating the average error of Eq. (23) and updating the weights after

seeing each sample. If at every iteration the order of patterns presented is randomized, stochastic backpropagation proves to work very well in practice. Another practical point, typically used in deeper networks and for computational reasons when dealing with very large training sets, is to update the weights after minibatches of the training data have been passed through the network, instead than as after every individual sample. The labeled pixels in the minibatch are re-drawn every few iterations.

Let's now provide an intuition on how backpropagation of weights works. Consider a weight connecting input $\mathcal{I}$ to output $\mathcal{O}$ at iteration t, $w_{\mathcal{IO}}(t)$. The output can be a linear combination, a nonlinearity or the final output. Backpropagation basically decides how to modify $w_{\mathcal{IO}}(t)$ by looking how much its variation impacts the loss $L(t)$ for the minibatch considered: this corresponds to a correction $\Delta w_{\mathcal{IO}}(t)$ applied to the weight value $w_{\mathcal{IO}}(t)$, as:

$$w_{\mathcal{IO}}(t+1) = w_{\mathcal{IO}}(t) + \Delta w_{\mathcal{IO}}(t). \tag{25}$$

The $\Delta w_{\mathcal{IO}}(t)$ correction is proportional to the derivative of the error with respect to the weight, multiplied by a learning rate η, which rules how much we modify the weights at each iteration:

$$\Delta w_{\mathcal{IO}}(t) = -\eta \frac{\partial L(t)}{\partial w_{\mathcal{IO}}} \tag{26}$$

This is also called a *gradient*. The general idea behind backpropagation is that each weight $w_{\mathcal{IO}}(t)$ is modified in the opposite direction of maximal gradient with respect to the current error. A gradient is the direction of change in the error function if we modify slightly the value of a specific weight: a positive gradient means that if we change the weight a tiny little bit, the error should decrease by a factor proportional to the step taken. The parameter η decides the length of such step, i.e. by how much we change the weights. It must be carefully tuned, to avoid both stepping too far from the current solution (with the risk of ending up in a worse one and not converging) and not changing anything at all (when η is too small). The gradient for each weight is estimated in turn, and propagated backward layer by layer: this way backpropagation allows to compute all gradients (for each weight in the network) in a single backward pass[1].

Summing up, the whole backpropagation procedure is performed sequentially for each training sample (or for minibatches of samples) and for each iteration (one iteration is achieved when all training samples have been seen by the network):

- The forward pass: the weights remain fixed and training samples $\mathbf{x}_i$ are passed through the network, from the inputs to the final outputs. At the end we have a prediction y_i^*. If there is a single output neuron, y_i^* is a scalar, if there are several output neurons (e.g. one per class in multiclass classification), y_i^* is a vector of scores, one per class. Finally, the loss is computed, for instance using Eq. (23).

- The backward pass: the loss is used to compute the gradient of the weights. The gradient is backpropagated recursively through the network and all the weights are updated using the update rule in Eq. (25). The sense of propagation is from the outputs (where $L(t)$ is computed) all the way back to the inputs.

3.1.4. Neural networks in remote sensing

Neural networks have met the remote sensing community interest since the end of the '90*s*, where application of MLP can be found mainly in land cover classification (Benediktsson et al., 2005), change detection (Pratola et al., 2013) and biophysical parameters estimation (Fang and Liang, 2003; Baret et al., 2007) and have remained a state of art approach ever since, against which new methodologies are compared.

In classification, feed-forward neural networks are victims of the success of kernel methods and are mostly considered for benchmarking against new methods both for land cover (Shao and Lunetta, 2012; Khatami et al., 2016) and forestry (Feret and Asner, 2013) applications. A rapid and fast renewal of interest is being observed with deep learning (see the discussion in the next section, here we limit ourselves to classical MLPs).

In biophysical parameters estimation, neural networks are among the most used algorithms for model inversion and a valid competitor to lookup tables on outputs of radiative transfer models: they have been recently used in studies

[1]For the mathematical details of the gradient computation in neural networks, we refer the reader to the really intuitive Stanford course by Karpathy (classes 3 and 4 of `http://cs231n.github.io/`).

considering vegetation parameters (Verrelst et al., 2012), biomass estimation (Cutler et al., 2012) surface temperature (Shwetha and Kumar, 2016), soil moisture (Rodríguez-Fernández et al., 2015), hydrological parameters (Blackwell, 2011; Blackwell and Milstein, 2014) or the derivation of inherent optical properties of oceans (Ioannou et al., 2013). As for classification, they are a gold standard methodological benchmark, as seen in the intercomparison reported in Verrelst et al. (2012) for vegetation parameters or the one in Englhart et al. (2012) for forest biomass estimation.

A last category that is starting to renew interest is spectral unmixing using neural network methods. As abundance estimation can be seen as a model inversion process (under some additional constraints as positivity and sum-to-one), neural networks can be designed to perform the estimation, as pointed out in a recent review (Heylen et al., 2014) or in specific applications, such as the estimation of fractional land cover types in high resolution images (Mitraka et al., in press).

3.2. Convolutional neural networks : the rebirth of neural networks

Since the methodological remote sensing community had focused most efforts on kernel methods for more than a decade, one could have argued that the interest for developing neural networks models was over at the end of the $2000s$. This was mainly due to the general opinion that neural networks were harder to train than convex models such as SVMs (see Section 2.1, page 4) However, some developments in computer vision decided otherwise. A specific type of neural networks, tailored to the processing of images and developed in the '90s, resurfaced: *convolutional neural networks* (CNN, LeCun et al. (1998)). Taking advantage of the advances in data availability and computing resources (in particular graphic processing units), which allowed to design deeper neural network architectures, CNN recently won most data processing competitions in computer vision by pulverizing all state of the art and became a new standard in data processing. Major data mining companies and major universities followed on these findings and focused most of their research efforts in that direction, thus creating a wave of interest for deep learning and CNN, that are, presently, the most researched ones in computer science and vision. This section provides a brief introduction to the main principles of CNN and pointers to the very recent research in deep learning for remote sensing.

3.2.1. Intuition

CNN architectures are specific to images. They use d-dimensional convolutions as filters, and thus learn implicitly contextual information. As for MLPs, they are composed of building blocks connecting input images to a specific output (e.g. a class to be predicted for the image) via a sequence of layers (Fig. 7): the main difference is that each filter is a convolution, i.e. a $(m \times m \times d)$ filter performing a linear combination of nearby pixels, both in color and spatial terms.

In CNNs, there are three types of layers:

- *Convolutional layers*, involving filters that are shared across the image (Fig. 8). In the example of Fig. 7, the first layer learns 5 filters, the second 10 and the third 40. This means that a filter in a convolutional layer is convolved over the whole input image and the activations recorded, but the filters are shared across the image. This actually helps keeping the computational cost feasible (modern CNNs involve millions of parameters to be learned, but if we had to learn different parameters for each location in one image it would become impossible). A convolutional layer is the more often composed of three building blocks (Fig. 8):

    1. A convolution filter. We have a $(r \times c \times d)$ image and we learn a series of nf different $(3 \times 3 \times d)$ filters. Each filter is convolved through the image and for each local data cube (in orange in INPUT) produces a single value, corresponding to a linear combination of the INPUT and filter values. For the orange cube, this leads to the orange square in the second image.
    2. A nonlinearity. After convolution, a nonlinearity (often a ReLU, see Eq. (22)) is generally applied as in traditional neural networks.
    3. A spatial pooling step. We want the convolutional network to be invariant to translations, and also to reduce spatial information, while increasing the size of the feature space. Both reasons actually work for the success of the network, the former since we cannot know a priori the position of the object the image is representing, and the second since we want to learn more and more abstract (and complex) representations of the data in successive operations. we do so with a spatial pooling, which corresponds to summarizing

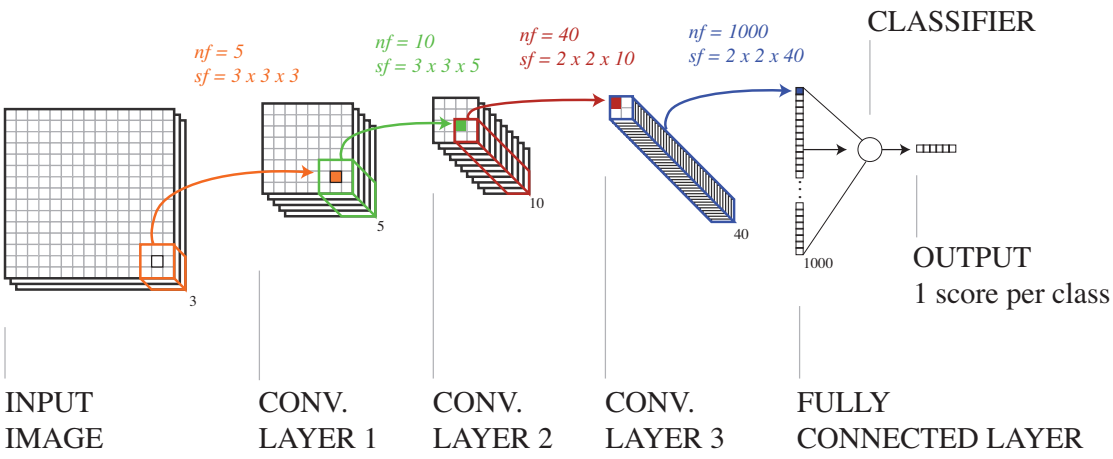


Figure 7: Flowchart of a CNN architecture with three convolutional layers (see Fig. 8 for details of a single layer) and one fully connected layer at the end. At each layer, each filter considers the whole d- dimensional image and computes convolutions of size $(m \times m \times d)$. This correspond to the colored parallelepiped at each step. The output is a nf-dimensional image, downsampled by a factor two (we assume a (2×2) max pooling at each layer). The information in the parallelepiped ends up being summarized in the pixel of the same color at the following step.

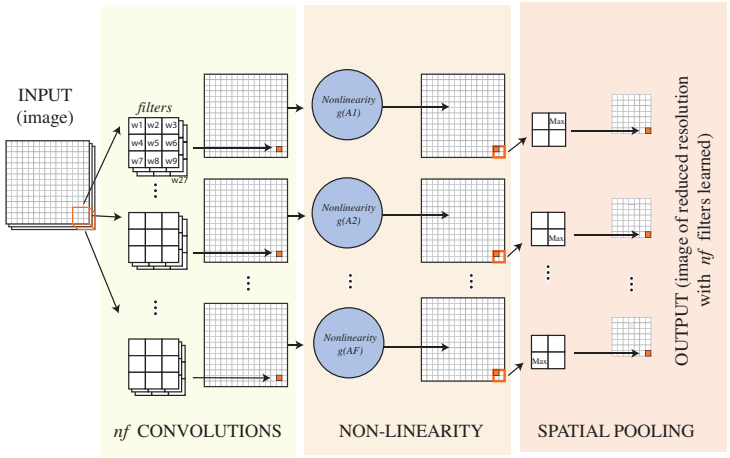


Figure 8: One typical architecture of one convolutional layer (a building block of Fig. 7), with a set of nf convolutions $(3 \times 3 \times d)$, a nonlinearity (e.g. a ReLU, Eq. (22)) and a spatial downsampling via a (2×2) max pooling filter (reduces the number of pixels by half). The output is an image with half the rows, half the columns as the original image, and as many layers as there were filters (in this case, nf).

local information with a statistic, typically the maximum (we then speak of *max pooling*) or the mean (*average pooling*). If the pooling is done with a (2×2) convolution, the spatial resolution with respect to the previous feature map is reduced by a factor two[2], where every pixel receives the maximal value observed in the (2×2) window.

A set of nf filters, with half spatial resolution as the input, are the output of the layer (and become the input of the next layer). So note, that the dimensionality d increases from layer to layer (in the first, it is the number of bands, but then becomes nf_l, the number of filters learned at each layer).

- *Fully connected layers*. Fully connected layer use all features in the input and squeeze them into a single value in the output. They are actually conceptually the same as a hidden layer of neurons in the MLP above.

[2]This is true only if the original convolution filter is applied with a half padding and a stride of 1, but we will not discuss these concepts in this review. To know everything about them, please read the very clear technical report of Dumoulin and Visin (2016).

The output vector will have as many dimensions as the number of neurons in the fully connected layer. In the example of Fig. 7, we learn 1000 such filters, leading to a 1000-dimensional input space for the prediction layer, discussed next.

- *Prediction layers*. This is generally the last layer of the network, which as in MLPs maps the high dimensional vector outputted by the last fully connected layer into a $ncl \times 1$ vector, with *ncl* being the number of classes to be predicted (or a single value in the case of regression). The loss function will then be assessed using this score as for MLP prior to backpropagation. This layer can be replaced by more fancy classifiers for prediction at test time (for examples using structured prediction classifiers as CRFs, see Section 5.

The good thing about CNNs, is that they can be trained by backpropagation as MLPs! By the chain rule in mathematics, CNN can be trained layer by layer and the loss backpropagated all the way to the inputs (as in MLPs). So, if you know how to train a MLP, you know how to train a CNN. It just has much more parameters to be learned, especially in the fully connected layers: taking the example of Fig. 7, each cell in the matrix within the blue box will have a corresponding weight for each filter learned. Since we are generally learning thousands of filters in the fully connected layer, this layer alone can involve millions of parameters to be estimated.

As mentioned above, CNNs have become the new gold standard in vision, and a lot of interest for these methods is being raised in remote sensing. This often raises the question: *are CNNs the silver bullet to solve any problem for remote sensing*? Our answer is a definite no: to train a CNN properly, a lot of training data are needed (to estimate millions of parameters, we'll need millions of training samples to avoid overfitting). If strategies to lessen the burden are applied (e.g. data augmentation (also known as jittering, Krizhevsky et al. (2012)), dropout (Srivastava et al., 2014), weight decay, unsupervised pre-training (Erhan et al., 2010)), CNNs still need a lot of labels to train, so they should be deployed in situations that are coherent with that requirement. For small and scarcely labeled datasets, there are very valid alternatives that are much more efficient (train fast, generalize well) as those reviewed in this chapter (kernels, ensemble or sparse methods). The bright side of this is that with CNNs it is now possible to tackle real big data problems and bring machine learning for remote sensing to a whole new scale, involving high resolution global problems.

3.2.2. CNN and remote sensing: a review

Research in remote sensing is starting to consider deep learning and to explore its potential for large scale labeling. Initially, research focused on the problem of image classification, where a single semantic label is attributed to an entire patch. Most papers start from pre-trained models, and then include a supervised fine-tuning stage (Penatti et al., 2015; Marmanis et al., 2015; Castelluccio et al., 2015). Such a fine tuning step seems to be necessary in order for general-purpose architectures from computer vision (e.g. VGG (Simonyan and Zisserman, 2015) or AlexNet (Krizhevsky et al., 2012)), which provide rich representations but specific to natural images, to adapt to remote sensing problems.

More recently, the appearence of publicly available new large scale datasets from data processing competitions (Rottensteiner et al., 2014; Campos-Taberner et al., in press) provided the datasets necessary to train deep CNN models to succeed in semantic labeling, i.e. the task of attributing a label to each pixel in the image. Romero and Camps-Valls (2016) proposed an unsupervised training strategy based on sparse coding to extract relevant features prior to classification. In Campos-Taberner et al. (in press) the results of the Data Fusion Contest 2015 are discussed: the winning team used that same model (Romero and Camps-Valls, 2016) to understand complimentarities between LiDAR and color data, while the runner-up team provides a detailed comparison of the use of pre-trained models in a land-use classification setting. In Paisitkriangkrai et al. (2015), authors propose a system based on i) a CNN model and ii) a random forest classifier trained using standard appearance descriptors (including the NDVI index and a histogram of normals of the DSM). The results of both are then fused and smoothed by the application of a conditional random field (See Section 5.2.2, page 26). In a subsequent paper, Sherrah (2016) trains two CNN models, one for color and one for height data, fuses them and smoothes the result with a CRF. These two models predict a single label per patch and slides through the whole the image at inference time. In Volpi and Tuia (in press), authors ease this point by using deconvolution layers (i.e. layers that increase the spatial support instead of decreasing it): as a result a scores vector is obtained for each pixel in the patch.

Other applications of interest in recent literature can be found in the estimation of sea ice concentration (Wang et al., 2016), change detection between videos and images (Vakalopoulou et al., 2016) or in the multimodal detection of urban trees using aerial and terrestrial images (Wegner et al., 2016).

4. Manifold learning methods

Typically, the feature space of remote sensing data is of very high dimensionality. This is particularly true for hyperspectral images with large numbers of spectral bands. Because of the high redundancy of the bands, the true dimensionality is however much lower. Moreover, due to the complex interactions of light with the earth surface and atmosphere, the relation between image bands is highly nonlinear. Hyperspectral pixels therefore live on a low-dimensional nonlinear submanifold of the spectral space. To illustrate this, Fig. 9 shows a 2-D scatterplot of a hyperspectral image of partly submerged grassland. Many analysis procedures rely on the use of a distance metric

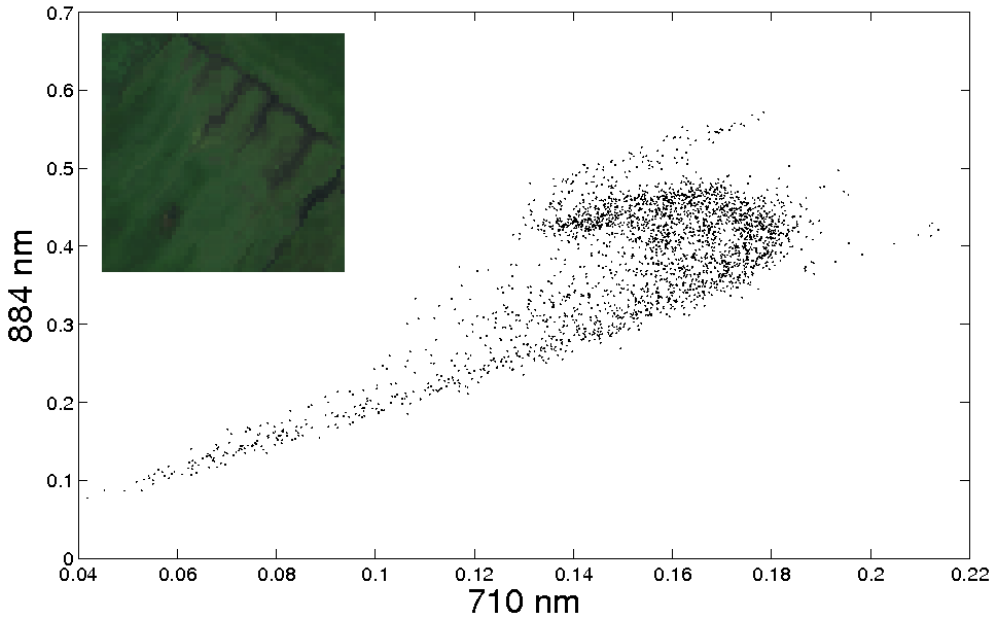


Figure 9: Scatter plot of 2 bands (710 nm and 884 nm) of partly submerged grassland. The data manifold has a highly nontrivial shape, indicating complex nonlinear interactions.

in the feature space. Metric-based classifiers in particular will work in a more optimal way, when using the metric of the submanifold (geodesic distance metric) rather than the Euclidean distance metric of the full feature space.

The term 'manifold learning' is generally reserved for nonlinear dimensionality reduction methods that project the data onto the submanifold. Manifold learning methods embed high-dimensional data samples into a lower dimensional domain by preserving the local structure between neighboring samples. We will concentrate in this section on supervised manifold learning methods. Supervised manifold learning methods for data classification map high-dimensional data samples to a lower dimensional domain in a structure-preserving way, while increasing the separation between different classes. Most supervised manifold learning methods compute the embedding only from the initially available labeled training data. However, the generalization of the embedding to novel points, i.e., the out-of-sample extension problem, becomes especially important in classification applications.

The manifold characterizes the underlying probability distribution of the data. A discrete representation of the manifold is often obtained by using graphs, where each data sample is represented by a node, and edges represent the local structure between neighboring samples. As an example, Fig. 10 shows a manifold-based representation of a spiral data set, a highly nonlinear 2-D manifold in a 3-D feature space. The discrete representation of this manifold is given by a k-nearest-neighbor (kNN) graph on the data. To generate such a graph, the Euclidean distance between any two points is calculated and every point is connected to its K nearest neighboring points. The weight of every edge is the corresponding Euclidean distance. The graph needs to be symmetrized and connected. The graph geodesics are then determined as the shortest paths along the weighted graph between these two points, calculated e.g. by the Dijkstra algorithm (Dijkstra, 1959). The lengths of the graph geodesics will approximate the true geodesic distances as measured along the surface of the data manifold.

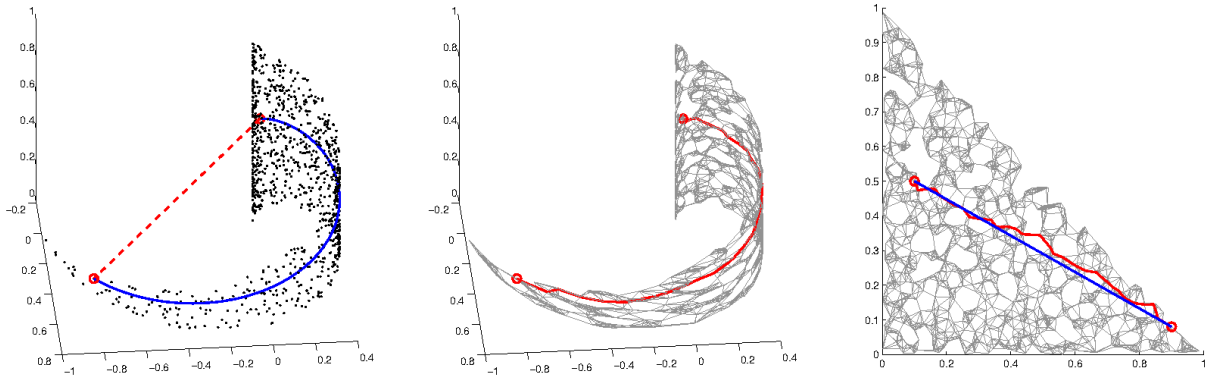


Figure 10: Graph-based manifold representation of a spiral; a: data cloud (dots), and true geodesic distance (full line) versus Euclidean distance (dashed line) between 2 data points; b: kNN graph (gray lines) and graph geodesic between two points; c: unfolded manifold and projected graph geodesic versus true geodesic.

A graph can also be represented in a matrix form. In the case of the kNN graph, this is the distance matrix containing the Euclidean distances between all samples connected to each other, and 0 everywhere else. This matrix can also be an adjacency matrix, containing 1 between all samples that are connected with an edge and 0 elsewhere. Spectral graph theory is a well-established algebraic theory (von Luxburg, 2007) and studies the properties of graphs via the eigenvalues and eigenvectors of their associated graph matrices. In the next section, we explain the general framework of spectral graph theory for manifold learning.

4.1. Spectral graph theory for manifold learning

Assume a D-dimensional feature space and N training samples $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]$. The embedding of $\mathbf{X}$ is given by $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_N]$, a $d \times N$-dimensional matrix with $d < D$. The goal is to find the optimal embedding by preserving the local structure of the data. This is generally formulated as an optimization problem:

$$\min_{\mathbf{Z}} \sum_{ij} \mathbf{M}_{ij} \mathbf{z}_i \mathbf{z}_j = \min_{\mathbf{Z}} \mathrm{Tr}(\mathbf{Z}\mathbf{M}\mathbf{Z}^T), \tag{27}$$

where $\mathbf{M}$ is the $N \times N$ graph matrix. This problem can be solved by solving the eigenvalue problem

$$\mathbf{M}\mathbf{z} = \lambda \mathbf{z} \tag{28}$$

and the embedding is then given by the eigenvectors corresponding to the smallest d eigenvalues (in fact, the eigenvector corresponding to the smallest eigenvalue is the unit vector and is discarded).

4.1.1. Local Linear Embedding (LLE)

As an example, consider the embedding algorithm Local Linear Embedding (Roweis and Saul, 2000). The basic idea of this algorithm is to describe the local structure of a data sample by representing it as a linear combination of its neighbors. Assume that $\mathbf{x}_{ij}$ are the neighbors of $\mathbf{x}_i$. Then, the optimal representation $\mathbf{W}$ is found by minimizing the reconstruction error:

$$\min_{\mathbf{W}} \sum_{ij} \|\mathbf{x}_i - \sum_j \mathbf{W}_{ij} \mathbf{x}_{ij}\|^2 \tag{29}$$

To preserve the the neighborhood relations in the low dimensional space, the obtained coefficient matrix $\mathbf{W}$ is kept during the embedding:

$$\min_{\mathbf{Z}} \sum_{ij} \|\mathbf{z}_i - \sum_j \mathbf{W}_{ij} \mathbf{z}_{ij}\|^2 \tag{30}$$

This optimization corresponds to Eq. (27) with $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})$.

4.1.2. Laplacian Eigenmaps (LE)

Another example of a local embedding is the Laplacian Eigenmaps method (Belkin and Niyogi, 2001). Here, the local structure is built by the pairwise distances between neighboring samples, eventually weighted by a Gaussian kernel. If the adjacency matrix $\mathbf{W}$ is constructed by appropriate neighborhood relationships, the optimal embedding minimizing the distances between each sample and its neighbors is obtained by:

$$\min_{\mathbf{Z}} \sum_{ij} (\mathbf{z}_i - \mathbf{z}_{ij})^2 \mathbf{W}_{ij} = \min_{\mathbf{Z}} \mathbf{Z}^T \mathbf{L} \mathbf{Z} \tag{31}$$

with $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian matrix and $\mathbf{D}$ is the diagonal degree matrix: $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. This optimization corresponds to Eq. (27) with $\mathbf{M} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, the normalized graph Laplacian matrix.

Other examples of local manifold learning methods that can be described using the spectral graph theoretic framework as well are:

- Isomap (Tenenbaum et al., 2000). This is an isometric embedding, which preserves the pairwise distances between neighboring samples. The pairwise geodesic distances in the submanifold are approximated by the shortest paths in the kNN graph.
- Local Tangent Space Alignment (LTSA) (Zhang et al., 2007) describes the local geometry by the local tangent space of each data sample and finds an embedding that aligns the tangent hyperplanes of the manifold.

4.1.3. Supervised manifold learning

Up till now, all the described manifold learning algorithms were unsupervised methods. One particular way of including class label information is to define the local structure of a sample by considering only neighboring samples of the same class. In this way, in Li et al. (2009), a supervised version of LLE is proposed, based on Linear Discriminant Analysis in the embedded submanifold. A supervised version of LE is proposed in Raducanu and Dornaika (2012). The graph Laplacian is split into a within and a between-class graph. Then, the local margin between samples of different classes is maximized simultaneously with the minimization of distances between samples of the same class.

Since manifold learning methods only map labeled samples, an important aspect of a supervised manifold learning is the generalization to new data samples. A kernel view on this aspect regards manifold learning as the learning of the eigenfunctions of a data-dependent kernel (Bengio et al., 2004). The embedding is defined by the eigenvectors of the kernel matrix $\mathbf{F} = \mathbf{I} - \mathbf{M}$ corresponding to the largest eigenvalues (summarized in the diagonal $d \times d$ matrix $\mathbf{\Lambda}$). $\mathbf{F}$ is obtained by a kernel function computed over pairs of training samples:

$$f(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{F}_{ij} = \delta_{ij} - \mathbf{M}_{ij} \tag{32}$$

where δ_{ij} is the Kronecker delta. The embedding of a new data sample $\mathbf{x}_0$ can then be obtained by using the Nyström formula:

$$\mathbf{z}_0^T = \sum_{i=1}^{N} f(\mathbf{x}_0, \mathbf{x}_i) \mathbf{z}_i^T \mathbf{\Lambda}^{-1} \tag{33}$$

In (Bengio et al., 2004), specific kernel functions are defined for the different manifold learning algorithms LLE, LE, Isomap and Local Tangent Space Alignment.

4.1.4. Manifold regularization

An interesting strategy to combine manifold learning with supervised classification is *manifold regularization* (Belkin et al., 2006), in which the optimization term in Eq. (27) is applied as a regularizer in a supervised classifier, hereby exploiting the local manifold geometry of the data:

$$\min_{\mathbf{Z}} V(\mathbf{z}_i, y_i) + \lambda \mathbf{Z}^T \mathbf{L} \mathbf{Z} \tag{34}$$

where V is the loss function corresponding to the classifier and y_i is the class label of sample $\mathbf{z}_i$. The adjacency matrix is generally composed of labeled as well as unlabeled samples, resulting in a semi-supervised approach.

4.1.5. Manifold alignment

Another concept is *manifold alignment*, in which the goal is to find a common manifold representation for two data sets, which may be dissimilar but are generated by similar processes. By learning projections from each of the original domains of the datasets to the shared manifold, knowledge can be transferred from one domain to the other. The adjacency matrix of Eq. (31) can be extended to include both domains by:

$$G = \begin{pmatrix} \lambda S_1 & (1-\lambda)W \\ (1-\lambda)W^T & \lambda S_2 \end{pmatrix} \tag{35}$$

where S_1 and S_2 are the adjacency matrices of the two domains and W represents the binary correspondence matrix between data points of both domains, i.e. $W_{ij} = 1$ if there is a correspondence between data point i from the first domain and data point j from the second domain, and 0 otherwise. λ weights the relative importance of the correspondence. As previously, the obtained optimization problem is solved by a generalized eigenvalue problem using the graph laplacian of G. A supervised version requires full knowledge about the correspondence matrix. Unsupervised (Wang and Mahadevan, 2005) and semisupervised (Ham et al., 2005b) manifold alignment methods were developed as well.

4.2. Manifold learning for remote sensing

The manifold learning methodology has been applied to the analysis of hyperspectral images (Bachmann et al., 2005, 2009), and in particular for hyperspectral unmixing (Heylen et al., 2011). A recent overview paper (Lunga et al., 2014) describes the state of the art of manifold learning for the purpose of feature extraction. In this section however, we will concentrate on the application of manifold learning for HSI classification.

In Ma et al. (2010b), the kernel view of supervised manifold learning has been applied to HSI classification using supervised LTSA. In Ma et al. (2010a), a combination of local manifold learning and kNN classification was proposed for HSI classification. In that paper, two strategies are proposed: in the first, kNN is applied using the new distance metric obtained by any unsupervised manifold learning dimensionality reduction method. In the second strategy, supervised manifold learning is applied, where d is chosen as the number of classes, and all training samples from the same class are mapped onto a single point in the embedded manifold. In Chapel et al. (2014), Laplacian eigenmaps are applied for HSI classification in a supervised manifold learning framework. A local submanifold is obtained from the training samples for each class separately. Then, a similarity measure between new test samples and the classes is defined by characterizing the perturbation that a submanifold undergoes when including the test sample. The test sample is classified into the class that gives the smallest perturbation. In Ma et al. (2015), a graph-based semisupervised learning approach is proposed that makes use of manifold learning for the graph construction. LLE as well as LTSA were applied.

A particularly interesting approach is the combination of sparse and manifold-based representations (for more information on sparse learning methods, see Section 7). In Tang et al. (2014), manifold-based regularization terms are incorporated in a sparsity-based objective fuction to enforce smoothness along sparse representations of local neighbors. Rather than sparsely representing test samples, in Ni and Ma (2015), the tangent plane is used to sparsely represent the local manifold of each test sample. In Huang and Yang (2015), sparse discriminant embedding is proposed as a supervised dimensionality reduction method for hyperspectral image classification. The method combines sparse regularization with graph-based discriminant analysis, optimally utilizing the manifold data structure.

The high within-class spectral variability of HSI makes that training samples at one location cannot be applied at other locations. (Semi-)supervised domain adaptation aims to adapt a classifier that is trained in one domain, to another domain, by adapting the manifold using unlabeled samples from that domain. Manifold regularization is an interesting technique to apply domain adaptation (Kim and Crawford, 2010). When confronted to many spatially disconnected domains, possibly also acquired by different sensors, manifold alignment techniques have been proposed (Tuia et al., 2014b). An extension of that method that takes into account locality preservation has been proposed (Yang and Crawford, 2016a): a global alignment is accomplished by bridging pair relationships, while locality preservation is induced by incorporating similar local clusters in the alignment process. Based on this work, in Yang and Crawford (2016b), 2 local manifold alignment methods were proposed for classification of multitemporal hyperspectral images in a common manifold space. In Tuia and Camps-Valls (2016), a kernelized version of the semisupervised manifold alignment is proposed.

In Zhang et al. (2014), the concept of supervised manifold learning is adapted to the problem of hyperspectral target detection. Hereby, the manifold dimensionality reduction is driven by minimizing the distance between target-target sample pairs and maximizing the distance between target-background sample pairs. Moreover, a sparse formulation is proposed to restrict the number of nonzero elements in the adjacency matrix, to overcome overfitting due to small training sample sizes that are typical in target detection. Mishne et al. (2015) describes a target detection method using a graph-based formulation of the manifold learning paradigm, hereby accounting for the local variability of target training samples.

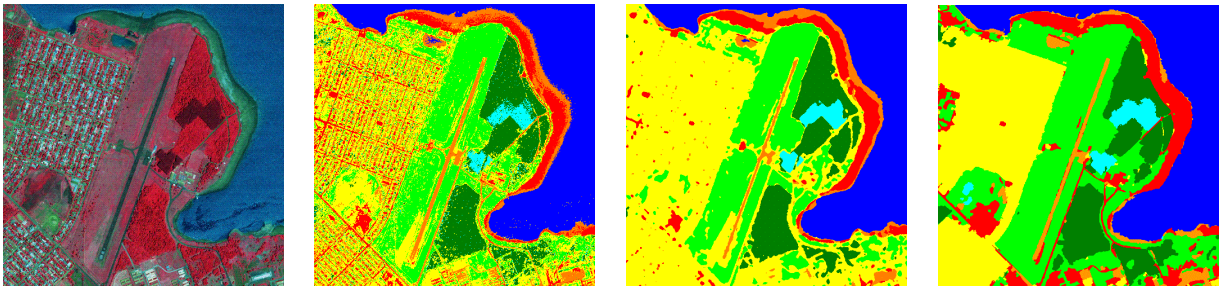
The concept of manifold learning is not only applicable to hyperspectral images but can be applied to other imaging modalities as well. In Huang et al. (2014), a hierarchical version of the supervised manifold learning approach is proposed, and applied to object classification of VHR images. A classification-oriented hierarchical manifold is obtained by first constructing submanifolds of lower level subclasses, after which parent manifolds are generated by sharing features that represent major classes. In Liu et al. (2016a), a graph-based formulation of the semisupervised manifold learning classification method is proposed for classification of Polarimetric SAR data. Target detection in SAR data is the topic of research in Huang et al. (2016), where a tensor extension of manifold representations is proposed. In Ebtehaj et al. (2015), a LLE approach is proposed for the retrieval of rainfall from passive microwave data. Finally, in Zhang et al. (2016b), a manifold regression method, based on LE is proposed for the analysis of chemical contamination from chemical spill data.

5. Structured output learning methods

Especially when thematic products should be mapped at high or very high spatial resolutions, the spatial information associated with the input image(s) and the expected spatial structure of the output results are of major importance (Nowozin and Lampert, 2010; Schindler, 2012). However, in the case of many machine learning and pattern recognition techniques (e.g., the aforementioned SVM and neural networks), basic formulations were developed and theorems were proved under the assumption of independent and identically distributed (i.i.d.) samples (Bishop, 2006; Vapnik, 1998). With regard to the application to image data, this assumption implies neglecting the (usually very high) correlations among neighboring pixels, the possible presence of quasi-periodic patterns and textures, the geometrical arrangement of regions and edges, and the general non-stationarity of the spatial behavior of the pixel intensities. If satellite image time series (SITS) are considered, neglecting dependencies also implies assuming that multiple images taken at different times over the same geographical area are independent. In the language of signal processing, this approach basically means applying a white stationary model to data drawn from well-correlated and mostly non-stationary sources. On one hand, the use of i.i.d. models bears the advantage of simplicity and has been found accurate and time-efficient in many learning problems in remote sensing, especially at moderate spatial resolutions (Landgrebe, 2003). On the other hand, learning and modeling methods have also been proposed to characterize and favor the expected dependencies among the samples in the output mapping result, according to the spatial, geometrical, or spatio-temporal information in the input imagery.

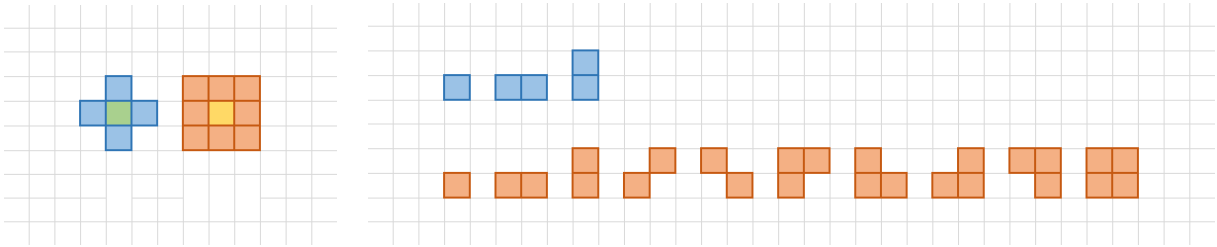
From a machine learning perspective, these methods fall under the broad family of *structured output learning* or *structured prediction*. This family includes techniques aimed at predicting an output characterized by some dependency structure instead of individual i.i.d. samples (Nowozin and Lampert, 2010). In the typical application to remote sensing image classification, explicit phrases such as *spatial-contextual* or *spectral-spatial* classification are often used when the desired output structure is associated with the spatial information in the imaged scene (Fauvel et al., 2013; Moser et al., 2013). Examples of the impact of these approaches are shown in Fig. 11, which regards land cover mapping from an IKONOS multispectral image at 4-m resolution (Fig. 11(a)). An SVM obtains the map in Fig. 11(b), which captures fairly well the discrimination among the classes but cannot detect the – spatially very heterogeneous – urban area as a whole on a pixelwise basis. Figs. 11(c) and (d) show results of structured prediction methods that incorporate spatial dependency models and substantially improve the spatial regularity of the land cover map and the discrimination of the urban class.

A major approach to address structured prediction is by using *probabilistic graphical models*, which encode the desired dependency properties through suitable graphs, usually associated with Markovian properties of various kinds (Koller and Friedman, 2009). The next subsections will summarize the key ideas and the current trends about these and related approaches. Here, we only recall that, in the case of image classification, an alternative to



(a) (b) (c) (d)

Figure 11: Example of result of structured output learning: (a) input VHR optical image (IKONOS, 4-m resolution, false-color display); (b) non-contextual SVM classification; (c) contextual classification based on the combination of an MRF model and an SVM (Moser and Serpico, 2013); (d) contextual classification based on a multiscale region-based MRF model with adaptive textures (Moser et al., 2013). Color legend: urban, herbaceous, shrubs and bushes, forest, bare soil, built-up area, and water.



(a) (b)

Figure 12: Examples of neighborhoods and cliques on the pixel lattice: (a) the blue pixels make the first-order (or 4-connected) neighborhood of the central green pixel, and the orange pixels make the second-order (or 8-connected) neighborhood of the central yellow pixel; (b) blue and orange pixels indicate cliques in the first- and second-order neighborhoods, respectively.

spatial-contextual classifiers is to apply purely pixelwise classifiers to sets of features that are related to spatial information (Landgrebe, 2003). In the usual pattern-recognition pipeline (Bishop, 2006), this approach conceptually moves the effort of capturing spatial dependencies from the classification stage to the feature extraction stage. Texture analysis (Maillard, 2003) and mathematical morphology (Ghamisi et al., 2015; Fauvel et al., 2013) provide well-known families of techniques for the latter stage. Although interpixel correlations in the extracted spatially-sensitive features are again very high, the use of these techniques together with pixelwise classifiers has been found powerful in the application to many remote sensing data modalities, including hyperspectral and multispectral VHR imagery. More details can be found in the review papers (Ghamisi et al., 2015) and (Fauvel et al., 2013).

5.1. The basics of Markov random fields

A major category of probabilistic graphical models for spatial-contextual information is represented by *Markov random field (MRF)* models (Kato and Zerubia, 2011; Wang et al., 2013). They are a family of 2D stochastic processes that generalize to data arranged on a 2D pixel lattice the usual notion of Markov chain, which have been popular for decades in the modeling of 1D sequences of data (e.g., for speech recognition) and events (e.g., in queueing theory) (Derin and Kelly, 1989; Rabiner, 1989).

5.1.1. Key ideas, definitions, and main properties

Let $\mathcal{I}$ be a 2D regular pixel lattice and let a *neighborhood system* be defined on $\mathcal{I}$. This means that a relationship "$\sim$" is defined on pairs of pixels such that $i \sim j$ indicates that pixels i and j are neighbors ($i, j \in \mathcal{I}$). Typical examples are the first-order (or 4-connected) and the second-order (or 8-connected) neighborhood systems, in which the neighbors of a pixel are its four adjacent pixels and its eight surrounding pixels, respectively (see Fig. 12(a)). In general, any neighborhood system is supposed to satisfy two properties:

$$i \sim j \iff j \sim i, \qquad i \nsim i \qquad \forall i, j \in \mathcal{I}, \tag{36}$$

i.e., being neighbors is a reflexive relationship and no pixel is neighbor to itself. From a graph-theoretic viewpoint, this is equivalent to introducing a graph in which nodes coincide with pixels and there is an edge between two nodes if and only if the corresponding pixels are neighbors (Wang et al., 2013). Given the neighborhood system, a *clique* $\mathcal{C}$ is a subset of pixels ($\mathcal{C} \subset \mathcal{I}$) that are mutually neighbors, i.e.:

$$i, j \in \mathcal{C} \Longrightarrow i \sim j. \tag{37}$$

The cliques corresponding to first- and second-order neighborhoods are shown in Fig. 12(b).

Let us focus here on the classification of an image defined on the lattice $\mathcal{I}$; extension to multitemporal classification of SITS, to regression, or to other learning tasks will be mentioned later in this section. Let $\mathbf{x}_i$ and y_i be the feature vector and the class label of pixel $i \in \mathcal{I}$, respectively, and let $\mathcal{X} = \{\mathbf{x}_i\}_{i\in\mathcal{I}}$ and $\mathcal{Y} = \{y_i\}_{i\in\mathcal{I}}$ collect the feature vectors and labels of all image pixels, respectively. The approach taken by Markovian structured prediction is probabilistic, i.e., $\mathcal{X}$ and $\mathcal{Y}$ are assumed to be (realizations of) stochastic processes supported on the pixel lattice $\mathcal{I}$, and a Bayesian approach to learning is used. In the case of a pixelwise classifier for i.i.d. data $(\mathbf{x}_i, y_i)$ ($i \in \mathcal{I}$), the Bayesian maximum *a-posteriori* (MAP) rule separately maximizes, with respect to y_i and on each pixel $i \in \mathcal{I}$, the pixelwise posterior probability distribution $P(y_i|\mathbf{x}_i)$ of the label y_i given the feature vector $\mathbf{x}_i$ (Bishop, 2006). In the case of non-i.i.d. data, the MAP rule should in principle be formulated as the maximization of the joint posterior distribution $P(\mathcal{Y}|\mathcal{X})$ of all labels given all feature vectors in the image, with respect to the whole label configuration $\mathcal{Y}$. This is a computationally intractable task, in general. However, it becomes tractable under suitable restrictions on the dependence structure of the considered random fields: it is here that MRFs play a crucial role with respect to classification.

Specifically, $\mathcal{Y}$ is an MRF if the following properties are satisfied (Kato and Zerubia, 2011):

- *Markovianity:* the probability distribution of the label of each pixel, when conditioned to the labels of all other pixels, can be restricted to the distribution conditioned only to the labels of the neighbors:

$$P(y_i|y_j, j \neq i) = P(y_i|y_j, j \sim i) \qquad \forall i \in \mathcal{I}. \tag{38}$$

- *Positivity:* the joint distribution of all labels is strictly positive, i.e., $P(\mathcal{Y}) > 0$ for all label configurations $\mathcal{Y}$.

The Markovianity property is basically the 2D counterpart of the usual finite-memory property of 1D Markov chains (Derin and Kelly, 1989). It focuses the modeling of spatial interactions on the basis of the neighborhood system. The positivity property means that no output classification map is *a priori* forbidden. It is postulated mostly as a technical assumption for the theorems and properties mentioned below to hold. It is not restrictive, though.

In itself, the definition of an MRF regards local (i.e., neighborhood-based) properties of the random field $\mathcal{Y}$. However, a major result of the theory, namely the Hammersley-Clifford theorem, also provides a well-defined and efficient global formulation. This theorem says that $\mathcal{Y}$ is an MRF if and only if its joint distribution can be expressed as:

$$P(\mathcal{Y}) = \frac{e^{-U(\mathcal{Y})}}{Z}, \tag{39}$$

where Z, named *partition function*, is a normalization constant and the function $U(\cdot)$, named *(prior) energy*, can be expanded as a sum of local contributions:

$$U(\mathcal{Y}) = \sum_{\mathcal{C}} V_{\mathcal{C}}(\mathcal{Y}_{\mathcal{C}}), \tag{40}$$

in which the sum is taken over all cliques in the considered neighborhood system and the contribution $V_{\mathcal{C}}(\cdot)$ associated with clique $\mathcal{C}$, named *potential*, only depends on the subset $\mathcal{Y}_{\mathcal{C}} = \{y_i\}_{i\in\mathcal{C}}$ of the labels of the pixels in $\mathcal{C}$ (Geman and Geman, 1984). This theorem provides an explicit representation of the global imagewise distribution of the random field (see (39)), a representation that is, in turn, related to the local properties associated with neighborhoods and cliques and quantified by the potentials (see (40). It is this capability to jointly provide a global model for the dependency structure of the desired learning output in terms of a computationally efficient local formalization that makes MRF-based approaches especially useful.

Let us also assume that the feature vectors are mutually independent when conditioned to the labels (*conditional independence assumption*), i.e., the global class-conditional distribution $p(\mathcal{X}|\mathcal{Y})$ (named *likelihood*) can be factored out as follows (Dubes and Jain, 1989):

$$p(\mathcal{X}|\mathcal{Y}) = \prod_{i\in\mathcal{I}} p(\mathbf{x}_i|y_i), \tag{41}$$

where $p(\mathbf{x}_i|y_i)$ is the pixelwise class-conditional statistics of the feature vectors. If this assumption and an MRF model for $\mathcal{Y}$ hold, then the global posterior distribution can be written as:

$$P(\mathcal{Y}|\mathcal{X}) \propto e^{-U(\mathcal{Y}|\mathcal{X})}, \qquad \text{where :} \qquad U(\mathcal{Y}|\mathcal{X}) = -\sum_{i\in\mathcal{I}} \ln p(\mathbf{x}_i|y_i) + \sum_{\mathcal{C}} V_{\mathcal{C}}(\mathcal{Y}_{\mathcal{C}}). \tag{42}$$

For example, if only pairwise clique potentials are nonzero, the global posterior distribution is determined by:

$$U(\mathcal{Y}|\mathcal{X}) = -\sum_{i\in\mathcal{I}} \ln p(\mathbf{x}_i|y_i) + \sum_{i\sim j} V_{ij}(y_i, y_j). \tag{43}$$

Therefore, the output field $\mathcal{Y}$ turns out to be an MRF even when conditioned to the input field $\mathcal{X}$ (Dubes and Jain, 1989). The phrase "*hidden MRF (HMRF)*" is sometimes used to stress that an MRF model is assumed for the hidden (or latent) field $\mathcal{Y}$ to be estimated, given the observed field $\mathcal{X}$. The distribution of $\mathcal{Y}$ given $\mathcal{X}$ is determined by the *posterior energy* $U(\mathcal{Y}|\mathcal{X})$, which includes both pixelwise terms related to class-conditional statistics and contextual terms related to the clique potentials.

In particular, according to (42), Bayesian MAP classification is equivalent to a *minimum energy* problem, i.e., to minimizing $U(\mathcal{Y}|\mathcal{X})$ with respect to $\mathcal{Y}$. Many techniques have been proposed for this task. Among these, we recall the following major approaches:

- *Simulated annealing* is a stochastic minimization algorithm that involves sampling multiple realizations of $\mathcal{Y}$ through Gibbs or Metropolis sampling. It converges to the global minimum under certain assumptions although with usually long computation times (Geman and Geman, 1984).

- *Iterated conditional mode (ICM)* is a deterministic technique that is initialized with a preliminary label configuration and converges to a local minimum (Besag, 1993). It is not computationally intensive but the convergence solution is generally suboptimal and possibly affected by the initialization.

- *Graph cuts* methods are based on a reformulation of the minimum-energy problem as a min-flow/max-cut problem on the corresponding graph. They are proven to reach a global minimum in the case of binary classification (Greig et al., 1989) and a local minimum endowed with strong optimality properties in the multiclass case (Szeliski et al., 2008). Thanks to these properties and to their remarkable computational efficiency, graph cuts have been growing more and more prominent in the last decade. The well-known swap-move and α-expansion algorithms belong to the graph cuts family (Boykov et al., 2001).

- *Belief propagation* methods are based on the idea of passing "messages" along the edges of the graph to progressively decrease the energy. They include the well-known loopy belief propagation (LBP) (Ihler et al., 2005) and tree re-weighted message passing (TRW) (Kolmogorov, 2006) algorithms. Convergence properties vary on individual algorithms and energy functions.

As an alternative to MAP, the *marginal posterior mode (MPM)* criterion, which maximizes, with respect to y_i on each pixel $i \in \mathcal{I}$, the marginal posterior distribution $P(y_i|\mathcal{X})$ of the label of each pixel given all feature vectors, is also sometimes used, given an MRF model (Kato and Zerubia, 2011). While it generally requires time-consuming random sampling in the case of conventional non-hierarchical MRFs (Dubes and Jain, 1989), it proves advantageous in the case of hierarchical MRFs (see below Section 5.2.3).

5.1.2. Modeling through MRFs

From a modeling perspective, we stress first that the conditional independence assumption (41) is customary in MRF-based classification and is accepted to favor analytical tractability (Dubes and Jain, 1989). It does not imply that the feature vectors are independent in themselves, and it splits the modeling task into: (i) encoding the desired dependency structure and constraints through the contextual prior of the output field $\mathcal{Y}$, expressed in terms of clique potentials; and (ii) modeling the likelihood of the input field $\mathcal{X}$ given $\mathcal{Y}$ on a pixelwise basis.

The possibility to define energy terms and potentials quite freely makes this modeling framework flexible and powerful. Well-known examples include:

- *Spatial smoothing models* – A customary example is the isotropic Potts (or multilevel logistic, MLL) model, in which pairwise cliques are involved and the following potential is used:

$$V_{ij}(y_i, y_j) = \beta\Big[1 - \delta(y_i, y_j)\Big], \tag{44}$$

where $\delta(\cdot)$ is the Kronecker symbol[3] and β is a positive parameter (Kato and Zerubia, 2011). In the minimization of the posterior energy, this potential penalizes class transitions between neighboring pixels, i.e., it favors equal labeling inside homogeneous regions. The spatial smoothing effect intensifies as β increases. Anisotropy can also be incorporated through different penalties across distinct spatial directions (Dubes and Jain, 1989).

- *Edge-preserving smoothing models* – Especially when β is high, the Potts model may yield to oversmoothing borders between homogeneous regions and small spatial details. The same comment holds for analogous spatially regularizing models. To overcome this drawback, edge-preserving MRFs encode edge information in the potentials. The idea is not to penalize class transitions between neighboring pixels when an edge passes (or is likely to pass) between them. This idea has been formalized using line processes (Geman and Geman, 1984), interaction functions (Li, 1995), total variation functionals (Rudin et al., 1992), and adaptive neighborhoods (Smits and Dellepiane, 1999).

- *Data fusion models* – When multiple input information sources are available (e.g., both an optical and a SAR image of the same geographical area) and/or multiple typologies of context can be defined (e.g., both a spatial and a temporal context in SITS analysis), composite energy functions can be defined (Schistad Solberg et al., 1996; Melgani and Serpico, 2003). For example, in the case of nonzero potentials only for pairwise cliques:

$$U(\mathcal{Y}|\mathcal{X}) = -\sum_{s=1}^{S}\sum_{i\in\mathcal{I}} \alpha^s \ln p^s(\mathbf{x}_i^s|y_i) + \sum_{t=1}^{T}\beta^t \sum_{i\overset{t}{\sim}j} V_{ij}^t(y_i, y_j), \tag{45}$$

where S and T are the numbers of input sources and of types of context, respectively, $\mathbf{x}_i^s$ is the feature vector of pixel $i \in \mathcal{I}$ in the sth source with its associated class-conditional statistics $p^s(\cdot)$, "$\overset{t}{\sim}$" is the neighborhood relationship corresponding to the tth type of context with its associated pairwise potential $V_{ij}^t(\cdot)$, α^s and β^t are positive weight parameters ($s = 1, 2, \ldots, S; t = 1, 2, \ldots, T$). Besides characterizing the desired output structure, this formulation also makes it possible to use MRF models as *data fusion* tools (Gomez-Chova et al., 2015).

Further examples of models that incorporate region or texture structures in image classification or generalize to regression through continuous-valued MRFs will be mentioned in Section 5.2.

The pixelwise energy contributions in (42), (43), or (45) can be determined by applying any parametric or nonparametric probability density estimator (Bishop, 2006). Alternately, they can be easily reformulated, up to additive terms that do not depend on $\mathcal{Y}$, in terms of the pixelwise posterior probabilities $P(y_i|\mathbf{x}_i)$ and prior probabilities $P(y_i)$. The latter can be estimated using training samples (Landgrebe, 2003). The former can be derived using any noncontextual classifier that provides posterior probability estimates, including Bayesian decision theory, neural networks (see Section 3), multinomial logistic regression (Bohning, 1992), and random forest (see Section 6). An SVM does not provide a probabilistic output *per se*, however approximate posteriors can be derived by postprocessing the SVM output through the algorithms in (Wu et al., 2004).

Markovian energies usually include parameters (e.g., the α's and β's in (44) and (45), which generally have to be optimized before or jointly with energy minimization. The resulting *model selection* task is not straightforward because maximum likelihood estimation is unfeasible apart from very specific MRF models (Ibanez and Simo, 2003). Approaches based on pseudo-likelihood functions (Besag, 1993), Monte Carlo or mean-field approximations (Ibanez and Simo, 2003), expectation-maximization (Moser and Serpico, 2009), or mean square error (De Giorgi et al., 2015; Serpico and Moser, 2006) have been proposed. More details on this model selection problem can be found in (Kato and Zerubia, 2011) and (Gimenez et al., 2015).

[3] This means that $\delta(a, b) = 1$ for $a = b$ and $\delta(a, b) = 0$ otherwise.

5.2. *Current trends of structured prediction in remote sensing*

In this section, we summarize the recent and current trends of structured output learning in remote sensing. In most of the cases, these methods are aimed at addressing image classification tasks with various input data modalities. However, applications to multitemporal analysis and regression can be remarked as well.

5.2.1. *Markov random fields in remote sensing*

MRF models have been popular for a long time in remote sensing for classification (Schistad Solberg et al., 1996; Jackson and Landgrebe, 2002), change detection (Moser and Serpico, 2009), multitemporal analysis (Melgani and Serpico, 2003), texture extraction (Rallier et al., 2004), and image restoration (Denis et al., 2009). The interest towards them has further intensified in the past few years, especially because of the ever growing availability of VHR data and the correspondingly increasing need for structured prediction methods. For example, in (Sun et al., 2015a) and (Khodadadzadeh et al., 2014), MRFs for spectral-spatial classification of VHR hyperspectral images are introduced by combining multinomial logistic regression with total variation functionals and mixed pixel modeling, respectively. In (Masjedi et al., 2016), an MRF is proposed to combine composite kernels and Wishart distributions and take benefit from both parametric and nonparametric approaches to polarimetric SAR data classification. In (Sun et al., 2015b), MRFs are used to incorporate contextual information into active learning by measuring uncertainty as a function of the discrepancy between pixelwise and MRF-based labeling. In (Yousif and Ban, 2014), nonlocal probabilistic modeling is incorporated into a Markovian change detection scheme to reduce oversmoothing. In (Bhatt et al., 2014), a continuous-valued MRF is developed to incorporate spatial information into Bayesian unmixing of hyperspectral data. In (Qin, 2014), an MRF is introduced to address the detection of changes in the faces of 3D buildings using stereo pairs of VHR images. In (Vakalopoulou et al., 2016) and (Karantzalos et al., 2014), Markovian energies are defined to address registration of multitemporal images.

Recent works have also made use of the data fusion capabilities of the MRF framework. Formulations as in (45)) are used, for example, to fuse the information conveyed by different feature sets in VHR image classification (Lu et al., 2016); to address change detection from multisensor VHR optical-SAR data (Gomez-Chova et al., 2015); to favor temporal consistency between land-cover maps on different years (Wang et al., 2015a); and to restore missing data in a partly cloud-covered image by using similarity with respect to a reference cloud-free image (Cheng et al., 2014). In (Gerke and Xiao, 2014), a decision tree is used to fuse optical and LiDAR data from an airborne platform and to determine the pixelwise energy term of an MRF. Multilayer MRF models (namely, the multicue MRF, the conditional multilayer mixed MRF, and the fusion MRF), which are aimed at fusing multiple features or multitemporal data, are discussed in (Sziranyi and Shadaydeh, 2014) and (Benedek et al., 2015).

Finally, substantial attention has been given to the combination of kernel and Markovian learning, with the purpose of taking benefit of the flexibility and robustness to dimensionality issues of the former and of the structured prediction capability of the latter. A basic way is to derive probabilistic pixelwise terms from the SVM output through the algorithm in (Wu et al., 2004) (see Section 5.1.2). The integration of the MRF and SVM approaches is also addressed in (Moser and Serpico, 2013) by proving that the application of the Markovian minimum-energy rule in the Hilbert space associated with a kernel is equivalent to the use of an appropriate composite Markovian kernel. The method is extended in (Moser and Serpico, 2014; De Giorgi et al., 2014) and (Wehmann and Liu, 2015) by integrating a dedicated kernel for complex-valued polarimetric SAR data, a graph cuts reformulation, and a multitemporal generalization, respectively.

5.2.2. *Conditional random fields and other graphical models*

Conditional random field (CRF) models are a generalization of MRFs and have lately been receiving increasing attention in remote sensing. In the case of a CRF, the Markovianity and positivity properties are assumed to hold for the conditional distribution of $\mathcal{Y}$ given $\mathcal{X}$ directly (Sutton and McCallum, 2011). This means that an MRF-like formulation is introduced for the posterior distribution $P(\mathcal{Y}|\mathcal{X})$ without passing through a prior MRF model and a likelihood model. For example, if only pairwise clique potentials are nonzero, the global posterior distribution of a CRF is expressed as:

$$P(\mathcal{Y}|\mathcal{X}) \propto e^{-U(\mathcal{Y}|\mathcal{X})}, \qquad \text{where :} \qquad U(\mathcal{Y}|\mathcal{X}) = \sum_{i \in \mathcal{I}} V_i(y_i|\mathcal{X}) + \sum_{i \sim j} V_{ij}(y_i, y_j|\mathcal{X}). \tag{46}$$

Compared to (43), pairwise potentials $V_{ij}(\cdot)$ are allowed to also use image data and not only labels, and pixelwise (unary) potentials $V_i(\cdot)$ can generally depend on the features of multiple pixels while the conditional independence assumption is no more necessary. Therefore CRFs generally provide increased modeling flexibility as compared to MRFs. Similar comments hold if higher-order clique potentials are used as well.

Examples of CRFs introduced in the recent remote sensing literature include spatial models for the classification of VHR images (Zhong et al., 2014a); combinations with the random forest and rotation forest ensemble classifiers (see Section 6) in the classification of hyperspectral (Li et al., 2015a) and LiDAR (Niemeyer et al., 2014) data; integration with composite kernel expansions (Wu et al., 2014); and formulations with higher-order clique potentials for road extraction (Wegner et al., 2015). Continuous-valued Gaussian CRFs have also been developed to address aerosol retrieval through the fusion of the measurements of multiple satellite sensors (Djuric et al., 2015), and joint denoising and classification of hyperspectral imagery (Zhong and Wang, 2014).

In addition to MRFs and CRFs, various other probabilistic graphical models have recently been applied to learning problems in remote sensing:

- *Pairwise Markov models* and *triplet Markov models* are further generalizations of MRFs, in which Markovianity is assumed for the joint distribution of $(\mathcal{X}, \mathcal{Y})$ and for the distribution of a triplet $(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$, where $\mathcal{Z}$ is an auxiliary stochastic process, respectively (Pieczynski, 2007). The triplet Markov model is especially useful when nonstationary behaviors should be modeled. For example, in (Lian et al., 2014), this concept is formalized in a CRF framework to define a conditional triplet Markov field for the unsupervised classification of SAR images.

- In a *hidden Markov model (HMM)*, a 1D Markov process is to be learned from a sequence of observations (Rabiner, 1989). Elegant and efficient parameter estimation and learning techniques (e.g., the Baum-Welch algorithm) can be formulated for HMMs and are well-known for 1D signal analysis. Popularity for 2D image analysis is much less, especially as compared to MRFs, which natively account for 2D lattice topologies. Nevertheless, applications of HMMs can be found in the recent remote sensing literature: to model temporal behavior and favor temporal consistency in SITS classification (Abercrombie and Friedl, 2016; Siachalou et al., 2015); to detect landmines in ground penetrating radar data (Manandhar et al., 2015); and to formalize model selection for support vector regression in combination with particle filters (Insom et al., 2015).

- *Bayesian networks* or *belief networks* are probabilistic graphical models that characterize the dependencies among a finite collection of random variables on the basis of a directed acyclic graph and of appropriate Markovianity properties (Kopparapu and Desai, 2001). Bayesian networks have recently been developed to model the relationships among bio-geophysical variables and phenomena with the aim of leaf area index retrieval (Quan et al., 2015); to perform multitemporal analysis for flood detection from VHR SAR, interferometric SAR, and ground data (D'Addabbo et al., 2016), and for the monitoring of infrastructured areas (Chen et al., 2016b); and for cloud detection and classification in the application to solar energy production (Alonso-Montesinos et al., 2016).

- A *Markov chain random field (MCRF)* is a Markov chain whose state space may be one or multi-dimensional and whose state transitions are determined by interactions with neighbors (Li et al., 2013). MCRF models have recently been applied within Bayesian schemes for land cover mapping (Li et al., 2014b) and updating (Li et al., 2013).

- *Random walker* methods first construct, on the pixel lattice, a graph in which an edge between a pair of pixels is weighted according to their spectral difference; then, segmentation is addressed by emulating a random walk from unlabeled pixels to seed pixels, as inspired by discrete potential theory and electrical circuits (Grady, 2006). In remote sensing, random walker approaches for structured prediction have been combined with SVM (Kang et al., 2015), neural networks and active learning (Bencherif et al., 2015), segmentation and multitemporal classification (Guo et al., 2015).

5.2.3. Hierarchical, multiscale, and multiresolution models

Among the main current trends of probabilistic graphical models in remote sensing, *multiscale* and *multiresolution* models are of primary importance (Willsky, 2002). The main reason is again the need for accurately characterizing

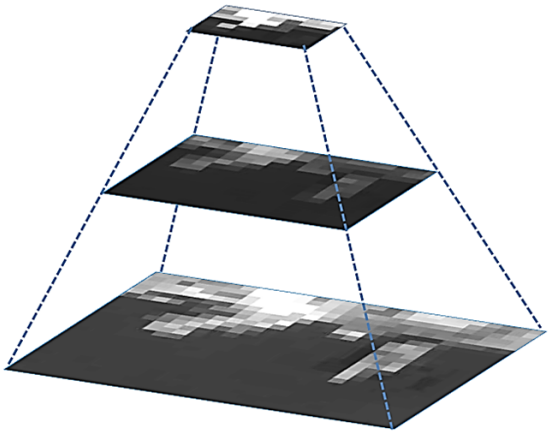


Figure 13: Example of quadtree for hierachical classification with three scale levels.

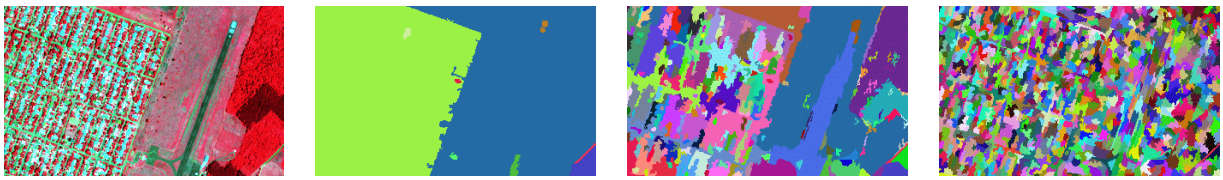
spatial information in VHR data. The rationale is that complementary information is appreciated at different spatial scales. While fine scale observations provide lots of geometrical detail but are heterogeneous and noise-sensitive, coarse scale observations capture only large image regions and covers but with strong robustness to noise and outliers. Graphical models have been proposed to take benefit from these complementary information sources both when input data are collected by sensors with different spatial resolutions and when multiscale features are extracted from an input single-resolution image.

Recent examples include: a multiscale CRF for subpixel mapping from hyperspectral images, which relates the coarser scale lattice of the input data and the finer scale lattice of the subpixel output (Zhao et al., 2015a); a hierarchical CRF for semi-supervised SAR image classification based on a mean-field approximation of the MPM solution and on generalized Gamma distributions (Zhang et al., 2015b); the combination of parametric models for the statistics of radar returns from vessels and sea clutter and of a hierachical prior for ship detection purposes (Song et al., 2016b); and the integration of a linear mixture model with an MRF for multiresolution optical image classification (Storvik et al., 2005; Moser et al., 2016).

Many multiscale or multiresolution graphical models have been introduced on the basis of hierarchical graphs (Willsky, 2002), such as quad-trees (Laferte et al., 2000), binary partition trees (Poggi et al., 2005; Kurtz et al., 2012), or even more irregular topologies (Scarpa et al., 2009). For example, a quadtree is associated with a collection of coregistered pixel lattices such that a pixel in the ith lattice corresponds to a 2×2 cell of four pixels in the $(i+1)$th lattice (see Fig. 13). The relationship between the "father" pixel in the ith lattice and its four "son" pixels in the $(i+1)$th lattice is well described by a tree, which allows data at multiple scales to be easily framed within a unique graphical model. With this topology, *hierachical MRF* models typically involve Markovianity properties along the scale axis (Laferte et al., 2000). Hierarchical MRFs have been recently developed for the classification of optical (Voisin et al., 2014), multisensor optical-SAR (Hedhli et al., 2015), and multitemporal (Hoberg et al., 2015; Hedhli et al., 2016) multiresolution data, as well as for multiscale change detection (Moser et al., 2011). In (Gaetano et al., 2014), a hierachical tree-structured MRF, which is based on an unbalanced binary tree endowed with appropriate split and merge rules, is applied to land cover characterization from a VHR SAR SITS.

5.2.4. Region-based and object-based methods

Structured prediction methods that operate on groups of similar pixels rather than on individual pixels are also topical. Especially when VHR optical images of urban, infrastructured, or agricultural areas are to be classified, the desired output structure should match the well-defined geometrical properties (shapes, borders, sizes, etc.) of the input scene. *Region-based* and *object-based image analysis (OBIA)* integrate classification and segmentation algorithms (Blaschke et al., 2008). A segmentation method identifies homogeneous regions, also known as segments or superpixels, in the input image (see the example in Fig. 14). These regions may or may not match semantically well-defined objects of the scene. Nevertheless, they are expected to characterize the geometrical structure associated



False-color display | Coarse scale | Intermediate scale | Fine scale

Figure 14: Examples of segmentation results at coarse to fine spatial scales, obtained by applying the method in (Felzenszwalb and Huttenlocher, 2004) to an IKONOS image (4-m spatial resolution). Colors in the three segmentation maps indicate segments.

with the image. Among the numerous approaches to segmentation, watershed methods (Vincent et al., 1991), graph-based region growing (Felzenszwalb and Huttenlocher, 2004), and mean-shift clustering (Fukunaga and Hostetler, 1975) are currently quite popular in remote sensing applications. Region-based structured prediction methods use segment information into the modeling of spatial information. This may be accomplished by regularizing the result of a pixelwise graphical model as a function of the segments (e.g., (Moser et al., 2013; Ghamisi et al., 2014)), by constructing a probabilistic model on a graph of segments rather than on a graph of pixels (e.g. (Zhang et al., 2015a; Yang, 2013)), or in more sophisticated ways.

Several techniques combining probabilistic graphical models and segmentation results have been proposed in the recent remote sensing literature. While they are generally effective in the discrimination of classes with well-defined geometries (e.g., in agricultural and urban scenes), they may generally be less accurate in the application to non-geometrically textured classes (e.g., a dense forest area). Random fields for classification are constructed on graphs of superpixels in (Volpi and Ferrari, 2015a,b), already discussed in Section 2.3.3, as well as in (Yang, 2013; Qin and Fang, 2014; Zhang et al., 2015a) using CRFs, in (Zheng and Wang, 2015; Liu et al., 2015a) using MRFs, in (Wu et al., 2015b) using a triplet Markov field, and in (Li et al., 2015e) using Cauchy graph embeddings. Pairwise potentials are defined in (Yang, 2013) and (Zheng and Wang, 2015) as functions of the common boundary of neighboring superpixels and possibly of their sizes. Structured prediction methods that make use of superpixels to favor edge-preserving behaviors are developed by formulating case-specific pairwise potentials and a region-based prior in (Zhao et al., 2015b), by using Gaussian mixtures in (Ghamisi et al., 2014), and by combining hierarchical segmentation, fuzzy logic, and MRFs in (Golipour et al., 2016). In (Zhong et al., 2014b), two CRF-based classifiers with partly complementary unary potentials are combined through decision fusion at the level of segments.

Hierchical random fields that take benefit not only of multiscale representations (see Section 5.2.3) but also of superpixels have also been developed, mostly for VHR optical image classification. In (Moser et al., 2013), multiscale segmentation maps are fused with local spatial context and adaptive texture features through MRF models. In (Tuia et al., 2016), a CRF is formulated to model a bidirectional hierarchy between pixels and superpixels.

5.2.5. Other graphical models and extensions

The SSVM method (Tuia et al., 2011) and, more generally, the combination of kernel and structured output learning (Volpi and Ferrari, 2015a,b; Huo et al., 2015) have been discussed in Section 2.3.3. A further approach to model dependencies within the kernel framework is to use *graph kernels*, which are kernel functions defined on the nodes of a graph and applied, for instance, in SVM classifiers (Camps-Valls et al., 2007). Compared to probabilistic graphical models, this approach shares the graph-theoretic viewpoint in the characterization of pixel interactions but differs in how the graph is used for learning, i.e., through kernels and Hilbert spaces rather than Bayesian Markovian reasoning. Recent examples of graph kernel approaches in remote sensing include the recursive graph kernel for higher-order spatial interactions in (Camps-Valls et al., 2010) as well as the SITS classification method in (Rejichi et al., 2015), which integrates an expert system, a spatial-object temporal adjacency graph aimed at capturing temporal evolution, and a graph kernel meant to measure graph similarity and related to random walks.

A generalization of graphs is represented by *hypergraphs*. Compared to a graph, a hypergraph replaces edges, which are meant to link pairs of nodes, by hyperedges, which are subsets of nodes and are meant to connect multiple nodes at once. In (Bai et al., 2015), a hypergraph is introduced to model spectral and spatial information in feature selection for semisupervised hyperspectral image classification. In (Jian et al., 2016), change detection is accomplished by applying similarity metrics to a set of hypergraphs, each associated with the image taken on one of the observation

times.

5.2.6. Marked point processes

Finally, an advanced and analytically elegant methodological tool to capture complex dependencies and interactions is given by *marked point process (MPP)* models. They stem from stochastic geometry and allow modeling random populations of objects or structures (e.g., buildings, canopies, road segments) in the imaged scene (Descombes and Zerubia, 2002). They pair Poisson processes, which indicate object locations, with additional random parameters, which describe object geometry, and allow geometrical constraints on the target structures to be encoded. Inference can be addressed using Monte Carlo Markov chains. As compared to probabilistic graphical models and especially to MRFs, MPPs use again a probabilistic "likelihood and prior" and a minimum-energy formalization but they do not need a predefined reference graph, which makes them especially advantageous when the spatial distribution of the target objects is irregular and complex. MPP models have recently been developed for vehicle and traffic segment detection in LiDAR point clouds (Borcs and Benedek, 2014), for target detection and tracking in time series of inverse SAR images (Benedek and Martorella, 2014), and for tree (Zhou et al., 2013) and boat (Craciun and Zerubia, 2014) detection from VHR optical images.

6. Ensemble learning methods

In machine learning, an ensemble is a combination of multiple learning algorithms with the purpose of improving the classification performance over each one of the individuals. In other words, an ensemble learner combines a number of so-called weak learners to generate a strong learner (Rokach, 2010) (see Fig. 15). In a sense, the concept of combining several classifiers can also be referred to as decision fusion. While decision fusion is focused more on the specific rules to combine classification results, this chapter will focus more on the way the ensemble is composed.

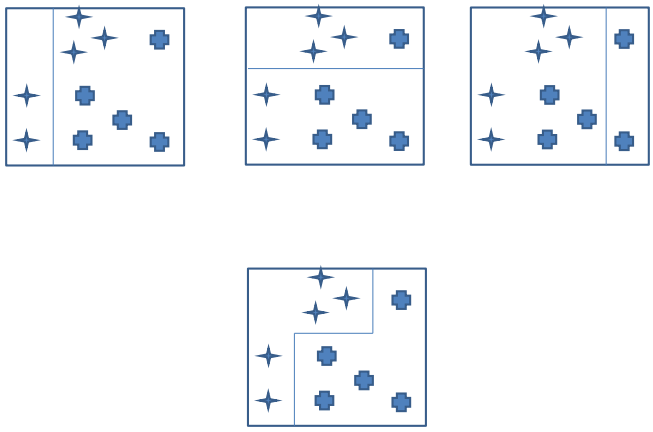


Figure 15: Illustration of 3 classifiers with their respective decision boundaries. The bottom classifier is obtained by majority voting on the three classifiers on top.

The family of ensemble methods is probably the oldest from the methods described in this chapter. Ensemble learning is as mature area of research in which important research progress has been made from 1990 on. Nevertheless, ensemble methods remain very popular tools for classification still nowadays. In the rest of this section, we provide some basic concepts in ensemble learning, as well as an overview of recent researches based on the corpus appearing in the recent state of the art on remote sensing analysis.

6.1. A taxonomy of ensemble methods

Suppose that there are L classifiers available. Denote the L classifiers as $f^l(\cdot)$, with $l = \{1, \ldots, L\}$. For a test sample $\mathbf{x}_i$, with class label $y_i \in \{1, \cdots, C\}$ each classifier l will be either correct ($f^l(\mathbf{x}_i) = y_i$) or wrong ($f^l(\mathbf{x}_i) \neq y_i$).

An ensemble learning method is supposed to be more accurate that the individual classifiers. For this, a necessary condition is that the individual classifiers are accurate and diverse. An accurate classifier is one that has an error rate which is better than random guessing. Two classifiers are diverse if they behave differently on the same test sample.

Ensemble methods use a number of basic principles:

- In *Stacking* or stacked generalization (Wolpert, 1992), the basic principle is that the predictions of several learning algorithms are used as input to a higher level classifier (see Fig. 16).

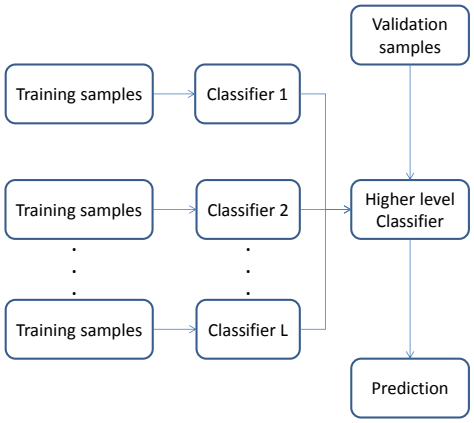


Figure 16: The principle of stacking

- *Bagging* or bootstrap aggregating: hereby, L training sets are generated from the original training set by random sampling with replacement (bootstrapping) (Breiman, 1996). Each training set $\{\mathbf{x}_i^l, y_i^l\}_{i=1}^{n_l}$, composed of n_l labeled examples, is classified separately by classifier $f^l(\cdot)$. The final decision is then performed by a higher level classifier, or a fusion rule, the most straightforward being a majority voting (see Fig. 17). The very popular classifier Random Forests (Breiman, 2001) is based on the bagging principle: specifically, a random forest is a bagging of decision trees[4], where each tree only screen a part of the original feature space (see the 'Random subspace methods' item below).

- *Boosting* (Schapire, 1990) is an iterative procedure that adaptively manipulates the training set to generate multiple classifiers. Each training sample is assigned a weight, and at each iteration l, the weighted error on the training set is minimized by a classifier $f^l(\cdot)$. The weighted error from $f^l(\cdot)$ is then applied to update the weights on the training set, in the sense that more weight is given to wrongly classified samples and less weight to correctly classified samples. The final classifier $f(\cdot)$ is constructed by a weighted vote of the individual classifiers: $f(\mathbf{x}_i) = \sum_l w_l f^l(\mathbf{x}_i)$, where the weights w_l are given by the accuracy of the corresponding classifier (see Fig. 19). The popular ADABOOST (Freund and Schapire, 1999) algorithm is a boosting algorithm using decision trees as base classifiers.

[4] A decision tree (Quinlan, 1986) is a basic classifier in a tree-like shape that contains internal nodes, branches and leaves which are the endnodes of the tree. In each internal node, all features are individually tested and the feature that provides the best split is chosen to split up the node into different branches, leading to the next level of internal nodes. The splitting stops when all samples at a node have the same class label. This node then becomes a leaf and denotes the class label.

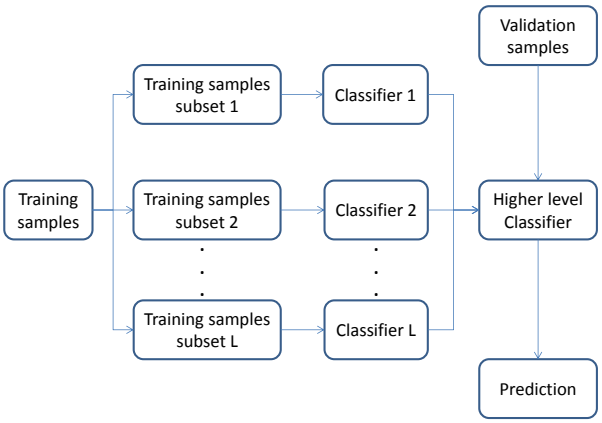


Figure 17: The principle of bagging

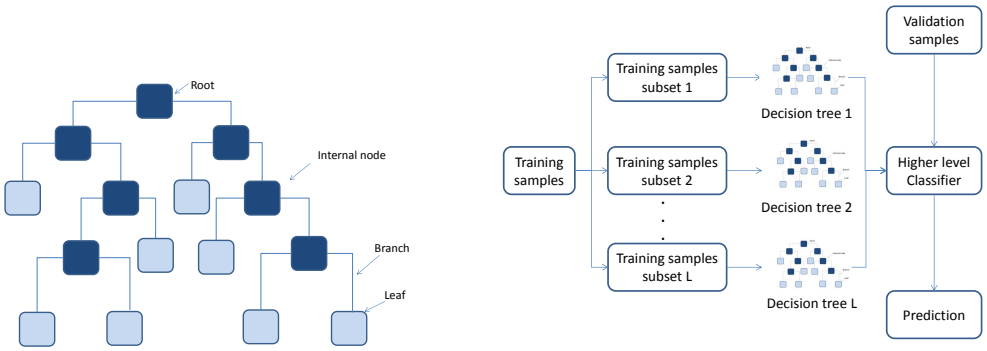


Figure 18: a) a decision tree; b) Random Forest classifier

- *Random Subspace methods* (Ho, 1998) train classifiers on randomly chosen subspaces of the original feature space. A particular Random Subspace method is *Random Forest* (Breiman, 2001). In this method, the principle of bagging is applied to generate multiple decision trees. Rather than using standard decision trees, at each node, a random subset selection of features is chosen (sometimes referred to as feature bagging) from which the best feature is obtained to perform the splitting. As discussed above, the training examples for each decision tree are also subsampled (see the 'Bagging' item above). An extension of Random Forest is given by the Extreme

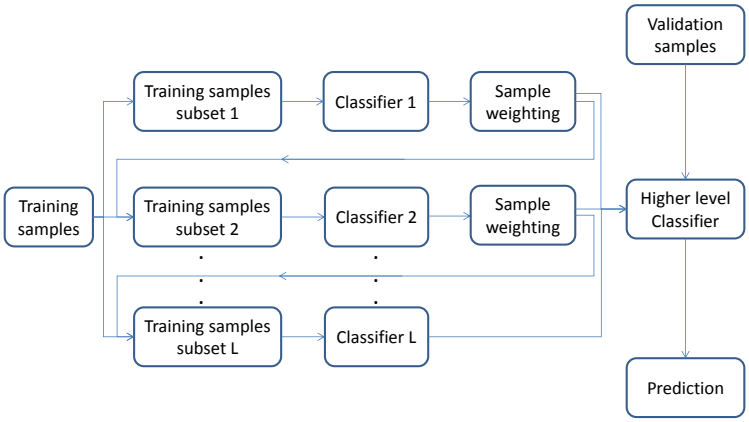


Figure 19: The principle of boosting

Randomized Tree or ExtraTrees (Geurts et al., 2006). These are trained like the Random Forest, with that difference that at each node, the splitting is randomized, in the sense that for each feature a randomly selected threshold value is chosen, after which the best split is chosen. Another difference with Random Forest is that with ExtraTrees, all training samples are used at each node.

Usually, the superior performance of ensemble classifiers over single classifiers is explained by the Bias-Variance trade-off. The bias and variance are given by:

$$\begin{aligned} \text{Bias} &= y - \frac{1}{L}\sum_{l=1}^{L} f^l(\mathbf{x}) = y - \hat{f}(\mathbf{x}) \\ \text{Variance} &= \sum_{l=1}^{L}\left(f^l(\mathbf{x}) - \hat{f}(\mathbf{x})\right)^2 \end{aligned} \tag{47}$$

A high bias causes a learner to miss relevant relations between features and output and thus causes underfitting. The variance is the error of a classifier caused by the sensitivity of the classifier to fluctuations of the training set. A high variance causes overfitting. In the case of classification, the zero-one loss can be applied, i.e. for classifier f, $\mathcal{L}(f(\mathbf{x}_i, y_i) = 0$ if $f(\mathbf{x}_i) = y_i$ and $\mathcal{L}(f(\mathbf{x}_i, y_i) = 1$ if $f(\mathbf{x}_i) \neq y_i$. The bias is then a measure for the loss of the ensemble relative to the optimal loss, while the variance is a measure for the average loss relative to the optimal loss. Due to the avaraging term in the definition of the variance, bagging will in general reduce the variance without affecting the bias too much. The process of boosting is shown to reduce the bias as well (Freund and Schapire, 1999). Random Forest and ExtraTrees ensemble methods were specifically designed to reduce the variance even further.

6.2. Why ensemble methods are so popular in remote sensing

Ensemble methods have a relevant history in remote sensing and are still very popular nowadays in remote sensing applications. There are three main reasons for this popularity.

- The first reason is a pragmatic one: many ensemble learning methods obtain superior classification accuracies compared to most baseline classifiers. They have been shown to obtain similar accuracies across a large number of real-world datasets to other state of the art classifiers such as the SVM (Farnandez-Delgado et al., 2014), which is a popular classifier in remote sensing applications.

- The reduced variance makes that ensemble methods typically show better generalization performance. This property is favorable for cases where the training set is not perfectly representative for general scenarios. These situations appear typically for high-dimensional data and large data sizes, along with limited training data, situations which occur in many remote sensing applications (Zhong and Wang, 2007) and in hyperspectral images in particular (Ham et al., 2005a) .

- By construction, ensemble methods have low training complexity. For high-dimensional data, it is often infeasible to learn a single strong classifier. The bagging principle allows each of the base classifiers to be trained on a small subset of the training data and is a highly parallelizable process, which dramatically decreases training times.

To demonstrate the popularity of ensemble learning methods in the domain of remote sensing, we now give an overview of the recent literature (from 2014 on) on the development and application of these methods for the analysis of remote sensing data. Ensemble learning methods have been developed and applied for different analysis strategies on remote sensing data.

One particular analysis task is multisource remote sensing data fusion. A recent overview on ensemble learning methods for data fusion in remote sensing is given in Merentitis and Debes (2015). In Zhang et al. (2015d), the problem of classification of multisource (hyperspectral and Lidar) remote sensing data is treated. For this, an ensemble multiple kernel active learning approach is proposed. Each source of information is classified using a separate kernel that is adaptively optimized using an active learning process. At the end, a fusion strategy is applied that makes a final decision based on the probabilistic outputs of each learning process.

The most important remote sensing analysis task for which ensemble methods are applied nowadays is the classification of hyperspectral images. In Merentitis et al. (2014), the classification of hyperspectral images is addressed from a bias-variance decomposition point of view. In Xia et al. (2016b), a class-specific Rotation Forest is proposed for hyperspectral image classification, in which the principal component data transformation step is performed for each class separately. In Sun et al. (2015c), the problem of imbalanced hyperspectral image classification is treated. Hereby, the number of pixels belonging to the different classes is imbalanced such that traditional classification methods are prone to assign more pixels to the larger classes. The bagging principle of ensemble learning is applied for partitioning the larger classes into different smaller groups. SVM is the selected base classifier, for which the maximum margin criterion is adopted to guide the ensemble learning method.

One particular development in hyperspectral image classification is spectral-spatial hyperspectral image classification. Rather than only using the spectral reflectance values of a hyperspectral pixel to perform the classification, contextual information from the neighborhood of a pixel is included in the feature set. Since this process dramatically increases the dimensionality of the data, ensemble methods are of particular interest. In Xia et al. (2015b), different random subspace ensemble methods (Random Forest, Rotation Forest and Rotation Subspace) combined with 2 base classifiers (decision tree and extreme learning machines) are compared for the specific task of spectral-spatial classification of hyperspectral images. The applied contextual features are based on the extended multi-attribute profiles. In Bao et al. (2016), morphological attribute profiles generate the contextual information. These features rely on a number of filtering parameters for which it is not easy to obtain optimal values. Therefore, a series of attribute profiles using different parameter settings are used as input in a Rotation Forest ensemble. In Hang et al. (2016), the contextual information is contained in a matrix-based spectral-spatial feature representation, after which matrix-based discriminant analysis is performed to learn the feature subspace for classification. Random sampling is used to generate a subspace ensemble method for the final classification.

In Xia et al. (2015a), integration of ensemble methods with contextual models is proposed. A Rotation Forest ensemble is applied on the spectral features. The contextual information is then modeled by a MRF prior after which a maximum a posteriori problem is solved using the α-expansion graph cuts optimization method (see the Section on structured outputs 5.1.1 for details on the α-expansion). In Damodaran et al. (2015), the contextual information is included by using MRF as well. However, here, rather than a fixed ensemble of classifiers, classifiers are dynamically selected for each input pixel, by exploiting the local information content of the pixel. In Li et al. (2015a), hyperspectral image classification is performed by combining Rotation Forest with multiclass AdaBoost. Moreover, the obtained probabilities are fed into a CRF to incorporate contextual information.

Another specific problem in hyperspectral image classification tasks is the availability of sufficient numbers of training samples. Specific ensemble strategies were developed that work well in the case of very small training sizes.

In Ayerdi and Romay (2016), a new ensemble classifier called anticipative hybrid Extreme Rotation Forest is proposed for hyperspectral image classification, and combined with a semisupervised strategy to include unlabeled samples in the training process. In the ensemble strategy, all individual classifiers are first trained on a small amount of training samples, after which they are ranked according to their accuracy results and a probabilistic model is built for the selection of different classifier architectures. These are then trained properly with the remaining training samples. In Xia et al. (2015c), Random Forest ensemble classification of hyperspectral images is proposed, in which disjoint feature subspaces, that serve as input for the classifier, are obtained by performing semi-supervised feature extraction. To alleviate he problem of limited training samples, a rotation-based SVM ensemble is proposed in Xia et al. (2016a).

A specific task in hyperspectral image classification is target detection, which can be regarded as a binary classification problem. A typical problem in target detection is that labeled data are even more difficult to obtain. In Dong et al. (2015), target detection is solved by learning a distance metric to separate target pixels from the background. A multimetric method is constructed by Random Forests. In Fei et al. (2015) an automatic target recognition system was developed for underwater mine classification, making use of composite relevance measures for feature selection. These generate a large number of classifiers, which are combined using an ensemble learning scheme in the framework of the Dempster-Shafer theory.

Ensemble learning methods are not only popular for hyperspectral image analysis. VHR remote sensing data sets are regularly analysed using ensemble learning techniques, e.g. for scene classification and object recognition. In Tokarczyk et al. (2015), the problem of semantic classification of VHR remote sensing images is treated by ensemble methods. Here, a boosting tree classifier is applied for textural feature generation and selection. In Zhang et al. (2016a), scene classification on VHR is performed by a NN ensemble, using a gradient boosting random CNN framework that combines many deep neural networks. In Huang et al. (2015), several ensemble learning strategies including Random Forest and tree bagging methods are used for urban water-body identification from VHR images.

Finally, we report a number of recent specific remote sensing applications that were treated with the use of ensemble learning strategies. In Wu et al. (2015a), regression ensemble methods (e.g. Random Forest) were applied to estimate the stem volume of individual trees based on waveform Lidar data. In Wang et al. (2015d), Adaboost is applied for point cloud classification of Laser Scanning data. In Bray and Link (2015), neural networks and Random Forest classifiers were applied to identify unexploded ordnance based on Magnetometry data. In Gokon et al. (2015), decision trees were applied for change detection, and in particular the detection of destroyed buildings after the Japanese tsunami based on high-resolution radar (TerraSAR-X) data, and in Klesk et al. (2015), a boosted decision tree is applied for landmine detection from Ground Penetrating Radar data.

7. Sparse learning methods

Because of continous technical improvements in sensor design, most signals and images are (spatially, spectrally, temporally) sampled at ever increasing resolutions. This delivers higher quality images containing more useful information, but at the same time increases the amount of redundant information. In order to see the useful information more clearly, it is important to remove the redundant information as much as possible. This can be done by transforming the data in a representation that excludes much of the redundant information.

It has been recognized for a long time that signals can be compactly represented. Since 1990, complete multiscale and multi-orientational orthogonal representations, such as wavelets, ridgelets, curvelets and contourlets have been extremely popular. Nowadays, overcomplete representations are acknowledged for their stability, compactness and robustness. One particular approach using overcomplete representations is compressive sensing, in which the sparsity of a signal is exploited reconstructing it by a small number of samples (Donoho, 2006). The basic idea of such sparse representations is to describe signals as a sparse linear combination of elements, referred to as atoms, in an overcomplete dictionary. When applied to supervised classification, the dictionary is represented by the training samples, and the underlying assumption is that each data sample can be linearly represented by a small number of training samples from the same class.

Remote sensing data and hyperspectral images in particular are generally high dimensional and highly redundant. This makes the problem of hyperspectral image classification a perfect candidate to apply overcomplete representations to.

7.1. Sparse learning classification

7.1.1. The sparse representation classifier (SRC)

Suppose that we want to classify a hyperspectral image with d spectral bands, containing pixels $\mathbf{x} \in \mathbb{R}^{\mathbf{d}}$. Assume that the number of classes is C and that for each class $c = \{1, \cdots, C\}$, labeled training pixels $\mathbf{x}_{ci}$, with $i = \{1, \cdots, n_c\}$ are available. Denote $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_C] \in \mathbb{R}^{d \times n}$ where the training set of class c is given by the n_c samples from that class $\mathbf{X}_c = [\mathbf{x}_{c1}, \mathbf{x}_{c2}, \cdots, \mathbf{x}_{cn_c}]$ and $n = \sum_{c=1}^{C} n_c$ is the total number of training samples.

When a pixel $\mathbf{x}^*$ is supposed to belong to class c, it is assumed that it can be represented as a linear combination of the training pixels of that class:

$$\begin{aligned} \mathbf{x}^* &= \sum_{i=1}^{n_c} \mathbf{x}_{ci}\alpha_{ci} \\ &= [\mathbf{x}_{c1}, \cdots, \mathbf{x}_{cn_c}][\alpha_{c1}, \cdots, \alpha_{cn_c}]^T \\ &= \mathbf{X}_c\alpha_c \end{aligned} \tag{48}$$

where α_c is the vector of coefficients corresponding to the training samples in class c. Since the label of $\mathbf{x}^*$ is unknown a priori, it can only be represented as a linear combination of all training samples:

$$\begin{aligned} \mathbf{x}^* &= [\mathbf{X}_1, \cdots, \mathbf{X}_c][\alpha_1, \cdots, \alpha_C]^T \\ &= \mathbf{X}\alpha \end{aligned} \tag{49}$$

Ideally, all coefficients are zero, except those corresponding to training samples from the class that $\mathbf{x}^*$ belongs to. This leads to the following optimization problem:

$$\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_q \text{ s.t. } \mathbf{X}\alpha = \mathbf{x}^* \tag{50}$$

or, when allowing noise $\mathbf{z}$ with bounded energy $\|\mathbf{z}\| < \epsilon$ in Eq. (49):

$$\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_q \text{ s.t. } \|\mathbf{X}\alpha - \mathbf{x}^*\|_2 \leq \epsilon \tag{51}$$

Using the Lagrangian multiplier notation, this becomes:

$$\hat{\alpha} = \arg\min_{\alpha} \|\mathbf{X}\alpha - \mathbf{x}^*\|_{\mathbf{2}} + \lambda\|\alpha\|_{\mathbf{q}} \tag{52}$$

The second term acts as a regularizer, since the least squares solution becomes unstable for large dictionaries. The l_q-norm applied on the regularizer determines the amount of sparsity. For $q = 0$, it counts the number of nonzero elements in α, but the optimization problem becomes NP-hard. It can be appoximately solved by (orthogonal) matching pursuit (OMP) (Tropp and Gibert, 2007). These are greedy algorithms that iteratively select an atom from the dictionary that maximizes the correlation with the current residual of the signal. Alternatively, it can be replaced by the l_1-norm, leading to a linear programming problem, which can be solved by several methods such as Lasso (least absolute shrinkage and selection operator) (Tibshirani, 1996).

Once the sparse vector α is obtained, $\mathbf{x}^*$ is reconstructed C times, each time using only the coefficients of one particular class, obtaining the following residuals:

$$r_c(\mathbf{y}) = \|\mathbf{X}_c\alpha_c - \mathbf{x}^*\|^2 \tag{53}$$

Finally, $\mathbf{x}^*$ is assigned to the class corresponding to the lowest residual:

$$\hat{c} = \arg\min_{c} r_c(\mathbf{x}^*) \tag{54}$$

Firstly developed for face recognition (Wright et al., 2009), SRC has been succesfully used in different applications and has been recently applied to the problem of hyperspectral image classification (Chen et al., 2011). The succes of SRC is the high redundancy and low coherency of the training samples making up the dictionary, which leads to low reconstruction errors and high sparsity levels, making the classifier robust. There are however a number of downsides to SRC, which are particulary important for hyperspectral image data.

- As described above, the greedy algorithms to solve SRC typically exploit the correlation between the residual and the training samples, rather than the actual distance. A test sample may have high correlation to training samples that are separated. In such scenario, SRC will likely select atoms from the dictionary whose class label is different from the test sample, which results in classification errors. This problem is likely to occur in hyperspectral data, since these typically contain highly correlated samples from different classes.
- Another drawback from the SRC is that the dictionary contains all training samples from all classes. The SRC therefore does not make use of class label information to calculate the sparse coefficients. Merely in the last step, where the residuals are employed, the class label information is used. This is a limitation in the particular situation where samples of different classes are highly correlated, such as in hyperspectral images.
- That the dictionary contains the training samples from all classes at once becomes even more of a drawback for small dictionaries. In remote sensing applications, collecting training data is typically expensive and time consuming, which makes that typical training sizes are small. This will even more weigh on the lack of class label information in the SRC.

To solve these issues, specific methodologies have been recently developed.

7.1.2. Collaborative representation classification (CRC)

It has been argued that, rather than the sparsity constraint, it is the collaborative nature of the optimization that improves classification accuracies. By collaborative, it is meant that, rather than just a sparse number of atoms, all the atoms in the dictionary collaborate to represent a test sample. The collaborative representation classifier can be implemented as a regularized least squares problem:

$$\hat{\alpha} = \arg\min_{\alpha} ||\mathbf{X}\alpha - \mathbf{x}^*||_2 + \lambda||\mathbf{I}\alpha||_2 \tag{55}$$

where $\mathbf{I}$ is the identity matrix. In contrast to the l_1-norm regularizer of the SRC, a closed-form solution of this optimization problem exists and is given by:

$$\hat{\alpha} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{x}^* \tag{56}$$

This makes the collaborative representation classifier much more time efficient than the SRC. Final classification is obtained by using the same strategy as in the SRC, by assigning the test sample to the class with the smallest residual.

In Li et al. (2014a), a class-specific collaborative representation classifier was proposed for hyperspectral image classification. Here, the sparse codes are optimized for each class c separately:

$$\hat{\alpha}_c = \arg\min_{\alpha_\mathbf{c}} ||\mathbf{X}_c\alpha_c - \mathbf{x}^*||_2 + \lambda||\mathbf{\Gamma}_c\alpha_c||_2 \tag{57}$$

and the identity matrix in the regularizer was replaced by a nonuniform distance-weighted Tikhonov matrix:

$$\mathbf{\Gamma}_c = \begin{vmatrix} ||\mathbf{x}^* - \mathbf{x}_{c1}||^2 & & 0 \\ & \ddots & \\ 0 & & ||\mathbf{x}^* - \mathbf{x}_{cn_c}||^2 \end{vmatrix} \tag{58}$$

This optimization also has a closed-form solution:

$$\hat{\alpha}_c = (\mathbf{X}_c^T\mathbf{X}_c + \lambda\mathbf{\Gamma}_c^T\mathbf{\Gamma}_c)^{-1}\mathbf{X}_c^T\mathbf{x}^* \tag{59}$$

7.1.3. Structured sparsity learning

Sparse learning can only work when the dictionary atoms are highly redundant. This is however not the case when the training sample sizes are small. As a consequence, samples belonging to different classes may have similar sparse codes while samples belonging to the same class may have totally different sparse codes. To alleviate this problem, one can try to enforce samples belonging to the same class to have similar sparse codes.

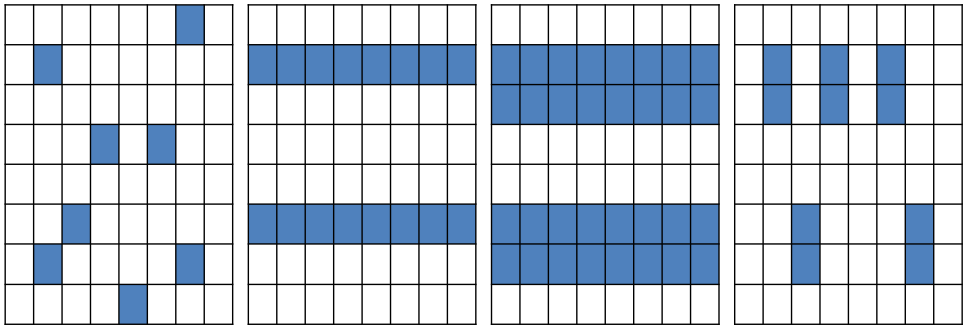


Figure 20: Sparse coefficient matrix $\mathbf{A} \in \mathbb{R}^{n \times \Omega}$, with n the number of dictionary atoms and Ω the number of pixels in a local region. From left to right: sparse representation, joint sparse representation, group sparse representation and sparse group sparse representation.

Hyperspectral images are spatially highly correlated, which means that neighboring pixels are likely to belong to the same classes. One particular methodology enforces neighboring pixels that are likely to belong to the same class, to have similar sparse codes by enforcing a structured sparsity constraint. In this way, the sparse codes of all the neighbors are simultaneously obtained. The joint sparsity prior assumes that all the sparse codes of the neighbors share the same set of dictionary atoms. The optimization problem of SRC becomes:

$$\hat{\mathbf{A}} = \arg\min_{A} ||\mathbf{X}\mathbf{A} - \mathbf{\Omega}||_2 + \lambda ||\mathbf{A}||_q \tag{60}$$

where $\mathbf{\Omega}$ is the collection of all pixels in a window centered around $\mathbf{x}^*$. $\mathbf{A} \in \mathbb{R}^{n \times \Omega}$ is now a sparse coefficient matrix and $||\mathbf{A}||_q = \sum_{i=1}^{n} ||\mathbf{A}_i||_2$, where $\mathbf{A}_i$ is the i-th row of $\mathbf{A}$. The joint sparsity prior forces the sparse coefficient matrix to have as few nonzero rows as possible. We will refer to this classifier as the joint sparse representation classifier (JSRC).

In Sun et al. (2014), it has been shown that the use of the joint sparsity prior obtains smoother hyperspectral land cover maps than the SRC. Moreover, other structured sparsity priors have been proposed. It makes sense to group together the dictionary atoms (i.e. the training pixels) that belong to the same class. In this way, test samples can be enforced to be represented by these groups rather than by individual atoms. A collaborative group sparsity prior enforces the coefficients to have a groupwise sparsity pattern:

$$\hat{\mathbf{A}} = \arg\min_{A} ||\mathbf{X}\mathbf{A} - \mathbf{\Omega}||_2 + \lambda \sum_{c=1}^{C} w_c ||\mathbf{A}_c||_2 \tag{61}$$

where w_c is the weight of each group (class) and is usually set to be the square root of the number of training samples of the group to compensate for different group sizes. A sparse group sparsity prior enforces the sparse codes to be not only groupwise sparse, but also sparse within each active group:

$$\hat{\mathbf{A}} = \arg\min_{A} ||\mathbf{X}\mathbf{A} - \mathbf{\Omega}||_2 + \lambda_1 \sum_{c=1}^{C} w_c ||\mathbf{A}_c||_2 + \lambda_2 \sum_{c=1}^{C} w_c ||\mathbf{A}_c||_1 \tag{62}$$

Fig. 20 depicts the sparsity of the matrix $\mathbf{A}$ for these different priors.

7.1.4. Dictionary Learning

Rather than stacking all the training samples and using these as a fixed dictionary, another option is to learn the dictionary itself to obtain a better representation of the data set (Aharon et al., 2006). In this way, the optimization problem becomes:

$$\hat{\alpha} = \arg\min_{\alpha, \mathbf{D}} ||\mathbf{D}\alpha - \mathbf{x}^*||_2 + \lambda ||\alpha||_q \tag{63}$$

where the dictionary $\mathbf{D}$ needs to be learned simultaneously with the sparse codes. This optimization problem can be solved iteratively, where in each iteration, Eq. (63) is optimized alternately with respect to either the dictionary or the sparse coding by fixing the other.

The updating of the sparse coding, given the dictionary is solved by the sparse approximation methods from above. To solve Eq. (63) with respect to the dictionary, unsupervised dictionary learning methods such as the method of optimal direction (Engan et al., 1999), K-SVD (Aharon et al., 2006) and Online dictionary learning (Mairal et al., 2009) were developed. The first two methods are batch methods that iteratively optimize the dictionary for a complete set of test samples, while the latter method iteratively updates the dictionary each time a new test sample is presented. These methods are able to generate even lower reconstruction errors than SRC with a fixed dictionary.

However, for classification purposes, it makes sense to construct supervised dictionary learning methods, since lower reconstruction errors do not necessarily lead to higher classification accuracies. Therefore, supervised dictionary learning methods need to be developed, in which the dictionary is updated to lower the reconstruction errors, and at the same time improve classification accuracies. This can be done e.g. by discriminative dictionary learning, where the reconstruction error contributed by atoms from the correct class is minimized, while the reconstruction error contributed by the other atoms is maximized. Another option is incoherent dictionary learning, that tries to eliminate atoms in the dictionary that are shared by samples from different classes.

A more efficient way is to optimize the dictionary along with the parameters of the applied classifier (e.g. the decision boundaries). The label consistent K-SVD combines the dictionary and the classifier parameters into one single parameter space, hereby minimizing reconstruction and classification errors simultaneously (Zhuolin et al., 2013). All these methods however still do not guarantee an optimal classification error, since a lower cost function still can be obtained for a reduced reconstruction error and an increased classification error.

Instead, in an appropriate optimization method, the dictionary should be driven by the minimization of the classification error. In task driven dictionary learning (Mairal et al., 2012), the dictionary is optimized by minimizing the classification risk $\mathcal{L}(\mathbf{D}, \mathbf{W}, \mathbf{x})$, which is formulated in terms of the dictionary atoms, the sparse coefficients and the classifier's parameters $\mathbf{W}$:

$$\mathcal{L}(\mathbf{D}, \mathbf{W}, \{\mathbf{x}\}_{i=1}^{N}) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{W}\alpha_i\|^2 + \mu\|\mathbf{W}\|^2 \tag{64}$$

where $\mathbf{y_i} \in \mathbb{R}^C$ is a binary vector representing the class label of sample $\mathbf{x}_i$, α_i is the sparse code of the sample, and N is the total amount of training samples. The first term in the sum represents the classification error for a training sample $\mathbf{x}$, measured by a linear regression, where the sample is represented by its sparse code as feature vector. $\mu > 0$ is a regularization parameter to avoid overfitting of the classifier. $\mathcal{L}(\mathbf{D}, \mathbf{W}, \mathbf{x})$ is then minimized with respect to the dictionary $\mathbf{D}$ and the classifier parameters $\mathbf{W}$. This guarantees that the learned features (sparse codes) are optimal for the trained classifier.

7.1.5. Sparse unmixing

A particular analysis task for hyperspectral images is spectral unmixing, in which a pixel is assumed to be a mixture of a (limited) number of pure spectra, the so-called endmembers. These mixtures appear for different reasons, amongst them the limited spatial resolution of the sensor, the existence of materials consisting of intimate mixtures and atmospheric scattering effects. To analyse such a mixed pixel, one needs to acquire information on the number of endmembers, the endmember spectra and the fractional abundances with which the mixture is built up from these endmembers. The topic of spectral unmixing is treated in depth in another chapter of this book. One of the particular state of the art methodologies for spectral unmixing however is sparse learning based unmixing, and therefore deserves a special mention in this chapter.

One of the most applied spectral mixture models is the linear mixture model, in which a pixel is assumed to be a linear mixture of the endmembers, with fractional abundances that are positive and sum up to one. When the endmembers are known in advance (e.g. spectra collected on the ground by a field spectro-radiometer), unmixing amounts to finding the optimal subset of endmembers in a (potentially very large) spectral library that best models the mixed pixel. In a sparse representation setting, the spectral library takes the role of the dictionary and the obtained sparse codes represent the fractional abundances (Iordache et al., 2011).

7.2. Sparse learning for remote sensing

Most of the recent developments in sparse representations for remote sensing are being done in the area of sparse representation-based classification of hyperspectral images.

Fu et al. (2016) developed a shape-adaptive JSRC for hyperspectral image classification. Rather than fixed windows for the local neighborhood, adaptive sizes and shapes are applied. These are obtained from using a shape-adaptive algorithm to the first PCA-band to constructs a shape-adaptive local smooth region for each test sample. Similarly, Zhang et al. (2014) proposed a nonlocal weighted JSRC-based method, where a weight matrix reflects the correlation between a central pixel and its neighborhood. This is further refined in Chen et al. (2016a), where the weights are optimized simultaneously with the sparse coding in the JSRC.

Li and Du (2013) developed a joint collaborative representation for spectral-spatial hyperspectral image classification. Rather than using a joint structured sparsity prior as in JSRC, the test samples are spatially averaged around a local window to exploit local spatial correlations. In Xiong et al. (2015), the same method is applied using a weighted averaging of the test samples. He et al. (2015) applied a SRC for spectral-spatial hyperspectral image classification, by applying a spatial translation-invariant wavelet representation on the test sample as well as on the dictionary elements. By a probabilistic approach, it is analyzed that the sparsity recovery and the classification accuracy of the wavelet representation-based SRC is improved over the regular SRC. A similar approach was proposed in Jia et al. (2015a), where a Gabor representation was applied on both the test sample and the dictionary atoms in a CRC. In Li et al. (2015b), a superpixel-level SRC framework with multitask learning is proposed for hyperspectral imagery. The local neighborhood of the test sample is obtained by a superpixel segmentation. On the obtained superpixels, a JSRC is applied. The whole framework is then described as a multitask learning problem, where for each feature, subdictionaries are constructed and a collaborative coding of the test sample is obtained. The minimal total residual, which is the sum over the obtained residuals for each of the applied features, decides on the test sample's class label.

Jia et al. (2015b) applied a low-rank representation to the local neighborhood of a test sample. Hereby, the spatial adjacency matrix is decomposed into the sum of a low-rank matrix and an error matrix. The optimization problem is to simultaneously minimize the rank of the first matrix and the elements of the error matrix. The obtained representation is used as input to a SRC classifier. On the obtained sparse representation-based probability estimates, a graph-cut segmentation algorithm was applied. This algorithm takes into account local spatial correlations and is applied for spectral-spatial classification of hyperspectral images.

In Soltani-Farani et al. (2015), a joint structured sparsity prior is included in a SRC method, where the dictionary is learnt from the data using a (unsupervised) block coordinate descent strategy. The method is explained in the framework of sparse unmixing. The joint sparsity prior is obtained by partitioning the pixels in spatial neighborhoods, called contextual groups. It is then assumed that pixels from the same contextual groups are represented by a common set of atoms from the dictionary. In Soltani-Farani and Rabiee (2015), a similar approach is presented, where it is assumed that fractional abundances of neighboring pixels are distributed according to a common Laplacian scale mixture prior.

To incorporate class label information in the SRC, one can generate separate dictionaries for each of the classes. In Cui and Prasad (2015), a class-dependent SRC for hyperspectral image classification is proposed where the class-dependence is incorporated at the level of the OMP solver. Rather than only using the residual for each class, the distance from the test sample to the training samples is simultaneously used to decide on the test samples labeling. This is done by employing the OMP and a kNN classifier for each class individually. In Bo et al. (2016), first a SCR is applied, after which the obtained codes are applied to generate meta-features for a multiclass SVM. Finally, decision fusion is applied on the SCR and SVM results by using a multiplicative fusion rule.

In Li et al. (2015d), two collaborative representation-based Nearest Neighbor classifiers are proposed for hyperspectral image classification. On the one hand, sparse codes are calculated based on the entire dictionary of training samples using the CRC (Eq. (56)), after which a majority voting on the k largest elements of the obtained $\hat{\alpha}$ decides on the label of the test sample. On the other hand, the kNN classifier is used to obtain a class-specific dictionary that is then used in the class-dependent CRC (Eq. (58)). The test sample is then assigned to the class with the smallest representation error. In the same line, in Zou et al. (2015), an SRC-based kNN classifier is proposed. Rather than using the class-specific residuals, a majority voting on the k largest elements of the obtained $\hat{\alpha}$ decides on the label of the test sample.

Sun et al. (2015d) propose a sparse hyperspectral image classification strategy for small training sizes by combining the concepts of task-driven dictionary learning and structured sparsity constraints. The authors enforce joint and

Laplacian structured sparsity priors on the task-driven dictionary learning method. The proposed method benefits both from the supervised dictionary learning approach and the inclusion of spatial information by the joint sparsity of local neighborhoods. In Wang et al. (2015c), the task-driven dictionary learning strategy is extended to a semisupervised approach in which the unlabeled samples are exploited along with a limited number of labeled samples to improve the classification of the task-driven setting. Unlabeled samples are included in the process by assigning to them an adaptive confidence probability as the likelihood of a logistic regression. A classification-oriented loss function is then minimized over both labeled and unlabeled samples using the task-driven dictionary learning approach.

Sparse methods are also popular in feature selection: selecting the correct features, rather than the right samples, can also be a desirable behavior that can be achieved through sparsity: in Tuia et al. (2014a), authors propose to use a ℓ_1-based selection strategy that explores the space of possible spatio-spectral filters, in order to build incrementally the best possible feature input space for classification. This approach is then extended to a multiclass logic in Tuia et al. (2015), where a $\ell_1\ell_2$-based selection criterion is used to discover the relevant features, which are useful (thus, lead to non-zero coefficients in a classifier) for more than one class. Moreover, the search is extended to more abstract feature representations by letting relevant features being re-filtered, mimicking the functioning of convolutional neural networks (see Section 3.2) on cascades of pre-defined filters types.

Another area where sparse representation-based classifiers are being developed is the area of hyperspectral target detection, which can be regarded as a one-class classification problem. In Zhang et al. (2015c), target detection is performed by a sparse representation-based binary hypothesis model. The sparse code of a test sample is calculated from a dictionary of background training samples and target training samples respectively. The residual errors decide on wether the test sample is a target or belongs to the background. In Song et al. (2016a), a kernel SRC is proposed for target detection by developing a one-class classifier based on the SRC. To improve the data separability between the target and outlier classes, the training samples taken from the target class are mapped into a high-dimensional feature space using a kernel function to build the learning dictionary. In He et al. (2016), target detection is performed by a robust low-rank sparse representation. In Xu et al. (2016b), a low-rank representation is used to model the background, with a sparsity-inducing regularizer to characterize the pixel's local representation. In Xu et al. (2016a), a sparse multitemporal dictionary learning method is developed for cloud detection and removal.

Sparse representations and dictionary learning were recently applied on multispectral Landsat images, for the purpose of image superresolution (Song et al., 2015) and domain adaptation (Roy et al., 2015). Sparse representation were applied as well on VHR remote sensing images for object detection (Yokoya and Iwasaki, 2015), scene classification (Chaib et al., 2016) and vehicle detection (Chen et al., 2016c). Another application field is SAR image classification (Hou et al., 2016; Xie et al., 2016).

8. Conclusions and future directions

In this chapter, we have described the contribution of machine learning to remote sensing data analysis. Based on the frequency with which machine learning methods have appeared in the recent relevant literature, we followed a taxonomy comprising 6 groups of methods: kernel learning methods, neural network learning methods, manifold learning methods, structured output learning methods, ensemble learning methods and sparse learning methods. The large majority of the state of the art deals with the problem of classification of remote sensing data.

The first conclusion that can be drawn is that many new methods are being developed for each of the covered groups of machine learning. On the one hand, new developments from the general machine learning community are picked up quickly and adapted for the specific purpose of remote sensing data analysis. Many of these methods deal with the problem of image classification, and focus on specific properties of remote sensing images, such as small training sizes and high spectral, spatial and temporal variability. On the other hand, dedicated methods are being developed exclusively for remote sensing data analysis as well. Examples are methods for spectral-spatial classification, target and anomaly detection, change detection and scene classification. Finally, specific remote sensing applications are being treated using more mature methodologies (e.g. SVM's, kernel methods and ensemble learning methods).

The recent trends are expected to continue for the coming years. A recent new development is the combination of ideas from different machine learning groups. Examples are the use of more involved classifiers in ensemble learning or neural network methods or the inclusion of sparsity in manifold learning methods or the integration of kernel

methods and probabilistic graphical models. In this way, the advantages of different groups of machine learning methods can be combined. Most likely, these ideas will be further developed in the near future. Models implying multimodal data, typically multi-view, multi-scale and multi-resolution methods, will become more and more the norm, to answer the needs of analysts that can now access a wide variety of sensor data. Also, because of the progress that has been and is being made on classification by the newly developed methodologies and by their recent inclusion in commercial software, these will find their way in different application areas.

References

Abercrombie, S., Friedl, M., Feb 2016. Improving the consistency of multitemporal land cover maps using a hidden markov model. IEEE Trans. Geosci. Remote Sens. 54 (2), 703–713.

Aharon, A., Elad, M., Bruckstein, A., 2006. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. 54 (11), 4311–4322.

Alonso-Montesinos, J., Martnez-Durbn, M., del Sagrado, J., del guila, I., Batlles, F., 2016. The application of Bayesian network classifiers to cloud classification in satellite images. Renewable Energy 97, 155–161.

Atkinson, P. M., Pardo-Iguzquiza, E., Chica-Olmo, M., Febr. 2008. Downscaling cokriging for super-resolution mapping of continua in remotely sensed images. IEEE Trans. Geosci. Remote Sens. 46 (2), 573–580.

Ayerdi, B., Romay, M. G., May 2016. Hyperspectral image analysis by spectral-spatial processing and anticipative hybrid extreme rotation forest classification. IEEE Trans. Geosci. Remote Sens. 54 (5), 2627–2639.

Bachmann, C., Ainsworth, T., Fusina, R., March 2005. Exploiting manifold geometry in hyperspectral imagery. IEEE Trans. Geosci. Remote Sens. 43 (3), 441–454.

Bachmann, C. M., Ainsworth, T. L., Fusina, R. A., Montes, M. J., Bowles, J. H., Korwan, D. R., Gillis, D. B., March 2009. Bathymetric Retrieval From Hyperspectral Imagery Using Manifold Coordinate Representations. IEEE Trans. Geosci. Remote Sens. 47 (3), 884–897.

Bai, X., Guo, Z., Wang, Y., Zhang, Z., Zhou, J., June 2015. Semisupervised hyperspectral band selection via spectral-spatial hypergraph model. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (6), 2774–2783.

Bao, R., Xia, J., Mura, M. D., Du, P., Chanussot, J., Ren, J., March 2016. Combining morphological attribute profiles via an ensemble method for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 13 (3), 359–363.

Baret, F., Hagolle, O., Geiger, B., Bicheron, P., Miras, B., Huc, M., Berthelot, B., Niño, F., Weiss, M., Samain, O., Roujean, J., Leroy, M., 2007. LAI, fAPAR and fCover CYCLOPES global products derived from VEGETATION. Part 1: Principles of the algorithm. Remote Sensing of the Environment 110 (3), 275–286.

Belkin, M., Niyogi, P., 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. Adv. Neural. Inf. Process. Syst. 14, 586–691.

Belkin, M., Niyogi, P., Sindhwani, V., 2006. Manifold regularization: a geometric framework for learning form labeled and unlabeled samples. J. Mach. Learn. Res. 7, 2399–2434.

Bencherif, M., Bazi, Y., Guessoum, A., Alajlan, N., Melgani, F., AlHichri, H., March 2015. Fusion of extreme learning machine and graph-based optimization methods for active classification of remote sensing images. IEEE Geosci. Remote Sens. Lett. 12 (3), 527–531.

Benedek, C., Martorella, M., 2014. Moving target analysis in ISAR image sequences with a multiframe marked point process model. IEEE Trans. Geosci. Remote Sens. 52 (4), 2234–2246.

Benedek, C., Shadaydeh, M., Kato, Z., Szirnyi, T., Zerubia, J., 2015. Multilayer Markov random field models for change detection in optical remote sensing images. ISPRS J. Photogramm. Remote Sens. 107, 22–37.

Benediktsson, J., Palmason, J. A., Sveinsson, J. R., 2005. Classification of hyperspectral data from urban areas based on extended morphological profiles. IEEE Trans. Geosci. Remote Sens. 43 (3), 480–490.

Benediktsson, J., Pesaresi, M., Arnason, K., Sep. 2003. Classification and feature extraction for remote sensing images from urban areas based on morphological transformations. IEEE Trans. Geosci. Remote Sens. 41 (9, 1), 1940–1949.

Bengio, Y., Paiement, J., Vincent, P., Delalleau, O., Le Roux, N., Ouimet, M., 2004. Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering. In: Thrun, S and Saul, K and Scholkopf, B (Ed.), Adv. Neural. Inf. Process. Syst. 16. Vol. 16 of Adv. Neural. Inf. Process. Syst. pp. 177–184, 17th Annual Conference on Neural Information Processing Systems (NIPS), Canada, Dec. 08, 2003.

Besag, J., 1993. Statistical analysis of dirty pictures. J. Applied Stat. 20 (5-6), 63–87.

Bhatt, J., Joshi, M., Raval, M., 2014. A data-driven stochastic approach for unmixing hyperspectral imagery. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 7 (6), 1936–1946.

Bioucas-Dias, J., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N., Chanussot, J., June 2013. Hyperspectral remote sensing data analysis and future challenges. IEEE Geosci. and Remote Sens. Mag. 1 (2), 6–36.

Bioucas-Dias, J. M., Plaza, A., Dobigeon, N., Parente, M., Du, Q., Gader, P., Chanussot, J., 2012. Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 5 (2, SI), 354–379.

Bishop, C., 2006. Pattern recognition and machine learning. New York, NY, Springer-Verlag.

Blackwell, W. J., 2011. Hyperspectral microwave atmospheric sounding using neural networks. In: Chen, C. C. (Ed.), Signal and image processing for remote sensing. CRC press.

Blackwell, W. J., Milstein, A., 2014. Aneural network retrieval technique for high resolution profiling of cloudy atmospheres. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 7 (4), 1260–1271.

Blaschke, T., Lang, S., Hay, G., 2008. Object-Based Image Analysis. Springer.

Bo, C., Lu, H., Wang, D., Feb 2016. Hyperspectral image classification via jcr and svm models with decision fusion. IEEE Geosci. Remote Sens. Lett. 13 (2), 177–181.

Bohning, D., 1992. Multinomial logistic regression algorithm. Annals of the Institute of Statistical Mathematics 44 (1), 197–200.

Borcs, A., Benedek, C., 2014. Extraction of vehicle groups in airborne lidar point clouds with two-level point processes. IEEE Trans. Geosci. Remote Sens. 53 (3), 1475–1489.
Boser, B., Guyon, I., Vapnik, V., 1992. A training algorithm for optimal margin classifiers. In: 5th ACM Workshop on Computational Learning Theory. Pittsburgh, USA, pp. 144–152.
Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Machine Intell. 23 (11), 1222–1239.
Bray, M., Link, C., Feb 2015. Learning Machine Identification of Ferromagnetic UXO Using Magnetometry. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (2), 835–844.
Breiman, L., 1996. Bagging predictors. Mach. Learn. 24, 123–140.
Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.
Bruzzone, L., Bovolo, F., March 2013. A Novel Framework for the Design of Change-Detection Systems for Very-High-Resolution Remote Sensing Images. P. IEEE 101 (3, SI), 609–630.
Bruzzone, L., Chi, M., Marconcini, M., Nov. 2006. A novel transductive SVM for semisupervised classification of remote-sensing images. IEEE Trans. Geosci. Remote Sens. 44 (11, 2), 3363–3373.
Bruzzone, L., Marconcini, M., 2010. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. IEEE Trans. Pattern Anal. Mach. Intell. 32 (5), 770–787.
Campos-Taberner, M., Romero-Soriano, A., Gatta, C., Camps-Valls, G., Lagrange, A., Saux, B. L., Beaupère, A., Boulch, A., Chan-Hon-Tong, A., Herbin, S., Randrianarivo, H., Ferecatu, M., Shimoni, M., Moser, G., Tuia, D., in press. Processing of extremely high resolution LiDAR and optical data: Outcome of the 2015 IEEE GRSS Data Fusion Contest. Part A: 2D contest. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.
Camps-Valls, G., Bandos Marsheva, T., Zhou, D., 2007. Semi-supervised graph-based hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 45 (10), 3044–3054.
Camps-Valls, G., Bruzzone, L., 2005. Kernel-based methods for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 43 (6), 1351–1362.
Camps-Valls, G., Bruzzone, L., 2009. Kernel Methods for Remote Sensing Data Analysis. J. Wiley & Sons, NJ, USA.
Camps-Valls, G., Gómez-Chova, L., Muñoz-Marí, J., Rojo-Álvarez, J. L., Martínez-Ramón, M., 2008a. Kernel-based framework for multi-temporal and multi-source remote sensing data classification and change detection. IEEE Trans. Geosci. Remote Sens. 46 (6), 1822–1835.
Camps-Valls, G., Gómez-Chova, L., Muñoz-Marí, J., Vila-Francés, J., Calpe-Maravilla, J., 2006. Composite kernels for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 3 (1), 93–97.
Camps-Valls, G., Muñoz Marí, J., Gómez-Chova, L., Guanter, L., Calbet, X., 2011. Nonlinear statistical retrieval of atmospheric profiles from MetOp-IASI and MTG-IRS infrared sounding data. IEEE Transactions on Geoscience and Remote Sensing 49.
Camps-Valls, G., Muñoz-Marí, J., Gómez-Chova, L., Richter, K., Calpe-Maravilla, J., 2008b. Biophysical parameter estimation with a semi-supervised support vector machine. IEEE Geoscience and Remote Sensing Letters 6 (2), 248–252.
Camps-Valls, G., Shervashidze, N., Borgwardt, K., 2010. Spatio-spectral remote sensing image classification with graph kernels. IEEE Geosci. Remote Sens. Lett. 7 (4), 741–745.
Camps-Valls, G., Tuia, D., Bruzzone, L., Benediktsson, J. A., 2014. Advances in hyperspectral image classification. IEEE Signal Proc. Mag. 31, 45–54.
Castelluccio, M. P. G., Sansone, C., Verdoliva, L., 2015. Land use classification in remote sensing images by convolutional neural networks. arXiv 1508, 1–11.
Chaib, S., Gu, Y., Yao, H., Feb 2016. An informative feature selection method based on sparse pca for vhr scene classification. IEEE Geosci. Remote Sens. Lett. 13 (2), 147–151.
Chapel, L., Burger, T., Courty, N., Lefevre, S., Apr. 2014. PerTurbo Manifold Learning Algorithm for Weakly Labeled Hyperspectral Image Classification. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 7 (4), 1070–1078.
Chen, C., Chen, N., Peng, J., March 2016a. Nearest regularized joint sparse representation for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 13 (3), 424–428.
Chen, H., Hua, Y., Ren, Q., Zhang, Y., 2016b. Comprehensive analysis of regional human-driven environmental change with multitemporal remote sensing images using observed object-specified dynamic Bayesian network. J. Applied Remote Sens. 10 (1).
Chen, Y., Nasrabadi, N., Tran, T., 2011. Hyperspectral image classification using dictionary-based sparse representation. IEEE Trans. Geosci. Remote Sens. 49 (10), 3973–3985.
Chen, Z., Wang, C., Wen, C., Teng, X., Chen, Y., Guan, H., Luo, H., Cao, L., Li, J., Jan 2016c. Vehicle detection in high-resolution aerial images via sparse representation and superpixels. IEEE Trans. Geosci. Remote Sens. 54 (1), 103–116.
Cheng, Q., Shen, H., Zhang, L., Yuan, Q., Zeng, C., 2014. Cloud removal for remotely sensed images by similar pixel replacement guided with a spatio-temporal MRF model. ISPRS J. Photogramm. and Remote Sens. 92, 54–68.
Cover, T. M., 1965. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Transactions on Electronics and Computers EC-14 (3), 326–334.
Craciun, P., Zerubia, J., 2014. Towards efficient simulation of marked point process models for boat extraction from high resolution optical remotely sensed images. pp. 2297–2300.
Crawford, M. M., Tuia, D., Hyang, L. H., 2013. Active learning: Any value for classification of remotely sensed data? Proceedings of the IEEE 101 (3), 593–608.
Cui, M., Prasad, S., May 2015. Class-dependent sparse representation classifier for robust hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 53 (5), 2683–2695.
Cusano, C., Napoletano, P., Schettini, R., 2015. Remote sensing image classification exploiting multiple kernel learning. IEEE Geosci. Remote Sens. Lett. 12 (11), 2331–2335.
Cutler, M., Boyd, D., Foody, G., Vetrivel, A., 2012. Estimating tropical forest biomass with a combination of SAR image texture and Landsat TM data: An assessment of predictions between regions. ISPRS Journal of Photogrammetry and Remote Sensing 70, 66–77.
D'Addabbo, A., Refice, A., Pasquariello, G., Lovergine, F., Capolongo, D., Manfreda, S., 2016. A Bayesian network for flood detection combining

SAR imagery and ancillary data. IEEE Trans. Geosci. Remote Sens. 54 (6), 3612–3625.
Damodaran, B., Nidamanuri, R., Tarabalka, Y., June 2015. Dynamic ensemble selection approach for hyperspectral image classification with joint spectral and spatial information. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (6), 2405–2417.
Datcu, M., Seidel, K., Walessa, M., Sep. 1998. Spatial information retrieval from remote-sensing images - Part 1: Information theoretical perspective. IEEE Trans. Geosci. Remote Sens. 36 (5, 1), 1431–1445.
De Giorgi, A., Moser, G., Serpico, S., 2014. Contextual remote-sensing image classification through support vector machines, Markov random fields and graph cuts. pp. 3722–3725.
De Giorgi, A., Moser, G., Serpico, S., 2015. Parameter optimization for Markov random field models for remote sensing image classification through sequential minimal optimization. Vol. 2015-November. pp. 2346–2349.
de Morsier, F., Borgeaud, M., Gass, V., Thiran, J. P., Tuia, D., in press. Kernel low-rank and sparse graph for unsupervised and semi-supervised classification of hyperspectral images. IEEE Trans. Geosci. Remote Sens. PP (99), 1–11.
Demir, B., Minello, L., Bruzzone, L., 2014. Definition of effective training sets for supervised classification of remote sensing images by a novel cost-sensitive active learning method. IEEE Trans. Geosci. Remote Sens. 52 (2), 1272–1284.
Demir, B., Persello, C., Bruzzone, L., 2011. Batch mode active learning methods for the interactive classification of remote sensing images. IEEE Trans. Geosci. Remote Sens. 49 (3), 1014–1031.
Denis, L., Tupin, F., Darbon, J., Sigelle, M., 2009. SAR image regularization with fast approximate discrete minimization. IEEE Trans. Image Process. 18 (7), 1588–1600.
Derin, H., Kelly, P., 1989. Discrete-index Markov-type random processes. Proc. IEEE 77 (10), 1485–1510.
Descombes, X., Zerubia, J., 2002. Marked point process in image analysis. IEEE Signal Process. Mag. 19 (5), 77–84.
Dijkstra, E., 1959. A note on two problems in connexion with graphs. Num. Math. 1, 269–271.
Djuric, N., Radosavljevic, V., Obradovic, Z., Vucetic, S., 2015. Gaussian conditional random fields for aggregation of operational aerosol retrievals. IEEE Geosci. Remote Sens. Lett. 12 (4), 761–765.
Dong, Y., Du, B., Zhang, L., Apr. 2015. Target detection based on random forest metric learning. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (4), 1830–1838.
Donoho, D., 2006. Compressed sensing. IEEE Trans. Inf. Theory 54 (4), 1289–1306.
Dubes, R., Jain, A., 1989. Random field models in image analysis. J. Appl. Stat. 16 (2), 131–164.
Dumoulin, V., Visin, F., 2016. A guide to convolution arithmetic for deep learning.
URL `http://arxiv.org/abs/1603.07285`
Ebtehaj, A., Bras, R., Foufoula-Georgiou, E., July 2015. Shrunken locally linear embedding for passive microwave retrieval of precipitation. IEEE Trans. Geosci. Remote Sens. 53 (7), 3720–3736.
Engan, K., Aase, S., Husoy, J., 1999. Method of optimal directions for frame design. In: ICASSP '99: 1999 IEEE Iinternational Conference on Acoustics, Speech and Signal Processing, Proceedings Vols I-VI. International Conference on Acoustics Speech and Signal Processing ICASSP. IEEE; IEEE Signal Proc Soc, pp. 2443–2446, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 99), PHOENIX, AZ, MAR 15-19, 1999.
Englhart, S., Keuck, V., Siegert, F., 2012. Modeling aboveground biomass in tropical forests using multi-frequency SAR data - A comparison of methods. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 5 (1), 298–306.
Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., 2010. Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11, 625–660.
Fang, H., Liang, S., 2003. Retrieving leaf area index with a neural network method: simlation and validation. IEEE Trans. Geosci. Remote Sens. 41 (9), 2052–2063.
Farnandez-Delgado, M., Cernadas, E., Barro, S., Amorim, D., 2014. Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. 15, 3133–3181.
Fauvel, M., Chanussot, J., Benediktsson, J. A., Oct. 2006. Decision fusion for the classification of urban remote sensing images. IEEE Trans. Geosci. Remote Sens. 44 (10, 1), 2828–2838.
Fauvel, M., Chanussot, J., Benediktsson, J. A., 2009. Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas. EURASIP Journal on Advances in Signal Processing 11.
Fauvel, M., Tarabalka, Y., Benediktsson, J. A., Chanussot, J., Tilton, J. C., 2013. Advances in Spectral-Spatial Classification of Hyperspectral Images. P. IEEE 101 (3, SI), 652–675.
Fei, T., Kraus, D., Zoubir, A., Jan 2015. Contributions to automatic target recognition systems for underwater mine classification. IEEE Trans. Geosci. Remote Sens. 53 (1), 505–518.
Felzenszwalb, P., Huttenlocher, D., 2004. Efficient graph-based image segmentation. Int. J. Comp. Vision 59 (2), 167–181.
Feret, J.-B., Asner, G., 2013. Tree species discrimination in tropical forests using airborne imaging spectroscopy. IEEE Trans. Geosci. Remote Sens. 51 (1), 73–84.
Freund, Y., Schapire, R., 1999. A short introduction to boosting. J. Jpn. Soc. Artif. Intell. 14 (5), 771–780.
Fu, W., Li, S., Fang, L., Kang, X., Benediktsson, J., Feb 2016. Hyperspectral image classification via shape-adaptive joint sparse representation. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 9 (2), 556–567.
Fukunaga, K., 1990. Introduction to statistical pattern recognition. Boston, MA, Academic.
Fukunaga, K., Hostetler, L., 1975. The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans. Inform. Theory 21 (1), 32–40.
Gaetano, R., Amitrano, D., Masi, G., Poggi, G., Ruello, G., Verdoliva, L., Scarpa, G., 2014. Exploration of multitemporal COSMO-Skymed data via interactive tree-structured MRF segmentation. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 7 (7), 2763–2775.
Gamba, P., Dell'Acqua, F., Lisini, G., Trianni, G., 2007. Improved VHR urban area mapping exploiting object boundaries. IEEE Trans. Geosci. Remote Sens. 45 (8), 2676–2682.
Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Machine Intell. PAMI-6 (6), 721–741.

Gerke, M., Xiao, J., 2014. Fusion of airborne laserscanning point clouds and images for supervised and unsupervised scene classification. ISPRS J. Photogr. Remote Sens. 87, 78–92.
Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Mach. Learn. 63 (1), 3–42.
Ghamisi, P., Benediktsson, J., Ulfarsson, M., 2014. Spectral-spatial classification of hyperspectral images based on hidden Markov random fields. IEEE Trans. Geosci. Remote Sens. 52 (5), 2565–2574.
Ghamisi, P., Dalla Mura, M., Benediktsson, J., 2015. A survey on spectral-spatial classification techniques based on attribute profiles. IEEE Trans. Geosci. Remote Sens. 53 (5), 2335–2353.
Gimenez, J., Frery, A., Flesia, A., 2015. When data do not bring information: A case study in Markov random fields estimation. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (1), 195–203.
Gokon, H., Post, J., Stein, E., Martinis, S., Twele, A., Muck, M., Geiss, C., Koshimura, S., Matsuoka, M., June 2015. A method for detecting buildings destroyed by the 2011 tohoku earthquake and tsunami using multitemporal terrasar-x data. IEEE Geosci. Remote Sens. Lett. 12 (6), 1277–1281.
Golipour, M., Ghassemian, H., Mirzapour, F., Feb 2016. Integrating hierarchical segmentation maps with mrf prior for classification of hyperspectral images in a bayesian framework. IEEE Trans. Geosci. Remote Sens. 54 (2), 805–816.
Gómez-Chova, L., Camps-Valls, G., Bruzzone, L., Calpe-Maravilla, J., 2010. Mean map kernel methods for semisupervised cloud classification. IEEE Trans. Geosci. Remote Sens. 48 (1), 207–220.
Gómez-Chova, L., Camps-Valls, G., Muñoz-Marí, J., Calpe, J., 2008. Semi-supervised image classification with Laplacian support vector machines. IEEE Geosci. Remote Sens. Lett. 5 (4), 336–340.
Gomez-Chova, L., Jenssen, R., Camps-Valls, G., 2012. Kernel entropy component analysis for remote sensing image clustering. IEEE Geosci. Remote Sens. Lett. 9 (2), 312–316.
Gomez-Chova, L., Tuia, D., Moser, G., Camps-Valls, G., Sept 2015. Multimodal classification of remote sensing images: A review and future directions. P. IEEE 103 (9), 1560–1584.
Grady, L., 2006. Random walks for image segmentation. IEEE Trans. Pattern Anal. Machine Intell. 28 (11), 1768–1783.
Greig, D. M., Porteous, B. T., Seheult, A. H., 1989. Exact maximum a posteriori estimation for binary images. J. Royal Stat. Soc. Series B (Method.) 51 (2), 271–279.
Gu, J., Jiao, L., Yang, S., Liu, F., Hou, B., Zhao, Z., 2016. A multi-kernel joint sparse graph for sar image segmentation. IEEE J. Sel. Topics Appl. Earth Observ. 9 (3), 1265–1285.
Gu, Y., Wang, C., You, D., Zhang, Y., Wang, S., Zhang, Y., 2012. Representative multiple-kernel learning for classification of hyperspectral imagery. IEEE Trans. Geosci. Remote Sens. 7 (50), 2852–2865.
Gu, Y., Wang, S., Jia, X., 2013. Spectral unmixing in multiple-kernel hilbert space for hyperspectral imagery. IEEE Trans. Geosci. Remote Sens. 51 (7), 3968–3981.
Gu, Y., Zhang, Y., Zhang, J., May 2008. Integration of spatial-spectral information for resolution enhancement in hyperspectral images. IEEE Trans. Geosci. Remote Sens. 46 (5), 1347–1358.
Guo, Z., Du, S., Zhao, W., 2015. Using random walker for knowledge transfer in classifying multi-temporal VHR images. Int. J. Remote Sens. 36 (17), 4332–4343.
Gurram, P., Kwon, H., 2013. Contextual svm using hilbert space embedding for hyperspectral classification. IEEE Geosci. Remote Sens. Lett. 10 (5), 1031–1035.
Ham, J., Chen, Y., Crawford, M., Ghosh, J., 2005a. Investigation of the random forest framework for classification of hyperspectral data. IEEE Trans. Geosci. Remote Sens. 43 (3), 492–501.
Ham, J., Lee, D. D., Saul, L. K., 2005b. Semisupervised alignment of manifolds. In: AISTATS.
Hang, R., Liu, Q., Song, H., Sun, Y., Feb 2016. Matrix-based discriminant subspace ensemble for hyperspectral image spatial-spectral feature fusion. IEEE Trans. Geosci. Remote Sens. 54 (2), 783–794.
Haykin, S., 1999. Neural Networks. Prentice-Hall.
He, L., Li, Y., Li, X., Wu, W., May 2015. Spectral-spatial classification of hyperspectral images via spatial translation-invariant wavelet-based sparse representation. IEEE Trans. Geosci. Remote Sens. 53 (5), 2696–2712.
He, Y., Li, M., Zhang, J., Yao, J., Feb 2016. Infrared target tracking based on robust low-rank sparse learning. IEEE Geosci. Remote Sens. Lett. 13 (2), 232–236.
Hedhli, I., Moser, G., Serpico, S., Zerubia, J., 2015. New hierarchical joint classification method for SAR-optical multiresolution remote sensing data. pp. 759–763.
Hedhli, I., Moser, G., Serpico, S., Zerubia, J., 2016. A new cascade model for the hierarchical joint classification of multitemporal and multiresolution remote sensing data. IEEE Trans. Geosci. Remote Sens. (in print).
Heylen, R., Burazerovic, D., Scheunders, P., Jun. 2011. Non-Linear Spectral Unmixing by Geodesic Simplex Volume Maximization. IEEE J. Sel. Topics Signal Proc. 5 (3), 534–542.
Heylen, R., Parente, M., Gader, P., 2014. A review of nonlinear hyperspectral unmixing methods. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 7 (6), 1844–1868.
Hinton, G. E., 1989. Connectionist learning procedures. Artificial Intelligence 40, 185–234.
Ho, T., 1998. The random subspace method for constructing decision forests. IEEE Trans. Patt. Anal. Mach. Intell. 20 (8), 832–844.
Hoberg, T., Rottensteiner, F., Queiroz Feitosa, R., Heipke, C., 2015. Conditional random fields for multitemporal and multiscale classification of optical satellite imagery. IEEE Trans. Geosci. Remote Sens. 53, 659–673.
URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6841049
Hou, B., Ren, B., Ju, G., Li, H., Jiao, L., Zhao, J., Jan 2016. Sar image classification via hierarchical sparse representation and multisize patch features. IEEE Geosci. Remote Sens. Lett. 13 (1), 33–37.
Huang, H., Yang, M., Sept 2015. Dimensionality reduction of hyperspectral images with sparse discriminant embedding. IEEE Trans. Geosci. Remote Sens. 53 (9), 5160–5169.
Huang, H.-B., Huo, H., Fang, T., March 2014. Hierarchical Manifold Learning With Applications to Supervised Classification for High-Resolution

Remotely Sensed Images. IEEE Trans. Geosci. Remote Sens. 52 (3), 1677–1692.
Huang, X., Qiao, H., Zhang, B., Feb 2016. Sar target configuration recognition using tensor global and local discriminant embedding. IEEE Geosci. Remote Sens. Lett. 13 (2), 222–226.
Huang, X., Xie, C., Fang, X., Zhang, L., May 2015. Combining pixel- and object-based machine learning for identification of water-body types from urban high-resolution remote-sensing imagery. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (5), 2097–2110.
Huang, X., Zhang, L., 2013. An svm ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery. IEEE Trans. Geosci. Remote Sens. 51 (1), 257–272.
Huo, L. Z., Tang, P., Zhang, Z., Tuia, D., 2015. Semisupervised classification of remote sensing images with hierarchical spatial similarity. IEEE Geosci. Remote Sens. Letters 12 (1), 150–154.
Ibanez, M., Simo, A., 2003. Parameter estimation in Markov random field image modeling with imperfect observations. a comparative study. Patt. Recogn. Lett. 24 (14), 2377–2389.
Ihler, A., Fisher III, J., Willsky, A., 2005. Loopy belief propagation: Convergence and effects of message errors. J. Machine Learning Res. 6.
Inglada, J., AUG 2007. Automatic recognition of man-made objects in high resolution optical remote sensing images by SVM classification of geometric image features. ISPRS J. Photogramm. Remote Sens. 62 (3), 236–248.
Insom, P., Cao, C., Boonsrimuang, P., Liu, D., Saokarn, A., Yomwan, P., Xu, Y., Sept 2015. A support vector machine-based particle filter method for improved flooding classification. IEEE Geosci. Remote Sens. Lett. 12 (9), 1943–1947.
Ioannou, I., Gilerson, A., Gross, B., Moshary, F., Ahmed, S., 2013. Deriving ocean color products using neural networks. Remote Sensing of the Environment 134, 78–91.
Iordache, M., Bioucas-Dias, J., Plaza, A., 2011. Sparse unmixing of hyperspectral data. IEEE Trans. Geosci. Remote Sens. 49 (6), 2014–2039.
Izquierdo-Verdiguier, E., Gmez-Chova, L., Bruzzone, L., Camps-Valls, G., 2014. Semisupervised kernel feature extraction for remote sensing image analysis. IEEE Transactions on Geoscience and Remote Sensing 52 (9), 5567–5578.
Izquierdo-Verdiguier, E., Laparra, V., Gmez-Chova, L., Camps-Valls, G., Sept 2013. Encoding invariances in remote sensing image classification with svm. IEEE Geosci. Remote Sens. Lett. 10 (5), 981–985.
Jackson, Q., Landgrebe, D., 2002. Adaptive Bayesian contextual classification based on Markov random fields. IEEE Trans. Geosci. Remote Sens. 40 (11), 2454–2463.
Jia, S., Shen, L., Li, Q., Feb 2015a. Gabor feature-based collaborative representation for hyperspectral imagery classification. IEEE Trans. Geosci. Remote Sens. 53 (2), 1118–1129.
Jia, S., Zhang, X., Li, Q., June 2015b. Spectral -spatial hyperspectral image classification using $\ell_{1/2}$ regularized low-rank representation and sparse representation-based graph cuts. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (6), 2473–2484.
Jian, P., Chen, K., Zhang, C., 2016. A hypergraph-based context-sensitive representation technique for VHR remote-sensing image change detection. Int. J. Remote Sens. 37 (8), 1814–1825.
Kang, X., Li, S., Fang, L., Li, M., Benediktsson, J., 2015. Extended random walker-based classification of hyperspectral images. IEEE Trans. Geosci. Remote Sens. 53, 144–153.
URL `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6815639`
Karantzalos, K., Sotiras, A., Paragios, N., 2014. Efficient and automated multimodal satellite data registration through MRFs and linear programming. pp. 335–342.
Kato, Z., Zerubia, J., 2011. Markov random fields in image segmentation. Foundations and Trends in Signal Process. 5 (1-2), 1–155.
Khatami, R., Mountrakis, G., Stehman, S., 2016. A meta-analysis of remote sensing research on supervised pixel-based land-cover image classification processes: General guidelines for practitioners and future research. Remote Sensing of the Environment 177, 89–100.
Khodadadzadeh, M., Li, J., Plaza, A., Ghassemian, H., Bioucas-Dias, J., Li, X., 2014. Spectral-spatial classification of hyperspectral data using local and global probabilities for mixed pixel characterization. IEEE Trans. Geosci. Remote Sens. 52 (10), 6298–6314.
Kim, W., Crawford, M., 2010. Adaptive classification for hyperspectral image data using manifold regularization kernel machines. IEEE Trans. Geosci. Remote Sens. 48 (11), 4110–4121.
Klesk, P., Godziuk, A., KApr.uziak, M., Olech, B., July 2015. Fast analysis of c-scans from ground penetrating radar via 3-d haar-like features with application to landmine detection. IEEE Trans. Geosci. Remote Sens. 53 (7), 3996–4009.
Koller, D., Friedman, N., 2009. Probabilistic Graphical Models. MIT Press.
Kolmogorov, V., 2006. Convergent tree-reweighted message passing for energy minimization. IEEE Trans. Pattern Anal. Machine Intell. 28 (10), 1568–1583.
Kopparapu, S. K., Desai, U. B., 2001. Bayesian Approach to Image Interpretation. Springer.
Krizhevsky, A., Sutskever, I., Hinton, G., 2012. ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems.
Kurtz, C., Passat, N., Gancarski, P., Puissant, A., 2012. Extraction of complex patterns from multiresolution remote sensing images: A hierarchical top-down methodology. Pattern Recogn. 45 (2), 685–706.
Kwon, H., Nasrabadi, N., 2007. A comparative analysis of kernel subspace target detectors for hyperspectral imagery. EURASIP Journal of of Advances in Signal Proc. 2007 (29250).
Laferte, J.-M., Perez, P., Heitz, F., 2000. Discrete Markov image modeling and inference on the quadtree. IEEE Trans. Image Process. 9 (3), 390–404.
Landgrebe, D. A., 2003. Signal theory methods in multispectral remote sensing. Wiley.
LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE.
Leiva-Murillo, J., Gomez-Chova, L., Camps-Valls, G., jan. 2013. Multitask remote sensing data classification. IEEE Trans. Geosci. Remote Sens. 51 (1), 151 –161.
Li, F., Xu, L., Siva, P., Wong, A., Clausi, D., June 2015a. Hyperspectral image classification with limited labeled training samples using enhanced ensemble learning and conditional random fields. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (6), 2427–2438.
Li, J., Marpu, P. R., Plaza, A., Bioucas-Dias, J., Benediktsson, J. A., in press. Generalized composite kernel framework for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens.

Li, J., Zhang, H., Zhang, L., Oct 2015b. Efficient superpixel-level multitask joint sparse representation for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 53 (10), 5338–5351.
Li, S., 1995. On discontinuity-adaptive smoothness priors in computer vision. IEEE Trans. Pattern Anal. Machine Intell. 17 (6), 576–586.
Li, W., Du, Q., 2013. Joint within-class collaborative representation for hyperspectral image classification. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 52 (6), 3399–3411.
Li, W., Du, Q., Xiong, M., 2015c. Kernel collaborative representation with tikhonov regularization for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 12 (1), 48–52.
Li, W., Du, Q., Zhang, F., Hu, W., Feb 2015d. Collaborative-representation-based nearest neighbor classifier for hyperspectral imagery. IEEE Geosci. Remote Sens. Lett. 12 (2), 389–393.
Li, W., Tramel, W., Prasad, S., Fowler, J., 2014a. Nearest regularized subspace for hyperspectral classification. IEEE Trans. Geosci. Remote Sens. 52 (1), 477–489.
Li, W., Zhang, C., Dey, D., Willig, M., 2013. Updating categorical soil maps using limited survey data by Bayesian Markov chain cosimulation. The Scientific World Journal 2013.
Li, W., Zhang, C., Willig, M., Dey, D., Wang, G., You, L., 2014b. Bayesian Markov chain random field cosimulation for improving land cover classification accuracy. Math. Geosci. 47 (2), 123–148.
Li, Y., Tan, Y., Deng, J., Wen, Q., Tian, J., 2015e. Cauchy graph embedding optimization for built-up areas detection from high-resolution remote sensing images. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (5), 2078–2096.
Li, Z., Shi, W., Shi, X., Zhong, Z., Dec. 2009. A Supervised Manifold Learning Method. Comput. Sc. Inf. Syst. 6 (2), 205–215.
Lian, X., Wu, Y., Zhao, W., Wang, F., Zhang, Q., Li, M., 2014. Unsupervised SAR image segmentation based on conditional triplet Markov fields. IEEE Geosci. Remote Sens. Lett. 11 (7), 1185–1189.
Liu, G., Zhao, Z., Zhang, Y., 2015a. Image fuzzy clustering based on the region-level Markov random field model. IEEE Geosci. Remote Sens. Lett. 12 (8), 1770–1774.
Liu, H., Wang, Y., Yang, S., Wang, S., Feng, J., Jiao, L., Apr. 2016a. Large polarimetric sar data semi-supervised classification with spatial-anchor graph. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 9 (4), 1439–1458.
Liu, J., Wu, Z., Li, J., Plaza, A., Yuan, Y., 2016b. Probabilistic-kernel collaborative representation for spatial-spectral hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 54 (4), 2371–2384.
Liu, J., Wu, Z., Wei, Z., Xiao, L., Sun, L., 2013. Spatial-spectral kernel sparse representation for hyperspectral image classification. IEEE J. Sel. Topics Appl. Earth Observ. 6 (6), 2462–2471.
Liu, K. H., Lin, Y. Y., Chen, C. S., 2015b. Linear spectral mixture analysis via multiple-kernel learning for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 53 (4), 2254–2269.
Lu, Q., Huang, X., Li, J., Zhang, L., 2016. A novel MRF-based multifeature fusion for classification of remote sensing images. IEEE Geosci. Remote Sens. Lett. 13 (4), 515–519.
Lunga, D., Prasad, S., Crawford, M., Ersoy, O., Jan 2014. Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning. IEEE Signal Proc. Mag. 31 (1), 55–66.
Ma, L., Crawford, M., Yang, X., Guo, Y., May 2015. Local-manifold-learning-based graph construction for semisupervised hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 53 (5), 2832–2844.
Ma, L., Crawford, M. M., Tian, J., Nov. 2010a. Local Manifold Learning-Based k-Nearest-Neighbor for Hyperspectral Image Classification. IEEE Trans. Geosci. Remote Sens. 48 (11), 4099–4109.
Ma, L., Crawford, M. M., Tian, J. W., Apr. 1 2010b. Generalised supervised local tangent space alignment for hyperspectral image classification. Electron. Lett. 46 (7), 497–U47.
Maillard, P., 2003. Comparing texture analysis methods through classification. Photogramm. Eng. Remote Sens. 69 (4), 357–367.
Mairal, J., Bach, F., Ponce, J., 2012. Task-driven dictionary learning. IEEE Trans. Patt. Anal. Mach. Intell. 34 (4), 791–804.
Mairal, J., Bach, F., Ponce, J., Sapiro, G., 2009. Online dictionary learning for sparse coding. In: ICML'09 Proceedings of the 26th Annual International Conference on Machine Learning. pp. 689–696.
Manandhar, A., Torrione, P., Collins, L., Morton, K., Apr. 2015. Multiple-instance hidden markov model for gpr-based landmine detection. IEEE Trans. Geosci. Remote Sens. 53 (4), 1737–1745.
Manolakis, D., Truslow, E., Pieper, M., Cooley, T., Brueggeman, M., Jan. 2014. Detection Algorithms in Hyperspectral Imaging Systems. IEEE Sign. Process. Mag. 31 (1), 24–33.
Marmanis, D., Datcu, M., Esch, T., Stilla, U., 2015. Deep learning earth observation classification using imagenet pretrained networks. IEEE Geoscience and Remote Sensing Letters 13 (1), 105–109.
Masjedi, A., Javad Valadan Zoej, M., Maghsoudi, Y., Feb 2016. Classification of polarimetric sar images based on modeling contextual information and using texture features. IEEE Trans. Geosci. Remote Sens. 54 (2), 932–943.
Matasci, G., Longbotham, N., Pacifici, F., M., K., Tuia, D., 2015a. Understanding angular effects in VHR imagery and their significance for urban land-cover model portability: a study of two multi-angle in-track image sequences. ISPRS J. Int. Soc. Photo. Remote Sens. 107, 99–111.
Matasci, G., Volpi, M., Kanevski, M., Bruzzone, L., Tuia, D., 2015b. Semisupervised transfer component analysis for domain adaptation in remote sensing image classification. IEEE Trans. Geosci. Remote Sens. 53 (7), 3550–3564.
Melgani, F., Bruzzone, L., 2004. Classification of hyperspectral remote sensing images with support vector machines. IEEE Trans. Geosci. Remote Sens. 42 (8), 1778–1790.
Melgani, F., Serpico, S., 2003. A Markov random field approach to spatio-temporal contextual image classification. IEEE Trans. Geosci. Remote Sens. 41 (11 PART I), 2478–2487.
Merentitis, A., Debes, C., Sept 2015. Many hands make light work - on ensemble learning techniques for data fusion in remote sensing. IEEE Geosci. and Remote Sens. Mag. 3 (3), 86–99.
Merentitis, A., Debes, C., Heremans, R., 2014. Ensemble learning in hyperspectral image classification: Toward selecting a favorable bias-variance tradeoff. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 7 (4), 1089–1102.
Mishne, G., Talmon, R., Cohen, I., May 2015. Graph-based supervised automatic target detection. IEEE Trans. Geosci. Remote Sens. 53 (5),

2738–2754.
Mitraka, Z., Del Frate, F., Carbone, F., in press. Nonlinear spectral unmixing of Landsat imagery for urban surface cover mapping. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.
Moser, G., Angiati, E., Serpico, S., 2011. Multiscale unsupervised change detection on optical images by Markov random fields and wavelets. IEEE Geosci. Remote Sens. Lett. 8 (4), 725–729.
Moser, G., De Giorgi, A., Serpico, S., 2016. Multiresolution supervised classification of panchromatic and multispectral images by Markov random fields and graph cuts. IEEE Trans. Geosci. Remote Sens. (in print).
Moser, G., Serpico, S., 2009. Unsupervised change detection from multichannel SAR data by Markovian data fusion. IEEE Trans. Geosci. Remote Sens. 47 (7), 2114–2128.
Moser, G., Serpico, S., 2014. Kernel-based classification in complex-valued feature spaces for polarimetric SAR data. pp. 1257–1260.
Moser, G., Serpico, S., Benediktsson, J., 2013. Land-cover mapping by Markov modeling of spatial-contextual information in very-high-resolution remote sensing images. Proc. IEEE 101 (3), 631–651.
Moser, G., Serpico, S. B., 2013. Combining support vector machines and Markov random fields in an integrated framework for contextual image classification. IEEE Trans. Geosci. Remote Sens. 51 (5), 2734–2752.
Mountrakis, G., Ima, J., Ogole, C., 2011. Support vector machines in remote sensing: A review. ISPRS J. Photogramm. Rem. Sens. 66 (3), 247–259.
Nair, V., Hinton, G., 2010. Rectified linear units improve restricted boltzmann machines. In: International Conference on Machine Learning.
Nasrabadi, N. M., 2014. Hyperspectral Target Detection. IEEE Sign. Process. Mag. 31 (1), 34–44.
Ni, D., Ma, H., Apr. 2015. Classification of hyperspectral image based on sparse representation in tangent space. IEEE Geosci. Remote Sens. Lett. 12 (4), 786–790.
Nielsen, A. A., 2011. Kernel maximum autocorrelation factor and minimum noise fraction transformations. IEEE Trans. Image Proc. 20 (3), 612–624.
Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. ISPRS J. Photogramm. Remote Sens. 87, 152–165.
Nowozin, S., Lampert, C., 2010. Structured learning and prediction in computer vision. Foundations and Trends in Computer Graphics and Vision 6 (3-4), 185–365.
Paisitkriangkrai, S., Sherrah, J., Janney, P., Van-Den Hengel, A., 2015. Effective semantic pixel labelling with convolutional networks and conditional random fields. In: IEEE/CVF International Conference on Computer Vision and Pattern Recognition Workshops, Earthvision.
Penatti, O., Nogueira, K., dos Santos, J. A., 2015. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In: IEEE/CVF Computer Vision and Pattern Recognition Workshops, Earthvision.
Persello, C., Bruzzone, L., 2016. Kernel-based domain invariant feature selection in hyperspectral images for transfer learning. IEEE Trans. Geosci. Remote Sens. PP (99).
Pieczynski, W., 2007. Multisensor triplet Markov chains and theory of evidence. Int. J. Approx. Reason. 45 (1), 1–16.
Poggi, G., Scarpa, G., Zerubia, J., 2005. Supervised segmentation of remote sensing images based on a tree-structured MRF model. IEEE Trans. Geosci. Remote Sens. 43 (8), 1901–1911.
Pratola, C., Del Frate, F., Schiavon, G., Solimini, D., 2013. Toward fully automatic detection of changes in suburban areas from vhr sar images by combining multiple neural-network models. IEEE Trans. Geosci. Remote Sens. 51 (4), 2055–2066.
Qin, R., 2014. Change detection on LOD 2 building models with very high resolution spaceborne stereo imagery. ISPRS J. Photogramm. Remote Sens. 96, 179–192.
Qin, R., Fang, W., 2014. A hierarchical building detection method for very high resolution remotely sensed images combined with DSM using graph cut optimization. Photogramm. Eng. Remote Sens. 80 (9), 873–883.
Quan, X., He, B., Li, X., Dec 2015. A bayesian network-based method to alleviate the ill-posed inverse problem: A case study on leaf area index and canopy water content retrieval. IEEE Trans. Geosci. Remote Sens. 53 (12), 6507–6517.
Quinlan, J., 1986. Induction of decision trees. Mach. Learn. 1 (1), 81–106.
Rabiner, L., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE 77 (2), 257–286.
Raducanu, B., Dornaika, F., Jun. 2012. A supervised non-linear dimensionality reduction approach for manifold learning. Pattern Recogn. 45 (6), 2432–2444.
Rakotomamonjy, A., Bach, F., Canu, S., Grandvalet, Y., 2008. SimpleMKL. J. Mach. Learn. Res. 9, 2491–2521.
Rallier, G., Descombes, X., Falzon, F., Zerubia, J., 2004. Texture feature analysis using a Gauss-Markov model in hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 42 (7), 1543–1551.
Rejichi, S., Chaabane, F., Tupin, F., May 2015. Expert knowledge-based method for satellite image time series analysis and interpretation. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (5), 2138–2150.
Rodríguez-Fernández, N., Aires, F., Richaume, P., Kerr, Y., Prigent, C., Kolassa, J., Cabot, F., Jiménez, C., Mahmoodi, A., Drusch, M., 2015. Soil moisture retrieval using neural networks: Application to SMOS. IEEE Trans. Geosci. Remote Sens. 53 (11), 5991–6007.
Rokach, L., 2010. Ensemble-based classifiers. Artif. Intell. Rev. 33 (1-2), 1–39.
Romero, A. und Gatta, C., Camps-Valls, G., 2016. Unsupervised deep feature extraction for remote sensing image classification. IEEE Transactions on Geoscience and Remote SensingIn press.
Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J. D., Breitkopf, U., Jung, J., 2014. Results of the isprs benchmark on urban object detection and 3d building reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing 93 (0), 256–271.
Roweis, S., Saul, L., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326.
Roy, M., Melgani, F., Ghosh, A., Blanzieri, E., Ghosh, S., June 2015. Land-cover classification of remotely sensed images using compressive sensing having severe scarcity of labeled patterns. IEEE Geosci. Remote Sens. Lett. 12 (6), 1257–1261.
Rudin, L., Osher, S., Fatemi, E., 1992. Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 60 (1-4), 259–268.
Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986a. Learning internal representations by error propagation. In: Rumelhart, D. E., McCelland, J. L. (Eds.), Parallel distributed processing: explorations in the microstructure of cognition, I: foundations. MIT press.

Rumelhart, D. E., Hinton, G. E., WIlliams, R. J., 1986b. Learning representations of back-propagation errors. Nature 323, 533–536.
Scarpa, G., Gaetano, R., Haindl, M., Zerubia, J., 2009. Hierarchical multiple Markov chain model for unsupervised texture segmentation. IEEE Trans. Image Process. 18 (8), 1830–1843.
Schapire, R., 1990. The strength of weak learnability. Mach. Learn. 5 (2), 197–227.
Schindler, K., 2012. An overview and comparison of smooth labeling methods for land-cover classification. IEEE Trans. Geosci. Remote Sens. 50 (11), 4534–4545.
Schistad Solberg, A., Taxt, T., Jain, A., 1996. A Markov random field model for classification of multisource satellite imagery. IEEE Trans. Geosci. Remote Sens. 34 (1), 100–113.
Schölkopf, B., Smola, A., 2002. Learning with Kernels. MIT press, Cambridge (MA).
Schroder, M., Rehrauer, H., Seidel, K., Datcu, M., Sep. 1998. Spatial information retrieval from remote-sensing images - Part II: Gibbs-Markov random fields. IEEE Trans. Geosci. Remote Sens. 36 (5, 1), 1446–1455.
Serpico, S., Moser, G., 2006. Weight parameter optimization by the Ho-Kashyap algorithm in MRF models for supervised image classification. IEEE Trans. Geosci. Remote Sens. 44 (12), 3695–3705.
Shao, Y., Lunetta, R. S., 2012. Comparison of support vector machine, neural network, and CART algorithms for the land-cover classification using limited training data points. ISPRS Journal of Photogrammetry and Remote Sensing 70, 78–87.
Sherrah, J., 2016. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. arXiv:1606.02585.
Shwetha, H., Kumar, D., 2016. Prediction of high spatio-temporal resolution land surface temperature under cloudy conditions using microwave vegetation index and ANN. ISPRS Journal of Photogrammetry and Remote Sensing 117, 40–55.
Siachalou, S., Mallinis, G., Tsakiri-Strati, M., 2015. A hidden Markov models approach for crop classification: Linking crop phenology to time series of multi-sensor remote sensing data. Remote Sens. 7 (4), 3633–3650.
Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference in Learning Representation. Vol. abs/1409.1.
Smits, P., Dellepiane, S., 1999. Discontinuity-adaptive Markov random field model for the segmentation of intensity sar images. IEEE Trans. Geosci. Remote Sens. 37 (1 II), 627–631.
Soltani-Farani, A., Rabiee, H., Jan 2015. When pixels team up: Spatially weighted sparse coding for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 12 (1), 107–111.
Soltani-Farani, A., Rabiee, H., Hosseini, S., Jan 2015. Spatial-aware dictionary learning for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 53 (1), 527–541.
Song, B., Li, P., Li, J., Plaza, A., Apr. 2016a. One-class classification of remote sensing images using kernel sparse representation. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 9 (4), 1613–1623.
Song, H., Huang, B., Liu, Q., Zhang, K., March 2015. Improving the spatial resolution of landsat tm/etm+ through fusion with spot5 images via learning-based super-resolution. IEEE Trans. Geosci. Remote Sens. 53 (3), 1195–1204.
Song, S., Xu, B., Li, Z., Yang, J., March 2016b. Ship detection in sar imagery via variational bayesian inference. IEEE Geosci. Remote Sens. Lett. 13 (3), 319–323.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15 (1), 1929–1958.
Storvik, G., Fjortoft, R., Solberg, A., 2005. A Bayesian approach to classification of multiresolution remote sensing data. IEEE Trans. Geosci. Remote Sens. 43 (3), 539–547.
Sun, L., Wu, Z., Liu, J., Xiao, L., Wei, Z., March 2015a. Supervised spectral -spatial hyperspectral image classification with weighted markov random fields. IEEE Trans. Geosci. Remote Sens. 53 (3), 1490–1503.
Sun, S., Zhong, P., Xiao, H., Wang, R., 2015b. An MRF model-based active learning framework for the spectral-spatial classification of hyperspectral imagery. IEEE J. Sel. Topics Signal Process. 9 (6), 1074–1088.
Sun, T., Jiao, L., Feng, J., Liu, F., Zhang, X., March 2015c. Imbalanced hyperspectral image classification based on maximum margin. IEEE Geosci. Remote Sens. Lett. 12 (3), 522–526.
Sun, X., Nasrabadi, N., Tran, T., Aug 2015d. Task-driven dictionary learning for hyperspectral image classification with structured sparsity constraints. IEEE Trans. Geosci. Remote Sens. 53 (8), 4457–4471.
Sun, X., Qu, Q., Nasrabadi, N., Tran, T., 2014. Structured priors for sparse-representation-based hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 11 (7), 1235–1239.
Sun, Z., Wang, C., Wang, H., Li, J., 2013. Learn multiple-kernel svms for domain adaptation in hyperspectral data. IEEE Geosci. Remote Sens. Lett. 10 (5), 1224–1228.
Sutton, C., McCallum, A., 2011. An introduction to conditional random fields. Foundations and Trends in Machine Learning 4 (4), 267–373.
Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C., 2008. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. IEEE Trans. Pattern Anal. Machine Intell. 30 (6), 1068–1080.
Sziranyi, T., Shadaydeh, M., 2014. Segmentation of remote sensing images using similarity-measure-based fusion-MRF model. IEEE Geosci. Remote Sens. Lett. 11 (9), 1544–1548.
Tang, Y. Y., Yuan, H., Li, L., Dec. 2014. Manifold-Based Sparse Representation for Hyperspectral Image Classification. IEEE Trans. Geosci. Remote Sens. 52 (12), 7606–7618.
Tenenbaum, J., de Silva, V., Langford, J., 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323.
Thoonen, G., Mahmood, Z., Peeters, S., Scheunders, P., 2012. Multisource classification of color and hyperspectral images using color attribute profiles and composite decision fusion. IEEE J. Sel. Topics Appl. Earth Observ. 5 (2), 510–521.
Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. B Met. 58 (1), 267–288.
Tokarczyk, P., Wegner, J., Walk, S., Schindler, K., Jan 2015. Features, color spaces, and boosting: New insights on semantic classification of remote sensing images. IEEE Trans. Geosci. Remote Sens. 53 (1), 280–295.
Tropp, J., Gibert, A., 2007. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory 53 (12), 4655–4666.

Tuia, D., Camps-Valls, G., 2011. Urban image classification with semisupervised multiscale cluster kernels. IEEE J. Sel. Topics Appl. Earth Observ. 4 (1), 65–74.
Tuia, D., Camps-Valls, G., 2016. Kernel manifold alignment for domain adaptation. PLoS One 11 (2), e0148655.
Tuia, D., Camps-Valls, G., Matasci, G., Kanevski, M., 2010. Learning relevant image features with multiple kernel classification. IEEE Trans. Geosci. Remote Sens. 48 (10), 3780 – 3791.
Tuia, D., Courty, N., Flamary, R., 2015. Multiclass feature learning for hyperspectral image classification: sparse and hierarchical solutions. ISPRS J. Int. Soc. Photo. Remote Sens. 105, 272–285.
Tuia, D., Muñoz-Marí, J., Kanevski, M., Camps-Valls, G., 2011. Structured output SVM for remote sensing image classification. J. Signal Proc. Sys. 65 (3), 457–468.
Tuia, D., Persello, C., Bruzzone, L., 2016. Recent advances in domain adaptation for the classification of remote sensing data. IEEE Geosci. Remote Sens. Mag. 4 (2), 41–57.
Tuia, D., Ratle, F., Pacifici, F., Kanevski, M., Emery, W. J., 2009. Active learning methods for remote sensing image classification. IEEE Trans. Geosci. Remote Sens. 47 (7), 2218–2232.
Tuia, D., Ratle, F., Pozdnoukhov, A., Camps-Valls, G., 2010. Multi-source composite kernels for urban image classification. IEEE Geosci. Remote Sens. Lett. 7 (1), 88–92.
Tuia, D., Verrelst, J., Alonso, L., Pérez-Cruz, F., Camps-Valls, G., 2011. Multioutput support vector regression for remote sensing biophysical parameter estimation. IEEE Geoscience and Remote Sensing Letters 8 (4), 804–808.
Tuia, D., Volpi, M., Copa, L., Kanevski, M., Muñoz-Marí, J., 2011. A survey of active learning algorithms for supervised remote sensing image classification. IEEE J. Sel. Topics Signal Proc. 5 (3), 606–617.
Tuia, D., Volpi, M., dalla Mura, M., Rakotomamonjy, A., Flamary, R., 2014a. Automatic feature learning for spatio-spectral image classification with sparse SVM. IEEE Trans. Geosci. Remote Sens. 52 (10), 6062–6074.
Tuia, D., Volpi, M., Moser, G., 2016. Getting pixels and regions to agree with conditional random fields. In: IEEE International Geoscience and Remote Sensing Symposium, IGARSS. Beijing, China.
Tuia, D., Volpi, M., Trolliet, M., Camps-Valls, G., 2014b. Semisupervised manifold alignment of multimodal remote sensing images. IEEE Trans. Geosci. Remote Sens. 52 (12), 7708–7720.
Vakalopoulou, M., Platias, C., Papadomanolaki, M., Paragios, N., Karantzalos, K., 2016. Simultaneous registration, segmentation and change detection from multisensor, multitemporal satellite image pairs. In: IEEE International Geoscience and Remote Sensing Symposium, IGARSS. Beijing, China.
Vapnik, V., 1998. Statistical Learning Theory. Wiley-Interscience, NJ, USA.
Verrelst, J., Alonso, L., Camps-Valls, G., Delegido, J., Moreno, J., 2011. Retrieval of canopy parameters using gaussian processes techniques. IEEE Transactions on Geoscience and Remote Sensing 49.
Verrelst, J., Alonso, L., Camps-Valls, G., Delegido, J., Moreno, J., May 2012. Retrieval of Vegetation Biophysical Parameters Using Gaussian Process Techniques. IEEE Trans. Geosci. Remote Sens. 50 (5, 2), 1832–1843.
Verrelst, J., Muñoz, J., Alonso, L., Delegido, J., Rivera, J., Camps-Valls, G., Moreno, J., 2012. Machine learning regression algorithms for biophysical parameter retrieval: Opportunities for Sentinel-2 and -3. Remote Sensing of the Environment 118, 127–139.
Vincent, L., Vincent, L., Soille, P., 1991. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Machine Intell. 13 (6), 583–598.
Voisin, A., Krylov, V., Moser, G., Serpico, S., Zerubia, J., 2014. Supervised classification of multisensor and multiresolution remote sensing images with a hierarchical copula-based approach. IEEE Trans. Geosci. Remote Sens. 52 (6), 3346–3358.
Volpi, M., Camps-Valls, G., Tuia, D., 2015. Spectral alignment of cross-sensor images with automated kernel canonical correlation analysis. ISPRS J. Int. Soc. Photo. Remote Sens. 107, 50–63.
Volpi, M., Ferrari, V., 2015a. Semantic segmentation of urban scenes by learning local class interactions. In: IEEE CVPR Workshop "Looking from above: when Earth observation meets vision". Boston, MA.
Volpi, M., Ferrari, V., 2015b. Structured prediction for urban scene semantic segmentation with geographic context. In: Joint Urban Remote Sensing Event (JURSE). Lausanne, Switzerland.
Volpi, M., Matasci, G., Kanevski, M., Tuia, D., 2014. Semi-supervised multiview embedding for hyperspectral data classification. Neurocomputing 145, 427–437.
Volpi, M., Tuia, D., in press. Dense semantic labeling with convolutional neural networks. IEEE Trans. Geosci. Remote Sens.
Volpi, M., Tuia, D., Camps-Valls, G., Kanevski, M., 2012a. Unsupervised change detection with kernels. IEEE Geosci. Remote Sens. Lett. 9 (6), 1026–1030.
Volpi, M., Tuia, D., Kanevski, M., 2012b. Memory-based cluster sampling for remote sensing image classification. IEEE Trans. Geosci. Remote Sens. 50 (8), 3096–3106.
von Luxburg, V., 2007. A tutorial on spectral clustering. Stat. Comput. 17 (4), 395–416.
Wang, C., Komodakis, N., Paragios, N., 2013. Markov random field modeling, inference and learning in computer vision and image understanding: A survey. Comput. Vis. Image Und. 117 (11), 1610–1627.
Wang, C., Mahadevan, S., 2005. Manifold alignment without correspondence. In: IJCAI'09 Proceedings of the 21st international jont conference on Artifical intelligence. pp. 1273–1278.
Wang, J., Zhao, Y., Li, C., Yu, L., Liu, D., Gong, P., 2015a. Mapping global land cover in 2001 and 2010 with spatial-temporal consistency at 250m resolution. ISPRS J. Photogramm. Remote Sens. 103, 38–47.
Wang, L., Hao, S., Wang, Q., Atkinson, P. M., 2015b. A multiple-mapping kernel for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 12 (5), 978–982.
Wang, L., Scott, A., Xu, L., Clausi, D., 2016. Sea ice concentration estimation during melt from dual-pol SAR scenes using deep convolutional neural networks: A case study. IEEE Trans. Geosci. Remote Sens. 54 (8), 4524–4533.
Wang, Q., Gu, Y., Tuia, D., in press. Discriminative multiple kernel learning for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens.

Wang, Z., Nasrabadi, N., Huang, T., March 2015c. Semisupervised hyperspectral classification using task-driven dictionary learning with laplacian regularization. IEEE Trans. Geosci. Remote Sens. 53 (3), 1161–1173.
Wang, Z., Zhang, L., Fang, T., Mathiopoulos, P., Tong, X., Qu, H., Xiao, Z., Li, F., Chen, D., May 2015d. A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification. IEEE Trans. Geosci. Remote Sens. 53 (5), 2409–2425.
Wegner, J., Montoya-Zegarra, J., Schindler, K., 2015. Road networks as collections of minimum cost paths. ISPRS J. Photogramm. Remote Sens. 108, 128–137.
Wegner, J. D., Branson, S., Hall, D., Schindler, K., Perona, P., 2016. Cataloging public objects using aerial and street-level images - urban trees. In: Computer Vision and Pattern Recognition. Las Vegas, NV.
Wehmann, A., Liu, D., 2015. A spatial-temporal contextual Markovian kernel method for multi-temporal land cover mapping. ISPRS J. Photogramm. Remote Sens. 107, 77–89.
Willsky, A., 2002. Multiresolution Markov models for signal and image processing. Proc. IEEE 90 (8), 1396–1458.
Wolpert, D., 1992. Stacked generalization. Neural Networks 5 (2), 241–259.
Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y., 2009. Robust face recognition via sparse representation. IEEE Trans. Patt. Anal. Mach. Intell. 31 (2), 210–227.
Wu, J., Jiang, Z., Luo, J., Zhang, H., 2014. Composite kernels conditional random fields for remote-sensing image classification. Electron. Lett. 50 (22), 1589–1591.
Wu, J., Yao, W., Choi, S., Park, T., Myneni, R., Nov 2015a. A comparative study of predicting dbh and stem volume of individual trees in a temperate forest using airborne waveform lidar. IEEE Geosci. Remote Sens. Lett. 12 (11), 2267–2271.
Wu, T.-F., Lin, C.-J., Weng, R., 2004. Probability estimates for multi-class classification by pairwise coupling. J. Mach. Learn. Res. 5, 975–1005.
Wu, Y., Wang, F., Zhang, Q., Niu, F., Li, M., 2015b. Fast algorithm based on superpixel-level conditional triplet Markov field for successive-approximation resistor image segmentation. IET Radar, Sonar and Navigation 9 (8), 1097–1105.
Xia, J., Chanussot, J., Du, P., He, X., May 2015a. Spectral-spatial classification for hyperspectral data using rotation forests with local feature extraction and markov random fields. IEEE Trans. Geosci. Remote Sens. 53 (5), 2532–2546.
Xia, J., Chanussot, J., Du, P., He, X., March 2016a. Rotation-based support vector machine ensemble in classification of hyperspectral data with limited training samples. IEEE Trans. Geosci. Remote Sens. 54 (3), 1519–1531.
Xia, J., Dalla Mura, M., Chanussot, J., Du, P., He, X., Sept 2015b. Random subspace ensembles for hyperspectral image classification with extended morphological attribute profiles. IEEE Trans. Geosci. Remote Sens. 53 (9), 4768–4786.
Xia, J., Falco, N., Benediktsson, J. A., Chanussot, J., Du, P., Apr. 2016b. Class-separation-based rotation forest for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 13 (4), 584–588.
Xia, J., Liao, W., Chanussot, J., Du, P., Song, G., Philips, W., July 2015c. Improving random forest with ensemble of features and semisupervised feature extraction. IEEE Geosci. Remote Sens. Lett. 12 (7), 1471–1475.
Xie, W., Jiao, L., Zhao, J., Feb 2016. Polsar image classification via d-ksvd and nsct-domain features extraction. IEEE Geosci. Remote Sens. Lett. 13 (2), 227–231.
Xiong, M., Ran, Q., Li, W., Zou, J., Du, Q., June 2015. Hyperspectral image classification using weighted joint collaborative representation. IEEE Geosci. Remote Sens. Lett. 12 (6), 1209–1213.
Xu, M., Jia, X., Pickering, M., Plaza, A. J., May 2016a. Cloud removal based on sparse representation via multitemporal dictionary learning. IEEE Trans. Geosci. Remote Sens. 54 (5), 2998–3006.
Xu, Y., Wu, Z., Li, J., Plaza, A., Wei, Z., Apr. 2016b. Anomaly detection in hyperspectral images based on low-rank and sparse representation. IEEE Trans. Geosci. Remote Sens. 54 (4), 1990–2000.
Yang, H., Crawford, M., Feb 2016a. Domain adaptation with preservation of manifold geometry for hyperspectral image classification. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 9 (2), 543–555.
Yang, H., Crawford, M., Jan 2016b. Spectral and spatial proximity-based manifold alignment for multitemporal hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 54 (1), 51–64.
Yang, Y., 2013. Land cover classification of high resolution images using superpixel-based conditional random fields. Int. J. Applied Math. Statist. 47 (17), 311–319.
Yokoya, N., Iwasaki, A., May 2015. Object detection based on sparse representation and hough voting for optical remote sensing imagery. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (5), 2053–2062.
Yousif, O., Ban, Y., 2014. Improving SAR-based urban change detection by combining MAP-MRF classifier and nonlocal means similarity weights. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 7 (10), 4288–4300.
Zhang, F., Du, B., Zhang, L., March 2016a. Scene classification via a gradient boosting random convolutional network framework. IEEE Trans. Geosci. Remote Sens. 54 (3), 1793–1802.
Zhang, G., Jia, X., Hu, J., Nov 2015a. Superpixel-based graphical model for remote sensing image mapping. IEEE Trans. Geosci. Remote Sens. 53 (11), 5861–5871.
Zhang, H., Li, J., Huang, Y., Zhang, L., 2014. A nonlocal weighted joint sparse representation classification method for hyperspectral imagery. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 7 (6), 2057–2066.
Zhang, H., Mendoza-Sanchez, I., Miller, E., Abriola, L., Jan 2016b. Manifold regression framework for characterizing source zone architecture. IEEE Trans. Geosci. Remote Sens. 54 (1), 3–17.
Zhang, L., Zhang, L., Tao, D., Huang, X., Febr. 2014. Sparse Transfer Manifold Embedding for Hyperspectral Target Detection. IEEE Trans. Geosci. Remote Sens. 52 (2), 1030–1043.
Zhang, P., Li, M., Wu, Y., Li, H., Sept 2015b. Hierarchical conditional random fields model for semisupervised sar image segmentation. IEEE Trans. Geosci. Remote Sens. 53 (9), 4933–4951.
Zhang, T., Yang, J., Zhao, D., Ge, X., March 2007. Linear local tangent space alignment and application to face recognition. Neurocomputing 70 (7-9, SI), 1547–1553.
Zhang, Y., Du, B., Zhang, L., March 2015c. A sparse representation-based binary hypothesis model for target detection in hyperspectral images. IEEE Trans. Geosci. Remote Sens. 53 (3), 1346–1354.

Zhang, Y., Yang, H. L., Prasad, S., Pasolli, E., Jung, J., Crawford, M., 2015d. Ensemble multiple kernel active learning for classification of multisource remote sensing data. IEEE J. Sel. Topics Appl. Earth Observ. 8 (2), 845–858.

Zhao, J., Zhong, Y., Wu, Y., Zhang, L., Shu, H., 2015a. Sub-pixel mapping based on conditional random fields for hyperspectral remote sensing imagery. IEEE J. Sel. Topic Signal Process. 9 (6), 1049–1060.

Zhao, J., Zhong, Y., Zhang, L., May 2015b. Detail-preserving smoothing classifier based on conditional random fields for high spatial resolution remote sensing imagery. IEEE Trans. Geosci. Remote Sens. 53 (5), 2440–2452.

Zheng, C., Wang, L., 2015. Semantic segmentation of remote sensing imagery using object-based Markov random field model with regional penalties. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 8 (5), 1924–1935.

Zhong, P., Wang, R., 2007. A multiple conditional random fields ensemble model for urban area detection in remote sensing optical images. IEEE Trans. Geosci. Remote Sens. 45 (12), 3978–3988.

Zhong, P., Wang, R., 2014. Jointly learning the hybrid CRF and MLR model for simultaneous denoising and classification of hyperspectral imagery. IEEE Trans. Neural Netw. Learning Systems 25 (7), 1319–1334.

Zhong, Y., Lin, X., Zhang, L., 2014a. A support vector conditional random fields classifier with a Mahalanobis distance boundary constraint for high spatial resolution remote sensing imagery. IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. 7 (4), 1314–1330.

Zhong, Y., Zhao, J., Zhang, L., 2014b. A hybrid object-oriented conditional random field classification framework for high spatial resolution remote sensing imagery. IEEE Trans. Geosci. Remote Sens. 52 (11), 7023–7037.

Zhou, J., Proisy, C., Descombes, X., le Maire, G., Nouvellon, Y., Stape, J.-L., Viennois, G., Zerubia, J., Couteron, P., 2013. Mapping local density of young eucalyptus plantations by individual tree detection in high spatial resolution satellite images. Forest Ecology and Management 301, 129–141.

Zhuolin, J., Zhe, L., Davis, L., 2013. Label consistent k-svd: learning a discriminative dictionary for recognition. IEEE Trans. Patt. Anal. Mach. Intell. 35 (11), 2651–2664.

Zou, J., Li, W., Du, Q., Dec 2015. Sparse representation-based nearest neighbor classifiers for hyperspectral imagery. IEEE Geosci. Remote Sens. Lett. 12 (12), 2418–2422.