Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2018-53-AC2, 2018
© Author(s) 2018. This work is distributed under
the Creative Commons Attribution 4.0 License.

Geoscientific Model Development Discussions

Interactive comment on "Topological Data Analysis and Machine Learning for Recognizing Atmospheric River Patterns in Large Climate Datasets" *by* Grzegorz Muszynski et al.

Grzegorz Muszynski et al.

muszyna23@gmail.com

Received and published: 26 October 2018

General comments

C: The paper presents a method to detect atmospheric rivers (ARs) in climate datasets. Unlike most existing methods, this one relies on marching learning and learns a classification rule for the detection of ARs based on a training dataset. In my opinion, one novelty of the paper lies in the choice of the features used for the classification. From maps of integrated water vapor, new features are constructed from topological data analysis that could me more suited for the problem. In general, I think that the paper is

well written and I appreciate the pedagogical effort made to clearly explain the methodology as well as the illustrations of cases where the algorithm performs well and not so well. Hence, I don't see any major reasons not to published the paper. I only have a few comments and suggestions that I think could benefit the paper.

A: We thank the reviewer for the valuable comments, which we think will improve the quality of the paper.

Specific comments

C: I find the use of the term "threshold-free" is maybe not the most appropriate. While I understand that in most of the cases, "threshold-free" means that the method does not rely on a fixed, predetermined, arbitrary threshold for the detection of ARs, thresholds are still used several times during the proposed procedure. Indeed, the goal of the SVM step is still to learn a threshold to separate the ARs from non-ARs from the training set and the topological features. The topological features are also constructed from a set of thresholds. (And to be more provocative, for now, the labels in the training set were also generated by an AR detection methods using thresholds). For me, the value of the paper is that it shows that if a we have a good training dataset, there is more efficient way to build this decision threshold than manually tinkering parameters of the classifier/detector.

A: We thank the reviewer for pointing out the potential confusion in the term "threshold-free". To clarify, we are specifically referring to the deficiency of most existing AR-detection techniques that use a fixed, predetermined, arbitrary threshold on certain physical variables in order to detect ARs. In our approach, the topological feature extraction is threshold free in the sense that we do not choose any fixed or predetermined thresholds to calculate features for the detection of ARs. In particular, topological features (invariants), in this case connected components/regions, are computed under all

possible values of certain parameter, here the Integrated Water Vapour (IWV). This implies that the connected components do not rely on specific choice of a threshold value, i.e. “threshold-free”.

However, the term “threshold-free” may be confusing for readers because indeed the training data uses a heuristic algorithm that has built-in thresholds on IWV. We mention on page 15, lines 11-13 and on page 18, lines 3-5 that an imperfect “ground truth” training set generated by the AR detection heuristic implemented in TECA (Prabhat, et al., 2015) is biased by using the fixed threshold based criteria for AR identification. In future work, we plan to test the method by training the classifier on datasets that are manually labelled. This should circumvent the problem of the classification results biased by fixed thresholds used for generating the ground truth data.

On the other hand, we would like to clarify that the SVM does not learn a threshold from the topological features to separate the ARs from non-ARs. Instead, the SVM finds a transformation of the topological vectors into a high dimensional space where ARs and non-ARs are separable by a suitable hyper-plane (for clarity this has been included in the paper on page 8 , lines 11-15).

C: In the same way, I am not sure I understand the following sentence from the abstract and the conclusion (p17, l-14-15) : “We anticipate that because the method is threshold-free, it can be applied to different climate change scenarios without any tuning”. If the statistical relationships between features and the target variable change through time, should not you retrain the SVM as the other methods have to reevaluate their thresholds?

A: Yes, if the distribution of data and target variables change over time, the SVM model should be retrained.

However, by “... it can be applied to different climate scenarios without any tuning.” we refer to “Stage 1” of the method, i.e. topological feature extraction. To clarify, there is no

need to determine any threshold criteria for this topology-based AR detection method. Hence, when the spatial resolution of the climate model changes or a different climate scenario is examined, there is no parameter re-tuning, unlike in the case of heuristic methods used by most other AR-detection methods, e.g., TECA (Prabhat, et al., 2015).

C: I think the explanation on the SVM could be improved if Figure 7 was split into 2: the first figure would illustrate the (linear) SVM and the different quantities in equations (3) and (4) (see for e.g. https://en.wikipedia.org/wiki/Support_vector_machine#/media/File:Svm_max_sep_hyperplane_with_margin.png). The second figure would focus more on the “kernel trick”. For instance, it would show a case were a linear classifier could not separate the two classes in a 2D space but would managed to do it if data were mapped into a 3D space.

A: We thank the reviewer for this excellent suggestion of dividing Figure 7 into two figures (i.e., Fig. 7 and Fig. 8). These two figures has been attached to this response and will be included in our revised manuscript.

Fig. 7: An example of linear SVM that finds the optimal hyperplane $w^T\phi(x) + b = 0$, its maximum-margin $\frac{2}{\sqrt{w^T w}}$ separating samples from two classes in data (blue dots and red stars), and all other quantities in the equations (3), (4). ζ is a variable defining how much on the ‘wrong’ side of the hyperplane a sample is: if it is $1 > \zeta > 0$, the point is classified correctly, but by less of a margin than the optimal hyperplane was found, else if it is more than $\zeta > 1$, the point is classified incorrectly. The magenta dot indicates an example of misclassified sample from the class of blue dots. Support vectors help to find the margin for the optimal linear hyperplane. $\phi(x)$ is a linear transformation in this case.

Fig. 8: a) An example of no clear linear separation between two classes (e.g., ARs and non-ARs) in data. This case cannot be solved using linear SVM. b) In a situation where the set of two class samples is not linearly separable in the original space the

SVM introduces the notion of a 'kernel function induced feature space' which casts the data into a higher dimensional space where the data is separable.

C: (P9, l7) "The kernel function that maps the input space into a higher dimensional space ...". I think the sentence can be a little bit nuanced. As far as I understand, the kernel function returns the inner product between two points projected into higher dimensional space by a mapping function phi. Each kernel function is implicitly associated with a mapping function phi (which does not need to be known for an actual application and that's one of the strong point of kernel methods). That's why the function phi is called a kernel induced implicit mapping.

A: Yes, this sentence should be rewritten to avoid confusion. We rephrase as follows: "The samples $\{x_i\}$, where $x_i \in R^n$, from the training set are mapped into a high dimensional feature space F by means of the transformation $\phi(x_i)$, where $\phi(x) : R^n \rightarrow F$. This transformation makes the samples of two groups (ARs and Non-ARs) separable, as shown in Figure 8. Then, the similarity between observations x_i and x_j is computed by kernel function $K(x_i, x_j)$ that can be expressed as an inner product $\langle \phi(x_i), \phi(x_j) \rangle_F$ in the feature space F. Hence, it is sufficient to know $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_F$ rather than $\phi(x)$ explicitly (Burges, 1998).

C: (P9, l12) "applying loose grid-search and fine grid-search for these two parameters". Do you use grid search with some kind of cross-validation scheme?

A: Yes, we used a stratified k-fold cross-validated grid-search to find optimal values of parameters C and γ regarding the SVM classification performance (Hsu et al., 2003). We split a training set into k folds of equal size in a such manner that k folds do not overlap one another. Then, iteratively, one fold was chosen for testing and the remaining k-1 folds were used for training the classifier. Since a grid-search is time-consuming we used a coarse grid-search on training sets to identify the range of C

and γ values with respect to classification performance. Then, we conducted a fine grid-search for the identified range of the parameters to select the optimal parameters regarding classification performance.

C: I think it should be clearly mentioned in the main text or in a table how many data points were used in the training set and the test sets. We could try to deduce it from confusion matrices but it is not very practical.

A: This comment will be included in the revised manuscript.

C: In the same way, for table 3, 4, etc . . . , the number of snapshots mentioned, is it for the test or training sets?

A: The number of snapshots mentioned in the manuscript represents the total number of samples of both classes (AR & Non-AR) after resampling was applied to the original datasets due to the imbalanced class problem. Resampling to solve the class imbalance problem has been mentioned on page 11, lines 13-16.

C: (P18, l1), Authors compare the computing time of their algorithm with the one of Liu et al. (2016) thats uses deep learning. How do both methods compare in terms of performances ?

A: Since both models were trained and tested on different datasets, containing data on different geographical regions, the performance of both models cannot be directly compared.

C: For the sake of reproducibility, it would be nice to at least provide in supplementary materials, details about the actual implementation of the methods. For instance, the

programming language used, the potential external softwares/packages/libraries used and for which step of the method.

A: We will provide details about the implementation of the method in supplementary materials of the revised manuscript. The TDA algorithm is implemented in C++ and is compatible with TECA software (Prabhat et al., 2015). The SVM model was imported from Python skitlearn modules are available at http://portal.nersc.gov/project/m1517/cascade/doi/GMD_2018/GMD_2018.html.

References

1) Burges, Christopher JC. “A tutorial on support vector machines for pattern recognition.” Data mining and knowledge discovery 2.2 (1998): 121-167.
2) Prabhat, S. Byna, et al. “TECA: Petascale pattern recognition for climate science.” International Conference on Computer Analysis of Images and Patterns, 2015.
3) Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. “A practical guide to support vector classification.” (2003): 1-16.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2018-53, 2018.

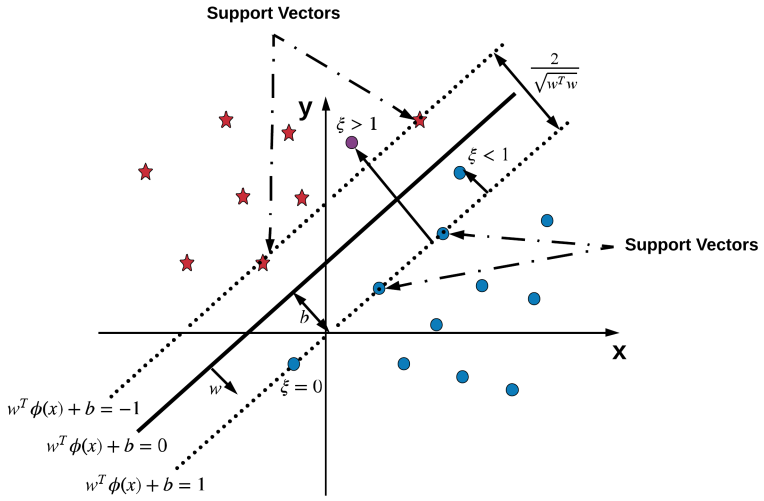

Fig. 1. Figure 7

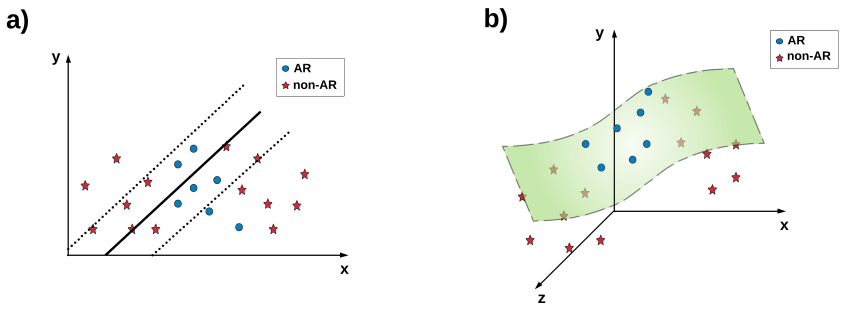

Fig. 2. Figure 8