Cluster-Centric Anomaly Detection and Characterization in Spatial Time Series

by

Hesam Izakian

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Anomaly detection in spatial time series is a challenging problem with numerous potential applications. A comprehensive anomaly detection approach not only should be able to detect and identify the emerging anomalies, but it also has to characterize the essence of these anomalies by visualizing the structures revealed within data in a way, which is understandable to the end-user. In this study, a cluster-centric framework for anomaly detection and characterization in spatial time series has been developed. For this purpose, the time series part of data is divided into a set of subsequences and the available spatio-temporal structures within the generated subsequences are discovered through a fuzzy clustering technique.

Since in spatial time series, each datum is composed of features dealing with the spatial and the temporal (one or more time series) components, clustering of data of this nature poses some significant challenges, especially in terms of a suitable treatment of different components of the data. We propose an extended version of the Fuzzy C-Means (FCM) clustering by introducing a composite distance function with adjustable weights (parameters) controlling the impact of different components in the clustering process. Three optimization criteria - a reconstruction error, a prediction error, and an agreement level are introduced and used as a vehicle to quantify the performance of the clustering method.

By comparing the revealed structures (clusters) in spatial time series in successive time intervals, one assigns an anomaly score to each cluster measuring the level of *unexpected* changes in data. Moreover, through developing some fuzzy relational

dependencies, the propagation of anomalies can be visualized in an understandable way to the end-user. To illustrate the proposed technique in this study, several datasets including synthetic and real-world data have been investigated. Experimental studies show that the proposed technique is able to find incident anomalies and quantify the propagation of anomalies over time.

Preface

Chapter 4 of this thesis has been published as H. Izakian, and W. Pedrycz, "Anomaly Detection in Time Series Data using a Fuzzy C-Means Clustering," *Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, Edmonton, Canada, pp. 1513-1518, 2013, IEEE Press. I was responsible for developing the idea, the data collection and analysis, and the manuscript composition. W. Pedrycz was the supervisory author and was involved with concept formation and manuscript composition.

Chapters 5 and 6 of this thesis have been published as H. Izakian, W. Pedrycz, and I. Jamal, "Clustering spatio-temporal data: An augmented fuzzy C-Means," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 5, pp. 855 - 868, 2013. I was responsible for developing the idea, the data collection and analysis, and the manuscript composition. W. Pedrycz was the supervisory author and was involved with concept formation and manuscript composition. I. Jamal also contributed in concept formation.

Chapter 7 of this thesis has been published as H. Izakian and W. Pedrycz, "Agreement-Based Fuzzy C-Means for Clustering Data with Blocks of Features," *Neurocomputing*, vol. 127, pp. 266-280, 2014. I was responsible for developing the idea, the data collection and analysis, and the manuscript composition. W. Pedrycz was the supervisory author and was involved with concept formation and manuscript composition.

Chapters 8 and 9 of this thesis have been published as H. Izakian and W. Pedrycz, "Anomaly Detection and Characterization in Spatial Time Series Data: A Cluster-Centric Approach," *IEEE Transactions on Fuzzy Systems*, DOI: 10.1109/TFUZZ.2014.2302456, 2014. I was responsible for developing the idea, the data collection and analysis, and the manuscript composition. W. Pedrycz was the supervisory author and was involved with concept formation and manuscript composition.

Acknowledgments

I would like to express my deepest respect and gratitude to my supervisor, Professor Witold Pedrycz, for his patience, innovations, enthusiasm, and supports during the four years Ph.D. study. It has been an honor to be his Ph.D. student.

I would like to acknowledge the financial support provided by the Alberta Innovates - Technology Futures and Alberta Advanced Education & Technology.

And finally, I would like to thank my wife and my parents for their support, encouragement, and patience.

Table of Contents

List of Tables

List of Figures

List of Symbols

$\boldsymbol{x}_k$	the kth data object
$\boldsymbol{x}_k(s)$	spatial part of $\boldsymbol{x}_k$
$\boldsymbol{x}_k(t)$	temporal part of $\boldsymbol{x}_k$
$\boldsymbol{x}_k(j)$	jth part of $\boldsymbol{x}_k$
$\hat{\boldsymbol{x}}_k$	reconstruction of kth data object
$\boldsymbol{v}_i$	the ith prototype
U	partition matrix
m	fuzzification coefficient
c	number of clusters
N	number of objects
n	length of a sequence
J	FCM objective function
$D[j]$	jth part of dataset (jth data source)
$d(.,.)$	distance function
σ	standard deviation
E	error
Q	performance index
z_k^t	kth particle in tth generation of PSO algorithm
y_k^t	kth velocity vector in tth generation of PSO
$pbest_k^t$	personal best corresponding to the particle z_k^t in PSO
$gbest^t$	global best of the population in tth generation of PSO
W_i	ith time window
$\boldsymbol{ux}_k$	membership degrees corresponding to $\boldsymbol{x}_k$
R	fuzzy relation
α	learning rate

1. Introduction

Spatial time series are commonly encountered in numerous application areas such as aerology, agriculture, and medical science. In this type of data, each datum is composed of two components namely a spatial part comprising location information (e.g., x-y coordinates or latitude-longitude pairs) and temporal part including one or more time series describing some temporal phenomena reported in successive time steps. Figure 1.1 shows the essence of spatial time series where, for each spatial x-y coordinate in the map, a time series is available.

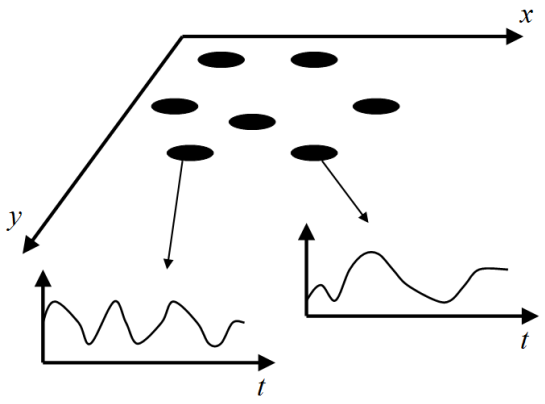


Figure 1.1. The essence of spatial time series

Anomaly detection in this type of data refers to detecting any *unexpected* changes in a subsequence of a set of spatially adjacent time series. This problem occurs in numerous application areas. For example, aerologists are interested to detect anomalies in climate patterns to predict future consequences; public health officers are interested in detecting anomalies of disease incidence to control possible future outbreaks, etc. A general framework for sequence-based anomaly detection involves using a sliding window to generate a set of subsequences and then determining those subsequences which exhibit the highest differences in comparison with the others as anomalies [1]. Three main scenarios can be distinguished here:

1) In the simplest case, there is only a single time series and the objective is to find a subsequence in this time series showing the highest difference from all other subsequences.
2) In the second case, there are a number of time series and the objective is to find a subsequence of a set of time series exhibiting high dissimilarity from other parts of data. In fact, in this problem we consider a set of time series at the same time to quantify dissimilarity.
3) Finally, the last case is concerned with spatial time series. Here we have to tackle another constraint present in the problem, which deals with the spatial position of time series when defining dissimilarity between subsequences. The objective is to find subsequences of a set of spatially neighboring time series, showing a high difference from the other parts of data.

Anomaly detection in spatial time series is a challenging problem since defining a spatial neighborhood of a set of time series encountering some unexpected changes is not straightforward. Moreover, the definition of unexpected changes may not be precise since we do not know what type of changes is expected and what type is not. Because of the problems identified above, using a brute force method to consider all possible states (spatially adjacent time series with subsequences encountering unexpected changes) is not efficient and for large size of data such approach is not feasible at all.

In this study, we develop a cluster-centric framework for anomaly detection and characterization in spatial time series. For this purpose, the time series part of data is divided into a set of subsequences and the available structures within the resulting spatio-temporal subsequences are discovered through a clustering technique. Clustering is an efficient instrument for visualizing and understanding the structure present within data. Fuzzy C-Means (FCM) proposed by Dunn [2] and Bezdek [3] is one of the most commonly-used clustering techniques in fuzzy set community in which, instead of assigning data to individual cluster, the Boolean-like nature of assignment is relaxed by admitting membership grades.

To cope with the specificity of the spatial time series, the generic objective

function of the FCM requires a thorough examination and revision of its formulation. For this purpose, an augmented version of Euclidean distance is employed in the FCM objective function. The augmented distance function is endowed with a substantial level of flexibility so that the contributions coming from the temporal and spatial parts of the data could be carefully balanced and optimized. The resulting flexibility is exploited to optimize three performance indexes, namely a reconstruction criterion, a prediction criterion, and an agreement criterion. To deal with the reconstruction criterion is essential when assessing the quality of clusters– information granules and quantifying their role being played in the processes of information granulation and de-granulation. The prediction aspects are of interest when forecasting a temporal component of the data given their specific location (spatial information). Agreement criterion is useful, when the objective is to reveal a general structure over all data sources having a high level of *agreement* among the available structures in separate data sources.

Clustering spatio-temporal subsequences in successive time steps, will lead to discovering a chain of structures within data and visualizing the dynamics available in spatial time series over time. One may quantify the level of unexpected changes within a part of data (in terms of an anomaly score) through comparing its structure with the revealed structures in the past (previous time steps). The comparison technique can be different for different natures of data and depends on the definition of anomaly from the user's point of view. Moreover, through the visualization of structure available within data in different time intervals, one may realize and quantify the propagation of anomalies over time. The proposed anomaly detection and characterization technique in this study is a user-friendly framework by strongly support the visualization of dynamics within data.

1.1. Research objectives and originality

The key objectives of this research are:

- Designing efficient clustering techniques to reveal and visualize the available structures– information granules within spatial univariate and spatial multivariate time series,
- Investigating the impact of different representation of time series on structures revealed through clustering techniques,
- Developing an efficient technique to assign anomaly scores to spatio-temporal clusters and quantifying the level of occurred unexpected changes in the structure of data,
- Constructing relations between clusters present in successive time windows to visualize and quantify the propagation of anomalies over time, and
- Providing a general framework for anomaly detection and characterization in spatial time series.

Detecting anomalies in spatial time series using spatio-temporal clustering is a novel idea studied in this research. The method introduced here, visualizes structures present in different time windows to make them understandable to the end-user. Moreover, using the fuzzy relation-based model of relationships, the revealed clusters in spatio-temporal subsequences can be tracked from the structures identified in the past, leading to a thorough temporal analysis of propagation of anomalies.

This research exhibits a significant level of originality:

- A new anomaly detection technique within time series data using a reconstruction criterion is developed.
- A new clustering technique for spatial univariate time series using reconstruction and prediction criteria is introduced.
- An agreement based fuzzy clustering approach for spatial multivariate time series is introduced.
- A new technique for assigning anomaly scores to the spatio-temporal clusters to quantify the level of unexpected changes is developed.

- A new fuzzy relation-based technique to visualize the propagation of anomalies in spatial time series during time evolution is introduced. And in overall:
- A general framework for anomaly detection and characterization in spatial time series has been developed.

1.2. Dissertation Organization

The subsequent chapters are structured as follows:

Chapter 2 A General Framework for Anomaly Detection and Characterization in Spatial time series

A general framework for anomaly detection and characterization is discussed. The framework comprises a number of components, each responsible for fulfilling one step in detecting and characterizing incident anomalies.

Chapter 3 Background and Literature Review

Some fundamentals about time series processing techniques including different representation methods and distance functions are discussed. Moreover, a number of techniques for clustering time series and spatio-temporal data, reported in the literature are reviewed. Finally, a number of anomaly and event detection approaches in the literature for time series and spatio-temporal data are reported.

Chapter 4 Anomaly Detection in Time Series Using a Fuzzy C-Means Clustering

In this chapter, a novel approach for anomaly detection in time series using a fuzzy C-Means clustering is proposed. Anomalies are divided into two categories: anomalies in amplitude and anomalies in shape. Then a general framework for detecting anomalies for both groups is introduced.

Chapter 5 Clustering Spatial Time Series Using a Reconstruction Criterion

An augmented version of fuzzy C-Means is introduced for clustering spatial time series. A composite distance function is employed and a reconstruction error is considered to control the impact of each part of data (spatial and temporal) in the clustering process.

Chapter 6 Clustering Spatial Time Series Using a Prediction Criterion

The same as the previous chapter, the proposed augmented fuzzy C-means technique is employed for clustering spatial time series. However, to find a sound balance between the effects of each part of data in the clustering process, a prediction criterion is considered.

Chapter 7 Clustering Spatial Time Series Using an Agreement Criterion

The proposed techniques in the previous chapters are suitable for clustering spatial univariate time series. In this chapter, the composite distance function is extended to admit different data sources (time series) for clustering. A Particle Swarm Optimization approach is employed to find a near-optimal impact of different data sources in the clustering process. This technique can be applied for clustering both spatial univariate and spatial multivariate time series.

Chapter 8 Anomaly Detection in Spatial Time Series

An anomaly detection method for spatial time series is proposed. It takes into account the historical behavior of data as well as the available structure (in form of clusters) inside the local data in various time intervals.

Chapter 9 Anomaly Characterization in Spatial Time Series

A gradient-based fuzzy relation is introduced to find the relationships between available structures within data in successive time intervals. This technique is able to quantify and visualize the propagation of anomalies over time.

Chapter 10 Conclusions and Future Works

We draw conclusions from our works in this chapter. Some directions for future works are suggested.

2. A General Framework for Anomaly Detection and Characterization in Spatial Time Series

As discussed in the previous chapter, anomaly detection and characterization in spatial time series is a challenging problem. To develop a general framework for this problem, one may split it into a number of blocks. Each block performs a set of processing over the data and sends the result to the next blocks. In this form, based on the nature of data and the application purpose, the end-user may interact with each separate block of the framework and choose some suitable parameters and methods.

2.1. Overall scheme of the proposed framework

Figure 2.1 shows the overall scheme of the proposed framework in this study.

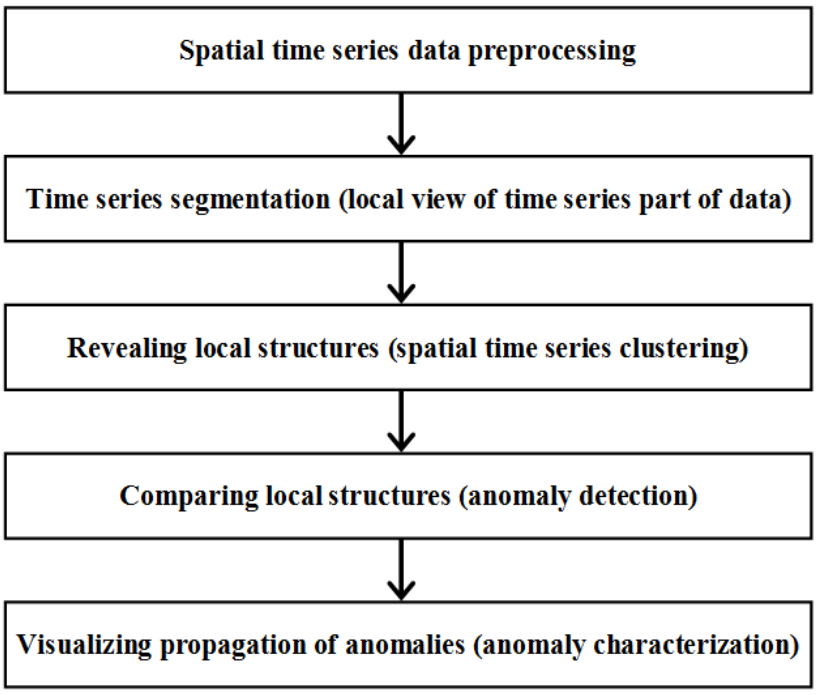


Figure 2.1. Overall scheme for the proposed framework for anomaly detection and characterization in spatial time series.

The blocks considered in this framework are as follows:

Spatial time series preprocessing: This block is composed of a set of preliminary processing to prepare the data for the next steps. If the spatial part of data is expressed using latitude/longitude pairs or postal codes, one may map them in this step to x-y coordinates to be used in Euclidean space. Another task that can be considered here is representing the time series part of data using an efficient technique to reduce the length of time series and decrease the impact of noisy data. The representation method of time series can be selected by the end-user based on the application purpose.

Time series part segmentation: This block provides a local view of time series part of data. For this purpose, a sliding window with a predefined length can be used. The sliding window moves thorough the time series part of data and generates a set of subsequences. Considering the spatial part of data along with the generated temporal subsequences, a set of spatio-temporal subsequences can be constructed. The length of the time window and its movement in each step can be selected by the end-user and depends on the nature of time series and the application purpose.

Revealing local structures: Considering the generated spatio-temporal subsequences in the previous block, this block discovers and visualizes the available local structure inside each set of spatio-temporal subsequences. For this purpose, in this research an augmented fuzzy C-Means technique has been considered and three criteria namely, a reconstruction, a prediction and an agreement are proposed. The first two criteria are used for clustering spatial univariate time series, while the third one is applicable for clustering univariate and multivariate spatial time series. The discovered structures are in form of a set of partition matrices describing the membership degrees of data points to cluster centers. The end-user in this block may select a number of parameters comprising the fuzzification coefficient, a suitable number of clusters for the generated

spatio-temporal subsequences, etc. Moreover, an appropriate criterion for spatio-temporal clustering should be selected by the end-user in this block.

Comparing local structures: This block is responsible for comparing the revealed local structures (clusters) provided in the previous block. In fact, in this block we try to find any unexpected changes in the structure of data. For this purpose, one may compare the local structures revealed in different local parts of data with their historical behavior and assign an anomaly score quantifying the level of unexpected changes. Since each cluster includes a set of spatio-temporal subsequences, one may assign an anomaly score to each single spatio-temporal subsequence and the estimated anomaly scores can be aggregated inside each cluster. Selecting a suitable method to assign an anomaly score to each single subsequence can be based on the nature of a data and the meaning of unexpected changes from the user's point of view.

Visualizing propagation of anomalies: Although finding the anomalous parts of data and quantifying the level of their unexpected changes is important, however, characterizing the incident anomalies and visualizing their propagation over time is equally important in many applications. In this research, a gradient-based fuzzy relation technique is employed for mapping local clusters in successive time steps, and visualizing the dynamics available in data in an understandable way to the end-user.

2.2. Summary

In this chapter, we briefly described the overall scheme of the proposed framework for anomaly detection and characterization in spatial time series. The framework is composed of a number of separate blocks, each responsible for performing a set of sub-tasks. Using this structure is beneficial and the end-user is able to understand and interact with the system in all steps of anomaly detection and characterization process. In other words, the end-user may try different

parameters and methods in each block, based on the application purpose and the nature of data to achieve some appropriate results.

3. Background and Literature Review

In this chapter, some fundamentals about time series processing techniques comprising different representation methods and distance functions are discussed. Then, a number of techniques for clustering time series and spatio-temporal data proposed in the literature have been reviewed. Next, a number of anomaly and event detection approaches proposed in the literature for time series and spatio-temporal data are reported.

3.1. Representation methods and distance functions in time series

There are a number of methods proposed in the literature to represent time series. In general, such representation methods are categorized into data-adaptive and non-data-adaptive techniques [4–7]. Adaptive piecewise constant approximation [5], piecewise linear approximation [8], singular value decomposition [9] and symbolic aggregate approximation [4] are examples of data-adaptive methods. Discrete Fourier transform [10], Chebyshev polynomials [11], discrete wavelet transform [12, 13] and piecewise aggregate approximation [14] are well-known methods belonging to the second category.

In this chapter, we describe three commonly studied methods to represent time series, namely Discrete Fourier Transform (DFT), Piecewise Aggregate Approximation (PAA), and Discrete Wavelet Transform (DWT). They can be viewed as sound representatives of the large set of the methods existing in the literature. In what follows, we review them briefly.

Discrete Fourier transform: The discrete Fourier transform models the time series using a set of sine and cosine waves. It represents the time series in a frequency domain. For a time series $\boldsymbol{x}$ of length n, DFT is composed of n complex numbers, each describing a sine/cosine wave given by

$$f_k = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} x_i \exp(-j2\pi ki/n) \quad k = 0, 1, \cdots, n-1, \tag{3.1}$$

where $j = \sqrt{-1}$. The original time series can be reconstructed by running an inverse transform given by

$$x_i = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} f_k \exp(j2\pi ki/n) \quad i = 0, 1, \cdots, n-1. \tag{3.2}$$

Faloutsos et al. [10] employed DFT for indexing time series. They noted that the most important features of each sequence are the first k (real and imaginary) coefficients ($k << n$) of the DFT transform, while the other coefficients assume values close to zero. By having these k coefficients, the original time series can be reconstructed with a little loss of information.

Piecewise aggregate approximation: This method provides a simple and efficient way of time series representation in time domain offering a substantial dimensionality reduction [14]. PAA divides the time series $\boldsymbol{x}$ with length n into k ($k << n$) segments of equal length and determines the mean value of data points lying within each segment as the representatives of the original time series. More formally, we have the representation in the form of a vector $\boldsymbol{f}$ whose coordinates are expressed as follows

$$f_i = \frac{k}{n} \sum_{j=\frac{n}{k}i}^{\frac{n}{k}(i+1)-1} x_j, \quad i = 0, 1, \cdots, k-1. \tag{3.3}$$

Discrete wavelet Transform: Wavelets are basis functions that describe time series in a time-frequency joint representation. In [12] and [13], DWT is used as an efficient representation method to index time series. A well-known method to calculate the DWT coefficients is a pyramid algorithm [15]. In this method, the length of time series, n, has to be a power of two. For time series that do not satisfy this condition, zero padding is realized. DWT converts the time series into two types of coefficients resulting from low pass filters (also called scaling function) and high pass filters (also called wavelet function) each in length $n/2$

given by

$$a_i = \frac{1}{2}\sum_{j=0}^{n-1} c_{2i-j+1}\, x_j, \quad i = 0, 1, \cdots, \frac{n}{2} - 1, \tag{3.4}$$

$$f_i = \frac{1}{2}\sum_{j=0}^{n-1} (-1)^j c_{j-2i}\, x_j, \quad i = 0, 1, \cdots, \frac{n}{2} - 1, \tag{3.5}$$

where $\boldsymbol{a} = [a_0, a_1, \cdots, a_{n/2-1}]^{\mathrm{T}}$ are scaling coefficients and $\boldsymbol{f} = [f_0, f_1, \cdots, f_{n/2-1}]^{\mathrm{T}}$ are wavelet coefficients present at the first level. To calculate the wavelet coefficients at the next level, the above calculations are performed over the scaling coefficients $\boldsymbol{a}$. The procedure is repeated until the required number of iteration has been reached. For each wavelet function there are a number of non-zero coefficients. For example, for the Haar wavelet, the non-zero coefficients are $c_0 = c_1 = 1$.

One has to stress that the representation method of time series is problem-dependent. For example, one may be interested to analyze time series based on their frequency characteristics (using DFT), time characteristics (where PAA could be of interest), or time-frequency joint characteristics (DWT).

Distance functions (distances, for brief) used in time series can be divided into three general categories: $\mathrm{L_p}-$norm distances, elastic measures, and statistical measures. Euclidean distance ($\mathrm{L_2}$) has been widely used as a dissimilarity measure [7] and is suitable to compare equal-length time series. Dynamic time warping distance [16] is an elastic measure used to determine an optimal match between two time series by stretching or compressing their segments, and concentrates on the similarity of time series with respect to their shapes. Longest common subsequence [17] is another example of the elastic-based distance measures. This method uses the length of the longest subsequence occurring in two time series to quantify their similarity. Edit distance of real-number sequences [18] which is another elastic-based distance measure, considers the number of insert, delete and replace operations that are required to convert one sequence to another for expressing the similarity. Pearson coefficient is a

statistical method, used to quantify the correlation between two time series. The Kullback-Liebler distance [19] is another statistical measure useful in expressing the dissimilarity between two time series represented by their Markov chain. A comparison between a number of representation methods and similarity measures used for various types of time series was reported in [7] in the problem of indexing time series. The suitability of each similarity measure is application-oriented. Nevertheless the Euclidean distance is in common usage.

3.2. Clustering time series

Time series have been investigated in a variety of problems of data mining such as clustering [21–23, 25, 31, 98], classification [99, 100], forecasting [101–103], and modeling [104, 126–130]. Based on the type of data being used, the time series clustering methods can be split into three categories [20, 21], namely those using raw time series [22–25], model-based methods [19, 26, 27], and representation-based methods [20, 28–31].

Methods using raw time series: Golay et al. [22] proposed two cross correlation based similarity measures of raw functional MRI data in order to provide functional maps of human brain activity using the Fuzzy C-Means method. The effect of different preprocessing methods and different number of clusters on the clustering performance was discussed. By representing time series through piecewise linear functions, Möller-Levet et al. [23] proposed a short time series distance determined as the sum of squared distances between the corresponding slopes encountered in two time series. The clustering was realized with the use of the FCM. In [24] a one-nearest neighbor network, based on dynamic time warping distance is built, where each node represents a certain time series and each link denotes neighbor relationship between nodes. In the next step, the time series of higher degrees (terms of the graph notation) are subject to clustering. The method can reduce the size of data and exhibits good performance in terms of efficiency and effectiveness. In [25], authors adopted a dynamic time warping distance for

the K-Means clustering by defining an averaging method (referred to as a DTW barycenter averaging). First, a global averaging method was proposed for time series, and then a strategy to reduce the length of the resulting average has been employed to improve the performance.

Model-based methods: Ramoni et al. [19] developed a Bayesian method to cluster time series. This method models the time series as Markov chains and uses the symmetric Kullback-Liebler distance between transition matrices as the similarity measure. The task of clustering was viewed as a Bayesian model selection problem to find the most suitable set of clusters. An entropy-based heuristic search strategy was used to improve efficiency. In [26] Kalpakis et al. first modeled the time series as an autoregressive model and then used LPC cepstral coefficients to capture the important features of the model. They used a partition around medoids clustering [32] to cluster the time series and showed the efficiency of the feature extraction method. In [27] time series are modeled using ARMA processes [33] and an expectation-maximization algorithm was used for clustering. Moreover, Bayesian information criterion [34] was used to determine the number of clusters in data.

Representation-based methods: A Haar wavelet based anytime K-Means clustering was proposed in [31]. This method exploits the multi-resolution property of wavelets. In the first step, an initial clustering was performed with a very coarse resolution of the data (the first level of representation). The results were used to initialize clustering at a slightly finer level of approximation. This process was repeated until the clustering results stabilize or until the wavelet representation was the same as the raw data (the last level of representation). This method was faster than the original K-Means and the quality of the clustering was often better. In [28] the variances of time series through their wavelet decomposition are used as the similarity measure and the FCM method considered as the clustering mechanism. The authors showed that this method will distinguish between time series with patterns of different variability as well

as time series with switching patterns. D'Urso and Maharaj [29] used the autocorrelation of time series as a representation technique, and then the FCM algorithm has been adopted to cluster time series in new feature space. Also in [30] they proposed using the estimated cepstrum of time series as robust and efficient features in fuzzy clustering. In [20] Yang and Chen developed an unsupervised ensemble learning model for time series clustering by combining rival-penalized competitive learning (RPCL) networks with different representations of time series including piecewise local statistics, piecewise discrete wavelet transform, polynomial curve fitting, and discrete Fourier transform. Moreover, the authors stressed that the joint usage of different representations becomes beneficial to improve the quality of results. A comprehensive survey of different methods of time series clustering is reported in [21].

3.3. Clustering spatio-temporal data

In real world applications we encounter with different kinds of spatio-temporal data. Kisilevich et al. [35] divided spatio-temporal data into five categories including spatio-temporal events, geo-referenced variables, geo-referenced time series, moving objects, and trajectories.

In spatio-temporal event data, there is a set of events, each occurred in a spatial location and coming with its timestamp. Clustering this type of data aims at finding a set of events that are close to each other in both space and time. One of the commonly used methods for clustering this type of data is scan statistics [36] [37]. In this method, one moves a cylindrical window of variable size and shape, across a geographical region to detect clusters of events with the highest likelihood ratios. In [38], an extended version of FCM has been proposed to find circular clusters of hotspots in spatio-temporal GIS data. For each timestamp, the events are clustered based on their spatial location and then a comparison between occurred clusters in successive time stamps has been performed to conclude some interpretations about events. Wang et al. [39] proposed two spatio-temporal

clustering methods called ST-GRID and ST-DBSCAN to detect seismic events in China and neighboring countries. The ST-GRID method used a multi-dimensional grid that covers the entire spatio-temporal feature space. Then, by merging the dense neighbor cells, spatio-temporal clusters were formed. ST-BDSCAN extended DBSCAN [40] by redefining density reachability using spatial and temporal radius. Both methods exploited an ordered k-dist graph [40] to determine their parameters.

Geo-referenced time series are composed of a set of fixed geographical coordinates, each corresponding to one or more time series. Geo-referenced variables data form a special case of geo-referenced time series where only the most recent point of time series is available. Clustering this type of data aims at grouping objects based on their spatial closeness and temporal similarities. In [41], FCM has been used to cluster weather time series. The Pearson correlation coefficient was employed as the similarity measure expressing closeness of two time series and a method to determine the number of clusters has been proposed. However, the method does not involve the spatial part of data in the clustering process. Deng et al. [42] proposed a density based spatio-temporal clustering. In this method, a spatial proximate network has been constructed using Delaunay triangulation and a spatio-temporal autocorrelation analysis was employed to define the spatio-temporal neighborhood. In [43], an extended version of FCM was proposed for image segmentation by considering the spatial location of pixels. This method has been considered by Coppi et al. [44] for clustering spatio-temporal data. In this approach, a spatial penalty term that was calculated using a spatial contiguity matrix has been added to the objective function to guarantee an approximate spatial homogeneity of the clusters.

Trajectories capture the movement behavior of a set of spatial objects in the form of time series. When only the most recent position of the objects is available, the data are called moving objects data. Clustering of this kind of data, aims at discovering a behavior of a collection of objects e.g., those occurring in urban traffic or animals' migration. In [45], the Euclidean distance between trajectories was used as a dissimilarity measure whereas OPTICS [46] has been extended to

cluster trajectories. Two methods, Trajectory-OPTICS and a time-focused version of that (called TF-OPTICS) were proposed. In [47], a probabilistic regression model for trajectory detection was proposed and expected maximization algorithm [48] was employed to model trajectories. Kalnis et al. [49] proposed algorithms to discover moving clusters in spatio-temporal data. In these methods, the set of objects of a moving cluster change over time. At each time step, the location of objects has been considered as a snapshot and a spatial clustering method like DBSCAN was used for clustering. Two snapshot clusters in consecutive time steps were considered as moving clusters if a value of their Jaccard coefficient exceeds a certain threshold. A fuzzy clustering for three-way data was proposed in [50]. In this structure, each data point was composed of objects, attributes and situations. The data are clustered based on not only individual time instances, but also the similarity between structures has been considered in different time steps. A survey of clustering spatio-temporal data is reported in [35].

3.4. Objects with blocks of features

Clustering spatial time series can be considered as clustering data with blocks of features coming from distinct data sources. In this point of view, each part of data including spatial part and each time series part (especially when dealing with multivariate time series) construct a block of features coming from a distinct data source.

Clustering objects with blocks of features originating from distinct sources or different data sites has been considered in number of studies coming usually under the name of collaborative clustering [51–55] and consensus-based clustering [56–68].

In collaborative clustering there are some communications between different data sources, and the algorithm looks for structure in each source by considering some hints coming from some other sources. These hints take on a format of partition matrices [53], prototypes [55] or proximity matrices [51, 52]. On the other hand,

in consensus-based clustering techniques usually the available information about the existing structure in data sources is collected in the form of cluster labels or partition matrices, and a new feature space (or similarity measure) is constructed using these guidance mechanisms. Subsequently the algorithm re-clusters the data using the new feature space.

Strehl and Ghosh [56] proposed normalized mutual information to evaluate the shared information among initial clusters. Three heuristics, namely Cluster-based Similarity Partitioning Algorithm (CSPA), HyperGraph Partitioning Algorithm (HGPA), and Meta-CLustering Algorithm (MCLA) were proposed to form consensus with a high level of shared information. As the initial clusters in these methods were crisp clusters, authors in [58] extended the above heuristics to deal with fuzzy clusters as initial clusters for building consensus. In [60], authors modeled the initial clusters coming from different data sources using a bipartite graph. A graph partitioning method was used to form final consensus. In [61], initial clusters of different data sources were viewed as independent sources of evidence of structures in data, and a voting mechanism was considered to generate a similarity matrix among objects. Finally, the objects were clustered using a hierarchical agglomerative clustering algorithm by considering the new similarity measure.

Ayad and Kamel [62] proposed a cumulative voting algorithm for different number of clusters to build computationally efficient consensus. In this method, a probabilistic mapping was introduced for cluster label alignment. In [65] a voting mechanism has been formulated as a multi-response regression problem to form consensus from an aggregated ensemble representation. In [64] authors proposed two fast and efficient centroid-based ensemble merging algorithms that combine partitions of data and are scalable to large datasets. In [66] a partition relevance analysis was considered to estimate the significance of partition matrices before combining them, and a new similarity measure between partition matrices was proposed.

In [59] a consensus-driven fuzzy clustering was proposed. In this method, some proximity matrices were constructed using partition matrices from different data

sites. The objective of the algorithm was to from a consensus over a data site to preserve its original structure, while minimize the distance of its corresponding proximity matrix from the other proximity matrices available in other data sites. A gradient-based method was used to realize optimization and form the final consensus results. Pedrycz [69] proposed a method to cluster semantically distinct families of variables. In this method, a prediction criterion has been used to optimize the effect of variables in the clustering process.

3.5. Anomaly detection in time series data

Most studies reported in the literature deal with anomaly detection in time series and do not consider spatio-temporal data. These methods can be divided into a set of categories comprising similarity-based, clustering-based, classification-based, and modeling-based techniques.

Similarity-based techniques: One straightforward method for anomaly detection in time series is to assign an anomaly score to each time series according to its similarity to the other time series existing in dataset. A suitable distance function or resemblance measure can be considered as a similarity/dissimilarity measure. In [70], an anomaly detection technique has been proposed for light curves in catalogues of periodic variable stars. By considering N time series $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$ present in the dataset, the anomaly score of a certain time series $\boldsymbol{x}_i$ was expressed as

$$\Re^2_{\boldsymbol{x}_i} = \frac{1}{N-1} \sum_{\substack{j=1,2,...,N \\ j \neq i}} r^2(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{3.6}$$

where $r(\boldsymbol{x}_i, \boldsymbol{x}_j)$ was the cross correlation present between time series $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. A lower value of the score of the cross correlation corresponds to a higher level of anomaly of the time series. Keogh et al. [1, 94] proposed detecting the maximal different subsequence within a longer time series (called discords) using a 1-nearest neighbor (1-NN) technique. Formally, by considering a time series of

length k, $t_1, t_2, ..., t_k$, a discord with length n, ($n < k$) was a subsequence $t_p, t_{p+1}, ..., t_{p+n-1}$ for $1 \le p \le k - n + 1$ with highest distance to its non-overlapping nearest neighbor. By representing time series using a symbolic aggregate approximation, authors proposed an algorithm that was faster then the brute force method. In [95], the distance of each time series to its *k*th nearest neighbor was proposed as anomaly score. In [71], a compression-based dissimilarity measure was proposed for anomaly detection in time series. For two sequences $\boldsymbol{x}$ and $\boldsymbol{y}$, the dissimilarity measure was expressed as

$$CDM(\boldsymbol{x}, \boldsymbol{y}) = \frac{C(\boldsymbol{xy})}{C(\boldsymbol{x}) + C(\boldsymbol{y})}, \tag{3.7}$$

where, $C(\boldsymbol{x})$ was size (given in bytes) of compressed file containing time series $\boldsymbol{x}$, and $C(\boldsymbol{xy})$ was size of compressed file containing concatenated sequences $\boldsymbol{x}$ and $\boldsymbol{y}$. Considering $\boldsymbol{x}$ as a subsequence of a longer time series $\boldsymbol{y}$, CDM ($\boldsymbol{x}$, $\boldsymbol{y}$) was considered as anomaly score of $\boldsymbol{x}$.

In [72], authors proposed a method to detect outliers in spatial and temporal data. Dimensionality reduction was performed during the preprocessing step using a L_2-norm and a global outlier was estimated for each separate location and separate time stamp. For this purpose, a distance-based outlier detection approach has been considered for temporal part of data and a neighborhood-based outlier detection method was exploited for spatial part of data. Moreover, spatially anomalous units encountering a high deviation from the historical trends are considered as spatio-temporal outliers.

Clustering-based techniques: Clustering is another method used for anomaly detection in time series. In this method, time series are clustered using an appropriate clustering technique and the revealed cluster centers are exploited to assign an anomaly score to each time series. In [73], a Fuzzy C-Means clustering was used to cluster a set of time series and a reconstruction criterion [74] was employed to reconstruct time series with the aid of the revealed cluster centers. Finally, a reconstruction error was used to assign an anomaly score to each time series. In [75], a set of training sequences was clustered using a *k*-medoids

clustering, and for each test sequence its inverse similarity to its closest medoid was considered as the anomaly score. Formally, an anomaly score, $\text{Score}(\boldsymbol{x}_k)$, was assigned to each test sequence $\boldsymbol{x}_k$ as

$$\text{Score}(\boldsymbol{x}_k) = \left(\min_{\forall i} (S(\boldsymbol{x}_k, \boldsymbol{v}_i)) \right)^{-1}, \tag{3.8}$$

where $\boldsymbol{v}_i$ was *i*th medoid and *S*(.) denotes a similarity measure computed for two sequences.

Classification-based techniques: Classification techniques also are of interest for anomaly detection in time series. A common method in this category is to train a classifier using a set of training normal time series and then use the classifier to assign an anomaly score to each test time series. In [76], the time series are projected onto a phase space and then novel events in time series are interpreted as outliers of normal distribution of vectors in the phase space. A single-class support vector machine was employed as the outlier detector. Dasgupta and Forrest [77] proposed an anomaly detection inspired by the negative selection mechanism of the immune system. Normal data were considered as "self" and anomalies were considered as "non-self" patterns. Moreover, Gao et al. [78] proposed using a neural network for event extraction in time series. They showed that neural network could characterize the properties of homeostatic dynamics and model the dynamic relation between endogenous and exogenous variables in financial time series.

Modeling-based techniques: Time series modeling techniques form another group of anomaly detection approaches reported in the literature. Autoregressive (AR) model [33] is one of the commonly used techniques for this purpose. AR model assumes that a value of the time series in time *t*, x_t, can be approximated using the values of its *p* values present in the previous time instants. Formally, we have

$$x_t = \sum_{i=1}^{p} q_i x_{t-i} + \varepsilon_t, \tag{3.9}$$

where p is the order of the model, q_i, i=1,2,…,p are its parameters, and ε_t is white noise. Takeuchi and Yamanishi [79] proposed a two-stage time series learning model to detect change points in time series. Considering $\{x_t \mid t = 1,2,...\}$ as the input time series, at the first step of the algorithm a sequence of probability density functions $\{p_t \mid t = 1,2,...\}$ was constructed using an AR model and for each point x_t in time series a logarithmic loss score was calculated as $\text{Score}(x_t) = -\log p_{t-1}(x_t \mid x^{t-1})$. A higher score for x_t indicates that this point is an outlier with a higher likelihood. In the next step, using a sliding window a new time series was constructed as an average of the calculated scores obtained in the previous step. The new time series was fitted again using an AR model and new loss scores were calculated for the new time series. Higher values of the scores for the points in the new time series indicate that they are change points with a higher probability. In [80] a self-organizing map (SOM) was employed to characterize the time evolution in AR processes. The regions of the map that AR process was expected to move were identified and the anomalous changes of AR process were detected. The method was applied to a real-world industrial process. In [81], multivariate time series are modeled using a weighted graph representation, where each node of the graph corresponds to a data point or a subsequence in a time series and each edge was weighted through a similarity measure between nodes. Considering p being the number of variables in multivariate time series, the similarity between time stamps i and j in time series was calculated with the aid of a Radial Basis Function (RBF) as follows:

$$K(i,j) = \exp\left[-\frac{\sum_{k=1}^{p}(x_{ik} - x_{jk})^2}{\sigma^2}\right]. \tag{3.10}$$

To calculate the connectivity of each node, the constructed graph was considered as a Markov chain with a transition matrix S, where the element in ith row and jth column denotes the transition probability from node i to node j. The transition matrix was normalized and the connectivity value of each node was calculated and nodes with a low value of connectivity were considered as anomalies. Khatkhate et al. [82] proposed modeling time series through a hidden Markov

model (called D-Markov machine model) from a symbolic representation. For each time epoch t_k, an anomaly measure was defined as

$$\hat{M}(t_k) = \sum_{l=1}^{k} d(\boldsymbol{p}^l, \boldsymbol{p}^{l-1}), \tag{3.11}$$

where $d(.,.)$ was a distance function, while probability vectors $\boldsymbol{p}^1, \boldsymbol{p}^2, \ldots$ are obtained at epochs $t_1, t_2, \ldots$ based on the respective time series. In [83] a dynamic Bayesian network was employed to develop two automated anomaly detection techniques. These methods can be applied to single sensor data streams (called uncoupled detection) as well as several data streams at once (called coupled anomaly detection). The efficiency of the proposed methods was investigated for two wind speed data streams to perform a data quality assurance and control. Dereszynski and Dietterich [84] proposed a real-time data quality control in sensor networks. This method models the spatial relationships among sensors using a Bayesian network. To exploit the temporal correlations, the model was extended to a dynamic Bayesian network. It was able to detect failure observations and predict their true values. In [85] an expectation-based scan statistics [86] was proposed in order to monitor a set of spatially located time series for detecting emerging spatial patterns. For this purpose, expected number of events was calculated and a set of spatial regions containing significantly high number of events was detected. Moreover, authors in [97] proposed an entropy based data analysis for detecting anomalies in complex aerospace systems. A survey of anomaly detection approaches for time series and point data can be found in [87], and [96], respectively.

3.6. Event detection in spatio-temporal data

A category of anomaly detection techniques in spatio-temporal data (mainly for disease anomalies) are reported in the literature as event (outbreak) detection. These techniques are mainly divided into two categories, namely statistical methods and model based methods [114]. In statistical methods, by comparing the number of disease incidence in some selected regions and the number of disease

cases in the whole area of the map, the algorithm tries to find any abnormal behavior in the system. Knox test [115], scan statistics [36] and cumulative sum methods [116] are some popular statistical approaches. On the other hand, in model based techniques, the algorithm based on some disease-related variables (e.g., climate situation etc.) tries to model the number of expected disease cases in the system and then by comparing it with the number of disease incidence makes a decision about occurring an abnormal situation. Generalized linear mixed model [117] and Bayesian modeling [118] are some well-known techniques in this category.

Knox proposed a test to detect any unusual space-time interaction of disease incidence. This technique checks whether the number of disease cases in a specific space and time interval are higher than the number of expected (usual) disease incidence. Aldstadt [119] employed a modified version of Knox test to detect infectious disease outbreaks. Scan statistics proposed in [36] checks different space-time intervals over a map by moving a cylindrical window with different shape and size, where the cylinder radius defines the spatial search area and cylinder height defines the temporal search area. The method tries to find areas with number of cases that are statistically significant. This method has been used in [120] for analyzing West Nile virus in New York state. Cumulative sum methods monitor sequential observations of a variable (e.g., number of disease cases in a region) and cumulate the deviations of the variable from the expected mean, and if this cumulating exceeds a pre-specified threshold, an event will be announced. In [121] Sonesson used a version of Cumulative sum for Tularemia disease incidence in Sweden.

Generalized linear mixed model uses regression frameworks to model disease counts using exponential statistical distributions. In this method, the number of expected disease cases is estimated using some disease-related variables, and if the difference between the number of disease incidence and number of expected disease cases is more than a threshold, an event will be announced. Kosmider [122] used a generalized linear mixed model to detect Salmonella outbreaks in British livestock. Bayesian networks were used extensively in event detection

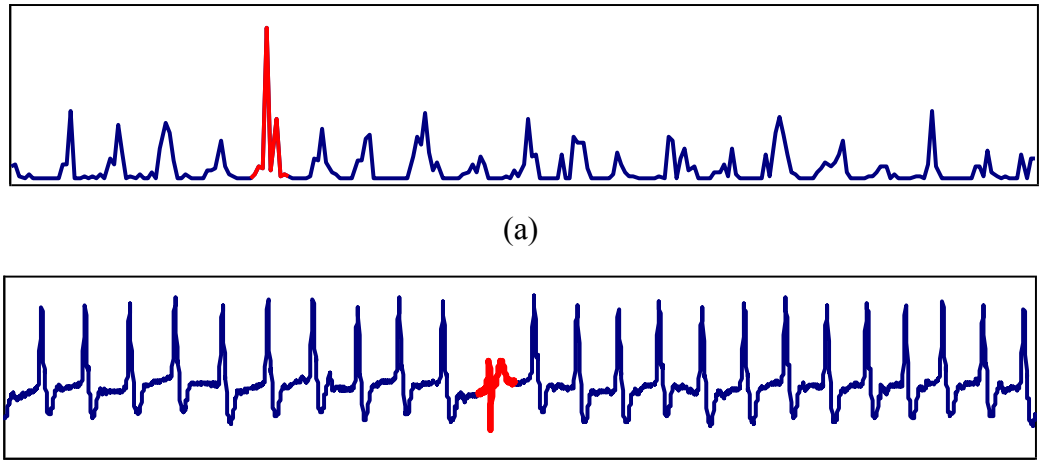
problem. In this technique, based on the relations between different variables and disease cases, a Bayesian network can be established, and using Bayesian rules the probability of incidence of disease cases can be estimated. If the number of disease cases is more than the expected number of disease cases, an alarm will occur as an event. In [123] Neill et al. used a Bayesian network scan statistics for event detection problem.

4. Anomaly Detection in Time Series Using a Fuzzy C-Means Clustering

One of the preliminary steps in detecting and characterizing anomalies in spatial time series is developing some techniques to detect anomalies in time series part of data. There are a number of techniques proposed in the literature for this purpose and some of them are reviewed in Chapter 3. Selecting a suitable technique for anomaly detection in time series depends on the nature of time series, the application purpose, and the definition of anomaly from the user‘s point of view.

In time series, anomaly can be considered as the occurrence of any unexpected changes in a subsequence of data. The term “unexpected change” makes sense when we compare the available pattern in a subsequence with the existing patterns in the entire time series. As the result, one common approach for anomaly detection in time series is the use of a fixed length sliding window and generating a set of subsequences of time series. In the next step, one may use different techniques to detect and characterize anomalies i.e. assigning an anomaly score to each subsequence.

Anomalies occurring in time series can be a result of a change in the amplitude of data (e.g., a heavy rainfall in a week of a year), or it may be a change in the shape (e.g., occurring an arrhythmia within a set of normal heartbeats in ECG signals). In this chapter, we categorize anomalies into two types: anomalies in shape and anomalies in amplitude. Figure 4.1(a) coming from [124] shows an anomaly in amplitude of precipitation time series belonging to one of climate stations in The United States, and Figure 4.1(b) coming from [125] shows an anomaly in shape within an ECG signal. The anomalous parts are highlighted in both figures.



(a)

(b)

Figure 4.1. The essence of anomalies in time series. (a) Anomaly in amplitude, and (b) anomaly in shape.

In this chapter, we propose a unified framework to detect both types of anomalies. For this purpose, after generating a set of subsequences of time series using a sliding window, a fuzzy C-Means clustering has been employed to reveal the available structure within data. Then, a reconstruction criterion [74], is considered to reconstruct the original subsequences from the determined cluster centers (prototypes) and partition matrix. For each subsequence, an anomaly score has been assigned based on the difference between the original subsequence and its reconstructed version. In the case of anomalies in amplitude, the original representation of time series along with the Euclidean distance function is used in the clustering process, while for shape anomalies, first a representation of subsequences is considered to capture the shape information and then, the Euclidean distance in the new feature space has been employed.

The idea of assigning an anomaly score to each subsequence based on its quality of reconstruction from revealed information granules– clusters is novel and promising. Moreover, providing a uniform framework to detect different types of anomalies, namely amplitude and shape anomalies is beneficial.

4.1. Problem formulation

Let us consider a time series $\boldsymbol{x} = x_1, x_2, ..., x_p$ of length p. We aim at finding a set of subsequences of $\boldsymbol{x}$ with length q, having highest amount of unexpected changes (in shape or amplitude) in terms of anomaly score. For this purpose, a sliding window with length q moves thorough the time series and generates a set of subsequences. Consequently, there will be N subsequences coming in the form

$$\begin{aligned} &\boldsymbol{x}_1 = x_{11}, x_{12}, ..., x_{1q} \\ &\boldsymbol{x}_2 = x_{21}, x_{22}, ..., x_{2q} \\ &\vdots \\ &\boldsymbol{x}_N = x_{N1}, x_{N2}, ..., x_{Nq} \end{aligned} \quad . \tag{4.1}$$

Note that in each movement, the sliding window moves r time steps. As the result, the number of subsequences, N is

$$N = \frac{p-q}{r} + 1 \, . \tag{4.2}$$

Considering a low value for r guarantees that no anomalous subsequences are missed, but processing a high amount of subsequences is time consuming. On the other hand, considering a high value for r (e.g., $r = q$) generates lower number of subsequences and processing time will be lower, but there is a risk of losing some anomalous subsequences. A trade-off between accuracy and processing time can be considered. Selecting the value of r being proportional to the length of subsequences is a reasonable choice, i.e. selecting a higher value of r for longer subsequences and lower value of r for shorter subsequences. The length of sliding window, q is another important parameter that can be selected based on the application purpose. One may consider different values for this parameter to find some appropriate results.

As mentioned earlier, the objective of this chapter is to assign an anomaly score to each subsequence and select the subsequences with higher anomaly scores as anomalous parts of time series. To handle this task, a fuzzy clustering-based method has been employed.

4.2. Anomaly detection using a fuzzy C-Means clustering

4.2.1. Fuzzy C-Means clustering

Fuzzy C-Means (FCM) proposed by Dunn [2] and Bezdek [3] is one of the most popular and efficient objective function-based clustering techniques that has been applied successfully in different applications including clustering spatial, temporal, and spatio-temporal data (refer e.g., to [93, 98, 105–108]).

FCM partitions a set of N data $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N$ in into c $(1 < c < N)$ clusters. The result is a set of c prototypes $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_c$ and a partition matrix, $U=[u_{ik}]$, $i=1,2..,c$, $k=1,2,\ldots,N$, $u_{ik} \in [0,1]$, $\sum_{i=1}^{c} u_{ik} = 1 \; \forall k$, and $0 < \sum_{k=1}^{N} u_{ik} < N \; \forall i$, describing the membership degrees of the objects to the prototypes. This structure arises through the minimization of the following objective function:

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} \left\| \boldsymbol{v}_i - \boldsymbol{x}_k \right\|^2, \tag{4.3}$$

where, m $(m > 1)$ is a fuzzification coefficient and $\|\cdot\|$ denotes the Euclidean distance function. Table 4.1 shows the FCM algorithm.

Table 4.1. The FCM algorithm.

Set m and c, and initialize the partition matrix randomly,
Repeat
Compute the cluster centers $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_c$ as follows:

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m} \boldsymbol{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}}. \tag{4.4}$$

Update the partition matrix as follows:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\left\| \boldsymbol{v}_i - \boldsymbol{x}_k \right\|}{\left\| \boldsymbol{v}_j - \boldsymbol{x}_k \right\|} \right)^{2/(m-1)}}. \tag{4.5}$$

Until there is no a significant change in U

4.2.2. Anomaly detection

Clustering subsequence $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$ will lead to generating a set of prototypes, representing the normal structure of subsequences. Each normal subsequence in dataset is similar to one or more prototypes or it can be similar to a combination (in form of a weighted average) of prototypes. The more the subsequence is similar to the prototypes, the less anomalous it is. To evaluate how much a subsequence is similar to the revealed prototypes (or their combination) a reconstruction criterion has been considered in this chapter.

Pedrycz and de-Oliveira [74] proposed that FCM can be considered as an encoding scheme of data and the original data points (here subsequences) can be decoded (reconstructed) using the estimated cluster centers and partition matrix. Assuming that $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, ..., \hat{\boldsymbol{x}}_N$ are the reconstructed version of subsequences $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$ respectively, by minimizing the following sum of distances:

$$F = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} \left\| \boldsymbol{v}_i - \hat{\boldsymbol{x}}_k \right\|^2, \tag{4.6}$$

one may arrive at [74]:

$$\hat{\boldsymbol{x}}_k = \frac{\sum_{i=1}^{c} u_{ik}^{m} \boldsymbol{v}_i}{\sum_{i=1}^{c} u_{ik}^{m}}. \tag{4.7}$$

After calculating the reconstructed version of each subsequence using (4.7), the reconstruction error in (4.8), that is a squared Euclidean distance between a subsequence and its reconstructed version is considered as the evaluation criterion to estimate how much a subsequence is similar to the prototypes. In other words, for each subsequence the calculated reconstruction error using (4.8) is considered as its anomaly score.

$$E_k = \left\| \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k \right\|^2. \tag{4.8}$$

Figure 4.2 shows the overall scheme of the proposed method. As mentioned earlier, our objective in this chapter is to provide a unified framework to detect anomalies in amplitude and anomalies in shape.

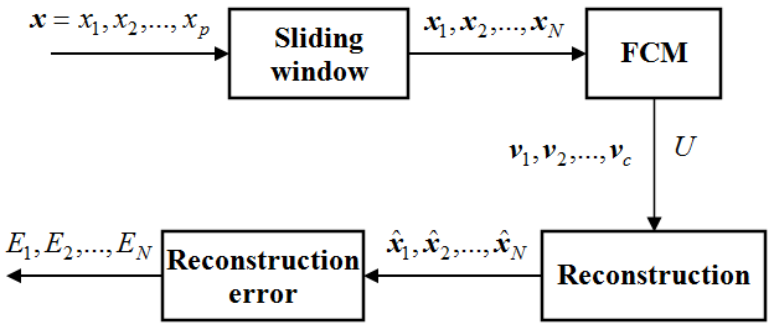


Figure 4.2. Overall scheme of the proposed anomaly detection.

As shown in Figure 4.2, the starting point of the proposed approach is generating a set of subsequences, $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$ using a sliding window. When the objective is to detect anomalies in amplitude, the Euclidean distance can be considered as a suitable dissimilarity measure and the generated subsequences can be employed in clustering process without any further preprocessing or representation. On the other hand, when detecting anomalies in shape is of concern, the generated subsequences cannot be employed directly in clustering. The reason is that the generated subsequences are not synchronized and the Euclidean distance function is not suitable to evaluate the similarity between time series with respect to their shape information. Although there are number of viable distance functions to measure the dissimilarity of asynchronous time series with respect to their shapes (e.g. dynamic time warping distance [16]), one has to be aware of the challenges we may encounter for optimizing the FCM objective function in dealing with those distance functions.

To compare subsequences based on their shape information, each subsequence is normalized to have a zero mean and a standard deviation equal to one. Then, each normalized subsequence is represented using a set of autocorrelation coefficients. Considering $\boldsymbol{x}_k$ as a subsequence with length q, its autocorrelation coefficient for lag s can be estimated using (4.9).

$$y_{k,s} = \frac{\sum_{t=s+1}^{q} (x_{k,t} - \bar{x}_k)(x_{k,t-s} - \bar{x}_k)}{\sum_{t=1}^{q} (x_{k,t} - \bar{x}_k)^2}. \tag{4.9}$$

As a matter of fact, autocorrelation coefficients estimate how much a signal matches its time-shifted version. By considering different legs $s = 1,2,...,q-1$, each subsequence is represented in a new feature space with length q-1. This representation of time series captures the shape information and removes existing shifts in asynchronous time series and the Euclidean distance function can be used efficiently to compare the subsequences in the new feature space. The idea of using autocorrelation representation of time series for fuzzy clustering was originally proposed in [29].

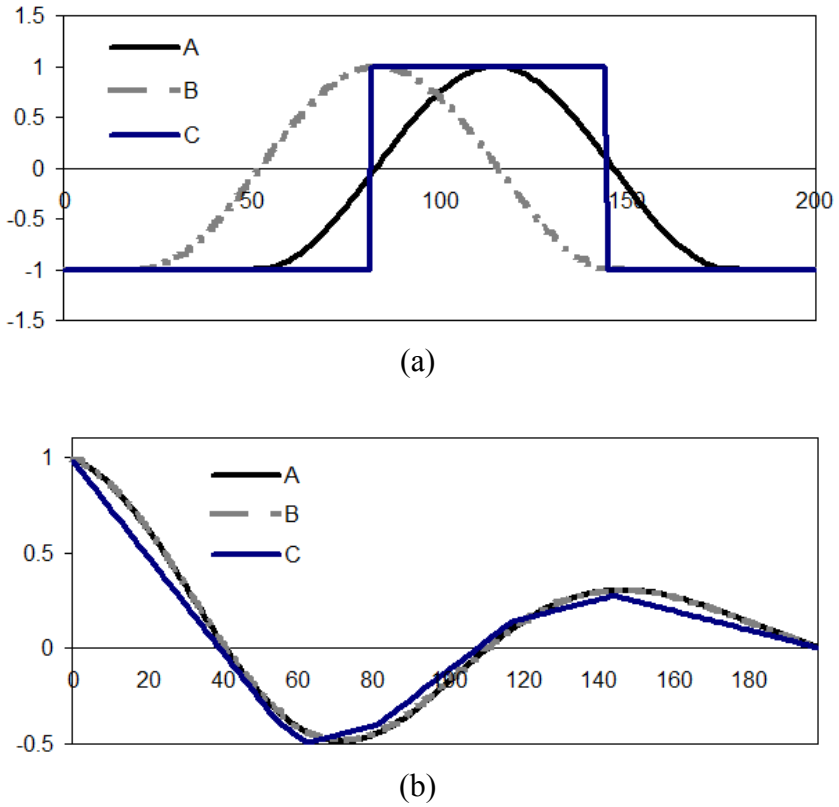


Figure 4.3. (a) Three time series and (b) their autocorrelation representation.

For illustrative purposes, let us consider Figure 4.3(a). In this figure, **A** is a sine wave, **B** is a shifted version of **A**, and **C** is a square shaped wave and is synchronized with **A**. Considering the Euclidean distance function to measure the dissimilarities between time series in this figure, we have $\|A - B\| > \|A - C\|$.

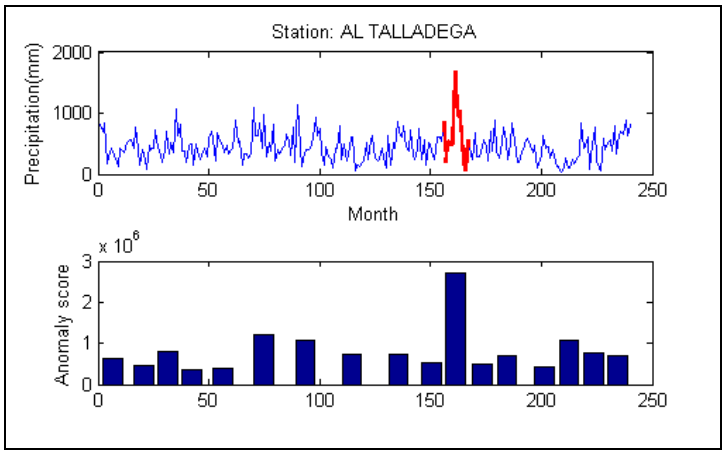
Although the time series **A** and **B** are quite similar, their Euclidean distance has a high amount because they are not synchronized. Figure 4.3(b) shows the autocorrelation representation of time series shown in Figure 4.3(a). In this figure we have $\|A - B\| < \|A - C\|$. The reason is that the autocorrelation function removes the available shifts in time, and easily asynchronous time series can be compared with each other in this new feature space with the use of the Euclidean distance function.

4.3. Experimental studies

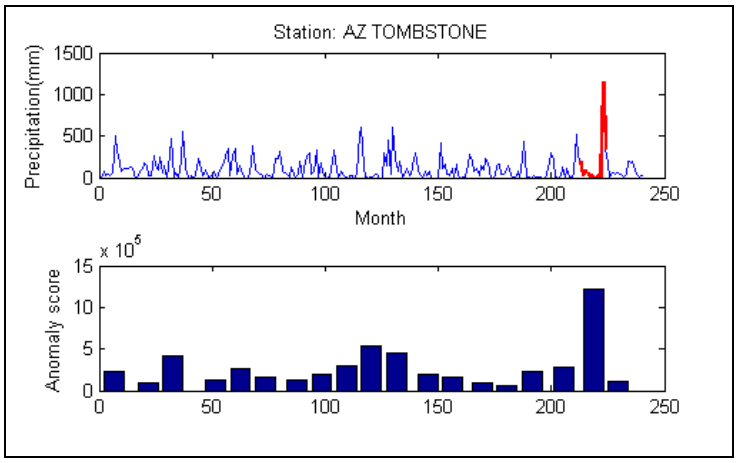
To illustrate the performance of the proposed method, two real datasets, one for anomaly detection in amplitude, and one for anomaly detection in shape are investigated.

4.3.1. Anomalies in amplitude

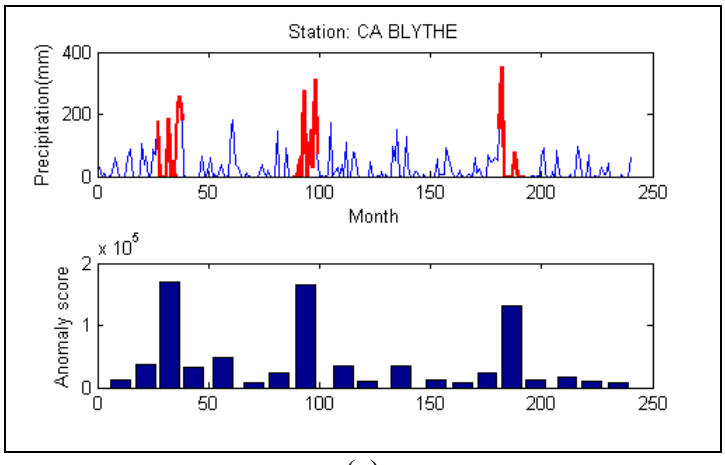
The United States monthly precipitation dataset [124] from 1990 to 2009 is considered. The length of time series in this dataset was 240 and four stations with some visible anomalies were chosen in our experiments. In FCM algorithm, the number of clusters, *c* as well as the fuzzification coefficient, *m* was set to 2. Moreover, since this dataset comprises monthly data, the length of sliding window was set to 12 that is equivalent to one year, and in each movement the sliding windows moves one time step. Figure 4.4(a)-(d) shows the results. Each figure is composed of two parts: the time series and the anomaly scores estimated for subsequences. Since the sliding window generates overlapping subsequences, for each set of overlapping subsequences, only the anomaly score corresponding to the most anomalous subsequence is shown. Moreover, in each time series the subsequences with higher anomaly scores are highlighted.


Station: AL TALLADEGA
Precipitation(mm)
2000
1000
0
0
50
100
150
200
250
Month
x 10^6
Anomaly score
3
2
1
0
0
50
100
150
200
250

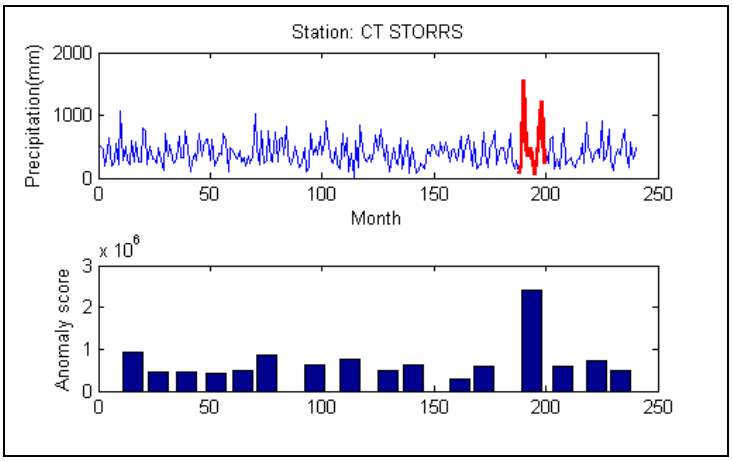
(a)


Station: AZ TOMBSTONE
Precipitation(mm)
1500
1000
500
0
0
50
100
150
200
250
Month
x 10^5
Anomaly score
15
10
5
0
0
50
100
150
200
250

(b)


Station: CA BLYTHE
Precipitation(mm)
400
200
0
0
50
100
150
200
250
Month
x 10^5
Anomaly score
2
1
0
0
50
100
150
200
250

(c)



(d)

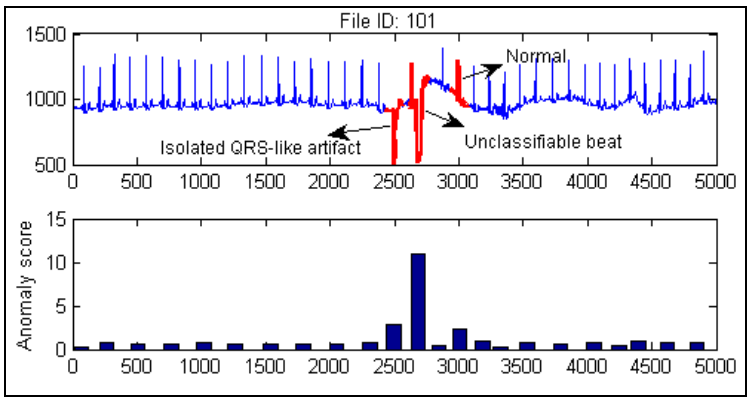
Figure 4.4. Monthly precipitation time series along with the estimated anomaly scores for different subsequences. In each figure, the subsequences with higher anomaly scores are highlighted.

As shown in these figures, the available anomalies in amplitude are detected using the proposed approach. Moreover, for all the other parts of data an anomaly score has been assigned to measure in which degree they are unusual in amplitude.

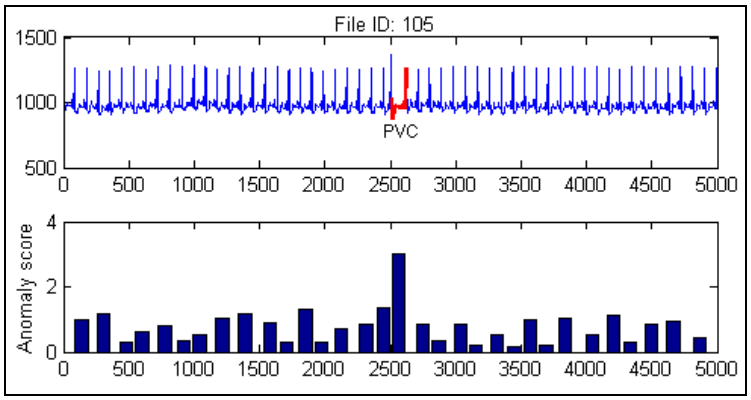
4.3.2. Anomalies in shape

The MIT-BIH arrhythmia dataset [125] for shape anomaly detection is considered. This dataset is composed of 48 half-hour annotated ECG signals. Four excerpts from the ECG signals in this dataset comprising some visible anomalies were selected, and similar to the previous experiment, in FCM algorithm the number of clusters as well as fuzzification coefficient was set to 2. To reduce the processing time, the excerpts are resampled from 360Hz to 128Hz. The length of each excerpt in our experiments is 5000 and the length of sliding window was set to around 1.2 times of average length of RR peaks to make sure that longer beats (e.g. PVC) can be incorporated in one subsequence. Moreover, the sliding window moves around 5% of the length of subsequences in each movement. After generating the subsequences, normalization has been employed and each

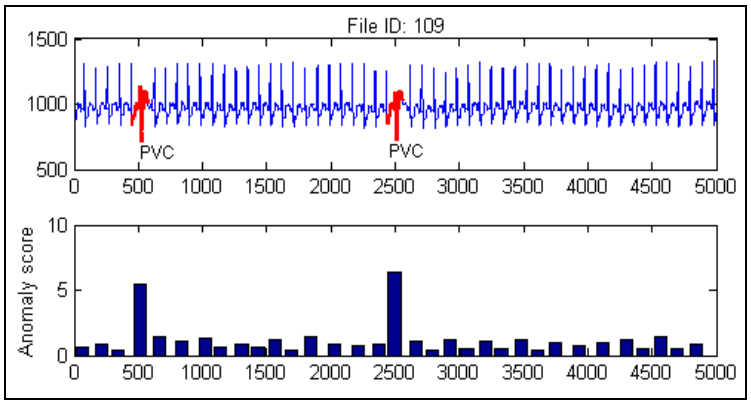
subsequence is represented using its autocorrelation coefficients. The clustering was applied over the new feature space.



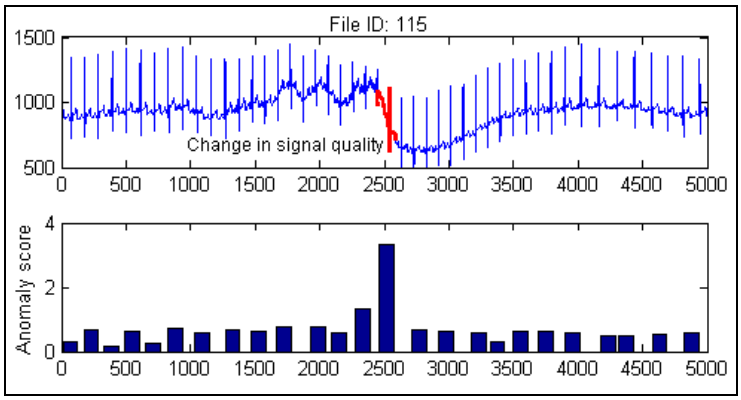
(a)



(b)



(c)



(d)

Figure 4.5. Some excerpts from MIT-BIH arrhythmia dataset for detecting anomalies in shape. In each figure, the subsequences with higher anomaly scores are highlighted.

Figure 4.5(a)-(d) shows the signals along with the estimated anomaly scores determined for different subsequences. In each signal, the subsequences with higher anomaly scores are highlighted and their corresponding annotation has been reported. As it can be seen from these figures, in most cases the detected anomalies are in type of PVC that is one of the most common arrhythmia heartbeats. In Figure 4.5(a) a normal beat has a high anomaly score. However as observed, this heartbeat is different from other normal beats in shape.

4.3.3. Parameter analysis

Parameters that have a direct impact on the performance of the proposed method are: length of sliding window, length of each movement of sliding window, and number of clusters and fuzzification coefficient in FCM. In this sub-section, we investigate the effect of the length of sliding window, q and propose a simple approach to find an optimal one. For the other parameters a similar procedure can be realized. Figure 4.6 is an excerpt of file 207 from MIT-BIH arrhythmia dataset and contains a visible anomaly in shape.

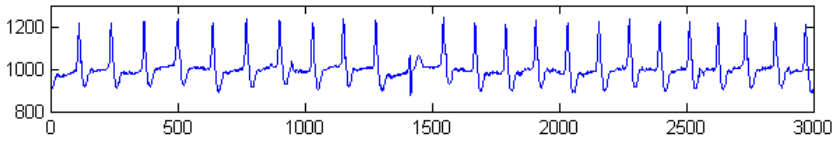


Figure 4.6. An excerpt from file 207 in MIT-BIH arrhythmia dataset.

Assume that h contains the calculated anomaly scores for all subsequences within a time series. We define a confidence term as

$$f = \frac{h_a}{\bar{h}} \tag{4.12}$$

where $\bar{h}$ is the average of anomaly scores in h, and h_a is the anomaly score corresponding to the anomalous subsequence i.e. the maximum score in h. This term is used to evaluate the performance of the proposed method. A higher value of f means that the proposed method assigned a high anomaly score to the anomalous subsequence and lower scores to the non-anomalous subsequences. As the result, each parameter that can maximize this performance index is more suitable. Note that here we assumed that there is only one anomalous subsequence in time series. In the case of more anomalous subsequences, one may define h_a as the average of anomaly scores corresponding to the anomalous subsequences.

Let us consider the length of each movement, r, equal to 5% of the length of sliding window, and the number of clusters and the fuzzification coefficient in FCM equal to 2. Figure 4.7 shows the amount of f for different length of sliding windows for the time series shown in Figure 4.6.

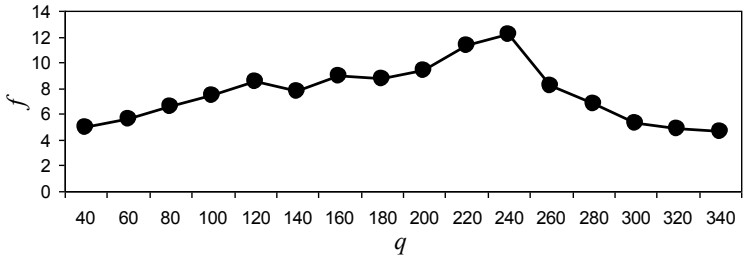


Figure 4.7. Different length of sliding windows vs. performance index.

As shown in this figure, the performance index, f has its optimal value at q=240, while for smaller and larger sliding windows it has lower amounts. The reason is that for small windows the anomalous part of time series cannot fit into a subsequence, and for large windows, the anomalous part of time series along with some non-anomalous parts has to be considered in one subsequence.

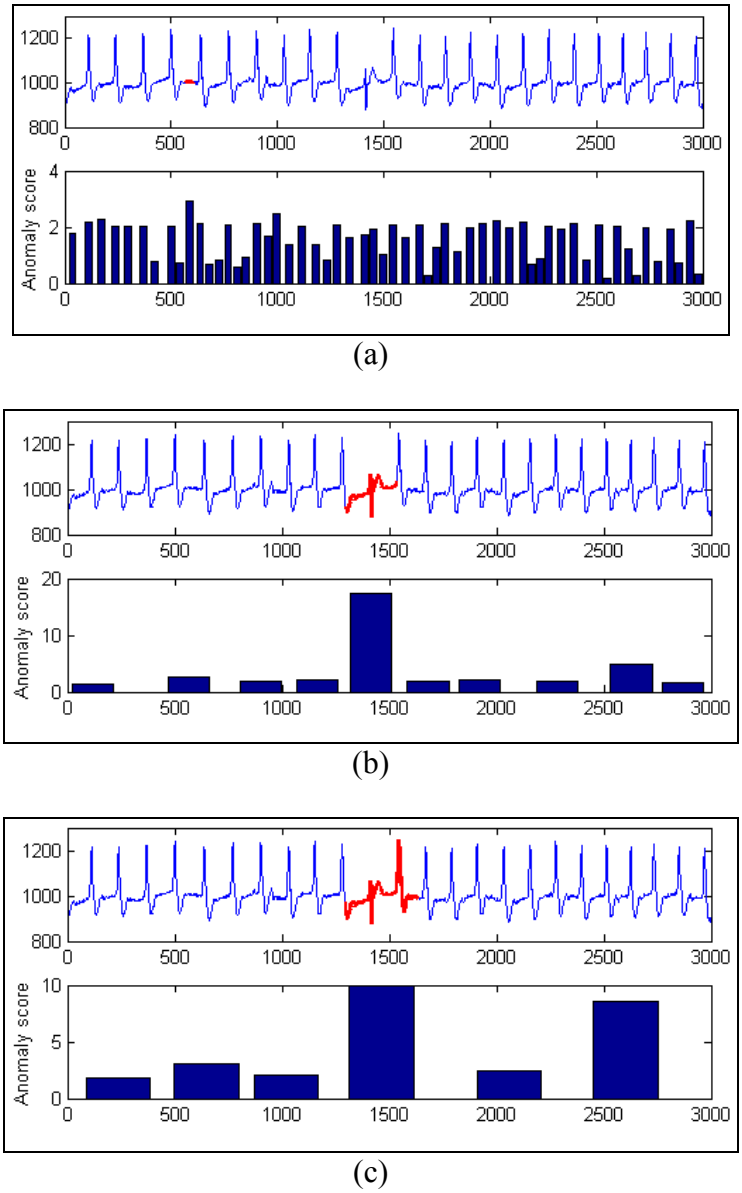


Figure 4.8. Detected anomalies in time series for different size of sliding windows. (a) q=40, (b) q=240, and (c) q=340.

Figure 4.8 illustrates this problem. In Figure 4.8(a) the size of sliding window was 40. We can see that the proposed approach even cannot find the anomalous part of time series. Moreover, most of anomaly scores are in a same range. In figure 4.8(b), the size of sliding window was set to 240 and we can see that the anomalous part of time series has a large anomaly score in comparison with the other parts. Finally, in Figure 4.8(c), the size of sliding window was 340 and as shown in this figure, some non-anomalous parts of time series have been considered as anomaly and the anomaly score corresponding to the detected anomalous subsequence is close to some non-anomalous subsequences.

4.4. Summary

A unified framework for detecting anomalies in amplitude and shape of time series is introduced. Using a fixed length sliding window a set of subsequences are generated, and the Fuzzy C-Means clustering is considered to reveal the available normal structures within subsequences. To measure the dissimilarity of each subsequence to different cluster centers, a reconstruction criterion is used and the calculated reconstruction error has been considered as anomaly score for each subsequence. For detecting anomalies in amplitude, the original representation of time series is considered, while for shape anomalies an autocorrelation representation of time series is used.

5. Clustering Spatial Time Series Using a Reconstruction Criterion

Since in spatial time series there are different data sources (say, spatial part and one or more time series part), clustering of this type of data poses some significant challenges. First, the diverse dimensionality and the range of the features originating from the corresponding data sources may easily lead to bias towards some data sources when carrying out clustering. Moreover, each data source comes with its own structure and a notion of distance could have a different meaning. It becomes apparent that in comparison with the generic FCM, we seek for clustering capable of dealing with the diversity of the blocks of features.

5.1. Problem formulation

Let us assume that there are N data $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_N$, each comprising its spatial and temporal components. The ith data $\boldsymbol{x}_i$ is represented as a concatenation of its spatial and temporal parts, namely $\boldsymbol{x}_i = [\boldsymbol{x}_i(s) | \boldsymbol{x}_i(t)]^{\mathrm{T}}$, where $\boldsymbol{x}_i(s)$ is the spatial part of $\boldsymbol{x}_i$, and $\boldsymbol{x}_i(t)$ denotes the temporal part (or its representation) of the same data point. Assume that there is one time series for each spatial location (spatial univariate time series), by considering r features in the spatial part (usually r=2) and q features in the temporal one, we have

$$\boldsymbol{x}_i = [\boldsymbol{x}_i(s) \,|\, \boldsymbol{x}_i(t)]^{\mathrm{T}} = [x_{i1}(s), \cdots, x_{ir}(s) | \, x_{i1}(t), \cdots, x_{iq}(t)]^{\mathrm{T}}. \tag{5.1}$$

Our interest is in the augmentation of the FCM algorithm so that the spatio-temporal nature of the data can be fully utilized in the clustering process. The aim of the FCM is to construct a collection of c information granules– clusters with the structure of data described by a collection of c prototypes $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_c$ and a fuzzy partition matrix $U = [u_{ik}]$, $i = 1, 2, ..., c$, $k = 1, 2, ..., N$. The objective

function of the FCM for a distance function d can be expressed as

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} d^2(\boldsymbol{v}_i, \boldsymbol{x}_k), \tag{5.2}$$

where m $(m > 1)$ is a fuzzification coefficient and the distance d is usually viewed as the Euclidean distance or its relatives such as the weighted Euclidean or the Mahalanobis distance [3]. When it comes to the spatial time series, the key point is to prudently capture a notion of distance which will clearly distinguish between the spatial and the temporal components in the problem at hand. Likewise we may like to accommodate a crucial possibility to strike a sound tradeoff between the distance determined with regard to the spatial and the temporal part of the feature vector. This is accomplished by forming an additive form of the distance function composed of the two components

$$d_{\lambda}^2(\boldsymbol{v}_i, \boldsymbol{x}_k) = \left\| \boldsymbol{v}_i(s) - \boldsymbol{x}_k(s) \right\|^2 + \lambda \left\| \boldsymbol{v}_i(t) - \boldsymbol{x}_k(t) \right\|^2, \quad \lambda \geq 0. \tag{5.3}$$

This augmented distance allows us control the effect of each part of data in the determination of the overall Euclidean distance and helps strike a sound balance between the impact of the spatial and temporal components of the data. When λ=0, the spatial component is considered and the temporal part is completely ignored. The higher the value of λ is, the more substantial is the impact of the temporal part of the spatial time series on the discovery of the structure. Subsequently, the above distance function is used in the objective function

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} d_{\lambda}^2(\boldsymbol{v}_i, \boldsymbol{x}_k). \tag{5.4}$$

Carrying out the optimization of J we arrive at the following expressions for the prototypes and the partition matrix

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m} \boldsymbol{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}}, \tag{5.5}$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\dfrac{d_{\lambda}^2(\boldsymbol{v}_i, \boldsymbol{x}_k)}{d_{\lambda}^2(\boldsymbol{v}_j, \boldsymbol{x}_k)} \right)^{1/(m-1)}}. \tag{5.6}$$

Let us explain the process of deriving (5.6) and (5.5) from the FCM objective function expressed in (5.4). We insert the proposed distance function (5.3) into the FCM objective function. We have:

$$J=\sum_{i=1}^{c}\sum_{k=1}^{N}u_{ik}^{m}\left(\left\|\boldsymbol{v}_i(s)-\boldsymbol{x}_k(s)\right\|^2+\lambda\left\|\boldsymbol{v}_i(t)-\boldsymbol{x}_k(t)\right\|^2\right). \tag{5.7}$$

To calculate the membership degrees, we define the augmented objective function, where the constraints are handled by Lagrange multiplier, γ, for data points q=1,2,…,N:

$$L=\sum_{i=1}^{c}u_{iq}^{m}\left(\left\|\boldsymbol{v}_i(s)-\boldsymbol{x}_q(s)\right\|^2+\lambda\left\|\boldsymbol{v}_i(t)-\boldsymbol{x}_q(t)\right\|^2\right)-\gamma(\sum_{i=1}^{c}u_{iq}-1). \tag{5.8}$$

We have

$$\frac{\partial L}{\partial u_{rq}}=m\,u_{rq}^{m-1}\left(\left\|\boldsymbol{v}_r(s)-\boldsymbol{x}_q(s)\right\|^2+\lambda\left\|\boldsymbol{v}_r(t)-\boldsymbol{x}_q(t)\right\|^2\right)-\gamma=0. \tag{5.9}$$

From (5.9) we have:

$$\gamma^{\,1/(m-1)}=u_{rq}\left(m\left(\left\|\boldsymbol{v}_r(s)-\boldsymbol{x}_q(s)\right\|^2+\lambda\left\|\boldsymbol{v}_r(t)-\boldsymbol{x}_q(t)\right\|^2\right)\right)^{1/(m-1)}. \tag{5.10}$$

Since in FCM we have $\sum_{j=1}^{c}u_{jq}=1$, we get

$$\sum_{j=1}^{c}\frac{\gamma^{\,1/(m-1)}}{\left(m\left(\left\|\boldsymbol{v}_j(s)-\boldsymbol{x}_q(s)\right\|^2+\lambda\left\|\boldsymbol{v}_j(t)-\boldsymbol{x}_q(t)\right\|^2\right)\right)^{1/(m-1)}}=1, \tag{5.11}$$

and

$$\sum_{j=1}^{c}\frac{u_{rq}\left(m\left(\left\|\boldsymbol{v}_r(s)-\boldsymbol{x}_q(s)\right\|^2+\lambda\left\|\boldsymbol{v}_r(t)-\boldsymbol{x}_q(t)\right\|^2\right)\right)^{1/(m-1)}}{\left(m\left(\left\|\boldsymbol{v}_j(s)-\boldsymbol{x}_q(s)\right\|^2+\lambda\left\|\boldsymbol{v}_j(t)-\boldsymbol{x}_q(t)\right\|^2\right)\right)^{1/(m-1)}}=1. \tag{5.12}$$

From (5.12) we have

$$u_{rq}=\frac{1}{\sum_{j=1}^{c}\left(\dfrac{\left\|\boldsymbol{v}_r(s)-\boldsymbol{x}_q(s)\right\|^2+\lambda\left\|\boldsymbol{v}_r(t)-\boldsymbol{x}_q(t)\right\|^2}{\left\|\boldsymbol{v}_j(s)-\boldsymbol{x}_q(s)\right\|^2+\lambda\left\|\boldsymbol{v}_j(t)-\boldsymbol{x}_q(t)\right\|^2}\right)^{1/(m-1)}}. \tag{5.13}$$

To calculate the prototypes we split (5.7) into two objective functions J_1 and J_2 as follows:

$$\begin{aligned} J_1 &= \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^m \left\| \boldsymbol{v}_i(s) - \boldsymbol{x}_k(s) \right\|^2, \\ J_2 &= \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^m \lambda \left\| \boldsymbol{v}_i(t) - \boldsymbol{x}_k(t) \right\|^2 \end{aligned}. \tag{5.14}$$

The minimization of J_1 and J_2 leads to the minimization of (5.7). To determine $\boldsymbol{v}_r(s)$ coming from J_1 we have:

$$\frac{\partial J_1}{\partial \boldsymbol{v}_r(s)} = 2\sum_{k=1}^{N} u_{rk}^m (\boldsymbol{v}_r(s) - \boldsymbol{x}_k(s)) = 0. \tag{5.15}$$

Finally we obtain

$$\boldsymbol{v}_r(s) = \frac{\sum_{k=1}^{N} u_{rk}^m \boldsymbol{x}_k(s)}{\sum_{k=1}^{N} u_{rk}^m}. \tag{5.16}$$

In the same manner, the prototypes corresponding to J_2 can be computed. Since we have $\boldsymbol{v}_r = [\boldsymbol{v}_r(s) \mid \boldsymbol{v}_r(t)]$, as the result:

$$\boldsymbol{v}_r = \left(\frac{\sum_{k=1}^{N} u_{rk}^m \boldsymbol{x}_k(s)}{\sum_{k=1}^{N} u_{rk}^m} \mid \frac{\sum_{k=1}^{N} u_{rk}^m \boldsymbol{x}_k(t)}{\sum_{k=1}^{N} u_{rk}^m} \right) = \frac{\sum_{k=1}^{N} u_{rk}^m \boldsymbol{x}_k}{\sum_{k=1}^{N} u_{rk}^m} \tag{5.17}$$

As usual, (5.5) and (5.6) are used in an iterative way in which the partition matrix and the prototypes are updated in a consecutive fashion. While the weight factor (λ) offers a badly needed flexibility to the method and could help in its optimization, it becomes crucial to arrive at a constructive way of selecting its optimal value. In what follows, we introduce a reconstruction criterion using which the factor's value becomes optimized.

5.2. Reconstruction error as evaluation criterion

A reconstruction criterion (RC) [74] to evaluate the structures revealed from clustering spatial time series is considered in this chapter. Figure 5.1 highlights the essence of this criterion.

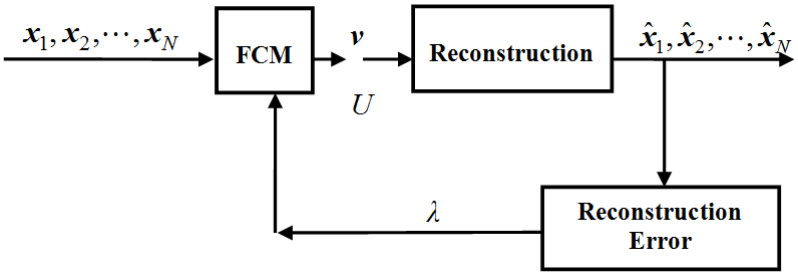


Figure 5.1. Overall scheme of evaluation of the clustering process completed with the aid of reconstruction criterion.

The essence of this evaluation process is to *reconstruct* the original data using the cluster prototypes and the partition matrix by minimizing the following sum of distances [74]

$$F = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m} \left\| \boldsymbol{v}_i - \hat{\boldsymbol{x}}_k \right\|^2, \tag{5.18}$$

where $\hat{\boldsymbol{x}}_k$ is the reconstructed version of $\boldsymbol{x}_k$. Zeroing gradient of F with respect to $\hat{\boldsymbol{x}}_k$, we get

$$2\sum_{i=1}^{c} u_{ik}^{m} \left(\boldsymbol{v}_i - \hat{\boldsymbol{x}}_k \right) = 0, \tag{5.19}$$

and then we have

$$\hat{\boldsymbol{x}}_k = \frac{\sum_{i=1}^{c} u_{ik}^{m} \boldsymbol{v}_i}{\sum_{i=1}^{c} u_{ik}^{m}}. \tag{5.20}$$

Once the reconstruction has been completed, viz. $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_N$ were constructed with the use of (5.20), the quality of reconstruction regarded as a function of λ is expressed in the form

$$E(\lambda) = \sum_{k=1}^{N} \left\| \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k \right\|^2 = \sum_{k=1}^{N} \left\| \boldsymbol{x}_k(s) - \hat{\boldsymbol{x}}_k(s) \right\|^2 + \sum_{k=1}^{N} \left\| \boldsymbol{x}_k(t) - \hat{\boldsymbol{x}}_k(t) \right\|^2, \tag{5.21}$$

where

$$\left\| \boldsymbol{x}_k(s) - \hat{\boldsymbol{x}}_k(s) \right\|^2 = \frac{1}{r} \sum_{j=1}^{r} \frac{\left(x_{kj}(s) - \hat{x}_{kj}(s) \right)^2}{\sigma_j^2}, \tag{5.22}$$

and

$$\left\|\boldsymbol{x}_k(t)-\hat{\boldsymbol{x}}_k(t)\right\|^2=\frac{1}{q}\sum_{j=1}^{q}\frac{\left(x_{kj}(t)-\hat{x}_{kj}(t)\right)^2}{\sigma_j^2}, \tag{5.23}$$

and σ_j^2 is the variance of jth feature. Given that commonly the spatial part and the temporal part are expressed in spaces of very different dimensionalities (typically $r<<q$), in these two we use the normalized Euclidean distances in order to avoid any bias towards any particular component of the distance. The reconstruction error $E(\lambda)$ is a function of λ and its minimum is determined by a systematic sweeping through a certain range of the values of λ. This approach, instead of any more sophisticated one-dimensional search is considered because learning about the form of this index as a function of λ is also of interest. Table 5.1 shows the pseudocode of the proposed algorithm.

Table 5.1. Clustering spatial time series using reconstruction criterion.

Given:
$\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_N$: spatial time series
c: number of clusters
m: fuzzification coefficient

Output:
U: a $c \times N$ partition matrix
$\boldsymbol{v}=\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots \boldsymbol{v}_c$: set of spatio-temporal prototypes

Algorithm:
for each λ in range [0, M] **do** //M is a large number
  Randomly initialize partition matrix U
  Repeat
    Calculate spatio-temporal prototypes using (5.5)
    Update partition matrix U using (5.6)
    Calculate the objective function J using (5.4)
  Until there is no significant change in U
  Reconstruct the spatio-temporal data using (5.20)
  Calculate reconstruction error using (5.21)
end
Select the partition matrix and prototypes corresponding to the minimum reconstruction error as the final result.

5.3. Experimental studies

Two datasets, namely a synthetic data and the Alberta temperature dataset have been considered in this section to illustrate the proposed technique.

5.3.1. Synthetic data

In this section, we investigate the behavior of the clustering results quantified in terms of the reconstruction criterion for two synthetic datasets. Figure 5.2(a) shows the spatial component of these datasets where P1, P2, P3 and P4 are groups of associated with four categories of time series of length of 256 samples. We considered two scenarios. In the first one, Figure 5.2(b), the time series are clearly distinguishable while those shown in Figure 5.2(c) exhibit a significant level of overlap (less distinguishable data). The generated time series in these figures are a kind of increasing and decreasing time series encountered in control charts patterns [88]. In Figure 5.3, we presented one of the time series along with its corresponding representations, namely DFT(32), PAA(32), and DWT(32). The notion DFT(32) means the DFT representation with length 32.

We systematically sweep through the range of values of λ to find its value where the reconstruction error attains its minimum. Table 5.2 presents the optimal values of λ along with the corresponding reconstruction error reported for several number of clusters, c=2, 3 and 4, and different representation methods with length 8, 16 and 32. Note that the reported reconstruction error is a sum of squared Euclidean distances between the original extracted features and the reconstructed features (see (5.21)). In all experiments, the value of the fuzzification coefficient m was set to 2.

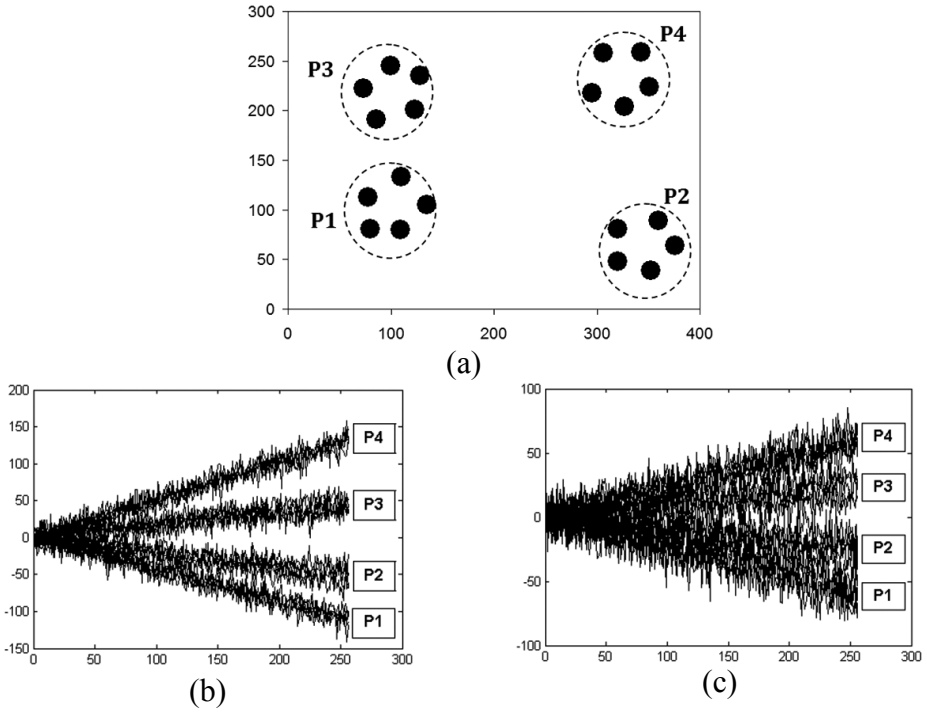


Figure 5.2. Synthetic spatio-temporal data: (a) spatial component, (b) temporal component of more distinguishable dataset, and (c) temporal component of less distinguishable dataset.

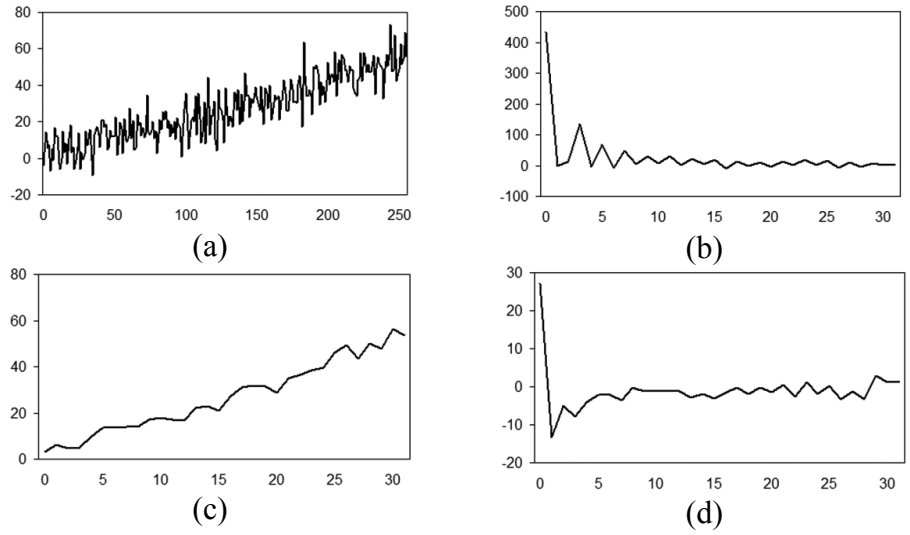


Figure 5.3. (a) A selected time series, and its representations with the use of: (b) DFT(32), (c) PAA(32), and (d) DWT(32).

Table 5.2. Optimal values of λ and associated reconstruction error for the synthetic datasets

(a) more distinguishable time series			
Representation	*c*=2	*c*=3	*c*=4
DFT(8)	0.058, 17.77	0.031, 10.83	0, 6.44
DFT(16)	0.048, 18.69	0.031, 11.48	0, 7.05
DFT(32)	0.048, 19.8	0.03, 12.58	0, 8.08
PAA(8)	0.95, 12.84	1, 5.78	0, 1.21
PAA(16)	0.5, 13.12	0.5, 6.07	0, 1.5
PAA(32)	0.25, 13.48	0.25, 6.42	0, 1.83
DWT(8)	7, 13.19	7.5, 5.97	0, 1.32
DWT(16)	8.5, 14.72	7.5, 7.68	0, 3.13
DWT(32)	10, 19.39	6.5, 12.41	0, 7.84
(b) less distinguishable time series			
Representation	*c*=2	*c*=3	*c*=4
DFT(8)	0.4, 18.92	0.1, 11.99	0, 7.5
DFT(16)	0.75, 20.32	0.11, 13.66	0, 8.84
DFT(32)	1, 21.29	0.11, 14.44	0, 9.76
PAA(8)	4.5, 13.64	3.5, 6.25	80, 1.81
PAA(16)	2.5, 14.2	1.5, 6.91	40, 2.3
PAA(32)	1.5, 14.82	0.85, 7.51	20, 3
DWT(8)	40, 14.44	25, 7.08	1000, 2.43
DWT(16)	55, 19.1	25, 11.69	450, 7.06
DWT(32)	10000, 23.27	20, 16.24	0, 11.74

The table visualizes the effect of different parameters on the optimal value of λ and the resulting reconstruction error. Among different representation methods, the DFT representation has the lowest value of the optimal λ, while the DWT assumes the highest value. The reason is that the magnitude of features is different depending on the representation method used.

As shown in this table, given a higher dimensionality of the representation space used for the temporal part of data, the optimal value of λ will occur in a lower amount to prevent bias towards temporal part in the clustering process. With the increase of the number of clusters, the reconstruction error is reduced. Having more visible structure in the more distinguishable dataset (Figure 5.2(b)), its reconstruction error usually is lower than the one reported for the less distinguishable dataset.

Let us investigate how λ impacts the effect arising from the temporal and spatial components of the data. We use the less distinguishable dataset, Figure 5.2(c), set

the number of clusters to 2, and use PAA(16) as the representation method of the time series. Figure 5.4 shows the results in the form of a contour plot of the obtained membership functions. The values λ=0 and λ=10,000 are treated as the extreme cases: when λ=0, the spatial part is involved in clustering while the second boundary focuses on the temporal part of the data. It becomes visible that the changes of λ lead to the shift of the contour plots which are reflective of the growing impact of the temporal or spatial component of the data. In the sequel, we investigate the impact of λ on the reconstruction error. In the series of experiments, we set the number of clusters to c=3. The DFT(16) is used as the representation method. Figure 5.5 displays the plot of reconstruction error vs. different values of λ. The optimal value of λ is clearly visible.

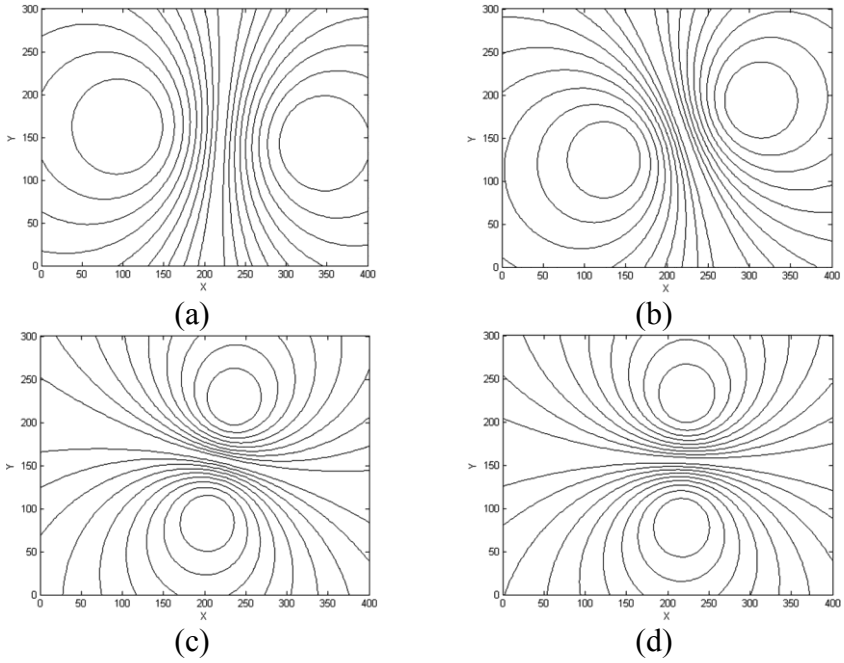


Figure 5.4. Contour plots of membership functions for selected values of λ and c=2, PAA(16) representation and less distinguishable dataset. (a) λ=0, (b) λ=1 (c) λ=3, and (d) λ=10,000.

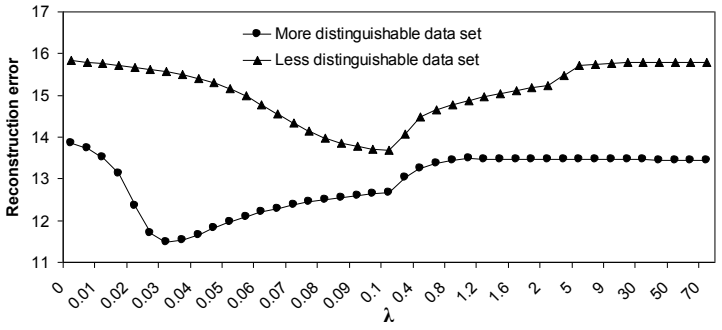


Figure 5.5. Plots of reconstruction error vs. λ for c=3 and DFT(16) representation.

Figure 5.6 shows the constructed clusters for λ=0, optimal value of λ, and λ=70, and *c*=3. The stars shown in these figures represent spatial prototypes.

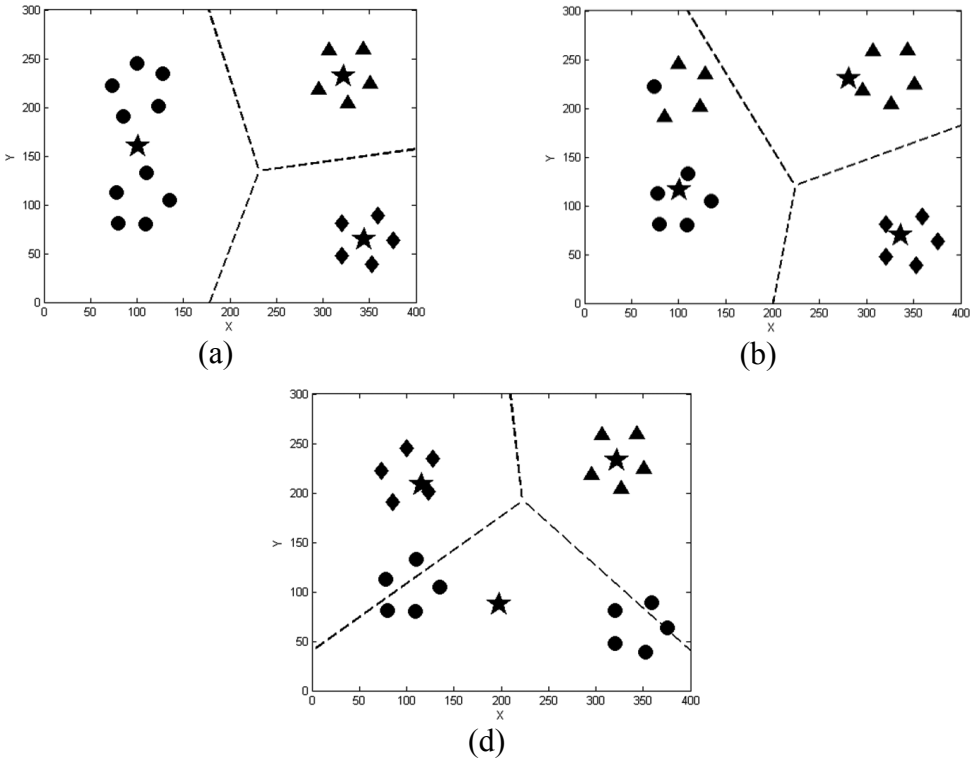
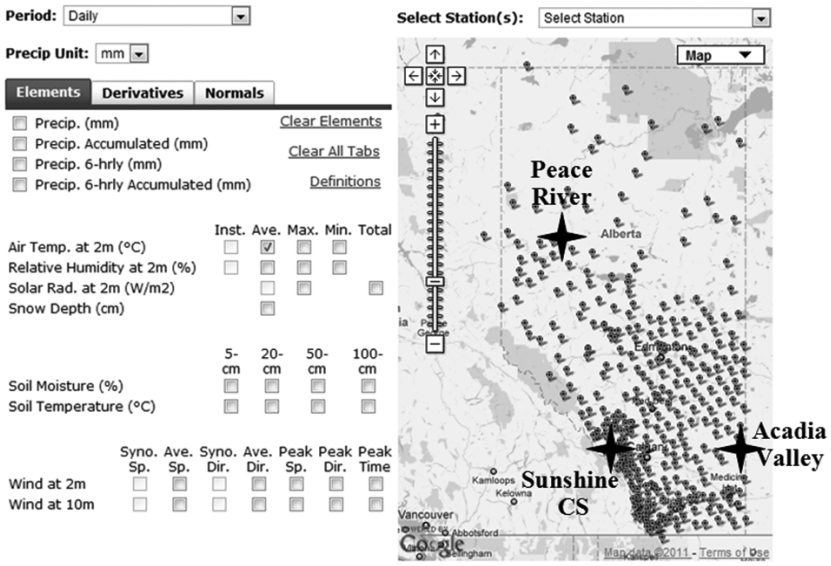


Figure 5.6. Clusters obtained for the less distinguishable dataset for c=3, DFT(16) and different values of λ: (a) λ=0 (b) optimal value of λ and (c) λ=70.

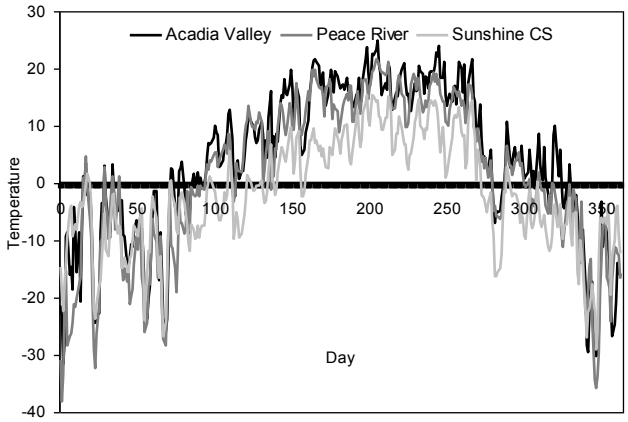
5.3.2. Alberta temperature data in different seasons

In this section we investigate the proposed method in application to the Alberta temperature dataset including daily average temperature. Alberta agriculture and rural development, provides updated agriculture-related data including daily temperature, humidity, precipitation, etc. The data are recorded by a number of stations located within the province of Alberta. For each station the geographical coordinates in the form of its latitude and longitude is provided. These data are available online at www.agric.gov.ab.ca. Figure 5.7(a) shows a snapshot of the system with three highlighted stations located in South East, South West, and North West Alberta. Figure 5.7(b) shows the average daily temperature recorded at these stations in 2009. As can be seen from this figure, different stations located in different parts of province come with different temperature patterns. Therefore grouping (clustering) these stations based on their locations and their daily average temperature (or any other variable e.g. precipitation) generates some useful insights with potential applicability to various domains. We consider the temperature data recorded for 2009 - 2011 at 246 stations located across Alberta. Notice that in the experiments, in the first step we project latitude and longitude coordinates to Cartesian coordinates to be used in the calculations of the Euclidean distance.

We split the daily average temperature data recorded in 2009 into four seasons: Spring, Summer, Fall and Winter, and run the experiments using the reconstruction criterion while the number of clusters varies from 2 to 5. The length of each time series is about 90 (depends on season) and for each representation method, the length of 8 has been chosen. Table 5.3 summarizes the results.



(a)



(b)

Figure 5.7. (a) A snapshot of the Alberta Agriculture and Rural Development system and three highlighted stations (www.agric.gov.ab.ca), and (b) Daily average temperature in year 2009 for the highlighted stations.

What could have been expected, when forming more clusters, the reconstruction error is reduced. Furthermore, from this table we can see that in some cases the optimal value of λ is equal to 0. This means that involving temporal information in these cases does not help the method to reconstruct data in a more accurate way. Figure 5.8 shows the contour plot of the membership degrees of the clusters obtained for different seasons of the year.

Table 5.3. The optimal value of λ and the associated reconstruction error for 246 stations in the Alberta temperature dataset in different seasons of 2009.

	c=2	c=3	c=4	c=5
Spring				
DFT(8)	0.2, 273.93	0.3, 192.4	0.02, 165.61	0.085, 136.63
PAA(8)	25, 277.55	8.5, 193.81	15, 159.11	0.95, 138.19
DWT(8)	200, 259.9	55, 169.43	20, 145.72	7.5, 114.79
Summer				
DFT(8)	0.02, 273.18	0.35, 187.62	0.55, 137.95	0.35, 122.49
PAA(8)	2250, 295.98	45, 177.52	50, 106.34	45, 80.64
DWT(8)	0, 299.93	125, 207.85	150, 127.48	250, 110.8
Fall				
DFT(8)	0.15, 299.67	0.1, 213.05	0.95, 162.13	0.15, 145.26
PAA(8)	175, 303.05	65, 187.41	55, 112.52	85, 86.04
DWT(8)	875, 326.29	300, 228.69	200, 144.92	400, 124.9
Winter				
DFT(8)	0.3, 286.08	0, 191.91	4.5, 133.48	4, 114.54
PAA(8)	5.5, 336.48	100, 233.43	50, 132.13	50, 115.83
DWT(8)	150, 280.23	175, 188.65	350, 140.42	400, 118.86

For different seasons we encounter different structures. This is quite reasonable because in some seasons several locations on the map are similar in temperature while in some other seasons they might be very different. Moreover we can see that the Spring clusters are similar to the Winter clusters while Summer clusters are similar to the Fall clusters. The reason is that in the Spring and Winter, the temperature is low in most parts of Alberta, so that there is no significant difference in temperature at most stations. As the result, the spatial part of data has more effect on the resulting clusters. On the other hand, in the Summer and Fall, the magnitude of temperature in the Rocky Mountains area (south west Alberta) is significantly different from the temperature recorded in some other areas (as can be seen from Figure 5.7(b)), so that the temporal part of the data has more effects. Figure 5.9 shows the clusters obtained for Summer 2009 data, optimal value of λ and c=3. The stars denote the spatial prototypes. There are clear differences between the clusters when using different representations of the time series. This is not surprising as different representation methods capture different facets of the time series. Also for each representation method the

distinguishability of the features can be different and as a result for different representation methods the revealed structures in temporal part of data can be more or less significant.

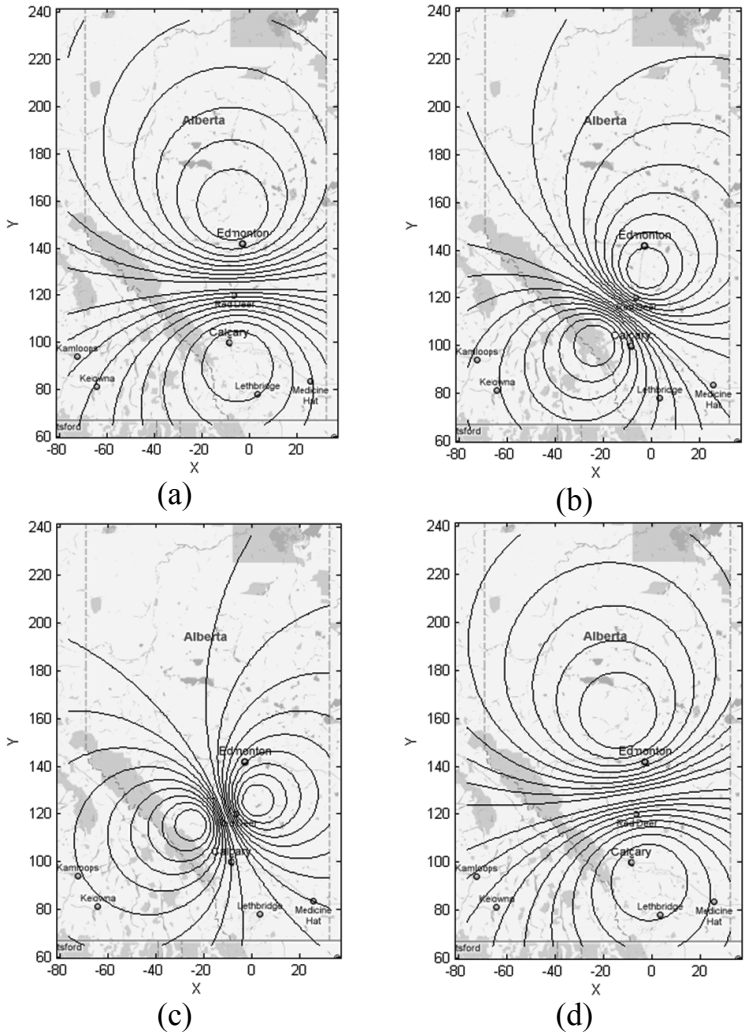


Figure 5.8. Clusters visualized in the form of contour plot of the membership degrees for successive seasons of 2009, c=2 and PAA(8) representation: (a) Spring, (b) Summer, (c) Fall, and (d) Winter.

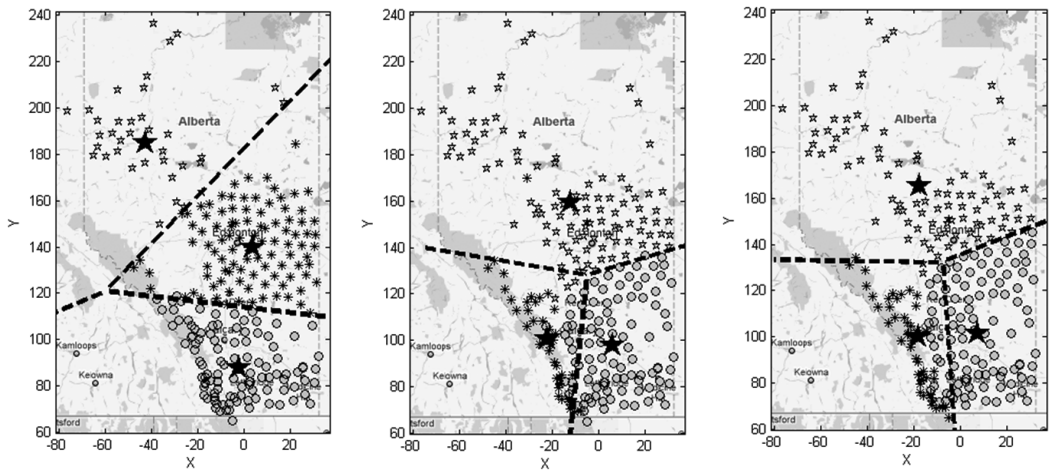


Figure 5.9. Clusters of spatio-temporal data - Summer 2009 data, *c*=3 and (a) DFT(8), (b) PAA(8), and (c) DWT(8). The optimal values of λ are 0.35, 45, and 125 respectively.

5.4. Summary

We have introduced the concept and algorithmic framework of fuzzy clustering for spatial univariate time series. It was shown that given a different nature of spatial and temporal components of the data, their different treatment is realized through a flexible distance function where a parameter λ controlling the influence of temporal and spatial components is optimized through the minimization of a reconstruction criterion. The optimal value of this parameter can be achieved by systematically sweeping through the range of values. The proposed technique examined over a synthetic and a real dataset. Experimental results show that using different representation of time series, one may obtain different clustering results. This is quite convincing because different representation techniques capture different characteristics of data leading to different results.

6. Clustering Spatial Time Series Using a Prediction Criterion

In the previous chapter we developed a spatial time series clustering with the aid of a reconstruction criterion. In this chapter a prediction criterion [69] has been considered as the evaluation criterion.

6.1. Problem formulation

Let us recall from the previous chapter that for clustering N spatial univariate time series, the objective function of FCM by considering the proposed composite distance function can be rewritten as:

$$J = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m}\left(\left\|\boldsymbol{v}_i(s)-\boldsymbol{x}_k(s)\right\|^2 + \lambda\left\|\boldsymbol{v}_i(t)-\boldsymbol{x}_k(t)\right\|^2\right),\ \lambda \geq 0. \tag{6.1}$$

Moreover, carrying out the optimization of J we arrive at the following expressions for the prototypes and the partition matrix

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m}\boldsymbol{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}}, \tag{6.2}$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\dfrac{\left\|\boldsymbol{v}_i(s)-\boldsymbol{x}_k(s)\right\|^2 + \lambda\left\|\boldsymbol{v}_i(t)-\boldsymbol{x}_k(t)\right\|^2}{\left\|\boldsymbol{v}_j(s)-\boldsymbol{x}_k(s)\right\|^2 + \lambda\left\|\boldsymbol{v}_j(t)-\boldsymbol{x}_k(t)\right\|^2}\right)^{1/(m-1)}}. \tag{6.3}$$

As usual, these two formulas are used in an iterative way in which the partition matrix and the prototypes are updated in a consecutive fashion. As discussed earlier, the parameter λ, plays an important role in controlling the effect of each part of data in the clustering process. To find an optimal value for this parameter, a prediction criterion (PC) is considered in this chapter.

6.2. Prediction error as evaluation criterion

The essence of the prediction criterion is to *predict* the temporal component of the data by using the available spatial structure. Figure 6.1 shows the overall scheme of this criterion.

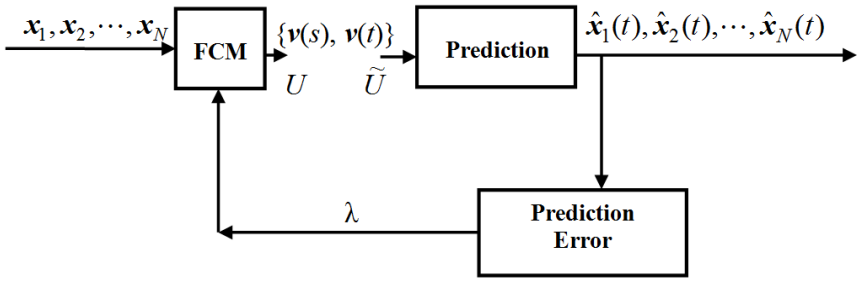


Figure 6.1. Overall scheme of evaluation of the clustering process completed with the aid of prediction criterion.

Starting with an initial value of λ, one may cluster the spatial time series using a FCM technique. The result of this clustering will be a set of cluster centers and a partition matrix. Since each spatial time series is composed of a spatial and a temporal part, the cluster centers (prototypes) are composed of a spatial part, $\boldsymbol{v}(s)$, and a temporal part , $\boldsymbol{v}(t)$, as well. Using the spatial part of data along with the spatial part of the calculated cluster centers, we form a new partition matrix, denoted by $\widetilde{U}$, as follows [69]

$$\widetilde{u}_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{\|\boldsymbol{v}_i(s)-\boldsymbol{x}_k(s)\|}{\|\boldsymbol{v}_j(s)-\boldsymbol{x}_k(s)\|}\right)^{2/(m-1)}},\ i=1,2,\ldots,c, \text{ and } k=1,2,\ldots,N. \tag{6.4}$$

With the use of this new partition matrix and the temporal part of the cluster centers , $\boldsymbol{v}(t)$, we minimize the following sum of distances

$$F = \sum_{i=1}^{c}\sum_{k=1}^{N}\widetilde{u}_{ik}^{m}\|\boldsymbol{v}_i(t)-\hat{\boldsymbol{x}}_k(t)\|^2, \tag{6.5}$$

where $\hat{\boldsymbol{x}}_k(t)$ is the predicted temporal part of the *k*th data. By zeroing the gradient of *F* with respect to $\hat{\boldsymbol{x}}_k(t)$ we have

$$\hat{\boldsymbol{x}}_k(t) = \frac{\sum_{i=1}^{c} \tilde{u}_{ik}^{m} \boldsymbol{v}_i(t)}{\sum_{i=1}^{c} \tilde{u}_{ik}^{m}}. \tag{6.6}$$

The quality of prediction is evaluated using the following prediction error

$$E(\lambda) = \sum_{k=1}^{N} \left\| \boldsymbol{x}_k(t) - \hat{\boldsymbol{x}}_k(t) \right\|^2 = \frac{1}{q} \sum_{k=1}^{N} \sum_{j=1}^{q} \frac{\left(x_{kj}(t) - \hat{x}_{kj}(t) \right)^2}{\sigma_j^2}. \tag{6.7}$$

It takes on a form of the sum of the normalized Euclidean distances between the temporal part of the data and the predicted temporal part. Similar to the reconstruction criterion, in the previous chapter, the intent is to minimize $E(\lambda)$ by adjusting the value of λ. Table 6.1 shows the pseudocode of the proposed algorithm.

Table 6.1. The pseudocode of the clustering method using prediction criterion.

Given:
$\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_N$: spatio-temporal data
c: number of clusters
m: fuzzification coefficient

Output:
U: a $c \times N$ partition matrix
$\boldsymbol{v} = \boldsymbol{v}_1, \boldsymbol{v}_2, \cdots \boldsymbol{v}_c$: set of spatio-temporal prototypes

Algorithm:
for each λ in range [0, M] **do** //M is a large number
  Randomly initialize partition matrix *U*
  Repeat
    Calculate spatio-temporal prototypes using (6.2)
    Update partition matrix *U* using (6.3)
    Calculate the objective function *J* using (6.1)
  Until there is no significant change in *U*
  Generate new partition matrix $\tilde{U}$ using (6.4)
  Predict the temporal part of data using (6.6)
  Calculate prediction error using (6.7)
end
Select the partition matrix and prototypes corresponding to the minimum prediction error as the final results.

6.3. Experimental studies

In this section, we investigate the behavior of the clustering results quantified in terms of the prediction criterion. The datasets investigated in the previous chapter are considered here as well.

6.3.1 Synthetic data

Let us consider the less and the more distinguishable synthetic datasets discussed in the previous chapter and examine the behavior of the prediction criterion over these two datasets. Figure 6.2 shows the datasets.

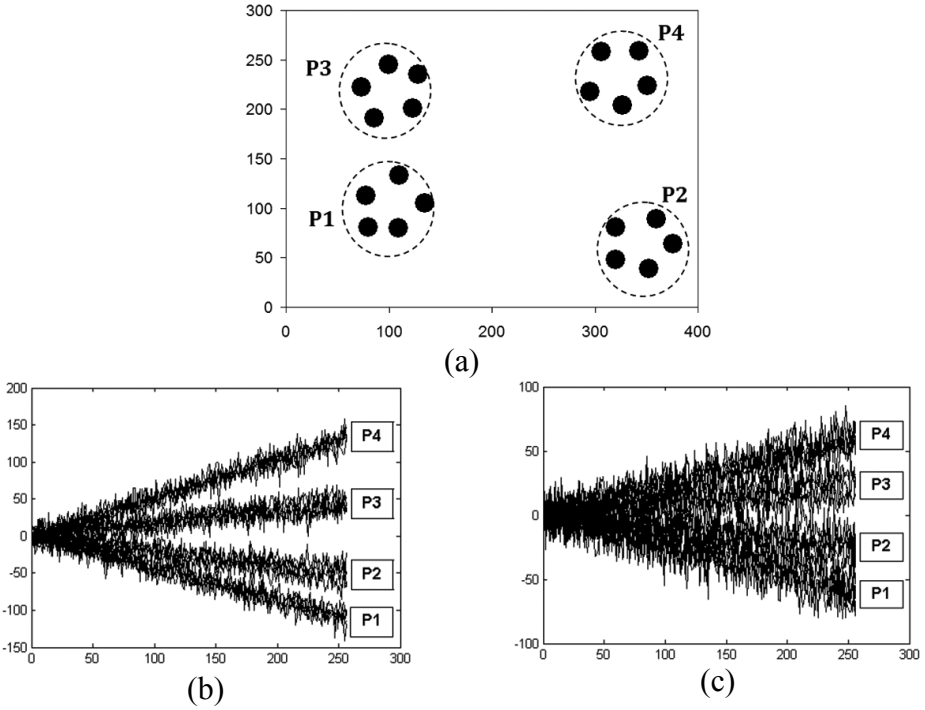


Figure 6.2. Synthetic spatio-temporal data: (a) spatial component, (b) temporal component of more distinguishable dataset, and (c) temporal component of less distinguishable dataset.

Similar to the reconstruction criterion, here we systematically sweep through the range of values of λ to find its value where the prediction error attains its

minimum. Table 6.2 presents the optimal values of λ along with the corresponding prediction error reported for several number of clusters, *c*=2, 3 and 4, and different representation methods with length 8, 16 and 32.

Table 6.2. Optimal values of λ and the associated prediction error for the synthetic datasets.

(a) more distinguishable time series			
Representation	*c*=2	*c*=3	*c*=4
DFT(8)	0.048, 7.7	0.14, 8.85	0.014, 5.46
DFT(16)	0.044, 8.52	0.098, 9.86	0.024, 6.07
DFT(32)	0.043, 9.52	0.09, 10.68	0.038, 7.1
PAA(8)	1.5, 3.18	3, 6.07	0.95, 0.23
PAA(16)	0.65, 3.39	1.5, 6.25	0.55, 0.53
PAA(32)	0.3, 3.73	0.7, 6.49	0.2, 0.85
DWT(8)	9, 3.52	20, 6.18	6.5, 0.34
DWT(16)	10, 5.16	25, 7.84	7.5, 2.15
DWT(32)	9.5, 9.45	20, 11.23	4, 6.86
(b) less distinguishable time series			
Representation	*c*=2	*c*=3	*c*=4
DFT(8)	0.17, 8.8	0.15, 10.05	0.039, 6.52
DFT(16)	0.18, 10.39	0.12, 11.19	0.045, 7.86
DFT(32)	0.18, 11.26	0.12, 11.98	0.048, 8.78
PAA(8)	5, 4.07	4, 6.67	6.5, 0.83
PAA(16)	2.5, 4.57	2, 6.99	15, 1.32
PAA(32)	1.5, 5.23	0.95, 7.5	1, 2.02
DWT(8)	35, 5.06	25, 6.82	2000, 1.47
DWT(16)	35, 9.16	20, 9.75	25, 6.1
DWT(32)	35, 13.2	20, 13.57	8, 10.75

We can see that most of the conclusions obtained when dealing with the reconstruction criterion hold here. There is an exception, however. Sometimes with the increase in the number of clusters, the error does not decrease. For example, the value of the error for *c*=3 is higher than the one for *c*=2. The reason is that for the generated datasets, by considering the number of clusters *c*=3 the "position" of the spatial part of prototypes and the "structure" of the temporal part of prototypes are not efficient for prediction, as the predicted time series are the weighted (calculated using the position of spatial part of prototypes in form of $\widetilde{U}$) average of temporal parts of prototypes.

Let us investigate the impact of λ on the prediction error. We set the number of clusters to *c*=3, and consider DFT(16) as the representation method. Figure 6.3 displays the plot of prediction error vs. different values of λ. The optimal value of λ is clearly visible in this figure.

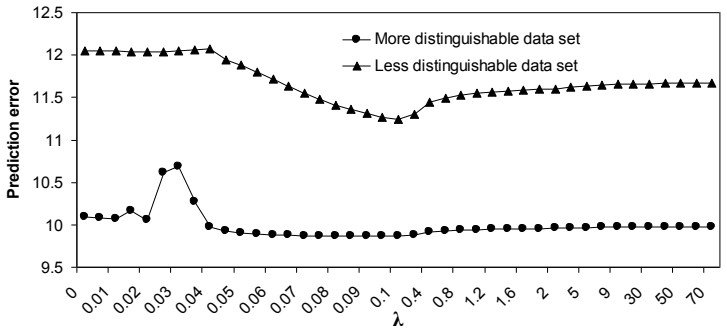


Figure 6.3. Plots of prediction error vs. λ for *c*=3 and DFT(16) representation of time series part of data.

Figure 6.4 shows the constructed clusters for the less distinguishable datasets, for λ=0, optimal value of λ, and λ=70 for the prediction criterion. The number of clusters in *c*=3. The stars represent spatial prototypes. The obtained results are similar to the results achieved by the reconstruction criterion.

6.3.2. Alberta daily average temperature data for 2009 to 2011

We considered daily average temperature, for 246 stations in Alberta in the time period 2009 to 2011 (see Figure 5.7) and built the clusters to investigate the prediction criterion. Table 6.3 shows the optimal amount of λ and its corresponding prediction error for these 246 stations and number of clusters *c*=2, 4, 6, 8 and 10. The length of time series in each dataset is 365 and the length of representation methods is set to 32. As shown in this table, usually for a higher number of clusters the prediction error is reduced. Moreover, λ has lower values for the DFT representation and higher amounts for the DWT representation because of the magnitude of the features in these techniques.

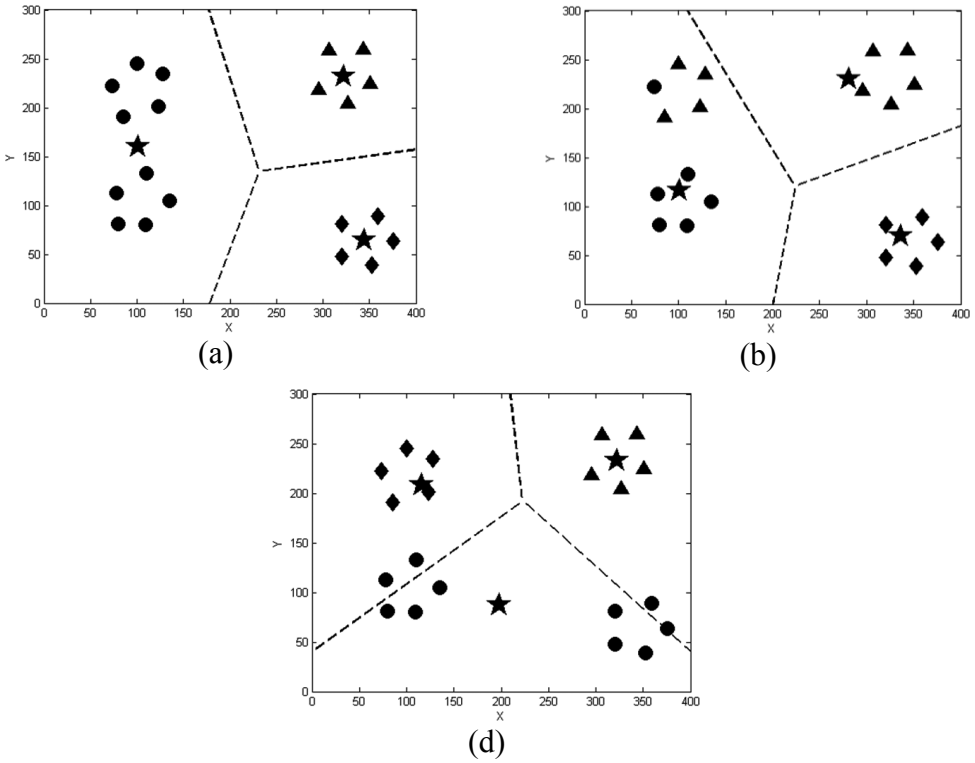


Figure 6.4. Clusters obtained for the less distinguishable dataset for c=3, DFT(16) and different values of λ: (a) λ=0, (c) optimal value of λ, and (c) λ=70.

Table 6.3. Prediction criterion for Alberta temperature dataset for 2009 to 2011. Each cell comprises two entries: the optimal value of λ, and the associated prediction error.

	c=2	c=4	c=6	c=8	c=10
2009					
DFT(32)	0.35, 157.39	0.8, 117.28	0.5, 101.5	0.07, 84.48	0.02, 76.92
PAA(32)	40, 199.72	10, 113.25	2.5, 103.71	2.5, 90.08	2, 80.38
DWT(32)	275, 131.13	125, 89.77	70, 77.2	5.5, 67.82	20, 61.18
2010					
DFT(32)	0, 170.51	1.5, 121.04	0.2, 96.9	0.09, 82.03	0.02, 69.26
PAA(32)	20, 189.84	15, 104.22	2, 97.87	3.5, 82.35	20, 73.32
DWT(32)	100, 126.53	125, 85.42	45, 74.87	20, 65.25	30, 60
2011					
DFT(32)	0.005, 190.34	0.65, 168.33	0.1, 154.17	0.45, 140.74	0.025, 133.3
PAA(32)	60, 205.67	15, 118.22	25, 105.68	2.5, 94.49	1, 80.26
DWT(32)	90, 134.45	175, 99.75	100, 88.87	10, 73.68	35, 64.54

The plots in Figure 6.5 illustrate the obtained clusters for c=4. The results are different depending upon the value of λ.

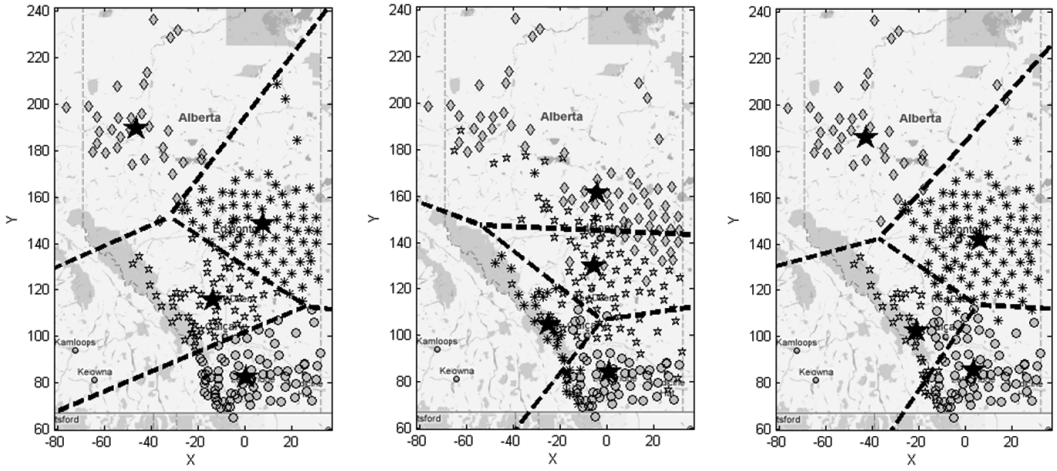


Figure 6.5. Plot of spatio-temporal clusters for 2009 for (a) λ=0, (b) λ=10,000, and (c) optimal value of λ. The number of clusters *c*=4 and DFT(32) used as the representation method.

The use of the optimal value of λ gives rise to clusters that form a sound balance between the effect of the spatial and temporal components in the clustering process.

6.4. Prediction abilities

In this section we focus on the prediction capabilities of the prediction criterion in dealing with spatial time series. Let us consider a part of the 2009 Alberta temperature dataset as the training samples $\boldsymbol{x}^{train}$, and the others as testing samples $\boldsymbol{x}^{test}$, and predict the temporal part of the testing samples based on their spatial coordinates. The procedure of this experiment is as follows:

1) Cluster the training samples using the augmented FCM and prediction criterion to find the optimal clusters (using optimal λ). The result is a set of spatio-temporal prototypes in the form of $\left\{\boldsymbol{v}^{train}(s) \mid \boldsymbol{v}^{train}(t)\right\}$.
2) Using the spatial part of the testing samples $\boldsymbol{x}^{test}(s)$, and the spatial part of

the calculated prototypes $\boldsymbol{v}^{train}(s)$, calculate the new partition matrix $\widetilde{U}$ using (6.4).

3) Predict the temporal part of the testing samples using $\widetilde{U}$ and the temporal part of the calculated prototypes $\boldsymbol{v}^{train}(t)$.

In this experiment we consider $N_{test} = 74$ (around 30%) stations of the 2009 Alberta temperature dataset as the testing samples and the other stations as training samples.

Table 6.4 shows the average prediction error for the testing set (called testing error), average prediction error for training set (training error) and an average error rate, for different representations and different number of clusters over 100 independent runs. The error rate is defined as:

$$E = \frac{testing\ error}{training\ error}. \tag{6.8}$$

In Table 6.4, with the increase of the number of clusters, both testing and training errors are reduced. This is quite reasonable since having more clusters means having more prototypes and more information about data, and as a result, the prediction can be more accurate. Moreover, because the clustering is performing on training samples, the defined error rate in (6.8) is always higher than 1 and by increasing the number of clusters the reduction in training error is higher than the reduction in testing error, so that the rate of testing error to training error is increased.

Figure 6.6(a) shows an example of selected stations as testing samples (star symbols) and the others as training samples. Three stations **a**, **b**, and **c** from testing samples have been labeled in this figure. Figure 6.6(b) shows the optimal clustering of the training samples. The optimal value of λ was equal to 0.65. In this figure two prototypes, P1 and P2 are labeled. The number of clusters was set to 5 and DFT(32) representation of time series was used.

Table 6.4. Average and standard deviation of testing error, training error, and error rate reported over 100 independent runs.

PAA(32)			
	Testing error	Training error	Error rate
c=2	0.834 ± 0.040	0.810 ± 0.009	1.030 ± 0.057
c=3	0.578 ± 0.106	0.519 ± 0.046	1.105 ± 0.181
c=4	0.505 ± 0.074	0.457 ± 0.028	1.107 ± 0.192
c=5	0.484 ± 0.060	0.430 ± 0.029	1.138 ± 0.220
DFT(32)			
	Testing error	Training error	Error rate
c=2	0.649 ± 0.020	0.639 ± 0.015	1.016 ± 0.049
c=3	0.567 ± 0.034	0.545 ± 0.014	1.042 ± 0.075
c=4	0.505 ± 0.033	0.471 ± 0.014	1.074 ± 0.091
c=5	0.465 ± 0.031	0.424 ± 0.013	1.099 ± 0.100
DWT(32)			
	Testing error	Training error	Error rate
c=2	0.544 ± 0.022	0.535 ± 0.011	1.019 ± 0.060
c=3	0.469 ± 0.022	0.453 ± 0.012	1.036 ± 0.072
c=4	0.391 ± 0.026	0.365 ± 0.013	1.075 ± 0.084
c=5	0.361 ± 0.024	0.332 ± 0.011	1.092 ± 0.095

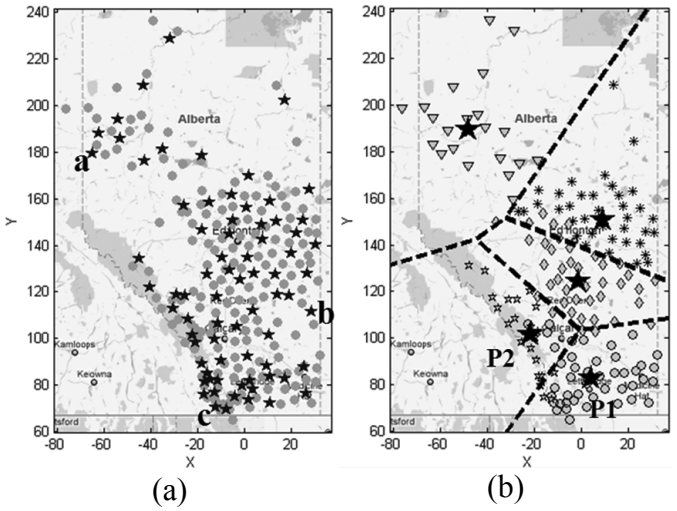


Figure 6.6. (a) The selected testing samples with three labeled stations **a**, **b** and **c** for prediction, (b) clusters of training samples with two labeled prototypes P1 and P2.

Figure 6.7 shows the reconstructed time series by the original features (32 DFT features) and predicted features. Using the prediction criterion, the temporal part

of the stations **a** and **b** has been predicted with a high accuracy. But the prediction for station **c** is not accurate. The reason is that, this station is between two clusters P1 and P2 (see Figure 6.6) with two very different temporal patterns. In fact, the spatial part of **c** is close to P1, but its temporal part is close to P2.

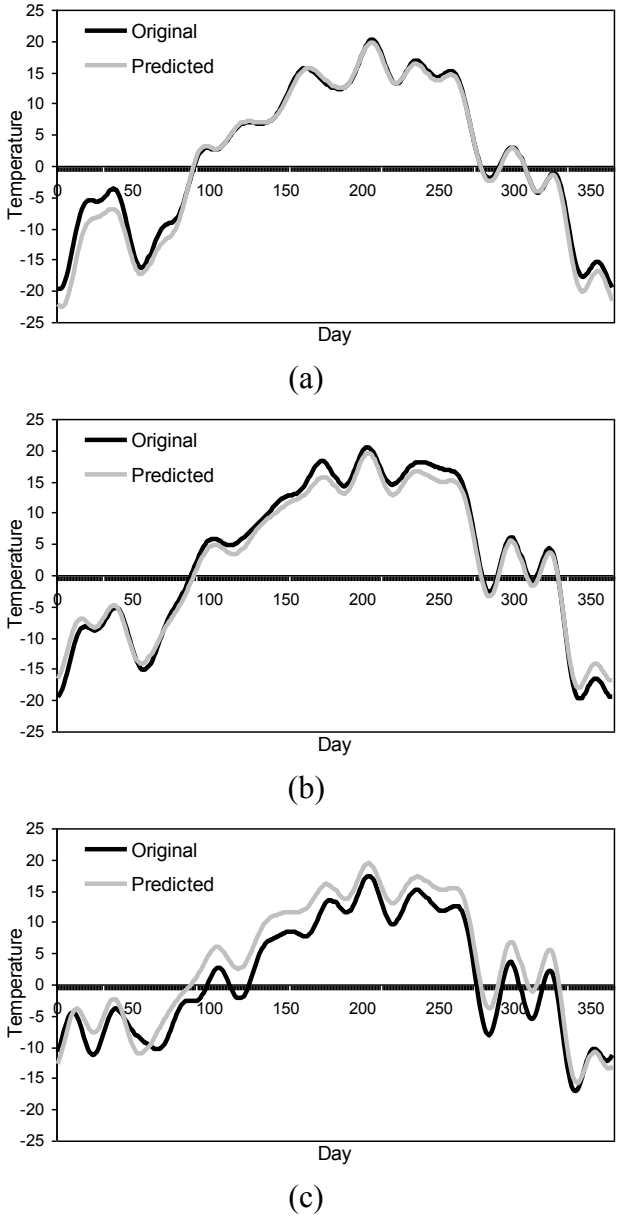


Figure 6.7. Original and predicted time series for (a) station **a**, (b) station **b**, and (c) station **c**.

Figure 6.8 shows the original and predicted time series of station **c** along with the time series corresponding to the prototypes P1 and P2. Both predicted and original time series of station **c** are almost between the time series corresponding to P1 and P2. P1 has more effect on prediction, because the spatial part of station **c** is closer to the spatial part of P1, and as a result P1 has a higher weight (in the form of membership degree $\widetilde{U}$) for prediction. One may consider more clusters to achieve more accurate prediction. For example, the prediction error for station **c** with number of clusters 2, 5, 8 and 12 is 1.283, 1.240, 0.684, and 0.511, respectively.

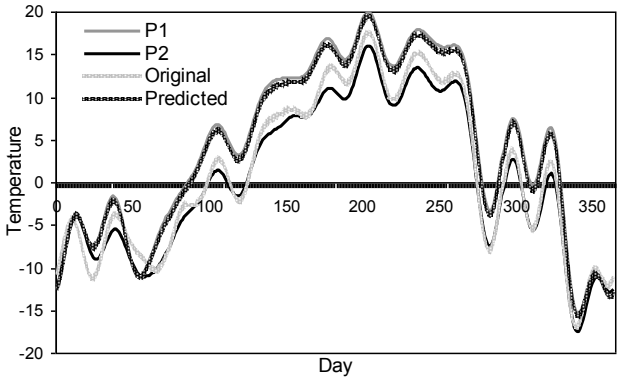


Figure 6.8. Original and predicted time series for station **c** (in Figure 6.6(a)) and the time series corresponding to the prototypes P1 and P2.

In the next step, we consider the entire data as training samples, and predict the temporal part of some unseen spatial coordinates in the map. The procedure is the same as used in the previous experiment. Figure 6.9 shows two generated spatial points **a** and **b** in the map. Also for each point, number of stations is selected as their neighbors. Figure 6.10(a) and 6.10(b) show the predicted time series for **a** and **b** along with the time series corresponding to their neighbors. As seen from these figures the predicted time series for points **a** and **b** are similar to their neighbors (time series).

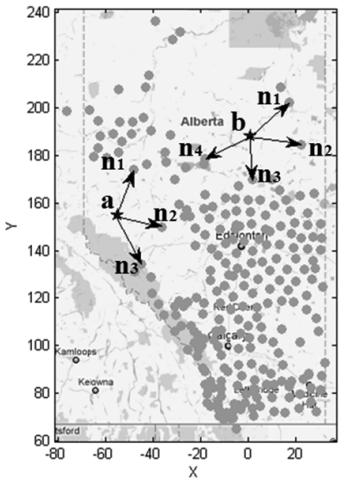
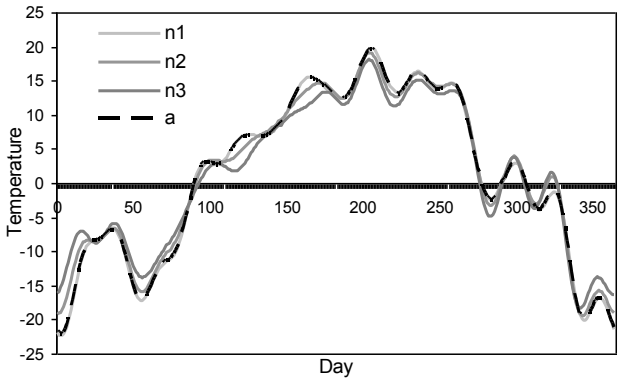
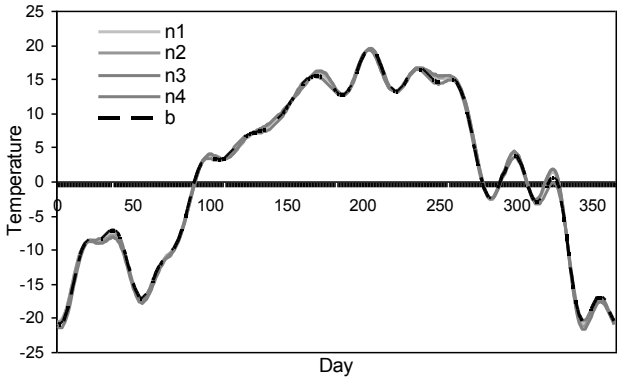


Figure 6.9. Generated two unseen spatial points **a** and **b** and their neighbors in the map.



(a)



(b)

Figure 6.10. Predicted time series and the time series corresponding to the neighbors of (a) station **a**, and (b) station **b** highlighted in Figure 6.9.

The prediction criterion that has been used in this study is different from the time series forecasting methods proposed in literature in both methodology and purpose. Our prediction criterion predicts the time series based on their spatial location and the time series formed in the cluster centers. Also in this method the objective is finding an optimal tradeoff to regulate the interaction between spatial and temporal patterns in the clustering process and not forecasting the time series for the future time steps. Time series forecasting methods proposed in literature (e.g. [33, 89, 90]) usually assume that the times series follow a linear or nonlinear model and try to find the parameters of the corresponding model using historical data. Then the generated model is used to forecast the time series in the future.

6.5. Comparative studies

Pham [43] proposed a spatial model of FCM (called RFCM), for image segmentation. This method uses a spatial penalty on membership degrees. The proposed objective function is as follows:

$$V = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m} \left\| \boldsymbol{v}_i(t) - \boldsymbol{x}_k(t) \right\|^2 + \frac{\beta}{2} \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m} \sum_{c' \in M_i} \sum_{j \in N_k} u_{c'j}^{m}, \tag{6.9}$$

where N_k denotes the neighbors of station k, and $M_i = \{1,2,...,c\} - \{i\}$. (6.9) is composed of two parts: the FCM objective function for temporal part of data and a spatial regularization term. β is a weight to control the effect of each part in clustering (like λ in our method). The above objective function can be minimized by calculating partition matrix and prototypes in an iterative process. Let us assume the kth object has a high membership degree to ith cluster. Minimizing (6.9) leads to the reduction of the membership degrees of objects in N_k to the cluster centers in M_i. Coppi et al. [44] extended this method to cluster spatial time series. To compare our method (the reconstruction and prediction criteria) with the RFCM, we propose the following evaluation criterion

$$Q = \frac{J(\boldsymbol{x}(s) \mid U)}{J(\boldsymbol{x}(s))} + \frac{J(\boldsymbol{x}(t) \mid U)}{J(\boldsymbol{x}(t))}, \tag{6.10}$$

where U is the optimal partition matrix in spatio-temporal clustering (resulted from optimal λ in our methods and optimal β in RFCM). $J(\boldsymbol{x}(s) \mid U)$ is the FCM objective function for spatial part of data by considering U as its partition matrix and calculating new prototypes. $J(\boldsymbol{x}(s))$ is the FCM objective function resulting from clustering spatial part of data separately. Also $\boldsymbol{x}(t)$ denotes the temporal part of data. In fact $J(\boldsymbol{x}(s))$ and $J(\boldsymbol{x}(t))$ are two normalization terms. The intuition behind the proposed criterion is that we consider a clustering as an "appropriate" clustering, if it is suitable for both spatial part and temporal part of data. The lower value of Q, the spatio-temporal clusters are more appropriate. Notice that since in clustering spatial (or temporal) part of data separately, we do not consider the other part, the resulting partition matrix will be the optimal one for that part and obviously we will have: $J(\boldsymbol{x}(s) \mid U) \geq J(\boldsymbol{x}(s))$ and $J(\boldsymbol{x}(t) \mid U) \geq J(\boldsymbol{x}(t))$ and as a result, always in (6.10) we have: $Q \geq 2$. We calculated Q for reconstruction criterion, prediction criterion and RFCM. In RFCM to find the optimal value of β a heuristic can be used. In [43] and [44] different values of β in a range is checked to optimize an objective function. This objective function is minimizing a cross validation error in [43] and maximizing a spatial autocorrelation in [44]. Since the evaluation criterion in this comparison is Q in (6.10), we check different values of β and select the one that can minimize it. Table 6.5 shows the comparison for different representations and different number of clusters for Alberta temperature data in 2009.

As can be seen from this table, in most of cases, reconstruction and prediction criteria have a lower value of Q. The reason is that these methods consider the same importance for each part of data in clustering, while RFCM pays less attention to the spatial part. In fact, in RFCM the spatial part of data has been used for smoothing the temporal clusters (like spatial smoothing of pixels in image processing). Also we can see that for different representation methods there are different amounts of Q. The reason is that each representation method captures a

special kind of features, and based on these features the temporal structures are different.

Table 6.5. Comparison of reconstruction criterion (RC), prediction criterion (PC) and RFCM over the evaluation criteria (6.10) for different representations and number of clusters.

	DFT(32)			PAA(32)			DWT(32)		
	RC	PC	RFCM	RC	PC	RFCM	RC	PC	RFCM
c=2	2.347	2.179	2.296	2.304	2.219	2.314	2.277	2.151	2.276
c=3	2.894	2.894	2.91	2.664	2.813	2.919	2.306	2.301	2.762
c=4	2.464	2.454	3.144	2.425	2.433	3.124	2.309	2.305	2.904
c=5	2.437	2.44	3.233	2.438	2.426	3.214	2.296	2.307	2.622

6.6. Summary

In this chapter, a prediction criterion for evaluating spatial time series clusters is employed. The essence of this criterion is to predict the temporal part of data using their spatial information. Similar to the previous chapter, a composite distance function has been considered in the FCM objective function to control the effect of each part of data in the clustering process. The prediction error has been employed to hit a sound balance between the effect of spatial part and time series part of data. The proposed technique has been investigated over a synthetic and the Alberta temperature data. Experimental studies show that using the prediction criterion, one may predict the time series part of data with the use of the spatial coordinates with a high accuracy. Furthermore, the proposed prediction and reconstruction criteria are compared to an existing clustering technique reported in the literature. Experimental results indicate the efficiency of the proposed criteria in this study for clustering spatial time series data.

7. Clustering Spatial Time Series Using an Agreement Criterion

In previous chapters we introduced using a reconstruction and a prediction criterion in clustering spatial univariate time series. In this chapter, we propose a technique that is suitable for clustering both spatial univariate and spatial multivariate time series. For this purpose, clustering spatial time series is considered as clustering data with blocks of features coming from distinct data sources. In this point of view, the spatial and each time series part of data (especially in multivariate time series) form a separate block of features coming from a distinct data source. Figure 7.1 visualizes the essence of the problem in which we aim at clustering objects with features coming from distinct data sources.

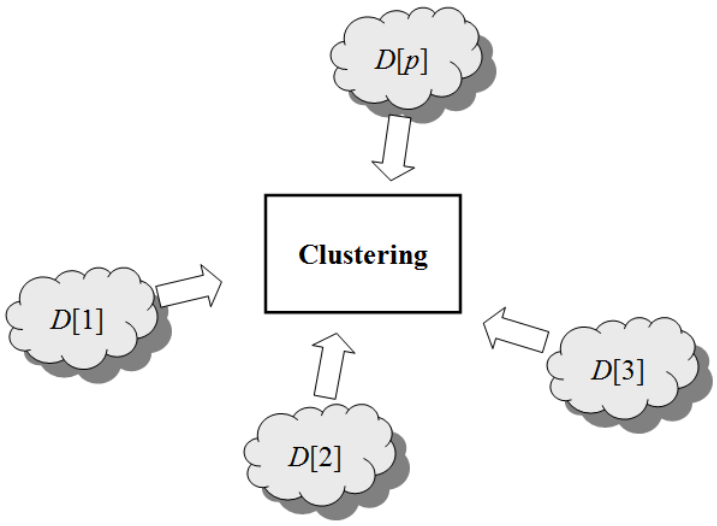


Figure 7.1. The essence of the agreement-based clustering.

We investigate the use of an augmented distance function in which distances computed for the individual blocks of features are aggregated (concatenated) by means of some weights. These weights are used to control the impact coming from each block of features to the clustering process. A significant problem here

is selecting a suitable criterion to optimize the impact of each part of data (blocks of features) in the clustering process. One of the alternatives sought here comes in the form of agreement-based clustering where clustering is intended to form a structure while achieving a significant level of structural "agreement" among all blocks.

7.1. Problem formulation

Let us consider N objects $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$ whose features are coming from p data sources (blocks) $D[1]$, $D[2]$,…,$D[p]$. Each data source describes the objects from a different point of view. By concatenating these features, each object is described with the use of $\boldsymbol{x}_k = \left[\boldsymbol{x}_k(1) | \boldsymbol{x}_k(2) | ... | \boldsymbol{x}_k(p)\right]^{\mathrm{T}}$, k=1, 2,…, N, where $\boldsymbol{x}_k(j)$ is the feature vector corresponding to jth data source, $D[j]$, for kth object. Since in each data source like $D[j]$ there are r_j features, altogether we have the following representation:

$$\boldsymbol{x}_k = \left[x_{k1}(1), x_{k2}(1), \cdots, x_{kr_1}(1) | ... | x_{k1}(p), x_{k2}(p), \cdots, x_{kr_p}(p)\right]^{\mathrm{T}}. \tag{7.1}$$

Note that the number of features in different data sources can be different. We propose a Fuzzy C-Means clustering for this type of data. To deal with the data structure represented in (7.1), we define the following distance function between object $\boldsymbol{x}_k$ and prototype $\boldsymbol{v}_i$

$$\begin{aligned} d^2_{\lambda_{1,...,p}}(\boldsymbol{v}_i, \boldsymbol{x}_k) = \lambda_1 \left\|\boldsymbol{v}_i(1) - \boldsymbol{x}_k(1)\right\|^2 + \lambda_2 \left\|\boldsymbol{v}_i(2) - \boldsymbol{x}_k(2)\right\|^2 + \\ ... + \lambda_p \left\|\boldsymbol{v}_i(p) - \boldsymbol{x}_k(p)\right\|^2, \quad \sum_{j=1}^{p} \lambda_j = 1, \quad 0 \le \lambda_j \le 1 \end{aligned}. \tag{7.2}$$

Using the distance function specified above, the impact of each data source in the clustering process can be easily controlled. Assigning $\lambda_j = 0$, removes the contribution of data source $D[j]$ to the overall clustering process, while $\lambda_j = 1$, removes the contribution of other data sources and considers only $D[j]$ in the clustering process. Higher values of λ_j increase the impact of $D[j]$ and decrease

the impact of the other data sources in the clustering process. Considering (7.2) as the distance function, the FCM objective function is expressed as

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} d_{\lambda_{1,\ldots,p}}^{2}(\boldsymbol{v}_i, \boldsymbol{x}_k). \tag{7.3}$$

The minimization of J is realized through an iterative process in which we successively compute the prototypes and the partition matrix in the form:

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m} \boldsymbol{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}}, \tag{7.4}$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{\lambda_{1,\ldots,p}}^{2}(\boldsymbol{v}_i, \boldsymbol{x}_k)}{d_{\lambda_{1,\ldots,p}}^{2}(\boldsymbol{v}_j, \boldsymbol{x}_k)} \right)^{1/(m-1)}}. \tag{7.5}$$

7.2. Agreement criterion

In previous section, we described a fuzzy clustering approach to deal with data with blocks of features coming from distinct sources. As the weights $\lambda_1, \lambda_2, \ldots, \lambda_p$ in the introduced distance function control the effect (impact) of each data source in the clustering process, the quality of clusters directly depends on them. In this section, we propose an evaluation criterion to optimize these embedded weights.

Since our objective is to reveal a general structure over all data sources, this structure should have a high level of "agreement" among the available structures in separate data sources. To measure the level of agreement, the FCM objective function has been considered. Assuming that U is the partition matrix resulting from clustering objects (with blocks of features) using the proposed distance function in (7.2), the quality of the clusters can be quantified using the following evaluation criterion

$$Q = \frac{J(D[1]|U)}{J(D[1])} + \frac{J(D[2]|U)}{J(D[2])} + \ldots + \frac{J(D[p]|U)}{J(D[p])}, \tag{7.6}$$

where $J(D[j]|U)$ is the value of the FCM objective function for data source $D[j]$ by considering U as its partition matrix and calculating its prototypes. $J(D[j])$ stands for the FCM objective function obtained when clustering data source $D[j]$ separately. In fact, (7.6) expresses how much the revealed general structure, U, is suitable (acceptable) for each separate data source in terms of the corresponding FCM objective function. Because the feature spaces for distinct data sources exhibit various magnitudes and dimensionalities, $J(D[j])$, j=1,2,…, p used as denominator in (7.6) serves as a normalization term. Moreover, since in clustering each data source separately, the other sources are not taken into account, the resulting partition matrix is the optimal one for this particular data source and obviously $J(D[j]|U) \geq J(D[j])$, and as the result the inequality $Q \geq p$ always holds. In the case $Q = p$, the available structures determined for distinct data sources are in a perfect agreement and the resulting structure by the proposed method is exactly the same as the available structures in various data sources. Lower value of Q indicates that the formed general structure is at a higher level of agreement with distinct data sources, while higher value of Q is indicative of a lower level of agreement. Therefore, the problem of finding optimal weights $\lambda_1, \lambda_2, ..., \lambda_p$ can be considered as an optimization problem: determining the values of $\lambda_1, \lambda_2, ..., \lambda_p$ in order to minimize Q. Since checking all the possible combinations of values of the weights is time consuming (especially for higher number of data sources), using a meta-heuristic algorithm to find near-optimal weights could be a viable alternative. There are numerous works reported in the literature exploiting the merits of evolutionary algorithms in clustering. In this study, a Particle Swarm Optimization (PSO) is used as an efficient population based searching algorithm to find (near) optimal weights.

7.3. Particle Swarm Optimization (PSO) as a searching algorithm

PSO [113] is a population based optimization technique inspired by bird flocking and fish schooling. It starts with a number of potential solutions (called particles)

and in some iteration tries to improve the quality of particles using some searching strategies. Since the problem search space here is a *p*-dimensional vector with elements in range [0, 1], each particle is encoded as a vector with *p* elements $\lambda_1, \lambda_2, ..., \lambda_p$ following the constraints imposed in (7.2). In the first step of the algorithm, number of particles and their corresponding velocity vectors (with *p* elements in a pre-specified range) are generated randomly. For each particle, the encoded weights are used to produce a general structure over all data sources, and the proposed evaluation criterion in (7.6) is considered as the quality (fitness) of that particle. In each iteration of the algorithm, the velocity vectors and the particles are updated using (7.7) and (7.8) respectively.

$$y_{ki}^{t+1} = w.y_{ki}^{t} + c_1 r_{1i}(pbest_{ki}^{t} - z_{ki}^{t}) + c_2 r_{2i}(gbest_{ki}^{t} - z_{ki}^{t}), \quad y_{ki}^{t} \in [y_{\min}, y_{\max}] \tag{7.7}$$

$$z_{ki}^{t+1} = z_{ki}^{t} + y_{ki}^{t+1}, z_{ki}^{t} \in [0,1] \tag{7.8}$$

where, k=1, 2,…, n, i= 1, 2, …, p, y_{ki}^{t} is the ith element of the velocity of the kth particle in tth step, z_{ki}^{t} is the ith element of the kth particle in tth step of the algorithm, n is the number of particles in the swarm and p is the dimensionality of the search space (number of data sources here). Also *pbest* (personal best) is the best solution the particle has revealed and *gbest* (global best) is the best solution the whole swarm has obtained during the search process, w is inertia weight, r_{1i} and r_{2i} are random values in range [0, 1] sampled from a uniform distribution and c_1 and c_2 are acceleration coefficients, controlling the impact of *pbest* and *gbest* on the search process. The algorithm improves the quality of solutions in a number of iterations and finally, the best particle (with the best fitness value) is considered as the (near) optimal weights. Figure 7.2 shows the overall scheme of the algorithm. At the first step of the algorithm, PSO generates a set of particles each comprising p weights, $\lambda_1, \lambda_2, ..., \lambda_p$. For each particle, these weights are exploited to cluster the objects with distinct data sources (using the composite distance function in (7.2)) and the quality of clusters is evaluated using the

proposed criterion in (7.6) which serves as the fitness function of the PSO. In the next step, PSO manipulates the generated particles using calculated fitness values to improve their quality.

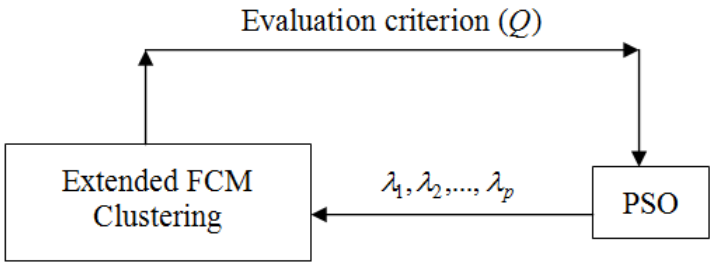


Figure 7.2. The overall scheme of the proposed agreement-based clustering.

7.4. Experimental studies

In this section, we illustrate the proposed method by using a synthetic dataset, and the Alberta climate data.

7.4.1. Synthetic data

For illustrative purposes and in order to clarify the performance of the proposed evaluation criterion (7.6), we generated five data sources and investigated the behavior of the proposed method. Figure 7.3(a)-(e) show the data sources $D[1]$ to $D[5]$.

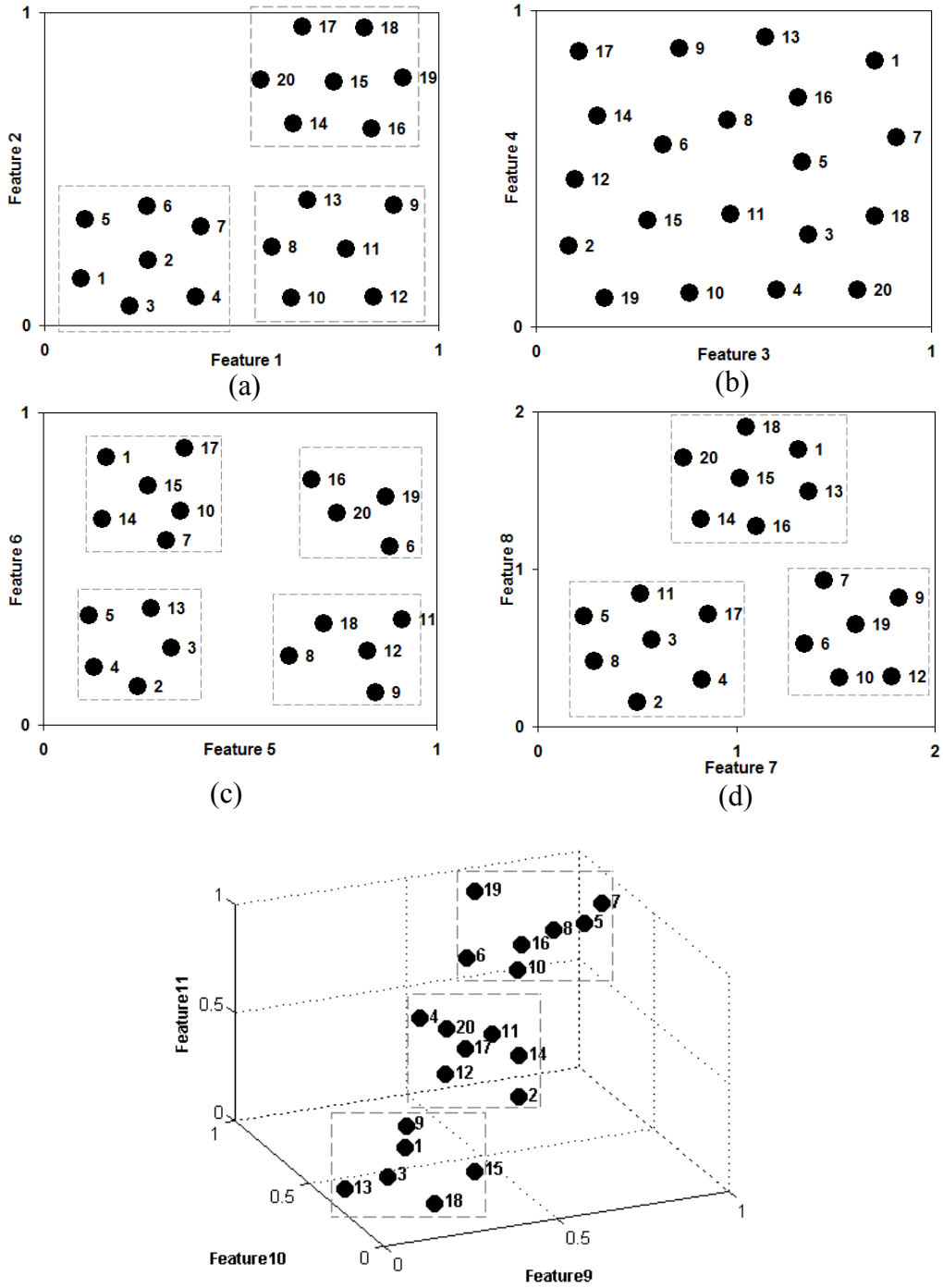


Figure 7.3. Five synthetic data sources. (a) $D[1]$, (b) $D[2]$, (c) $D[3]$, (d) $D[4]$, and (e) $D[5]$.

As shown in these figures, each object is composed of 11 features that are associated with five data sources with the following geometries:

- D[1] is a two-dimensional data with features in range [0, 1] and has a visible structure for three clusters.
- D[2] is a two-dimensional data with features in range [0, 1], but there is no a visible structure in this data source .
- D[3] is a two-dimensional data with features in range [0, 1] and has a visible structure for four clusters.
- D[4] is a two-dimensional data with features in range [0, 2] and has a visible structure for three clusters.
- D[5] is a three-dimensional data with features in range [0, 1] and has a visible structure for three clusters.

Strong (more distinguishable) structure versus weak (less distinguishable) structure: In this experiment, we consider two data sources D[1] and D[2] and investigate the effect of the values of λ_1 and λ_2 on the evaluation criterion (Q) to form a general structure for number of clusters c=3. The fuzzification coefficient was set to m=2 in all experiments. Figure 7.4 shows the values of Q versus different values of λ_1. Because of having two data sources in this experiment, we have $\lambda_2 = 1 - \lambda_1$.

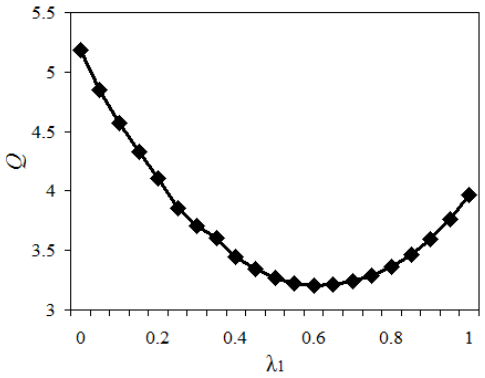


Figure 7.4. Evaluation criterion (Q) versus different values of λ_1 in the formation of the general structure over D[1] and D[2].

As can be seen from this figure, the optimal weights are λ_1=0.6 and λ_2=0.4, that means $D[1]$ has higher impact on forming globally acceptable clusters. The prototypes for these data sources before forming the general structure are as follows:

$$\begin{array}{ll} \boldsymbol{v}_1[1]=[0.717 \quad 0.243] & \boldsymbol{v}_1[2]=[0.699 \quad 0.277] \\ \boldsymbol{v}_2[1]=[0.734 \quad 0.798] \text{ and } & \boldsymbol{v}_2[2]=[0.563 \quad 0.768] \\ \boldsymbol{v}_3[1]=[0.236 \quad 0.219] & \boldsymbol{v}_3[2]=[0.201 \quad 0.376] \end{array}$$

Once the overall general structure has been formed, the updated prototypes are changed to:

$$\begin{array}{ll} \boldsymbol{v}_1[1]=[0.668 \quad 0.323] & \boldsymbol{v}_1[2]=[0.621 \quad 0.444] \\ \boldsymbol{v}_2[1]=[0.732 \quad 0.752] \text{ and } & \boldsymbol{v}_2[2]=[0.432 \quad 0.578] \\ \boldsymbol{v}_3[1]=[0.279 \quad 0.231] & \boldsymbol{v}_3[2]=[0.392 \quad 0.419] \end{array}$$

As it can be seen, the prototypes corresponding to data source $D[2]$ exhibit more changes in comparison with the prototypes describing $D[1]$. Since $D[1]$ has a more visible structure, its FCM objective function $J(D[1])$, has lower value in comparison with $D[2]$ and as a result the algorithm pays more attention to $D[1]$ to achieve lower values for $J(D[1]|U)/J(D[1])$. Also one may note that the situation where $\lambda_1 = 0$ is the worst case in this experiment because of the existing a stronger structure in $D[1]$.

Let us consider $D[1]$ and $D[3]$ and form the general structure for the number of clusters set to c=3 and c=4. Figure 7.5(a) and 7.5(b) show the effect of λ_1 and $\lambda_3 = 1 - \lambda_1$ on the evaluation criterion. The optimal value of λ_1 is 0.55 for c=3 (Figure 7.5(a)) and 0.45 for c=4 (Figure 7.5(b)).

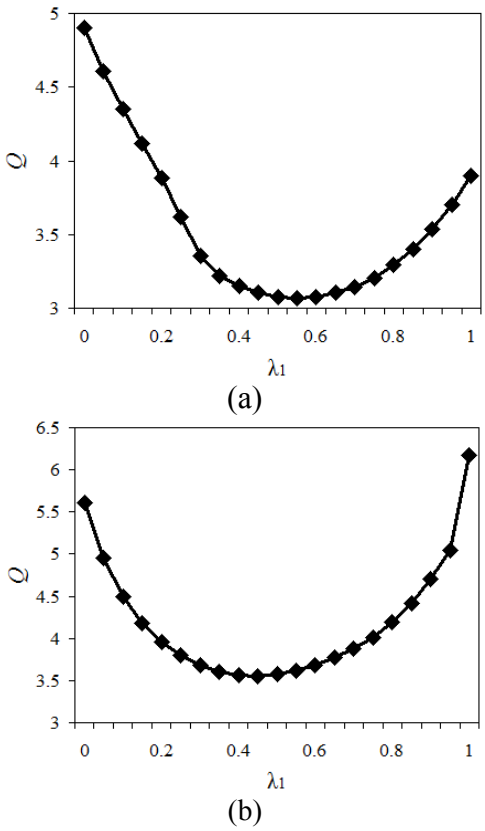


Figure 7.5. Evaluation criterion (Q) versus different values of λ_1 in forming general structure over $D[1]$ and $D[3]$. (a) c=3, and (b) c=4.

For c=3, $D[1]$ has a stronger (more visible) structure and we have $\lambda_1 > \lambda_3$, while for c=4, as $D[3]$ has more visible structure, we have $\lambda_1 < \lambda_3$.

Data sources with different magnitudes of features: Let us consider $D[1]$ and $D[4]$. Both of data sources have a visible structure for c=3. The magnitude of features in $D[1]$ is in range [0, 1], while for $D[4]$ it is in range [0, 2]. Figure 7.6 shows Q for different values of λ_1. Similar to the previous experiments, we have $\lambda_4 = 1 - \lambda_1$.

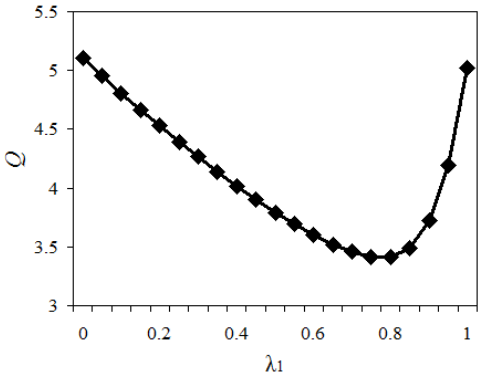


Figure 7.6. Q versus different values of λ_1 in forming general structure over $D[1]$ and $D[4]$ for c=3.

The optimal value of Q (see Figure 7.6) occurred around $\lambda_1 = 0.75$ and $\lambda_4 = 0.25$. The reason is that the magnitude of features in $D[1]$ is lower than the magnitude of features in $D[4]$ and the algorithm assigns a higher value to λ_1 to prevent bias towards $D[4]$ in the formation of the general structure.

Data sources with different number of features: In this experiment we consider $D[1]$ and $D[5]$. $D[5]$ has three features and a visible structure for c=3. Figure 7.7 shows the values of Q for different values of λ_1. The optimal Q occurs for higher value of λ_1 ($\lambda_1 > \lambda_5$). The reason is the same as in the previous experiment: considering higher value for λ_1 in order to prevent bias towards $D[5]$ in the clustering process.

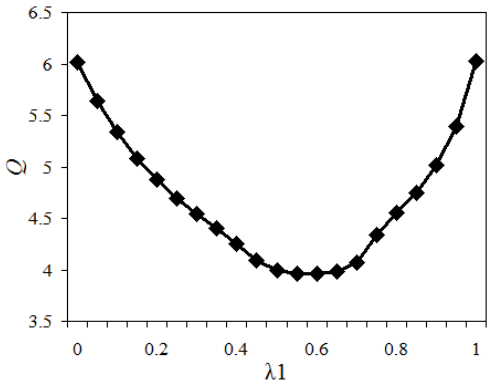


Figure 7.7. Q versus different values of λ_1 in forming general structure over $D[1]$ and $D[5]$ for c=3.

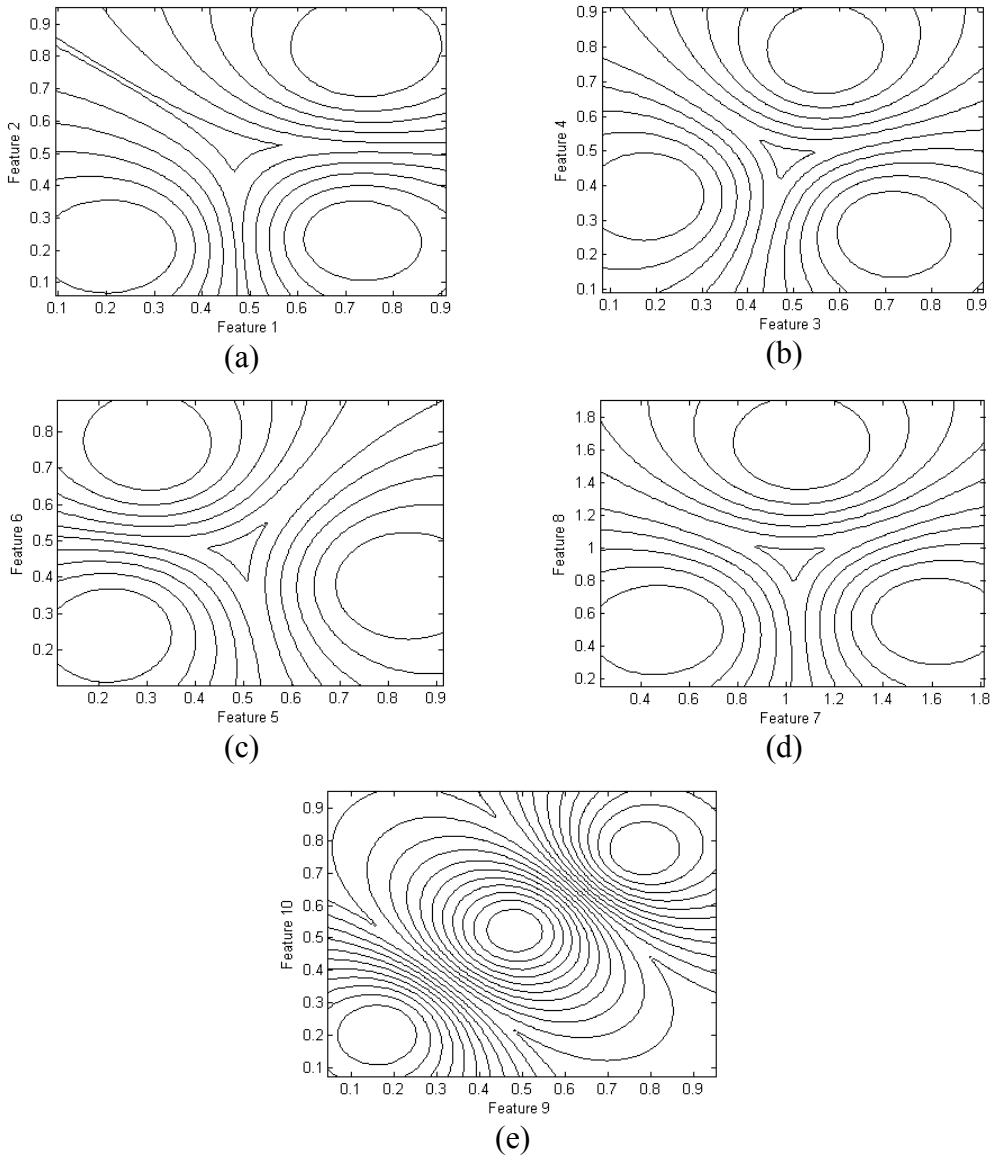


Figure 7.8. Contour plot of membership degrees before forming the general structure. (a) $D[1]$, (b) $D[2]$, (c) $D[3]$, (d) $D[4]$ and (e) $D[5]$.

Forming general structure for $D[1]$ *to* $D[5]$: In this experiment we consider all data sources to form a general structure for number of clusters c=3 and c=4. For the PSO algorithm the following parameters after a fine-tuning has been chosen: number of particles n=5 (equal to the number of data sources), $c_1 = c_2 = 2$, number of iterations= $10n = 50$, range of velocity elements=[-0.3, +0.3]. For the

number of clusters c=3, the optimal weights are as follows $[\lambda_1 = 0.288,\ \lambda_2 = 0.184,\ \lambda_3 = 0.235,\ \lambda_4 = 0.087,\ \lambda_5 = 0.209]$,

and for c=4, the optimal weights are

$[\lambda_1 = 0.252,\ \lambda_2 = 0.185,\ \lambda_3 = 0.307,\ \lambda_4 = 0.072,\ \lambda_5 = 0.184]$.

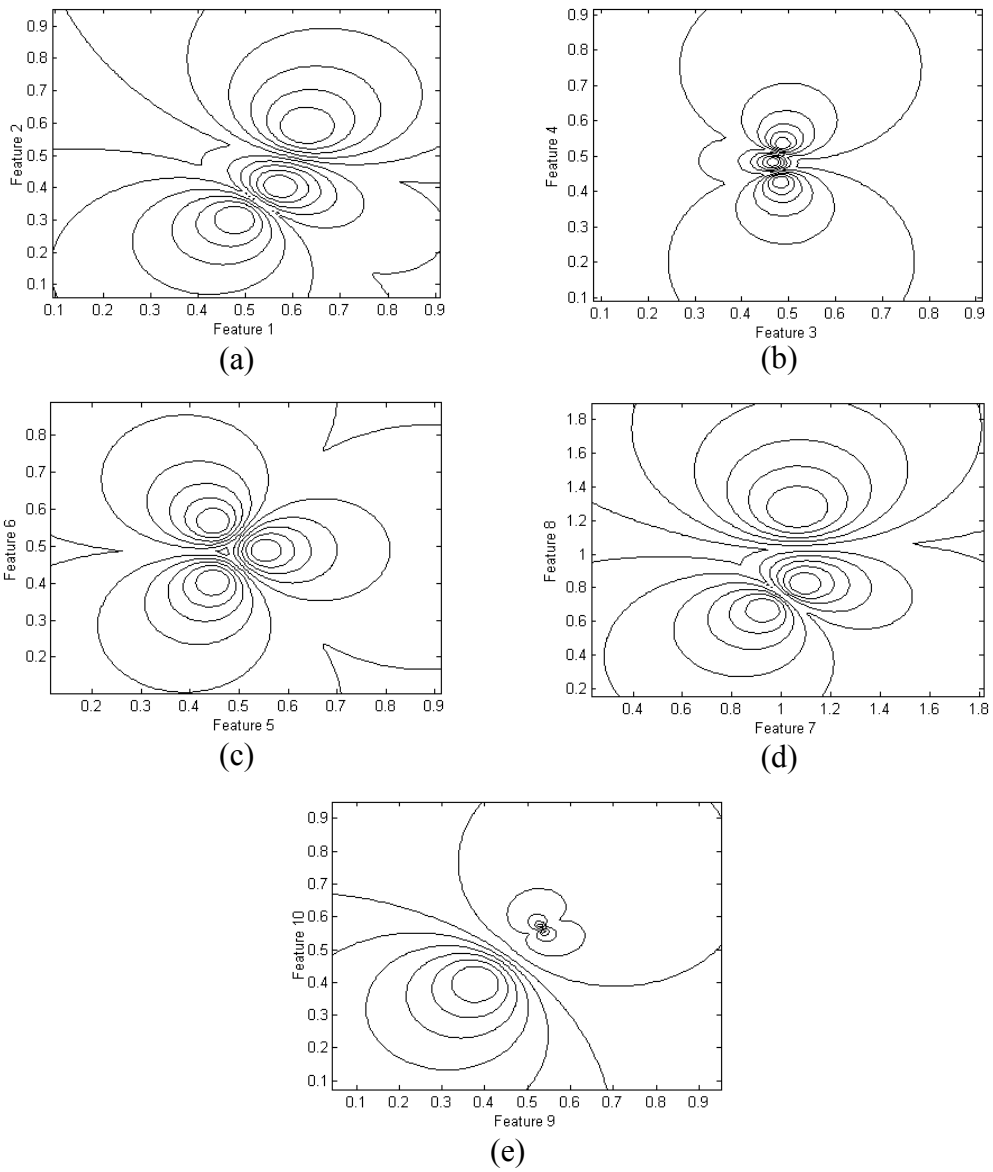


Figure 7.9. Contour plot of membership degrees after forming general structure. (a) D[1], (b) D[2], (c) D[3], (d) D[4] and (e) D[5].

Overall, $D[1]$ and $D[3]$ have higher weights in comparison with the weights associated with other data sources. $D[2]$, $D[4]$ and $D[5]$ have lower weights because of their weak structure, higher range of features and higher dimensionality, respectively. Also for c=3, $\lambda_1 > \lambda_3$, while for c=4 $\lambda_1 < \lambda_3$ because of the existing structures in these data sources. Figure 7.8(a)-(e) shows the clusters in each data source (visualized in the form of the contour plot of membership degrees) for c=3 and Figure 7.9(a)-(e) shows the clusters after forming general structure. For $D[5]$ the clusters are plotted over the first two features.

Obviously, there is a significant change in the initial clusters after forming the general structure. Also Figure 7.10 shows the PSO convergence process for c=3 and c=4. The most significant improvements have been observed in the first few generations. Moreover, for c=4, Q has a higher value than c=3. The reason is that having more clusters means more details about the available structures in the separate data sources. As the result, the level of agreement between data sources is decreased.

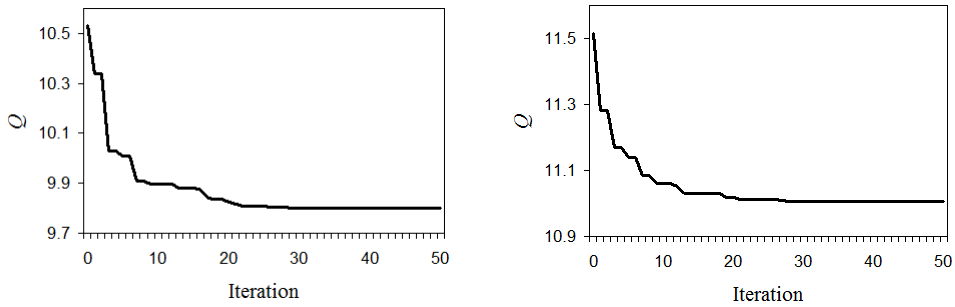


Figure 7.10. Convergence of PSO optimization process for (a) c=3 and (b) c=4.

7.4.2. Alberta climate data

The dataset used in this experiment is composed of 173 stations located in Alberta. For each station, its spatial coordinates, and the recorded daily average temperature, daily precipitation, and daily average humidity in the form of time

series have been provided. These data are available online at www.agric.gov.ab.ca. Figure 7.11 shows a snapshot of the system with one highlighted station along with its temperature, precipitation, and humidity time series in 2010.

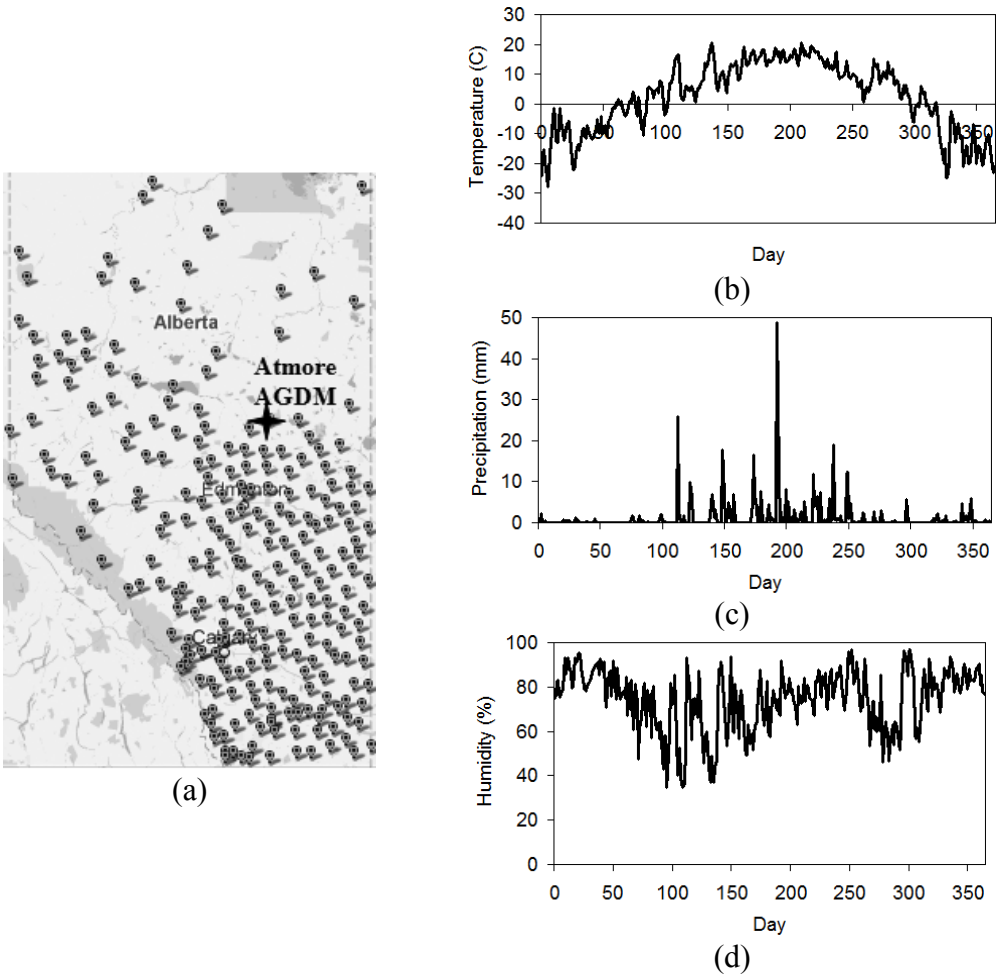


Figure 7.11. A snapshot of the Alberta agriculture system. (a) A set of stations in Alberta along with one highlighted station, (b) temperature time series corresponding to the highlighted station in 2010, (c) precipitation time series, and (d) humidity time series

Since in this dataset, for each station there are four sources of data (spatial coordinates, temperature, precipitation, and humidity) we use the proposed method to form some general structures over all data sources. DFT and PAA representations of time series with length 8, 16 and 24, and number of clusters 2,

3, 4, and 5 are considered. Note that the length of time series representation is application-dependent and higher length of representation includes more details about time series, while lower length of representation hides details. For the PSO algorithm the number of particles is set equal to the number of data sources (n=4) and the number of iterations is 10n.

To assess the effectiveness of the proposed method we compared it with three following scenarios:

1) In the first scenario, for each separate data source we calculate its partition matrix using FCM and then consider the following criterion to evaluate the average level of agreement among the available structures revealed in separate data sources:

$$Q_{\text{avg}} = \frac{1}{p}\sum_{i=1}^{p}\sum_{j=1}^{p}\frac{J(D[j]\,|\,U[i])}{J(D[j])}, \tag{7.9}$$

where $U[i]$ is the partition matrix calculated using FCM for data source $D[i]$, $J(D[j]\,|\,U[i])$ is the FCM objective function for $D[j]$ by considering $U[i]$ as its partition matrix, and $J(D[j])$ is FCM objective function for the separate data source $D[j]$.

2) In the second scenario, we consider the data source with the highest level of agreement with other data sources and evaluate the level of its agreement using the following criterion:

$$Q_{\min} = \min\sum_{j=1}^{p}\frac{J(D[j]\,|\,U[i])}{J(D[j])}, \text{ for } i=1,2,\ldots,p, \tag{7.10}$$

3) Finally, in the third scenario, we consider "standard" FCM (all the weights in (7.2) are set to be equal to 1) to cluster all data sources and use (7.6) to evaluate the level of agreement among data sources (Q_{FCM}).

Table 7.1 compares the results in terms of the average level of agreements among separate data sources (Q_{avg}), highest available level of agreement among separate

data sources ($Q_{\min}$), level of agreement achieved by FCM (Q_{FCM}), and the level of agreement achieved by optimal weights (Q). For the last one, the results are reported as the average and standard deviation in 40 independent runs.

Table 7.1. Experimental results for Alberta climate data in 2010. DFT and PAA representations with length 8, 16 and 24, and number of clusters c=2, 3, 4, and 5 have been considered. For the optimal weights, the results are reported in the form of average and standard deviation of Q in 40 independent runs.

c	Representation	Q_{avg}	$Q_{\min}$	Q_{FCM}	Q
2	DFT(8)	5.485	5.184	5.23	4.716±0.034
	DFT(16)	4.952	4.712	4.945	4.504±0.017
	DFT(24)	4.909	4.789	4.834	4.479±0.047
	PAA(8)	5.231	5.079	4.849	4.562±0.013
	PAA(16)	4.835	4.768	4.634	4.41± 0.012
	PAA(24)	4.825	4.718	4.626	4.438±0.045
3	DFT(8)	6.7	6.363	6.208	5.213±0.124
	DFT(16)	5.656	5.324	5.619	4.837±0.074
	DFT(24)	5.496	5.211	5.121	4.793±0.071
	PAA(8)	6.299	6.01	5.662	4.925±0.042
	PAA(16)	5.481	5.41	4.995	4.672±0.027
	PAA(24)	5.488	5.113	5.007	4.691±0.018
4	DFT(8)	7.689	6.839	6.929	5.575±0.202
	DFT(16)	6.091	5.562	5.591	4.946±0.074
	DFT(24)	5.828	5.382	5.424	4.842±0.074
	PAA(8)	6.892	6.332	5.67	5.08± 0.033
	PAA(16)	5.792	5.521	5.258	4.765±0.029
	PAA(24)	5.73	5.398	5.221	4.773±0.02
5	DFT(8)	7.992	7.682	6.79	5.684±0.168
	DFT(16)	6.353	5.683	5.774	5.071±0.064
	DFT(24)	6.054	5.505	5.549	4.983±0.056
	PAA(8)	7.329	6.733	5.853	5.184±0.047
	PAA(16)	5.98	5.663	5.344	4.857±0.033
	PAA(24)	5.955	5.5	5.282	4.871±0.075

As shown in this table, the proposed method can produce the structures with a higher level of agreement among all data sources. In most cases, increasing the number of clusters (granularity) increases the value of Q (which means reduces the level of agreement). In fact, by increasing the number of clusters, more details

about the available structures in each separate data source is considered and as the result the level of agreement between structures in different data sources is decreased. On the other hand, by increasing the length of time series representation, the clusters are built with higher level of agreement because by increasing the length of representation of time series, the degree of overlap between clusters and as a result the FCM objective function (that has been used in (7.6) as denominator) is increased and the value of Q is decreased. Furthermore, different parameters (e.g. number of clusters, type and length of representation, etc.) have various impacts on the available structures in each data source and affect the level of agreements achieved by the proposed method over data sources. Figure 7.12(a)-(d) shows the clusters revealed for the Alberta climate dataset for different data sources separately, Figure 7.12(e) shows the clusters revealed by FCM over all data sources, and Figure 7.12(f) shows the clusters produced by the proposed method (optimal weights) over all data sources. Number of clusters is c=2 and the DFT(24) representation of the time series is considered. Moreover, Figure 7.13 shows the clusters for Alberta climate data for c=3 and PAA(24) representation. In fact, in both Figures 7.12(f) and 7.13(f) the revealed clusters are the ones that have the highest agreement with the available structures in distinct data sources.

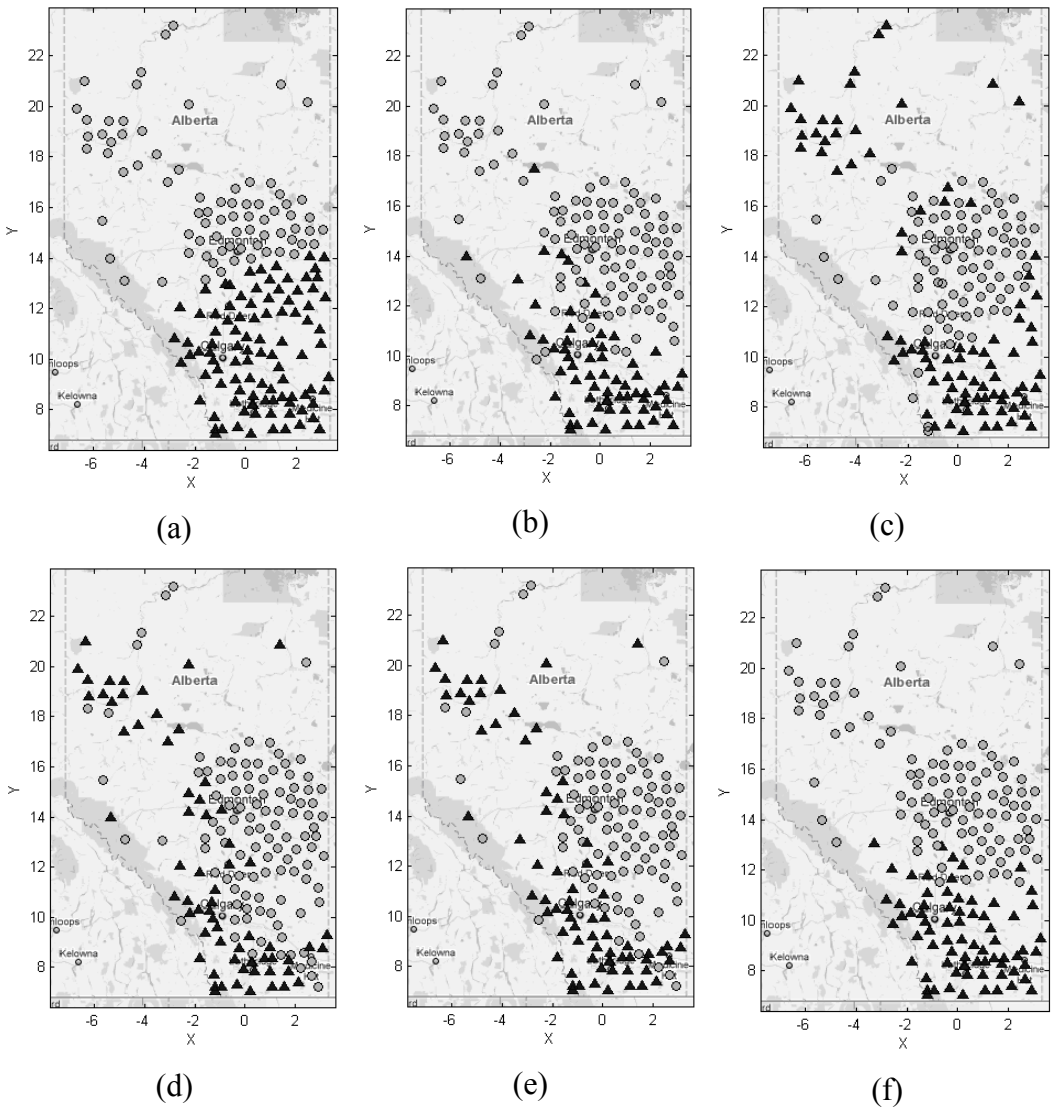


Figure 7.12. Revealed clusters for Alberta climate data for (a) spatial part of data, (b) temperature part, (c) precipitation part, (d) humidity part, (e) all parts and using FCM method and (f) all parts using the optimal weights. Number of clusters c=2 and DFT(24) representation has been used.

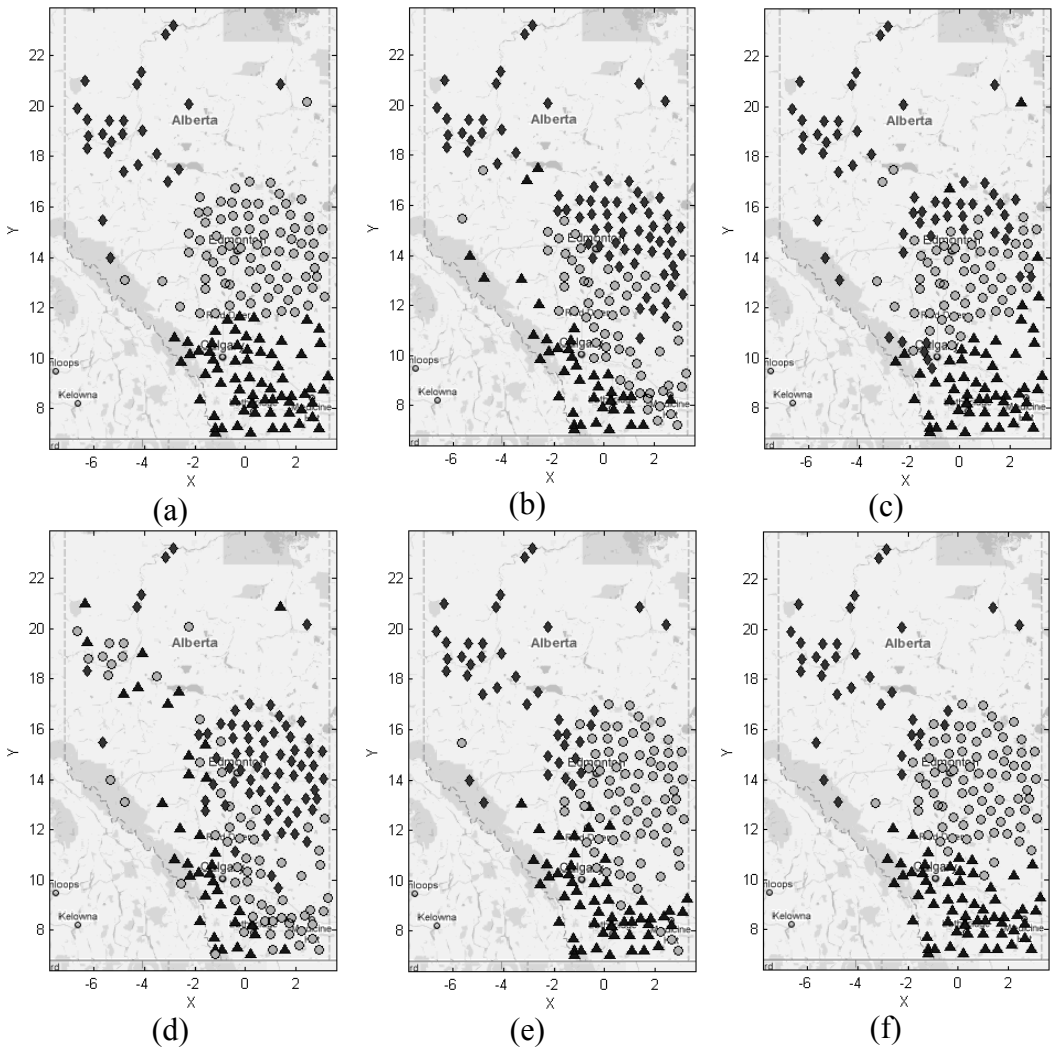


Figure 7.13. Revealed clusters for Alberta climate data (a) spatial part of data, (b) temperature part, (c) precipitation part, (d) humidity part, (e) all parts and using FCM method and (f) all parts using the optimal weights. Number of clusters c=3 and PAA(24) representation has been used.

7.5. Summary

In this chapter, we have proposed a fuzzy clustering approach to deal with data with blocks of features coming from different sources. This technique is suitable for clustering spatial univariate and spatial multivariate data. A distance function has been proposed to control the effect of each source in the clustering process

and the FCM objective function has been adopted to cope with the new distance function. An evaluation criterion is introduced and a particle swarm optimization is employed to find the optimal weights embedded in the new distance function. The proposed method has been studied over a synthetic and a real dataset. Experimental results show that the introduced method reveals interesting structures from data with blocks of features coming from distinct sources.

8. Anomaly Detection in Spatial Time Series

In this chapter, we introduce a novel technique for anomaly detection in spatial time series. Our objective is to detect any unexpected changes in a subsequence of a set of spatially neighboring time series. For this purpose, the clustering techniques introduced in previous chapters are used as a powerful instrument to reveal and visualize the available structure within spatial time series.

8.1. Problem formulation

Let us consider N data $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N$ each comprising a spatial part and a time series part. Figure 8.1 shows the overall scheme of the proposed method by presenting a bird's eye view at the introduced approach.

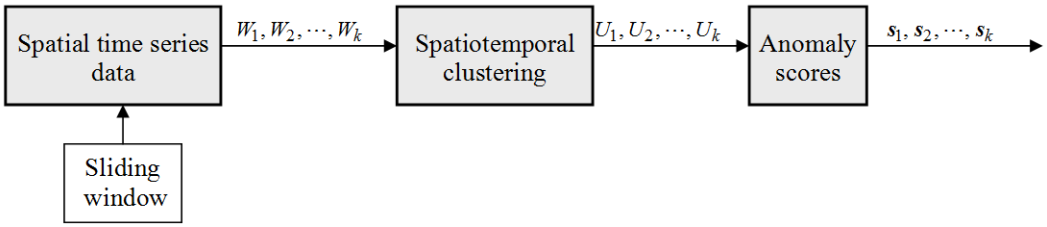


Figure 8.1. The overall scheme of the proposed method for anomaly detection in spatial time series.

At the first step, a sliding window moves across the time coordinate of data. Since there are N spatial time series, the time window at each step includes N subsequences. By considering the spatial information and the generated subsequences, we form a set of spatio-temporal subsequences $W_1, W_2, \cdots, W_k$. In fact, the sliding window allows us to look at the data at different time intervals. At the second step, the available structure in each set of spatio-temporal subsequences W_i, i=1,2,…k is revealed using a spatio-temporal clustering approach developed in the previous chapters. The result of this step is a collection

of partition matrices $U_1, U_2, \cdots, U_k$, each describing a set of clusters existing within the spatio-temporal subsequences. Next, as shown in Figure 8.1, we need to develop a technique to assign anomaly scores to the revealed clusters in different time windows. This technique should compare the revealed structure in each time window with the structures discovered in previous time intervals and assign an anomaly score to each cluster quantifying the level of unexpected changes in the structure of the data.

8.2. Anomaly evaluation in revealed structures

To assign an anomaly score to the revealed clusters inside time windows $W_1, W_2, \cdots, W_k$ (see Figure 8.1), for each single subsequence inside a time window, an anomaly score is estimated based on its historical behavior. Next, the estimated anomaly scores are aggregated to determine an anomaly score for each cluster inside each time window.

Let us consider the *j*th time window W_j. Since there are *N* spatial time series in dataset, W_j contains *N* subsequences. To assign an anomaly score to each subsequence, there are a number of methods proposed in the literature and some of them were reviewed in Chapter 3. Moreover, in Chapter 4 a novel technique proposed for this purpose. The strategy to estimate an anomaly score for a subsequence depends on the nature of data and the application purpose. One strategy can be considering the average distance between the subsequence and the subsequences located in previous time windows. The second strategy can be using a 1- nearest neighbor technique i.e. considering the distance between the subsequence and its nearest subsequence in the previous time windows. Furthermore, in periodic time series, the anomaly score for a subsequence can be considered as its distance from its corresponding subsequence in the previous time period.

Assume that $\boldsymbol{x}_{kj}$ is a subsequence of spatial time series $\boldsymbol{x}_k$ falling within the window W_j and f_{kj} is its anomaly score estimated using an anomaly detection

technique in time series. After computing an anomaly score for each single subsequence inside W_j, the anomaly scores can be aggregated to estimate an anomaly score for each cluster. Assuming that U is the partition matrix resulting from clustering of spatio-temporal data corresponding to the time window W_j, the anomaly scores for the clusters located in W_j, $\boldsymbol{s}_j = \{s_i, i = 1,2,...,c_j\}$, can be estimated using

$$s_i = \frac{\sum_{k=1}^{N} u_{ik} f_{kj}}{\sum_{k=1}^{N} u_{ik}}, \quad i = 1,2,...,c_j, \tag{8.1}$$

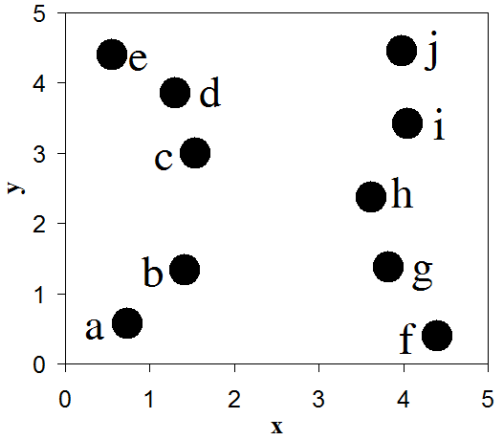
where, c_j is number of clusters in W_j and f_{kj} is anomaly score estimated for kth spatio-temporal data, $\boldsymbol{x}_k$, inside time window W_j. Higher value of s_i indicates that the subsequences belonging to ith cluster of W_j are more anomalous. On the other hand, a lower value of s_i indicates that the subsequences corresponding to this cluster are similar to the subsequences in their previous time intervals (based on the selected anomaly evaluation technique) and then the level of unexpected changes (anomalies) is lower.

8.3. Experimental studies

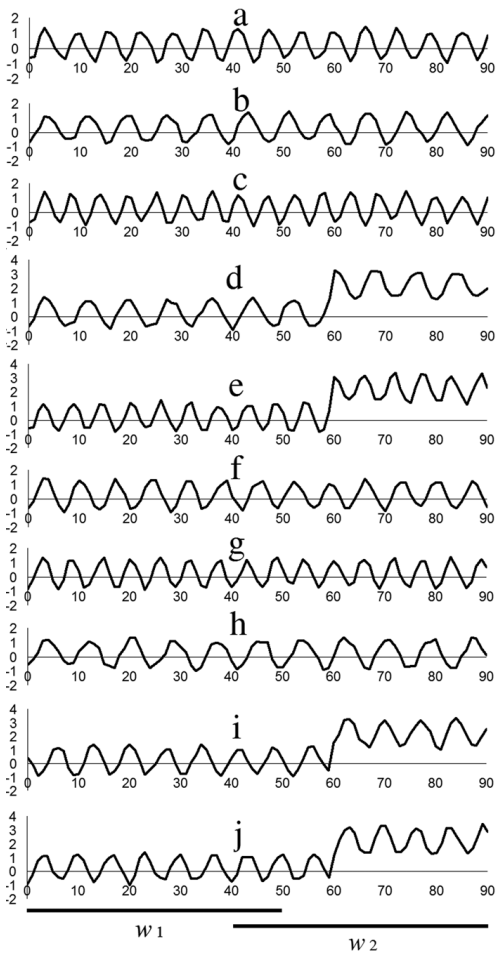
In this section, to illustrate the proposed approach a synthetic dataset as well as the Alberta temperature dataset has been studied.

8.3.1. Synthetic dataset

A synthetic dataset with 10 spatial time series labeled as “a” to “j” has been generated. Figure 8.2(a) shows the spatial coordinates and Figure 8.2(b) shows the time series part of data.



(a)

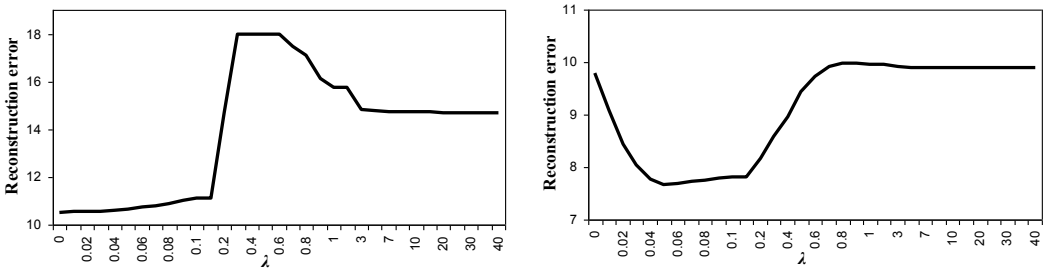


(b)

Figure 8.2. Synthetic dataset: spatial part (a), and the associated time series (b).

The length of each time series is 90 and there are some visible changes in the temporal part of spatial time series d, e, i, and j at time moment 61. In fact, these points are located spatially in topmost of y coordinate. Two time windows, one covering time steps from 1 to 50 (called W_1) and another one covering time steps from 41 to 90 (called W_2) have been considered in this experiment. By concatenating the spatial part of data with the specified time series parts, two sets of spatio-temporal subsequences are formed. Note that, selecting the length and position of time windows is application-dependent and the end-user may assign values to these parameters in an interactive manner when analyzing some initial results. To cluster the generated spatio-temporal data corresponding to time windows W_1 and W_2, the reconstruction criterion (described in Chapter 5) is considered and the number of clusters is varied from 2 to 4, and the fuzzification coefficient, *m*, was set to 2.0. Different values of λ in range [0, 100] have been considered leading to the optimal value of this parameter. For this value, the corresponding clusters have been selected.

Figure 8.3 displays the values of the reconstruction error for *c*=3 and different values of λ. For the segments W_1 and W_2, the optimal values of λ are 0.00 and 0.05, respectively.



(a) (b)

Figure 8.3. Reconstruction error vs. different values of λ for clustering spatial time series for (a) W_1 and (b) W_2. The number of clusters was *c*=3.

Figure 8.4 shows the revealed spatio-temporal clusters using the reconstruction criterion in the form of contour plot of membership degrees for W_1 and W_2 and number of clusters c=2, 3, and 4.

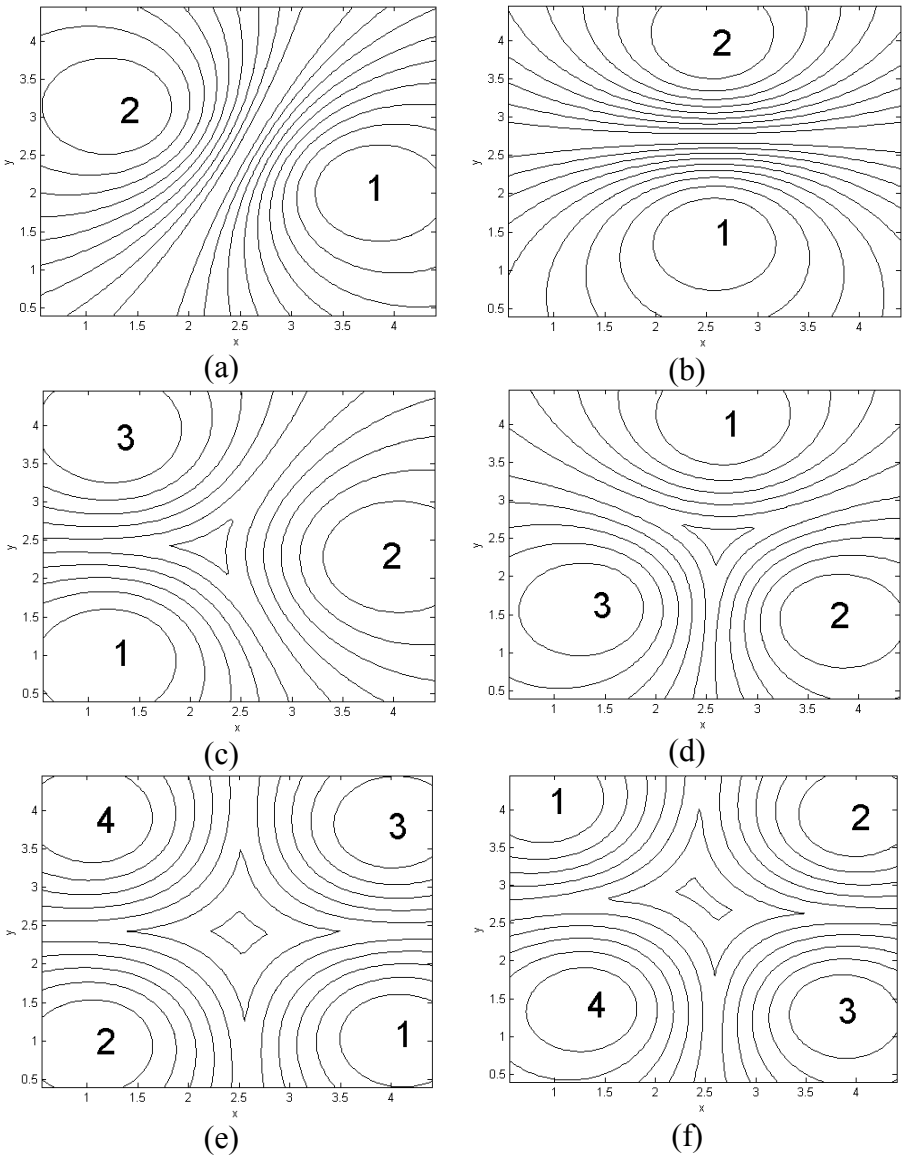


Figure 8.4. Spatio-temporal clusters of the generated synthetic dataset for c=2, 3, and 4. Figures (a), (c), and (e) correspond to time window W_1, and (b), (d), and (f) correspond to W_2.

For each time window, its corresponding subsequences in time series part along with the spatial part of data have been considered for clustering purpose. The number inside each cluster represents the order of this cluster in its corresponding partition matrix. There are some visible differences between clusters within time window W_1 and clusters inside W_2 because of existing some changes in time series part of data in time steps 61 - 90.

Following the scheme presented in Figure 8.1, now let us assign an anomaly score to the revealed clusters. Since there is no historical data for the subsequences located inside W_1, we assume that they are normal and their corresponding anomaly scores are zero. For subsequences located inside W_2 we calculate their anomaly scores as the squared Euclidean distance between each subsequence and the corresponding subsequence in the previous time window. In this technique, each subsequence inside W_2 which is different from its corresponding subsequence inside W_1, will receive a high anomaly score indicating a high level of unexpected changes. In the next step, the calculated anomaly scores for each subsequence are aggregated inside each cluster using (8.1). Table 8.1 shows the values of the anomaly score corresponding to clusters inside time window W_2 for the number of clusters c=2, 3, and 4.

As shown in this table, the second cluster in Figure 8.4(b), the first cluster in Figure 8.4(d), and the first and the second cluster in Figure 8.4(f) exhibit a high anomaly score indicating a high level of anomaly in the time series part of data in these clusters.

Table 8.1. Anomaly scores of spatio-temporal clusters inside time window W_2 for the number of clusters varying from 2 to 4.

Case study	Clusters			
	1	2	3	4
c=2	53.73	**125.75**	-	-
c=3	**130.33**	56.39	66.64	-
c=4	**111.47**	**139.88**	45.19	56.26

8.3.2. Alberta temperature data

In this sub-section, we look at the Alberta daily temperature data for the first 50 days of years 2009 and 2010 (www.agric.gov.ab.ca). The spatial time series corresponding to year 2010 are used to realize anomaly detection while the data corresponding to 2009 are assumed to be normal and are used in calculating anomaly scores. A number of stations located in the Western part of Alberta are selected and their daily temperature is increased 20 Celsius in days 25 to 45 of 2010 to produce some anomalies. A sliding window with length 30, moving 20 time steps in each movement is considered in this experiment. The following time windows are realized: W_1 for days 1 to 30 and W_2 for days 21 to 50. By concatenating the spatial part of data to the temporal subsequences W_1 and W_2, two spatio-temporal subsequences are realized for each year. Different number of clusters, c=2, 3, and 4 and reconstruction criterion have been considered in this experiment. Figure 8.5 shows the revealed clusters for time window W_1 in 2009 and 2010, and different number of clusters, and Figure 8.6 shows the revealed clusters for W_2. The number shown inside each cluster indicates the cluster's order in its corresponding partition matrix.

As shown in Figure 8.5 and Figure 8.6 the revealed clusters in years 2009 and 2010 for time window W_1 are quite similar, while for time window W_2 (which is contains the anomalous part of data) for number of clusters c=3 and c=4, the revealed clusters are different. However, for c=2 the clusters are similar. In fact, in time window W_2, for c=3 and 4, the anomalous part of data constructs a separate cluster, while for c=2 the number of clusters is not enough to construct a separate cluster for the anomalous part of data.

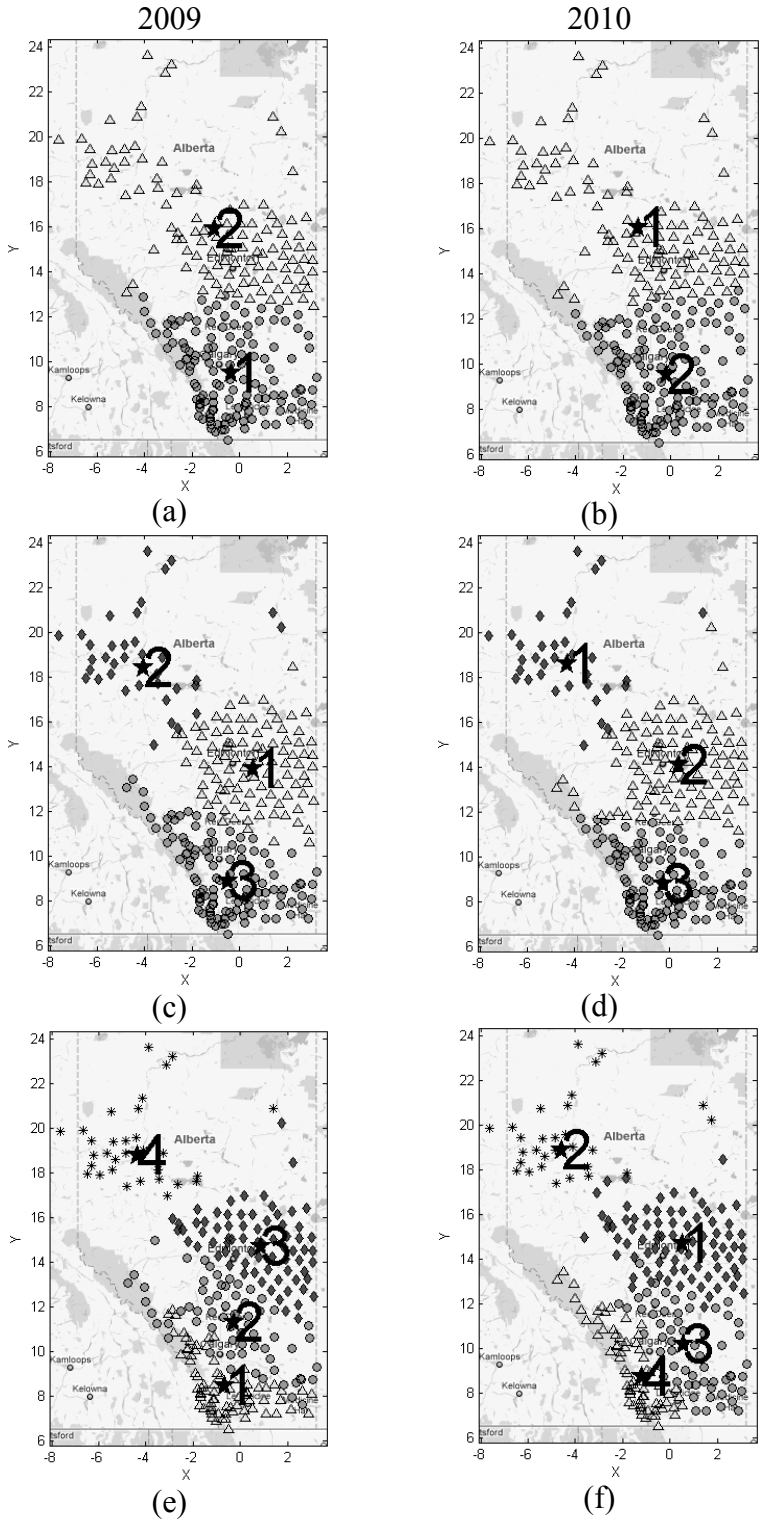


Figure 8.5. The revealed clusters for time window W_1 in 2009 and 2010, and different number of clusters. The numbers inside each cluster indicates its order in its corresponding partition matrix.

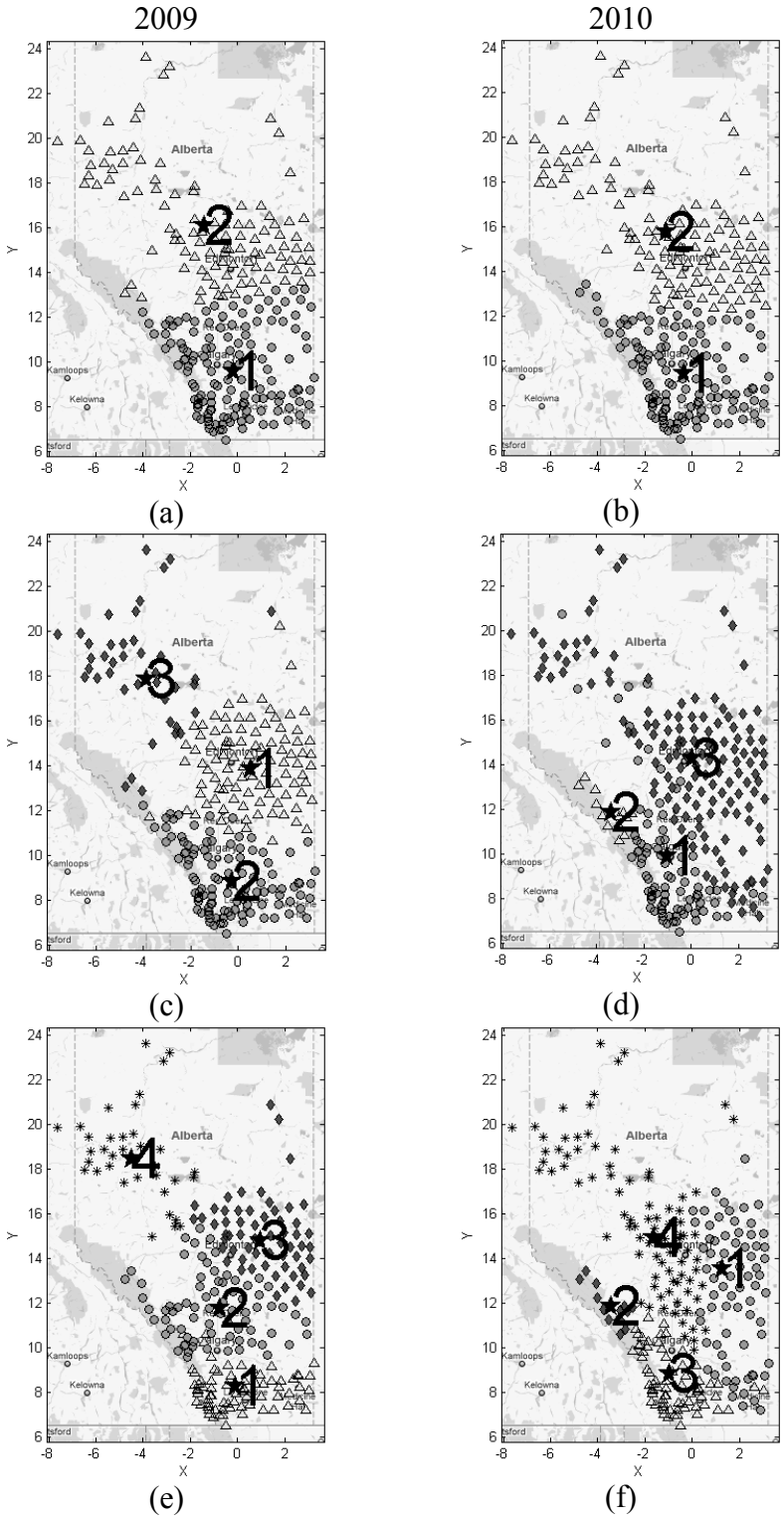


Figure 8.6. The revealed clusters for time window W_2 in 2009 and 2010, and different number of clusters.

Table 8.2. Estimated anomaly scores for each spatio-temporal cluster revealed for the Alberta temperature dataset in 2010 and different time windows.

Time window	Case study	Cluster			
		1	2	3	4
W_1	c=2	3520	2965	-	-
	c=3	3748	3356	2854	-
	c=4	3385	3732	3105	2816
W_2	c=2	3045	3299	-	-
	c=3	2553	**8736**	2897	-
	c=4	2878	**9102**	2467	2878

Table 8.2 shows the calculated anomaly scores for each time window in 2010. As shown in this table, for time window W_1, there is no any anomalous cluster, while in time window W_2, when the number of clusters is c=3 and 4, the second cluster (in both cases) are detected as anomalous clusters. Note that in this time window, for the number of clusters c=2, both clusters are considered as normal. One may conclude that the number of clusters in this case is not enough to capture anomalous characteristics of data. As the result, the anomaly scores estimated for each single subsequence in anomalous part of data are distributed among the two revealed clusters, leading to a not significant overall anomaly score for each cluster. One may employ a cluster validity index technique to find a suitable number of clusters for each time window.

8.4. Summary

A novel technique for anomaly detection for spatial time series is proposed in this chapter. A sliding window with a fixed length has been considered to generate a set of spatio-temporal subsequences in successive time steps. Then a spatio-temporal clustering has been employed to reveal existing structure within each time window. Considering the historical behavior of each single subsequence inside each time window, an anomaly score has been estimated and then, the estimated anomaly scores are aggregated to assign an anomaly score to each cluster inside each time window. This technique examined over a synthetic and Alberta temperature dataset. We have showed that the proposed method detects

the incident anomalies in form that is understandable and user-friendly by strongly supporting the visualization and comprehension of the revealed structures.

9. Anomaly Characterization in Spatial Time Series

Although detecting anomalous part of data is critical in many applications, analyzing and characterizing the detected anomalies (e.g., identifying the source of anomalies and visualizing anomaly propagation over time) is equally important. In this chapter, we introduce a technique to discover the available relations among structures within data in different time intervals. Although one may compare the revealed clusters in different time intervals using some techniques reported in the literature (e.g., [109–112]), the proposed technique in this study shows some advantages as a vehicle to visualize the propagation of anomalies over time and space.

9.1. Problem formulation

In this chapter, we add another component to the proposed framework for anomaly detection and characterization in spatial time series. As the result, Figure 8.1 (in the previous chapter) can be refined as shown in Figure 9.1.

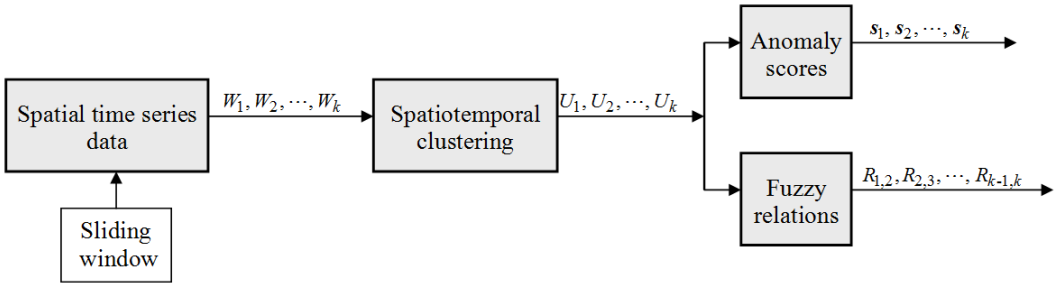


Figure 9.1. The overall scheme for anomaly detection and characterization in spatial time series.

As shown in this figure, the detected structures positioned in different time windows come in the form of a family of partition matrices. We are interested in analyzing the relationships available between these structures. As the FCM algorithm is initialized randomly, the order of clusters encountered in these

matrices could be different for different runs and different time windows. Furthermore, there might be different number of clusters for various time windows. As the result, we have to form a fuzzy relation using which we map clusters present in a partition matrix U_1 to the clusters in U_2. The entries of this fuzzy relation should describe degrees to which clusters in U_1 are related to clusters in U_2. Using this technique, a chain of relationships among different clusters in various time intervals can be constructed and the anomalous clusters can be tracked and analyzed from structures revealed in previous time intervals.

9.2. A gradient based fuzzy relation for anomaly characterization

Let us assume two partition matrices U_1 and U_2 of dimensionality $c_1 \times N$ and $c_2 \times N$ respectively, where N is number of data, c_1 is number of clusters in U_1, and c_2 is number of clusters in U_2. Each spatial time series $\boldsymbol{x}_k$ is described in U_1 as a collection of membership degrees $\boldsymbol{ux}_k = [ux_{1,k}, ux_{2,k}, ..., ux_{c_1,k}]^{\mathrm{T}}$, k=1,2,…,N, and similarly, each spatial time series $\boldsymbol{y}_k$ is represented in U_2 through $\boldsymbol{uy}_k = [uy_{1,k}, uy_{2,k}, ..., uy_{c_2,k}]^{\mathrm{T}}$, k=1,2,…,N. Our intent is to form a relational dependency that maps clusters present in U_1 onto the clusters occurring in U_2. We construct a fuzzy relation R of dimensionality $c_1 \times c_2$ whose entries are optimized in a way that the following performance index becomes minimized:

$$Q = \sum_{k=1}^{N} \left\| \boldsymbol{ux}_k - R \circ \boldsymbol{uy}_k \right\|^2 = \sum_{k=1}^{N} \sum_{i=1}^{c_1} \left(ux_{i,k} - \max_{j=1,2,...,c_2} (r_{i,j} \ \mathrm{t} \ uy_{j,k}) \right)^2 , \qquad (9.1)$$

where R= $[r_{i,j}]$, i=1,2,…, c_1, j=1,2,…, c_2, is a fuzzy relation to be determined, $\circ$ denotes a sup-t composition with "t" being a t-norm, and $ux_{i,k}$ is the ith element of $\boldsymbol{ux}_k$. By considering a gradient descent optimization approach to minimize (9.1), we update the entries of R in an iterative fashion for i=1,2,…, c_1, and j=1,2,…, c_2. For the an element in R, say $r_{s,t}$ we have

$$r_{s,t}(iter+1)=\left\langle r_{s,t}(iter)-\alpha\frac{\partial Q}{\partial r_{s,t}(iter)}\right\rangle, \tag{9.2}$$

where, $\langle\ \rangle$ indicates that the resulting values of $r_{s,t}(iter+1)$ are confined to the [0,1] interval; α is a positive learning rate controlling intensity of learning, and *iter* stands for iteration index. By choosing the max-min composition operator (although the solution can be derived for different types of max-t and min-s compositions involving various t-norms and t-conorms), we have

$$\frac{\partial Q}{\partial r_{s,t}}=2\left(\max_{j=1,2,\ldots,c_2}\left(\min(r_{i,j},uy_{j,k})\right)-ux_{i,k}\right)\times\varphi_{s,t,i,k}, \tag{9.3}$$

where

$$\varphi_{s,t,i,k}=\frac{\partial}{\partial r_{s,t}}\max_{j=1,2,\ldots,c_2}\left(\min(r_{i,j},uy_{j,k})\right). \tag{9.4}$$

In virtue of the nature of the minimum and maximum operations we have

$$\varphi_{s,t,i,k}=\begin{cases}1 & \text{if } i=s \text{ and } r_{s,t}<uy_{t,k} \text{ and } r_{s,t}>\max\limits_{\substack{j=1,2,\ldots,c_2,\\ j\neq t}}\left(\min(r_{s,j},uy_{j,k})\right)\\ 0 & \text{otherwise}\end{cases}. \tag{9.5}$$

Using the above optimization, the fuzzy relation *R* can be estimated. Each row in *R* corresponds with a given cluster in U_1 and each column in *R* stands for a cluster in U_2. As an example, let us assume that the following fuzzy relation is given:

$$R=\begin{bmatrix}0.2 & 0.9 & 0.3\\ 0.4 & 0.2 & 0.8\\ 0.1 & 0.05 & 0.2\end{bmatrix}. \tag{9.6}$$

This fuzzy relation indicates that the first cluster in U_1 is related to the first, second and the third cluster in U_2 with degrees of 0.2 and 0.9, and 0.3, respectively. The second cluster in U_1 is related to the first, second and third clusters in U_2 with degrees 0.4 and 0.2, and 0.8, respectively. When a chain of structures in successive time steps (resulting from clustering data in different time windows) is considered, the above fuzzy relations characterize the behavior of clusters (linkages among them) and specify the origin of an arbitrary cluster.

9.3. Experimental studies: A simulated outbreak scenario

The North American Animal Disease Spread Model (NAADSM) [91] is used in order to simulate a livestock disease outbreak across the province of Alberta. NAADSM is a unit (herd)-based stochastic state-transition spread simulation model for contagious diseases among animals. In this method, each infected susceptible unit may have four disease states comprising latent period, sub-clinically infectious period, clinically infectious period, and naturally immune period. To simulate a disease spread among a group of units in a map, a set of spread parameters should be determined. Some of these parameters are: duration of each disease state in the form of a probability density function, rate of animal shipment, movement distance, shipping delay, wind direction, maximum distance of spread, airborne transport delay, etc. Also, some spread control policies including quarantine, destruction, and vaccination can be determined.

We considered 246 stations in Alberta agriculture and rural development system (www.agric.gov.ab.ca) and for each station a number of cattle herds have been generated randomly. Moreover, the population of each herd is considered as a random number in a certain range. Using NAADSM and considering some values for the above-mentioned parameters, an outbreak with a period of 100 days has been simulated. In the resulting outbreak dataset, for each station the spatial coordinates are provided in the form of latitude-longitude and there is a time series with length 100 measuring the rate of infected herds within each station for each day. Figure 9.2(a) shows the spatial part of the simulated data. As shown in this figure, an outbreak (triangles) occurred in Southern part of the province. The highlighted station in this figure (named Del Bonita) is the start point of the outbreak. Figure 9.2(b) shows the rate of infected herds corresponding to this station during the simulation, and Figure 9.2(c) shows the rate of infected herds in the entire province for 100 days.

In the first step of the experiments, the latitude-longitude pairs are mapped to Cartesian coordinates to be used in the calculations of the Euclidean distance. In selecting time windows to generate subsequences, two parameters, namely the

length of time windows and the length of overlap between two successive time windows should be considered. As discussed earlier, these parameters can be selected by the end-user in an interactive manner when running the system and analyzing initial solutions. Moreover, this selection can be based on the application and the nature of data. For example, in periodic time series the length of time windows can be equal to the length of the period of time series.

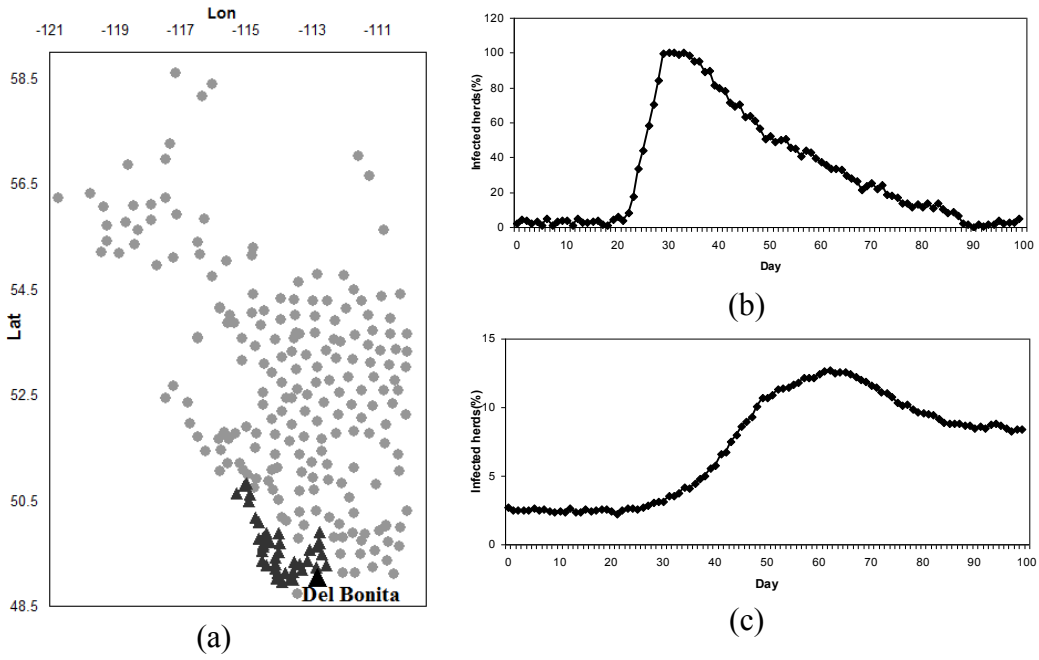


Figure 9.2. (a) The spatial part of simulated outbreak, (b) time series corresponding to the station Del Bonita, and (c) the rate of infected herds across the province in 100 days.

In this experiment, we simply used a sliding window of length 20 and in each step the window moves in 10 days. As the result, the subsequences inside the following time windows are considered: W_1: days 1 to 20, W_2: days 11 to 30, W_3: days 21 to 40, W_4: days 31 to 50, W_5: days 41 to 60, W_6: days 51 to 70, W_7: days 61 to 80, W_8: days 71 to 90, and W_9: days 81 to 100.

In the next step, the spatial part of data is concatenated to the above-generated temporal subsequences and the resulting spatio-temporal subsequences are clustered using the reconstruction criterion discussed in Chapter 5. Different

values of parameter λ in range [0, 100] are considered for this purpose. A challenging problem here is to find an appropriate number of clusters for each time window (cluster validity index). Although for this problem, numerous approaches have been reported in the literature (see for example ref. [92]), but most of them are not suitable for data having different parts with different natures (e.g., spatial time series here). In this study, we employed the reconstruction criterion in order to find an appropriate number of clusters for each set of subsequences inside time windows. For this purpose, the temporal part of data inside each time window along with the spatial part is clustered using the reconstruction criterion for different number of clusters, and the number of clusters that can reduce the reconstruction error effectively has been chosen. In fact, lower amount of reconstruction error specifies a higher quality of clusters in terms of granulation and de-granulation [93]. Figure 9.3 shows the reconstruction error vs. number of clusters c=1 to 12 for the time windows W_1 to W_9. For c=1 we simply considered the average value of data objects (both spatial part and temporal part) as the cluster center and the membership degree of each data object to that cluster center is set to 1. Also in all experiments the fuzzification coefficient, m, was set to 2.

As shown in this figure, for all the defined time windows, usually increasing the number of clusters decreases the reconstruction error and after some steps this reduction in reconstruction error has been flattened. Consequently, we select the number of clusters at the point where the values of the reconstruction error start to exhibit a saturation effect (no further substantial changes of error are reported when increasing the values of c). Moreover, some automations (e.g. using BIC [34]) can be realized for this purpose.

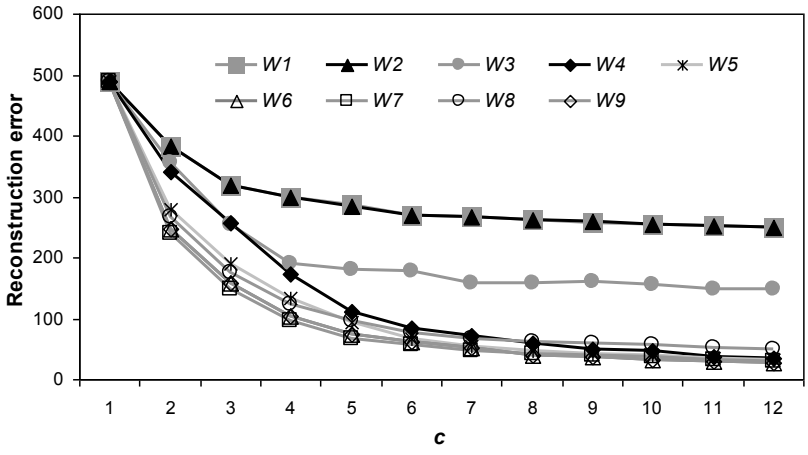


Figure 9.3. Reconstruction error vs. different number of clusters for windows W_1 to W_9.

Table 9.1 shows the selected number of clusters for different time windows. Note that in most cases determining the number of clusters is application-dependent and the user can choose the number of clusters based of the nature of the problem under consideration.

Table 9.1. The selected number of clusters for different time windows.

Time window	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9
c	3	3	4	5	5	5	5	5	5

Figures 9.4(a)-(i) show the revealed spatio-temporal clusters for different time windows. The stars represent spatial cluster centers and the number positioned next to each cluster center represents the order of that cluster in its corresponding partition matrix. As shown in these figures, the outbreak has been started in time window W_3 in the Southern part of the province around the Del Bonita station (see Figure 9.2(a)). In time window W_4, the outbreak moves in two ways: Northern part and Western (left-hand) part of the map. As can be seen from the clusters coming from next time windows, it continues to propagate to the Western part of the province. In fact, one of the advantages of the proposed technique is that it can visualize the dynamic changes (migration) of anomalies over time.

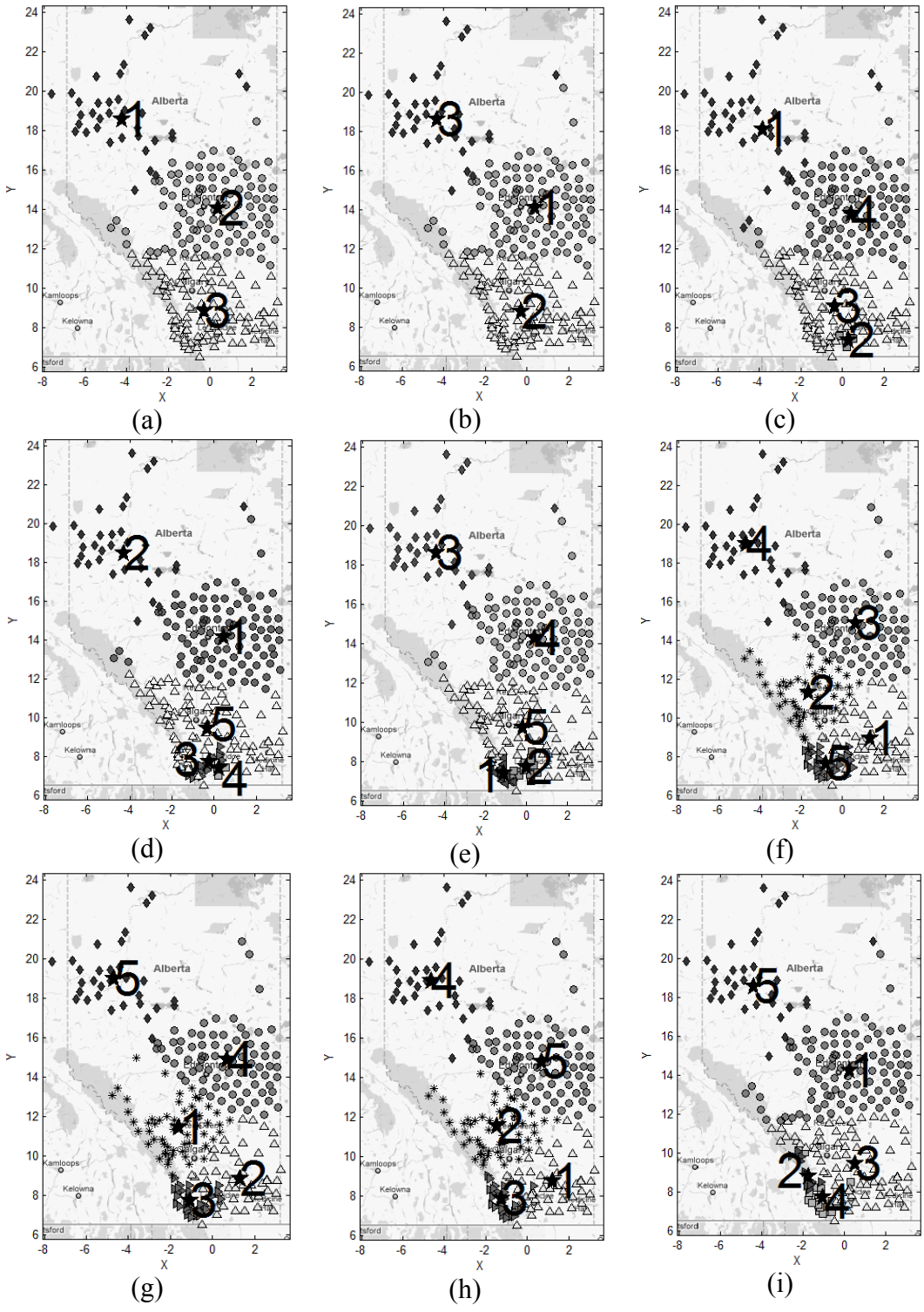


Figure 9.4. The revealed spatio-temporal clusters for time windows (a) W_1, (b) W_2, (c) W_3, (d) W_4, (e) W_5, (f) W_6, (g) W_7, (h) W_8, and (i) W_9.

Now let us estimate an anomaly score for the revealed clusters in different time windows. For this purpose, for each subsequence inside a time window W_j, its anomaly score is considered as the average squared Euclidean distance to its previous subsequences. Formally, considering $\boldsymbol{x}_{kj}$ as a subsequence of spatial time series $\boldsymbol{x}_k$ falling within the window W_j, its anomaly score is expressed as follows

$$f_{kj} = \frac{1}{j-1}\sum_{i=1}^{j-1}\left\|\boldsymbol{x}_{kj} - \boldsymbol{x}_{ki}\right\|^2 . \tag{9.7}$$

The intuition behind this measure is that, in disease data usually normal subsequences are very similar to the subsequences present in the previous time intervals. As the result, (9.7) generates a high score for anomalous subsequences, while normal subsequences exhibit a lower score.

After computing an anomaly score for each single subsequence inside each time window, the anomaly scores are aggregated inside each cluster using the described technique in the previous chapter (using (8.1)). Table 9.2 reports the estimated anomaly scores calculated for different clusters revealed for time windows W_2 to W_9.

Table 9.2. Anomaly scores reported for different clusters in time windows W_2 to W_9.

Time widow	Clusters				
	1	2	3	4	5
W_2	0.01	0.04	0.01	-	-
W_3	0.07	**6.47**	0.05	0.04	-
W_4	0.10	0.16	**4.89**	**7.54**	0.13
W_5	**7.70**	**7.97**	0.23	0.14	0.18
W_6	0.56	0.37	0.17	0.16	**7.73**
W_7	0.27	0.50	**6.29**	0.13	0.12
W_8	0.63	0.38	**5.45**	0.29	0.24
W_9	0.21	**7.07**	0.53	**2.98**	0.30

Since W_1 is the first generated time window and there is no historical data for that, the subsequences inside this time window considered as normal. As shown in

this table, the first anomalous cluster has been detected in time window W_3 and cluster 2 present in this time window is an anomalous one with the anomaly score of 6.47. In the next time window, W_4, both clusters 3 and 4 exhibit a high anomaly score and in W_5, clusters 1 and 2 are anomalous clusters. In the time window W_6 to W_8 there is one anomalous cluster, and finally in the window W_9 two anomalous clusters have been found.

Now let us analyze the movement of clusters over time (anomaly propagation) using the proposed fuzzy relational model in this chapter. In the gradient-based method, considering a high value for learning rate, α, may lead to some oscillations in the produced values of the performance index and may eventually lead to a danger of falling into local optima. In contrast, by selecting a very small value of the learning rate we end up with a very slow learning. Different values of this parameter have been examined and finally its value was set to 0.01 and the learning has been terminated once there was no significant reduction observed in the performance index.

Table 9.3 shows the estimated fuzzy relations obtained for successive structures corresponding to the generated time windows. The fuzzy relation obtained for the transition from W_1 to W_2 indicates that clusters 1, 2 and 3 in W_1, in the next time step move to clusters 3, 1, and 2 in W_2, respectively. Figures 9.4(a) and 9.4(b) visualize these transitions. Considering fuzzy relation from W_2 to W_3, one may conclude that cluster 2 from W_2 moves to cluster 2 and 3 of W_3. Since cluster 2 of W_3 exhibits a high anomaly score, this indicates that some parts of cluster 2 of W_2 in the next step encountered some anomalies in the temporal part of data.

Using the fuzzy relations presented in Table 9.3, one may visualize the evolution of clusters in different time windows using a graph-oriented representation. In Figure 9.5 nodes represent clusters, edges stand for relations (associations) between clusters, and each layer of nodes presents single-step windows. The numbers displayed over the nodes of the graph represent anomaly scores reported in Table 9.2, and the level of shading of the edges corresponds to the membership

value present in the corresponding entry of the fuzzy relation. In other words, for the membership grades close to 1, the edges are black and for values of membership close to zero the links are almost invisible.

Table 9.3. Fuzzy relation between any two consecutive revealed structures.

Time window	Fuzzy relation (R)
W_1 to W_2	$\begin{pmatrix} 0.02 & 0 & 0.95 \\ 0.93 & 0.05 & 0.04 \\ 0.04 & 0.93 & 0.01 \end{pmatrix}$
W_2 to W_3	$\begin{pmatrix} 0.05 & 0.05 & 0 & 0.90 \\ 0.02 & 0.93 & 1.00 & 0.02 \\ 0.92 & 0.02 & 0 & 0 \end{pmatrix}$
W_3 to W_4	$\begin{pmatrix} 0.07 & 0.88 & 0.08 & 0.01 & 0 \\ 0 & 0 & 0 & 0.96 & 0 \\ 0.10 & 0.03 & 0.68 & 0.02 & 0.80 \\ 0.77 & 0.09 & 0.15 & 0.02 & 0.15 \end{pmatrix}$
W_4 to W_5	$\begin{pmatrix} 0.09 & 0.08 & 0.02 & 0.97 & 0.04 \\ 0.08 & 0.07 & 0.98 & 0 & 0 \\ 0.13 & 0.54 & 0 & 0 & 0 \\ 0.05 & 0.31 & 0 & 0 & 0 \\ 0.44 & 0.08 & 0.01 & 0.01 & 0.97 \end{pmatrix}$
W_5 to W_6	$\begin{pmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.46 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.38 \\ 0.03 & 0.05 & 0.03 & 0.98 & 0.03 \\ 0.14 & 0.19 & 0.94 & 0.02 & 0.03 \\ 0.82 & 0.66 & 0.03 & 0.01 & 0.04 \end{pmatrix}$
W_6 to W_7	$\begin{pmatrix} 0.01 & 0.95 & 0.06 & 0.01 & 0.01 \\ 0.96 & 0.02 & 0.06 & 0.02 & 0.01 \\ 0.03 & 0.01 & 0.02 & 0.97 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.97 \\ 0.01 & 0.01 & 0.83 & 0.01 & 0.01 \end{pmatrix}$
W_7 to W_8	$\begin{pmatrix} 0.03 & 0.99 & 0.03 & 0.01 & 0.02 \\ 0.95 & 0.01 & 0.04 & 0.01 & 0.01 \\ 0.04 & 0.01 & 0.91 & 0.01 & 0.01 \\ 0.02 & 0.02 & 0.02 & 0.01 & 0.96 \\ 0.01 & 0.01 & 0.01 & 0.98 & 0.01 \end{pmatrix}$
W_8 to W_9	$\begin{pmatrix} 0.01 & 0.09 & 0.65 & 0.07 & 0.01 \\ 0.17 & 0.12 & 0.37 & 0.06 & 0.02 \\ 0.01 & 0.56 & 0.01 & 0.76 & 0.01 \\ 0.01 & 0.06 & 0.01 & 0.02 & 0.96 \\ 0.75 & 0.08 & 0.04 & 0.05 & 0.02 \end{pmatrix}$

The constructed graph in Figure 9.5 shows the evolution of clusters in time windows W_1 to W_9. Using this structure one may track normal and anomalous clusters. Let us consider cluster 3 from time window W_1. As shown in Figure 9.5, it moves to cluster 2 in the next time window, W_2. In W_3 this cluster split into clusters 2 and 3. Cluster 2 is anomalous and is related to anomalous cluster 4 in time window W_4. On the other hand, some data of cluster 3 in W_3 encounter with some anomalies in the next time window resulting the appearance of cluster 3 in W_4. Both anomalous clusters in W_4 merge into cluster 2 in time window W_5. A new anomalous cluster (cluster 1) has emerged in this time window from cluster 5 of W_4. Anomalous clusters 1 and 2 of W_5 are merged into cluster 5 in time window W_6. It moves to cluster 3 in W_7, and then moves to cluster 3 in W_8. Finally, this cluster splits into clusters 2 and 4 in W_9.

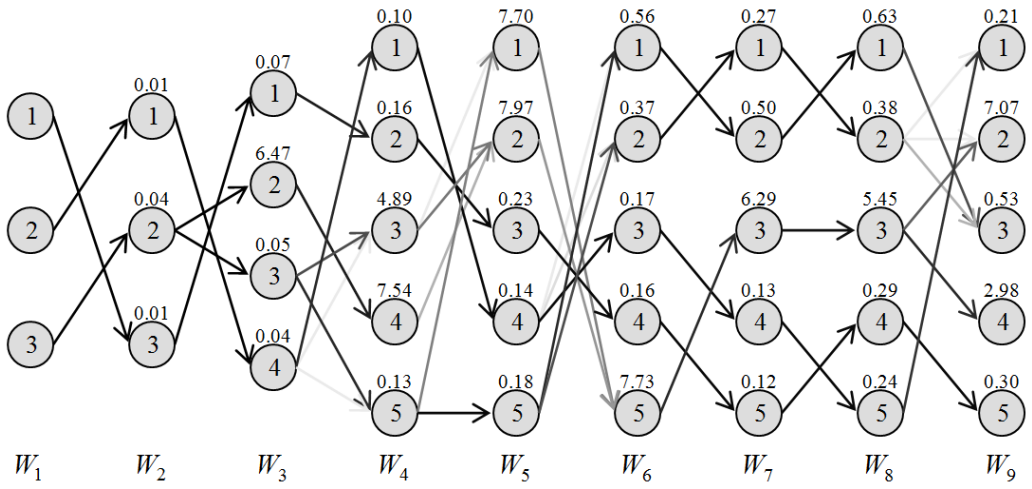
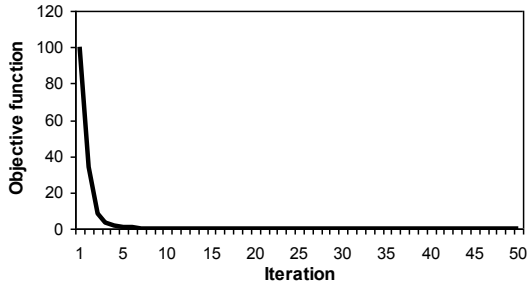


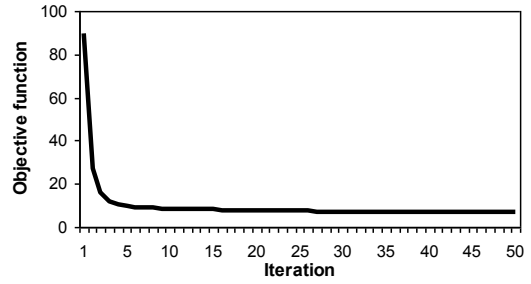
Figure 9.5. Graph representation of anomaly scores and fuzzy relations reported in Table 9.2 and 9.3.

Figures 9.6(a) and 9.6(b) show the amount of the performance index defined in (9.1) to calculate the fuzzy relations from W_1 to W_2, and from W_2 to W_3 respectively. The final value of the objective function in Figure 9.6(a) is very close to zero, while in Figure 9.6(b) the amount of the final objective function is

higher. The reason is that the revealed structures in time windows W_1 and W_2 are very similar, so that the available structure in W_1 can be estimated using the structure in W_2 and the estimated fuzzy relation. On the other hand, the available structures in time windows W_2 and W_3 are different and the resulting objective function has a higher extent.



(a)



(b)

Figure 9.6. The values of the minimized objective function reported in 50 iterations of the learning scheme: optimization of the relationships from W_1 to W_2 (a), and W_2 to W_3 (b).

9.4. Summary

In this chapter, we added a new component to the proposed framework for anomaly detection and characterization in spatial time series. A gradient based fuzzy relation technique has been developed to find existing relationships between local structures in successive time steps. Using this approach, one may visualize and quantify the propagation of anomalies over time. The technique is illustrated using an outbreak scenario simulated using NAADSM over a set of agriculture

stations in Alberta. We have showed that the proposed method detects and characterizes the incident anomalies in an understandable way for the end-user.

10. Conclusions and Future Works

In this study, we developed a general framework for anomaly detection and characterization in spatial time series. The framework comprises a number of components (blocks) each fulfilling a set of sub-tasks. At the first step of the method, a sliding window moves across the time coordinate of data generating a set of spatio-temporal subsequences. This component allows us to look at the data locally. Next, the available structure inside the generated spatio-temporal subsequences are revealed and visualized through a spatio-temporal clustering. Three criteria, namely a reconstruction, a prediction and an agreement have been investigated for evaluating the revealed spatio-temporal clusters. Dealing with the reconstruction criterion is of interest when evaluating the quality of clusters in the processes of information granulation and de-granulation. The prediction criterion can be considered when forecasting a temporal component of the data given their spatial location. And finally, the agreement-based criterion is useful, when the objective is to reveal a general structure over all data sources having a high level of agreement among the available structures in separate data sources.

After discovering the spatio-temporal structures using the proposed clustering techniques, the next step is to assign an anomaly score to each cluster measuring the level of unexpected changes inside the structure of data. For this purpose, an anomaly score can be assigned to each single subsequence inside each time window and the estimated anomaly scores can be aggregated inside each cluster. Finally, a gradient-based fuzzy relation technique is proposed to quantify the available relations between structures of data in successive time steps, leading to a visualization of propagation of anomalies over time. The proposed framework in this study is general and the end-user can interact with the system to determine different parameters and methods for anomaly detection based on the nature of data and the application purpose. Moreover, this framework supports strongly, the visualization of structure inside data, so that the end-user can fully understand the changes and dynamics within the data.

The main contributions of our study are as follows:

- A new clustering technique for spatial time series data through a reconstruction criterion is proposed.
- A new clustering technique for spatial time series using a prediction criterion is introduced.
- An agreement-based fuzzy clustering for spatial multivariate time series is proposed.
- A new technique for assigning anomaly scores to spatio-temporal clusters for quantifying the level of unexpected changes in data is developed.
- A new fuzzy relation-based technique to visualize the propagation of anomalies in spatial time series over time is introduced.

The proposed framework can be further investigated for future extension as follows:

- Developing some other spatio-temporal clustering techniques to reveal structures within spatial time series (univariate and multivariate).
- Investigating different distance functions for various parts of data (e.g., Euclidean distance for spatial part, and dynamic time warping distance for time series part) and its impact on clustering. One has to be aware of the challenges of refinements of the generic FCM method to cope with the diversity of distance measures different from the Euclidean one.
- Developing some efficient heuristics to find optimal length of time windows for time series segmentation (generating spatio-temporal subsequences).
- Developing new techniques for comparing revealed spatio-temporal clusters in various time steps and assigning anomaly scores to quantify the level of unexpected changes.

- Extending the proposed framework for anomaly detection and characterization for other types of spatio-temporal data (e.g., spatio-temporal event data, trajectory data etc.).

Bibliography

[1] E. Keogh, J. Lin, and A. Fu, "Hot SAX: Efficiently finding the most unusual time series subsequence," *Proceeding of Fifth IEEE International Conference on Data Mining*, 2005, pp. 226-233.

[2] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no.3, pp. 32-57, 1974.

[3] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York, Plenum, 1981.

[4] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107-144, Aug. 2007.

[5] K. Chakrabarti, E. Keogh, S. Mehrotra, M. Pazzani, "Locally adaptive dimensionality reduction for indexing large time series databases," *ACM Transactions on Database Systems*, vol. 27, no. 2, pp. 188-228, Jun. 2002.

[6] C. A. Ratanamahatana, E. Keogh, A.J. Bagnall, and S. Lonardi, "A novel bit level time series representation with implications for similarity search and clustering," *Proceedings of the 9th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, Hanoi, Vietnam, 2005. pp. 771-777.

[7] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, 2008, pp. 1542-1552.

[8] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," *Proceedings of the IEEE International Conference on Data Mining*, 2001, pp. 289-296.

[9] F. Korn, H.V. Jagadish, and C. Faloutsos, "Efficiently supporting ad-hoc queries in large datasets of time sequences" *Proceedings of the ACM SIGMOD*

international conference on Management of data, New York, USA, 1997, pp. 289-300.

[10] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," *Proceedings of the ACM SIGMOD international conference on Management of data*, 1994, pp. 419-429.

[11] Y. Cai and R. Ng, "Indexing spatio-temporal trajectories with chebyshev polynomials," *Proceedings of the ACM SIGMOD international conference on Management of data*, 2004, pp. 599-610.

[12] K.-P. Chan and A.W.-C. Fu, "Efficient time series matching by wavelets," *Proceedings of the 15th International Conference on Data Engineering*, 1999, pp. 126-133.

[13] K.-P. Chan, A.W.-C. Fu, and C. Yu, "Haar wavelets for efficient similarity search of time-series: with and without time warping," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 3, pp. 686-705, May/Jun. 2003.

[14] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Journal of Knowledge and Information Systems*, vol. 3, no. 3, pp. 263-286, Aug. 2001.

[15] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, Jul. 1989.

[16] D. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series" *Proceedings of the Workshop on Knowledge Discovery in Databases*, 1994, pp. 359-370.

[17] M. Vlachos, D. Gunopulos, and G. Kollios, "Discovering similar multidimensional trajectories," *Proceedings of the 18th International Conference on Data Engineering*, 2002, pp. 673 - 684.

[18] L. Chen, M.T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," *Proceedings of the ACM SIGMOD international conference on Management of data*, 2005, pp. 491-502.

[19] M. Ramoni, P. Sebastiani, And P. Cohen, "Bayesian clustering by dynamics," *Machine Learning*, vol. 47, no. 1, pp. 91-121, 2002.

[20] Y. Yang and K. Chen, "Time series clustering via RPCL network ensemble with different representations," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 41, no. 2, pp. 190-199, Mar. 2011.

[21] T. Warren Liao, "Clustering of time series-a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857-1874, Nov. 2005.

[22] X. Golay, S. Kollias, G. Stoll, D. Meier, A. Valavanis, and P. Boesiger, "A new correlation-based fuzzy logic clustering algorithm for fMRI," *Magnetic Resonance in Medicine*, vol. 40, no. 2, pp. 249-260, 1998.

[23] C.S. Möller-Levet, F. Klawonn, K.-H. Cho, and O. Wolkenhauer, "Fuzzy clustering of short time series and unevenly distributed sampling points," *LNCS, Proceedings of the IDA*, 2003, pp. 28-30.

[24] X. Zhang, J. Liu, Y. Du, and T. Lv, "A novel clustering method on time series," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11891-11900, Sept. 2011.

[25] F. Petitjean, A. Ketterlin, and P. Gancarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, vol. 44, no. 3, pp. 678-693, Mar. 2011.

[26] K. Kalpakis, D. Gada, and V. Puttagunta, "Distance measures for effective clustering of ARIMA time-series," *Proceedings of the IEEE International Conference on Data Mining*, 2001, pp. 273-280.

[27] Y. Xiong and D. Yeung, "Time series clustering with ARMA mixtures," *Pattern Recognition*, vol. 37, no. 8, pp. 1675-1689, Aug. 2004.

[28] E.A. Maharaj, P. D'Urso, and D.U.A. Galagedera, "Wavelet-based fuzzy clustering of time series," *urnal of Classification*, vol. 27, no. 2, pp. 231-275, 2010.

[29] P. D'Urso, and E.A. Maharaj, "Autocorrelation-based fuzzy clustering of time series," *Fuzzy Sets and Systems*, vol. 160, no. 24, pp. 3565-3589, Dec. 2009.

[30] E.A. Maharaj and P. D'Urso, "Fuzzy clustering of time series in the frequency domain," *Information Sciences*, vol. 181, no. 7, pp.1187-1211, Apr. 2011.

[31] M. Vlachos, J. Lin, E. Keogh, and D. Gunopulos, "A wavelet based anytime algorithm for k-means clustering of time series," *Proceedings of the Workshop on Clustering High Dimensionality Data and Its Applications*, 2003, pp. 23-30.

[32] L. Kaufman and P.J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*, Wiley, New York, 1990.

[33] G.E.P. Box, and G. Jenkins, *Time series analysis: forecasting and control*, Holden-Day, 1976.

[34] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461-464, 1978.

[35] S. Kisilevich, F. Mansmann, M. Nanni, S. Rinzivillo, "Spatio-temporal clustering," *Data mining and knowledge discovery handbook*, Part 6, pp. 855-874, 2010.

[36] M. Kulldorff, "Prospective time periodic geographical disease surveillance using a scan statistic," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 164, no. 1, pp. 61-72, 2001.

[37] H. Izakian, W. Pedrycz, "A new PSO-optimized geometry of spatial and spatio-temporal scan statistics for disease outbreak detection," *Swarm and Evolutionary Computation*, vol. 4, pp. 1-11, Jun. 2012.

[38] F. Di Martino, S. Sessa, "The extended fuzzy C-means algorithm for hotspots in spatio-temporal GIS," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11829-11836, Sept. 2011.

[39] M. Wang, A. Wang, A. Li, "Mining spatial-temporal clusters from geo-databases," *Proceedings of the second International Conference on advanced data mining and Applications*, 2006, pp. 63 - 270.

[40] M. Ester, HP Kriegel, J. Sander, X Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Data mining and knowledge discovery*, 1996, pp. 226-231.

[41] Z. Liu, R. George, "Fuzzy cluster analysis of spatio-temporal data," *Proceedings of the ISCIS 18th International symposium*, Antalya, Turkey, 2003, pp. 984-991.

[42] M. Deng, Q. Liu, J. Wang, Y. Shi, "A general method of spatio-temporal clustering analysis," *Science China, Information Sciences*, doi: 10.1007/s11432-011-4391-8.

[43] D.L. Pham, "Spatial models for fuzzy clustering," *Computer vision and image understanding*, vol. 84, no. 2, pp. 285-297, 2001.

[44] R. Coppi, P. D'Urso, P. Giordani, "A fuzzy clustering model for multivariate spatial time series," *Jurnal of Classification*, vol. 27, no.1, pp. 54-88, Mar. 2010.

[45] M. Nanni, D. Pedreschi , "Time-focused clustering of trajectories of moving objects*," Journal of Intelligent Information Systems*, vol. 27, no. 3, pp. 267-289, Nov. 2006.

[46] M. Ankerst, M. M. Breunig, HP. Kriegel, J. Sander, "OPTICS: ordering points to identify the clustering structure," *Proceedings of the ACM SIGMOD international conference on Management of data*, Philadelphia, 1999, pp. 49-60.

[47] S. Gaffney, P. Smyth, "Trajectory clustering with mixtures of regression models," *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999, pp. 63-72.

[48] A. P. Dempster , N. M. Laird , D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B (Methodological),* vol. 39, no. 1, pp. 1-38, 1977.

[49] P. Kalnis, N. Mamoulis, S. Bakiras, "On discovering moving clusters in spatio-temporal data," *Proceedings of the International symposium on spatial and temporal databases*, 2005, pp. 364-381.

[50] M. Sato and Y. Sato, "On a multicriteria fuzzy clustering method for 3-way data," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 2, no. 2, pp. 127-142, Jun. 1994.

[51] W. Pedrycz, V. Loia, S. Senatore, "P-FCM: a proximity-based fuzzy clustering," *Fuzzy Sets and Systems,* vol. 148, no. 1, pp. 21-41, 2004.

[52] V. Loia, W. Pedrycz, S. Senatore, "Semantic Web Content Analysis: A Study in Proximity-Based Collaborative Clustering," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 6, pp. 1294-1312, Dec 2007.

[53] W. Pedrycz, "Collaborative fuzzy clustering," *Pattern Recognition Letters*, vol. 23, no. 14, pp. 1675-1686, 2002.

[54] L. F. S. Coletta, L. Vendramin, E. R. Hruschka, R. J. G. B. Campello, W. Pedrycz, "Collaborative fuzzy clustering algorithms: Some refinements and design guidelines," *IEEE Trans. Fuzzy Syst*., vol. 20, no. 3, pp. 444 - 462, Jun. 2012.

[55] W. Pedrycz, P. Raia, "Collaborative clustering with the use of Fuzzy C-Means and its quantification," *Fuzzy Sets and Systems*, vol. 159, no. 18, pp. 2399 - 2427, 2008.

[56] A. Strehl, J. Ghosh , "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583-617, 2002.

[57] W. Pedrycz, K. Hirota, "Forming consensus in the networks of knowledge," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 5, pp. 657-666, 2007.

[58] K. Punera, J. Ghosh, "Consensus based ensembles of soft clusterings," *Applied Artificial Intelligence*, vol. 22, no. 7, pp. 780-810, 2008.

[59] W. Pedrycz, K. Hirota, "A consensus-driven fuzzy clustering," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1333-1343, 2008.

[60] X. Z. Fern, C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," *21th International Conference on Machine Learning*, Banff, Canada, 2004.

[61] A. L.N. Fred, A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835-850, Jun 2005.

[62] H. Ayad, M.S. Kamel, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 160-173, Jan 2008.

[63] N. Ilc, A. Dobnikar, "Generation of a clustering ensemble based on a gravitational self-organizing map," *Neurocomputing*, vol. 96, no. 1, pp. 47-56, 2012.

[64] P. Hore, L. O. Hall, D. B. Goldgof, "A scalable framework for cluster ensembles," *Pattern Recognition*, vol. 42, no. 5, pp. 676 - 688, 2009.

[65] H.G. Ayad, M.S. Kamel, "On voting-based consensus of cluster ensembles," *Pattern Recognition*, vol. 43, no. 5, pp.1943-1953, 2010.

[66] S. Vega-Pons, J. Correa-Morris, J. Ruiz-Shulcloper, "Weighted partition consensus via kernels," *Pattern Recognition* vol. 43, no. 8, pp. 2712-2724, 2010.

[67] F. Wang, C. Yang, Z. Lin, Y. Li, Y. Yuan, "Hybrid sampling on mutual information entropy-based clustering ensembles for optimizations," *Neurocomputing,* vol. 73, no. 7-9, pp.1457-1464, 2010.

[68] A.L.V. Coelho, E. Fernandes, K. Faceli, "Inducing multi-objective clustering ensembles with genetic programming," *Neurocomputing*, vol. 74, no. 1-3, pp. 494-498, 2010.

[69] W. Pedrycz and A. Bargiela, "Fuzzy clustering with semantically distinct families of variables: Descriptive and predictive aspects," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1952-1958, 2010.

[70] P. Protopapas, J.M. Giammarco, L. Faccioli, M.F. Struble, R. Dave, C. Alcock, "Finding outlier light curves in catalogues of periodic variable stars," *Monthly Notices of the Royal Astronomical Society*., vol. 369, no. 2, pp. 677-696, 2006.

[71] E. Keogh, S. Lonardi, C. A. Ratanamahatana, "Towards parameter-free data mining," *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge discovery and data mining*, pp. 206-215, 2004.

[72] M. Das, S. Parthasarathy, "Anomaly detection and spatio-temporal analysis of global climate system," *In Proceedings of the 3rd International Workshop on Knowledge Discovery from Sensor Data*, 2009, pp. 142-150.

[73] H. Izakian, W. Pedrycz, "Anomaly detection in time series using a Fuzzy C-Means clustering," *Proceedings of the Joint IFSA World Congress and NAFIPS Annual Meeting*, Canada, 2013, pp. 1513-1518.

[74] W. Pedrycz and J.V. de Oliveira, "A development of fuzzy encoding and decoding through fuzzy clustering," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 4, pp. 829-837, Apr. 2008.

[75] V. Chandola, V. Mithal, V. Kumar, "Comparative evaluation of anomaly detection techniques for sequence data," *8th IEEE International Conference on Data Mining*, Pisa, Italy, 2008, pp. 743-748.

[76] J. Ma, S. Perkins, "Time-series novelty detection using one-class support vector machines," *Proceedings of the International Joint Conference on Neural Networks*, 2003, pp. 1741- 1745.

[77] D. Dasgupta, S. Forrest, "Novelty detection in time series using ideas from immunology," *Proceedings of the International Conference on Intelligent Systems*, 1996.

[78] D. Gao, Y. Kinouchi, K. Ito, X. Zhao, "Neural networks for event extraction from time series: A back propagation algorithm approach," *Future Generation Computer Systems*, vol. 21, no.7, pp.1096-1105, Jul 2005.

[79] J. Takeuchi, K. Yamanishi, "A unifying framework for detecting outliers and change points from time series," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 4, pp. 482-489, Apr 2006.

[80] C. Brighenti, M. A. Sanz-Bobi, "Auto-regressive processes explained by self-organized maps: Application to the detection of abnormal behavior in industrial processes," *IEEE Transactions on Neural Networks,* pp. 2078- 2090, vol. 22, no. 12, Dec 2011.

[81] H. Cheng, P. Tan, C. Potter, S. Klooster, "A robust graph-based algorithm for detection and characterization of anomalies in noisy multivariate time series," *Proceedings of the IEEE International Conference on Data Mining Workshops*, Pisa, Italy, 2008, pp. 349- 358.

[82] A. Khatkhate, A. Ray, E. Keller, S. Gupta, S.C. Chin, "Symbolic time-series analysis for anomaly detection in mechanical systems," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 4, pp. 439-447, Aug 2006.

[83] D. J. Hill, B. S. Minsker, E. Amir, "Real-time Bayesian anomaly detection for environmental sensor data," *Proceedings 32nd Congress of the International Association of Hydraulic Engineering and Research*, 2007.

[84] E. W. Dereszynski, T. G. Dietterich, "Spatio-temporal models for data-anomaly detection in dynamic environmental monitoring campaigns," *ACM Transactions on Sensor Networks*, vol. 8, no. 1, Aug 2011.

[85] D. B. Neill, "Expectation-based scan statistics for monitoring spatial time series," *International Journal of Forecasting*, vol. 25, no. 3, pp. 498-517, Sept. 2009.

[86] M. Kulldorff, R. Heffernan, J. Hartman, R. Assunção, F. Mostashari, "A space-time permutation scan statistic for disease outbreak detection," *PLoS Medicine*, vol. 2, no. 3, pp. 216-224, Mar. 2005.

[87] V. Chandola, A. Banerjee, V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 832-839, May 2012.

[88] D.T. Pham and A.B. Chan "Control Chart Pattern Recognition using a New Type of Self Organizing Neural Network" *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 212, no. 2, 1998, pp. 115-127.

[89] M. H. Magalhães, R. Ballini, F A. C. Gomide, "Granular Models for Time-Series Forecasting" *Handbook of Granular Computing (eds W. Pedrycz, A. Skowron and V. Kreinovich)* 2008.

[90] H. G. Seedig, T. A. Runkler, R. Grothmann, "Forecasting of clustered time series with recurrent neural networks and a fuzzy clustering scheme," *International joint Congress on neural networks*, Atlanta, 2009, pp. 1360-1367.

[91] N. Harvey, A. Reeves, M. Schoenbaum, F. Zagmutt-Vergara, C. Dubé, A. Hill, B. Corso, B. McNab, C. Cartwright, M. Salman, "The North American Animal Disease Spread Model: A simulation model to assist decision making in evaluating animal disease incursions," *Preventive Veterinary Medicine*, vol. 82, no. 3-4, pp. 176-197, Dec 2007.

[92] W. Wang, Y. Zhanga, "On fuzzy cluster validity indices," *Fuzzy Sets and Systems*, vol. 158, no. 19, pp. 2095 - 2117, Oct 2007.

[93] H. Izakian, W. Pedrycz, I. Jamal, "Clustering spatio-temporal data: An augmented fuzzy C-Means," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 5, pp. 855 - 868, Oct. 2013.

[94] E. Keogh, J. Lin, A. W. Fu, H. V. Herle, "Finding unusual medical time-series subsequences: Algorithms and applications," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 3, July 2006, pp. 429-439.

[95] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large datasets," *Proceedings of the ACM SIGMOD international conference on Management of data*, pp. 427-438, 2000.

[96] V. Chandola, A. Banerjee, V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol 41, no 3, July 2009, pp. 1-72.

[97] A. Agogino, K. Tumer, "Entropy based anomaly detection applied to space shuttle main engines," *IEEE Aerospace Conference*, 2006.

[98] P. D'Urso, "Fuzzy clustering for data time arrays with inlier and outlier time trajectories," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 5, pp. 583-604, Oct. 2005.

[99] V. Petridis and A. Kehagias, "Predictive modular fuzzy systems for time-series classification," *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 3, pp. 381-397, Aug. 1997.

[100] S. M Arafat and M. Skubic, "Modeling fuzziness measures for best wavelet selection," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 5, pp. 1259-1270, Oct. 2008.

[101] Z. Chen, S. Aghakhani, J. Man, and S. Dick, "ANCFIS: A neuro fuzzy architecture employing complex fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 305-322, Apr. 2011.

[102] S. Chen and C. Chen, "TAIEX forecasting based on fuzzy time series and fuzzy variation groups," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 1-12, Feb. 2011.

[103] Y.C. Cheng, S.T. Li, "Fuzzy time series forecasting with a probabilistic smoothing hidden Markov model," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 2, pp. 291 - 304, Apr. 2012.

[104] A. Lemos, W. Caminhas, and F. Gomide, "Multivariable Gaussian evolving fuzzy modeling system," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 91-104, Feb. 2011.

[105] W. Ma, L. Jiao, M. Gong, C. Li, "Image change detection based on an improved rough fuzzy c-means clustering algorithm," *International Journal of Machine Learning and Cybernetics*, DOI: 10.1007/s13042-013-0174-4.

[106] H. Izakian, W. Pedrycz, "Agreement-based fuzzy C-means for clustering data with blocks of features," *Neurocomputing*, vol. 127, pp. 266-280, March 2014.

[107] W. Pedrycz, "Proximity-based clustering: a search for structural consistency in data with semantic blocks of features," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 5, pp. 978 - 982, Oct. 2013.

[108] X. X. Zhang, H. X. Li, C. K. Qi, "Spatially constrained fuzzy-clustering-based sensor placement for spatio-temporal fuzzy-control system," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 5, pp. 946-957, Oct 2010.

[109] D. T. Anderson, A. Zare, S. Price, "Comparing fuzzy, probabilistic, and possibilistic partitions using the Earth Mover's distance," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 4, pp. 766 - 775, Aug. 2013.

[110] R. Campello, "A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment," *Pattern Recognition Letters*, vol. 28, no. 7, pp. 833-841, May 2007.

[111] D. Anderson, J. Bezdek, M. Popescu, and J. Keller, "Comparing fuzzy, probabilistic and possibilistic partitions," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 5, pp. 906-918, Oct 2010.

[112] E. Hullermeier, M. Rifqi, S Henzgen, R. Senge, "Comparing fuzzy partitions: A generalization of the rand index and related measures," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 3, pp. 546- 556, Jun 2012.

[113] J. Kennedy and R.C. Eberhart, Particle Swarm Optimization, *Proc. of the IEEE International Conference on Neural Networks*, pp.1942-1948, 1995.

[114] C. Robertson, T.A. Nelson, Y.C. MacNab, A.B. Lawson, "Review of methods for space-time disease surveillance," *Spatial and Spatio-temporal Epidemiology* 1 (2010) 105-116.

[115] E. Knox, "The detection of space-time interactions," *Appl Stat* 1964;13:25-9.

[116] P. Rogerson, "Surveillance systems for monitoring the development of spatial patterns," *Statistics in Medicine*, vol. 16, no. 18, pp.2081-2093, 1997.

[117] K. Kleinman, R. Lazarus, R. Platt, "A generalized linear mixed models approach for detecting incident clusters of disease in small areas with an application to biological terrorism," *American Journal of Epidemiology*, vol. 159, no. 3, pp. 217-24, 2004.

[118] N. Best, S. Richardson, A. Thomson, "A comparison of Bayesian spatial models for disease mapping," *Statistical Methods in Medical Research*, vol. 14, no. 1, pp. 35-59, 2005.

[119] J. Aldstadt, "An incremental Knox test for the determination of the serial interval between successive cases of an infectious disease," *Stochastic Environmental Research and Risk Assessment,* vol. 21, no. 5, pp.487-500, 2007

[120] G. D. Johnson, "Prospective spatial prediction of infectious disease: experience of New York State (USA) with West Nile Virus and proposed directions for improved surveillance," *Environmental and Ecological Statistics*, vol. 15, pp. 293-311, 2008.

[121] C. Sonesson, "A CUSUM framework for detection of space-time disease clusters using scan statistics," *Statistics in Medicine*, vol. 26:4770-89, 2007.

[122] R. Kosmider, L. kelly, S. Evans, G. Gettinby, "A statistical system for detecting Salmonella outbreaks in British livestock," *Epidemiology & Infection,* vol. 134, pp. 952-960, 2006.

[123] D. B. Neill, G. F. Cooper, K. Das, X. Jiang, and J. Schneider, "Bayesian network scan statistics for multivariate pattern detection," *In J. Glaz, V. Pozdnyakov, and S. Wallenstein, eds., Scan Statistics: Methods and Applications*, 221-250, 2009.

[124] The United States Historical Climatology Network (USHCN), Available online at: http://cdiac.ornl.gov/epubs/ndp/ushcn/ushcn.html.

[125] A.L. Goldberger, L.A.N. Amaral, L. Glass, J.M. Hausdorff, P.Ch. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.K. Peng, H.E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation* 101 (23) (2000) e215-e220.

[126] Y. Yoshinari, W. Pedrycz, K. Hirota, "Construction of fuzzy models through clustering techniques," *Fuzzy Sets and Systems*, vol. 54, no. 2, pp. 157-165, Mar. 1993.

[127] M. F. Azeem, M. Hanmandlu, N. Ahmad, "Structure identification of generalized adaptive neuro-fuzzy inference systems," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 5, pp. 666-681, Oct. 2003.

[128] A. Celikyilmaz, I. B. Turksen, "Enhanced fuzzy system models with improved fuzzy clustering algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 3, pp. 779-794, Jun. 2008.

[129] B. Hartmann, O. Banfer, O. Nelles, A. Sodja, L. Teslic,I. Skrjanc, "Supervised hierarchical clustering in fuzzy model identification", *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 6, pp. 1163-1176, Dec. 2011.

[130] W. Pedrycz, H. Izakian, "Cluster-Centric Fuzzy Modeling," *IEEE Transactions on Fuzzy Systems*, DOI: 10.1109/TFUZZ.2014.2300134, 2014.