Feature Selection in Land-Cover Classification using EO-learn

Filip Koprivec
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
filip.koprivec@ijs.si

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

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

ABSTRACT

Applying machine learning to Big Data can be a cumbersome task which requires a lot of computational power and memory. In this paper we present a feature selection technique for land-cover classification in earth observation scenario. The technique extends the state-of-the-art feature extractors by pruning the dimensionality of the required feature space and can achieve almost optimal results with 10-fold reduction of the number of features. The approach utilizes a genetic algorithm for generation of optimal feature vector candidates and multi-objective optimization techniques for candidate selection.

Keywords

remote sensing, earth observation, machine learning, feature selection, classification

1. INTRODUCTION

Earth observation (EO) has become one of the major sources of Big Data. European Sentinel-2 mission, which acquires global data with 5-day revisit time, reports a total of 6.4 PB of satellite imagery products being available to the users via Copernicus services [2], whereas the total cumulative amount of EO data available from European Space Agency (ESA) is estimated to exceed 140 PB.

A huge amount of data have motivated EO and machine learning communities to invest into methodologies to work with such high volumes. Since 2016, as observed in Big Data from Space conferences, the community has tackled and solved the problem of storing, pre-processing and applications of basic machine learning and extensive deep learning algorithms for mainly solving classification problems. Processing pipelines have been established and are used regularly for solving different EO tasks [6].

The research has already approached the limits of the accuracy of the models. Our research has therefore focused on trade-off between model accuracy (of the current state-of-the-art) and processing efficiency. The approach is expected to be used in systems, which require a fast response with reasonably good results. Possible approaches include the use of fast classification techniques (i.e. Very Fast Decision Trees), which were taken from the field of stream mining, and optimization of the feature selection process.

This paper presents an early attempt to provide effective feature selection in land-cover classification. We illustrate that it is possible to significantly reduce the dimensionality of feature space of the state-of-the-art feature extractors [1, 8, 9] applied to a time-series of satellite images. Experimental data has been acquired by EO-learn library from PerceptiveSentinel[1] project.

2. DATA

Acquiring EO data is achieved using services provided by European Space Agency (ESA). For our experiments we have used Sentinel-2 missions data. This data includes scalar features from 13 different sensors with a resolution from 10 m × 10 m to 20 m × 20 m. A more detailed description of data available within Sentinel-2 missions is provided in [6]. EO-learn library [3] presents an abstraction layer over ESA services, which provide access and basic pre-processed (i.e. atmospheric correction, cloud detection and similar) products.

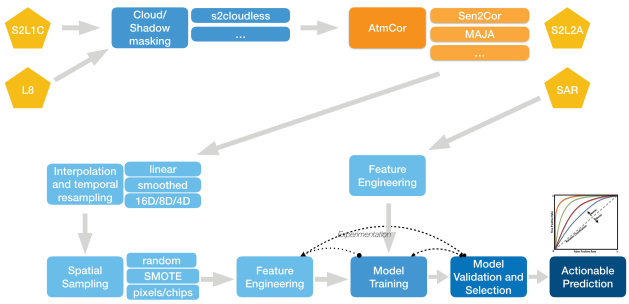

Figure 1: Data flow (acquisition and pre-processing) with EO-learn library using Sentinel-2 data. EO-learn modules are depicted with light blue containers.

Figure 1 depicts the data flow in a typical experiment. The top row depicts components for Level-1 and Level-2 pre-processing, which include cloud detection and atmospheric corrections. Products are being stored in the cloud and are accessed via EO-learn library. EO-learn modules are independent and can communicate with one another through a unified data structure (`EOPatch`) that can include satellite

[1]http://www.perceptivesentinel.eu/

imagery data, enriched features, metadata and even corresponding vector data. For example: a feature engineering module for calculating normalized differential vegetation index (NDVI) from raw data would take `EO-patch` including the original 13 bands as an input and would output the same patch with an added NDVI index. Such modules are reusable and are being accumulated in the EO-learn library and made available to the community. Complex data processing and analytics pipelines can therefore be established literally within minutes.

3. METHODOLOGY

Based on satellite imagery our task is to classify land-cover in Slovenia. For this task we are using a time-series of images from the same year, which capture the dynamics of growth of particular vegetation and enable better accuracy of the models than a single image. Labels for building classification models have been acquired from a patch of land-use data (Slovenian LPIS data). The models can be applied to a wider area, where ground-truth data is not available and can even uncover some ground truth data mistakes (or generalizations). Our goal is to solve the task as fast as possible yet still accurate.

We base our methodology on the extraction of the state-of-the-art features from Sentinel-2 dataset. On top of this dataset we perform intelligent feature selection procedure based on multi-objective optimization approach.

3.1 Feature Engineering

We have acquired a time-series of satellite imagery for year 2017 and selected 27 small tiles ($1\,\text{km} \times 1\,\text{km}$) from Slovenia randomly (ensuring, that appropriate distribution of different land-covers was consistent). We have performed cloud detection and then provided linear interpolation (simply because it is the fastest) over the remaining data points for each of the bands and additional indices. From these interpolated data we have extracted the phenological features suggested by Valero et al. [8]. The features have been calculated from following indices: NDVI, NDWI, EVI, SAVI, ARVI and SPI[2] [5, 6]. These indices provide various information from the time-series which are important for land-cover classification (i.e. speed of growth, length of maximum index interval, etc.). All together we have used 108 different features within our experiments. Some examples of the features are depicted in Figure 2.

3.2 Feature Selection

A feature selection algorithm should choose a limited amount of features out of the pool of 108, which would still provide enough information for almost optimal classification of land-cover. We employed a modification of the POSS genetic optimization algorithm [7] for the task. The algorithm would select a candidate solution (a selection of features) and slightly modify (mutate) it. The mutations have to be considered carefully, since the number of selected features must be kept as small as possible. The problem can be formulated as $f : 2^N \to \mathbb{R}$, where N is the number of all features. We are looking for a subset $A \subseteq 2^N$ that optimizes (minimizes or maximizes) the selected criterion function.

[2]normalized differential vegetation index, normalized differential water index, extended vegetation index, soil-adjusted vegetation index, atmospherically resistant vegetation index and standardized precipitation index

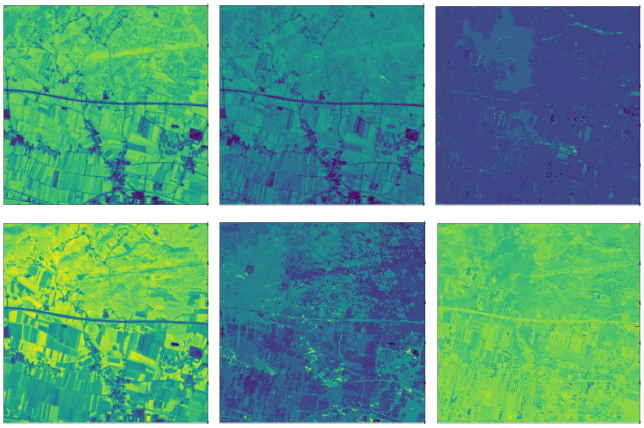

Figure 2: A sample of features extracted from a time-series of images: standard deviation of NDVI, max difference in NDVI in a sliding window, length of time interval where max mean value is attained (with specified tolerance), mean NDVI and rate of NDVI time-series change corresponding to the longest positive interval.

A naïve genetic algorithm without proper weighting of a number of features behaves poorly on most tested classifiers. If optimizing only the accuracy score (i.e. F_1), the algorithm would almost always converge towards selecting all the features (since the dataset is large and there is generally no danger of overfitting). We modified the POSS algorithm to search possible feature space and optimize the number of selected features as well as the accuracy score with a 2-dimensional multi-objective optimization.

The main idea of the algorithm is as follows. We have N features, which we encode into a solution candidate $S = \{f_1, f_2, \ldots, f_N\}$. A bit f_i represents whether the i-th feature is selected (value 1) in the candidate solution or not (value 0). We keep the current optimal elements on a 2-dimensional Pareto front, which is determined by $1 - F_1$ score and number of selected features (for illustration see Figure 4). This approach can easily be extended to any other fixed dimension. $1 - F_1$ is selected for convenience in selection (elements on Pareto front are those that are not comparable to any others in the current Pareto front, as determined by strict product order for each dimension, strict or non-strict is just a matter of preference when considering equality, but non-strict version more naturally excludes duplicates). In each iteration, the algorithm uniformly samples an item from the Pareto front and tries to improve it. Each bit f_i is then flopped with probability $\frac{1}{N}$, where N is the number of features.

This newly constructed candidate is then evaluated for its performance (F_1). All the items on the Pareto front are then compared with this new item. If there exists no such item that is comparable or bigger from the new item, the new item is on the Pareto front and is subsequently added to it. All items that are comparable or smaller than new item

are removed from the Pareto front, as they are (strictly) Pareto sub-optimal. Strictness is useful since it removes the duplicates (in a non-strict product weak ordering, even if the relation is non-linear, as in the case in the Pareto front, the product ordering is antisymmetric) [5].

4. RESULTS

Results of the early feature selection experiments are depicted in Figures 3 and 4. We have tested the methodology with the most popular classification techniques used in remote sensing (apart from deep learning): gradient boosting (LightGBM implementation [4]), random forests and logistic regression (baseline). Gradient boosting has proven to be a superior method whereas logistic regression performed the worst. SVM classifier was not considered since its training time complexity $\mathbb{O}(N^3)$ is too high for frequent re-training, needed in the feature selection algorithm.

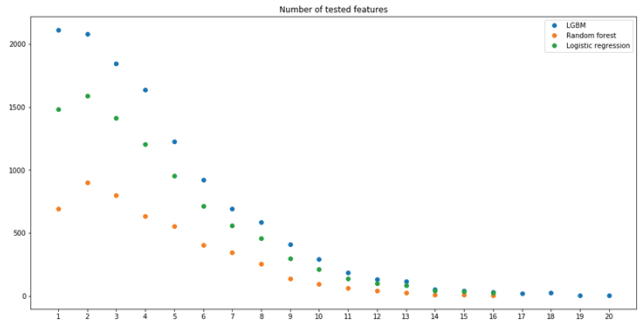

Figure 3: Number of tested candidates (y) per number of features (x). Gradient boosting is depicted with blue, random forests with orange and logistic regression with green dots.

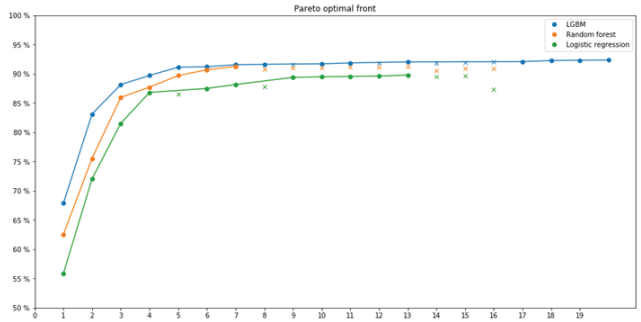

Figure 4: The best candidate F_1 score (y) per number of features (x). The lines depict the Pareto front for a particular classification algorithm. The optimal number of features for random forests and logistic regression is smaller than the size of the longest tested feature vector. Gradient boosting is depicted with blue, random forests with orange and logistic regression with green colour.

Figure 3 depicts the number of tested candidates per number of features. Number of features starts to decline sharply, but is a bit jumpy. This represents an expected behaviour considering the random nature of the feature selection algorithm and incremental difficulty of greatly increasing the number of features.

Figure 4 shows that smaller number of tested examples with a high number of features does not significantly affect F_1 score (considering small changes of a random element on the Pareto front, this seems reasonable). The same figure also shows, that already with a careful selection of just a few "good" features, classification produces quite good results. The figure also nicely depicts part of Pareto front and shows that high quality of feature selection might also improve the classification in some cases.

A clear plateau shape can be seen in Figure 4, hinting, that there is a reasonable choice of a subset of features. Selecting a small, but optimal subset of all features can yield good accuracy score of the classification algorithm, with decreased memory and computation footprint. The most important consequence of using an optimal subset of features is, that it saves a lot of time for data preparation (not extracting unneeded features, not sending/saving unneeded data) and most importantly makes the model reasonably small and fast, which allows usage even on a plethora of low computational power devices.

In the results presented above, LightGBM classification algorithm performance is unmatched by either random forest or logistic regression. This is an expected result since boosting can skew the feature space and can inherently introduce non-linear features into the model. The most illustrative case for the strength of proper feature selection is however seen in the case of random forest algorithm. We can observe from Figure 4 that already with 7 wisely chosen features (out of 108) one can achieve the optimal F_1 classification score. The reduced number of features speeds up the feature extraction step (less features need to be calculated) and modeling (less data is needed, fewer features are considered) and reduces the memory consumption demand.

5. CONCLUSIONS AND FUTURE WORK

This is the early paper on feature selection used for land-cover classification. It shows great potential of the methodology and up to 15-fold reduction of the number of needed phenological features in order to still achieve state-of-the-art accuracy. The methodology could be used with potentially great benefits also on other types of feature vectors in land-cover classification (i.e. with resampled index values), where it would automatically find the features that can distinguish between various land-cover classes. The main underlying reason for our research lies in the provision of computationally effective methods for faster, easier and cheaper EO data analysis.

There are still research challenges to be considered in this work. Firstly, benefits of feature reduction to the computational tasks should be examined in depth. The most important phase of the process is the inference phase (land-cover classification on large areas). However, preliminary results indicate that speed-up and memory consumption might be smaller than expected based on common sense.

Feature selection should be tested with other faster classification methods (i.e. based on incremental learning [5]), which trade accuracy for the faster computation. The latter might be beneficial in particular use cases (i.e. on-the-fly classification for on-line EO browsers like SentinelHub or

large scale classification). A comprehensive study of benefits within full-stack pipelines (from data acquisition to inference) should be conducted.

Earth observation community has striven towards achieving optimal accuracy of the classification algorithms in the past few years. Especially deep learning algorithms have shown to require vast amounts of computational time, which is sometimes difficult to obtain. Presented work, together with research into computationally effective classification methods, might be a step towards sacrificing some of the accuracy in order to achieve final results sooner and with less struggle.

6. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the ICT program of the EC under project PerceptiveSentinel (H2020-EO-776115). The authors would like to thank Sinergise for their contribution to EO-learn library along with all help with data analysis.

7. REFERENCES

[1] Belgiu, M., and Csillik, O. Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis. *Remote Sensing of Environment 204* (2018), 509 – 523.

[2] European Space Agency. Mission status report 152. `https://sentinel.esa.int/documents/247904/3720568/Sentinel-2-Mission-Status-Report-152-25-May-28-June-2019.pdf`. Accessed: 2019-08-01.

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

[4] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* (2017), pp. 3146–3154.

[5] Koprivec, F., Kenda, K., Čerin, M., Bogataj, M., and Peternelj, J. H2020 Perceptive Sentinel - Deliverable 4.6 Stream Mining Models for Earth Observation. Reported 31st March 2019.

[6] 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.

[7] Qian, C., Yu, Y., and Zhou, Z.-H. Subset selection by pareto optimization. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 1774–1782.

[8] 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.

[9] 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), 1 – 13.