J Sign Process Syst (2011) 65:301–310
DOI 10.1007/s11265-010-0483-8

Structured Output SVM for Remote Sensing Image Classification

Devis Tuia · Jordi Muñoz-Marí · Mikhail Kanevski · Gustavo Camps-Valls

Received: 15 January 2010 / Revised: 31 March 2010 / Accepted: 31 March 2010 / Published online: 28 April 2010

Abstract Traditional kernel classifiers assume independence among the classification outputs. As a consequence, each misclassification receives the same weight in the loss function. Moreover, the kernel function only takes into account the similarity between input values and ignores possible relationships between the classes to be predicted. These assumptions are not consistent for most of real-life problems. In the particular case of remote sensing data, this is not a good assumption either. Segmentation of images acquired by airborne or satellite sensors is a very active field of research in which one tries to classify a pixel into a predefined set of classes of interest (e.g. water, grass, trees, etc.). In this situation, the classes share strong relationships, e.g. a tree is naturally (and spectrally) more similar to grass than to water. In this paper, we propose a first approach to remote sensing image classification using structured output learning. In our approach, the output space structure is encoded using a hierarchical tree, and these relations are added to the model in both the kernel and the loss function. The methodology gives rise to a set of new tools for structured classification, and generalizes the traditional non-structured classification methods. Comparison to standard SVM is done numerically, statistically and by visual inspection of the obtained classification maps. Good results are obtained in the challenging case of a multispectral image of very high spatial resolution acquired with QuickBird over a urban area.

Keywords Structured output learning · Support vector machines · Kernel methods · Land use classification

1 Introduction

Remotely sensed images allow Earth Observation with unprecedented accuracy. New satellite sensors acquire images with high spectral and spatial resolution, and the revisiting time is constantly reduced. Processing data is becoming more complex in such situations and many problems are posed from a machine learning perspective, but image segmentation is probably the most critical and important application. The characteristics of the acquired images allow the characterization, identification, and classification of the land-covers [16]. However, traditional classifiers such as Gaussian maximum likelihood or artificial neural networks are

This work was partly supported by the Swiss National Science Foundation grants 100012-113506 and PBLAP2-127713, and by the Spanish Ministry of Science TEC2006-13845/TCM, CSD2007-00018, and AYA2008-05965-C04-03.

D. Tuia (✉) · M. Kanevski
Institute of Geomatics and Analysis of Risk,
University of Lausanne, Lausanne, Switzerland
e-mail: devis.tuia@unil.ch

M. Kanevski
e-mail: mikhail.kanevski@unil.ch

J. Muñoz-Marí · G. Camps-Valls
Image Processing Laboratory (IPL),
University of València, València, Spain

J. Muñoz-Marí
e-mail: jordi.munoz@uv.es

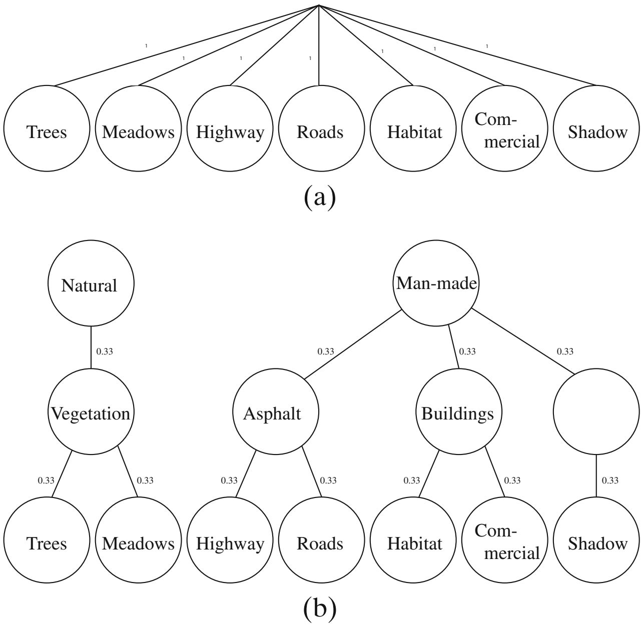
G. Camps-Valls
e-mail: gustavo.camps@uv.es

affected by the high input sample dimension, tend to overfit data in the presence of noise or perform poorly when a low number of training samples are available [11, 12].

In the recent years, the use of support vector machines (SVMs) [17] for remote sensing image classification has been paid attention basically because the method integrates in the same classification procedure: (1) a *feature extraction* step, as samples are mapped to a higher dimensional space where a simpler (linear) classification is performed, becoming nonlinear in the input space; (2) a *regularization* procedure by which model's complexity is efficiently controlled; and (3) the minimization of an upper bound of the generalization error, thus following the Structural Risk Minimization (SRM) principle. The application of SVMs have demonstrated very good performance in multispectral, hyperspectral, and multi-source image classification [5–7, 15]. Kernel methods rely on the definition of a distance measure between input samples (pixels) in a proper Hilbert space.

In standard kernel machines, relations between the outputs is not considered explicitly, which constitutes a theoretical limitation of the approach. In this paper, we propose the use of a recent machine learning framework, the *structured output learning*, to improve the quality of remote sensing image classification by taking into account such relations.

Suppose we are given a set of n labeled pixels in a remotely sensed image $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathcal{X} \times \mathcal{Y}$. In the classical SVM image classification scenario, the similarity between pixels $\mathbf{x}_i$ and $\mathbf{x}_j$ is used to predict the outputs y_i using the kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle$. The output vector accounts for a land-use or land-cover class, e.g. class label y_i for pixel $\mathbf{x}_i$ may refer to 'grass', 'soil', or 'water'. This way of proceeding assumes independence between the outputs, as illustrated in Fig. 1a. This choice is justified by the fact that, in principle, no assumptions about the distribution of the outputs may be done. However, a pixel can be associated with a *structured output* that considers a more complex relational information, for instance, through a hierarchical tree structure showing several classification levels. Figure 1b illustrates this idea. Considering the structure relating the possible classification outputs, direct output class dependencies are assumed and also different forms of penalizing misclassification errors can be achieved according to the designed structure. This way, it seems obvious that predicting 'Highway' for a 'Tree' pixel should be penalized more than predicting 'Grass', because the latter are intuitively more related classes, and in fact, input samples (spectra) are more similar.



(a)

(b)

Figure 1 Examples of **a** classical non structured and **b** structured output representation.

In the literature, this output information is indirectly exploited through the use of cost matrices and balancing constraints, typically encoded a priori. This approach is not useful when few (or not representative) labeled samples are available. Besides, these are second-order output class relations, and they are not learned from data but fixed before learning. Structured output learning [3] formalizes the problem of output space relations. This methodological framework aims at predicting complex objects, such as trees, sequences or web queries, where, contrarily to usual machine learning algorithms, the relationship between the associated outputs plays a role in the prediction.

Several ways can be considered to introduce interdependencies in the output space. In this paper, Structured Support Vector Machines (SSVM) are considered. The first attempts at SSVM classifiers can be found in the machine learning literature [13, 19, 20, 22]. In [1], an excellent review of the state of the art in structured learning is presented. Despite the confinement of these methods in the machine learning community, the first applications of structured output learning appeared in other disciplines, such as natural language learning [13] or object localization [4]. However, its application to image classification has not been presented so far, and currently no remote sensing applications of SSVM (or of structured learning in general) can be found. Nonetheless, the manifold where natural and remote sensing images lie is typically smooth and dominated by strong local relations. Therefore, similar classes should be nearby in the manifold and, by considering output relations, structured learning may bring relevant information to the classification task.

This paper introduces the concept of structured learning for the simplest case of multiclass classification. Two implementations are considered: the first with a simple, yet effective, modification of the SVM loss function, and the second introducing the tree structure in the measure of similarity, the kernel. The presented framework is evaluated on a very high resolution (VHR) image classification problem. Section 2 illustrates the modification to the standard SVM to obtain the SSVM formulation. Section 3 presents the experimental setup of the experiments presented in Section 4. Section 5 concludes the work.

2 Structured Output Learning

This section revises the theory and novelties introduced by the structured learning paradigm. Then, we develop the SSVM algorithm. A simple hierarchical structure is designed to include output relations in the context of remote sensing image classification.

2.1 Basic Concepts

The aim of structured learning for classification is, as for the canonical classification setting, to learn a function h such as

$$h : \mathcal{X} \rightarrow \mathcal{Y}, \tag{1}$$

where $\mathcal{X}$ is the space of inputs and $\mathcal{Y}$ is the space of structured outputs. Using an i.i.d. training sample $\mathbf{X} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)\}$, h is the function minimizing the empirical risk

$$R_E^{\Delta}(h) = \frac{1}{n}\sum_{i=1}^{n} \Delta\big(y_i, \bar{y}_i\big), \tag{2}$$

where $\bar{y}_i$ is the prediction and y is the correct class assignment. The quality of the model is evaluated by a loss function: $\Delta(y_i, \bar{y}_i) = 0$ if the label is correctly assigned and $\Delta(y_i, \bar{y}_i) \geq 0$ otherwise. Note that the classical 0/1 loss is a particular case of this function returning the loss 1 for each wrong output. Coming back to the h functions, they are of the form

$$h(x) : \arg\max_{y \in \mathcal{Y}} f(\mathbf{x}, y), \tag{3}$$

where $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a joint function between inputs and outputs evaluating how well a certain prediction matches the observed output. The joint function f can be represented as $f(\mathbf{x}, y) = \mathbf{w}^{\top}\Psi(\mathbf{x}, y)$, where $\mathbf{w}$ is a weight vector and the joint input–output mapping Ψ relates inputs $\mathbf{x}$ and outputs y.

In order to account for a structured output, or object, two main ingredients of the SVM must be modified: the *mapping* Ψ and the *loss function* Δ. It is clear that such modification leads to SSVM formulations that are application-dependent and must take advantage of the specificities of the particular task. The examples considered in this paper address the problem of multiclass spectral image classification.

2.2 The Joint Input–Output Mapping

Figure 1b shows the tree structure for a multiclass classification task with seven classes (lower level in the tree), even though 13 classes can be created from the structure by grouping physically-similar classes into superclasses (upper levels). The goal is to encode the structure of the tree into the mapping $\phi(\mathbf{x})$, resulting into a joint input–output mapping $\Psi(\mathbf{x}, y)$. In [1], a mapping considering tree-structures is proposed for taxonomies. Consider a taxonomy as a set of elements $\mathcal{Z} \supseteq \mathcal{Y}$ ordered by a partial order $\prec$, and let $\beta_{(y,z)}$ be a measure of similarity respecting the order $\prec$. The representation of the outputs $\Lambda(y)$ can be generalized as

$$\lambda_z(y) = \begin{cases} \beta_{(y,z)} & \text{if } y \prec z \text{ or } y = z \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

For instance, in the guiding example of Fig. 1 and using $\beta_{(y,z)} = 1$, the class “Meadows” will be represented as

$$\Lambda(\text{Meadows}) = \begin{pmatrix} 0 & \\ 1 & \text{(Meadows is equal to Meadows)} \\ 0 & \\ 0 & \\ 0 & \\ 0 & \\ 0 & \\ 1 & \text{(Vegetation is a superclass of Meadows)} \\ 0 & \\ 0 & \\ 1 & \text{(Natural is a superclass of Meadows)} \\ 0 & \end{pmatrix}$$

This way, the similarity between two outputs sharing common superclasses will be higher than between outputs that are distant in the tree. Then, we can define the joint input–output feature map via a tensor product

$$\Psi(\mathbf{x}, y) = \Phi(\mathbf{x}) \otimes \Lambda(y) \tag{5}$$

This formulation introduces a weight vector $\mathbf{w}_z$ for every node in the hierarchy. The inner product of the joint feature map decomposes into kernels over input and output space (using the properties proposed in [17]):

$$\begin{aligned}\left\langle \Psi(\mathbf{x}, y), \Psi\left(\mathbf{x}', y'\right)\right\rangle &= K_{\otimes}((\Phi(\mathbf{x}), \Lambda(y)), \left(\Phi\left(\mathbf{x}'\right), \Lambda\left(y'\right)\right)\\ &= \left\langle \Lambda(y), \Lambda\left(y'\right)\right\rangle K\left(\mathbf{x}, \mathbf{x}'\right) \qquad (6)\end{aligned}$$

In order to illustrate this principle, consider a three-class problem with the structure shown in Fig. 2. Classes 4, 5 and 6 are the superclasses giving the tree structure.

In the non-structured version of the algorithm (equivalent to usual multiclass classification), the Λ and Ψ vectors take the form:

$$\Lambda(\mathrm{y})^{\top} = \begin{pmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{pmatrix} \qquad \Psi(\mathrm{y})^{\top} = \begin{pmatrix} \mathbf{x}\,0\,0 \\ 0\,\mathbf{x}\,0 \\ 0\,0\,\mathbf{x} \end{pmatrix}$$

Taking into account the structure shown in Fig. 2, Λ and Ψ become:

$$\Lambda(\mathrm{y})^{\top} = \begin{pmatrix} 1\,0\,0\,1\,0\,1 \\ 0\,1\,0\,1\,0\,1 \\ 0\,0\,1\,0\,1\,1 \end{pmatrix} \qquad \Psi(\mathrm{y})^{\top} = \begin{pmatrix} \mathbf{x}\,0\,0\,\mathbf{x}\,0\,\mathbf{x} \\ 0\,\mathbf{x}\,0\,\mathbf{x}\,0\,\mathbf{x} \\ 0\,0\,\mathbf{x}\,0\,\mathbf{x}\,\mathbf{x} \end{pmatrix}$$

The linear dot product between the two first classes will result in $2\left\langle \mathbf{x}, \mathbf{x}'\right\rangle$, while between the classes 1 and 3 (and 2 and 3) it is of $\left\langle \mathbf{x}, \mathbf{x}'\right\rangle$ only. Thus, using a joint input–output mapping, output structure participates to the similarity between samples.

2.3 The Loss Function

To define the loss function, we can modify the classical 0/1 loss by exploiting the tree-based output structure.

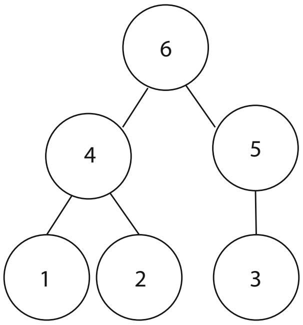


Figure 2 Toy example of structure.

The proposed tree-based loss assumes a common superclass in the tree at level $l = \{1, ..., L\}$ as follows:

$$\Delta(y, \bar{y}) = \begin{cases} (l-1)/L & \text{if } y^l = \bar{y}^l \\ 1 & \text{otherwise} \end{cases} \qquad (7)$$

Using this loss, errors predicting 'far away' classes are penalized more than 'close' errors. A class predicted correctly will receive a loss of zero ($l - 1 = 0$), while the prediction of a class not sharing any superclass with the true class will receive a loss of 1.

The loss function presented in Eq. 7 assumes equal distance between the classes and their superclasses: this can be refined by constructing ad hoc class distances from the labeled data or by learning inter-class distances through clustering or bag kernels [21].

2.4 The *n*-slack and 1-slack SSVM

The modification of the loss and the mapping allows us the integration of output-space similarities into the kernel function. However, to exploit this new source of information, the whole SVM must be reformulated: it is easy to see that the mapping $\Psi(\mathbf{x}, y)$ cannot be computed for test points, for which the class membership is unknown. In order to solve this general problem in structured learning, specific (structured) SVM formulations must be developed. Several strategies have been proposed for the SSVM [2, 14, 18–20], but the formulation of [19] is the most general as it includes the rest as particular cases. This formulation is usually referred to as the *n-slack Structured-outputs SVM* (*n*-SSVM), since it assigns a different slack variable to each of the n training examples. Specifically, in the margin-rescaling version of [19], the position of the hinge is adapted while the slope is fixed. Each possible output is considered and the model is constrained iteratively by adding constraints on the $(\mathbf{x}, \mathbf{y})$ pairs that most violate the SVM solution (note that $\mathbf{y}$ has become a vector containing all possible outputs). In other words, a sort of regularization is done, restricting the set of possible functions h. This way, the formulation becomes:

$$\min_{\mathbf{w}, \xi} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i \qquad (8)$$

$$\forall i : \xi_i \geq 0$$

$$\underbrace{\forall \bar{y} \in \mathcal{Y}}_{-1-}, \underbrace{\forall i}_{-2-} : \underbrace{\left\langle \mathbf{w}, \Psi\left(\mathbf{x}_i, y_i\right)\right\rangle}_{-3-} - \underbrace{\left\langle \mathbf{w}, \Psi\left(\mathbf{x}_i, \bar{y}_i\right)\right\rangle}_{-4-}$$

$$\geq \underbrace{\Delta\left(y_i, \bar{y}\right)}_{-5-} - \xi_i \qquad (9)$$

The objective is the conventional regularized risk used in SVMs. The constraints state that for each incorrect label (–1–) and for each training example $(\mathbf{x}_i, y_i)$ (–2–), the score $\langle \mathbf{w}, \Psi(\mathbf{x}_i, y_i)\rangle$ of the correct structure y_i (–3–) must be greater than the score $\langle \mathbf{w}, \Psi(\mathbf{x}_i, \bar{y}_i)\rangle$ of all incorrect structures $\bar{y}$ (–4–) by a required margin (–5–). If the margin is violated, the slack variable ξ_i becomes non zero.

The above quadratic program involves a very large, possibly infinite number of linear inequality constraints. This high number of inequalities is too large to be optimized explicitly. Alternatively, the problem is solved by using *delayed constraint generation* where only a finite and small subset of constraints is taken into account. Optimizing over a subset of the constraints enlarges the feasible set and will yield a solution which provides a lower bound on the objective. The algorithm works iteratively by finding the most violated constraints. Since exact solution would imply a huge amount of constraints, a violating constraint is added to the model only if the violation if greater than a precision threshold ϵ. Nonetheless, this algorithm is not efficient: it iterates over all training examples and adds several constraints at each iteration. It has been proven in [19] that a greedily constructed model of the n-slack SSVM requires $\mathcal{O}(\frac{n}{\epsilon^2})$ constraints. Although it could be efficient when the most violated constraints can be found quickly, the n-SSVM is computationally expensive when working with many training samples. To solve this issue, the n-slack algorithm has been reformulated in [13] to a model several orders of magnitude faster: the 1-*slack SSVM* (1-SSVM). As suggested by its name, the model has a unique slack variable ξ applied to all the constraints. The interested reader can find the proof for the equivalence with the n-slack formulation in [13]. The n cutting planes of the previous model (one for each training example) are replaced by a single cutting plane for the sum of the hinge-losses. In this sense, Eqs. 8 and 9 can be replaced by

$$\min_{\mathbf{w},\xi} \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{n}\xi \tag{10}$$

$$\forall \bar{y} \in \mathcal{Y} : \frac{1}{n}\sum_{i=1}^{n}\left[\left\langle \mathbf{w}, \Psi\left(\mathbf{x}_i, y_i\right)\right\rangle - \left\langle \mathbf{w}, \Psi\left(\mathbf{x}_i, \bar{y}_i\right)\right\rangle\right]$$

$$\geq \frac{1}{n}\sum_{i=1}^{n}\Delta\left(y_i, \bar{y}\right) - \xi \tag{11}$$

where $\xi = 1/n\sum_i \xi_i$.

Algorithm 1 shows the 1-SSVM, as proposed in [13]. Starting with an empty working set of constraints $\mathcal{W} = \emptyset$, the algorithm computes the solution over the current $\mathcal{W}$, finds the most violated constraint (one, for all the training points) and adds it to the working set. The algorithm terminates when no constraint is added in the previous iteration, i.e. when all the constraints are fulfilled up to a precision ϵ. Unlike the n-SSVM, only one constraint is added at each iteration. This new formulation has $|\mathcal{Y}|^n$ constraints, one for each possible combination of labels $[\bar{y}_1, \ldots, \bar{y}_n]$, but only one slack variable ξ is shared across all constraints.

Algorithm 1 1-slack SSVM with margin rescaling [13]

Inputs

- Initial training set $\mathbf{X} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ $(n \times d + 1)$.
- Set of constraints $\mathcal{W} \leftarrow \emptyset$.
- Precision ϵ (scalar).

```
repeat
    solve the SVM using Eqs. 10 and 11
    Result are parameters $(\mathbf{w}, \xi)$
    for $i = 1, \ldots, n$ do
        predict the label $\bar{y}_i \leftarrow \arg\max_{\bar{y}\in\mathcal{Y}}\{\Delta(y_i, \bar{y}) + \langle \mathbf{w}, \Psi(x_i, \bar{y})\rangle\}$
    end for
    $\mathcal{W} \leftarrow \mathcal{W} \cup \{(\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_n)\}$
until Eq. 11 is fulfilled with $\epsilon$ precision for all the active constraints $\left(\text{i.e. } \frac{1}{n}\sum_i \Delta(y_i, \bar{y}_i) - \frac{1}{n}\sum_i \left[\langle \mathbf{w}, \Psi(\mathbf{x}_i, y_i)\rangle - \langle \mathbf{w}, \Psi(\mathbf{x}_i, \bar{y}_i)\rangle\right] \leq \xi + \epsilon\right)$.
```

3 Data and Experimental Setup

Experiments have been carried out on a four-bands optical image of the city of Zurich (Fig. 3a). The image, acquired by the sensor QuickBird, in 2006 has size (828 × 889) pixels. QuickBird is a high-resolution commercial earth observation satellite, owned by DigitalGlobe and launched in 2001. The satellite collects panchromatic imagery at 60–70 cm resolution and multispectral imagery at 2.4 and 2.8-m resolutions. The latter was pansharpened using Bayesian Data Fusion [9] to attain a spatial resolution of 0.6 m. Seven classes of interest have been highlighted by photointerpretation and 254,469 pixels have been carefully labeled (Fig. 3b). For analysis, these pixels have been randomly split into three sets: for training (1,400 pixels, 200 per class), model selection (5,000) and validation (248,769). The input variables are the four spectral bands and

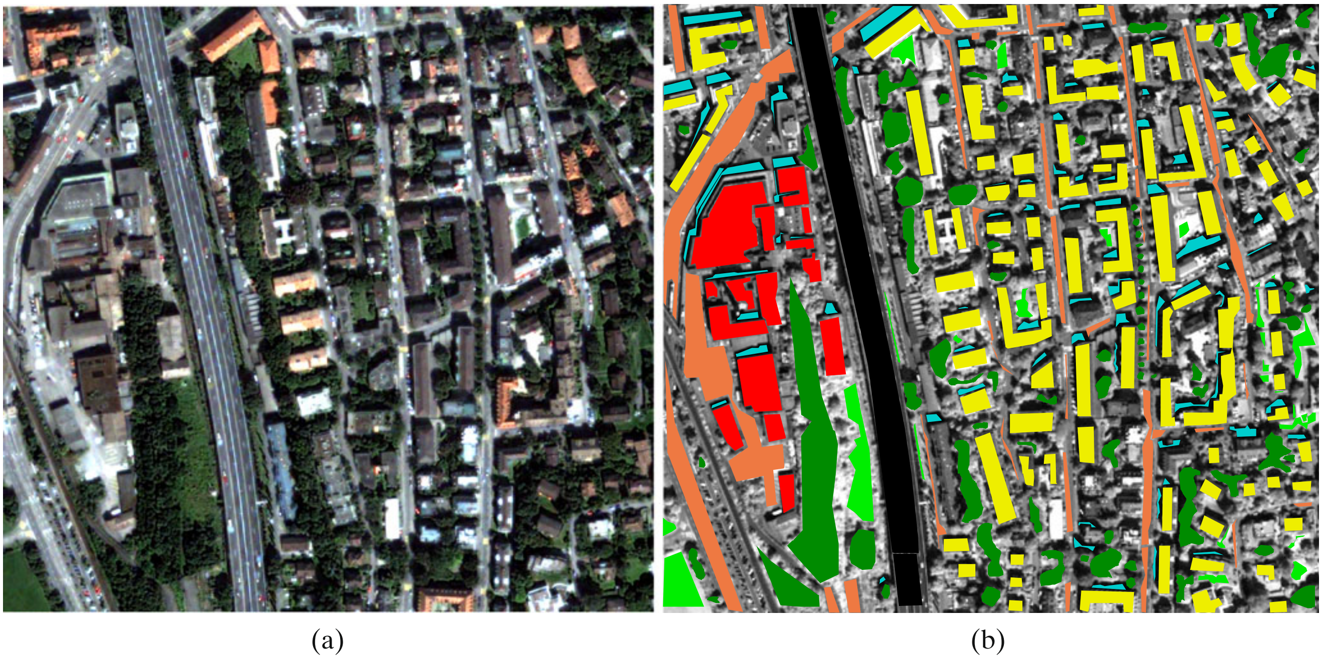


Figure 3 Multispectral very high resolution Quickbird image acquired over Zurich. **a** RGB composition of the image and **b** ground survey of the seven classes of interest identified: 'trees' (*Dark green*), 'meadows' (*light green*), 'highway' (*black*), 'road' (*brown*), 'residential' (*orange*), 'commercial' (*red*) 'and shadow' (*blue*).

six morphological features extracted from the panchromatic band (opening and closing with increasing size structured elements (5, 9, 13 pixels)).

Structure has been imposed using the tree shown in Fig. 1b. The empty class, superclass of 'shadow' has been added in order to give a three-level hierarchy for each of the seven classes and results in a normalized mapping. Three main experiments have been carried out:

1. `Loss`: only the loss function $\Delta(y, \bar{y}_i)$ is modified. This is investigated by three different modifications: first, the modification proposed in Section 2.3:
   - `Loss-Tree1`: encoding the similarity of the tree in Fig. 1b with $\beta = 0.33$.

   Then, one could decide to encode a greater similarity to classes sharing the first common superclass, inducing an asymmetric tree, or to derive the loss from training data, for instance by computing class mean vectors and computing a distance matrix between them. These algorithmical variations are analyzed in experiments `Tree2` and `Tree3`:
   - `Loss-Tree2`: decreasing β to 0.1 between classes sharing a direct superclass.
   - `Loss-Tree3`: loss assessed by distance between training class mean vectors.
2. `PSI`: the mapping $\Psi(\mathbf{x}_i, y_i)$ is tree weighted using Eq. 6, where $\langle \Lambda(y), \Lambda(y') \rangle$ is set to $1 - \Delta(y, y'_i)$. The loss used is the 0/1 loss.
3. `Loss-PSI`: both Δ and Ψ are modified using the modifications presented above. That give birth to three experiments using respectively the losses `Tree1`, `Tree2` and `Tree3`.

SSVM algorithm is implemented using the SVM^{struct} library,[1] and compared with the multiclass implementation of the SVM of [8], also using the same library.

4 Results and Discussion

Table 1 illustrates the results of the experiments mentioned above. The standard SVM, taken as a reference, results in an overall accuracy of 76.17% and an estimated kappa statistic of 0.711. Looking at the user's accuracies, most of the confusion is observed for the classes 'Residential' and 'Commercial': this was expected, because several residential buildings have a

[1] Available at http://svmlight.joachims.org/svm_struct.html.

Table 1 Classification accuracies (in %) and estimated kappa statistic (along with its 95% confidence intervals).

Loss	SVM	SSVM–`Loss`			SSVM–`PSI`	SSVM–`Loss-PSI`		
	0/1	`Tree1`	`Tree2`	`Tree3`	0/1	`Tree1`	`Tree2`	`Tree3`
Trees	85.07	82.41	**87.67**	84.96	**86.74**	84.78	**87.39**	83.64
Meadows	96.22	96.02	93.34	95.29	**96.44**	95.90	94.27	94.86
Highway	96.98	94.15	94.54	95.28	93.68	95.50	94.09	95.68
Roads	73.40	69.58	**75.76**	**77.24**	72.70	**75.15**	73.32	70.96
Residential	59.67	**71.22**	**68.48**	**63.71**	**62.90**	**66.61**	**67.85**	**65.84**
Commercial	67.09	57.54	50.20	59.75	**69.59**	55.60	52.38	62.32
Shadows	99.62	**99.77**	99.49	99.53	99.04	99.59	**99.77**	99.44
Overall	76.17	**77.30**[a]	**77.72**[a]	**77.07**[a]	**77.25**[a]	**77.21**[a]	**77.29**[a]	**76.66**[a]
Kappa	0.711	**0.722**	**0.726**	**0.721**	**0.724**	**0.722**	**0.722**	**0.715**
C.I.	[0.709; 0.713]	[0.720; 0.724]	[0.724; 0.728]	[0.719; 0.723]	[0.722; 0.726]	[0.720; 0.724]	[0.719; 0.724]	[0.713; 0.717]

In bold, the results outperforming the standard SVM

[a] Significantly improves the Standard SVM by the McNemar's test [10]

roof color similar to the commercial center. The main challenge of the method is thus to resolve the confusion between these two classes.

The `Loss` results show a significant improvement (tested by the McNemar's test) of the SVM solution of about 1% to 1.5%: in all these experiments, the

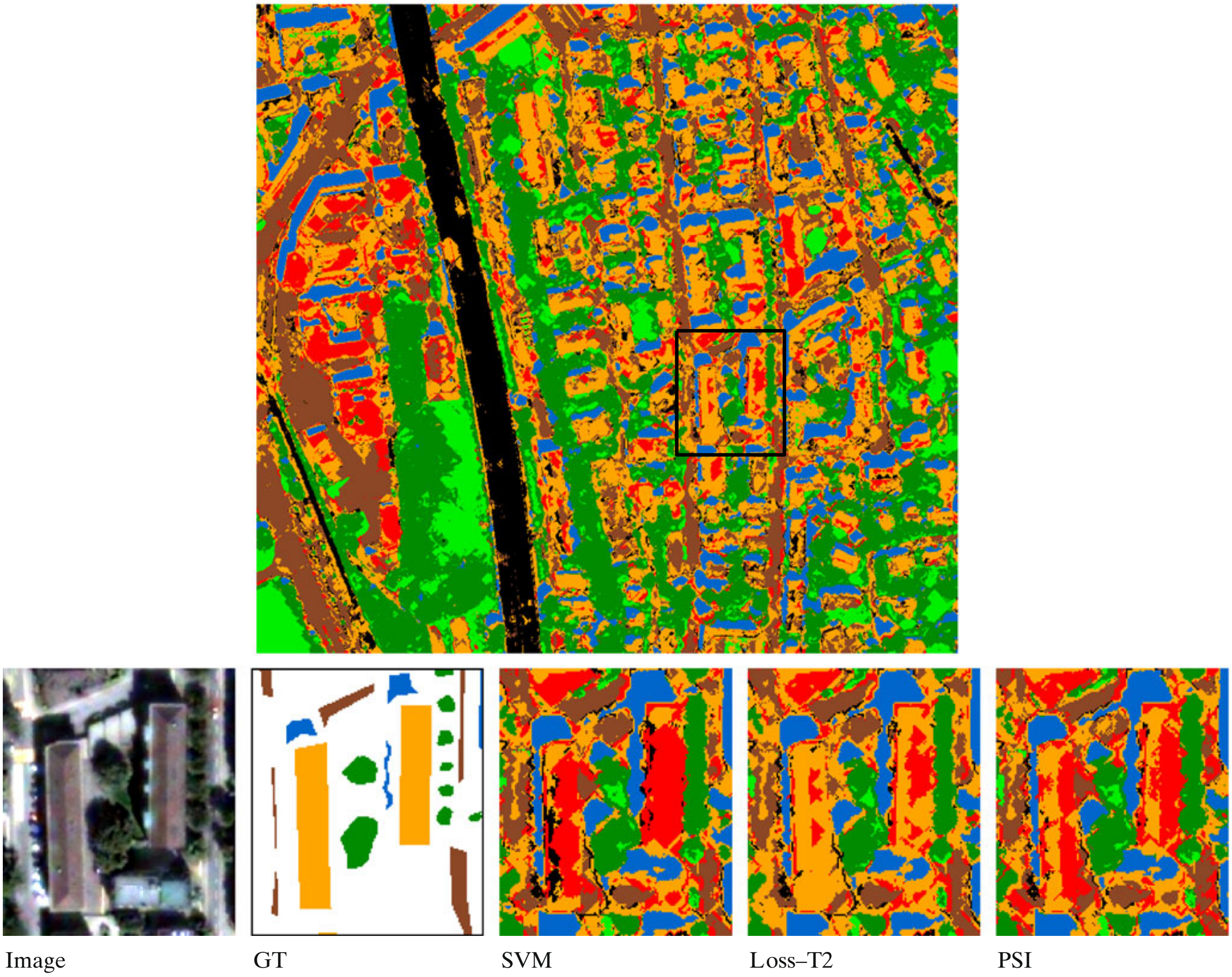


Figure 4 Classification map, `Loss-Tree2` experiment. *Bottom row*: detail of classification for the SVM, `Loss-Tree2` and `PSI` experiments.

Table 2 Classification accuracies (in %) and estimated kappa statistic (along with its 95% confidence intervals) for the second-level classification (four classes).

Loss	SVM	SSVM–`Loss`			SSVM–`PSI`	SSVM–`Loss-PSI`		
	0/1	`Tree1`	`Tree2`	`Tree3`	0/1	`Tree1`	`Tree2`	`Tree3`
Natural	96.86	95.03	96.24	95.84	**97.38**	96.61	96.07	95.19
Asphalt	84.53	80.88	**85.17**	**85.65**	82.19	**85.16**	83.54	82.18
Buildings	80.82	**86.59**	**82.89**	80.67	**84.87**	**83.35**	**83.51**	**83.01**
Shadows	99.62	**99.77**	99.49	99.54	99.04	99.60	**99.77**	99.44
Overall	87.00	**87.86**[a]	**87.85**[a]	86.99	**88.09**[a]	**88.14**[a]	**87.62**[a]	86.79
Kappa	0.812	**0.823**	**0.824**	0.812	**0.827**	**0.828**	**0.821**	0.808
C.I.	[0.810; 0.814]	[0.821; 0.825	[0.822; 0.826	[0.899; 0.814]	[0.825; 0.829]	[0.826; 0.830]	[0.819; 0.823]	[0.806; 0.810]

In bold, the results outperforming the standard SVM

[a] Significantly improves the Standard SVM by the McNemar's test [10]

class 'Residential' increases in accuracy, as well as the roads (`Tree2` and `Tree3`). Even though it is not a big gain, this improvement is achieved by modifying the loss function only and practically no additional computational cost is involved. Figure 4 illustrates the classification map obtained with the LS–T2 experiment: the strong confusion between residential and commercial buildings is visible on the left side of the image.

The `PSI` experiment shows a solution similar to the ones observed above. Nevertheless, the accuracy is improved for four classes, among which 'Commercial' and 'Residential'. This proves that the modification of the mapping Ψ allows to encode additional information about the pixels. The solution seems more stable: unlike in the previous experiments, there is no class where the accuracy is degraded.

The last experiments (`Loss-PSI`) combine the ideas of the previous ones: both the loss and the mapping are modified. The effect of the new loss seems to be stronger, since the solutions are similar to the ones obtained in the `Loss` experiments. With respect to the latter, the overall performance is lightly degraded (especially for the `Tree3` loss), probably due to the increase of the complexity of the model.

In the introduction, we stated that the interest of adding information about the output structure in taxonomies would help penalizing the prediction of classes whose outputs are distant in the tree structure. In other words, confusion between higher levels of the taxonomy should be avoided. To confirm this hypothesis, we have re-grouped the seven-classes predictions of Table 1 into the four classes of the second level of the tree of Fig. 1, namely 'Vegetation', 'Asphalt', 'Buildings' and 'Shadow'. Table 2 illustrates the results for the aggregated data. The gains in overall accuracy are, again, of the order of 1–1.5% and of 0.15 for the estimated kappa statistic. The `PSI` and the `Loss-PSI` with `Tree1` loss result in the best solutions, confirming that the modification of the mapping helps the coherence of the solution. On the contrary, the mean vectors loss (`Tree3`) does not result in an improved solution of the standard SVM (the result of the McNemar's test is negative, thus giving preference to the latter): a loss function computed on training data seems not to be satisfactory, probably because its information is redundant with the kernel $K(\mathbf{x}, \mathbf{x}')$, while the other losses introduce new independent information related to expert knowledge of the class similarities.

5 Conclusions

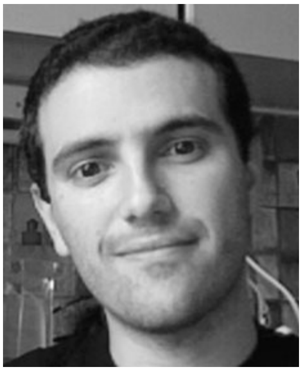
We proposed the use of structured learning for remote sensing image classification. The framework has been presented and analyzed theoretically. Also a structured SVM has been developed by embedding output label similarity in the machine through a simple hierarchical tree. The output relations are then used to modify the SVM loss function and the kernel function. Experiments on a VHR image classification problem showed good results, and encourage future research in the field of remote sensing structured learning. Ongoing work is focused on design of spatially structured output kernels.

Acknowledgements The authors would like to thank Prof. Thorsten Joachims, and Drs. Matthew Blaschko and Y. Altun for the fruitful discussions and advices.

References

1. Altun, Y., Hofmann, T., & Tsochantaridis, I. (2007). Support vector learning for interdependent and structured output spaces. In G. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, & S. Vishwanathan (Eds.), *Predicting structured data* (pp. 85–105). Cambridge: MIT press.

2. Altun, Y., Tsochantaridis, I., & Hofmann, T. (2003). Hidden Markov support vector machines. In *International conference on machine learning (ICML)* (pp. 3–10).
3. Bakır, G., Hofmann, T., Schölkopf, B., Smola, A. J., & Vishwanathan, S. (2007). *Predicting structured data*. Cambridge: MIT Press.
4. Blaschko, M., & Lampert, C. (2008). Learning to localize objects with structured output regression. In D. Forsyth, P. Torr, & A. Zisserman (Eds.), *Computer vision: ECCV 2008* (pp. 2–15). Heidelberg: Springer.
5. Camps-Valls, G., & Bruzzone, L. (2005). Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing, 43*, 1351–1362.
6. Camps-Valls, G., & Bruzzone, L. (2009). Kernel methods in remote sensing image processing. Hoboken: Wiley.
7. Camps-Valls, G., Gómez-Chova, L., Muñoz-Marí, J., Rojo-Álvarez, J. L., & Martínez-Ramón, M. (2008). Kernel-based framework for multi-temporal and multi-source remote sensing data classification and change detection. *IEEE Transactions on Geoscience and Remote Sensing, 46*(6), 1822–1835.
8. Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass problems. *Journal of Machine Learning Research, 2*, 265–292.
9. Fasbender, D., Radoux, J., & Bogaert, P. (2008). Bayesian data fusion for adaptable image pansharpening. *IEEE Transactions on Geoscience and Remote Sensing, 46*(6), 1847–1857.
10. Foody, G. M. (2004) Thematic map comparison: Evaluating the statistical significance of differences in classification accuracy. *Photogrammetric Engineering and Remote Sensing, 50*(5), 627–633.
11. Fukunaga, K., & Hayes, R. R. (1989). Effects of sample size in classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 11*(8), 873–885.
12. Hughes, G. F. (1968). On the mean accuracy of statistical pattern recognition. *IEEE Transactions on Information Theory, IT-14*(1), 55–63.
13. Joachims, T., Finley, T., & Yu, C. N. J. (2009). Cutting-plane training of structural SVMs. *Machine Learning, 77*, 27–59.
14. Joachims, T., Galor, T., & Elber, R. (2005). Learning to align sequences: A maximum-margin approach. In B. Leimkuhler (Ed.), *New algorithms for macromolecular simulation, LNCS* (Vol. 49). Heidelberg: Springer.
15. Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing, 42*(8), 1778–1790.
16. Richards, J. A., & Jia, X. (1999). *Remote sensing digital image analysis. An introduction* (3rd ed.). Berlin: Springer.
17. Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge: MIT Press.
18. Taskar, B., Guestrin, C., & Koller, D. (2003). Maximum-margin Markov networks. In *Advances in neural information processing systems (NIPS)*.
19. Tsochantaridis, I., Finley, T., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research, 6*, 1453–1484.
20. Tsochantaridis, I., Hofmann, T., Joachims, T., & Alturn, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Intl. conf. mach. learn.*
21. Tuia, D., & Camps-Valls, G. (2009). Semi-supervised remote sensing image classification with cluster kernels. *IEEE Geoscience and Remote Sensing Letter, 6*(1), 224–228.
22. Zien, A., Brefeld, U., & Scheffer, T. (2007). Transductive support vector machines for structured variables. In *Intl. conf. mach. learn.*



Devis Tuia was born in Mendrisio, Switzerland, in 1980. He received a diploma in Geography at the University of Lausanne in 2004, a Master of Advanced Studies in Environmental Engineering at the Federal Institute of Technology of Lausanne (EPFL) in 2005 and a PhD in Environmental Sciences at the University of Lausanne in 2009. He is currently a postdoc researcher at the Univeristy of València, Spain and the Univeristy of Colorado at Boulder, under a Swiss National Foundation program.

His research interests include the development of algorithms for feature selection and classification of very high resolution remote sensing images and socio-economic data using kernel methods. Particularly, his studies focused on the use of unlabeled samples and on the interaction between the user and the machine through active and semisupervised learning. Visit http://devis.tuia.googlepages.com for more information.

Jordi Muñoz-Marí was born in València, Spain in 1970, and received a B.Sc. degree in Physics (1993), a B.Sc. degree in Electronics Engineering (1996), and a Ph.D. degree in Electronics Engineering (2003) from the Universitat de València. He is currently an associate professor in the Department of Electronics Engineering at the Universitat de València, where teaches analysis of circuits and linear systems, introduction to programmable logical devices, digital electronic systems and microprocessor electronic systems. His research interests include the development of machine learning algorithms for signal and image processing. Visit http://gpds.uv.es/~jordi/ for more information.

Mikhail Kanevski received the Ph. D. degree in plasma physics from the Moscow State University in 1984 and Doctoral theses in computer science form the Institute of Nuclear Safety (IBRAE) of Russian Academy of Science in 1996. Until 2000, he was a Professor at Moscow Physico-Technical Institute (Technical University) and head of laboratory at the Moscow Institute of Nuclear Safety, Russian Academy of Sciences. Since 2004, he is a professor at the Institute of Geomatics and Analysis of Risk (IGAR) of the University of Lausanne, Switzerland. He is a principal investigator of several national and international grants.

His research interests include geostatistics for spatio-temporal data analysis, environmental modeling, computer science, numerical simulations and machine learning algorithms. Remote sensing image classification, natural hazards assessments (forest fires, avalanches, landslides) and time series predictions are the main applications considered at his laboratory.

Gustavo Camps-Valls was born in València, Spain in 1972, and received a B.Sc. degree in Physics (1996), a B.Sc. degree in Electronics Engineering (1998), and a Ph.D. degree in Physics (2002) from the Universitat de València. He is currently an associate professor in the Department of Electronics Engineering at the Universitat de València, where teaches electronics, advanced time series processing, and machine learning for remote sensing. His research interests are tied to the development of machine learning algorithms for signal and image processing with special focus on remote sensing data analysis. He conducts and supervises research within the frameworks of several national and international projects, and he is Evaluator of project proposals and scientific organizations. He is the author (or co-author) of 70 international journal papers, more than 80 international conference papers, several international book chapters, and editor of the books “Kernel methods in bioengineering, signal and image processing” (IGI, 2007) and “Kernel methods for remote sensing data analysis” (Wiley & sons, 2009). He is a referee of many international journals and conferences, and currently serves on the Program Committees of several International Conferences. Since 2007 he is member of the Data Fusion technical committee of the IEEE Geoscience and Remote Sensing Society, and since 2009 he is member of the Machine Learning for Signal Processing Technical Committee of the IEEE Signal Processing Society. Visit http://www.uv.es/gcamps for more information.