Usage of Incremental Learning in Land-Cover Classification

Jože Peternelj
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
joze.peternelj@ijs.si

Beno Šircelj
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
beno.sircelj@ijs.si

Klemen Kenda
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Jamova 39, 1000 Ljubljana, Slovenia
klemen.kenda@ijs.si

ABSTRACT

In this paper we present a comparison of a variety of incremental learning algorithms along with traditional (batch) learning algorithms in an earth observation scenario. The approach was evaluated with the earth observation data set for land-cover classification from Europe Space Agency's Sentinel-2 mission, the digital elevation model and the ground truth data of land use and land cover from Slovenia. We show that incremental algorithms can produce competitive results while using less time than batch methods.

Keywords

remote sensing, earth observation, incremental learning, machine learning, classification

1. INTRODUCTION

Land cover classification is one of the common and well researched tasks of machine learning (ML) in the Earth Observation (EO) community [1]. The challenge is to classify land into different types based on remote sensing data such as satellite images, radar data, information on weather [12] and altitude. The most commonly used data are satellite images, which may vary in acquisition period, resolution or wavelength. A plethora of algorithms have explored the potential of using a single-date image [3] and even time series of images for the task [11, 13]. Extensive work with state-of-the-art accuracy was performed using methods of deep learning [14]. The latter report a high computational effort in the learning and forecasting phase, which reduces their potential for continuous tasks requiring a timely response. There have also been efforts to reduce learning and prediction times using intelligent feature selection [6, 7]. To the best of our knowledge, no cases have been reported where stream models have been used in an EO scenario. The primary purpose of incremental learning would be to reduce the computational cost of classification, regression, or clustering techniques, which, when dealing with large data provided by Sentinel 2 and other sources, can be a significant cost to organizations trying to extract knowledge from that data. One of the advantages of incremental learning is that it is not necessary to load all the data into memory at once when creating a model. We only need to store the model and the part of the data we are processing. This could be especially useful in various EO scenarios, as the data from Copernicus services is estimated to exceed 150PB.

2. DATA

2.1 EO data

The Earth observation data were provided by the Sentinel 2 mission of the EU Copernicus programme, whose main objectives are land monitoring, detection of land use and land changes, support for land cover creation, disaster relief support and monitoring of climate change [2]. The data comprise 13 multi-spectral channels in the visible/near- infrared (VNIR) and short wave infrared (SWIR) spectral range with a temporal resolution of 5 days and spatial resolutions of 10m, 20m and 60m [8]. The Sentinel's Level-2A products (surface reflections in cartographic geometry) were accessed via the services of SentinelHub[1] and processed using `eo-learn`[2] library. Additionally, a digital elevation model for Slovenia (EU-DEM) with 30m resolution[3] was used.

2.2 LULC data

LULC (Land Use Land Cover) data for Slovenia is collected by the Ministry of Agriculture, Forestry and Food and is publicly available [10]. The data is provided in shapefile format, with each polygon representing a patch of land marked with one of the LULC classes. Originally there were 25 classes, but we introduced a more general dataset by grouping similar classes together. The frequencies of 8 newly grouped classes are shown in Figure 1.

2.3 Feature Engineering

The EO data were collected for the whole year. 4 raw band measurements (red, green, blue - RGB and near-infrared - NIR) and 6 relevant vegetation- related derived indices (normalized differential vegetation index - NDVI, normalized differential water index - NDWI, enhanced vegetation index - EVI, soil-adjusted vegetation index - SAVI, structure intensive pigment index - SIPI and atmospherically resistant vegetation index - ARVI) were considered. The derived indices are based on extensive domain knowledge and are used for assessing vegetation properties. One example is the NDVI index, which is an indicator of for vegetation health and biomass. Its value changes during the growth period of the plants and differs significantly from other unplanted

[1] `https://www.sentinel-hub.com/`
[2] `https://github.com/sentinel-hub/eo-learn`
[3] `https://www.eea.europa.eu/data-and-maps/data/eu-dem#tab-original-data`

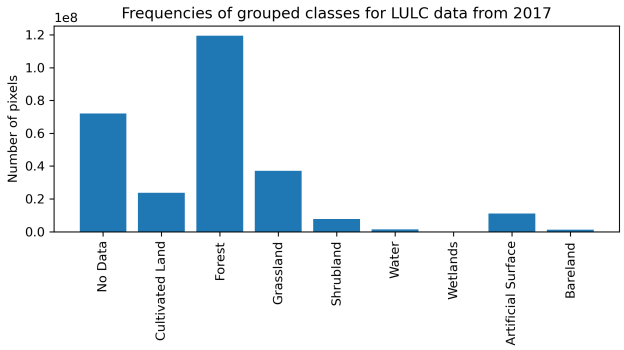

Figure 1: Frequencies of grouped classes for LULC data from 2017 show that the new simplified classification preserves the most common classes separated and merges the less common classes. Classes with the lowest frequencies were selected for oversampling.

areas. The NDVI is calculated as:

$$NDVI = \frac{NIR - red}{NIR + red}$$

Timeless features were extracted based on Valero et al. [11]. These features can describe the three most important crop stages: the beginning of greenness, the ripening period and the beginning of senescence [11, 13]. Annual time series have different shapes due to the phenological cycle of a crop and characterize the development of a crop. With timeless features, they can be represented in a condensed form.

For each pixel, 18 features per each of 10 time series were generated. From elevation data, the raw value and maximum tilt for a given pixel were calculated as 2 additional features. In total 182 features were constructed. From these features only a Pareto-optimal subset of 9 features was selected [6].

3. METHODOLOGY

Classification accuracy **(CA)** and **F1** score were calculated for **11** different ML methods, **6 batch** learning methods and **5 incremental** learning methods. All incremental learning methods are available in the `ml-rapids` (MLR)[4] library which has been developed in order to support the use of incremental learning techniques within eo-learn [4] library.

Hoeffding Tree (incremental)

Hoeffding tree (HT) is an incremental decision tree that can learn from massive streams. It assumes that the distribution of generating examples does not change over time. The Hoeffding tree begins as an initially empty leaf. Each time the new example arrives, the algorithm sorts it down the tree (it updates the internal nodes statistics) until it reaches the leaf. When it reaches the leaf, it updates the leaf statistics of all unused attributes. It then takes the best (A) and second-best (B) attributes based on standard deviation and calculates the ratio of their reductions. To find the best attribute to split a node the Hoeffding bound is used. First algorithm

[4] `https://github.com/JozefStefanInstitute/ml-rapids`

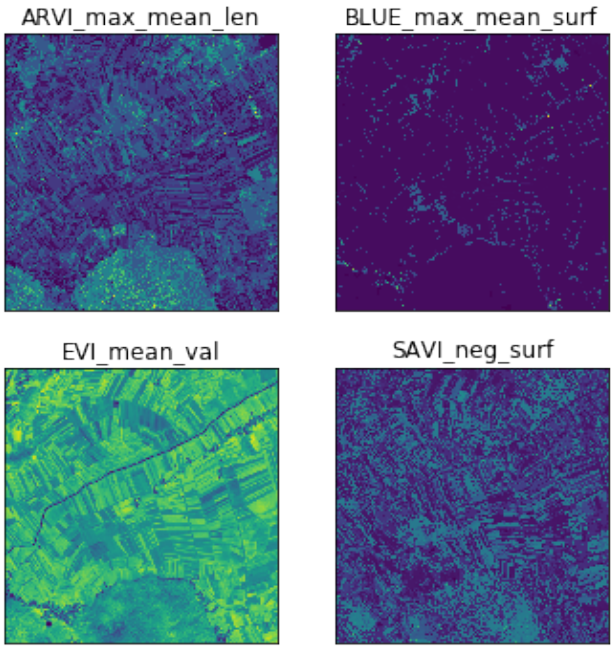

Figure 2: Example of some of the timeless features. `ARVI_max_mean_len` shows the length of maximum mean value in a sliding temporal neighbourhood of ARVI index. `BLUE_max_mean_surf` shows the surface of the flat interval area containing the peak using the blue raw band. `EVI_mean_val` shows mean value of EVI index and `SAVI_neg_sur` shows the maximum surface of the first negative derivative interval of SAVI index.

checks if the ratio is less than $1 - \epsilon$, where $\epsilon = \sqrt{\log \frac{1/\delta}{2n}}$ and $1 - \delta$ is desired confidence. If the ratio is small enough, meaning that attribute A is really better than attribute B, then the algorithm divides the node by that attribute.

Bagging of HT (incremental)

Given a standard training set D of size n, bagging generates m new training sets D_i, each of size n', by uniform sampling from D. Because the sampling is done with replacement, some observations can be repeated in each D_i. If $n' = n$, then for large n the set D_i is expected to have the fraction $(1 - 1/e)(\approx 63.2\%)$ of the unique examples of D, the rest being duplicates. Then, m HT models are fitted using the above m samples and combined by voting. To include a new sample, a random subset of models are selected according to Poisson distribution [9], and these models are updated with the sample in the same way as the HT model described above.

Naïve Bayes (incremental)

Naïve Bayes (NB) is a classification technique based on Bayes's Theorem. It lets us calculate the probability of data belonging to a given class, given prior knowledge. Bayes' Theorem is:

$$P(class|data) = \frac{P(data|class)\ times P(class)}{P(data)}$$

where $P(class|data)$ is the probability of class given the provided data. To add a new training instance, NB only needs to update relevant entries in its probability table.

Logistic Regression (incremental)
Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. A model with two predictors x_1 and x_2 and a binary variable Y, denoted by $p = P(Y = 1)$, which gives us the odds of the values belonging to the class p. The relationship between these terms can be modeled with the following equation:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}}$$

The parameters $\beta_0, \beta_1, \beta_2$ can be determined by stochastic gradient descend using logistic loss function.

Perceptron (incremental)
Perceptron is very similar to Logistic regression. It models a binary variable with the same activation function. The only difference is in the cost function that is used for gradient descend.

Batch learning methods
Batch learning methods learn from the whole training set and do not have to rely on heuristics (e.g. Hoeffding bound) or incremental approaches (like SGD) for building the model. The following batch methods have been tested: decision trees, gradient boosting (LGBM), random forest, perceptron, multi-layer perceptron, and logistic regression [5].

4. RESULTS

Results of the experiments are summarised in Figures 3, 4 and Table 1. Figures depict dependency of algorithm-specific F_1 score vs. its training and inference times. An ideal algorithm would be located in the top left corner, achieving full F_1 score with a training and inference time of 0. Any algorithm that has no other algorithm in its top-left quadrant (no algorithm is both more accurate and faster) belongs to a Pareto front, which means that this algorithm is optimal for a certain set of use-cases.

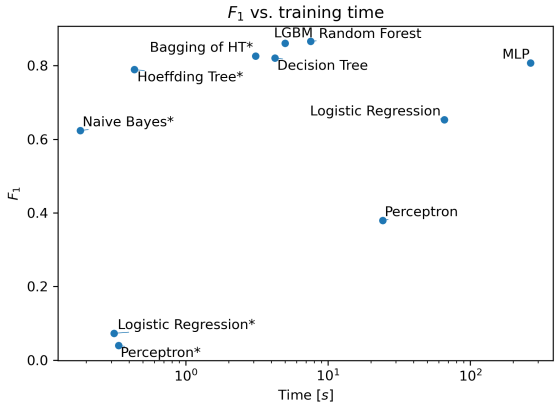

Figure 3: F1 score vs. training time of different models for predicting LULC classes. *Denotes incremental algorithms.

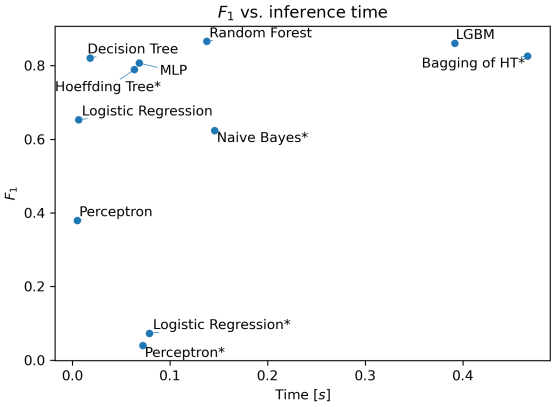

Figure 4: F1 score vs. inference time of different models for predicting LULC classes. *Denotes incremental algorithms.

We can observe that ml-rapid's Naïve Bayes, Hoeffding Tree, Bagging of HT, Decision Trees, LGBM and Random Forest belong to the Pareto optimal set of algorithms according to the training time and F1 score. Regarding inference times Logistic Regression, Decision Trees and Random Forest are the only Pareto optimal algorithms. The choice of algorithm depends on the available processing power and time. For a system that has a lot of time and resources available, it would be best to use Random Forest as it has the highest F1 score. In practice, this is not always feasible. For example, if the algorithm were used for an on-board system on the satellite, we could not afford to save all the data and would prefer to load only the model. With an incremental algorithm, the data could be collected, processed and discarded while the acquired knowledge would be stored in the model. Another preference for HT would be in a wrapper feature selection algorithm [6]. This type of algorithms do a lot of evaluations of the selected method. The main result is a subset of features that can later be used with other algorithms. The acquired set of features might be biased towards the method used, but the results would be obtained much faster.

From the confusion matrix of the HT algorithm shown in Figure 5, we can see that shrubland is often wrongly classified as forest, bareland or grassland and vice versa. This is mainly due to the unclear distinction between these classes (e.g. shrubland can be anything between bareland and forest) and poor ground truth data due to infrequent updates, low accuracy, and lack of detail (e.g. patch of land labeled as shrubland can also grassland and trees). The unclear distinction between certain classes may also explain confusion between wetlands and shrubland or wetlands and grassland, as wetlands may be covered with grass or shrubs. The lack of detail also contributes to misclassification between grassland and artificial surface, as not every small grassy area, such as park or lawn, is included in ground truth data. Finally, grass cultures, unused land overgrown by grass and rotation of crops are likely some of the reasons for confusion between cultivated land and grassland.

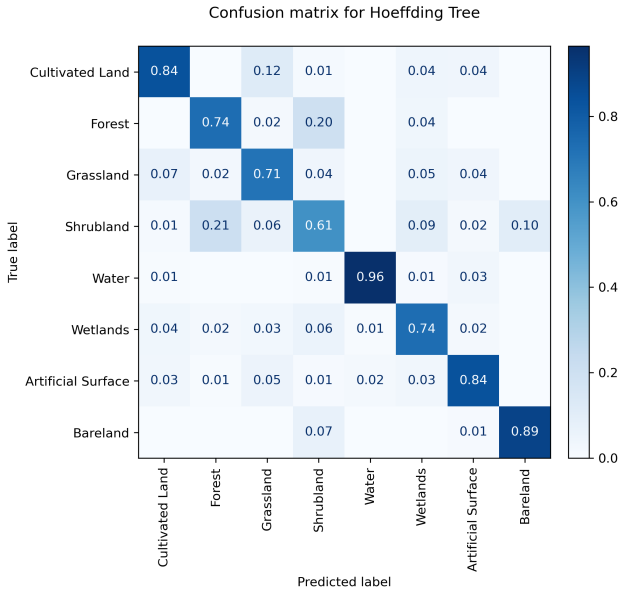

Figure 5: Confusion matrix of HT based model for predicting LULC classes.

	Training time	Inference time	CA	F1
LGBM	4.87	0.38	0.86	0.86
Decision Tree	4.18	0.02	0.82	0.82
Random Forest	7.53	0.14	0.87	0.87
MLP	264.67	0.07	0.81	0.81
Logistic Regression	63.50	0.01	0.67	0.65
Perceptron	24.05	0.01	0.45	0.38
Hoeffding Tree*	0.44	0.06	0.79	0.79
Bagging of HT*	3.07	0.46	0.83	0.83
Naïve Bayes*	0.18	0.15	0.64	0.62
Logistic Regression*	0.31	0.08	0.15	0.07
Perceptron*	0.33	0.07	0.14	0.04

Table 1: Comparison of models for predicting LULC classes. *Denotes incremental algorithms.

5. CONCLUSIONS

In our approach we have concentrated on effective processing. Our goal was to provide methods and workflows which can reduce the need for extensive hardware and processing power. Our goal was focused on use cases where a near state-of-the-art accuracy can be achieved with only a fraction of the processing power required by the state-of-the-art. We have researched stream mining algorithms. We have shown that these algorithms, even if they are not the most accurate or the fastest, take their place at the Pareto front in a multi-target environment, which means that some users might find them suitable for their needs and that they provide the best results for particular computational demand.

6. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the ICT program of the EC under project PerceptiveSentinel (H2020-EO-776115) and project EnviroLENS (H2020-DT-SPACE-821918).

7. REFERENCES

[1] D4.7 stream-learning validation report, May 2020. Perceptive Sentinel.

[2] Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., et al. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment 120* (2012), 25–36.

[3] Gómez, C., White, J. C., and Wulder, M. A. Optical remotely sensed time series data for land cover classification: A review. *ISPRS Journal of Photogrammetry and Remote Sensing 116* (2016).

[4] H2020 PereptiveSentinel Project. Eo-learn library. `https://github.com/sentinel-hub/eo-learn`. Accessed: 2019-09-06.

[5] Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[6] Koprivec, F., Kenda, K., and Šircelj, B. Fastener feature selection for inference from earth observation data. *Entropy* (Sep 2020).

[7] Koprivec, F., Peternelj, J., and Kenda, K. Feature Selection in Land-Cover Classification using EO-learn. In *Proc. 22th International Multiconference* (Ljubljana, Slovenia, 2019), vol. C, Institut "Jožef Stefan", Ljubljana, pp. 37–40.

[8] Koprivec, F., Čerin, M., and Kenda, K. Crop Classification using Perceptive Sentinel. In *Proc. 21th International Multiconference* (Ljubljana, Slovenia, 2018), vol. C, Institut "Jožef Stefan", Ljubljana, pp. 37–40.

[9] Oza, N. C. Online bagging and boosting. In *2005 IEEE international conference on systems, man and cybernetics* (2005), vol. 3, Ieee, pp. 2340–2345.

[10] Slovenian ministry of agriculture. Mkgp - portal. `http://rkg.gov.si/`. Accessed: 2020-08-11.

[11] Valero, S., Morin, D., Inglada, J., Sepulcre, G., Arias, M., Hagolle, O., Dedieu, G., Bontemps, S., Defourny, P., and Koetz, B. Production of a dynamic cropland mask by processing remote sensing image series at high temporal and spatial resolutions. *Remote Sensing 8(1)* (2016), 55.

[12] Čerin, M., Koprivec, F., and Kenda, K. Early land cover classification with Sentinel 2 satellite images and temperature data. In *Proc. 22th International Multiconference* (Ljubljana, Slovenia, 2019), vol. C, Institut "Jožef Stefan", Ljubljana, pp. 45–48.

[13] Waldner, F., Canto, G. S., and Defourny, P. Automated annual cropland mapping using knowledge-based temporal features. *ISPRS Journal of Photogrammetry and Remote Sensing 110* (2015).

[14] Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., and Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine 5*, 4 (2017), 8–36.