Deep constrained clustering applied to satellite image time series

Baptiste Lafabregue[1,2], Jonathan Weber[1],
Pierre Gançarski[2], and Germain Forestier[1]

[1] IRIMAS, University of Haute-Alsace, Mulhouse, France
[2] ICube, University of Strasbourg, Strasbourg, France
{baptiste.lafabregue,jonathan.weber,germain.forestier}@uha.fr
gancarski@unistra.fr

Abstract. The advent of satellite imagery is generating an unprecedented amount of remote sensing images. Current satellites now achieve frequent revisits and high mission availability and provide series of images of the Earth captured at different dates that can be seen as time series. Analyzing satellite image time series allows to perform continuous wide range Earth observation with applications in agricultural mapping, environmental disaster monitoring, etc. However, the lack of large quantity of labeled data generally prevents from easily applying supervised methods. On the contrary, unsupervised methods do not require expert knowledge but sometimes provide poor results. In this context, constrained clustering, which is a class of semi-supervised learning algorithms, is an alternative and offers a good trade-off of supervision. In this paper, we explore the use of constraints with deep clustering approaches to process satellite image time series. Our experimental study relies on deep embedded clustering and the deep constrained framework using pairwise constraints (must-link and cannot-link). Experiments on a real dataset composed of 11 satellite images show promising results and open many perspectives for applying deep constrained clustering to satellite image time series.

Keywords: time series, satellite images, remote sensing, clustering, constraints, deep embedded clustering

1 Introduction

Deep learning is widely used in a large panel of domains, and sees a high interest in the remote sensing community [25]. It achieves great results but is highly dependant on the amount of available data, and more specifically labeled data. Remote sensing produces a large amount of data which often lacks of annotations. This problem is even more pregnant for times series of satellite images. Satellite images can be freely acquired every 5 days, however, due to the complexity of the data and the lack of a well defined typology, it is difficult to use supervised approaches. Therefore multiple clustering methods are applied in the domain [1, 2]. But even if fully labelled data are not available, there are still a lot of background knowledge available from experts.

Constraints are a good way to exploit this knowledge, in this paper we will focus only on pairwise constraints, must-link (ML) and cannot-link (CL). They are widely used and well studied [3], which makes it easier to have a comparison baseline, as they are implemented by a lot of different methods. These constraints indicate that two instances should be assigned to the same cluster (must-link) or that they should be assigned to different clusters (cannot-link).

In this paper, we want to study if we can benefit from the advances in deep learning through a constraint-based framework that seems more appropriate to the remote sensing domain. We first introduce related works on clustering for time series and constrained clustering, respectively in section 2.1 and 2.2, then we present the deep constrained clustering framework and its adaptation to time series in section 3, then we compare the results to classic constrained clustering on remote sensing data in section 4. Finally, we discuss our results and the future works in the section 5.

2 Related Work

2.1 Clustering for time series

Different approaches for clustering of time series have been proposed through time, mostly based on some representation methods like Discrete Wavelet Transform [5] or similarity measures like Dynamic Time Warping [6]. Those methods are then usually used to be incorporated into some standard clustering algorithms from the k-means, k-medoid, spectral or hierarchical clustering families, as shown in the review by Aghabozorgi et al. [4]. In this field, some improvements are still made, with, e.g., the k-shape algorithm [7], which is based on an iterative refinement procedure that uses a normalized version of the cross-correlation measure. One of the difficulties with time series is the heterogeneity of the related topics and the types of data, starting from the number of features or the sequences length, to the type of correlation between elements, based on shape or structure, with different amplitudes.

This issue has been tackled in the supervised learning by the rise of representation learning through deep neural networks. Recently some deep clustering approaches have been proposed. They are essentially based on some end-to-end architectures that simultaneously learn an embedding for the data and a clustering assignment, through an autoencoder and a clustering layer plugged on the encoder output [8, 9]. A derived architecture have been developed for time series, that uses a 1D-CNN followed by a Bi-LSTM as autoencoder, to preserve the time dimension in the encoded features, they are then clustered by a similarity metric as a clustering layer [10].

2.2 Constraints in clustering

A lot of works have been done on constraints integration for clustering. Most of them rely on some extension of standard clustering algorithm like k-means [14] or spectral approaches [15], but there are also some dedicated methods like constrained programming clustering [16]. A comparative study has been done on the subject on time series [17].

In the deep learning domain, most of the semi-supervised methods refer to self-learning approaches or other way to include knowledge in supervised task. But recently some works have been proposed to include pairwise constraints in deep clustering networks [13, 12]. Both of these papers use constraints as an input for an extended loss function that tends to maximize the similarity of the encoded values between elements of a must link constraints and respectively minimize it for a cannot link constraints. Our work is based on the paper of Zhang et al. [13]. Most of our work consists in adapting this method to time series and study its results on satellite image time series. This method handles different types of constraints, but, as indicated above, we focus on ML and CL in this paper.

3 Deep Constrained Clustering Framework and its adaptation to time series

The Deep Constrained Clustering framework (DCC) presented by Zhang et al. [13] is based on a deep clustering method, the Deep Embedded Clustering (DEC) [8] and its improved version (IDEC) [9]. We first describe the IDEC framework, then how it is extended with constraints through the DCC framework and finally how we adapted it to time series.

3.1 Improved Deep Embedded Clustering

Deep Embedded Clustering (DEC) [8], in the initial step, trains an autoencoder ($x_i = g(f(x_i))$) and then removes the decoder. Then, it fine-tunes the remaining encoder ($z_i = f(x_i)$) by optimizing the Kullback Leiber divergence between two distributions Q and P. Q is a soft cluster assignment, where for each instance i we compute a vector q_i of length k, k being the number of expected clusters, where q_{ij} is the degree of belief that the instance i belongs to cluster j. P is the target distribution that is defined from Q as a "hard" assignment vector that assign i to only one cluster. We have the following loss, L_c, as clustering loss:

$$L_c = KL(P|Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{1}$$

where q_{ij} is the similarity between the embedded point z_i and cluster centroid μ_j measured by Student's t-distribution [18]:

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||^2)^{-1}}{\sum_j (1 + ||z_i - \mu_j||^2)^{-1}} \tag{2}$$

and p_{ij} is the target distribution defined as:

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})} \tag{3}$$

The set of centroids μ is initialized using a k-means on z. The improvement from IDEC [9] is to keep the decoder and the reconstruction loss L_r even after the initialization. The intuition behind it is that the clustering loss, by distorting the embedding space, may alter the representativeness of embedded features and thus the clustering performance, clusters being no more meaningful. Therefore, the clustering loss improves separability of clusters, while the reconstruction loss keeps clusters matched to the features learned by the autoencoder in the first step. The reconstruction loss is computed as the mean squared error between the input time series and the output of the autoencoder. So we have a combined loss that is defined as follow:

$$L = L_r + \gamma * L_c \tag{4}$$

where $\gamma > 0$ is a coefficient that controls the degree of distorting embedded space.

3.2 Constraints integration

The extension of DEC to incorporate constraints is based on the Deep Constrained framework (DCC) [13]. They propose four types of constraints, but we only took in consideration pairwise constraints, because they are supported by various types of constrained clustering algorithm.

The loss function used for must-link constraints set ML is:

$$l_{ML} = L_r - \gamma_{ML} * \sum_{(a,b)\in ML} log \sum_j q_{aj} * q_{bj} \tag{5}$$

In the same way, the loss function for cannot-link constraints set CL is:

$$l_{CL} = - \sum_{(a,b)\in CL} log(1 - \sum_j q_{aj} * q_{bj}) \tag{6}$$

In an intuitive way, ML loss prefers instances with same soft assignments and the CL loss prefers instances with opposite assignments. The ML loss is mitigated by a coefficient $\gamma_{ML} > 0$ and the addition of the reconstruction loss L_r in a similar way that the clustering loss L_c to prevent the method to assign all elements to only one cluster.

3.3 Application to satellite image time series

The main purpose of these experiments is to see if this new type of constrained clustering can be applied on satellite image time series and how it competes with the state of the art. We tested the original DCC framework, which is composed of fully connected layers. The network architecture remains unchanged, the time series is only flattened before being fed to the encoder. We also proposed a modified version of DCC with 1D-convolutional layers, as it proved to work well on time series supervised classification [19]. In this new version, we keep the original input dimension and the network is composed only of 1D-convolutional layers followed each time by a batch normalization layer and the embedding layer is preceded by a global average pooling layer. The embedding layer remains a fully connected layer. We did not include the method in [10] yet, as it relies on the choice of a similarity metric for the target distribution.

4 Experiments and results

4.1 Dataset and experimental setup

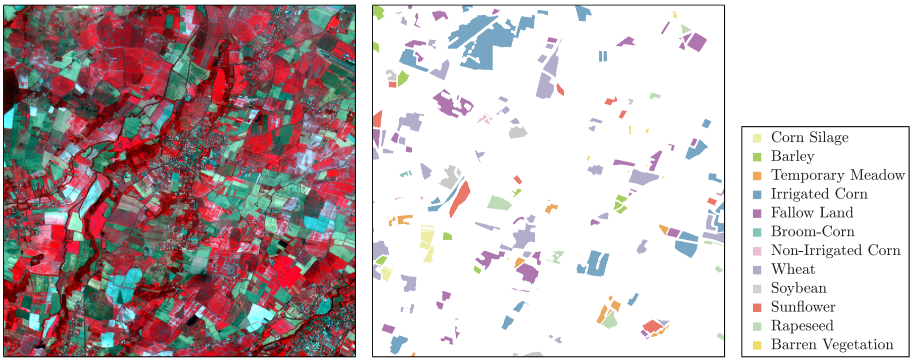
For these experiments, we apply these methods on crop classification which is an important field of research in remote sensing that has seen numerous study [21, 20]. The dataset is composed of 12 class of different kind of crops (wheat, corn silage, irrigated corn, ect, see Fig. 1c), located near Toulouse (Southwest France). The original set of images[3] is composed of 11 multispectral (green, red, and near-infrared) 1000×1000 pixel images non-uniformly sampled from 15/02/07–20/10/07 and captured by the Formosat-2 satellite. One of the images is presented in Fig. 1a. The dataset is composed of pixels randomly selected within the annotated areas (see Fig. 1b) that were then split into a train and test sets composed of 1974 and 9869 pixel time series respectively. The algorithms are evaluated using the test set. The train set is only used to set hyperparameters for the spectral clustering method (Spec). Constraints are generated from the test set by randomly sampling pairs of pixels and creating an ML or CL constraint depending upon their labels. The reference data is based on farmer's declaration to the EEA's Common Agricultural Policy. To test how methods benefit from constraints, we define three levels of constraints size: 5%, 15% and 50% of the cardinality of the dataset $N = 9869$ (a very small fraction of the number of possible constraints, $\frac{1}{2}N[N-1]$). For the evaluation we used the Adjusted Rand Index (ARI) and the constraints satisfaction rate (Sat.) averaged over ten runs. For each level of constraints, ten random sets of constraints were generated to be used at each run. This ensures that each method benefits from the same constraints.

4.2 Methods compared and parametrization

To have a comparison baseline, we add to the two deep constrained clustering methods, four standards constrained clustering methods. We use a constrained k-means algorithm (COP-KMeans) [14], a spectral method (Spec) [15], a declarative method (CPClustering) [16] and a collaborative method with 3 k-means agents (SAMARAH) [23]. To highlight the variability caused by the choice of metric, we use the Euclidean and the DTW metrics [6][4].

[3] Provided by the *Centre d'Études Spatiales de la Biosphère (CESBIO) Unité Mixte de Recherche CNES-CNRS-IRD-UPS*, Toulouse, France.

[4] Implementations for compared method available at from `https://icube-forge.unistra.fr/lampert/TSCC`



(a) Image (b) Reference Data (c) Class Labels

Fig. 1. An image from the time series: 12 classes, and 11 time points (t_4 displayed here).

Note that CPClustering does not require any initialization parameters. The others need at least the number of clusters, otherwise default parameters were used. The only exception is the Spec method which needs some hyperparameters that are defined by grid search on a the train set. For deep clustering, we follow the settings in DEC [8] and IDEC [9] but as the results were not stable we had to do some minor changes (see section 4.3 for more details). For the layers dimensions we set the embedding layer to a dimension of 2 instead of 10, because it seems to give more stability. For the DCC, the encoder network is set to dimensions $d - 500 - 500 - 2000 - 2$, where $d = l * f$ and l is the length of the input time series and f the number of features per time step, and for DCC-conv the dimensions are $l * t - 128 - 256 - 128 - 2$, the corresponding 1D-filters have a dimension of $8 - 5 - 3$, following Wang et al. [24]. For each of them the decoder is a mirror of the encoder. For the optimizer both of them are trained using a SGD with momentum of 0.9 and a decay value of $1e - 6$, to compensate the variability mentioned before. γ and γ_{ML} are both set to 0.1 as described in the original papers, these values should be set as small as the learning rate used is high, otherwise the effect of the distortion is too important if both are high, or negligible if both are low. In our case, the learning rate is set high (0.1) so the clustering loss should have less weight (0.1) than the reconstruction loss, more explanations can be found in [9]. [5]

4.3 Results

The results with and without constraints are presented in Table 1. Spec gives the best overall results, but that must be mitigated, as mentioned before, by

[5] The source code used for this paper: `https://github.com/blafabregue/DeepConstrainedClustering`

Table 1. Unconstrained and Constrained ARI and constraint satisfaction. The best performances for each measure, constraint fraction, and distance measure are highlighted in bold. Unconstrained satisfaction was measured using the 50% constraint sets.

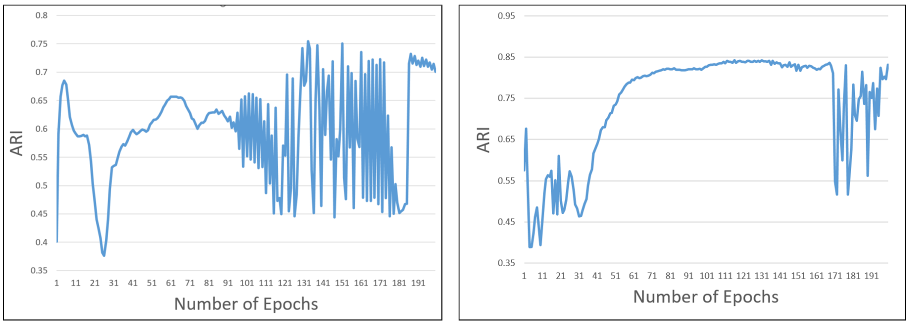
Method	Distance	Unconstrained		5%		15%		50%	
		ARI	Sat.	ARI	Sat.	ARI	Sat.	ARI	Sat.
COP-KMeans [14]	DTW	0.426	0.812	0.416	**1.00**	0.407	**1.00**	0.436	**1.00**
	Eucl.	0.420	0.807	0.406	**1.00**	0.443	**1.00**	0.369	**1.00**
Spec [15]	DTW	**0.531**	**0.840**	**0.683**	0.867	**0.725**	0.888	0.786	0.911
	Eucl.	**0.737**	**0.885**	0.671	0.854	0.702	0.875	0.781	0.916
CPClustering [16]	DTW	0.437	0.803	0.469	**1.00**	0.510	**1.00**	0.589	**1.00**
	Eucl.	0.681	0.413	0.650	**1.00**	0.542	**1.00**	0.510	**1.00**
SAMARAH [23]	DTW	0.406	0.802	0.597	0.870	0.637	0.867	0.681	0.878
	Eucl.	0.463	**0.817**	**0.691**	0.884	**0.714**	0.890	0.702	0.885
DCC [13]		0.703	0.885	0.550	0.852	0.448	0.816	0.615	0.862
DCC-Conv		0.508	0.833	0.497	0.844	0.491	0.819	**0.820**	0.936

the fact that it needs a training for its hyperparameters (for unconstrained its average results is at 0.367 against the retained configuration at 0.737). We can also point out that the result depends on the chosen similarity metric. This is also the case for the other metric-based methods at the exception of COP-KMeans which performs badly in both cases. The deep clustering only proves to be more efficient when the number of constraints are very high and only with the convolutionnal architecture. But the most intriguing point is the partially opposite behavior of the two architectures.

DCC gives relatively good results in unconstrained configuration, but the constraints have a strong negative effect. This effect has already been studied in constrained clustering, and can be observed for all other methods at the exception of SAMARAH. It was observed in Lampert et al. [17] that if the algorithm already captures well the structure of the data it will not benefit, or even have a negative effect, from constraints addition. This seems to be the case for DCC. This goes in an opposite conclusion than Zhang et al. [13], which concludes to no negative effect. In our case this can be explained by the noisy ground truth used (presence of trees or road across the field, problem of frontier pixels).

DCC-Conv, in the other hand, obtains good results with constraints, but only if the number of constraints is high enough. Indeed the way constraints are used, in the backpropagation, does not enforce the algorithm to respect constraints, and needs a lot of information to be sure that it propagates to the weights. On this point DCC-Conv really shows its specificity, it was observed in Lampert et al. [17] that methods does not necessarily benefits from a higher number of constraints, but mostly from informative and coherent constraints. In this case, the ARI seems linked to the number of constraints which may let think that the network starts to learn the dataset itself and not the structure, the algorithm being trained and learn on the same set.

Two other points have to be highlighted, that are not displayed in Tab 1. First the standard deviation strongly increases with constraints and stabilizes again when the number of constraints is higher, for DCC and DCC-Conv (i.e. for DCC-Conc, standard deviation is respectively for unconstrained, 5%, 15%, 50% at 0.005, 0.069, 0.010 and 0.015). This seems to show that the quality of constraints is important, the impact of noise might therefore be smoothed by the number of constraints. The second one is that the network is not stable during the training, this is the case for both architectures but in a higher amplitude for DCC. This tends to stabilize, due to the decay added in the optimizer, but the variability remains (i.e. for DCC, ARI goes from 0.55 to 0.74 to go down to 0.45). An illustration of this instability can be seen in figure 2. In a first step, the ARI increase smoothly, but then it starts to oscillate and struggle to converge again. This is mostly the case for constrained runs and but also, in a smaller amplitude for unconstrained runs. This seems to come mainly from the fact that the target distribution (the hard assignment) is updated regularly, so the objective function does not aim the same target.



(a) DCC with 50% constraints (b) DCC-Conv with 50% constraints

Fig. 2. Evolution of ARI measure through the training process compared between DCC and DCC-Conv.

5 Conclusion

The deep clustering demonstrates that it can achieve good results on remote sensing time series without and with constraints. Moreover, it does not require to choose a representation method as it is learned by the autoencoder, which makes it easier to handle for a domain expert. However the role played by the hyperparameters (i.e. dimensions of layer, especially the embedding layer, optimizer) needs some further investigations, as they seem to influence the output quality. There are also two main problems that we plan to study. First, DCC is not as robust as expected from the result of the previous studies. Second, the instability in the training and in the impact of constraints lowers the average result. We plan to study which factors may induce these problems, and further investigate the effect of noisy constraints, the size of the dataset and how constraints can be differently integrated. Finally, we also plan to study how the learned network behave on transfer learning.

Acknowledgments. This work was supported by the ANR TIMES project (grant ANR-17-CE23-0015) of the French Agence Nationale de la Recherche.

References

1. Khiali, L., Ndiath, M., Alleaume, S., Ienco, D., Ose, K., & Teisseire, M. (2019). Detection of spatio-temporal evolutions on multi-annual satellite image time series: A clustering based approach. International Journal of Applied Earth Observation and Geoinformation, 74, 103-119.
2. Rey, D. M., Walvoord, M., Minsley, B., Rover, J., & Singha, K. (2019). Investigating lake-area dynamics across a permafrost-thaw spectrum using airborne electromagnetic surveys and remote sensing time-series data in Yukon Flats, Alaska. Environmental Research Letters, 14(2), 025001.
3. Basu, S., Davidson, I., & Wagstaff, K. (Eds.). (2008). Constrained clustering: Advances in algorithms, theory, and applications. CRC Press.
4. Aghabozorgi, S., Shirkhorshidi, A. S., & Wah, T. Y. (2015). Time-series clustering–A decade review. Information Systems, 53, 16-38.
5. Chan, K. P., & Fu, A. W. C. (1999, March). Efficient time series matching by wavelets. In Proceedings 15th International Conference on Data Engineering (Cat. No. 99CB36337) (pp. 126-133). IEEE.
6. Sakoe, H., Chiba, S., Waibel, A., & Lee, K. F. (1990). Dynamic programming algorithm optimization for spoken word recognition. Readings in speech recognition, 159, 224.
7. Paparrizos, J., & Gravano, L. (2015, May). k-shape: Efficient and accurate clustering of time series. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (pp. 1855-1870). ACM.
8. Xie, J., Girshick, R., & Farhadi, A. (2016, June). Unsupervised deep embedding for clustering analysis. In International conference on machine learning (pp. 478-487).
9. Guo, X., Gao, L., Liu, X., & Yin, J. (2017, June). Improved Deep Embedded Clustering with Local Structure Preservation. In IJCAI (pp. 1753-1759).
10. Sai Madiraju, N., Sadat, S. M., Fisher, D., & Karimabadi, H. (2018). Deep Temporal Clustering: Fully Unsupervised Learning of Time-Domain Features. arXiv preprint arXiv:1802.01059.
11. Dau, H. A., Begum, N., & Keogh, E. (2016, October). Semi-supervision dramatically improves time series clustering under dynamic time warping. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 999-1008). ACM.
12. Ren, Y., Hu, K., Dai, X., Pan, L., Hoi, S. C., & Xu, Z. (2019). Semi-supervised deep embedded clustering. Neurocomputing, 325, 121-130.

13. Zhang, H., Basu, S., & Davidson, I. (2019). Deep Constrained Clustering-Algorithms and Advances. arXiv preprint arXiv:1901.10061.
14. Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001, June). Constrained k-means clustering with background knowledge. In Icml (Vol. 1, pp. 577-584).
15. Li, Z., Liu, J., & Tang, X. (2009, June). Constrained clustering via spectral regularization. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 421-428). IEEE.
16. Duong, K. C., & Vrain, C. (2017). Constrained clustering by constraint programming. Artificial Intelligence, 244, 70-94.
17. Lampert, T., Lafabregue, B., Serrette, N., Forestier, G., Crémilleux, B., Vrain, C., & Gancarski, P. (2018). Constrained distance based clustering for time-series: a comparative and experimental study. Data Mining and Knowledge Discovery, 32(6), 1663-1707.
18. Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), 2579-2605.
19. Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019). Deep learning for time series classification: a review. Data Mining and Knowledge Discovery, 1-47.
20. Garnot, V. S. F., Landrieu, L., Giordano, S., & Chehata, N. (2019). Time-Space tradeoff in deep learning models for crop classification on satellite multi-spectral image time series. arXiv preprint arXiv:1901.10503.
21. Sicre, C. M., Baup, F., & Fieuzal, R. (2014). Determination of the crop row orientations from Formosat-2 multi-temporal and panchromatic images. ISPRS journal of photogrammetry and remote sensing, 94, 127-142.
22. Kamvar, K., Sepandar, S., Klein, K., Dan, D., Manning, M., & Christopher, C. (2003, April). Spectral learning. In International Joint Conference of Artificial Intelligence. Stanford InfoLab.
23. Forestier, G., Gançarski, P., & Wemmert, C. (2010). Collaborative clustering with background knowledge. Data & Knowledge Engineering, 69(2), 211-228.
24. Wang, Z., Yan, W., & Oates, T. (2017, May). Time series classification from scratch with deep neural networks: A strong baseline. In 2017 International joint conference on neural networks (IJCNN) (pp. 1578-1585). IEEE.
25. Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. IEEE Geoscience and Remote Sensing Magazine, 5(4), 8-36.