

Clustering Dynamic Spatio-Temporal Patterns in the Presence of Noise and Missing Data

Xi C. Chen, James H. Faghmous, Ankush Khandelwal, Vipin Kumar

University of Minnesota
Minneapolis, MN, 55414, US
{chen, faghmous, ankush, kumar}@cs.umn.edu

Abstract

Clustering has gained widespread use, especially for static data. However, the rapid growth of spatio-temporal data from numerous instruments, such as earth-orbiting satellites, has created a need for spatio-temporal clustering methods to extract and monitor dynamic clusters. Dynamic spatio-temporal clustering faces two major challenges: First, the clusters are dynamic and may change in size, shape, and statistical properties over time. Second, numerous spatio-temporal data are incomplete, noisy, heterogeneous, and highly variable (over space and time). We propose a new spatio-temporal data mining paradigm, to autonomously identify dynamic spatio-temporal clusters in the presence of noise and missing data. Our proposed approach is more robust than traditional clustering and image segmentation techniques in the case of dynamic patterns, non-stationary, heterogeneity, and missing data. We demonstrate our method's performance on a real-world application of monitoring in-land water bodies on a global scale.

1 Introduction

Spatio-temporal data are rapidly becoming ubiquitous thanks to affordable sensors and storage. These information-rich data have the potential to revolutionize diverse fields such as the social, earth, and medical sciences where there is a need to extract and understand complex spatio-temporal phenomena and their dynamics. Additionally, the data in such scientific domains tend to be large and unlabelled. This highlights the importance of unsupervised methods in monitoring spatio-temporal dynamics with little or no human supervision.

Clustering is one of the most common unsupervised data mining techniques. It has enjoyed tremendous success, especially for static data [Jain and Dubes, 1988]. Yet, there is little work in the spatio-temporal setting where data is in the form of continuous spatio-temporal fields and the clusters are dynamic. Furthermore, spatio-temporal data that originate from earth-orbiting satellites, cell phones, and other sensors tend to be noisy, incomplete, and heterogeneous, making their analysis especially challenging [Faghmous and Kumar,

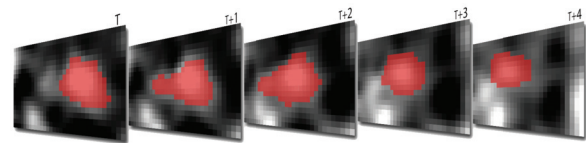


Figure 1: An example of a dynamic spatio-temporal cluster. In each time step (t_i) we must extract the cluster from the continuous field (red pixels) and track its evolution over time. Such patterns are common in the Earth Sciences, fMRI analysis, and other domains.

2013]. When dealing with continuous spatio-temporal data, the clusters are embedded in the continuous spatio-temporal field, where these clusters or objects have no clear boundaries. The goal is to isolate such clusters from the background and continuously monitor them over time (see Figure 1).

In this paper, we propose a novel spatio-temporal clustering paradigm to identify clusters in a continuous spatio-temporal field where clusters are dynamic and may change their size, shape, location, and statistical properties from one time-step to the next. Our paradigm stems from the observation that in numerous dynamic settings, although clusters may move or change shape, there are a number of points that do not change cluster memberships for a significant time-period. This observation allows us to autonomously extract dynamic clusters in continuous spatio-temporal data that may contain missing values, noise, or highly-variable features. We demonstrate our paradigm on a real-world application of monitoring in-land water bodies (e.g. lakes, dams, etc.) using remotely-sensed data on a global scale. We compare our method's performance to the K-MEANS and Expectation-Maximization (EM) clustering algorithms as well as the Normalized Cuts (NCUT) image segmentation algorithm, and find that our method's ability to leverage both spatial and temporal information makes it more robust to noise, missing data, and heterogeneity – common characteristics of emerging spatio-temporal datasets.

2 Background and Related work

2.1 Problem formulation

The goal of this work is to autonomously extract dynamic clusters from a continuous spatio-temporal field. An fMRI recording or remotely sensed data are examples of continuous

spatio-temporal fields, where each location has unique spatial coordinates and is characterized by uni/multi-dimensional time-series representing the evolution of the feature vector over time. Thus, given a continuous spatio-temporal field, our task is to identify all clusters in space and associate the clusters across time.

2.2 Existing clustering approaches

Clustering is a common data mining technique that groups similar points together to reveal high-level patterns in a dataset. Clustering algorithms may belong to two broad categories: feature-based clustering and constraint-based clustering.

Feature-based clustering algorithms group data based on their similarity in the corresponding feature space, without considering other information. Many popular clustering algorithms including K-MEANS [MacQueen and others, 1967], EM [McLachlan and Krishnan, 2007], LINKAGE (e.g., single-linkage [Sibson, 1973], complete linkage [De-fays, 1977]) and DBSCAN [Ester *et al.*, 1996] all belong to this category.

Constraint-based clustering algorithms assign data to clusters based on additional constraints other than similarity in the feature space. For example, many image segmentation algorithms (e.g. [Shi and Malik, 2000; Enright *et al.*, 2002]) can be considered to be clustering methods with spatial constraints. They cluster data into spatially connected patches such that data from the same cluster have similar feature values and are also spatially connected. Other clustering algorithms, such as trajectory clustering [Lee *et al.*, 2007], mining swarm/flock patterns [Li *et al.*, 2010] and moving clusters [Li *et al.*, 2004], are other examples of constraint-based clustering. They discover clusters of objects that have similar behavior over time.

Despite the wide applicability of these approaches, they do not address the fundamental needs of many spatio-temporal applications. For example, on the one hand, feature-based methods do not take into account spatial and temporal information that uniquely represent spatio-temporal clusters. Thus, in a feature-based setting, one either clusters the feature values or the spatial locations within data. On the other hand, in a constrained clustering setting, objects are already predefined and are grouped based on some constraints. However, in continuous spatio-temporal fields, there is no clear definition of an object, thus these methods have limited applicability. Finally, there have been works that cluster continuous spatio-temporal data into static clusters, such that the resulting clusters explain the data for the entire temporal duration (e.g. [Birant and Kut, 2007; Steinbach *et al.*, 2003]). Such approaches are not well-suited for the discovery of clusters over every time-step in the data, especially when the clusters are dynamic and may change size, shape, location and statistical properties over time.

A desirable solution is one that can isolate clusters that have similar feature values over space and time, while also keeping track of such clusters as they evolve. The most common approach to tackle this problem has been to either analyze the data in space and then aggregate/associate over time, or by analyzing over time and then smoothing over space

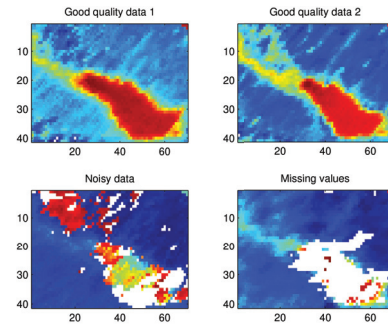


Figure 2: Examples of data challenges associated with spatio-temporal data. The data represent a remotely-sensed “wetness index” to estimate surface wetness. Each panel shows the wetness index of the same location at four different dates. As it can be seen, data quality often varies with time. The top row shows two dates where the data is of good quality, while the bottom row shows data with noisy and missing values (white pixels).

[Roddick and Spiliopoulou, 1999]. However, there is mounting evidence that such an approach may lead to false discoveries (e.g. [Davidson *et al.*, 2013; Faghmous *et al.*, 2014]).

2.3 Challenges

In addition to technical limitations, clustering spatio-temporal data faces significant data challenges in many real world applications. Figure 2 shows some of the common data challenges associated with analyzing spatio-temporal data. The data are routinely missing and noisy. Thus, analyzing the data on a snapshot-by-snapshot basis, or while disregarding spatial information would lead to inadequate performance.

Another significant challenge is heterogeneity in space and time [Faghmous and Kumar, 2014]. Heterogeneity in space refers to the case where data belonging to the different clusters may have the same feature values, despite being distinct “objects”. Temporal heterogeneity refers to the instance where the feature values that uniquely discriminate a cluster change over time for the same cluster. Figure 3 demonstrates the concept. The left panel shows the “wetness index” values for a region containing two lakes surrounded by land. On the top right panel, one notices a clearly distinguishable signal in the feature space (as seen by the bi-modal distribution of the feature values). However, in another time-step (bottom right panel) the feature space is not as informative due to spatio-temporal heterogeneity. Thus, relying solely on information in one time-step would yield inaccurate results.

3 A Spatio-Temporal Clustering Paradigm

To address the above-mentioned challenges, we propose a general spatio-temporal clustering paradigm that systematically leverages the very challenges that affect traditional methods to identify dynamic spatio-temporal clusters. Our paradigm consists of two main steps: identifying the most certain cluster memberships and iteratively finalizing the most uncertain points which will likely be at the cluster boundaries where dynamics occur. Figure 4 outlines the four key steps. The first step specifies the clustering objectives such as separating certain activity from the background, or

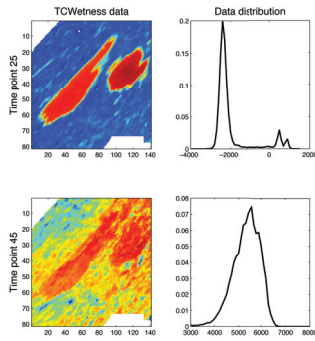


Figure 3: An example of data heterogeneity. Each row shows the “wetness index” values for the same lake at different time steps. The right panel shows the density of the feature values (pixel colors) on the left. In the top row, the water and land are easily distinguishable in the feature space however in the bottom row the two clusters are not distinguishable. Thus, relying solely on that time-step would yield inaccurate results.

labeling the clusters with target labels. The second step is to identify “core points”, which are the points that for a given time-window do not change cluster memberships. The key here is to choose an appropriate time window size. In practice, one could identify core points for each snapshot by examining data from the previous and upcoming time-steps. The third step is to finalize cluster memberships along the cluster’s boundary. Given the dynamic nature of the clusters and the uncertainty in the data, the boundary points are going to be more challenging to cluster. While the exact approach may differ, the idea is to use information from the core points (especially the ones that are spatially nearby) to finalize cluster assignments. The fourth and final step is to post-process the cluster result in case not all points have been labeled.

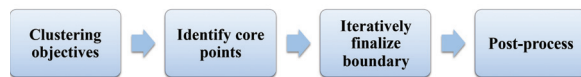


Figure 4: Our proposed four-step spatio-temporal clustering paradigm.

4 Proposed method

The proposed paradigm takes advantage of the fact that in many domains, although the clusters may move, there are “core points” that never change cluster memberships for a given time window. This is an important observation when the data may be missing or noisy between time-steps, as such, although clusters might not be separable during every single time-step, borrowing stronger signals from other time steps helps overcome some of these challenges.

While there are many ways to implement this paradigm, this section presents one such realization in practice. We hope that introducing this paradigm to the community will allow us to leverage our collective creativity to design a host of methods to analyze space-time data.

4.1 Clustering objectives

The first step in the paradigm is to articulate the objectives of our clustering analysis. In the case of global water monitoring, we are given a single-dimensional spatio-temporal field without any notion of water or land. The goal is to extract clusters and their dynamics over a fifteen-year period and then label each cluster as water and land in the post-process phase. We monitor surface water using a “wetness index”, known as TCWETNESS. TCWETNESS has been widely used in mapping and monitoring land use/land cover by the remote sensing community [Collins and Woodcock, 1996; Coppin and Bauer, 1996]. The steps used to produce TCWETNESS are discussed in detail in [Chen *et al.*, 2015].

4.2 Discover stable clusters

After specifying the clustering objective, the second step of the paradigm is to identify groups of data that rarely change cluster membership for a given time window. Points in any stable cluster are expected to be contiguous in space and also have similar temporal characteristics during the pre-defined time window. The main motivation behind stable clusters is that points are grouped together not only based on their spatial connectivity but also their long-term temporal similarity. This is critical in noisy and incomplete data as the features might not be informative at every time-step, but over a long enough period, similar time-series would emerge.

Our spatio-temporal method that identifies stable clusters is an extension of the traditional DBSCAN algorithm. DBSCAN [Ester *et al.*, 1996] is a density-based clustering algorithm. It groups data that are closely packed together in the feature space. The algorithm identifies “core points” that have at least m neighbors within an ϵ distance in the feature space. Unlike DBSCAN which only considers distances in the feature space, our approach seeks to associate points that are adjacent in space and have similar feature values over a non-trivial time window. Similar to the ST-DBSCAN method proposed by [Birant and Kut, 2007], we use both spatial and temporal information in finding ϵ -neighbors. The main distinction between our approach and [Birant and Kut, 2007] is that we are interested in identifying clusters for every time-step and associating these clusters across time, despite noise and missing data. While [Birant and Kut, 2007] returns only a single cluster for the entire time period. To do so, we propose a new spatio-temporal distance metric.

We use the notion of spatio-temporal ϵ -neighbors to denote two points that are spatially connected and have similar temporal characteristics. Thus, given a distance function that accounts for both spatial adjacency and temporal similarity, two points are spatio-temporal ϵ -neighbors if their spatial-temporal distance is smaller than ϵ . Points that have more than m ST ϵ -neighbors are defined as core points. Once the core points are identified, the algorithm iteratively associates the core points with their spatio-temporal ϵ -neighbors.

The steps of our method is shown in Fig. 5. The algorithm first detects core points (shown as red dots in Fig. (a)) based on a predefined distance function. Then it creates an edge between all core points and their own spatio-temporal ϵ -neighbors (as shown in Fig. (b)). Finally, it groups points that have been connected by an edge as clusters. In the given

example, as shown in Fig. (c), two clusters are discovered (orange and green) and there is a single point (yellow) that does not belong to any cluster.

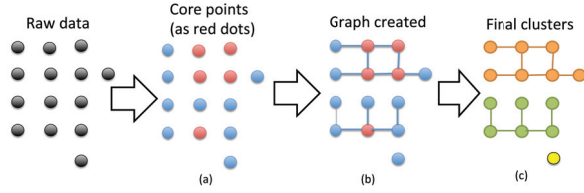


Figure 5: The steps of creating stable clusters from a spatio-temporal data using ST-DBSCAN.

The spatio-temporal distance function that we choose forces two spatio-temporal ϵ -neighbors to be spatially adjacent and have similar temporal characteristic. It is a function as below.

$$d_{st}(x, y) = \begin{cases} d_t(x, y) & \text{if } x \text{ and } y \text{ are spatial neighbors} \\ 0 & \text{otherwise} \end{cases}$$

where, $d_t(x, y)$ is a time-series distance function.

The choice of time series distance function is related to the application. Commonly used time-series distance functions include (but are not limited to) Euclidean distance, Pearson’s correlation and k th order statistic [Chen *et al.*, 2013]. Pearson’s correlation is preferred when the trend of time-series is more important than the actual values. Euclidean distance is susceptible to noise and outliers [Latecki *et al.*, 2005]. The k th order statistic distance is a time series distance function that is robust to outliers. However, it requires an estimation of the outlier properties. In the scenario where the property of the data changes over time and space, the k th order statistic distance is not suitable.

We propose to estimate the distance between two time series based on how similar they are over a period of time.

Definition 1 (Temporal similarity) *Two time series are temporally similar if they have the same expected value for the entire duration.*

To use temporal similarity, we assume that our data follow the additive white noise model, i.e., any real observation of object x at time t , $x(t)$, is the summation of its true value $\hat{x}(t)$ and a random white noise signal $n(x, t)$ as shown below.

$$x(t) = \hat{x}(t) + n(x, t)$$

When two objects x and y are temporally similar, their true value at any time are identical. Hence,

$$x(t) - y(t) = n(x, t) - n(y, t)$$

Since we assume the white noise model, the expectation of any noise is zero. Thus,

$$\begin{aligned} E(x(t) - y(t)) &= E(n(x, t) - n(y, t)) \\ &= E(n(x, t)) - E(n(y, t)) \\ &= 0 \end{aligned}$$

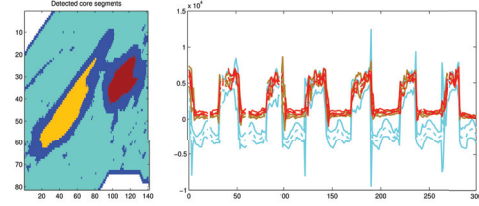


Figure 6: Core segments and their corresponding temporal profiles

Therefore, we can estimate the temporal similarity of two time series x and y as the p -value of the following test.

$$H_0 : E(\mathbf{x} - \mathbf{y}) = 0$$

$$H_a : E(\mathbf{x} - \mathbf{y}) \neq 0$$

Since outliers may negatively impact expectations, an alternate test can be used when the data are susceptible to outliers.

$$H_0 : \text{median}(\mathbf{x} - \mathbf{y}) = 0$$

$$H_a : \text{median}(\mathbf{x} - \mathbf{y}) \neq 0$$

Thus, under the white noise assumption, we propose that two time-series are similar if their difference over the given duration is centered around zero. We can use the p -value of such a hypothesis test, e.g., the Kolmogorov-Smirnov test [Massey Jr, 1951], as the measure of similarity.

An example of stable clusters discovered for an area containing two lakes is shown in Figure 6. The three stable clusters are highlighted in yellow, red and green. The dark blue locations around the clusters are points that we cannot yet assign to any cluster and we must rely on the third step in our paradigm to finalize cluster memberships. We refer to such points as “uncertain points”

The right panel of Figure 6 shows the temporal profile of the clusters in the figure’s left panel. We highlighted the time-series with the same color of the cluster they were assigned to. The yellow and red time-series have very similar temporal profiles and overlap for almost the entire record. However, notice that data in the light blue (land) pixels have different feature values from the yellow and red (water) pixels only during some periods. This is where our temporal similarity over a long time window helps overcome spatio-temporal heterogeneity.

Our temporal similarity measure is sensitive to the choice of time window length. Specifically, the window length w impacts the number of points that will be assigned to stable clusters and/or the number of uncertain points. If we set w to 1 then, our method would be similar to many of the traditional clustering algorithm that disregard time (e.g NCUT) and it would not be able to cluster time-steps where data are missing. If we choose a too broad w then we might include too much uncertainty from the highly variable data signal or the changing properties of the dynamic cluster, which would increase the proportion of “uncertain points”.

4.3 Growing and refining clusters for each time point

The third step of our paradigm finalizes cluster memberships by assigning “uncertain points” to clusters and correcting any



Figure 7: Illustrative example of layer based classifier

assignment mistakes from the previous step. This step relies on the cluster assignments from the previous step to build a spatial predictive model. We use the constructed model to predict the cluster membership of unassigned points based on their feature values in the current time-step. However, given the spatial-heterogeneity in the data, we propose a layer-based classification method which iteratively assigns the uncertain points to existing stable clusters based on spatial proximity and the feature value at the current time-step. Unlike the first step in the paradigm, this classification step only uses information for the current time-step.

Assuming we identified k stable clusters in the previous step, we build a k -class classifier to determine the probability that a given uncertain point belongs to one of the k stable clusters. The model is trained using the feature values of the points in the stable clusters, with each point having a feature value and stable cluster membership. The algorithm then tries to classify the first uncertain point which is at the boundary of a stable cluster using the learned classifier. By assuming clusters are spatially contiguous, an uncertain point is more likely to have the same label as its stable neighbor. Thus after we classify an uncertain point, if its resulting label is the same as its neighbor from a stable cluster we accept that labeling, and move on the next uncertain point. If the label assigned to the uncertain point is inconsistent with its neighbor from the stable cluster, we label the point as “unknown”. The idea is to delay an uncertain labeling until more data are available to make an unambiguous assignment.

Figure 7 illustrates our layer-based method. In this example, there are two stable clusters in green and light blue. The points in white are the uncertain points that we have not labeled. We attempt to assign these uncertain points to existing clusters in the layer-based fashion by first assigning the points that are adjacent to existing clusters. The first layer of uncertain points to be classified are highlighted in red in the left panel of Figure 7. Then these uncertain points (red points) are classified based on their feature values using a classifier trained on the feature values of the points in the stable clusters. The classification results in this example are shown in the middle panel of Figure 7. Finally, the algorithm checks for spatial consistency such that any newly labeled point should have the same label as its neighbor from the stable clusters. We relabel any points with inconsistent labels as “uncertain” and we repeat the classification procedure until no points change class membership.

While numerous classifiers could be used, we chose a local Bayesian classifier. For any given time, we consider observations that are spatially nearby and also in the same cluster to follow a Normal distribution. Then, for each neighborhood, we train a Bayesian classifier. Its conditional probability for

any cluster C as

$$p(x|x \in C, t) \sim N(\mu(C_R, t), \sigma(C_R, t))$$

where $\mu(C_R, t)$ is the sample mean of all points belonging to stable cluster C and within region R (i.e., a spatial region that is centered around x) at time t and similarly $\sigma(C_R, t)$ is the sample standard deviation of all points belonging to clustering C and in region R .

The prior probability of $p(x \in C|t)$ is a function of the spatial distance between x and the cluster C . It is independent of t . Specifically,

$$p(x \in C|t) = e^{-\frac{\|x, C\|_{space}}{\delta_d^2}}$$

where $\|x, C\|_{space}$ is the spatial distance between the point x and cluster C . δ_d is a parameter that controls the weight of the spatial constraint. The larger δ_d is, the smaller the impact of the spatial distance on the prior probability. Thus for every uncertain point at the boundary of stable clusters, we would assign it to the cluster with the highest probability.

5 Experimental results

To test the performance of our approach, by autonomously extracting all in-land water bodies from 166 lake regions on a global scale (see Figure 8). We compared our performance against that of Normal-cut (an image segmentation method) [Shi and Malik, 2000] and Gao et al.’s method [Gao *et al.*, 2012] – a K-MEANS based approach used by the water resources management community. To compare the performance of the three methods, we use ground truth data from the Shuttle Radar Topography Mission’s (SRTM) Water Body Dataset (SWBD), which consists of a global water body map for February 2000. The SWBD data contains most water bodies for a large fraction of the Earth (60° S to 60° N) and is publicly available through the MODIS repository as the MOD44W product. For verification purposes, we compare each algorithm’s output (i.e. water pixels and land pixels) for the February 18th 2000 snapshot of TCWETNESS (the closest MODIS date near the time SWBD was collected) against the SWBD data



Figure 8: Positions of the 166 lake regions analyzed.

We evaluate the performance of the algorithms on each lake region independently using the F_1 score – measure that conveys the balance between a model’s precision and recall [Pang-Ning *et al.*, 2006]. To do so, we consider the water

locations as the positive set and the land locations as the negative set. The F_1 score of each model can be obtained by comparing an algorithm’s output and its corresponding validation set [Pang-Ning *et al.*, 2006].

Figure 9 shows the overall performance of each algorithm across all 166 lakes. Overall, our proposed method is more robust compared to NCUT and Gao *et al.*’s method as seen by the higher median F_1 score as well as a smaller inter-quartile range (the blue box).

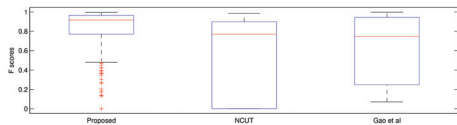


Figure 9: The performance of algorithms on the test 166 lakes.

To further understand our algorithm’s performance in the presence of noise and missing values, we repeated the same experiment except that we segmented the data into groups of increasing difficulty. In the first experiment we grouped the test data of February 2000 into the three groups based on the percentage of missing data in that snapshot. The three groups were: (i) regions that had less than 10% missing data; (ii) regions that had more than 10% but less than 60% missing data; and (iii) regions that had more than 60% missing data. Figure 10 shows the performance of each algorithm as a function of the percentage of missing data. We find that when the data are relatively complete (left panel of Figure 10) all three methods perform similarly. However, as the percentage of missing data increases, our proposed method outperforms the baseline methods. Note that in extreme cases where most of the data are missing NCUT breaks down completely, while our method is able to recover thanks to its reliance on information from multiple time-steps.

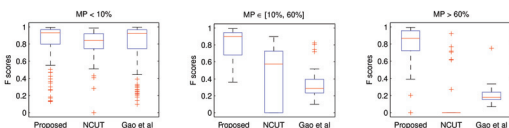


Figure 10: Performance of the three algorithms as a function of missing data.

In another experiment, we segmented the test data based on how noisy the data were. We considered the 107 lakes that had less than 10% missing data. We separated the 107 lakes into three groups based on their classification difficulty (e.g. how separable were the land and water pixels in the feature space). We used the Bhattacharyya coefficient (BC) [Comaniciu *et al.*, 2000] to measure how separable were the land and water TCWETNESS feature distributions. A BC measure of 0 means that the two distributions are completely separable. The three groups we evaluated were: (i) regions with a BC smaller than 0.05; (ii) regions with BC larger than 0.05 and smaller than 0.1; and (iii) regions with BC larger than 0.1 but smaller than 0.5. The performance of each algorithm on the different groups is shown in Figure 11. The results show that when water and land locations are high separable

in the feature space, all three methods perform equally well. However, when the features from a single time-step are not discriminative enough, using both spatial and temporal information is better than using only information from the feature space.

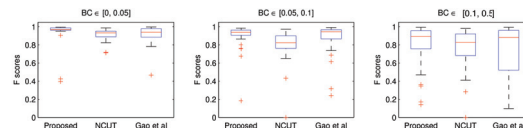


Figure 11: Performance of the three algorithms as a function of noise.

6 Conclusion and future work

In this paper, we introduced a new spatio-temporal paradigm to identify dynamic clusters from a continuous spatio-temporal field where data might be missing or noisy. The intuition behind this work is that although the cluster may move slightly from time-step to the next (and thus some points may change cluster membership), there are core points that never change clusters across the entire time. Our method used spatial contiguity and temporal similarity assumptions to overcome the limitations of non-space-time-aware methods especially in the presence of noise and missing values – two common characteristics of satellite products. We presented one implementation of the paradigm and expect numerous innovations on how to exactly carry it out. Our method can be used by domain scientists and sustainability experts to study the dynamics of in-land water availability and may potentially lead data driven resource management. One avenue of future work could be the automatic choice of the window size w . In our case we chose a time period of five years, but other more systematic ways to choosing the time window size should be explored.

Acknowledgments

This research was supported in part by the National Science Foundation Awards 1029711, 0905581, and 1464297. The NASA Award NNX12AP37G and an UMII MnDRIVE Fellowship. Access to computing facilities was provided by the University of Minnesota Supercomputing Institute.

References

- [Birant and Kut, 2007] Derya Birant and Alp Kut. St-dbscan: An algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007.
- [Chen *et al.*, 2013] Xi C Chen, Karsten Steinhaeuser, Shyam Boriah, Snigdhanu Chatterjee, and Vipin Kumar. Contextual time series change detection. In *SDM*, pages 503–511. SIAM, 2013.
- [Chen *et al.*, 2015] Xi Chen, Ankush Khandelwal, Sichao Shi, James Faghmous, Shyam Boriah, and Vipin Kumar. Unsupervised method for water surface extent monitoring using remote sensing data. *Machine Learning and Data Mining Approaches to Climate Science: Proceedings of*

- the Fourth International Workshop on Climate Informatics*, 2015.
- [Collins and Woodcock, 1996] John B. Collins and Curtis E. Woodcock. An assessment of several linear change detection techniques for mapping forest mortality using multi-temporal landsat {TM} data. *Remote Sensing of Environment*, 56(1):66 – 77, 1996.
- [Comaniciu *et al.*, 2000] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149. IEEE, 2000.
- [Coppin and Bauer, 1996] Pol R. Coppin and Marvin E. Bauer. Digital change detection in forest ecosystems with remote sensing imagery. *Remote Sensing Reviews*, 13(3-4):207–234, 1996.
- [Davidson *et al.*, 2013] Ian Davidson, Sean Gilpin, Owen Carmichael, and Peter Walker. Network discovery via constrained tensor analysis of fMRI data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 194–202. ACM, 2013.
- [Defays, 1977] Daniel Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, 1977.
- [Enright *et al.*, 2002] Anton J Enright, Stijn Van Dongen, and Christos A Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic acids research*, 30(7):1575–1584, 2002.
- [Ester *et al.*, 1996] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.
- [Faghmous and Kumar, 2013] James H Faghmous and Vipin Kumar. Spatio-temporal data mining for climate data: Advances, challenges, and opportunities. In W. Chu, editor, *Data Mining and Knowledge Discovery for Big Data: Methodologies, Challenges, and Opportunities*, pages 83–116. Springer, 2013.
- [Faghmous and Kumar, 2014] James H Faghmous and Vipin Kumar. A big data guide to understanding climate change: The case for theory-guided data science. *Big data*, 2(3):155–163, 2014.
- [Faghmous *et al.*, 2014] James H. Faghmous, Hung Nguyen, Matthew Le, and Vipin Kumar. Spatio-temporal consistency as a means to identify unlabeled objects in a continuous data field. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 410–416, 2014.
- [Gao *et al.*, 2012] Huilin Gao, Charon Birkett, and Dennis P. Lettenmaier. Global monitoring of large reservoir storage from satellite remote sensing. *Water Resources Research*, 48(9), 2012.
- [Jain and Dubes, 1988] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.
- [Latecki *et al.*, 2005] Longin Jan Latecki, Vasileios Megalooikonomou, Qiang Wang, Rolf Lakaemper, CA Ratanamahatana, and E Keogh. Partial elastic matching of time series. In *Data Mining, Fifth IEEE International Conference on*, pages 4–pp. IEEE, 2005.
- [Lee *et al.*, 2007] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM, 2007.
- [Li *et al.*, 2004] Yifan Li, Jiawei Han, and Jiong Yang. Clustering moving objects. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 617–622. ACM, 2004.
- [Li *et al.*, 2010] Zhenhui Li, Bolin Ding, Jiawei Han, and Roland Kays. Swarm: Mining relaxed temporal moving object clusters. *Proceedings of the VLDB Endowment*, 3(1-2):723–734, 2010.
- [MacQueen and others, 1967] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.
- [Massey Jr, 1951] Frank J Massey Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.
- [McLachlan and Krishnan, 2007] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.
- [Pang-Ning *et al.*, 2006] Tan Pang-Ning, Michael Steinbach, Vipin Kumar, et al. Introduction to data mining. *Pearson*, 2006.
- [Roddick and Spiliopoulou, 1999] John F Roddick and Myra Spiliopoulou. A bibliography of temporal, spatial and spatio-temporal data mining research. *ACM SIGKDD Explorations Newsletter*, 1(1):34–38, 1999.
- [Shi and Malik, 2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.
- [Sibson, 1973] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.
- [Steinbach *et al.*, 2003] Michael Steinbach, Pang-Ning Tan, Vipin Kumar, Steven Klooster, and Christopher Potter. Discovery of climate indices using clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 446–455. ACM, 2003.