Tree Based Representations For Fast Information Mining From VHR Images

Lionel Gueguen*, G.K. Ouzounis*, M. Pesaresi, P. Soille

Joint Research Centre, European Commission
Via E. Fermi, 2749, I-21027 Ispra (Va), Italy

The 8th Conference on Image Information Mining

*L. Gueguen and G. Ouzounis have been, since Sept. 2012, with DigitalGlobe Inc., Research & Development Department
Context

- Supervised classification is an interesting framework for grasping the information content of a VHR scene;
- Pixels are characterized by spectral, textural and geometric features, for classification;
- Usual classification techniques are slow when handling several millions of pixels;

Proposed solution:
- Tree-based segmentation reduces the number of elements to be handled;
- Tree-based clustering for performing fast classification, which allows for interactive image content exploration.
Outline

1. HIERARCHICAL SEGMENTATIONS
2. FAST SUPERVISED CLASSIFICATION
   - CLUSTERING BASED CLASSIFICATION
   - HIERARCHICAL CLUSTERING
   - FAST SUPERVISED CLASSIFICATION
3. EXPERIMENTS
   - HYPERSONTICAL CLASSIFICATION
   - VHR COMPONENT CLASSIFICATION
   - LEARNING FROM THIRD-PARTY SOURCE
4. CONCLUSION
Advantages of hierarchical segmentations

- It atomizes the image into homogeneous components and it reduces the number of elements in comparison to the number of pixels;
- It embeds multiple nested segmentations, from which it is easier to retrieve segments of interest;
- The segmentation is not impacted by a scale parameter.
Tree representation

- The root represents the component covering the full image domain;
- The leaves represent the smallest segments (which can be the pixels);
- A parent node is a segment formed as the union of its children segments;
- Each segment is described by a vector of spectral (average spectral response in the segment) and shape characteristics (2nd order moments).
Well-known tree representation

- **Max-Tree**: it encodes nested connected components that are brighter than their surrounding. Min-Tree encodes the connected components that are darker than their surrounding;

- **Level set Tree**: it encodes both types of components that are brighter or darker than their surrounding;

- **Alpha-Tree**: it encodes the hierarchical clustering of pixels, provided a dissimilarity measure between adjacent pixels;

- **Binary Partition Tree**: it encodes also hierarchical clustering of pixels, provided a dissimilarity measure between adjacent segments.
Outline

1. HIERARCHICAL SEGMENTATIONS

2. FAST SUPERVISED CLASSIFICATION
   - CLUSTERING BASED CLASSIFICATION
   - HIERARCHICAL CLUSTERING
   - FAST SUPERVISED CLASSIFICATION

3. EXPERIMENTS
   - HYPERSPECTRAL CLASSIFICATION
   - VHR COMPONENT CLASSIFICATION
   - LEARNING FROM THIRD-PARTY SOURCE

4. CONCLUSION
Clustering Based Classification

- Given $l$ training examples $\{\tilde{a}_i\}_{i=1}^l$ associated to $m$ classes $\{y_i\}_{i=1}^l$, $y_i \in \{1, \ldots, m\}$, train a classifier;
- Make use of the data distribution (semi-supervised classification) which can be encoded in any clustering;
- A naive Bayesian classifier is built for each cluster denoted by $N$, where the posterior probability of a random label $Y$:

$$p(Y = q \mid N) = \frac{h(y_i = q \mid \tilde{a}_i \in N) + 1}{|\{\tilde{a}_i \in N\}| + m} \quad (1)$$

where $h(y_i = q \mid \tilde{a}_i \in N)$ is the number of times the class $q$ is represented in $N$.
- The node class estimate is $\tilde{q} = \arg \max_q p(Y = q \mid N)$. 


Clustering Based Classification Properties

Advantages
- Low classification complexity for large clusters;
- Incremental with new training examples;

Drawbacks
- Optimal number of clusters to be set a priori;
- Higher classification errors;

Idea
Hierarchical clustering benefits from low complexity classification, but it gives freedom on clustering granularity.
Hierarchical clustering

- A hierarchical clustering is a tree based representation which organizes numerical elements into nested clusters;
- A set of $n$ $k$-dimensional elements $\{a_i\}_{i=1}^n$ can be organized in a tree $\mathcal{T}$;
- A node $N$ is associated to a subset/cluster of the elements $\{a_i^N\}_{i=1}^{N}$;
- As a node get further from the root, the cluster represented by the node gets smaller.
### Computing Hierarchical clustering

<table>
<thead>
<tr>
<th>Approach</th>
<th>Bottom-up</th>
<th>Top-down</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td>deterministic complete</td>
<td>low memory usage fast $O(n \log n)$ incremental</td>
</tr>
<tr>
<td><strong>Drawbacks</strong></td>
<td>memory usage slow $O(n^3)$</td>
<td>non-deterministic incomplete</td>
</tr>
<tr>
<td></td>
<td>non-incremental</td>
<td></td>
</tr>
<tr>
<td><strong>Algorithms</strong></td>
<td>single-linkage, complete-linkage ...</td>
<td>Hierarchical k-means, KD-Tree ...</td>
</tr>
</tbody>
</table>

What approach to take for handling millions of objects?
### Computing Hierarchical clustering

<table>
<thead>
<tr>
<th>Approach</th>
<th>Bottom-up</th>
<th>Top-down</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advantages</td>
<td>deterministic, complete</td>
<td>low memory usage, fast $O(n \log n)$ incremental</td>
</tr>
<tr>
<td>Drawbacks</td>
<td>memory usage, slow $O(n^3)$, non-incremental</td>
<td>non-deterministic, incomplete</td>
</tr>
<tr>
<td>Algorithms</td>
<td>single-linkage, complete-linkage ...</td>
<td>Hierarchical k-means, KD-Tree ...</td>
</tr>
</tbody>
</table>

**What approach to take for handling millions of objects?**
Fast Supervised Classification

Main Idea
- Find an initial fine clustering, such that two elements belonging to a cluster share similar features;
- Find a coarser clustering in order to maximize mutual information between the label $Y$ and the clusters $\tilde{N}$

- The main idea is in the spirit of Information Bottleneck;
- First, prune the initial hierarchical clustering with a Rate-Distortion criterion, where the distortion is given as the average dispersion per cluster. Apply a Rate-Distortion criterion in Top-Down manner;
- The Rate-Distortion depends on a trade-off parameter controlling the granularity of the leafs.
Fast Supervised Classification

- Further, prune the intermediate tree in order to increase the average mutual information between the clusters and the labels. Apply a Classification criterion in Bottom-Up manner.
- The final leaves of the tree represent a clustering which is optimized to predict the labels;
- Classify the elements of each leaf, attributing the estimated leaf class.
Fast Supervised Classification Properties

- Using a KD-Tree makes a very quick hierarchical clustering;
- As the tree is a set of clusters, the low classification complexity remains and it is still incremental;
- The trade-off enables to select the right clustering granularity;
- This setting is expected to cluster millions of elements in several minutes, while performing classification in less than a second.
Hyperspectral Classification

Description:

- ROSI hyper-spectral image of Pavia center;
- 1096 × 715 pixels × 103 bands;
- 103 bands are reduced to 6 principal components;
- 10% of the 148,152 labelled data points are used for training;
- 9 classes (Asphalt, Meadows, ...);
- CPU: Intel Xeon E5504 @2.00GHz;
- KD-Tree based classification is compared to SVM.
Hyperspectral Classification Results

<table>
<thead>
<tr>
<th></th>
<th>KD-Tree Based</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kappa</td>
<td>0.62</td>
<td>0.78</td>
</tr>
<tr>
<td>Overall Accuracy</td>
<td>0.82</td>
<td>0.91</td>
</tr>
<tr>
<td>Clustering (in s)</td>
<td>2.35 s</td>
<td>0 s</td>
</tr>
<tr>
<td>Training (in s)</td>
<td>0.07 s</td>
<td>5.33 s</td>
</tr>
<tr>
<td>Prediction (in s)</td>
<td>0.03 s</td>
<td>552 s</td>
</tr>
<tr>
<td>Total time (in s)</td>
<td>2.45 s</td>
<td>557.33 s</td>
</tr>
</tbody>
</table>
Built-up Detection in Kolkata

Description:

- WorldView-2 Multi-spectral scene of Kolkata, India;
- $16,384 \times 8820$ pixels $\times$ 4 bands (RGB,NIR1);
- The Luminance is decomposed by a Max-Tree;
- 8-dimension feature vector (area, compactness, linearity, contrast and spectral average);
- Features vector are organized in a KD-Tree, with a minimal leaf size of 256;
Interactive Built-up Detection in Kolkata

Features extraction characteristics:

- CPU: Intel Xeon E5504 @2.00GHz;
- 14,112,692 CCs are produced and characterized in 200 s;
- The features are clustered hierarchically in a KD-Tree in 140 s.

Interactive training/classification:

- 100,000 training examples from 10 ROI;
- Training the KD-Tree requires 0.5s;
- classification of 14,112,692 CCs requires 0.2s;
Close View of Built-up Detection in Kolkata

(a) Kolkata - Country side

(b) Built-up classification

**Figure**: a) A close view of the WV2 image in a country side area. b) The building detection corresponding to (a).
Close View of Built-up Detection in Kolkata

(a) Kolkata - City center

(b) Built-up classification

Figure: a) A close view of the WV2 image in a country side area. b) The building detection corresponding to (a).
Interactive Built-up Detection in Beirut

Description:

- WorldView-2 Multi-spectral scene of Beirut, Lebanon;
- $15,975 \times 8547$ pixels $\times 8$ bands;
- The spectral difference is decomposed by an Alpha-Tree;
- 12-dimension feature vector (area, compactness, linearity, contrast and spectral average);
- Features vector are organized in a KD-Tree, with a minimal leaf size of 256;
Reference data set:

- Manually digitized reference map indicating the presence of urban area;
- Reference resolution is 30m;
- Detection and reference are downgraded to 100m, before comparison.
Learning From MODIS

Description:

- 10 WV2 multispectral scenes in Lebanon;
- A MODIS 500-m Global Urban Extent layer (MODIS-GUE);
- All CCs are considered as training samples associated to labels given by (MODIS-GUE);

(a) WV2 footprints  
(b) MODIS-GUE  
(c) Learning Result
Close View on Results

(d) WV2 Zoom
(e) Learning Result

Figure: d) A close view from one of the WV2 scene over Lebanon. Credit DigitalGlobe 2008. e) The corresponding classification result by exploiting MODIS-GUE.
Computational Complexity

Time complexity to process a Multispectral tile of size 10,000 × 10,000 pixels × 8 bands (Intel Xeon E5504 @2.00GHz CPU).

<table>
<thead>
<tr>
<th></th>
<th>Time Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max-Tree</td>
<td>17.00 s</td>
</tr>
<tr>
<td>KD-Tree</td>
<td>195.00 s</td>
</tr>
<tr>
<td>Examp. collection</td>
<td>14.00 s</td>
</tr>
<tr>
<td>Training</td>
<td>1.00 s</td>
</tr>
<tr>
<td>Classification</td>
<td>0.04 s</td>
</tr>
<tr>
<td>Total</td>
<td>227.04 s</td>
</tr>
</tbody>
</table>

By extrapolation, we have a throughput of 6300 km²/hour with a single CPU.
Conclusion

- Fast classifier based on hierarchical clustering;
- Fast image decomposition into Connected Components;
- Experiments with VHR WV2 images for the detection of built-up;

Thank you for your attention.
Acknowledgement

This research was funded by the JRC Specific Programme of European Commission’s Seventh Framework Programme for Research and Technological Development (FP7). It was undertaken under the work programme of the Geo-Spatial Information Analysis for Security and Stability action, Global Security and Crisis Management unit, Institute for the Protection and Security of the Citizen.