Remote sensing image segmentation by active queries

Devis Tuia\textsuperscript{a,}\textsuperscript{*}, Jordi Muñoz-Marí\textsuperscript{b}, Gustavo Camps-Valls\textsuperscript{b}

\textsuperscript{a} Geographical Information Systems Laboratory (LaSIG), Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
\textsuperscript{b} Image Processing Laboratory (IPL), Universitat de València, Spain

1. Introduction

Remotely sensed images allow Earth Observation (EO) with unprecedented accuracy. New satellite sensors acquire images with high spectral and spatial resolution, and the revisiting time is constantly reduced. As a consequence, the accessibility to digital information has been enhanced and larger archives of digital remote sensing imagery are available for the user. The characteristics of the acquired multidimensional images enhance the characterization, identification and classification of the landcovers [34].

Besides sensor developments, classification algorithms have become more and more powerful, and can now treat most of the classification problems efficiently [7,41]. Nonetheless, most of these methods require supervision, i.e., the definition of a training set of labeled pixels that is used to train the model and learn the relationships between the spectra and the class labels. The definition of such a set may be difficult and time-consuming, especially when dealing with very high resolution (VHR) or hyperspectral images. For the former, the high level of detail makes the definition of representative pixels difficult, since the intraclass variance strongly increases with the finer spatial resolution. For the latter, the greater spectral detail makes choosing the relevant (and representative) pixels a complicated task in the high dimensional space concerned. Relevancy of the training set is thus an issue. In this optic, the user can decide to rely on image segmentation (either referred to as unsupervised or clustering methods) [18,20,27]: in a clustering setting, no labeled information is provided by the user and the model performs data partitioning using the observed features only. Two main approaches to clustering are found in the literature: partitioning methods that split the feature space into distinct regions and hierarchical methods [9], returning a hierarchical description of the data in the form of a tree or dendrogram.

Partitioning methods are the most studied in remote sensing: first, fuzzy clustering has been used in conjunction with optimization algorithms to cluster landcover regions [28,29]. This approach is extended to multiojective optimization in Bandyopadhyay et al. [4] where two opposing objective functions favoring global and local partitioning were used to enhance contextual regularization. Other regularization strategies include the fusion of multisource information [3,35], rule-based clustering [5] and Markov random fields [31]. Optimization using simulated annealing is considered in Hofmann et al. [19], while graph partitioning using normalized cuts is proposed in Shi and Malik [37] to optimize the cost of data clustering. Graph cuts have been reconsidered more recently [44] where the authors proposed a multistage technique cascading two clustering techniques, graph-cuts and fuzzy c-means, to train the expectation maximization (EM) algorithm. Hierarchical methods have been studied in order to include contextual constraints into the – basically spectral – hierarchical linkage [45]: multistage restricted
methods are proposed in [21,22] in order to perform first a region
growing segmentation to ensure spatial contiguity of the seg-
ments, and then to perform classical linkage to merge the most
similar segments. In Marcal and Castro [26], this idea is further
developed by adding to the aggregation rule several informative
features: spatial criteria accounting for cluster compactness,
cluster size, and the part of their boundaries that two clusters
share. Fractal dimension is used as a criterion of homogeneity in
Baatz and Schäpe [2]. Watershed segmentation on the different
resolution is proposed in Yang et al. [46] and in Hasanazadeh and
Kasaei [17], where it is used in conjunction with fuzzy partition-
ing to account for connectedness of segments. In Bruzzone and
Carlin [6], the concept of hierarchical segments has been used to
retrieve features at different scales for SVM classification.

Since clustering methods do not require supervision, they are
naturally adapted to solve remote sensing problems. However,
they are much more difficult to calibrate than supervised meth-
ods especially because the final aggregation is decided a posteriori
by the user, either by finding an appropriate number of clusters or
by deciding the level of pruning of a similarity tree. For both
cases, few robust criteria are available and the user typically turns
to heuristical criteria. Moreover, the outputs of clustering meth-
ods are often difficult to interpret since there is no explicit link
between the clusters found by the model and the classes desired
by the user. The former describes data similarities and the latter
are semantic interpretations of the objects of interest. Thus, the
resulting clusters may represent mixed semantic properties of the
scene, harming the labeling of the final classification map.

However, if a clustering model could achieve partitionment of
the data points matching the user’s required semantics, the
resulting model would join the precision of supervised methods
and the efficiency of unsupervised methods. Recent research
addresses this possibility by defining effective markers to classify
images through segmentation: in Tarabalka et al. [38], the authors
propose to use the most confident pixels classified by a commit-
tee of models as markers for segmentation. In a follow up article
[39], the authors use SVM posterior probabilities as markers. Even
if they provide very accurate classification results, these strategies
can be questioned since they still require the output of one or
several supervised models to initialize the segmentation. In this
work, we propose an intuitive way to consider this critical
problem for remote sensing image segmentation: we propose to
use semi-supervised labeling of the clustering result, similar to
what proposed for image database characterization in Grira et al.
[16]. By semi-supervised, we mean that we exploit the cluster
structure of data and the few labeled pixels provided by the user
all together to rank the remaining pixels to be labeled and
retrieve an optimal segmentation. In this paper, we exploit an
approach to learn the optimal data structure through efficient
labeling of an unsupervised classification result. As in Dasgupta
and Hsu [13], we convert the hierarchical clustering tree labeling
issue into a graph partitioning problem. Based on a hierarchical
segmentation of the image provided by the user, the algorithm
queries pixels from the most uncertain clusters and, depending on
the response of the user, decides whether they should be divided
into their subclusters or not: by interacting with the user, the
machine can find the best clustering of data and label the
resulting clusters optimally. Proceeding this way, the number of
clusters does not need anymore to match the number of classes
desired: the only requirement is the class consistency of every
single cluster.

To keep the classification routines approach fast and efficient,
an user may also decide to act on the selection of the training
pixels. If the data show a strong cluster structure, its considera-
tion in the choice of the queries allows to speedup convergence.
To perform such an inclusion, one may recur to active learning
building the training set by sampling the pixels showing maximal
uncertainty or information. By integrating these pixels, the efforts
are focused on confictive areas and the size of the hypothesis
space is thus reduced efficiently [36,40]. Active methods begin to
be considered in biomedical applications [12,32], image retrieval
[10,24,47], and in the remote sensing community [30,23,33,42],
but their application has been limited to supervised methods. For
a survey of applications of active learning algorithms in remote
sensing, refer to Tuia et al. [43]. However, this type of approaches
shows three major drawbacks for image segmentation: first, they
need a model running iteratively to rank the unlabeled pixels,
which is computationally very intensive, especially for nonlinear
models. Secondly, they need an initial training set that roughly
accounts for the data structure; and third, they disregard the
structure of the data: if the initial training set omits an important
part of the data structure, it will be difficult to discover by the
algorithm. In this paper, a different view on active learning for
image segmentation is proposed: unlike Tarabalka et al. [38,39],
we use active queries to label the hierarchical segmentation
result, without recurring to any supervised model. Several active
strategies to query the segments of a remote sensing image are
studied. With respect to the original method proposed by
Dasgupta and Hsu [13], we propose an alternative active segment
selection strategy along with a new active strategy to retrieve the
most informative pixels into this segment. The cost of dividing a
segment into subsegments is assessed through error bounds
and then the best segmentation is provided to the user. The
active segmentation strategy studied is independent from the
method used to cluster the data, and it does not need any training
or initial labeled set. Consequently, the proposed approach is
computationally very efficient for active remote sensing image
segmentation. In this sense, it represents a valid alternative
to traditional, supervised, active learning algorithms. We
tested the proposed method in a variety of settings for two
challenging remote sensing classification scenarios: hyperspectral
and VHR multispectral image segmentation. Spatial regulari-
zation is also studied, along with visual properties of the
classification maps.

The remainder of the paper is organized as follows. Section 2
briefly fixes the notation and present the prune-and-label frame-
work. In Section 3, different strategies for active pruning are
presented. Section 4 presents the data and the experimental setup
of the experiments considered in Section 5.

2. Cluster-based segmentation

In this section, the strategy to prune a hierarchical segmenta-
tion considered is presented. The aim of this strategy is to find the
best segmentation result from a pre-computed hierarchical clus-
tering by exploiting iterative queries to the user. At each iteration,
the pixels in each segment are assigned to the class that best
matches the observed labels (note that many clusters may be
labeled into the same class).

The main feature of the approach is that the labeled pixels
provided iteratively by the user increase the confidence about the
data structure: at the beginning, the algorithm is not confident
enough to assess which are the coherent clusters in the image,
thus a single cluster is accepted as a valid result. As long as
labeled pixels are provided, the algorithm can assess whether a
cluster is more coherent as it is or if the division in its two direct
children would provide more coherent clusters (in terms of class
membership). If the latter is true, the cluster structure is updated
and the mixed cluster is divided into its children.
The next sections detail this principle that leads to the prune-and-label strategy proposed.

2.1. Hierarchical image segmentation

A dataset can be described in an unsupervised manner by a hierarchical similarity tree, also known in statistical literature as dendrogram. Such a tree can be obtained by bottom-up (using hierarchical agglomeration, i.e., linkage) or in a top-down (using k-means iteratively) clustering.

The tree represents the multiscale structure of similarity between the samples. A 2-D example is given in Fig. 1: the data are merged at different levels according to their similarity, which is represented in the figure by the height of the tree. The tree itself can be described as a series of nodes \( v \), represented by the red numbered circles. The similarity between nodes is represented by a measure of distance. Each node has a size \( n_v \) accounting for the number of gathered data points or leaves. In the particular example of Fig. 1, \( n_{18}=3 \) and \( n_{17}=7 \). The root of the tree, \( v_{19} \), regroups the whole image and consists of \( n \) leaves \( \{w_{n-1}^{m}\} \in \mathbb{R}^n \). A complete tree is made of \( 2n-1 \) nodes, accounting for both the leaves and the upper nodes. A cut \( V \) on the tree is a complete partitioning of the dendrogram. It corresponds to the pruning of the tree into a series of nodes \( v \) represented by the height of the tree. The tree already represents the ensemble of the leaves. Depending on the depth of the tree pruning, the solution ranges from under to oversegmentation. During the learning process, different data points are queried and labeled by the user. For a given node \( v \), we define the number of labeled points in the node as \( n_v^l \) and the number of unlabeled points as \( n_v^u = n_v - n_v^l \).

2.2. Labeling a hierarchical tree: a pruning problem

As defined above, different cuts in the tree will provide different clustering of the data. In Fig. 1, three cuts of the tree are proposed. In traditional clustering, the concepts of over- and under-segmentation are used to describe the adequacy of the clustering obtained. In the first, the tree has been pruned excessively and a single object of the image is divided into different subregions; in the latter, several objects have been merged into a single cluster. The adequacy of clustering is very often assessed using cluster validity indexes [15,27]. In the setting proposed in this paper, cluster validity is assessed using the adequacy of the pruning to the classes expected by the user, as explained below.

An image segmentation problem consists of assigning pixels into \( c \) classes of interest accurately. These classes can be defined \textit{a priori} by the user or pre-defined by the problem (e.g., in land-use segmentation classes may be defined by the sensor). In any case, the aim is to classify the \( n_v \) pixels belonging to a node \( v \in V \) into a class \( y_v = l \), with \( l = 1, \ldots, c \). Since the tree already represents the different (multiscale) structures seen in the data, the optimal pruning \( V_{opt} \) should minimize the classification error on the leaves, and thus match the semantic classes desired by the user. By assessing the classification error, the adequacy of the current pruning can be estimated and the tree is pruned as long as dividing a cluster results in a decrease of the error.

The main problem is that no labels are available to evaluate the classification error and hence assess the adequacy of the current pruning. However, such labels can be obtained by user-machine interaction: the machine proposes to the user a list of pixels to be labeled and adapts the cut of the tree and the labeling of the nodes of the current pruning according to the received answer. This way, the tree structure and labeling are modified iteratively according to the semantic the user is looking for. The algorithm profits simultaneously from the data structure and from the expected class structure.

2.3. Prune-and-label algorithm

A way to quantify the interest of pruning a node into its subnodes is required. Consider a node \( v \), where a certain amount of pixels have already been sampled by the user. The relative proportions of the labels into each node \( v \) of the current pruning are \( p_{c,v} \), with \( l = 1, \ldots, c \). The label assigned to the leaves of this node will be the one maximizing this quantity. Similarly, the error of assigning the label \( l \) to the node can be assessed using the number of errors committed on the visited leaves:

\[
\epsilon_v = 1 - \max_l \{p_{c,v}\}
\]

(1)

Since the number of visited nodes is often limited (in particular at the beginning of the process), the confidence in the assignment of the label is often very poor. To correctly prune and label the tree, two concepts become crucial: the confidence and the admissibility of a label assignment.

2.3.1. The confidence on label assignment

Relying on observed probabilities could lead to biased conclusions, especially when few points have been sampled within a cluster. In order to assess the confidence of the label attributed to a cluster, one can use generalization bounds on the observed proportions. The estimated upper confidence bound \( p_{l,v}^{UB} \), and respectively the lower confidence bound \( p_{l,v}^{LB} \), can be formulated.
Therefore, the node of the minimal cuts of the two children of the node after pruning.

Subnodes $V$ among all the combinations of admissible labels $L$ should be considered. To penalize unlikely labelings, the focus can be put on admissible $(v)$ couples only, i.e., labels that make sense for a node, given the observed leaves. A label can be considered admissible for a given node if

$$1 - p_{v,l}^{UB} < 2 (1 - p_{v,l}^{LB}), \quad \forall l \neq l$$  

In other words, $I$ is an admissible label for node $v$ if the highest achievable error with such labeling, i.e., $1 - p_{v,l}^{UB}$, is smaller than twice the least achievable for any other class [13]. Note that several classes can be considered as admissible for the same node if they satisfy Eq. (4).

Once all the admissible couples $(v,l)$ have been found, their estimated error, or cost, is evaluated as

$$\hat{\epsilon}_{v,l} = \begin{cases} 1 - p_{v,l}^{LB}, & \text{if } (v,l) \text{ is admissible} \\ 1 & \text{otherwise} \end{cases}$$  

By doing so, the cost of an uncertain cluster will be higher than the one of another that has been extensively sampled. The optimization problem aims at finding the optimal pruning $V_{opt}$

$$V_{opt} = \arg \min_{V} \{ \hat{\epsilon}(V,L) \}$$  

where $\hat{\epsilon}(V,L)$ is the error related to a given pruning $V$ with nodes labeled according to admissible labels $L$. In other words, the selected pruning is the one minimizing the estimated error among all the combinations of admissible labels $L$ and prunings $V$. It can be shown [see [13]] that for a given node $v$ and its subnodes $vl$ and $vr$, this is equivalent to finding the minimum between the adjusted estimate of the error of Eq. (5) and the sum of the minimal cuts of the two children of the node after pruning. Therefore, the node $v$ is pruned only if the following inequality holds for all possible labels $l \in c$ and if at least one label is admissible for $v$, that is

$$\hat{\epsilon}_{v,l} > \hat{\epsilon}_{v,l} + \hat{\epsilon}_{v,r}$$  

In other words, if the pruning of a node into its two subnodes decreases the global cost, the tree is modified accordingly.

The prune-and-label routine is applied iteratively, sampling one (or several) pixels at each iteration. Once the user has returned true labels of the pixels, the proportions and admissibility of all the nodes are computed. With a bottom-up pass through the tree, all the costs are evaluated and Eq. (6) is minimized efficiently.

### 2.4. Computational complexity

The algorithm proposed is based on the two-steps procedure illustrated in Fig. 2. The computational load can thus be divided into two parts. The first contribution to the cost is related to the definition of the cluster structure: this cost depends on the clustering algorithm used and can thus be strongly reduced, depending on the algorithm selected. In this paper, we used the complete-link hierarchical clustering [45], which is at most $O(n^2 \log n)$ complex [1]. Note that this part is computed only once. After that, the iterative query part is linear, at most $O(2n)$ complex per iteration. This cost is related to the descent to find the node to label and the update of probabilities of the nodes including the labeled pixel. As the tree is pruned into subnodes, the cost per iteration decreases, as only subtrees $n_i$ are recurred. Summarizing, the global cost of the proposed approach with linkage clustering is at most $O(n^2 \log n)+\sum_{i=1}^{n} \ast O(2n_i)$, if we consider $\gamma$ queries.

This is very attractive compared to traditional active learning strategies, that re-train the classifier at each iteration. For these, the global complexity is at least $\gamma \ast O(C)$, where $O(C)$ is the complexity of the classifier, that increases during the iterations as labeled pixels are added to the training set. We say at least because the active learning heuristics can add computational cost if subroutines involving distance computation or clustering are used. Depending on the classifier, the base cost can be high: in the case of SVM, the cost is at least $O(n^2)$.

### 3. Active queries on hierarchical trees

Given the prune-and-label strategy described above, the selection of the node to sample from and of the leaf inside that node become...
crucial. This section presents active strategies to perform these selections efficiently. Fig. 2 summarizes the general framework.

3.1. Selecting a good node

One way to increase the efficiency of the algorithm is to define new strategies for the selection of the node \( v^* \) to sample from. This selection can be optimized by selecting \( v^* \) taking into account the confidence on its label.

A first, trivial strategy would be to merely sample from the node showing the largest number of leaves, i.e., to sample according to the node size \( n_v \) (CS-s0 hereafter). This strategy roughly corresponds to random sampling and will be used as comparison in the experiments below. At each iteration \( v \), the node to be sampled \( v^* \) is selected with probability proportional to the node size. We use proportionality, rather than maximization, to allow the model to make mistakes and thus discover new clusters that are not the largest.

More sophisticated strategies can be used by accounting for the bounds of Eq. (3). In an active learning setting, we would like to take advantage of uncertainties in the class labels and to avoid sampling from clusters whose class membership seems already clearly discovered. Since the bounds are constantly updated while sampling, their values can be used to direct the sampling towards the clusters showing maximal uncertainty.

The first approach (CS-s1 hereafter) selects the node to be sampled proportionally to its size and the certainty of its label, assessed by the lower bound on the winning class [13]: If the lower bound on the winning class of the cluster is small, the cluster will be sampled with high probability. This strategy takes advantage of large, pure, clusters: as soon as they are considered pure enough, the sampling is directed elsewhere. Since it still considers proportionality to the cluster size, this strategy remains conservative, in the sense that it guarantees, besides active learning, screening the larger clusters that may hide unseen data structures.

A second and more focused strategy (CS-s2 hereafter) consists of excluding the size term of the s1 strategy. In this case, as soon as a cluster is considered uncertain, it is divided and sampled until the purity of the subtree is considered satisfactory. This strategy treats extensively the uncertain clusters, by dividing them as long as the substructures are not stable in terms of attributed labels. The approach thus focuses on the most uncertain subtree increasing the number of cuts, but without coming back to a larger cluster whose uncertainty is yet to be discovered. As a consequence, this strategy may result in slower convergence in the first iterations, where a part of the data structure remains unseen. It should be preferred in situations where the user is interested in a rapid resolution for specific areas, rather than in global performance. Table 1 summarizes the sampling strategies considered in the experiments.

3.2. Selecting a leaf inside the selected node

Once the node to sample from \( v^* \) has been selected, a data point must be queried from the subtree rooted at \( v^* \). The choice of the data point can be seen as the problem of descending into the subtree rooted at \( v^* \) until a node containing a single sample is found. This data point is the one presented to the user for labeling. Since the tree is binary, each descent step consists in selecting between the left child node \( v_l \) and the right child node \( v_r \). Similar to the s0 strategy presented above, a random strategy to perform such choice is to choose the child node proportionally to the number of unlabeled points found in the left child node \( v_l \).

We will refer to this descent strategy as ‘CS-s1|d0’, where the [ ] symbol represents one of the node selection strategies presented in Section 3.1.

As for the node selection strategies, this descent can also be made active by taking into account the probability bounds. Similar to the s1 strategy presented above, the \( p_{lb}^{d0} \) bound can be exploited to direct the descent. Moreover, since this probability bound is updated from the leaves to the root at each iteration, up-to-date probabilities for all the child nodes \( v_l \) and \( v_r \) are available at each iteration. By using this last criterion, the probability to sample from a conflictive region of the subtree rooted at \( v \) is increased. We will refer to this descent strategy as ‘CS-sj|d1’. Table 1 summarizes the six sampling strategies considered in the experiments below.

4. Data and experimental setup

This section summarizes the remote sensing data used in the experiments and describes the conducted experiments. Three different remote sensing images are considered, accounting for the most common situations in the field: hyperspectral and very high spatial resolution. In addition, we will analyze the effect of introducing spatial regularization, not only to improve classification but to more interestingly demonstrate the advantage of active sampling under different yet meaningful tree structures.

4.1. Images considered

The proposed methodology has been tested on two challenging remote sensing image classification scenarios: that is, working with high dimensional hyperspectral images and with very high spatial resolution (VHR) images. The first is characterized by hundreds of narrow spectral channels, while the latter shows submetric spatial resolution with smaller spectral detail. The images considered in the experiments account for the following specific settings (see Fig. 3):

- **Hyperspectral1**: A 220-bands AVIRIS image taken over Indiana’s Indian Pine test site in June 1992. The image is 145 × 145 pixels, contains 16 crop type classes, and a total of 10,366 labeled pixels (see Table 2). This image is a classical benchmark to validate model accuracy and constitutes a very challenging classification problem because of the strong mixture of the classes’ signatures and unbalanced number of labeled pixels per class. The image can be obtained from [https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html](https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html).

- **Hyperspectral2**: The second test dataset considers an image acquired in 1999 by the HyMap airborne spectrometer over Barrax (Spain) during the DAISEX99 campaign. The image has 128 bands in the region 0.4–2.5 \( \mu \m \) and a spatial resolution of 5 m. The six classes of interest are: ‘Corn’, ‘Sugar beets’, ‘Barley’, ‘Wheat’, ‘Alfalfa’, and ‘Soil’ [8] (see details in Table 3).

- **VHR**: The last case study is a 4-bands multispectral image of the city of Zurich, Switzerland, representing a residential suburban area in the West part of the city. The image, acquired by the sensor QuickBird in 2006 has size 828 × 889 pixels. The original image has been pansharpened using Bayesian Data Processing on the original image.

### Table 1

<table>
<thead>
<tr>
<th>Subnode selection proportional to</th>
<th>Node selection proportional to</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_v )</td>
<td>((1-p_{lb})^{m_v}n_v )</td>
</tr>
<tr>
<td>((1-p_{lb})^{m_v}n_v )</td>
<td>((1-p_{lb})^{m_v}n_v )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( n_v )</th>
<th>( s0d0 )</th>
<th>( s1d0 )</th>
<th>( s2d0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>((1-p_{lb})^{m_v}n_v )</td>
<td>( s0d1 )</td>
<td>( s1d1 )</td>
<td>( s2d1 )</td>
</tr>
</tbody>
</table>
Fusion [14] 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 (see Table 4). For this last image, additional experiments have been led on the detail highlighted in Fig. 3c, with the (almost) exhaustive ground information shown in Fig. 8a.

### 4.2. Experimental setup

In the experiments presented in this paper, the hierarchical representation of the images is obtained using linkage clustering [45] with squared Euclidean distance $d^2(\psi_i, \psi_j)$ and Ward's agglomeration criterion. To limit the computational load of linkage, 10,000 pixels have been considered in each experiment. Prune and label algorithm has been implemented starting from the C code of Dasgupta and Hsu [13], available at http://cseweb.ucsd.edu/~djhsu/. To test robustness of the algorithm to different initial conditions, we varied the seed of the selection of the pixels within the clusters, resulting into 20 different sampling schemes.

<table>
<thead>
<tr>
<th>Class</th>
<th>Labeled pixels</th>
<th>Legend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfalfa</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>Corn-notill</td>
<td>1434</td>
<td></td>
</tr>
<tr>
<td>Corn</td>
<td>834</td>
<td></td>
</tr>
<tr>
<td>Grass/Pasture</td>
<td>497</td>
<td></td>
</tr>
<tr>
<td>Grass/Trees</td>
<td>747</td>
<td></td>
</tr>
<tr>
<td>Grass/Pasture-mowed</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Hay-windrowed</td>
<td>489</td>
<td></td>
</tr>
<tr>
<td>Oats</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Soybeans-notill</td>
<td>968</td>
<td></td>
</tr>
<tr>
<td>Soybeans-min</td>
<td>2468</td>
<td></td>
</tr>
<tr>
<td>Soybeans-clean</td>
<td>614</td>
<td></td>
</tr>
<tr>
<td>Wheat</td>
<td>212</td>
<td></td>
</tr>
<tr>
<td>Woods</td>
<td>1294</td>
<td></td>
</tr>
<tr>
<td>Bldg-Grass-Tree-Drives</td>
<td>380</td>
<td></td>
</tr>
<tr>
<td>Stone-steel towers</td>
<td>95</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Classes and number of samples of the ground truth of the AVIRIS Indian Pines dataset.

<table>
<thead>
<tr>
<th>Class</th>
<th>GT pixels</th>
<th>Legend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>52,813</td>
<td>Dark green</td>
</tr>
<tr>
<td>Vegetation</td>
<td>12,347</td>
<td>Light green</td>
</tr>
<tr>
<td>Highway</td>
<td>28,827</td>
<td>Black</td>
</tr>
<tr>
<td>Asphalt</td>
<td>43,005</td>
<td>Brown</td>
</tr>
<tr>
<td>Residential buildings</td>
<td>78,018</td>
<td>Yellow</td>
</tr>
<tr>
<td>Commercial buildings</td>
<td>25,389</td>
<td>Red</td>
</tr>
<tr>
<td>Shadows</td>
<td>14,071</td>
<td>Cyan</td>
</tr>
</tbody>
</table>

Table 3: Classes, number of samples and colors legend for Fig. 3e of the DAISEX image of Barrax.

<table>
<thead>
<tr>
<th>Class</th>
<th>GT pixels</th>
<th>Legend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>52,813</td>
<td>Dark green</td>
</tr>
<tr>
<td>Vegetation</td>
<td>12,347</td>
<td>Light green</td>
</tr>
<tr>
<td>Highway</td>
<td>28,827</td>
<td>Black</td>
</tr>
<tr>
<td>Asphalt</td>
<td>43,005</td>
<td>Brown</td>
</tr>
<tr>
<td>Residential buildings</td>
<td>78,018</td>
<td>Yellow</td>
</tr>
<tr>
<td>Commercial buildings</td>
<td>25,389</td>
<td>Red</td>
</tr>
<tr>
<td>Shadows</td>
<td>14,071</td>
<td>Cyan</td>
</tr>
</tbody>
</table>

Table 4: Classes, number of samples and colors legend for Fig. 3f of the QuickBird dataset.

Fig. 3. Datasets considered in the paper: (a) Indian Pines dataset, Indiana, USA; (b) HyMap image of the test set of Barrax, Spain; (c) QuickBird scene of suburbs of Zurich and detail used in the experiment presented in Section 5.2.4; (d)-(f) illustrate the ground survey data available. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
of the same hierarchical structure. The results are described using the following measures:

- **Observed performance**: The accuracy occurring while labeling all the pixels in a cluster with the winning label.
- **Number of cluster pruned**: This measure shows how far the hierarchical tree has been pruned and shows the complexity of the final clustering.

For the VHR images (QuickBird), experiments studying the effect of spatial regularization have been led by stacking the reflectance values of the pixels with the ones of their eight closest spatial neighbors (see Sections 5.2.2 and 5.2.4): this way, spatial neighborhood of the pixels is taken into account. The dimensionality of the input space increases from $\mathbb{R}^4$ to $\mathbb{R}^{36}$. For the hyperspectral images, these experiments have been omitted, since the dimensionality of the input space would largely increase.

## 5. Results and discussion

This section presents the results of the prune-and-label active learning algorithm proposed on the datasets presented in the previous section.
5.1. Hyperspectral imagery

Results on the AVIRIS and HyMap dataset are reported in the two first lines of Fig. 4. For all the sampling strategies selected, there is a strong decrease in the classification error in the first iterations, since the algorithm ‘discovers’ the class structure with the first queries. The advantage of selecting actively the node to sample from can be seen from the first iterations for the CS-s1.
strategy, that smoothly directs the sampling toward uncertain big clusters. The gain in accuracy increases constantly with respect to the CS-s0 strategy and saturates in the last iterations, where practically all the available pixels were sampled. Regarding the CS-s2 strategy, a more difficult start is observed for both experiments: since this strategy concentrates on uncertain areas, without ensuring a diverse sampling (encoded by the constraint on the size of the nodes $n_s$ in the s1 strategy), as soon as a mixed pixel is selected, the algorithm prunes as long as it is necessary to solve this part of the hierarchical tree. This results in a slower convergence than random in the first iterations; nonetheless, when the algorithm discovers a new part of the tree, the overall classification error rapidly decreases and the algorithm converges to an optimal solution slightly outperforming the CS-s1 strategy. This shows that this sampling strategy efficiently solves the problematic areas without wasting queries on larger clusters showing less uncertainty. With respect to the descent strategies, the active descent ‘d1’ is more effective than the random ‘d0’ in all the experiments proposed. Overall, the CS-s2 strategy partitions the hierarchical tree in a focused manner: as soon as a cluster is divided into two, the CS-s2 strategy focuses on the most uncertain subtree, thus increasing the number of cuts, but without coming back to larger cluster whose uncertainty is yet to be discovered. The number of clusters reported in Fig. 4 confirms these observations: the CS-s2 strategy quickly divides the set into a greater number of clusters compared to the two other strategies. Note that the number of clusters of the CS-s1 is greater than the one of the CS-s0, thus showing that the method focuses on uncertain clusters, rather than querying the largest cluster without maximizing the probability of a cut. The effect of the ‘d1’ descent can also be seen in these figures: by driving the selection of the query into the uncertain areas of the node, such descent allows more divisions (see the higher number of clusters in Fig. 4b) and a faster increase of the accuracy.

5.2. VHR imagery

The experiments carried out on the VHR image of Zurich confirm most of the observations made above. The curves of last row of Fig. 4 show the effective decrease of uncertainty of the active methods: after an initial deep increase of the accuracy, the slope of the random sampling decreases, certainly because of the contradictory pixels, related to the strong intraclass variance of the classes. However, the active strategies efficiently minimize the error, showing the same behavior observed in both the AVIRIS and HyMap images, with a better performance for the CS-s2, that solves the mixed clusters by isolating the typal subclusters found from the rest. Again, the active descent ‘d1’ strategy leads to more effective results outperforming the CS-s1 strategy by 2–3%. CS-s1 is immediately effective (in this case, already after 180 queries in terms of observed accuracy), while CS-s2 repeats the behavior observed above and needs about 700 queries to outperform random sampling. However, the slope of the learning curve is steeper for the latter strategy that, starting from 1000 queries, shows the best performance.

5.2.1. Per class sampling

Other interesting observations can be done by considering the evolution of the sampling of a single experiment: Fig. 5 illustrates, for each iteration, the percentage of the available pixels of a class that has been labeled. CS-s0 samples according to the size distribution of the classes, since it samples always from the bigger cluster. Therefore, the share of the available pixels grows linearly during the iterations, as shown in the first row of Fig. 5. The CS-s1 strategy shows three sampling phases: the first one, occurring between iterations 1 and 3000, ignores the classes ‘Trees’ and ‘Shadows’ and gives preference to the four others. In a second phase (3000→6000 for the ‘d0’ descent and from 2000 for the ‘d1’), the model treats in detail the classes ‘Commercial’ and ‘Roads’, that are spectrally very close to their counterparts ‘Residential’ and ‘Highway’. In the last phase, the remaining pixels of shadow and tree are sampled: this is due to the fact that the available pixels from the other classes have already been sampled and only pixels of these classes remain. Moreover, plots of last line of Fig. 4a show that at this point the active strategy has almost reached convergence. The CS-s2 (last row of Fig. 5) shows a more varying behavior: each time a mixed cluster is found, its inner structure is studied in detail until a more stable representation is found according to the pixels known. Right from the beginning, CS-s2 over samples the class ‘Commercial’, ‘Meadows’ and ‘Roads’ (iterations 1→300). Then, from iteration 3000, the class ‘Meadows’ is considered solved and the sampling rate decreases abruptly and more pixels are sampled from the class ‘Highway’. As for the CS-s1 strategy, pixels from classes ‘Tree’ and ‘Shadows’ start being sampled from iteration 7000, after convergence to the optimum. Note that the class ‘Residential’ is constantly sampled all over the process.

5.2.2. About spatial regularization

Regarding contextual regularization, left column of Fig. 6 shows the results of the same three sampling strategies applied to the data stack described in Section 4.2: the first striking remark is the improvement in performance of the algorithm when adding information about the pixels’ context. This matches general knowledge in remote sensing image classification. It is important to notice that, for both the spectral (solid) and the spatially constrained (dashed) experiments, the active strategies improve the CS-s0 sampling scheme in a similar way. Using spatial regularization, overall accuracy increases of about 10% for all the experiments, but the relationships among the different sampling strategies remain unchanged. This is thus related to the better description provided by the initial hierarchical structure, that better matches the class distribution: this can also be seen in the right hand side figure in Fig. 6, where the sampling strategies need less partitionments to reach better results, thus demonstrating that the problem has become easier for the prune-and-label strategies. However, we would like to stress that the proposed strategies work in all the scenarios and that a better hierarchical representation lead to faster convergence and more compact segmentations.

5.2.3. Comparison with model-based active learning

In this section, the proposed approach is compared to two state-of-the-art model-based active learning heuristics. Results are illustrated in Fig. 7 for breaking ties (BT, [25]) using a linear discriminant analysis (LDA) classifier (left panel) and margin sampling (MS, [36]), which is specific to large margin classifier as support vector machines. In both cases, we used the 36-dimensional vectors of the QuickBird image and started with 100 labeled pixels to train the initial model; then 50 iterations adding 100 pixels per iterations are performed. In the first case, LDA cannot converge to satisfactory results, because the problem is nonlinear: the model cannot perform better than accuracies of 70%. Margin sampling performs better than the proposed approach in the first queries. However, from 1000 queries on for the passive case and from 1650 queries on in the MS case, the proposed methods perform better, since it retrieves coherent segments of smaller size, while the model-based approaches

---

1 From now on, all the results reported consider the ‘d1’ descent strategy.
work in a pixel-based manner and saturate on an 85% accuracy. The better performance of SVM in the first stage has a double explanation: first, SVM is generally robust in solving nonlinearly separable problems with little labeled information and secondly the measures of accuracy used for both approaches are different. For MS, it is a pixel-based accuracy, while for the proposed approach, average accuracies over the segments are reported. This is why errors obtained in the first iteration are higher for the cluster-based methods since large clusters have not been pruned yet. For this reason, we should stress that this comparison is merely illustrative, since in the first part of the procedure the accuracy is skewed towards the pixel-based models, while in the last it is skewed towards the cluster-based.

Regarding computational time\(^2\) (cf. Section 2.4), the proposed approach required 660 s for running the linkage clustering and 1.2 s for the 5000 queries performed. Margin sampling using SVM required 610 s for 50 iterations querying 100 pixels per iteration, 1216 s when performing 100 iterations sampling 50 pixels per iteration and 2467 s when performing 200 iterations sampling 25 pixels per iteration. For a fixed number of queries and when using a slow clustering algorithm as linkage, the execution times are comparable to those obtained by a model-based approach with a small number of iterations. However, when the number of iterations increases the proposed approach is much faster than the traditional model-based heuristics. Note that querying large sets of pixels per iteration with MS is not recommended, because the queried pixels can suffer of problems of data collinearity. As a matter of fact, MS is optimal only when a single pixel is queried per iteration, but this increases the execution time considerably [42].

5.2.4. Segmentation maps

The last experiment presented is concerned with the detail of the Zurich VHR image highlighted in Fig. 3c. A ground-truth information was provided for this image patch in order to test the algorithm in realistic conditions, where the whole image is

---

\(^2\) The computations are performed using a 2.53 GHz Intel processor Core 2 Duo and 8 Gb of RAM.
clustered. In the discussion that follows, the spatially regularized data have been used with the ‘d1’ descent strategy.

This experiment allows the visualization of the classification result: Fig. 8 illustrates the classification outputs for different numbers of labeled pixels. From the maps, it is possible to notice how the CS-s0 strategy cannot handle properly the differences between the classes in the first iterations: after a few queries, the CS-s0d1 sampling strategy cannot find the residential and asphalt classes, while both the CS-s1d1 (roughly) and CS-s2d1 (more precisely, but with more noise) do: the building class is correctly handled within the first 50 queries and the road after 100 by the active strategies.

When the general class distribution is retrieved (see the 500 queries maps shown in the last column), CS-s0d1 still provide the greater confusion in the map. Such confusion happens because there are some small clusters having spectral properties similar to meadows and shadows, but that are, in fact, tree pixels. At this stage, the CS-s0d1 strategy provides the most smooth solution.

6. Conclusions

This paper presented an active learning method for semi-supervised labeling of segmented images. Unlike methods resorting model-based heuristics, the proposed approach relies on selecting the samples that more properly describe data density looking at its cluster structure. This strategy is motivated by the cluster assumption, which is reasonable in the context of remote sensing image classification.

The method combines active supervised and unsupervised clustering coupled using a tree-based cluster representation.
pruned with an active learning strategy. No model training is required by the algorithm. Good results have been obtained when dealing with both high spectral and very high spatial image resolution, providing thus a good alternative to users confronted with the tedious and uncertainty-prone process of pixel labeling. Since only the probability between the selected leaf and the root subnode have to be updated, the cost of the algorithm is sublinear, providing thus a very fast querying algorithm. To this, one must add the time spent computing the unsupervised clustering. However, its success relies on the quality of the hierarchical clustering retrieved from the image: even if active queries outperform passive selection in all the experiments considered, it is also true that the better the hierarchical representation, the faster the convergence of the approach. This conclusion will drive our further work to study alternative ways of building the hierarchical clustering.

Acknowledgments

This work has been supported by the Swiss National Science Foundation (Grants no. PBLAPF-127713 and PZ00PF-136827), and by the Spanish Ministry of Science and Innovation under projects TEC2008-13696, AYA2008-05965-C04-03, and CONSOLIDER/CSD 2007-00018.

References

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 Ph.D. in environmental sciences at the University of Lausanne in 2009. He was a post-doc researcher at both the University of València, Spain and the University of Colorado at Boulder under a Swiss National Foundation program. He is now senior researcher at the Laboratory of Geographic Information Systems at the EPFL Lausanne. 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. 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 he 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.

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 in the Universitat de València, where he teaches electronics, advanced time series processing, and machine learning for remote sensing. He is also a leading researcher at the Image Processing Laboratory (IPL), and has been a visiting researcher at the Remote Sensing Laboratory (Univ. Trento, Italy) and at the Max Planck Institute for Biological Cybernetics (Tübingen, Germany). 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 an evaluator of project proposals and scientific organizations. He is the author (or co-author) of 100 international peer-reviewed journal papers, more than 100 international conference papers, 20 international book chapters, and editor of the books “Kernel methods in bioengineering, signal and image processing” (IGI, 2007), “Kernel methods for remote sensing data analysis” (Wiley & sons, 2009), and “Remote Sensing Image Processing” (MC, 2011). He has been recently included in the list of highly cited researchers by ISI-Thomson Reuters. He is a referee of many international journals and conferences, and currently serves on the Program Committees of International Society for Optical Engineers (SPIE) Europe, International Geoscience and Remote Sensing Symposium (IGARSS), International Workshop on Artificial Neural Networks (IWANN), Machine Learning for Signal Processing (MLSP), and International Conference on Image Processing (ICIP). Since 2007 he is the 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. He is involved in the MTG-IRS Science Team (MIST) of the European Organisation for the exploitation of Meteorological Satellites (EUMETSAT). He is Associate Editor of the “ISRN Signal Processing Journal” and “IEEE Geoscience and Remote Sensing Letters”. Visit http://www.uv.es/gcamps for more information.