SCALABLE MACHINE LEARNING FOR FAST THEMATIC MAPPING IN WEB SERVERS

Javier Lozano, Naiara Aginako,
Marco Quartulli, Igor G. Olaizola

Vicomtech-IK4
Digital Television
and Multimedia Services Dept.
Paseo Mikeletegi 57, 20009 Donostia
Spain

Ekaitz Zulueta

University of Basque Country
Sys. Eng. and Automation Department
Nieves Cano 12, 1006 Gasteiz
Spain

ABSTRACT

We present a web mining processing service that returns supervised probabilistic classifications of Earth Observation (EO) data in tiled form, with the aim to create user-selection based thematic maps from remotely sensed raster imagery.

User interfaces supporting interactive navigation and model training and tuning are implemented in open HTML5 standards, while software interfaces among components conform to OGC standards.

Near real time operation in the servers is attained by exploiting efficient data structures for high dimensional indexing and search.

Index Terms— Remote Sensing, thematic mapping, web based mapping systems, Visual Analytics

1. INTRODUCTION

While EO data lack in principle explicit semantic meaning, machine learning / data mining algorithms can be used to generate thematic maps from raster data and implicit semantics in the form of training.

Earth observation mining systems are the subject of current research and development efforts [1, 2, 3, 4]. Classification is an foundational methodology for assigning semantic labels to raster data records [5, 6, 7, 1]. Visual Analytics methodologies are often considered for including the expert user in the loop [2, 8].

In this contribution, we build on previous work in the domain of Earth Observation analytics and mining [9, 10].

Trying to overcome the semantic gap, we propose a prototype that merges near real time supervised classification techniques for thematic geospatial layer generation with standardized web service back-ends supporting an interactive visualization and learning interface. We consider data available on the Basque Country, Spain, in the Open Data Euskadi repository (`http://opendata.euskadi.net/`).

In the present contribution, we use a collection of twelve image products from an airborne platform which compose a 14080 x 9840 pixel submetric resolution map of Donostia - San Sebastian.

2. METHODOLOGICAL APPROACH

We try to improve on the state of the art by considering supervised classification embedded in standardized tools with the aim to obtain a efficient solution for large volumes of data.

With respect to the last point, pixel based approaches [1] require processing very large data volumes. In this sense, to get an efficient response to the queries, the data organization is critical. In particular, nearest neighbor search can benefit hierarchical indexing structures. K-d trees are space partitioning data structures for point organization in k-dimension Euclidean spaces. They are based on sets of hyperplanes each perpendicular to one of the axes of the coordinate system. All nodes in the tree, including root and leaves, store a point and a space dividing hyperplane. To find the nearest neighbors, it is necessary to define a search scope, to determine the vicinity of the points, once the K-d tree is built. In this work the scope is variable in each selected pixel and depends on the definition of the target class by the user. The key benefit is the reduction in the computational cost to find the nearest neighbor from $O(n)$ to $O(log(n))$ in the average case. This directly affects the performance when dealing with Big Data archives.

With respect to unsupervised classification approaches [3], we focus on including the user in the training process. An interactive learning scheme allows a supervisor to define positive and negative examples by interacting with a web based geovisualization interface. As is typical in Visual Analytics methodologies, interaction events directly influence a probabilistic model of the thematic class of interest that is built on top of the K-d tree indexing structures.

Finally, we base our implementation on a strict adherence to current web standards such as HTML5 and OGC services. In such an environment, optimizing performance issues related to data communication and memory footprint in the client is of foremost importance. Click-and-drag operations in the client move the map view port as is typical of HTML-based geospatial interfaces. Events that impose an extension or a recomputation of the live area under analysis are handled by spawning new processing requests to the server. The system configuration aims at reducing these requests to a minimum, while avoiding an excessive load on the client memory.

3. IMPLEMENTED SOLUTION

We test our implementation strategy by developing a prototype in a high-level open source scripting language, Python, to leverage the extensive functionality made available in packages for N-dimensional array object management (`http://www.numpy.org`), optimization, spatial data structures and efficient algorithms (`http://www.scipy.org`) and image processing and graphics (`http://effbot.org/imagingbook/pil-index.htm`).

The web architecture relies on the Flask micro-framework for serving functionality (`http://flask.pocoo.org/`), and on the TileStache map server for managing and serving thematic tiles

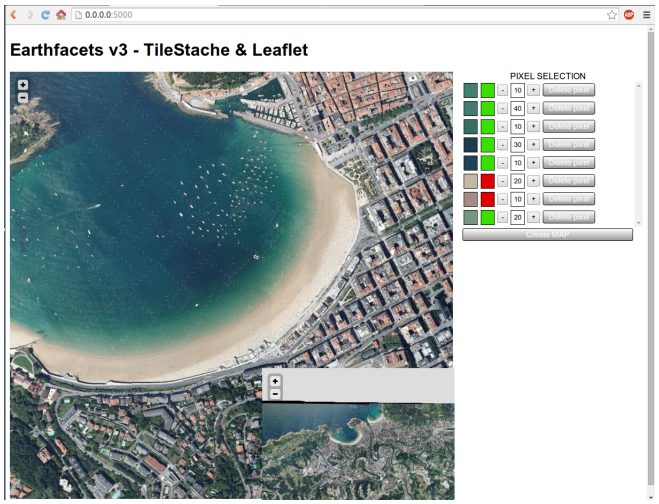

Fig. 1: User Interface. An interactive map viewer supports supervised training and output presentation. A configuration panel to the right allows the user to interactively manipulate the parameters of the learned model.

(`http://tilestache.org/`). User interface operations like zoom, drag and drop operations are made available by relying on these libraries.

The client side is composed by a web page divided in a map and a configuration panel. This interface allows the user interaction to select pixels of the map according to the semantics of the search. Configuration of model hyperparameters is available via the left pane. Based on HTML standards, the communication with the server is carried out through Asynchronous JavaScript and XML (AJAX). This way the web web can send data to the server and retrieve a request from it asynchronously, in the background.

4. PROTOTYPE OPERATION

The client interface, in figure 1, is built around an interactive map view that supports supervised training according to the semantics of the thematic class of interest and output presentation. A configuration panel presents a description of the training itself and allows the user to interactively manipulate some parameters of the learned model.

At start up time, the server creates tiles corresponding to the current active area. After this, K-d trees indexes are generated for training, and the map is presented for the user interaction. The user is able to select different training pixels according to semantic meaning of the search, and can subsequently tune the model use for realizing the query. To this end, the client transmits to the server the parameters for the model. The model is used on server to analyze and process the data: K-d tree indexes return the nearest neighbors according to it in the form of classified tiles. When all new tiles are created, the server returns to the client a layer identification number and loads the new layer in the map client.

An example of returned thematic map layer can be found in figure 2. Only pixels according to the proposed training data set are returned creating tiles.

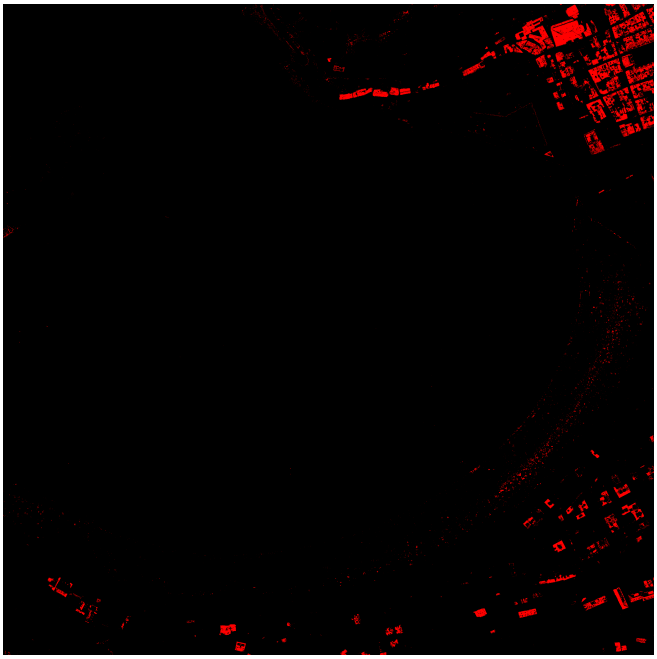

Fig. 2: Obtained supervised single class result map after processing user selected training pixels for Buildings class. Red pixels are pixels classified as beach, black ones are not classified or undefined.

Performance Statistics	Beach	Vegetation	Sea	Building	Street	Bare soil
Number of pixels	11	20	14	12	8	14

Table 1: Training volume for training data set.

5. METHODOLOGICAL EVALUATION

The first obtained results appear encouraging in terms of both visual inspection and quantitative performance evaluations. To this specific end, a test and validation area has been defined on the city of Donostia where the Vicomtech-IK4 research center is located so that field inspections can be used whenever necessary to verify the obtained results. In this area we defined six principal classes of interest: Buildings, Streets, Bare soil, Sea, Vegetation and Beach. In this section we present the results obtained in the different tests.

5.1. Information retrieval performance

A training set is selected on the map for each class according to its semantic meaning (table 1). The multi-class supervised classification process employed composes by probability multiple implicit single class thematic layers as the one shown in figure 2: it estimates and minimizes a per pixel probabilistic distance to the nearest training element in either the feature or the geographic space, resulting in a pure multiclass classification or in a multi-class classification with a significant segmentation component related to the spatial dimension

An example map obtained from the classification and merging process is presented in figure 3. As can be seen, a significant portion of the output pixels remains unclassified, particularly in the top and center with Sea class and on the bottom side with Building, Vegeta-

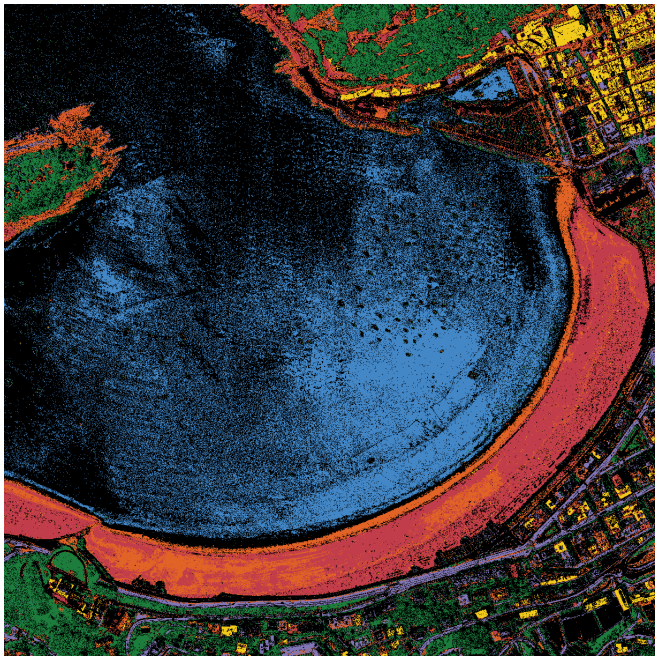

Fig. 3: Multiple Layer Class Merging Result: Building (yellow), Sea (blue), Vegetation (green), Bare soil (orange), Street (purple), and Beach (red). A significant portion of the output pixels remains unclassified (black).

Performance Statistics	Beach	Vegetation	Sea	Building	Street	Bare soil	Undefined
Precision	84,98	82,92	99,7	96,49	58,27	10,72	0,0
Recall	65,72	50,52	33,84	19,36	13,8	44,71	37,5
F1	74,12	62,78	50,52	32,25	22,31	17,30	0,0
Accuracy	87,59	84,93	48,75	83,49	76,62	84,04	39,53

Table 2: Performance measures for the training data set.

tion and Street classes. We consider this to represent a feature rather than a shortcoming in our approach. A better definition of the ground coverage classes can be obtained by extending the training, at the possible cost of a reduced Specificity. To obtain quantitative results from classified pixels we develop a ground truth image against which compare. We consider classical statistical measures of information retrieval performance like Precision, Recall, F1 and Accuracy.

Results are shown in table 2. We consider them to be encouraging: while Precision values overcomes 82 percent and Accuracy measures are around 80 percent, in most of the classes, Recall values range from about 13 percent to more than 65 percent, which implies a decrease of the F1 values with respect to good obtained Precision values. Receiver Operating Characteristic (ROC)analysis is performed for each class by means of Area Under the ROC Curve (AUC) calculation. Results shown in figure 4 are not very encouraging: only the Beach class exceeds the 0.7 value.

5.2. Indexing performance

We consider classifier implementations based on two libraries dedicated to machine learning and data mining, Scipy [11] and Scikit-learn [12]. An indexing algorithm is implemented in both libraries. We compare classifiers built on each of them and an implementation that foresees no indexing at all. In this last case, image pixels are scanned one by one during the classification procedure.

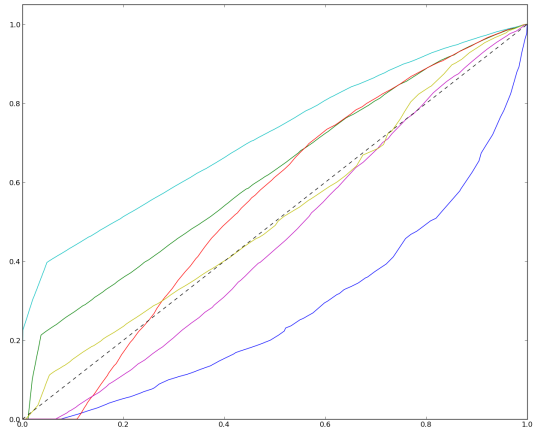

Fig. 4: Area Under Receiver Operating Characteristic (ROC) curve value comparative in the case of pure classification: Building (light blue), Sea (red), Vegetation (green), Bare soil (yellow), Street (purple), and Beach (dark blue). AUC values: Building 0.715, Sea 0.549, Vegetation 0.617, Bare soil 0.515, Street 0.454 and Beach 0.278

Performance Statistics	Beach	Vegetation	Sea	Building	Street	Bare soil	Undefined
Precision	84,98	87,56	99,70	98,53	58,27	11,01	00,00
Recall	65,72	41,99	33,83	11,92	13,80	40,78	50,00
F1	74,12	56,76	50,52	21,27	22,31	17,34	00,00
Accuracy	87,08	83,44	47,60	81,65	75,79	84,68	37,26

Table 3: Performance measures obtained with the K-d tree with batched query (Scikit-learn) engine for the training data set in the case of pure classification.

For measuring the classification engine performances in terms of processing times, we prepare two execution environments. The first is based on a traditional laptop with an Intel(R) Core(TM) i5 650 processor model running at 3.2 GHz and the second one is based on a notebook with Intel(R) Core(TM) i7- 4750HQ running at 2.00GHz and SSD Hard Drive, both with 8 GB of RAM. In each environment, we run the tests with the three indexing options.

A requirement to be able to compare different classification and indexing engines in terms of computational cost and time is to obtain comparable statistical measures of information retrieval performance. The obtained information retrieval performance values from the three algorithmic options are comparable, as per table 3 and per the comparison in figure 5.

At this point a comparison of the processing times is justified. As can be seen in table 4, the benefit of the use of indexing engine is clear. While required time for the indexing-less classifier is almost 7 hours and 45 minutes, the required time for the K-d tree-based implementation is around 25 minutes and the K-d tree batch query classifier system based on Scikit-learn only requires about a minute.

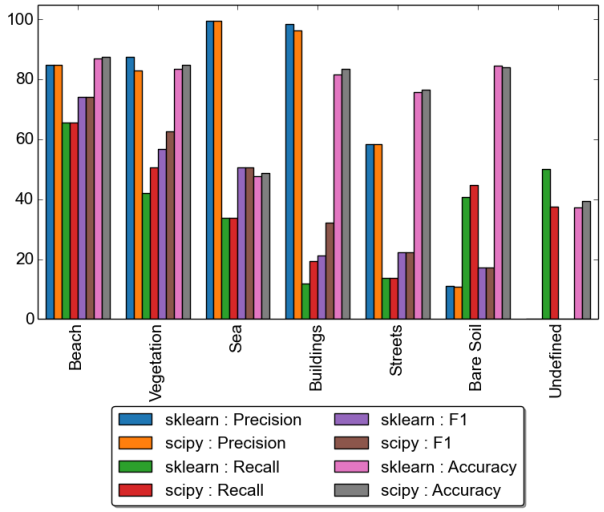

Fig. 5: Comparison of K-d tree (SciPy) and K-d tree batch queries (Scikit-learn) indexed classification obtained performance measures by class.

Indexing engine	i5-650 (3.2GHz) HH:mm:ss.00	i7-4750HQ (2.0GHZ) HH:mm:ss.00
None	07:43:36.63	06:00:35.91
K-d tree (Scipy)	00:25:43.75	00:14:21.65
K-d tree batch queries (Scikit-learn)	00:01:09.43	00:00:56.89

Table 4: Time performance measures for efficient classification. The results are obtained by an implementation in Python, an interpreted language. Improvement are possible by using compiled versions of the algorithm.

This represents the 5 and the 0.2 percent respectively of the processing time required by the index-less implementation.

The use of a Solid-State Drive might further improve these values.

6. CONCLUSIONS

We try to improve on existing systems for web based classification of remote sensing data by simplifying the architecture, extending the classification from unsupervised to supervised methods, and trying to attain real time performance by exploiting n-dimensional data indexing structures in the learning algorithm with the final aim of allowing users to interactively navigate and semantically map large extensions of geospatial Big Data from aerial and space-borne sensors.

Results are presented from a prototype implementation of this system. Qualitative and quantitative measures are reported for the performance of the obtained supervised method for thematic map definition.

7. REFERENCES

[1] Michael Schröder, Hubert Rehrauer, Klaus Seidel, and Mihai Datcu, "Interactive learning and probabilistic retrieval in remote sensing image archives," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 38, pp. 2288–2298, 2000.

[2] Quan Ho, Patrik Lundblad, Tobias Aström, and Mikael Jern, "A web-enabled visualization toolkit for geovisual analytics," *Information Visualization*, vol. 11, no. 1, pp. 22–42, 2012.

[3] Angel Ferran, Sergio Bernabe, Pablo G. Rodriguez, and Antonio Plaza, "A web-based system for classification of remote sensing data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 4, pp. 1934–1948, AUG 2013.

[4] Marco Quartulli and Igor García, "A review of EO image information mining," *ISPRS Journal of Photogrammetry and Remote Sensing*, pp. 11–28, January 2013.

[5] Ujjwal Maulik and Anasua Sarkar, "Efficient parallel algorithm for pixel classification in remote sensing imagery," *GeoInformatica*, vol. 16, no. 2, pp. 391–407, 2012.

[6] Soe W. Myint, Patricia Gober, Anthony Brazel, Susanne Grossman-Clarke, and Qihao Weng, "Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery," *Remote Sensing of Environment*, vol. 115, no. 5, pp. 1145 – 1161, 2011.

[7] Hugo Costa, Hugo Carre no, Fernando Baçao, and Mario Caetano, "Combining per-pixel and object-based classifications for mapping land cover over large areas," *Int. J. Remote Sens.*, vol. 35, no. 2, pp. 738–753, 2014.

[8] P. E. Keel, "Collaborative visual analytics: Inferring from the spatial organization and collaborative use of information," in *2006 IEEE Symposium On Visual Analytics Science And Technology*. IEEE, 2006, pp. 137–144.

[9] Javier Lozano, Marco Quartulli, Iñigo Tamayo, Maider Laka, and Igor G. Olaizola, "Visual analytics for built-up area understanding from metric resolution earth observation data," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, 2013.

[10] Javier Lozano, Naiara Aginako, Marco Quartulli, and Igor G.Olaizola, "Semi automatic remote sensing image layer generator based on web based visual analytics," *5th Jubilee International Conference on Cartography and GIS*, 2014.

[11] Eric Jones, Travis Oliphant, Pearu Peterson, et al., "SciPy: Open source scientific tools for Python," 2001–.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.