

Temporal and spatial approaches for land cover classification.

Ryabukhin Sergey

`sergeyryabukhin@gmail.com`

Abstract. This paper describes solution for Time Series Land Cover Classification Challenge (TiSeLaC). Using features extracted from satellite images time series (SITS) each pixel corresponding to $30m \times 30m$ area can be classified to one of general class (urban area, forest, water, etc.). The following approaches implemented and evaluated: classical data mining multiclass prediction, local context embedding and extracting shapes of temporal dynamics. Also the different cross-validation schemes considered to evaluate performance of approaches.

Keywords: image classification, satellite images time series, land cover classification

1 Introduction

The Time Series Land Cover Classification Challenge (TiSeLaC)¹ provided with time series of processed 23 Landsat 8 images acquired in 2014 above the Reunion Island (2866×2633 pixels at 30 m spatial resolution), provided at level 2A². Among the many land cover classes the following 9 most important classes are retained for task:

- Urban Areas
- Other built-up surfaces
- Forests
- Sparse Vegetation
- Rocks and bare soil
- Grassland
- Sugarcane crops
- Other crops
- Water

Further, the 81714 observations selected with stratified sampling. Every observation covers $30m \times 30m$, assigned to unique class and has following 10 features:

¹ <https://sites.google.com/site/dinoienco/tiselc>

² Level 2A is a monodate ortho-rectified image expressed in surface reflectance, provided with a cloud/cloud shadow/snow/water mask, <http://www.cesbio.ups-tlse.fr/multitemp/?p=3202>

- UltraBlue
- Blue
- Green
- Red
- NIR (Near-infrared)
- SWIR1 (Shortwave Infrared 1)
- SWIR2 (Shortwave Infrared 2)
- NDVI (Normalized Difference Vegetation Index)
- NDWI (Normalized Difference Water Index)
- BI (Brightness Index)

Also pixel coordinates (roughly related to longitude and latitude) for each point provided.

Let use $n_features = 10$ to denote the number of original features provided and $n_periods = 23$ to denote number of consecutive images in time series.

In difference with other competitions³ for classifying object on aerial or satellite images, there the full images were given, here only part of pixels are proposed. Also in other tasks the classification is applied on segments or whole images, while pixel-wise classification is proposed for this challenge.

In next sections temporal approach (section 2) and spatial context approach (section 3) are described. Also all approaches validated on different validation schemes which corresponds to different methods for data preparation (details in section 4). Finally in section 5 the scores for every approach listed and in section 6 an advantages and restrictions of above approaches analyzed.

2 Temporal approaches

Here the two methods applied: one is to encode temporal shape of every feature's by single category, then combine them to $n_features$ features, another is to encode each feature set snapshot to single category, then combine them to $n_periods$ features. Both encodings were implemented with clustering. Clustering algorithm MiniBatchKMeans is used with implementation from [1] based on paper [2]. While the clustering algorithm usually have many parameters to be tuned, here only different number of clusters tried.

2.1 Clustering temporal shapes

Every sequence consisting of $n_periods$ single channel measurement is treated as features and encoded with single cluster number. So what there is $n_features$ of new categorical features given. As it can be seen in Fig. 1 for $n_clusters = 10$ some of temporal shapes are well separated.

³ http://dataring.ru/competitions/fpi_sk_competition/
<https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection>
<https://www.kaggle.com/c/planet-understanding-the-amazon-from-space>

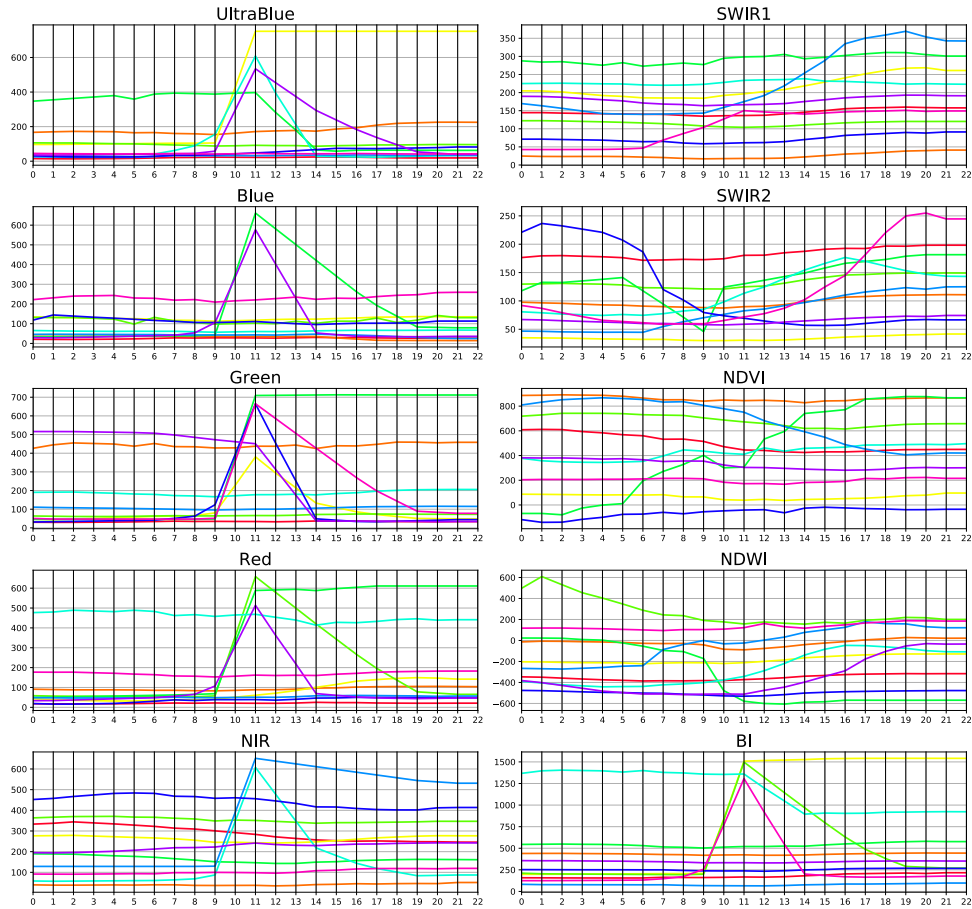


Fig. 1. Each feature clustered by 10 clusters. One line correspond to cluster center in different periods.

2.2 Clustering snapshots

From the other side not temporal shape but single period set of features were clustered. Here the task is to find clusters in $n_features$ -dimensional space. After combining all clusters for each period the new $n_periods$ categorical features are given. A few examples of such clusters are shown on Fig. 2

In each method clusters are combined and passed as categorical features to classifier (see Section 5).

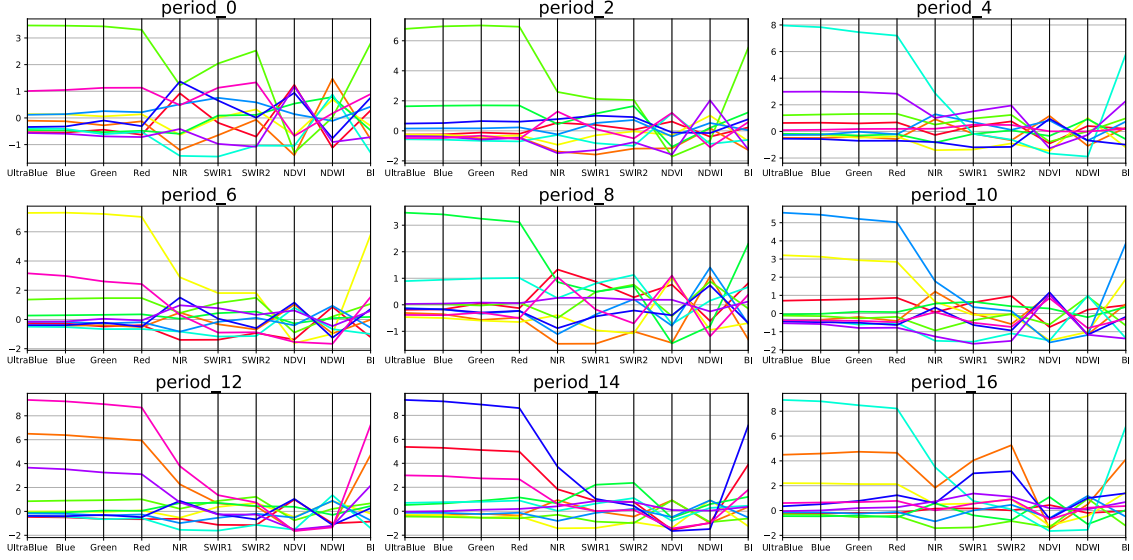


Fig. 2. Each snapshot clustered by 10 clusters. One line correspond to cluster center for each normalized feature.

3 Spatial context models

3.1 Neighbors embedding

Using spatial context is crucial approach in image classification tasks especially in aerial and satellite images. Despite the data is sparse as we here usually have no immediate neighbors pixels available, the following steps for searching the nearest neighbors and calculation of distribution properties among pixel area is applied:

1. For every point
 - search for the $n_neighbors$ nearest neighbors ordered by $L2$ distance
 - Fetch neighbors features: $F_{c,t,k}$, where
 - $c \in 1, \dots, n_features$ - number of features,
 - $t \in 1, \dots, n_periods$ - time,
 - $k \in 1, \dots, n_neighbors$ - number of neighbor
2. The point itself features also added
3. Train classifier on new dataset with $n_features * n_periods * (n_neighbors + 1)$ features.

See section 5 for results.

3.2 k-NN model

Here the non-parametric classification is applied to predict class based on nearest neighbors using coordinates provided.

4 Cross-validation schemes

The good cross-validation scheme is crucial part to construct robust and reusable classification model. Here a few schemes were designed to score every approach. Firstly the original proposed scheme with evenly distributed train and validation points are used. Further, to check how size of labeled samples affected accuracy of prediction the sub-sampling schemes are used (see Fig. 3):

- sub-sample train with ratio 0.7
- sub-sample train with ratio 0.4
- sub-sample train with ratio 0.1

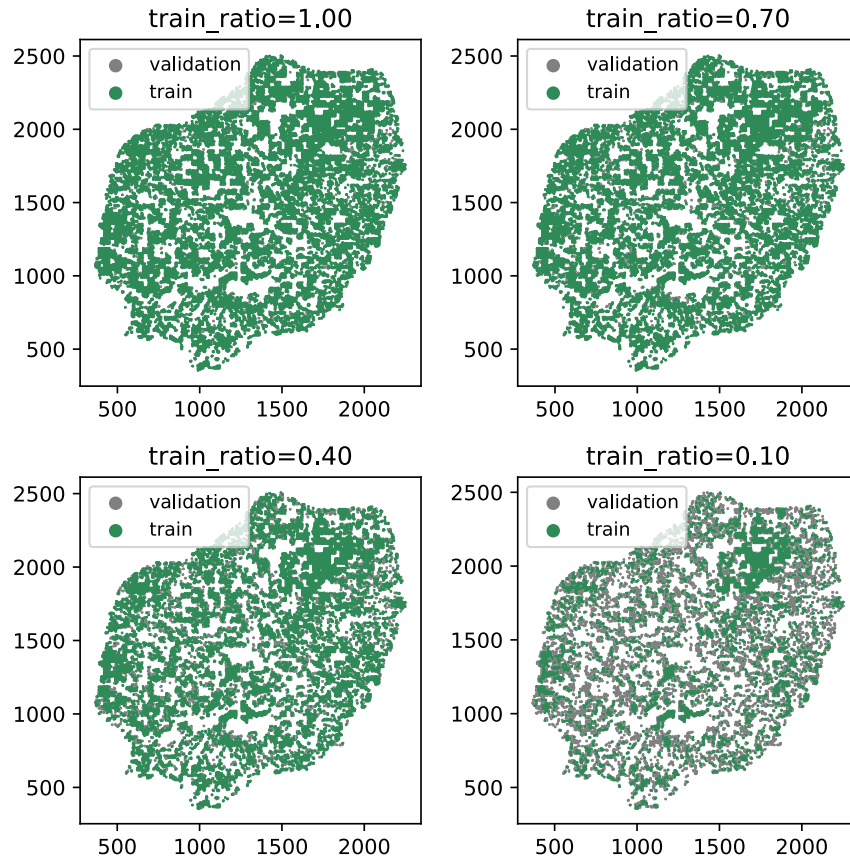


Fig. 3. Train and validation distribution on map for different train set size ratios.

Another scheme (see Fig. 4) is pursued the spatial separation of train and validation points and implemented by follows:

- split coordinate space to 100 (10x10) equal rectangles
- randomly split rectangles between train and validation

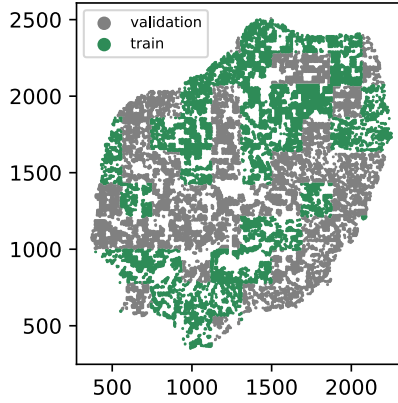


Fig. 4. Train and validation distribution on rectangle-separated cross-validation scheme.

In every schemes the size of validation set is the same and is equal to 1/3 of total sample.

5 Experimental results

For scoring prediction F1score with option 'weighted' is used:

$$F1score_weighted = \frac{\sum_{c=1}^C F1score(p_c, c) * |c|}{N}$$

where C count of classes, $c = 1, \dots, C$ is classes, p_c is predictions for objects from class c , $N = \sum_{c=1}^C |c|$ is size of test set.

The following cross-validation results are shown in **Table 1**:

- **Benchmark** For the simplest benchmark the Extremely Randomized Trees Classifier (ETC) ([3], with implementation from [1]) is trained without any preprocessing of original features.
- **Temporal feature clusters**, for ETC the best result for n_clusters=60 shown, also plot scores for different number of clusters is shown on Fig. 5
- **Snapshot clusters**, for ETC the best result for n_clusters=40 shown, also plot scores for different number of clusters is shown on Fig. 6

- **Neighbors embedding** for ETC the result for $n_neighbors$ for 1,4,9 is shown.
- **Coordinate neighbors** using Nearest Neighbors Classifier from [1], the results for $n_neighbors$ for 1,2,10 is shown.

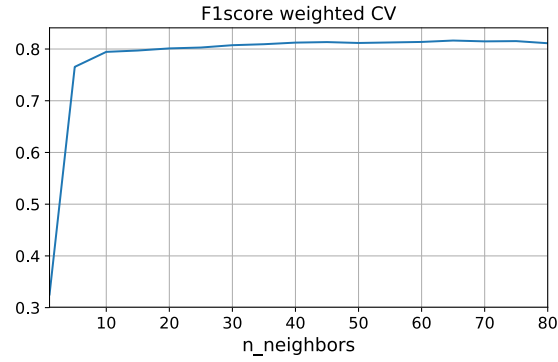


Fig. 5. Cross-validation scores for different number of clusters for temporal feature shapes.

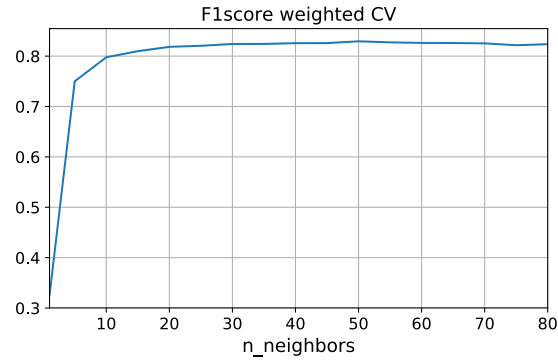


Fig. 6. Cross-validation scores for different number of clusters for snapshots.

6 Discussion

Despite the clustering discovered many patterns of seasonal reflectance dynamics, it shows worst results even in comparison with benchmark. Although different clustering models and distance metrics may help. Also the best scores for this

methods achieved at high number of clusters (~ 60), hence it looks like during clustering some important information is loosed.

The spatial approaches both outperform benchmark, and the 1-Nearest Neighbors Classifier show the best result among all other approaches. This approach very robust to train size decreasing and trained only on 10% of points it outperforms other approaches that employs full data set.

Also from spatial approaches the Nearest Embedding with 9 neighbors shows best result on cross-validation with rectangles, where k-Nearest Neighbors Classifier is not applicable. So this method can be recommended to use on completely new area for land cover classification.

Table 1. F1score weighted on different approaches (rows) and cross-validation schemes (columns)

Method	Classifier	Original Rectangles	Train 70%	Train 40%	Train 10%	
Benchmark	ETC30	0.8893	0.7797	0.8824	0.8719	0.8409
Cluster features time series	ETC30	0.8185	0.7063	0.7846	0.7746	0.7546
Cluster snapshots	ETC30	0.8294	0.7314	0.8223	0.8116	0.7816
Neighbors embedding 1	ETC30	0.9038	0.7975	0.8964	0.8834	0.8496
Neighbors embedding 4	ETC30	0.9056	0.8141	0.8968	0.8883	0.8569
Neighbors embedding 9	ETC30	0.9041	0.8171	0.8966	0.8862	0.8528
Coordinates only	1-NN	0.9850	NA	0.9787	0.9663	0.9071
Coordinates only	2-NN	0.9789	NA	0.9716	0.9553	0.8875
Coordinates only	10-NN	0.9523	NA	0.9393	0.9158	0.8335

References

1. 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*, 12:2825–2830, 2011.
2. D. Sculley. Web-scale k-means clustering. *WWW 2010: Proceedings of the 19th Annual International World Wide Web Conference.*, 2010.
3. Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, Apr 2006.