ABSTRACT

SENCAN, HUSEYIN. (Dis)Similarity-based Classification Of Cross Domain Multivariate Spatiotemporal Systems Using Dynamic Network Structures and Graph Edit Distances. (Under the direction of Dr. Chang S. Nam and Dr. Robert St. Amant.)

Many complex systems can be naturally represented as networks. Consider the human brain, where coherent physiological activity among neural elements in spatially distant brain locations creates cognitive functions; or climate, in which a grid of oscillators governs oceanic-atmospheric circulations; or finance, as a collection of stock markets; or social interactions in general. In the literature, topological structures of such networks have been analyzed extensively.

From a statistical learning viewpoint, dynamic changes in these network structures can be associated with events that emerge from the underlying structures. For instance, dynamic changes in brain network structure may correspond to an inherent cognitive dysfunction or a unique function. Changes in a climate network may correspond to a natural phenomenon such as an extreme event like a hurricane or a drought. Structural changes in a social network such as the science collaboration network may point to a political paradigm shift.

The first part of this dissertation evaluates classification and inference methods for spatiotemporal datasets (subject to scope of this thesis) and identifies suitable approaches for handling spatiotemporal patterns, as well as challenges and gaps in current learning algorithms. These act as motivation for the current work. Further, possible future directions for the analysis of spatiotemporal data are identified.

The second part of this dissertation describes a graph-theoretic approach for the classification of (extreme) events emerging from spatiotemporal physical systems based on underlying network structures. A novel, efficient dissimilarity-based pattern classification algorithm using the distances between network structures created from spatiotemporal processes as input is proposed to predict the outcome of such events as an alternative to existing feature-based classification methods.

Finally, the dissertation presents motivating examples and application areas for which the proposed methods are suitable, along with evaluation metrics that demonstrate their validity and superiority to existing approaches. Four datasets from two distinct real-world emergent phenomena have been formulated as spatiotemporal data classification
problems and selected as target applications: (1) from the cognitive science domain, predicting subjects’ mental states in two separate electroencephalogram (EEG)-based cognitive science experiments, and (2) from the climate domain, forecasting landfall behavior of North Atlantic and South West Indian ocean hurricane trajectories. Results show that network-based classification approaches for the target datasets outperform current generic and domain-specific feature selection algorithms at the 0.05α-level over four different classification tasks.
© Copyright 2016 by Huseyin Sencan

All Rights Reserved
(Dis)Similarity-based Classification Of Cross Domain Multivariate Spatiotemporal Systems Using Dynamic Network Structures and Graph Edit Distances

by
Huseyin Sencan

A dissertation submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

Computer Science

Raleigh, North Carolina
2016

APPROVED BY:

Dr. Matthias Stallmann
Dr. Dennis R. Bahler

Dr. Chang S. Nam
Co-chair of Advisory Committee

Dr. Robert St. Amant
Co-chair of Advisory Committee
BIOGRAPHY

Huseyin Sencan was born in Antalya, Turkey. After graduating from (Istanbul) Bogazici University Computer Education and Technology Department with honors degree in 2004, he continued his career as a Software Engineer for various industries. In 2006, he was enrolled (Istanbul) Marmara University Computer Engineering Department from which he received his M.Sc. degree in 2009. He started his Ph.D. in North Carolina State University Computer Science department in Fall 2009 and successfully defended his Written Preliminary in Fall 2012, then Oral Preliminary in Summer 2015 under the guidance of Dr. Robert St. Amant and Dr. Chang S. Nam. During his Ph.D., he worked in various research projects as a research assistant and application designer/developer; such as between 2012 and 2013 in ISE Brain-Computer Interface Laboratory, between 2011 and 2012 in CSC Knowledge Discovery Laboratory, between 2009 to 2010 in CSC AI-Game Design Laboratory.
ACKNOWLEDGEMENTS

I would like to express my appreciation my co-advisors, Dr. Robert St. Amant and Dr. Chang S. Nam for their guidance, support and patience throughout my research. I would not be able to complete this work without their help. I would also like to thank Dr. Matthias Stallmann and Dr. Dennis R. Bahler for accepting to be member of my committee and for their valuable comments and suggestions. I would also like to thank Dr. Michael Young, and Dr. Nagiza Samatova whom I had a chance to work with in the beginning of my PhD here at NC State. In addition, many thanks to my friends and fellow graduate students Can Babaoglu and Prairie Goodwin for their help and comments throughout my PhD. Last but not least, I would also like to thank my family for their continuous support and encouragement throughout my academic career.
# TABLE OF CONTENTS

List of Tables .................................................................................. vi
List of Figures .................................................................................. vii

Chapter 1 Introduction ..................................................................... 1
  1.1 Problem Statement and Research Questions ............................. 4
  1.2 Thesis Statement ................................................................... 5
  1.3 Dissertation Contributions .................................................. 5
  1.4 Dissertation Outline ............................................................ 6

Chapter 2 Background Work ............................................................ 7
  2.1 Spatial Datasets and Classification .......................................... 8
  2.2 Temporal (Time Series) Datasets and Classification ................. 10
  2.3 Spatiotemporal Datasets and Classification ............................. 11
    2.3.1 Classification of Human Cognitive States using EEG .......... 13
    2.3.2 Classification and Predictive Modeling in Climate .............. 16
  2.4 Network-based Approaches ................................................... 19
    2.4.1 Brain Networks .......................................................... 20
    2.4.2 Climate Networks ......................................................... 26

Chapter 3 Proposed Method: Similarity-Based Classification of Spatiotemporal Datasets using Networks ......................................................... 29
  3.1 Similarities Between Spatiotemporal Systems ......................... 30
  3.2 Similarities Between Networks .............................................. 31
  3.3 Graph Matching ................................................................... 33
    3.3.1 Exact Graph Matching .................................................. 33
    3.3.2 Inexact Graph Matching ................................................. 34
  3.4 Network Classification using (Dis)Similarities ......................... 37
    3.4.1 Learning with (indefinite-) Kernels and (pseudo-) Euclidian Space Embeddings of Dissimilarities .................. 39
    3.4.2 Dissimilarity Space Embeddings ...................................... 45
    3.4.3 (Nearest) Neighborhood-based Learning ......................... 46

Chapter 4 Motivating Examples and Datasets .................................... 47
  4.1 Hurricane Track Classification ............................................... 47
    4.1.1 Dataset I - Atlantic Ocean Hurricanes .......................... 48
    4.1.2 Dataset II- Indian Ocean Hurricanes .............................. 48
    4.1.3 Climate Data and Preprocessing ..................................... 49
  4.2 Classification of Human Cognitive States ............................... 50
    4.2.1 Dataset III - Expert Classification in Engineering Design .... 50
4.2.2 Dataset IV - Mental Task Classification ......................... 50
4.2.3 EEG Data and Preprocessing ...................................... 51

Chapter 5 Experimental Results and Discussion ....................... 53
  5.1 Classification Algorithms on Raw (Tabular) Data .................. 54
    5.1.1 Support Vector Machines ..................................... 54
    5.1.2 k-Nearest Neighbor Classifiers ............................... 56
    5.1.3 Decision Trees ................................................ 56
    5.1.4 Naive Bayes .................................................... 57
    5.1.5 Results ....................................................... 58
  5.2 Feature Selection Algorithms ...................................... 58
    5.2.1 t-Score ....................................................... 60
    5.2.2 F-Score ....................................................... 60
    5.2.3 Information Gain ............................................. 60
    5.2.4 Minimum redundancy, maximum relevance ....................... 60
    5.2.5 Fast Correlation-Based Filter ................................. 61
    5.2.6 CFS ........................................................ 61
    5.2.7 Results ....................................................... 61
  5.3 Dimensionality Reduction, Feature Extraction
    and Domain Specific Features ....................................... 64
    5.3.1 Principal Component Analysis ................................ 65
    5.3.2 Common Spatial Patterns ...................................... 65
    5.3.3 Climate Indices ............................................... 65
    5.3.4 Correlations / Coherences as Features ....................... 66
    5.3.5 Results ....................................................... 67
  5.4 Similarity-Based Classification .................................... 70
    5.4.1 Similarities across components of spatiotemporal systems ... 71
    5.4.2 Structural Similarities Between Networks and Classification .. 71
    5.4.3 Spectral Features ............................................. 74
    5.4.4 Multi-dimensional Scaling ................................... 76
    5.4.5 Converting Distances into Similarities ....................... 76
  5.5 Comparison of Feature and Similarity-based Approaches .......... 79

Chapter 6 Conclusions ..................................................... 81
  6.1 Contributions to Computer Science ................................ 82
  6.2 Contributions to Cognitive Science ................................ 84
  6.3 Contributions to Climate Science ................................ 85
  6.4 Limitations ....................................................... 85
  6.5 Future Work ...................................................... 87

Bibliography ............................................................... 89
LIST OF TABLES

Table 5.1  Overall cross validation accuracies for tabularly embedded data for each classification algorithms. Highest and significantly improved statistics are displayed in bold. .......................... 58

Table 5.2  Cross validation accuracies for filter and embedded feature selection algorithms across all four benchmark datasets. Highest and significantly improved statistics are displayed in bold. .............................. 62

Table 5.3  Cross validation accuracies for specific feature selection algorithms across four benchmark datasets. Highest and significantly improved statistics are displayed in bold. Climate indices are domain specific features for climate data. CSP as a supervised feature selection technique developed for brain EEG is also applied in climate data. PCA is a generic dimensionality reduction technique. Correlations between times series in climate data, and coherences in EEG data are implemented as problem/domain specific features. 66

Table 5.4  Cross validation accuracies for similarity-based learning algorithms using temporal similarities across spatiotemporal systems. ................................. 71

Table 5.5  Cross validation accuracies for classifiers using spectral graph features extracted from each network separately, and multi-dimensional scaling embeddings of graph edit distances. Highest and significantly improved statistics are displayed in bold. ................................................. 72

Table 5.6  Cross validation accuracies for (dis)similarity-based classification of graph edit distances after transformed into similarities through four different ($S_1$, $S_2$, $S_3$, $S_4$) linear and nonlinear transformation functions as listed in Equation 3.13. Highest and significantly improved statistics are displayed in bold. 77
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1</td>
<td>Overview of classification methods for spatiotemporal data.</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Vectorial embedding of three-dimensional spatiotemporal data into two-dimensional tabular format for the purpose of canonical data classification.</td>
<td>11</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Machine Learning Framework - Intertwined, three-step process: preprocessing, feature selection, and classification or regression.</td>
<td>13</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>Spatial and temporal filtering of spatiotemporal process (a) data is represented as it is acquired, (b) spatial locations that are making a difference in classification are selected, (c) time series data is filtered and summarized in a single point using temporal filtering techniques.</td>
<td>15</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Similarities between spatiotemporal processes: (a) first pairwise similarities at each spatial location ( s_i \in S ) are calculated separately between all sample (training) instances in ( X ), (b) calculated similarities for each spatial location ( s_i ) are represented as a separate similarity matrix ( K_i ), then (c) using a combination function ( f ) an aggregated similarity matrix is calculated and given, along with the corresponding class labels for each row, to a similarity-based classifier to build a classification model.</td>
<td>30</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Method overview for the proposed network-based classification framework: (a) each event with an outcome ( y_i ) is associated with a spatiotemporal process ( x_i ), (b) networks are generated based on the statistical correlations/coherences between time series, (c) the distances between pairwise network instances are computed using graph matching techniques and the resulting distance matrix ( D ) used as input to dissimilarity-based classifiers.</td>
<td>32</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Example 1 - edge distance calculation, spatial networks with a single edge.</td>
<td>36</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>Example 2 - edge distance calculation, spatial networks with multiple edges.</td>
<td>37</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>Metric and non-metric spaces and corresponding learning algorithms.</td>
<td>39</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Flowchart of the classification algorithms using dissimilarities ( D ) - embedded and kernel-based classification methods.</td>
<td>41</td>
</tr>
<tr>
<td>Figure 3.7</td>
<td>Classification using ( D ) as kernel, (a) dissimilarities are converted into similarities, then classified using (indefinite-) kernel-based approaches (b) dissimilarities are embedded into (pseudo-) Euclidian spaces then classified using vector-based classifiers.</td>
<td>44</td>
</tr>
<tr>
<td>Figure 3.8</td>
<td>Dissimilarity space embedding and classification (a) during training, a set of graphs are selected as part of the prototype set selection step, (b) testing instances are embedded agains this.</td>
<td>45</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Motivating Example I - (a) Patterns of North Atlantic and (b) Indian Ocean hurricane tracks.</td>
<td>48</td>
</tr>
</tbody>
</table>
Figure 4.2 Motivating Example II - Mental Transformation Task. First subject’s baseline signal is recorded, then a mental transformation task in 2d appears on the screen for 15secs, in the final screen, the subject is asked to choose matching object in 3d with the projected view.

Figure 5.1 Decision boundary for linear Support Vector Machines.

Figure 5.2 Decision boundary for nonlinear (kernel) Support Vector Machines.

Figure 5.3 Decision boundaries for filter-based feature selection algorithms over cognition dataset; xy coordinates represent features with the highest weights as a result of CFS feature selection. Local classifiers in (a) and (b) overfit the data in order to reduce the training error. Model-based parametric (c), linear (d), (e) and non-linear global classifiers (f)-(h) fit more generalizable decision boundaries but could not separate the data into two classes.

Figure 5.4 Decision boundaries for domain-specific (coherence) features on cognition dataset; xy coordinates represent coherence features with the highest weights as a result of CFS feature selection applied for visualization purposes. As local classifiers kNN (a) divides the space into local neighborhood regions, decision trees (b) divide the space into local rectangles. Parametric Naive Bayes (c) distinguish the data based on class distributions. Linear (d), (e), and nonlinear kernel classifiers (f)-(h) fit a global model on the data.

Figure 5.5 Performance comparison of local vs model-based classifiers on cognition data using coherences as features. On the left, the accuracies are shown when the same subject’s data is not allowed to be in both training and testing sets (heterogeneous data) at the same time, and on the right, random sampling where the test data is randomly selected among all instances and the data from the same user is allowed to be in both training and testing sets.

Figure 5.6 Decision boundaries for spectral features on the cognition dataset; xy coordinates represent eigenvalues corresponding to largest eigenvectors. The data is scattered in two dimensions. Local classifiers in (a) and (b) overfits the data in local regions, parametric (c) and model-based linear classifiers (d), (e) fit a global model to enable generalization. Nonlinear kernel-based classifiers in (f)-(h) create nonlinear decision boundaries for the data.

Figure 5.7 Classification decision boundaries for MDS embeddings; the coordinate system lies in the eigenspace where x and y axes correspond to the largest variations. The distribution of the data is mostly explained by the x-component. Local classifiers in (a) and (b) overfit the data. Parametric (c) and linear classifiers in (d)-(e) fit simple decision boundaries. Nonlinear classifiers in (f)-(h) create decision boundaries without gaining or loosing too much in generalization.
Figure 5.8  Similarity matrix representation for each sample dataset, similarities between instances acquired from (a) Cognition dataset, (b) Atlantic Ocean hurricanes, (c) Indian Ocean hurricanes, and (d) Graz-BCI experiment data.

Figure 5.9  Comparison of overall classification (SVM) accuracies of feature and similarity-based classifiers for each dataset. The average accuracies across all datasets are listed in the last column.

Figure 5.10  Statistical analysis of the results; (a) Kruskal Wallis test results: the mean accuracies and variances of four datasets from each method are drawn, (b) Post-hoc test: selected (highlighted in blue line) network-based approach is significantly better than any statistical and domain specific feature selection technique except correlation features and multidimensional scaling of edit distances (shown in gray.)
Chapter 1

Introduction

Analyzing dynamical systems from a physics point of view requires extensive amounts of information about system dynamics, such as the interactions between particles at a micro or reduced macro scale. Ideally, a dynamical system can be described by a set of differential equations [89]. However, some systems are too complicated to be described exactly, and some are not subject to such a reduction due to insufficient knowledge. In the literature, these are referred to as complex systems [52]. Typical examples of complex systems are (1) the human brain, where the cognitive functions such as learning or memory are outcomes of the dynamic interactions between distant neuron groups, and (2) climate, in which atmospheric, geographic and biological factors contribute to the changes in a nearly chaotic system. These two domains are the focus of this dissertation.

Data-driven statistical models that predict events by learning relationships with observed or simulated variables have had a significant impact on our lives, due to their success in many disciplines of science over the last several decades, in physics but also in a wide range of other domains [109]. With the increase of available data, accessible computational power, and the development of robust statistical models, we have seen a rapid transition in the treatment of problems that were formerly viewed as impractical to solve. As might be expected, data-driven models are also gaining ground in the analysis of complex systems, as an alternative or a complementary approach to current physics-based simulations.

Methods used in data-driven statistical analysis traditionally depend on the assumption that adjacent observations are independent and identically distributed. However, for the analysis of spatially and temporally extended dynamical systems, this is rarely
true. For instance, climate scientists collect oceanic and atmospheric data over a region from equidistant locations in predetermined time intervals, either by direct observation or using sensor satellite images. Economists constantly record daily stock market quotations to be able to follow the trends in financial markets. Brain imaging uses mapped electrodes attached to the subject’s scalp in order to collect signals with a fixed sample rate in a variable length experiment sessions. All of these datasets, constructed from complex systems, intrinsically have high auto-correlation and dependency between system variables. The systematic approach that aims to answer mathematical and statistical questions posed by these spatial and temporal correlations has its own unique branch in statistics and commonly referred to as spatiotemporal data analysis [102]. The first part of this dissertation explores a particular branch of spatiotemporal data mining, inference analysis in spatially and temporally extended dynamical systems.

Inference algorithms, in particular classification methods independent of the representation of the data they operate on (i.e., structured or tabular data), try to automate the classification process using a learned model built upon a sample prototype set; intuitively, such algorithms are comparable in some ways to our human ability to recognize general patterns by looking at examples. In this respect, the majority of general purpose classification algorithms start with a model generation step using a representative data called a training set in a particular domain. The goal for the learned model is to be able to generalize well to unseen real world examples contained in a test set. For instance, if the learning algorithm is designed to distinguish humans from the cars in a traffic video surveillance application, then we need a sample set of representations of cars and humans in our training algorithm, taken from existing footage. Then a model can utilize the derived characteristics or features of both classes to create a set of rules and equations, or it can look for specific templates to match; which approach is taken depends on the selected learning methodology and representation. In this respect, depending on the representation of objects in the example domain, the learning algorithms can be designed in two distinct ways: statistical and structural.

In the statistical approach, the model can be constructed either from the tabular raw data or from a set of features generated or selected from the raw data. (Feature generation and selection methods are also two distinct paradigms under the statistical machine learning framework and will be discussed in detail in later chapters.) In the car versus human recognition example above, the dimensions, weights, proportions, colors,
and so forth of each object can be declared as features and listed in tabular numerical form. Together, these form a feature (vector) space where each instance of the data can be represented as a point in $n$ dimensional Euclidian space. The calculated metric distances between these vectors feed a learning algorithm, in which the goal is to optimize the model by minimizing the targeted error function. Metric operations such as transformations, inner products, and defined norms that preserve the underlying Euclidian geometry give us a rich set of mathematical tools, resulting in a well established statistical learning theory. Nevertheless learning is inherently restricted to the generated/selected feature-vector space; it is therefore limited in terms of discovering possible structural relations between training instances.

Structural pattern recognition, on the other hand, mines the data in order to find primitives, definitive and relational substructures. Structural methods have been found most effective in domains in which inherent natural structure exists, such as molecular and protein datasets, handwriting recognition, fingerprint, and network classification problems [3, 46, 73]. In chemical compound classification, for instance, the existence or lack of a particular molecule within a structure may tell us whether a given chemical is part of a toxic family or not. Thus, in structural learning algorithms, features can be defined as substructures, primitives, motives, or communities, and the presence or absence of these features in a new sample determines the target class of that instance. However, the drawback of a discriminative motive searching approach is that learning is tied to binary-valued features. The lack of mathematical framework in decision making limits the broad use of structural learning algorithms in classification, despite their intuitive nature.

One of the main theories about human learning suggests that we learn from examples and that we differentiate things by looking at differences between objects, even if such differences are not explicitly defined in advance as features. This is in contrast to conventional statistical machine learning algorithms, which are designed to work with existing features [29]. For example, given an object, we can classify and label it based on how similar the object is to some reference point. This approach to considering implicit structural features leads to the formation of a class of hybrid learning methodologies called similarity- or dissimilarity-based pattern recognition algorithms, which typically utilize evolutionary learning principles to form a class description by stating that a class is a set of similar objects grouped together. Using any arbitrary proximity measure, a
(dis)similarity-based classification algorithm can be constructed without the need for preprocessing to extract or generate artificial features. In this respect, dissimilarity-based classification does two things: it combines the representative power of the structural and solid mathematical framework of statistical learning algorithms using the notions of proximity and dissimilarity space, and it eliminates the need for cumbersome feature generation or primitive substructure searching processes to a great extent.

In the second part of this dissertation, we explain the use of (dis)similarity-based classification algorithms to analyze complex network structures generated from domain independent spatiotemporal datasets using graph edit distance measures.

1.1 Problem Statement and Research Questions

A spatiotemporal process $x$ with an outcome $y$ occurring over a set of spatial locations and within a certain time period can be represented as

$$y \leftarrow x(s, t) : s \in S \text{ and } t \in T,$$

where $T$ is the time interval in which the process occurs and $S$ is the set of locations. If the goal is to predict the outcome of a future process $\hat{y}$, then using machine learning algorithms is the natural choice. However, the majority of modern machine learning algorithms are designed to work on tabular datasets where the inputs are composed of features $x = \{x_1, x_2, ..x_N\}$, with each ($x_i$) representing unique and independent characteristics of the data. The first research question is this: What techniques can be applied to fit spatiotemporal data into an existing standard machine learning framework, or what other alternative classification approaches can be utilized to predict the output of a spatiotemporal process (within the scope of this thesis)?

With advances in network theory, many complex systems are now being represented as networks. For example, the global climate system is represented by a grid of oscillators varying in some complex way [107]; the interactions between brain regions during task execution are represented as functional brain networks [13]. Recently, these networks have been analyzed with the goal of performing inference on them. The second research question is this: Can we use these network structures for the purpose of classification of events emerging from spatiotemporal processes instead of (or complementary to) feature-based approaches? Additionally, what are other possible techniques that we can utilize to classify such networks?
1.2 Thesis Statement

To answer the first question, we explore existing features and feature-based classification approaches in each domain (i.e. climate and cognitive science). Since the domains share similar spatiotemporal characteristics, we list the approaches that can be transferable from one domain to another and give an overview of methods to create a framework for common learning approaches. We show that spatiotemporal processes can be embedded into the standard machine learning framework by utilizing a set of spatial and temporal filtering techniques.

For the second research question, our hypothesis is that for inference in complex physical systems, network-based classification, where the interactions between distant spatial locations are the determining factor for overall system behavior, is a valid alternative to existing data mining algorithms for predicting emerging event characteristics. We also add that, for classification of spatial networks, (dis)similarity-based learning, using the proposed graph edit distance formulation, contributes to a more effective structural learning strategy compared to existing (spectral) graph embedding techniques.

1.3 Dissertation Contributions

The overall contributions of this research can be summarized as follows:

1. A unified framework has been created for classification in complex systems exhibiting spatiotemporal characteristics. Suitable and non-suitable approaches, challenges and gaps in the existing learning algorithms designed for classification in such systems have been identified.

2. Two distinct real-world emergent phenomena have been formulated as spatiotemporal (network) classification problems: (1) from the domain of cognitive science, specifically brain-computer interfaces (BCI), predicting cognitive state or expertise level of experiment subjects during task execution, and (2) from the climate domain, forecasting landfall behavior of hurricane trajectories.

3. A novel two-step graph edit distance calculation for spatial networks has been defined. In the first step, the concept of edge matching cost based on the spatial orientation
of the edges in a network structure is introduced, and in the second step, using the matrix of matching costs, graph edit distance between network pairs is formulated as an assignment problem.

4. It has been demonstrated that the graph-edit-based network classification approach outperforms existing machine learning techniques on problems in the two real-world spatiotemporal domains given above.

5. A detailed explanation of (dis)similarity-based learning approaches has been carried out, starting from arbitrary similarity or distance scores between instances in a prototype set, then following with directions for classification of those objects in a projected mathematical space. Each method has been analyzed to find suitable learning strategies for classification of spatial networks.

1.4 Dissertation Outline

The remainder of this document is structured as follows. Chapter 2 reviews existing research on spatiotemporal data classification. The methodology for network-based classification of such data is explained in Chapter 3, where an introduction to (dis)similarity-based learning approaches and corresponding space topologies is given. In Chapter 4, motivating examples and experiment datasets from two distinct spatiotemporal systems, brain-computer interfaces and climate, are introduced. Afterwards, evaluation metrics for current and foreseen results are discussed.
Chapter 2

Background Work

The first two sections in this chapter cover classification in spatial datasets and classification in temporal datasets, respectively. The third section is more extensive: it first describes feature-based approaches to classification in two domains, brain-related research and climate research; it then turns to network-based analysis of the same domains.

In Figure 2.1, a general framework for spatiotemporal classification in scientific datasets is given. Though the branches in Figure 2.1 represent existing approaches in the climate and cognitive science domains, the framework can be extended to any scientific data classification tasks for data that has both spatial and temporal dimensions. In the following sections each branch of the framework tree will be discussed through the example studies from both climate and cognitive science literature.

The majority of modern machine learning algorithms are designed to work on standard relational datasets where the inputs are composed of attributes, each ideally representing unique and independent characteristics of the data. As a general convention these attributes are called features and the algorithms utilizing those in learning referred to as feature-based statistical learning algorithms.

A well known example for feature-based classification on relational data is the IRIS dataset, where the features are the sepal and petal length and width information for three flower types [4]. The IRIS dataset is small and simple; for more complex datasets that may include higher dimensional features, another step, called feature selection or generation, might be required to filter out less informative attributes. A detailed overview of feature selection and projection methods for general purpose statistical learning algorithms can be found in well-known surveys [77, 69].
However, not all datasets are required to have independent and identically distributed discriminating feature sets listed in their respective dimensions. Many real-world datasets, especially scientific datasets, are recorded continuously in at least one of the space and time dimensions. Therefore adjacent recordings have dependency on each other; this is referred to as autocorrelation. In the analysis of spatiotemporal data, autocorrelation needs to be taken into consideration to account for the characteristics of the data and to produce objective results and meaningful analysis.

### 2.1 Spatial Datasets and Classification

Spatial data models using suitable geometric representations like points, lines, shapes and images define the properties of static objects in space. Environmental and ecological modeling using remote sensing satellite observations, geographic information systems (GIS), telecommunication and mobility applications utilizing GPS locations are well-known sources of vast amounts of spatial data. The increase in information resources and application areas has triggered the demand for effective machine learning algorithms for such data [99, 14].

For this reason, existing learning algorithms have been adapted for the analysis of spatial data. Khan et al. [61] use a modified version of the kNN (k-nearest neighbors) classification algorithm with a set of metric distances calculated between spatial images, to create a similarity score between training instances. Direct embedding of the spa-
tially distributed features sequentially in tabular form is another way to employ existing learning algorithms with such data. When it comes to the analysis of spatial features, however, overlooking autocorrelation that exists between training instances may cause poor inferences about the system being analyzed.

On the other hand, by incorporating spatial information into learning algorithms, more suitable and effective tools can be created for the analysis of spatial data [99]. Markov random fields (MRF) applied in image segmentation problems and spatial autoregression (SAR) models used in regional economy and ecological data applications are two examples of proven, successful implementation of spatial learning algorithms.

In SAR, the general logistic regression equation (Eq. 3.3),

\[ y = X\beta + \epsilon, \]  

is modified using the spatial contiguity matrix \( W \) and spatial dependency coefficient \( p \) (Eq. 2.2),

\[ y = \rho Wy + X\beta + \epsilon. \]  

The correction terms \( p, W \) and \( y \) introduced into the regression equation aim to remove systematic variations in the input data \( X \) and in the residual error function \( \epsilon \) due to spatial neighborhood relations.

A Markov random field-based Bayesian classifier is a model applied to image segmentation and land-use classification problems. Experience with such models suggests the importance of a spatial variable conditioned on the events only in its direct neighborhood.

In classification using Naive Bayes, the probability that a given instance \( X \) belongs to a class \( c_i \) can be computed from the conditional probabilities existing in the data (Eq. 2.3):

\[ P(c_i|X) = \frac{P(X|c_i)P(c_i)}{P(X)}. \]

MRF modifies this formula by incorporating the neighborhood class label function \( L \) to estimate the posterior probability of a point belonging to class \( c_i \) in the presence of spatial dependency:
\[ P(c_i|X, L) = \frac{P(X|c_i, L)P(c_i|L)}{P(X)}. \] (2.4)

2.2 Temporal (Time Series) Datasets and Classification

Temporal data, also referred to as time series data, captures changes in a variable over a time interval \( T \). The analysis of temporal data is studied extensively in statistics because of diverse resources in data and needs in application areas [101]. Sensors recordings of the human body for cardiovascular activity monitoring, safety measurements of a nuclear power plant, daily stock market values on Wall Street, and periodic sales numbers in a company are just few examples of time series data.

As in spatial analysis, time series classification also faces challenges due to the nature of the data. The characteristics of time series data include high dimensionality, temporal autocorrelation, noise, nonlinearity and nonstationarity [2]. Learning algorithms working in this domain needs to account for these challenges.

In general, time series classification algorithms are divided into three categories: feature-based, distance-based and model-based algorithms [33, 113].

The first approach preprocesses the data to generate a set of features, typically by utilizing domain or expert knowledge. Generic machine learning algorithms that perform well on standard datasets can also be used in feature-based time series classification tasks [120].

The second approach frequently employed in sequential data classification is to compute pairwise distances, which can either be standard (i.e. metric, such as Euclidian distance) or non-standard (i.e. non-metric, as in dynamic time warping [62]), between time series [9]. The calculated distances are given as input to a neighborhood-based classification algorithm. Although this approach has been criticized for being primitive since it does not involve any model building, it stands as an effective learning strategy in classification.

As a third approach, model-based learning algorithms are employed in classification of time series data. Hidden Markov Models and Neural Networks are promising for dealing with intrinsic challenges introduced by temporal data since they cope with the temporal aspect of the data internally through state transition diagrams [43]. These techniques
are more sophisticated compared to feature and distance-based algorithms, requiring in-depth analysis of the selected parameters prior to model building [66].

2.3 Spatiotemporal Datasets and Classification

In spatiotemporal datasets, variables extend in both spatial and temporal dimensions. Datasets originating from ecological and environmental dynamics, climate, social and daily interactions like transportation and traffic, physiological recordings of human body and brain such as functional magnetic resonance imaging (fMRI), electroencephalogram (EEG) and magnetoencephalography (MEG) are all examples having spatiotemporal characteristics. Despite advances in recording technologies and the emergence of new application areas, a general framework is lacking for spatiotemporal data classification methodologies in the current literature. One of the aims of this work is to draw a general picture of machine learning algorithms for spatiotemporal systems.

Machine learning in general is an intertwined, three-step process: preprocessing, feature selection, and classification, as pictured in Figure 2.3. Depending on the first two steps, the nature of the data may change, and therefore the type of classification algorithm that is most appropriate may change as well. For example, noisy data can be
preprocessed into noise-free data, so that a more complex algorithm can be employed without fear of overfitting. If the high dimensionality of input data is reduced using a feature selection algorithm, then a less robust, fuzzy classification algorithm can be employed. Selection of a classification method therefore depends on how the data is being processed and which features are extracted from it. This may explain the reason for the lack of a general framework for classification in spatiotemporal analysis, since domain-dependent preprocessing and feature selection is determining factor for selection of a classification approach.

A common but usually inefficient approach to any type of spatiotemporal system is to reorganize each instance of data into a one-dimensional vector representation called a vectorial embedding, as shown in Figure 2.2. Mourao et al. [78] applied vectorial embedding to fMRI data recorded in discrete time intervals at separate marker locations and fed the resultant data to a Support Vector Machine classifier as input to create classification model. The model discriminates between recordings of people assigned to perform a cognitive task from people in resting state. The algorithm is named “Spatial-Temporal SVM.” This is slightly misleading, because the algorithm does not incorporate any spatial or temporal information such as autocorrelation into its model-building step. The results of their study do not show any significant improvement over the comparison method that uses features generated by taking the mean values across temporal dimension. The usefulness of vectorial embedding in analysis of spatiotemporal data has not yet been proven.

Another commonly applied strategy for spatiotemporal data classification is to filter out one or both of the space and time dimensions of the input. This can be done by averaging over the temporal dimension, or by transforming time series data into more compact forms such as spectral coefficients, or by selecting fewer spatial locations as representative features. These examples of temporal and spatial filtering techniques are illustrated in Figure 2.4. However, these transformations are application dependent and requires domain knowledge in order to apply safely. In the reminder of this section, representative techniques for classification of spatiotemporal datasets contributing to this research will be investigated within their respective domains. Each section begins with a brief introduction to important classification problems in the domain, then continues with an overview of the current state of the art in analysis techniques.
Figure 2.3: Machine Learning Framework - Intertwined, three-step process: preprocessing, feature selection, and classification or regression.

2.3.1 Classification of Human Cognitive States using EEG

Cognitive state classification is an important application area for brain-related research and has many application areas. Designing brain computer interfaces, creating feedback loops for advance training systems such as driving and flight simulators, allowing disease diagnosis and prevention are just a few examples.

The data mining problems addressed by EEG-based cognitive science studies include sequential pattern mining (to deal with extracting repetitive patterns from data such as eye or muscle movements), associating relations of these patterns with cognitive brain functions [60], and outlier detection for differentiating abnormal brain states from normal brain functions (such as epileptic seizures [35, 50] or schizophrenia [67]). In this research, we focus on inference analysis (i.e. classification algorithms) for EEG-based cognitive architectures.

Extraction of human cognitive states for the purpose of the above-mentioned applications requires experiments conducted under laboratory conditions. While some applications collect data from subjects for diagnosis purposes in everyday non-laboratory settings, others follow experimental procedures known as the synchronous and asynchronous paradigms.

In asynchronous systems, a subject makes self-paced decisions to switch between available cognitive actions. The system is designed to capture and act according to these mental changes. In the more common synchronous systems, experiments are divided into fixed duration blocks called epochs. Each epoch is associated with the targeted task being performed. For instance, a task might be to imagine oneself moving to the left or to the right; each decision interval is associated with a clue shown to the subject on a computer screen. From the EEG data recorded, a system tries to differentiate between the two intended directions.

The input signal generated by those processes is an example of three-dimensional spatiotemporal data consisting of $M$ epochs, from $N$ EEG channels, recorded over a
fixed time interval $T$.

**Feature Selection for BCI Datasets**

The goal of feature selection in machine learning is to find a small subset of attributes that sufficiently represents a larger set. However, in some types of data, including EEG, such information is spread across all the dimensions of the input; therefore finding such a subset may not be possible using standard feature selection techniques based on statistical correlations. Handling feature selection in each dimension (i.e. spatial and temporal) separately and introducing an intermediate step (i.e. feature identification) carried by subject experts to account domain-specific characteristics is a common strategy applied for this type of data.

**Temporal Filtering**

One of the fundamental theories about the electrical activity of the brain [6] suggests that our brain rhythms change in accordance with the cognitive operation the brain performs. For instance, during the resting awake state, the brain follows a posterior basic rhythm called alpha modulation (7-14 Hz). During (imaginary) motor movements, the beta rhythm (15-30 Hz) is attenuated. Cognitive functions, on the other hand, activate higher frequency gamma waves (30-100 Hz). These modulatory characteristics of the signal allow for identification of frequency domain features in EEG-based systems, because frequency conveys information about the rhythmic activity of the brain during task execution. As a side effect, identification of this spectral component rules out the temporal dimension of the input signal by summarizing the data collected over a course of time in a single amplitude value.

**Spatial Filtering**

In addition to frequency domain transformations, spatial filtering is another common technique for feature selection in EEG data. The goal of any spatial filtering is to determine an $N$ dimensional unmixing matrix $W$ weighted towards more important spatial components. When this matrix is applied to input signal $x$ the result is a new signal $y$ with emphasis on more discriminating channels (Eq. 2.5).

$$y(t) = Wx(t) \quad (2.5)$$
A commonly used spatial filtering technique in the EEG domain is independent component analysis (ICA), in which linear projections of the signal are computed to generate statistically independent components [7]. In ICA, for $N$-channel EEG data, unmixing matrix $W$ consists of $N$ rows, each of which corresponds to an independent component in the signal. Selecting the first $M$ of these $N$ rows will reduce the dimension of the input signal by a factor of $(1 - \frac{N}{M})$. In many applications, ICA is simply used for artifact removal, due to its success in separating eye blinks and muscle movements from the raw signal; therefore spatial filtering part is mostly achieved by its supervised successor, the Common Spatial Patterns (CSP) algorithm.

CSP is a supervised spatial filtering technique proven to be effective in a number of EEG-based brain computer interface applications in comparison to previous approaches [12]. The underlying principle of CSP is to maximize the variances between signals that belong to separate classes via simultaneous diagonalization of the signal covariance matrices. The first and last entries of the resulting projection matrix $W$, sorted in descending order of the eigenvalues, explain the channels with largest variance and therefore highest discriminative characteristics.
Inference in EEG-based Systems

Although the spatial and temporal feature selection methods explained above can be used as standalone techniques, they are usually applied in combination with each other. For instance, frequency transformations can be performed after selection of discriminating spatial channels. However, the ultimate success of the learning algorithms depends on the selection of the right classification strategy. Lotte et al. [70] groups classification methods in machine learning domain according to their distinct characteristics: generative versus discriminative, static versus dynamic, stable, and regularized algorithms. Since EEG data is noisy and contain outliers, the regularization that minimizes the effect of outliers and limits the unnecessary complexity of the model is stated as a required component for this type of data. Nevertheless, researchers have tried a majority of the state-of-the-art classification techniques with different feature selection strategies for EEG data. Among others, Support Vector Machines (SVM) are repeatedly reported as a suitable and successful approach [79] due to SVM’s regulatory and generalizable characteristics. In addition to SVM, dynamic classifiers such as Hidden Markov Models (HMM) is also reported as an effective technique for the current domain by Obermaier et al. [85], because of their ability to cope with temporal aspect of the data inside the model.

2.3.2 Classification and Predictive Modeling in Climate

Climate is a system governed by physical, chemical and biological principles; the behavior of the system can be approximately predicted using set of differential equations known as model simulations. The complexity, resolution, initial conditions, variables and processes included in a model result in different simulations whose predictions about the distant and close future may vary marginally. One solution for the margin of error problem is averaging the simulation outputs to make commonsense predictions from model ensembles [103].

While model simulations base their predictions on a set of initial conditions which requires limited data to build upon, the size of climate data is actually enormous and growing exponentially. Remote sensor satellite images and output of model simulations such as reanalyses [54], in which all the available but noisy heterogeneous observational data is put through numerical simulations to generate large scale homogeneous datasets, are the main sources of the large volume of climate data. This opens the gates for data-
driven analysis of climate as an alternative or complementary approach for physics-based simulations. However, the nature of climate data is spatiotemporal and there is a gap in learning algorithms in data mining and machine learning to account for challenges such as high dimensionality and auto- and cross correlation between input variables.

Active research problems in climate domain are grouped into four main categories by Faghmous and Kumar in their recent study [32]: outlier and event detection, relationship mining, pattern mining, and predictive modeling. In outlier detection the goal is to model and differentiate anomalous events from the data such as the ones that have drastic effects on the ecology like significant land cover changes [56], droughts or abnormal temperature shifts [72].

Pattern mining tries to find motives or data clusters that share similar structure and exhibit similar behavior that could also give greater insights about the rest of the data. Pattern mining approaches in the climate field are applied to discover the dominant signal in the high dimensional data via empirical orthogonal function (EOF) [5] and Independent Component Analysis (ICA) [75] to find clusters exhibiting similar behavior such as spatiotemporal neighborhoods. An interesting study is on finding hurricane track patterns and identifying their characteristics across each distinct class of hurricanes [81]. Finding oceanic eddies is another open research problem because of their importance in carrying information across oceans and forming teleconnections among geographic regions [32].

Relationship mining applications seek out relationships between some variables and target events. For example, researchers have attempted to find correlations between sea surface temperatures and global land surface temperature values [58] and fire patterns in the Amazon [16]. They have also related vertical wind shear in the Atlantic Ocean to rainfall patterns in the African Sahel region [39]. Kim et al. [62] applied composite analysis to relate five years hurricane data to warming patterns of the Pacific. McCloskey et al. [74] take a different approach and relate Bermuda high pressure values to the curving patterns of Cape Verde hurricanes.

These applications focus on explanations of phenomena. In predictive modeling, on the other hand, given a set of data forming a process the goal is to create a model for the process and to find the output of future processes by looking at the data. Machine learning techniques are common in predictive modeling, such as in classification, by assigning the process to a class label, or in regression, by finding real-valued output as a result of a
learned equation. In the next subsections, the components of predictive modeling, feature selection and learning algorithms applied for climate data will be covered.

**Variable and Feature Selection for Climate Data**

For climate, spatiotemporal variables such as daily sea level pressures (SLP), sea surface temperatures (SST), vertical wind shear (VWS) and numerous other variables are continuously recorded by sensors (i.e. *in situ* and by satellite) and assimilated into model simulations by major climatology institutions like NCAR-NCEP [37] and NASA [38]. Data-driven predictive models utilize these variables either by directly incorporating them into their analysis [19], filtering relevant ones using statistical feature selection algorithms or through generated climate indices [49]. Feature selection, which can be a separate process defined by the domain experts or part of the learning algorithm, is one of the key factors for the success of a learning algorithm. The generic statistical feature selection algorithms will be covered in Chapter 5 in more detail. Climate indices computed using normalized values of atmospheric and oceanic variables such as SLP, SST etc. within selected local regions summarize climate behavior on a global scale. Defining climate indices as features serves two purposes in the analysis of climate data: (1) climate indices act as a feature set proven to be significant, with a core importance for any machine learning application, and (2) climate indices eliminate the spatial and also to some extent the temporal dimension of the data, which is an essential filtering step for the classification of spatiotemporal data.

**Inference Analysis for Climate Data**

To forecast the precipitation in a region, Coe and Stern [21] build a second-order Markov chain with the help of the temporal properties of historical data of the region. The output of the process predicts whether the corresponding region will have a high or low precipitation in any given time of the year. Regression algorithms are also employed to predict changes in tropical forest cover, the amount of rainfall in specified regions such as the African Sahel region, and temperature changes in global scale.

Forecasting hurricane counts and landfall behavior is another active research question and is also within the scope of this research. For hurricane related predictive analysis, the common question asked by researchers is whether the number of tropical cyclones occurring in a target region (e.g. North Atlantic or Pacific) can be forecast accurately prior
to a climate season. Elsner et al. [31] use May-June values of several climate indices as input features for Poisson regression analysis and find that the North Atlantic Oscillation (NAO) index is the leading factor in predicting the number of hurricanes hitting coastal regions of the US in a given hurricane season.

Chand et al. [15] model the prediction of tropical cyclone (TC) activity in the southwest tropical Pacific islands as a binary regression problem where the output values consist of high and low activity and the covariates of the regression are determined using Pearson correlation analysis. They report that the preseason vertical wind shear (VWS) and vorticity values of the target and sea surface temperatures (SST) values of Nino 4 regions are potential predictors for TC activity for the upcoming season. Similarly Chu et al. [20] employ correlation analysis to choose the regions where preselected features (i.e. SST, SLP VWS, PW and vorticity) show the highest correlation with North Pacific cyclone activity. The specified variables in selected regions are used as covariates in Bayesian regression analysis to predict the number of hurricanes before a season begins.

The predictive models in the climate literature reviewed in this section either choose predefined climate indices or select custom location and time periods to build index-like structures to analyze the relations between data and target events. The reason for this is that using climate indices as features simplifies model generation by avoiding the additional complexity introduced by spatial and temporal dependencies in the data, such as auto- and cross correlations, as well as spatial and temporal resolutions.

However, algorithms that can incorporate spatial and temporal relations within the data may shed light on the uncertainties that current models fail to express. There is also room for improvement in current spatiotemporal analysis techniques in favor of the novel approaches taking underlying characteristics of data into account such as structural dependencies and network-like behaviors.

### 2.4 Network-based Approaches

Recent advances in complex network analysis such as the formalization of small-world, scale-free properties and the discovery of these phenomena in many real world physical, biological, and social systems have changed the way that these systems are being analyzed. Movement is towards network-based approaches rather than traditional feature-based methods, because of their ability to incorporate structural information into model build-
ing. In the following two sections, we investigate network-based approaches in complex system analysis, specifically analysis of brain and climate dynamics. As in the previous section, below we briefly introduce problems the domain specific to network-based analysis and then outline the state of the art.

2.4.1 Brain Networks

The brain is one of the most complex systems known to mankind. Researchers have been trying to model, learn and replicate the inner workings of this complex system for decades.

Structural brain networks

It is well-established theory that neuronal elements such as neurons, axons, and synapses form a structural network known as connectome. The first anatomical nervous system connectome that has been modeled completely [117] belongs to a free-living (non-parasitic) roundworm, about 1 mm in length, called *C. elegans*. The corresponding neural network forms a graph with 302 neurons (i.e. nodes) and on average 14 synaptic connections per neuron (i.e. edges). It exhibits a small-world topology with a path length close to a random network and high clustering coefficient similar to a regular network [116].

More complex biological connectomes, for example the human brain, have enormous complexity with billions of neurons and trillions of synaptic connections. Currently, complete modeling of the human brain, to the same degree as *C. elegans*, is not a feasible goal. To date this remains a grand challenge. However, the structural connectivity of the human brain is still studied in nested spatial resolutions [104].

At the microscale, researchers use electron and light microscopy to map neuronal connections at the cellular level by reconstructing neuronal processes from volumetric electron microscopy (EM) images, with the help of segmentation algorithms [112]. At the mesoscale, anatomically distinct brain regions and connecting pathways are traced along the axonal projections using injected neuroanatomical markers to deliver whole-brain connectivity maps; this has been done for several nonhuman species [10]. At the macroscale, the connectional anatomy of the brain is being modeled using noninvasive neuroimaging techniques such as magnetic resonance imaging (MRI) that utilizes statistical correlations between cortex thicknesses of different brain areas [45], and diffusion
tensor imaging (DTI) that uses diffusion of water molecules to produce neural tract images across brain regions [53]. All these studies report similar results, finding the small-world property of brain networks at different spatial resolutions. This may lead to an inference that brain has evolved to an efficient and robust structure in which distributed specialized regions are responsible for delivering specific cognitive functions such as motor movements, speech, and vision.

**Functional brain networks**

Structural networks determine the static infrastructure of the brain. They are the highways between different brain areas and put physical constraints on cognitive brain functions. However, structural networks cannot fully explain the dynamic pathways activated during task executions. Therefore, functional networks also are a main topic of cognitive and clinical studies. Their investigation, separate from structural networks, focuses on understanding emerging activation patterns of the brain to reveal information beyond the anatomical structure.

Although continuous advances in brain imaging technologies provide new techniques to observe functional behavior of the brain, fMRI, MEG, and EEG are the most common ones used in clinical studies because of their noninvasive nature and solid theoretical foundations. Despite the fundamental differences in what these techniques measure in relation to activation, the network construction philosophies are similar. The functional networks generated from statistical interdependencies between activity patterns (i.e. electrical and magnetic responses for EEG and MEG and hemoglobin level in the blood for fMRI) reflect functional interactions between brain regions. For instance, statistically significant coherent activity in two sensor locations determines the existence of a functional edge between them.

Cognitive studies using brain imaging take the snapshot of the brain under comparative cases (for instance, a resting state versus task execution, or a healthy versus a dysfunctional brain) and then construct a network for each case. Most studies of functional brain networks point towards the direction of small-world characteristics; however, there are few studies that report findings supporting scale-free properties as well.
Inference analysis using brain networks

Practical applications use graph-theoretic analysis of functional brain networks to generate inferences about the brain. In inference analysis, the goal is to classify and characterize changes in the brain due to pathological disfunction (e.g. Alzheimer’s Disease, schizophrenia, epilepsy) for preventive and diagnostic purposes, or to detect modulations under different task executions in experimental settings, especially in cognitive science research.

There are two distinct paradigms for classification of brain networks, based on (1) topological features and (2) structural similarities. Another approach can be added, (3) embedding methods [91].

Classification via Topological Features

Topological properties characterizing global and local connectivity patterns of the brain, which are mainly used for knowledge discovery purposes, are also utilized as feature sets for machine learning algorithms. These features can be evaluated locally at the individual element level such as nodes or links, or at a larger scale (i.e. globally) for the entire network. For instance, as a local measurement, the degree of an individual node (i.e., the number of links connected to a particular vertex) determines the relative importance of that node; as a global measurement the degree distribution (i.e., the average node degree across the entire network) characterizes the resilience property of the network.

Rubinov et al. [95] give an extensive list of these network measurements at local and global scales. They group measurements according to what they characterize, such as the level of functional integration and segregation, centrality, resilience and connectivity. In inference analysis, these measurements are treated as outputs of a feature selection algorithm and fed into the classification and regression algorithms. For example, toward early diagnosis of Alzheimer’s disease in an EEG connectivity study, Dauwels et al. [24] use global synchrony and Granger causality values to distinguish mild cognitive impairment (MCI) patients from an age-matched control group.

Several studies combine vertex level (with cardinality $|m|$) and graph level (with cardinality $|k|$) features for a brain connectivity network (consisting of $|N|$ vertices) to identify a feature set consisting $|N \times m + k|$ dimensions [27, 30] and then apply another selection step to filter the most discriminative features for classification.

One drawback of topological analysis of brain networks is the unique characteristics
of constructed graphs, which may not be suitable for some of the topological feature sets such as average path length, because the artificially constructed networks are usually sparse and disconnected. Therefore it is not atypical to rule out some general-purpose topological features.

**Classification via Vectorial Embedding of Graphs**

Richiardi et al. [90] extract the upper triangle of weighted adjacency matrix $\beta$ of a underlying network $g_i$ to represent the information in feature vector form $F(g_i)$:

$$F(g_i) = \{\beta_{1,2}^i, \beta_{1,3}^i, ..., \beta_{n-2,n-1}^i\}^T. \quad (2.6)$$

Then a training set

$$F = \{F(g_1), F(g_2), ..., F(g_k)\} \quad (2.7)$$

composed of $k$ training graphs with dimensionality $\frac{|V|^2 - |V|}{2}$ is used as input to a feature selection algorithm.

As an alternative to direct embedding of an adjacency matrix, spectral embedding using singular value decomposition of $\beta$ can be applied to generate lower dimensional projection of input graphs [92]. The calculated projections (i.e. principle components) are given as input to standard machine learning algorithms for classification. Results show that applied dimensionality reduction technique improves the accuracy of adjacency matrix embedding in cognitive state classification task; however, no comparison with other graph classification algorithms is available.

**Classification via Graph Kernels**

Another family of learning algorithms for structural data classification is graph kernels, derived from kernel methods applied in vectorial data classification. The idea behind kernel methods for vectorial data is to find a real valued pairwise similarity value $k(x_i, x_j)$ for training instances in $X$ through a mapping $\phi : X \to H$: [98],

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_H \quad (2.8)$$

For tabular datasets, learning with kernel machines is well-established theory in the machine learning literature [47]. The “kernel trick” states that as long as we have a positive semi-definite (p.s.d.) kernel matrix $K$ (a.k.a. a valid kernel) representing pairwise simi-
larity values, we do not need to calculate or formalize the underlying mapping function \( \phi \) explicitly. The calculated kernel values will correspond to an inner product \( \langle \cdot, \cdot \rangle_H \) in some Reproducing Kernel Hilbert Space (RKHS).

In the last several decades of analysis of structural data, such as protein interaction, gene expression, and chemical compound classification tasks, many successful graph kernels have been suggested [36]. For structural data represented by graphs (i.e. \( g_i \) and \( g_j \)), the general kernel formulation in Eq. 2.8 becomes

\[
  k(g_i, g_j) = \langle \phi(g_i), \phi(g_j) \rangle_H, \tag{2.9}
\]

where \( \phi \) is a mapping \( \phi : \mathcal{G} \to \mathcal{H} \) from graph domain \( \mathcal{G} \) to a Hilbert space \( \mathcal{H} \).

For structured data, the proposed kernels can be grouped in three categories [100]: random walk and path-based kernels, limited size subgraphs-based kernels, and subtree pattern-based kernels. However, the suitability of these kernels for brain graphs is questionable, because the suggested kernels are engineered to find patterns occurring in any orientation in graphs such as protein interaction networks [112] or to utilize statistical properties of walk lengths occurring in large graphs such as World Wide Web networks [55]. Brain networks, on the other hand, have special characteristics.

In a brain network, each node has a unique node label associated with the region where the recording take place, and the graphs generated through the same experimental procedure have the same vertex correspondence. Nodes can also be represented as spatial coordinates rather than vertex labels; therefore, for kernel functions defined for functional brain networks to be relevant, a plausible requirement is that it incorporate spatial information into the kernel formulation in addition to structural resemblance. As another domain-specific challenge, brain recordings such as EEG or fMRI may include higher noise-to-signal ratio, manifested as missing edges in the network structure. There is a clear need for kernel methods engineered specifically for brain connectivity graphs. Recently, kernel suggestions for this type of networks have started to emerge, especially for fMRI data.

Mokhtari et al. [76] attempt to employ shortest path kernels utilizing walks with length one in combination with SVM classification, plus backward feature elimination algorithm, to improve the classification accuracy using the most discriminating edges of an fMRI connectivity graph. Pons et al. [100] propose Weisfeier Lehman kernels based on subgraph isomorphism for brain networks because of their expressive power on graph
similarity measurements and efficient computational time complexity. However, their results show that the suggested kernel evaluates the structural property of the underlying pattern better than location of the pattern, an important issue for graphs where spatial information has crucial importance, as with the brain [114].

Convolution kernels [44] applied to composite objects such as graphs can be interpreted as similarity functions defined over smaller units (e.g. individual nodes) of an object. Overall similarities are computed using the convolution operation:

\[
k(g, h) = \sum_{g_i \in g, h_i \in h} \prod_{1 \leq i \leq d} k_i(g_i, h_i).
\]

(2.10)

Takerkart et al. [108] use walks of length one as the parts forming composite brain graphs and define three similarity measures: structural, geometrical, and activation kernels. Using geometrical distances between edges makes the kernel suitable for spatial graphs, but utilizing the activation maps rather than functional connectivity limits its applicability only to the graphs generated from fMRI data.

In general, despite their theoretical validity, the challenge for graph kernels stems from their expressive power. It is not known whether a kernel function using local structural information can represent enough information about the overall graph similarities for the purpose of classification.

**Classification via Graph Distances**

While graph kernels are used to find efficient and expressive statistical measures such as walks on substructures of composite objects to express pairwise similarities, a conceptually similar method, (dis)similarity-based learning, takes a direct approach by calculating custom distances between graph topologies for the purpose of classification.

Richiardi et al. [93] compare the classification accuracy of vectorial graph embedding versus dissimilarity space learning techniques for fMRI connectivity graphs. For vectorial graph embeddings, they use upper triangle of weighted graph adjacency matrix \(\beta\). For dissimilarity space learning, distances between two graphs \(g, g' \in \mathcal{G}\) are calculated using the difference between edge weights:

\[
d(g, g') = \sum_{e_{i,j} \in g, g'} |\beta(i, j) - \beta'(i, j)|.
\]

(2.11)
In their study, using vectorial embedding of graph adjacency matrix outperform the proposed dissimilarity-based learning algorithm that uses the distance metric above. However, the primitive distance function used in dissimilarity computation takes all edge labels into account, including spurious ones; it can plausibly be improved towards better metrics.

### 2.4.2 Climate Networks

A network-based approach opens a new perspective for analyzing complex climate systems by capturing system dynamics within a single framework. Current applications of climate networks in data mining are for discovery of new and existing climate indices and association of dynamic changes in network structures to seasonal climate.

#### Network Construction

The intuition behind climate networks is that the global climate system can be represented by a grid of oscillators varying in some complex way [107]. Early studies in climate network analysis focused on the construction of global climate teleconnection patterns that are resilient over long periods of the time.

Tsonis et al. [111], inspired by complex network theory, were the first group to propose the idea of building climate networks by formulating teleconnections as linear correlations between signals at distant locations. Results of their study suggest that there exist stable networks through which information is transferred and fluctuations from this stable infrastructure may lead to major climatological changes. Their analysis also shows that the global climate network cannot be explained using only one type of network property such as scale-free, small-world, or random, because there exist subregions exhibiting different network topologies locally; for example, a tropical region may be fully connected, or higher altitude regions may have scale-free properties.

In addition to linear methods such as Pearson correlation, networks built from nonlinear interactions have also been considered. Donges et al. [26] created a 20th century global climate network from a multi-model simulation and global surface air temperature data (SAT), using nonlinear mutual information and betweenness centrality measures instead of linear correlations; their work reveals novel and significant patterns. They call the resultant structure of their study the “backbone of the climate network.”
Climate Index Discovery

A crucial component of climate research is climate indices, discovered early on by observation and more recently through eigenvalue analysis techniques such as SVD and PCA. Climate indices are accepted as the global impact points on earth climate and are also studied from the network point of view. Steinbach et al. [106] applied a shared nearest neighborhood algorithm to demonstrate that time series data from cluster centroids correlate well with known climate indices such as North Atlantic Oscillation (NAO) or Southern Oscillation Index (SOI), and centroids whose time series values do not correlate to any of the known indices are candidates for new potential climate indices.

Pressure anomalies having opposite polarity appearing at two different locations on earth at the same time are known as dipoles. Jawale et al. [106] propose cluster-based discovery of dynamic dipoles within the network that can be interpreted as spatially moving climate indices occurring in a sliding time window, as an alternative to indices build from fixed locations. It is reported that the dynamic dipole-based approach estimates land temperature values better than the corresponding SOI.

Capturing Dynamic Changes and Association Mining

The basic network construction and analysis techniques that use all available data to construct a single steady network across several decades may not be capable enough in addressing seasonal changes in climate. A recent trend in complex climate network analysis is to construct dynamic networks at various temporal resolutions and associate them with emerging climatological events.

A network construction method proposed by Tsonis et al. [110] uses four major climate indices as nodes to create a climate network with fewer links. The complex network analysis shows a high synchronization between these index variables; however, in periods when the coupling between them increases, the synchrony is disturbed and a new state emerges in the climate. They conclude that the observable climate goes through temporal variations which in turn affect global climate variables and structures.

Guez et al. create annual climate networks from air temperature and geopotential-height field values. Correlation analysis demonstrates that NAO variations influence the number of links in the network and therefore act as an indicator of dynamically changing structure [41]. Berezin et al., in a more recent study, observe fluctuations in network topologies associated with the different phases of climate. They also report that links
in equatorial regions are more susceptible to these changes than links in non-equatorial regions [8].

Another challenge in network analysis of climatological events is the scale of the constructed networks, which usually covers the entire globe. The location of an event in focus, however, may be affected by only a small portion of a global phenomenon. From this perspective regional networks higher in resolution but smaller in scale may be required for inference analysis and association mining applications. In a recent study, Yamasaki et al. create networks from temperature measurements at four separate geographical regions around the globe to observe dynamic changes in topology of subregions related to global temperature variations induced by El-Nino. They develop a method to observe fluctuations in network structures appearing as “blinking” links over short time intervals. Their results show that structural changes can be observed even within the regions whose temperature values are not affected by El Nino, especially on type of links that require longer time delays to form. They conclude that variations in climate (for instance due to the El Nino’s effect) are correlated with the changes in teleconnection patterns at regional scales [121].

**Inference Analysis for Climate Networks**

Although discovery of moving climate indices and associations between dynamic changes in network structures and emerging event datasets may serve well as a feature selection tool for the purposes of statistical learning theory, to our knowledge direct application of network structures to inference has not been done in general. The research described in this dissertation is one of the earliest examples in the climate domain.
Chapter 3

Proposed Method: Similarity-Based Classification of Spatiotemporal Datasets using Networks

In this section we propose a new methodology for classification of spatiotemporal data from a real-world, complex system using similarity-based classification techniques. Our approach is based on an argument that systems sharing the same underlying spatial structure but exhibiting different behavior at different time frames should be grouped based on their output behavior, and the underlying data should have similar characteristics within each group. Therefore, instead of looking for a specific signature, i.e. features that can be seen as the cause of the emerging behavior, this notion of similarity can be used to classify the spatiotemporal systems and allow us to predict future events by looking at the data within a period of time.

We propose two approaches to classify spatiotemporal systems using similarities. As a first, direct approach, the time series of spatiotemporal systems can be compared at each spatial location and their aggregate similarities can be calculated as a similarity score between two systems. Following the second approach, systems can be represented as structural entities such as networks, and these structural entities can be compared instead of individual time series. In the following two sections the reasoning behind each approach will be explained in detail.
Figure 3.1: Similarities between spatiotemporal processes: (a) first pairwise similarities at each spatial location \( s_i \in S \) are calculated separately between all sample (training) instances in \( X \), (b) calculated similarities for each spatial location \( s_i \) are represented as a separate similarity matrix \( K_i \), then (c) using a combination function \( f \) an aggregated similarity matrix is calculated and given, along with the corresponding class labels for each row, to a similarity-based classifier to build a classification model.

### 3.1 Similarities Between Spatiotemporal Systems

A spatiotemporal process \( x_i \) occurring over a set of spatial locations \( S \) during a discrete time period \( T \) can be represented as

\[
x_i(s, t) : s = s_1, s_2, \ldots s_{|S|} \text{ and } t = t_1, t_2, \ldots t_{|T|}.
\]

For example, the occurrence of natural phenomena such as hurricanes can be represented by a tuple \((x_i, y_i)\), where the \( y_i \) is the output of a process \( x_i \). The goal is to classify such phenomena by looking at the underlying spatiotemporal process \( x_i \in X \) and comparing with other processes \((X \setminus x_i)\) occurring in the same spatial region at different time...
In the first approach, as illustrated in Figure 3.1, the similarity between two process \( x_i \) and \( x_j \) at a spatial location \( s_k \in S \) can be calculated as the correlation (or coherence for EEG signals) between two time series at that location.

\[
sim(x_i(s_k), x_j(s_k)) = corr(x_i(s_k), x_j(s_k)) \tag{3.1}
\]

Then an overall similarity score between two spatiotemporal process can be represented as a function of similarities over all spatial locations.

\[
sim(x_i, x_j) = f(\{\sim(x_i(s_k), x_j(s_k))\}_{k=1}^{\|S\|}) \tag{3.2}
\]

In this respect, similarities can be perceived as kernels. Inspired by multiple kernel learning theory [40], the combination function \( f \) can be implemented as a simple or weighted averaging function. Here we use simple averaging and calculate a final similarity score between two processes as follows:

\[
sim(x_i, x_j) = \frac{1}{\|S\|} \sum_{k=1}^{\|S\|} sim(x_i(s_k), x_j(s_k)). \tag{3.3}
\]

Once the similarity score is calculated, using similarity-based classification algorithms explained in Section 3.4.1, a model for classification can be built to predict unknown (i.e. test) events.

### 3.2 Similarities Between Networks

In this section and following subsections, instead of computing similarities at each spatial location between two systems, we adopt a new methodology for network-based classification of emerging event datasets using dissimilarity-based classification techniques.

Mathematically, a network \( G \) is the representation of a changing, real-world complex system defined by a set of nodes \( (V) \) and links \( (E) \) between them. Although classification of emerging events from underlying network structures is a novel idea for analysis of spatiotemporal datasets, networks are fundamentally graphs with special characteristics. Classification of dynamic networks using structural matching is not different from solving a general graph classification problem; indeed they share the same theoretical framework.
Figure 3.2: Method overview for the proposed network-based classification framework: (a) each event with an outcome $y_i$ is associated with a spatiotemporal process ($x_i$), (b) networks are generated based on the statistical correlations/coherences between time series, (c) the distances between pairwise network instances are computed using graph matching techniques and the resulting distance matrix ($D$) used as input to dissimilarity-based classifiers.

In Figure 3.2, an overview of the proposed method for network classification is given. The process starts with processing of a time series signal spread across spatial regions. Connections in the networks are generated based on significant linear (i.e. Pearson) correlation (or coherence) values (i.e. $p < 0.05$) between two time series signals at pairwise locations. Once the networks are constructed, dissimilarities between network instances are calculated through graph edit distance measures. In the final step, using (dis)similarity-based classification techniques, a classification model is being learned and generated.
3.3 Graph Matching

Whether it is a graph or network, any structure-based classification algorithm (except the ones utilizing topological features and vectorial graph embedding methods explained in Chapter 2) depends on the notion of pairwise similarity between training examples to build a statistical model. The graph matching problem, studied extensively for several decades, can be divided into two main branches: exact and inexact matching algorithms [22].

Some problems such as biological or chemical compound classification require exact matching of structures or substructures so that a particular instance can be assigned to a target class (e.g. to a specific gene or to a toxin family). However, exact matching can be too restrictive to form a basis for general purpose structure-based classification problems. For instance, in classification of handwritten texts or real world objects from computer vision applications, instances from the same class may exhibit minor or major structural differences, either naturally or sometimes due to the noise. Thus, a structure-based classification algorithm should account for these differences rather than expecting exact matches between inputs.

3.3.1 Exact Graph Matching

Exact matching algorithms look for a one-to-one correspondence between pair of graphs. An exact matching algorithm is characterized by the fact that the mapping between the nodes of two graphs must be edge-preserving in a sense that, if two nodes in the first graph are linked by an edge, then they can be mapped to two matching nodes in the second graph that are linked by an edge as well. Graph isomorphism, the strictest form of exact graph matching, requires a one-to-one correspondence between each node of the first graph and each node of the second graph. A weaker form of matching, subgraph isomorphism, requires that isomorphism only needs to hold between one of the graphs and a induced subgraph of the other.

One major drawback of exact matching is the complexity of the calculation. Graph isomorphism is in \( NP \) and subgraph isomorphism is \( NP \)-complete. More important is the limited generalizability of the approach to the real-world structure-based classification problems due to the restrictive assumption of identicality of graph topologies.
3.3.2 Inexact Graph Matching

Inexact or error-tolerant graph matching algorithms, on the other hand, do not look for a one-to-one correspondence between graph instances. Instead, differences are penalized using a cost function according to the severity of variations between edge and node labels. Thus, the goal of an error-tolerant matching algorithm is to find a mapping that minimizes the cost of matching. There exist different approaches to solving the inexact graph matching problem in the literature [82], both in terms of the selected optimization problem and the definition of the cost function. In this research, graph edit distance-based matching is selected as the target algorithm due to its flexible implementation and generalizable theory.

Graph Edit Distance

Graph edit distance (ged) is defined over a set of edit operations to transform one graph into another. The edit operations are a set of edge substitutions, additions, and deletions. The distance is the total cost of operations $c(\gamma)$ spent between two graphs $G_1$ and $G_2$ to make them identical.

**Definition 3.3.1 (Graph Edit Distance)** Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs, $\gamma : \{E_1 \rightarrow E_2 \cup V_1 \rightarrow V_2\}$ be a mapping between edges and nodes of $G_1$ and $G_2$, and $\Gamma$ be the set of all possible mappings. The graph edit distance $d(G_1, G_2)$ is defined as

$$d(G_1, G_2) = \min_{\forall \gamma_i \in \Gamma} [c(\gamma_i)]$$

where $c(\gamma_i)$ is the total cost of edit operations for mapping $\gamma_i$.

A direct approach to find the minimum ged computation between graphs is to formulate the problem as a tree search algorithm and look for all possible mappings of nodes and edges from a source graph $G_i$ to target graph $G_j$. For instance, despite its slowness for comparison of large graphs, both complete and heuristic versions of the A* algorithm [94] have been proposed to find optimal or near-optimal solutions for the standard ged calculation.

Although there is a well-defined framework around ged computations, the real advantage of using graph edit distances is because of its flexibility with respect to custom cost
functions and edit operation definitions. Different problem domains may require specialized operations and existing edit operations may not always respond the needs of the current context.

**Graph Edit Distance for Spatial Graphs**

For spatial networks, such as functional brain or climate networks, constructed graphs have special characteristics such as unique node labels and dynamically changing edge structures. It is possible to find the optimal solution by searching a smaller portion of the search space via more efficient search algorithms without dealing with arbitrary insertion and adding operations. In this work, to find the distance between two spatial graphs, we calculate the cost of matching each edge from one graph to another. Then using the cost of all possible edge matchings, we select the matching with the minimum cost as the graph edit distance between two spatial graphs \( \text{ged}_{\text{spatial}} \). However, the distance between two edges does not have a standard definition, even though the distance between two vertices can be defined as the normed distance. Therefore an edge distance calculation formula is proposed here for the purpose of the current application and used instead of standard edit operations.

**Definition 3.3.2 (Matching Distance Between Edges)** Let \( e = (v_1, v_2) \in E_1 \) be an edge between \( v_1 \) and \( v_2 \) represented as spatial coordinates in source graph \( G_1 \) and \( e' = (v'_1, v'_2) \in E_2 \) be an edge between \( v'_1 \) and \( v'_2 \) of the target graph \( G_2 \). The matching distance between edges \( e \) and \( e' \) is defined as:

\[
d(e \rightarrow e') = \min(||v_1 - v'_1|| + ||v_2 - v'_2||, ||v_1 - v'_2|| + ||v_2 - v'_1||)
\]

(3.5)

The proposed distance formula calculates the minimum matching cost (i.e. distance) between any possible vertex combination. In the example shown in Figure 3.3, the minimum distance between two edges according to Equation 3.3, the minimum distance between two edges according to Equation 3.5 becomes:

\[
d(e \rightarrow e') = \min(||(0, 0) - (0, 0)|| + ||(0, 1) + (1, 1)||, ||(0, 0) - (1, 1)|| + ||(0, 1) + (0, 0)||)
\]

\[
d(e \rightarrow e') = \min(3, 2) = 2.
\]

Once all the edge distances from source graph to the target graph are calculated using the above definition, finding the matching with minimum cost from the source to the
target graph can be formulated as a bipartite graph matching problem.

**Definition 3.3.3 (Bipartite Graph)** A bipartite graph $G_b = (V, E)$ is a graph whose vertices can be divided into two disjoint (i.e. independent) sets $V'$ and $V''$ such that every edge connects a vertex in $V'$ to one in $V''$.

For spatial graphs where the edges are the connections between spatial locations (i.e. vertices), the edge matchings across two graphs can be represented as a bipartite graph $G_b$ where bipartite nodes of the graph are the edges of graphs $G_1$ and $G_2$ such that $V' = E_1$ and $V'' = E_2$ respectively.

**Definition 3.3.4 (Graph Edit Distances for Spatial Graphs)** Let $G_b$ be a bipartite graph with a vertex set $V = V' \cup V''$ where $V' = E_1$ and $V'' = E_2$. Then the graph edit distance between two spatial graphs $G_1$ and $G_2$ is the cost of mapping $\gamma_i$, minimizing the cost among all possible matchings $\Gamma$,

$$
\text{ged}_{\text{spatial}}(G_1, G_2) = \min_{\forall \gamma_i \in \Gamma} \sum_{\forall (e_i \rightarrow e_j) \in \gamma_i} d(e_i \rightarrow e_j). 
$$

(3.6)

In the example shown in Figure 3.4, each graph $G_1$ and $G_2$ has $n = 2$ edges, the cardinality of the mapping set becomes $|\Gamma| = n!$. Therefore finding the minimum graph edit distance requires computation of two possible matchings over $\Gamma = \{\gamma_1, \gamma_2\}$ where $\gamma_1 : \{(e_1 \rightarrow e'_1), (e_2 \rightarrow e'_2)\}$ and $\gamma_2 : \{(e_1 \rightarrow e'_2), (e_2 \rightarrow e'_1)\}$.

However, for networks with a larger number of edges (e.g. $n \gg 10$), finding the graph edit distance between two graphs is not a trivial computation for the standard tree search algorithms like $A^*$, because the time and space complexity of the search problem grows factorially.
Luckily, a computationally efficient alternative for the suggested graph edit distance computation can be found in the combinatorial optimization field where the minimum cost bipartite graph matching problem is reformulated as an assignment problem. The assignment problem was first introduced by Khun [64] and later refined by Munkres [80] to assign $n$ number of workers to $n$ number of jobs with a minimum cost in polynomial time. The original algorithm is developed for $n$-by-$n$ input namely to assign equal number of workers to equal number jobs. Bourgeois et al. [11] extended Kuhn-Munkres algorithm for $n$-by-$m$ rectangular matrices that consist of more workers than jobs to enable partial matching.

The modified Khun-Munkre’s algorithm, given below, requires the cost matrix $C : N \times M \rightarrow \mathbb{R}$, where $N = |E_1|$ and $M = |E_2|$ and $C_{ij} = d(e_i \rightarrow e_j)$, to be defined over all possible edge matchings as input. Using a set of matrix operations, it finds the permutation (i.e. $\gamma^*$) that minimizes the cost of matching in polynomial time.

The next step for the algorithm is to calculate the (normalized) cost of optimal matching $\gamma^*$ between two graphs $G_i$ and $G_j$ to update the value of the corresponding cell ($D_{ij}$) in the dissimilarity matrix $D$. In order to calculate entire content of dissimilarity matrix, the matching algorithm needs to be called for every possible pair of graphs $G_i$ and $G_j$ in set $\mathcal{G}$. Once the dissimilarity matrix for all training and testing graphs are calculated, the final step is to build a model in the dissimilarity space.

### 3.4 Network Classification using (Dis)Similarities

Learning based on pairwise object dissimilarities is another alternative, in addition to feature-based learning approaches, for solving pattern recognition problems. As explained in the previous sections, the range of dissimilarity definitions can be broad and may
Data: Cost matrix $C : N \times M \to \mathbb{R}$ of edge matchings between $g_i, g_j$

Result: $ged(G_i, G_j)$ and the permutation of optimum matching $\gamma$

1. if $C$ is not a square matrix then
2. Make the matrix square by adding $\infty$ cost for unassigned jobs;
3. end
4. For each row $r \in C$ subtract the smallest entry from every other entry of that row;
5. For each col $c \in C$ subtract the smallest entry from every other entry of that col;
6. while true do
7. Draw lines through appropriate rows and columns so that all the zero entries of the cost matrix are covered and the minimum number of such lines is used;
8. if the minimum number of covering lines $== N$ then
9. Optimal $\gamma$ assignment is found;
10. Exit Loop;
11. end
12. Determine the smallest entry not covered by any line;
13. Subtract this entry from each uncovered row, then add it to each covered col;
14. end
15. Calculate $ged(G_i, G_j)$ of matching $\gamma$;

Algorithm 1: Modified Khun-Munkres Algorithm for the Assignment Problem

change according to the nature of the problem being solved. Defined dissimilarity functions (e.g., $ged$) can be the result of metric operations or completely arbitrary. The problem with non-metric dissimilarity measures is that they may conflict with the intrinsic space requirements of existing machine learning algorithms, which may be built upon assumptions such as a representation of objects in Euclidian space. Therefore, in order to solve learning problems using dissimilarities, we either need to define a novel classification framework so that the input matrix $D$ does not violate the space requirements of the optimization being used, or modify non-metric distances to the closest metric versions to satisfy the requirements of the existing learning algorithms.

In this respect, methods that utilize pairwise dissimilarities can be divided into four groups: (1) learning algorithms adapting their geometry according to the input properties, (2) algorithms modifying their input in order to fit geometrical constraints, (3) classifiers working on dissimilarity spaces, (4) nearest neighbor-based approaches.
3.4.1 Learning with (indefinite-) Kernels and (pseudo-) Euclidian Space Embeddings of Dissimilarities

Mathematically, a space is a set with structure. Spaces constitute the boundaries of learning algorithms and determine the medium they operate in. An overview of the metric and non-metric spaces that the classification algorithms covered in this work operate on can be seen in Figure 3.5\(^1\). The nested relationships that exist between spaces are illustrated; for instance, Euclidean space is bounded by more strict rules than metric spaces. The spaces shown in dashed rectangles are generated by relaxation of some of the constraints in their corresponding metric versions.

In classical machine learning, the differences between points, objects are measured using some kind of a metric. A metric space is the one that satisfies all four constraints in the following definition:

**Definition 3.4.1 (Metric Space)** A metric space is a pair \((X,d)\) where \(X\) is a set and \(d\) is a distance function \(d : X \times X \to \mathbb{R}^+_0\) such that the following conditions are fulfilled for all \(x, y, z \in X\):

1. **Reflexivity** \(d(x, x) = 0\)
2. **Symmetry** \(d(x, y) = d(y, x)\)
3. **Definiteness** \(d(x, y) = 0 \Rightarrow (x = y)\)
4. **Triangle equality:** \(d(x, y) + d(y, z) \geq d(x, z)\).

\(^1\)Figure 3.5 and Definitions 3.4.2 to 3.4.7 are the modified versions of Figure 2.2. and corresponding definitions from [83]
For example the distances between instances represented as vectors can be measured using various kinds of metric such as city block, Euclidian, or max-norm, while the similarity between string sequences can be measured using Hamming metric formulations. Given a metric distance matrix, if it can be embedded into Euclidian space or represented in some Hilbert space (see Def. 3.4.2 - Def. 3.4.7), then existing machine learning algorithms such as SVMs can be applied successfully without changing the geometry of the learning algorithm.

**Definition 3.4.2 (Inner Product Space)** Let $X$ be a vector space over $\mathbb{C}$. A vector space with an inner product $(X, <\cdot,\cdot>)$ is an inner product vector space.

**Definition 3.4.3 (Hilbert Space)** An inner product space for which the induced norm gives a complete metric space is a Hilbert space.

**Definition 3.4.4 (Euclidian Space)** A Euclidian space $\mathbb{R}^p$ is a positive real metric space equipped with a positive definite inner product $<\cdot,\cdot>$.

However, when we think about similarities between objects, not all similarities may be the result of metric operations that we could fit into finer spaces, such as Inner Product Spaces or Hilbert Spaces. For example the edit distances between graphs satisfy the criteria from (1) to (3) of Def. 3.4.1, however they do not necessarily satisfy the (4)th criterion, triangular equality. Therefore they belong to a special case of metric spaces called semi-metric spaces. In semi-metric spaces the learning algorithms operating on Hilbert or Euclidian spaces need to be adapted to the geometry of pseudo-Euclidian and Krein spaces, which are described briefly in the following definitions:

**Definition 3.4.5 (Indefinite-Inner Product Space)** Let $\mathcal{V}$ be a vector space over $\mathbb{C}$. A vector space with indefinite inner product $(\mathcal{V}, <\cdot,\cdot>_{\mathcal{V}})$ is an inner product vector space. Indefinite inner product is a generalization of inner product with only symmetry and linearity conditions hold.

**Definition 3.4.6 (Krein Space)** A Krein space $\mathcal{K}$ is a vector space over $\mathbb{C}$ such that indefinite inner product is defined and $\mathcal{K}$ allows a orthogonal decomposition $\mathcal{K} = \mathcal{K}_+ + \mathcal{K}_-$ such that $\mathcal{K}_+$ and $\mathcal{K}_-$ are Hilbert spaces.
**Definition 3.4.7 (Pseudo-Euclidian Space)** A pseudo Euclidian space represented as $\mathcal{E} = \mathbb{R}^{p,q}$ is a real vector space equipped with indefinite inner product. $\mathcal{E}$ allows direct decomposition $\mathcal{E} = \mathcal{E}_+ + \mathcal{E}_-$ where $\mathcal{E}_+ = \mathbb{R}^p$ and $\mathcal{E}_- = \mathbb{R}^q$ and the inner product is positive definite on $\mathcal{E}_+$ and negative definite on $\mathcal{E}_-$.

In summary, dissimilarity-based learning starts with the generation of dissimilarity matrix $D : N \times N \rightarrow \mathbb{R}$. The first step is to analyze the geometric characteristics of this matrix to discover the space constraints. In Figure 3.6, the flow chart for the general learning strategies that can be employed using dissimilarity matrix $D$ is illustrated.

There are two main approaches to classification using dissimilarities: (1) the dissimilarities can be transformed into similarities then treated as inner products in some (Krein) Hilbert space, and kernel-based learning algorithms such as (indefinite-) SVM are employed; or (2) they can be embedded into (pseudo-) Euclidian spaces and treated as vectors in this configuration space. In the following sections, the guidelines for these two approaches will be discussed in detail.

**Learning with (Indefinite-) Kernels**

Kernel methods dealing with pairwise similarities rather than the explicit representations of the objects in higher dimensions is a popular tool in both structured and canonical data classification. A kernel function is a dot product, namely a similarity measure, in some higher dimensional feature space,

$$\kappa(x, x') = \langle \phi(x), \phi(x') \rangle_H, \quad (3.7)$$
where $\phi$ is a mapping $\phi : \mathcal{X} \rightarrow \mathcal{H}$. The power of kernel methods orginates from the theory that kernel matrices can be used without explicitly computing the mapping $\phi(x)$, as long as the kernel function satisfies the requirements of validity such as positive semi definiteness criteria. The reason behind this is that for any positive semi-definite (p.s.d.) kernel, there is a vector space configuration in some high dimensional Hilbert space that pairwise inner products are equal to $K$.

As a state-of-the-art kernel-based binary classification algorithm, SVM tries to optimize the following in its dual form,

$$
\begin{align*}
\text{maximize} & \quad 1^T\alpha - \frac{1}{2}\alpha^T \text{diag}(Y) K \text{diag}(Y) \alpha \\
\text{subject to} & \quad 0 \leq \alpha \leq c_1, \ y^T\alpha = 0.
\end{align*}
$$

where the hyper-paramaters $\alpha$ and $C$ are the weight vector and the regularization parameter respectively. $K$ is the symmetric kernel matrix required to be p.s.d. for the problem to be formulated as a convex optimization, and $Y$ is the class label vectors.

The above formulation of the SVM is also known as definite kernel learning. However, not all the similarity definitions necessarily satisfy the positive semi-definiteness criteria; some might be the result of some arbitrary assignment operations or can be indefinite.

There is a class of learning algorithms called indefinite kernel classification algorithms [18] operating on kernel Krein spaces (i.e. RKKS) that simply work by replacing $K$ with symmetric similarity matrix $S$. Although the optimization is no longer convex, empirical results show that replacing kernel matrix $K$ with a non-p.s.d. similarity matrix $S$ sometimes works as well as the definite case; however, the optimization converges to a stationary point instead of a global minimum [86]. Therefore, dealing with non-convex geometry and slow convergence to a non-optimal saddle point limits the use of indefinite kernel methods as a standard classification algorithm.

Alternatively, instead of changing the geometry of learning, the input matrix $S$ can be modified through a set of spectral operations (i.e. clip, flip, or shift) to obtain a p.s.d. matrix. Spectrum modification starts with the eigenvalue decomposition of symmetric similarity matrix $S$,

$$
S = U^T \Lambda U
$$

(3.9)
where $U$ is the orthogonal matrix and $\Lambda$ is the diagonal matrix of eigenvalues.

Pekalska et al. [87] suggest a spectrum flip operation that works by changing the sign of the negative eigenvalues of similarity matrix $S$ to positive, to obtain a \textit{p.s.d.} kernel. Instead of subtracting two Hilbert spaces in a Krein space they add them to obtain a modified similarity matrix in Hilbert space. The spectrum flip operation and reproduced similarity matrix $S_{\text{flip}}$ can be formulated as follows:

$$S_{\text{flip}} = U^T \text{diag}(|\lambda_1|, |\lambda_2|, \ldots, |\lambda_n|) U.$$  \hfill (3.10)

Wu et al. [119] propose a spectrum clip operation assuming that negative eigenvalues of a similarity matrix are due to noise and should be discarded from the input matrix. The similarity matrix is made \textit{p.s.d.} by clipping all negative eigenvalues to zero as in the following:

$$S_{\text{clip}} = U^T \text{diag}(\max(\lambda_1, 0), \ldots, \max(\lambda_n, 0)) U.$$  \hfill (3.11)

In shifting, negative eigenvalues are overcome by shifting entire matrix with a constant value, i.e. absolute value of the minimum eigenvalue,

$$S_{\text{shift}} = U^T \text{diag}(\lambda_1 + \min(\Lambda, 0), \ldots, \lambda_n + \min(\Lambda, 0)) U.$$  \hfill (3.12)

Chen et al. [17] give an overview of spectrum modification operations and compare their performance on various datasets. For each dataset, the methods (i.e. clip, shift, and flip) perform differently; therefore, depending on the characteristics of the input similarities, selection of the right modification strategy requires a validation process.

**Converting dissimilarities to similarities**

The similarity-based learning algorithms covered in the previous section work with pair-wise similarity measures ($S$) between samples. However the distances or dissimilarities between samples are quite the opposite of similarities. The most direct approach for transforming graph edit distances or any dissimilarity matrix ($D$) into similarities is to apply a monotonically decreasing function $f : \mathbb{R} \to \mathbb{R}$ over distances. Burke et al. [82]
propose four different trivial transformation functions for that purpose:

\[
\begin{align*}
S_1 &= f_2(D) = -D \\
S_2 &= f_1(D) = -D^2 \\
S_3 &= f_3(D) = \text{tanh}(-D) \\
S_4 &= f_4(D) = \exp(-\lambda D) \text{ where } \lambda > 0
\end{align*}
\]  

(3.13)

The result of the transformation is a symmetric similarity matrix \( S \); however, it is not guaranteed to be p.s.d., hence it may not constitute a valid kernel, and the indefinite kernel learning algorithms explained above can be a natural choice for classification.

**Embedding of Dissimilarities in (pseudo-) Euclidian Spaces**

Alternatively, instead of working in inner product space, one can map the dissimilarities \( D \) into a vector space \( X \) by preserving the original distances as much as possible. The assumption is the distance matrix \( D \) is the result of applied distance function \( d : X \times X \rightarrow D \) over the vector space \( X \). If \( D \) is a Euclidian distance matrix then,

\[
X = Q_k \Lambda_k^{1/2}
\]

(3.14)

is a \( k \leq n \) dimensional vector configuration of \( X \) in Euclidian space, where \( \Lambda_k \) is the diagonal matrix of \( k \) leading eigenvalues of \( D \) and \( Q_k \) is the matrix of corresponding eigenvectors. If \( D \) is a dissimilarity matrix of non-Euclidian distances, then a Euclidian
Figure 3.8: Dissimilarity space embedding and classification (a) during training, a set of graphs are selected as part of the prototype set selection step, (b) testing instances are embedded against this.

configuration $X$ cannot be generated directly. However, to map non-Euclidian distance matrices into Euclidian space, only $p \leq k$ positive eigenvalues of $D$ can be taken into account and the mapping

$$ X = Q_p \Lambda_p^{1/2} \quad (3.15) $$

becomes an approximation of the original dissimilarities in Euclidian space.

According to Definition 3.4.7, the pseudo-Euclidean space can be interpreted as combination of two Euclidian spaces $\mathcal{E} = \mathbb{R}^{p \times q}$. The distances in pseudo-Euclidean space can be calculated by subtracting embedded distances in negative space $\mathbb{R}^q$ from the embedded distances in positive space $\mathbb{R}^p$. Pekalska et al.[28] present this embedding process and pseudo-Euclidian formulation of well-known classification algorithms such as Fisher’s linear discriminant, generalized nearest mean and support vector machine classifiers.

### 3.4.2 Dissimilarity Space Embeddings

As an alternative to concrete (indefinite-) kernel and (pseudo-) space embeddings of dissimilarities, the abstract embedding of dissimilarities, namely dissimilarity space embedding, is also proposed, as illustrated in Figure 3.8.

In this new space, to build a learning model over a graph domain $\mathcal{G}$, in addition to training $T \subset \mathcal{G}$ and testing sets $S = \mathcal{G} / T$ of conventional classification algorithms, a prototype set $P \subseteq T$ needs to be defined to form the basis of the dissimilarity space.

Given a set of $n$ prototype graphs $P = \{p_1, p_2, ..., p_n\}$, and a dissimilarity matrix
$D : T \times P \rightarrow \mathbb{R}$, the mapping $\phi : \mathcal{G} \rightarrow \mathbb{R}^n$ of a graph $G$ into dissimilarity space $\mathcal{P}$ is defined as:

$$\phi(G) = (d(G,p_1), d(G,p_2), \ldots, d(G,p_n)) \text{ where } n = |P|, d(i,j) \in D. \quad (3.16)$$

The dissimilarity space $\mathcal{P}$ constructed via the prototype set $P \subset \mathcal{G}$ and embedding $\phi$ is Euclidian, with definite inner product $\langle \cdot, \cdot \rangle$. Therefore any conventional linear or non-linear classification algorithms (i.e. SVM, LDA) can be used to build a classifier function $f(\phi(G)) : \mathcal{G} \rightarrow \mathcal{Y}$ over the training set $T$ and can be validated through the test set $S$.

### 3.4.3 (Nearest) Neighborhood-based Learning

Given a dissimilarity matrix of networks $D : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$, a set of class labels $\mathcal{Y}$, and a labeled set of training instances $(G_j, y_j)$, the most direct learning approach to classify an unknown network instance $G_i$ is to look for the most similar instances through neighborhood relations. In $k$-nearest neighborhood learning [23], the goal is to assign an instance to a class based on majority of labels of its $k$ neighbors.

In simplest form, when $k$ is 1 the decision is based on the closest single neighbor of $G_i$. The decision function $f : \mathcal{G} \rightarrow \mathcal{Y}$ is defined as:

$$f(G_i) = y_j, \text{ where } j = \arg\min_j d(G_i,G_j) \text{ s.t. } i \neq j. \quad (3.17)$$

For $k > 1$, the decision function of a two class problem with class labels $\{-1, 1\}$, can be formulated as:

$$f(G_i) = \text{sign}(\sum_{i=1}^{k} y_{I(i)}). \quad (3.18)$$

where $I$ is an array of indices of the points ordered in an increasing order in terms of their distance to $G_i$ and $I(i)$ is the index of the $i$th element in that array.

In this respect, the nearest neighborhood learning is a primitive learning architecture since it does not include any model building step. The dissimilarity matrix itself is enough to create an effective classifier, and that is still a valid alternative approach to more advanced dissimilarity-based classification techniques explained in previous sections.
Chapter 4

Motivating Examples and Datasets

In this chapter we discuss the datasets that motivate the current study. Four datasets from two domains are tested to evaluate the effectiveness of the proposed methodology in comparison to alternative spatiotemporal data classification techniques covered in Chapter 2. The example problems are from climate and cognitive science. In the climate domain, the end trajectories of hurricane patterns are predicted, with respect to whether they will hit land or remain offshore in the North Atlantic or the Indian Ocean. In the cognitive science domain, the cognitive state of experiment participants is classified by their EEG patterns. In the first experiment, whether they are experts in engineering design problems is evaluated, and in the second experiment the goal is to predict what kind of mental task a participant is executing at any given time interval.

4.1 Hurricane Track Classification

Hurricanes lead to major natural disasters in the regions of landfall. To better understand and predict regional hurricanes, it is important to consider the factors influencing the dynamics of their tracks. McCloskey et. al. [74] analyze hurricane track patterns in North Atlantic. They find two distinct subgroups, as shown Figure 4.1: the Cape Verde and the Western Caribbean hurricanes, which differ in point of origin, track shape and driving climatic forces. This categorization shows also parallelism to existing hurricane track clustering studies in the literature [34, 42, 81], supporting their findings.

In this research, we address the important and complementary problem of forecasting whether the hurricane in its genesis region will eventually strike land or remain off-
Figure 4.1: Motivating Example I - (a) Patterns of North Atlantic and (b) Indian Ocean hurricane tracks.

shore. Existing real time hurricane track prediction studies, such as CLIPPER [84] and GFDL [65], focus on short-term predictions of two to five days. To complement these methods, we aim at (a) making mid-term predictions of five to ten days after the initial hurricane formation and (b) revealing contributing factors, to possibly improve climate models for seasonal tropical cyclone activity projections.

For this purpose, we analyze hurricane data from two distinct regions, the North Atlantic and the east Indian Ocean.

4.1.1 Dataset I - Atlantic Ocean Hurricanes

In the Atlantic, we focus on Cape Verde hurricanes that originate off the west coast of Africa and follow the curvature shown in Figure 4.1a. The prediction problem tries to discriminate (a) “Recurving” hurricanes that threaten North America, north of about 35°N but remaining offshore, from (b) “Straight Moving” hurricanes that make landfall in the Caribbean and North America south of this latitude. Hurricane locations from the “besttrack” (HURDAT) data are extracted to find the matching hurricanes and end result of the trajectories, whether they make landfall or not.

4.1.2 Dataset II- Indian Ocean Hurricanes

In the east Indian Ocean, hurricanes originate in close proximity to the center of the ocean follow two distinct pattern, (a) hurricanes moving towards the land and hitting Madagascar and the West India shores, and (b) hurricanes that curve towards the south
and stay in the ocean. Hurricane locations from the West Indian Ocean hurricane dataset are extracted and grouped into two groups (i.e. land hitting and recurving) according to their end trajectories.

### 4.1.3 Climate Data and Preprocessing

For climate data, noisy inputs due to remote observations and sensor readings are generally corrected via model simulations to generate homogeneous and interpolated datasets. We use the output of NCEP/NCAR Reanalysis 1 project data from 1948 to the present available in 4-times-daily format and as daily averages. For the Hurricane Track classification problem, we look at SLP data up to 30 days before each hurricane, and we create networks using the data within this time window.

Climate data is subject to a dominant seasonality signal, however, that masks out more relevant and important information within the data. Therefore for analysis covering inter-seasonal variations, an anomaly time series is constructed that removes the monthly mean values ($\mu$) from the input signal; this is used instead of the raw data for feature-based classification of the climate data.

\[
x^*_t,m = x_t - \mu_m \quad \text{for all } m \in \{1..12\} \tag{4.1}
\]

An alternative is to use standardized data (i.e. z-score), dividing the anomaly values by the standard deviations ($\sigma_m$) within the same period.

\[
x^*_t,m = \frac{x_t - \mu_m}{\sigma_m} \tag{4.2}
\]

For our analysis we use standard anomaly time series for removal of the seasonal signal from the data. For analysis capturing longer periods (i.e. decadal analysis) de-trending of the signal may also be required as part of the preprocessing, to avoid spurious positive or negative correlations between distant locations. Detailed discussions of anomaly time series construction and pre-processing of climate data exist in the literature [57].


4.2 Classification of Human Cognitive States

The second domain is cognitive science, in particular classification of human cognitive states. We test two datasets, the first one related to engineering design concepts and the second a public dataset related to mental task classification.

4.2.1 Dataset III - Expert Classification in Engineering Design

Traditionally, the concept of engineering design is introduced to students beginning with two-dimensional (2D) projected views (plan and elevation) of objects. Students are gradually exposed to the task of transforming information from 2D to 3D using isometric and perspective views of an object under study, thereby building an intuition for how to transform information between 2D and 3D [68], as illustrated in Figure 4.2.

In this research, the effect of experience in the transformation of CAD drawings from 2D to 3D and from 3D to 2D planes is investigated. Twelve object drawings are presented to subjects in two sessions in a cue-guided experimental paradigm. In the first session, the goal is to find matching 3D representation of an object presented in a third angle projection of a 2D CAD drawing. In the second session, an object in a 3D plane is shown and subjects are asked to find the corresponding 2D projection of the same figure. The experiments were conducted at North Carolina State University, Human Factors and Ergonomics laboratories over a two week period with 12 subjects where 2 of them were excluded due to an electrical problem affecting the output signal occurred during experiment.

As a classification problem, the question we are trying to answer, given an EEG data collected during task execution, is whether we can determine the expertise level of the subject (i.e, an expert or novice system user) by looking at the data in particular network structures generated from the data.

4.2.2 Dataset IV - Mental Task Classification

The last dataset comes from one of the largest EEG data repositories from Graz-BCI repository. This dataset includes recordings of nine individuals with severe motor disabilities who were asked to perform different types of imaginary actions; the goal of the research was to find the best mental task paradigm for binary BCI classification problems. Users carried out five different mental tasks: word association, mental arithmetic,
4.2.3 EEG Data and Preprocessing

In the first experiment, EEG data was collected using a 16-channel g.USBAmp amplifier with a sample rate set to 256 Hz and processed via a Bci2000 signal processing system. Electrode were placed on subjects using the international 10-20 reference method. Electrode positions included channels Fp1, Fp2, F3, Fz, F4, T3, C3, Cz, C4, T4, P3, Pz,
P4, O1, Oz and O2. The average activity of the earlobe was used as a reference. During data acquisition, a notch filter at 50 Hz was enabled in order to prevent line noise. After collection of the signals, we applied de-trending to get rid of low frequency noise and to better visualize the signals. Eye blinking and movement-related artifacts were removed from the signal using the Independent Component Analysis technique.

For the second EEG dataset, we use the data from Graz-BCI data repository. The data was recorded from 30 electrode channels using the international 10-20 system. Electrode positions included channels AFz, F7, F3, Fz, F4, F8, FC3, FCz, FC4, T3, C3, Cz, C4, T4, CP3, CPz, CP4, P7, P5, P3, P1, Pz, P2, P4, P6, P8, PO3, PO4, O1, and O2 using the g.tec GAMMAsys system. The signal was band pass filtered at 0.5-100 Hz (notch filter at 50 Hz) and sampled at a rate of 256 Hz.

To get the signal power over a frequency band we use the Fast Fourier Transform (FFT). To calculate the level of synchronization across different electrode locations and different frequency values, we use the coherence estimation function $\gamma$ as a normalized cross spectrum,

$$\gamma(f) = \frac{G_{uv}(f)}{2G_{uu}(f)G_{vv}(f)},$$ (4.3)

where $G_{uv}(f)$ is the cross spectrum between signals $X$ and $Y$ and $G_{uv}(f)$ and $G_{uv}(f)$ are the auto spectrum of the signals. We use signal power and (significant) coherence values as part of the feature-based and network-based classification algorithms in the next chapter.
Chapter 5

Experimental Results and Discussion

This section is structured as follows. We start with the canonical data classification. We describe state-of-the-art classification algorithms using direct embedding of the three-dimensional spatiotemporal data into a two-dimensional tabular format, without employing any additional logic to deal with the intrinsic structure in the data. Then we employ generic feature selection algorithms on the canonical datasets from existing machine learning repositories, to create a more compact and robust representation of the high-dimensional tabularly embedded data.

Our next approach is to look at each domain to find specific features and feature selection methods defined for that particular domain. We compare classification accuracies of these domain-specific features to generic, statistically selected features.

As an alternative to these feature-based approaches, as a next step we look at the similarities between spatiotemporal systems and utilize similarity-based classification algorithms. In this respect, we use two approaches. In the first one, we look for temporal similarities across spatial locations between two systems. In the second, we use a graph theoretical approach to calculate similarities between network instances associated with spatiotemporal processes.

Our results indicate that there is a significance difference in terms of classification accuracies between these approaches. As discussed throughout this section, the network-based approach or features extracted from network structures significantly improve on the classification accuracies in comparison to classifiers using generic and other domain-specific feature selection techniques.
5.1 Classification Algorithms on Raw (Tabular) Data

As explained in Chapter 2 and Figure 2.2, if there is data to classify and the characteristics of the datasets are not known in advance other than the class labels, the most straightforward approach is to represent the data in row matrix form and use off-the-shelf classification algorithms to see if the data is classifiable with no prior information.

The target spatiotemporal data described in the previous chapter consists of $n$ recordings from $m$ spatial locations, over a period of $T$ distinct time points. Therefore it has a 3-dimensional representation, $n \times m \times T$. In order to run classification algorithms that require tabular data, we reshape each dataset into a 2-dimensional representation (i.e. a matrix with dimensions $n \times m. T$). After that step, we apply four widely used linear and non-linear feature-based classification algorithms suggested by Lotte et al. [70] for EEG-based BCI data and Vasimalla [113] for feature-based classification of spatiotemporal data, as explained in the following subsections.

Another reason why we select those classification algorithms is to compare classifiers with different characteristics; specifically, each of these classifiers represents a different approach and assumptions about how to capture regularities in data. For example, $k$NN is a local classifier, a decision tree is a local classifier with a generated model, Naive Bayes is a parametric generative classifier, and SVM is a model-based discriminative classifier with regularization capability. Kernel-based SVMs produce nonlinear decision boundaries. However, all of these classifiers have static capabilities; classifiers with dynamic learning capabilities such as HMMs are outside of the scope of this work.

5.1.1 Support Vector Machines

Support Vector Machines (SVM) use a hyperplane to separate samples drawn from two classes. The goal of SVM is not only to separate the training samples; it also ensures that there is some distance (i.e. margin) between the samples (or vectors, in SVM terms) and the separating hyperplane for better generalization to unseen samples. The optimal classification occurs when the hyperplane maximizes the margin from the supporting vectors, as illustrated in Figure 5.1.

The problem with linear decision boundaries is that not all problems are linearly separable. We may need other methods for nonlinear cases. As illustrated in Figure 5.2, one commonly applied solution is to map input samples into higher dimensional space where
a linear decision boundary can be applied. This is called the “kernel trick.” Depending on the application, there exist many forms of kernel functions. Three generically applicable and widely studied kernel methods for canonical data are polynomial kernels, radial basis functions, and sigmoid kernels. The formula for polynomial kernel is:

\[ K(x_i, x_j) = (x_i x_j + 1)^q, \]  

where \( q \) is the degree of the polynomial. The radial-basis functions refers to the following equation,

\[ K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2), \]  

where \( \gamma \) is the radius of a spherical mapping. The sigmoid kernel is

\[ K(x_i, x_j) = \tanh(\gamma(x_i x_j) - \delta), \]  

where \( \gamma \) and \( \delta \) are model parameters and value of the function varies between \([-1, +1]\). The choice between these kernels is commonly not made in advance but instead is based on their empirical performance in a given domain.
5.1.2 \textit{k}-Nearest Neighbor Classifiers

\textit{k}-Nearest Neighborhood (also covered in Section 3.4.3 in more detail) is a local classification technique where the decision function (see Eq. 3.17 and Eq. 3.18) is based on the majority of the instance labels within close proximity (i.e. \textit{k} neighbors) of the sample point \(x\). The value of parameter \(k\) can be preset by the user (e.g. \(k\) could be 1, 3, 5 etc.) or it can be determined through a validation step where the value of \(k\) is the one minimizing the validation error.

The decision function for \(k\)NN is based on the distance between a test instance \(x\) being classified and a set of training samples. The most commonly used distance functions are metric, such as Euclidian; however, non-metric distance functions such as Mahalanobis can also be applied if it is known in advance that the distribution of the input samples is Gaussian. As above, in many domains where existing theory does not dictate specific values, the parameter values chosen for a \(k\)NN classifier are those that produce the best performance empirically.

5.1.3 Decision Trees

A decision tree is a local and model-based supervised learning algorithm where local regions for classes are defined based on a set of recursive splits of the data [88]. Internally in a decision tree, nodes are conjunction branches represented by “\textit{if-else statements},”
where the value of a selected feature at that branch directs the decision to the lower level nodes. Leaf nodes are decision nodes and lead to the class labels.

The recursive splitting of the decision tree continues either until every instance is classified correctly based on the branching or certain criterion of "purity" is met. Decision trees are widely used because of their interpretability and their ability to handle nonlinearity; however they may not be preferred if the local classifiers are not suitable for the problem domain and the data. One common case is when individual features are related by some implicit non-local function; the classic example of such a case is the parity function, for which a decision tree can be built but will fail to generalize to new data.

5.1.4 Naive Bayes

Naive Bayes classifiers are a family of parametric and probabilistic decision functions based on independence assumptions between input features. These classifiers assign class labels to problem instances represented as feature vectors. In general terms, Naive Bayes assigns an instance $x$ to a class $C_k$ with gives the highest probability based on a maximum a posteriori (MAP) decision rule:

$$
\hat{y} = \arg\max_{k \in \{1, \ldots, K\}} p(C_k) \prod_{i=1}^{n} p(x_i|C_k).
$$

(5.4)

When dealing with continuous data, a general approach would be to assume that data from each class is the result of some kind of probability distribution and that the goal of the classifier is to find parameters of these distributions that produce the most generalizable model. For example, if we assume a Gaussian distribution, given a value $v$ of a feature, then the probability of that feature is drawn from the class $c_i$ is

$$
p(x = v|c_i) = \frac{1}{\sqrt{2\pi \sigma_i^2}} e^{-\frac{(v-\mu_i)^2}{2\sigma_i^2}},
$$

(5.5)

where $\mu$ and $\sigma$ are the mean and variance of the Gaussian distribution. Independence between input features is a strong assumption that rarely holds perfectly in real-world data. Nevertheless Naive Bayes is sometimes surprisingly robust to violations of the assumption, and it is commonly applied due to its simplicity and efficiency.
### Table 5.1: Overall cross validation accuracies for tabularly embedded data for each classification algorithms. Highest and significantly improved statistics are displayed in bold.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Tabular</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k$-fold CV</td>
</tr>
<tr>
<td></td>
<td>D. Tree</td>
</tr>
<tr>
<td>Atlantic Ocean Hurricanes</td>
<td></td>
</tr>
<tr>
<td>Indian Ocean Hurricanes</td>
<td></td>
</tr>
<tr>
<td>Cognition</td>
<td></td>
</tr>
<tr>
<td>Graz BCI</td>
<td></td>
</tr>
</tbody>
</table>

#### 5.1.5 Results

Four generic classification algorithms and three SVM kernel functions were applied to the tabular data. The $k$-fold cross validation results for each method are given in Table 5.1. The highest and significantly different results calculated using a paired $t$-test are listed in bold for each dataset. To find the significance of the classifier over entire collection of datasets, a Kruskall-Wallis $H$-test (also known as a *one-way ANOVA on ranks*) was applied. As can be seen in the table, none of the classifiers were able discriminate the test instances accurately for the tabular datasets. For two datasets, the decision-tree was a slightly but significantly better method as a local classifier among other six. However, to generalize across all the datasets no method was significantly different from the others at the 0.05 level.

#### 5.2 Feature Selection Algorithms

A tabular representation of spatiotemporal data, where each column is treated as a feature (even though these may not match our intuitions about features in any other domain), results in a high-dimensionality dataset. Decision functions are likely to suffer from the curse of dimensionality. In order to reduce the overhead on the classification algorithms due to extra dimensions, and to discard uninformative and redundant columns, subsets of relevant attributes can be selected or generated.

The process of extracting relevant attributes from high-dimensional data is called
feature selection; feature selection has been studied extensively in the machine learning literature. In a classification framework, there are three main stream approaches for feature selection: wrapper, filter and embedded methods.

Filter methods rank features based on fitness criteria, such as the degree of correlation to target class, \( t \)-tests, or Fisher scores. The ranking criteria may change from one method to another depending on statistical assumptions and the type of data being evaluated. After the ranking step, the subset of columns showing highest scores and/or variability across each class are selected as features.

Filter methods do not take classifier logic into account during feature selection, which can limit the success of the classifier if the selection criteria does not meet the requirements of the classifier. For instance, Naive Bayes assumes independence between the input features, but univariate feature selection algorithms may come up with a feature set that is highly correlated. Wrapper methods, on the other hand, use classifier performance to evaluate the ranking of features. They implement a search strategy to find a subset of features that works best with the selected classifier. This search can be expensive, however, and in high-dimensional feature spaces, this leads to infeasible search complexities. Although there exist heuristic-based search alternatives such as hill climbing or genetic algorithms, wrapper models usually suffer from the curse of dimensionality.

To overcome the limitations of wrapper and filter methods, embedded methods include a feature selection step as part of the classification. Decision trees and methods based on regularization techniques are examples of embedded algorithms. We utilize them here for comparison with filter methods. Due to the infeasible computational complexities that wrapper methods introduce, we did not include wrapper methods as part of our comparison.

We used the Arizona State University Feature Selection repository, written in MATLAB, to test our data with well-known feature selection algorithms. Domain knowledge is rarely sufficient to determine \textit{a priori} the best feature selection algorithm for a dataset. We thus tested 6 filter methods and 1 embedded method for our datasets chosen as a representative sample over available techniques. In order to have a diverse set, three most common univariate (\( t \)-test, \( F \)-test, InfoGain), statistical feature selection algorithms are selected and implemented to find the most relevant attributes. To eliminate the redundancy effect that univariate selection techniques may suffer from in high dimensional inputs, three multivariate (mrMr, FCBF, CFS) selection techniques that tries to find
most relevant at the same time statistically independent feature sets are investigated to explore different characteristics of the input space.

In the following paragraphs, we briefly discuss the feature selection algorithms we tested.

5.2.1 \textit{t-Score}

\textit{t-Score} is a univariate filter algorithm for binary classification that measures the significance of the difference between the means ($\mu$) of feature values across classes, then assigns a \textit{t-score} for each feature for weighting.

\[
t_{\text{score}} = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}
\]  

(5.6)

5.2.2 \textit{F-Score}

\textit{F-Score} takes between- and within-class variances into account and tries to find feature sets that exhibit the most discrimination among all using the following score calculation formula,

\[
F_{\text{score}} = \frac{\sum n_i (\mu_i - \mu)^2}{\sum \sigma_i^2}.
\]  

(5.7)

5.2.3 \textit{Information Gain}

\textit{Information Gain} measures the reduction in uncertainty after adding a feature to the current set, formulated as

\[
IG(X,Y) = H(X) - H(X|Y),
\]  

(5.8)

where $H(X)$ is the entropy, a measure of uncertainty associated with a feature vector $X$, and $H(X|Y)$ is the entropy of $X$ after adding feature $Y$ into the set. The features are ranked then selected by how much they reduce entropy.

5.2.4 \textit{Minimum redundancy, maximum relevance}

\textit{Minimum redundancy, maximum relevance (mRMR)} tries to find a subset of features
that are most relevant but at the same time most diverse for a given dataset. mRMR has been proven to be effective in high-dimensional datasets such as gene expression, where many correlated (redundant) features exist [25]. For continuous data, relevance is calculated using an \( F \)-test, and redundancy is evaluated using statistical correlations such as Pearson’s \( r \) between features.

5.2.5 Fast Correlation-Based Filter

Fast Correlation-Based Filter (FCBF) is a two-level feature selection algorithm that measures feature-class and feature-feature correlations. For relevance calculation it utilizes linear correlation values. For redundancy decisions, a heuristic approach using a symmetric uncertainty score based on information gain principles is applied to keep least redundant features in the output set.

5.2.6 CFS

Correlation-based Feature Selection (CFS) is another multivariate feature selection algorithm based on redundancy and relevance concepts. CFS for each feature calculates a Merit-score over the redundancy and relevance values for each feature and uses greedy best-first search to explore the feature space to find the best subset of features. The redundancy of adding a feature to an existing set is computed using information gain as in Eq. 5.8; relevance is calculated using linear correlation values.

5.2.7 Results

As can be seen in Table 5.2, filter-based approaches improve the classification accuracies on average compared to previous approach using entire set of the features; however, accuracies are only slightly better than chance in most cases. Within each dataset, the significantly better classifiers and feature selection algorithms are shown in bold. The CFS and Info-Gain feature selection algorithm perform slightly better than other algorithms in some cases. Decision trees and nonlinear SVM are also superior in terms of classification accuracies. However, in terms of significance there is no single feature selection or classifier that works significantly better than the others over all four datasets according to a non-parametric Kruskal-Wallis \( H \)-test.
<table>
<thead>
<tr>
<th>Metric</th>
<th>Filtering</th>
<th>Feature Sel.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D. Tree</td>
<td>kNN</td>
</tr>
<tr>
<td><strong>k-fold CV</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atlantic Ocean Hurricanes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t$-Test</td>
<td>0.49</td>
<td>0.51</td>
</tr>
<tr>
<td>Fisher</td>
<td>0.40</td>
<td>0.43</td>
</tr>
<tr>
<td>Info-Gain</td>
<td>0.50</td>
<td>0.54</td>
</tr>
<tr>
<td>mRMR</td>
<td>0.44</td>
<td>0.50</td>
</tr>
<tr>
<td>CFS</td>
<td>0.37</td>
<td>0.40</td>
</tr>
<tr>
<td>FCBF</td>
<td>0.35</td>
<td>0.47</td>
</tr>
<tr>
<td>SBMLR</td>
<td>0.49</td>
<td>0.50</td>
</tr>
<tr>
<td>Indian Ocean Hurricanes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t$-test</td>
<td>0.48</td>
<td>0.48</td>
</tr>
<tr>
<td>Fisher</td>
<td>0.59</td>
<td>0.65</td>
</tr>
<tr>
<td>Info-Gain</td>
<td>0.48</td>
<td>0.48</td>
</tr>
<tr>
<td>mRMR</td>
<td>0.59</td>
<td>0.59</td>
</tr>
<tr>
<td>CFS</td>
<td><strong>0.70</strong></td>
<td>0.59</td>
</tr>
<tr>
<td>FCBF</td>
<td>0.59</td>
<td>0.44</td>
</tr>
<tr>
<td>SBMLR</td>
<td>0.50</td>
<td>0.63</td>
</tr>
<tr>
<td>Cognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t$-test</td>
<td><strong>0.64</strong></td>
<td>0.49</td>
</tr>
<tr>
<td>Fisher</td>
<td>0.53</td>
<td>0.52</td>
</tr>
<tr>
<td>Info-Gain</td>
<td>0.55</td>
<td>0.49</td>
</tr>
<tr>
<td>mRMR</td>
<td>0.54</td>
<td>0.49</td>
</tr>
<tr>
<td>FCBF</td>
<td>0.47</td>
<td>0.47</td>
</tr>
<tr>
<td>CFS</td>
<td>0.56</td>
<td>0.43</td>
</tr>
<tr>
<td>SBMLR</td>
<td><strong>0.63</strong></td>
<td>0.49</td>
</tr>
<tr>
<td>Graz BCI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t$-test</td>
<td>0.47</td>
<td>0.45</td>
</tr>
<tr>
<td>Fisher</td>
<td>0.48</td>
<td>0.50</td>
</tr>
<tr>
<td>Info-Gain</td>
<td>0.47</td>
<td>0.45</td>
</tr>
<tr>
<td>mRMR</td>
<td>0.46</td>
<td>0.48</td>
</tr>
<tr>
<td>FCBF</td>
<td>0.45</td>
<td>0.55</td>
</tr>
<tr>
<td>CFS</td>
<td>0.53</td>
<td>0.56</td>
</tr>
<tr>
<td>SBMLR</td>
<td>0.47</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table 5.2: Cross validation accuracies for filter and embedded feature selection algorithms across all four benchmark datasets. Highest and significantly improved statistics are displayed in bold.
Figure 5.3: Decision boundaries for filter-based feature selection algorithms over cognition dataset; $xy$ coordinates represent features with the highest weights as a result of CFS feature selection. Local classifiers in (a) and (b) overfit the data in order to reduce the training error. Model-based parametric (c), linear (d), (e) and non-linear global classifiers (f)-(h) fit more generalizable decision boundaries but could not separate the data into two classes.
When we compare the univariate feature selection methods with multivariate methods, no significant difference between them in terms of accuracies is observed. Similarly, embedded feature selection methods do not result in any better or worse accuracies compared with filter methods. However, we cannot generalize these findings for filter, embedded, or multivariate feature selection methods for different datasets or domains because the data in this form is not separable by selected classifiers. For detailed comparisons of these methods on different datasets and domains, corresponding publications in literature can be searched [51].

In Figure 5.3, the classifier boundaries on the cognition dataset, selected as an example, are drawn in a two-dimensional grid. For visualization purposes we can only draw the first two features with the highest weight as a result of the CFS algorithm. This does not show the complete picture of the classifier capabilities with more features (i.e. in higher dimensions), but it gives an intuition of classifier behavior in lower dimensions. As illustrated, the distributions of the sample points in the first two dimensions (i.e. the highest weighted features) as a result of CFS feature selection leave the data blended. The linear SVM and parametric Naive Bayes classifiers have no chance to classify the data in this form. Non-linear kernel-SVMs were also unable to distinguish the data. Local classifiers, on the other hand, overfit the training data, resulting in low accuracies in the testing phase.

5.3 Dimensionality Reduction, Feature Extraction and Domain Specific Features

Instead of applying feature selection techniques based on generic filtering to existing data, a different approach is to transform the data into a new space, to reduce the dimensionality or define new or domain specific features using expert or domain knowledge. For this purpose, we looked at the features generated by Principal Component Analysis (PCA), a commonly used algorithm for dimensionality reduction and feature transformation that works for data in any domain. For climate, we use indices specifically defined by climatologists over the years to explain climate phenomena. For EEG data, we have already transformed the signal into its frequency components. In addition to this transformation, we test the performance of Common Spatial Patterns (CSP) in both domains, a spatial feature selection method proven to be effective especially for EEG-based systems. In the
literature we find a small number of studies that use the correlations between time series as features, even though it results in a higher-dimensional feature space. Here we also employ correlations as features in the climate domain and coherences as features in the cognitive science domain. In the following subsections, we briefly explain the mechanics of each method.

5.3.1 Principal Component Analysis

PCA is a dimensionality reduction technique based on orthogonal transformation of data dimensions in order to map a set of correlated features into a set of uncorrelated features. The generated features are called principal components, and they are ordered in terms of how much they explain the variance in the original data. The first component, for instance, can be calculated by solving the following equation:

\[ w_{(1)} = \arg \max_w \left\{ \frac{w^T X^T X w}{w^T w} \right\}. \quad (5.9) \]

The additional principal components can be found by subtracting previously calculated components from the data and solving the same equation.

5.3.2 Common Spatial Patterns

Common Spatial Patterns filtering is a spatial feature selection algorithm that has been effectively applied in many EEG-based BCI applications. CSP aims at projecting multi-channel EEG data into a low-dimensional spatial subspace such that the goal of the transformation is to maximize the variance between input signals originated from different classes. The objective function for CSP is

\[ w = \arg \max_w \frac{||wX_1||^2}{||wX_2||^2}. \quad (5.10) \]

This equation can be solved by simultaneous diagonalization of covariance matrices of the input signal originating from each class.

5.3.3 Climate Indices

Climatologists use climate indices to explain the changes in a climate system. As explained in Chapter 2, there exist many different indices calculated from different atmo-
<table>
<thead>
<tr>
<th>Metric</th>
<th>Features</th>
<th>Atlantic Ocean Hurricanes</th>
<th>Indian Ocean Hurricanes</th>
<th>Cognition</th>
<th>Graz BCI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>D. Tree</td>
<td>kNN</td>
<td>N.Bay</td>
<td>SVM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>k-fold CV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Climate Indices</td>
<td></td>
<td>0.71</td>
<td>0.63</td>
<td>0.43</td>
<td>0.62</td>
</tr>
<tr>
<td>CSP</td>
<td></td>
<td>0.63</td>
<td>0.62</td>
<td>0.56</td>
<td>0.60</td>
</tr>
<tr>
<td>PCA</td>
<td></td>
<td>0.54</td>
<td>0.54</td>
<td>0.50</td>
<td>0.49</td>
</tr>
<tr>
<td>Correlation</td>
<td></td>
<td>0.72</td>
<td>0.73</td>
<td>0.57</td>
<td><strong>0.79</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Climate Indices</td>
<td></td>
<td>0.66</td>
<td>0.55</td>
<td>0.57</td>
<td>0.61</td>
</tr>
<tr>
<td>CSP</td>
<td></td>
<td>0.59</td>
<td>0.67</td>
<td><strong>0.69</strong></td>
<td><strong>0.69</strong></td>
</tr>
<tr>
<td>PCA</td>
<td></td>
<td>0.54</td>
<td>0.56</td>
<td>0.57</td>
<td>0.59</td>
</tr>
<tr>
<td>Correlation</td>
<td></td>
<td>0.56</td>
<td>0.56</td>
<td>0.54</td>
<td><strong>0.74</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSP</td>
<td></td>
<td>0.56</td>
<td>0.56</td>
<td>0.55</td>
<td>0.56</td>
</tr>
<tr>
<td>PCA</td>
<td></td>
<td>0.56</td>
<td>0.63</td>
<td>0.63</td>
<td>0.50</td>
</tr>
<tr>
<td>Coherence</td>
<td></td>
<td>0.68</td>
<td>0.76</td>
<td>0.76</td>
<td><strong>0.91</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSP</td>
<td></td>
<td>0.46</td>
<td>0.47</td>
<td>0.53</td>
<td>0.51</td>
</tr>
<tr>
<td>PCA</td>
<td></td>
<td>0.54</td>
<td>0.52</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>Coherence</td>
<td></td>
<td>0.49</td>
<td>0.62</td>
<td>0.58</td>
<td><strong>0.73</strong></td>
</tr>
</tbody>
</table>

Table 5.3: Cross validation accuracies for specific feature selection algorithms across four benchmark datasets. Highest and significantly improved statistics are displayed in bold. Climate indices are domain specific features for climate data. CSP as a supervised feature selection technique developed for brain EEG is also applied in climate data. PCA is a generic dimensionality reduction technique. Correlations between times series in climate data, and coherences in EEG data are implemented as problem/domain specific features.

Spheric variables that capture specific aspects of climatology. For the hurricane tracking classification problem, we use values of 11 different climate indices available from the NCAR/NCEP portal [1]; these are commonly applied in climate analytics as input features for classification problems. Since the climate indices are calculated monthly, we use specific index values for each hurricane, matching prior to event occurrence dates.

### 5.3.4 Correlations / Coherences as Features

In addition to features selected and generated from the input data, features specific to systems exhibiting oscillatory characteristics are also investigated. Correlations between
spatial locations in climate data and coherence in input signals between two electrode positions in EEG data [96] are defined as features distinct from the above-mentioned generic and domain-specific feature definitions.

5.3.5 Results

In Table 5.3, the classification accuracies for domain specific feature selection and feature generation algorithms are listed. For the hurricane classification problem, correlation-based features on average perform better than domain-specific features and PCA. In the cognition dataset, coherences in the signal between spatial locations as features outperform all other feature selection algorithms and result in very high classification accuracy, e.g. 91%. In the Graz BCI dataset, the accuracies for coherence features again perform better than the features generated by the CSP and PCA algorithms.

We conclude that correlations/coherences between spatial locations result in better predictors than CSP features and domain-specific indices. A statistical analysis (i.e. the Kruskal Wallis H-test) also shows that correlation/coherence-based features result in significantly better classifiers ($p < 0.05$) than CSP and PCA. As an unsupervised feature generation algorithm, PCA performs significantly worse than others. Compared to generic feature selection algorithms, domain specific features (i.e. climate indices and CSP) are better predictors, on average. In terms of model selection, linear and nonlinear SVMs perform significantly better than local and parametric classifiers.

The decision boundaries of coherence features are illustrated in Figure 5.4. For visualization purposes, we select the first two features of a CFS feature selection algorithm over coherence values. In comparison with the generic feature selection methods illustrated in Figure 5.3, the data instances are spread more heterogeneously in the feature space, and classifiers, especially model-based linear and nonlinear SVM, are able to separate the data from two classes. However when the cross validation accuracies are taken into consideration, linear, polynomial and $rbf$ kernels outperform sigmoid kernels. Even though on the entire dataset the decision boundaries look similar to those produced by nonlinear classifiers, the sigmoid did not perform well during the cross-validation phase; this was also the case for the local $k$NN and decision tree classifiers. This could be an indication of an overfitting effect of these classifiers compared to other kernel-based classifiers, which have better generalization ability.
Figure 5.4: Decision boundaries for domain-specific (coherence) features on cognition dataset; $xy$ coordinates represent coherence features with the highest weights as a result of CFS feature selection applied for visualization purposes. As local classifiers $k$NN (a) divides the space into local neighborhood regions, decision trees (b) divide the space into local rectangles. Parametric Naive Bayes (c) distinguish the data based on class distributions. Linear (d), (e), and nonlinear kernel classifiers (f)-(h) fit a global model on the data.
Figure 5.5: Performance comparison of local vs model-based classifiers on cognition data using coherences as features. On the left, the accuracies are shown when the same subject’s data is not allowed to be in both training and testing sets (heterogeneous data) at the same time, and on the right, random sampling where the test data is randomly selected among all instances and the data from the same user is allowed to be in both training and testing sets.

**Classifier and kernel method selection in feature-based classification:**

In terms of classifier selection, when we analyze the Table 5.3., it can be seen that model-based global classifiers such as linear and non-linear SVMs and model-based local decision trees outperform the local (kNN) and parametric (Naive Bayes) classifiers. However when we enable random sampling during cross validation, accuracies of local classifiers increase significantly as can be seen in Figure 5.9, since we allow more similar instances such as data from the same user in both test and training datasets. This finding indicates that when there are more similar instances between training and test sets, is a higher chance that local classifiers perform better.

In terms of kernel function selection for SVM classification, the accuracies calculated throughout the result section varies depending on the dataset. Therefore, there is no particular function outperforms all other kernel functions.

In literature, two main factors are pointed out for deciding which kernel to use. The first factor is related to sample size and number of features. Hsu et al. [48] discuss that when the number of features are too large compared to number of training instances (i.e. underdetermined problems) using linear kernel is a better choice because there is not
enough data to fit a nonlinear function. On the other hand, if available training instances are large compared to features, then, nonlinear kernels such as $rbf$ can be applied safely.

The second factor is the selection of model parameters of the optimization function, because linear kernels are special case of $rbf$ kernels and can be obtained chaining some of the model parameters [59]. One can directly start with $rbf$ and find the parameters that has the best validation accuracy. The resultant nonlinear kernel may be very similar to the linear one.

In this study, in order to compare the kernel functions with each other, the default parameter values for each optimization function is used. We suggest that, in selection of the kernel methods, the classification accuracies in validation step is a good indicator to understand the performance of a selected classifier, because good validation accuracy is also an indication for a better fit for the data, especially in cases visual analysis is not possible due to higher dimensions. Empirically, linear and polynomial kernels performs slightly better than $rbf$ kernels in selected datasets. Sigmoid kernels perform worse among other kernels in most of the experiments throughout this chapter. However, for a future reference, it could be possible to add a step for parameter search as a nested validation step prior to model building to find the best performance for each kernel under certain parameter configuration [105].

### 5.4 Similarity-Based Classification

Finally, we analyze spatiotemporal systems displaying distinct output behavior as integrated systems, instead of focusing on specific features at specific time and locations; it is reasonable to assume that similar processes will exhibit similar output behaviors. To analyze spatiotemporal systems from a similarity standpoint, we propose two main approaches. In the first, similarities between time series data at each spatial location of two processes are calculated as an input to similarity-based classification algorithms. In the second approach, entire systems are accepted as structural entities, such as networks formed throughout the time, and pairwise similarities between these structural entities are calculated for dissimilarity-based classification.
<table>
<thead>
<tr>
<th>Metric</th>
<th>Similarity-based classifiers</th>
<th>SVM</th>
<th>kNN</th>
<th>IndSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-fold CV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atlantic Ocean Hurricanes</td>
<td></td>
<td>0.50</td>
<td>0.42</td>
<td>0.54</td>
</tr>
<tr>
<td>Indian Ocean Hurricanes</td>
<td></td>
<td>0.52</td>
<td>0.57</td>
<td>0.52</td>
</tr>
<tr>
<td>Cognition</td>
<td></td>
<td>0.55</td>
<td>0.52</td>
<td>0.50</td>
</tr>
<tr>
<td>Graz BCI</td>
<td></td>
<td>0.60</td>
<td>0.44</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Table 5.4: Cross validation accuracies for similarity-based learning algorithms using temporal similarities across spatiotemporal systems.

5.4.1 Similarities across components of spatiotemporal systems

As a first approach, as explained in Section 3.1, the direct similarities between two spatiotemporal systems are calculated. To achieve this, at each spatial location \( k \in \{ s_1, s_2, ..., s_n \} \) across two spatiotemporal grids \( s_{g1}^k \) and \( s_{g2}^k \), similarity is defined as a linear combination of correlations (i.e. Pearson’s \( r \)) between time series as in Equation 3.2.

As can be seen in Table 5.4, direct similarity measures between spatiotemporal systems do not yield good classification accuracies. The reason for low classification accuracies might be related to phase differences between samples or due to the behavior of the time series; for example, they might not follow similar trends in different time periods. As a next step, we instead look for more robust similarities, such as structural similarities, between spatiotemporal systems.

5.4.2 Structural Similarities Between Networks and Classification

As an alternative to direct similarities between spatiotemporal systems, we can look at the structural similarities between network structures. One difference from the previous approach is that instead of similarities that may be arbitrary and thus spurious, we first look at the teleconnections occurring in longer time frames within a system. Using these teleconnection patterns, a network is associated with each event, then structural similarities between network structures are compared for the purpose of classification.
<table>
<thead>
<tr>
<th>Metric</th>
<th>Tabular Filtering</th>
<th>Feature Sel.</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-fold CV</td>
<td>D. Tree</td>
<td>kNN</td>
</tr>
<tr>
<td>Atlantic Ocean Hurricanes</td>
<td>spectral 0.54</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>MDS</td>
<td>0.63</td>
</tr>
<tr>
<td>Indian Ocean Hurricanes</td>
<td>spectral 0.76</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>MDS</td>
<td>0.65</td>
</tr>
<tr>
<td>Cognition</td>
<td>spectral 0.57</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td>MDS</td>
<td>0.66</td>
</tr>
<tr>
<td>Graz BCI</td>
<td>spectral 0.58</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td>MDS</td>
<td>0.43</td>
</tr>
</tbody>
</table>

Table 5.5: Cross validation accuracies for classifiers using spectral graph features extracted from each network separately, and multi-dimensional scaling embeddings of graph edit distances. Highest and significantly improved statistics are displayed in bold.

One direct way of classifying these networks (i.e. graphs), represented in adjacency matrix form, is through spectral embedding and using resultant spectral components as features for the classification algorithms. Alternatively, the dissimilarities between networks can be calculated for a more detailed comparisons of graphs.

To quantify the dissimilarities, graph edit distances between network structures are computed algorithmically as explained in Section 3.3.2. Once pairwise graph edit distances are available for every training instance represented in a distance matrix $D$, two main approaches can be employed to classify this matrix. Using classical scaling, the distances can be mapped to a Euclidian space $X$ or, alternatively, instead of dissimilarity-based classification, the distances in $D$ can be converted into similarities $S$, and similarity-based classification methods can be applied to the resulting matrix. In the following section, we first present the results for spectral embedding of network structures, then employ distance-based classification methods, and finally include the results for similarity-based classification of network structures.
Figure 5.6: Decision boundaries for spectral features on the cognition dataset; $xy$ coordinates represent eigenvalues corresponding to largest eigenvectors. The data is scattered in two dimensions. Local classifiers in (a) and (b) overfits the data in local regions, parametric (c) and model-based linear classifiers (d), (e) fit a global model to enable generalization. Nonlinear kernel-based classifiers in (f)-(h) create nonlinear decision boundaries for the data.
5.4.3 Spectral Features

Given a graph adjacency matrix $A$, some important information can be computed from this matrix. The spectral graph theory states that eigenvalues and eigenvectors of this matrix will be the same if two graphs are isomorphic, and this information can be used, for instance to match two arbitrary graphs. This is called spectral embedding of the graphs and works by mapping the eigenvalues of the graph in a coordinate system.

As stated above, if two graphs are isomorphic then their eigenvalues and corresponding eigenvectors will also be the same. This notion can be extended to non-isomorphic graphs such as if two graphs are similar then their spectrum will also be more similar at least compared to their non-similar counterparts. The spectral components of graphs, namely leading eigenvalues and eigenvectors of adjacency matrix (or closely related form, Laplacian matrix), have previously been used to define the characteristics of the graphs for clustering or classification purposes, especially for spatial graphs.

Wilson et al. [118] have used various graph representations, such as Adjacency, Laplacian, Heat and Common Path matrices to extract graph spectrum. We compared the binary, regular adjacency, and Laplacian representations of the graphs. The spectral features on the unweighted (binary) representation of the graphs produce classification results that are slightly better than chance; the problems that are addressed for spectral components, such as being sensitive to structural differences and missing nodes, affect classifier accuracies. The spectrum of weighted matrices, on the other hand, performs better than the unweighted (binary) counterparts and spectrum of Laplacian matrices. Therefore, we use the spectral features extracted from weighted matrices of the graphs. The results of the classification accuracies are listed in Table 5.5.

In Figure 5.6, the decision boundaries on the first two eigenvalues of each network instance for all classifiers are represented in $xy$-coordinates. As can be seen from the figure, the distribution of the embedded graph features is quite homogenous, and no linear separation is possible with these features. Non-linear and local classifiers seem to be producing more promising decision boundaries, but this results in low cross validation accuracies as can be seen in Table 5.5.
Figure 5.7: Classification decision boundaries for MDS embeddings; the coordinate system lies in the eigenspace where x and y axes correspond to the largest variations. The distribution of the data is mostly explained by the x-component. Local classifiers in (a) and (b) overfit the data. Parametric (c) and linear classifiers in (d)-(e) fit simple decision boundaries. Nonlinear classifiers in (f)-(h) create decision boundaries without gaining or loosing too much in generalization.
5.4.4 Multi-dimensional Scaling

As explained in the previous chapter, pairwise graph edit distances between brain and climate networks represented in a distance matrix $D$ can be used as a direct source for classification. Once we calculate the graph edit distances using the procedure explained in Section 3.3.2 we can use this pairwise distance matrix for the purpose of classification. We can use a neighborhood-based algorithm without a model, or we can embed these distances into artificially generated feature vectors using Multi-Dimensional Scaling (MDS).

MDS is a method for mapping the $n \times n$ distance matrix $D$ in low dimensional $n \times p$ configuration matrix $C$ such that the distances in this new space $D'$ are as close as possible to original distances in $D$. Rows of $C$ are the coordinates of $n$ points in $p$-dimensional space for some $p < n$.

In Figure 5.7, the first two dimensions of the vectors as a result of scaled distances are drawn on the $xy$-coordinates. The selected $x$ and $y$ dimensions also reflect the largest variance in the configuration. As can be seen in the figure, the distribution of the data along the $x$-axes helps to create generalizable decision boundaries, especially for kernel-based methods. In Table 5.5, we see the results of classification algorithms applied on the vectors obtained by scaling of graph edit distances. The results show that MDS outperforms classifiers based on embedded spectral graph features across all test datasets. In terms of classifier selection, on average the global classifiers perform better than local ones; however, the results are not significantly different.

5.4.5 Converting Distances into Similarities

Once the graph edit distances (i.e. dissimilarity measures) are at hand, using the transformation techniques explained in Section 3.4.1, they can be converted into similarities. As a next step, learning methods that work on similarity spaces can be employed as outlined in Section 3.4.1.

For this purpose we employ four different monotonically decreasing transformation functions, listed in Equation 3.13. In Figure 5.8, the converted similarity matrix for each dataset is displayed for visualization purposes. By looking at these visualizations, one can draw some conclusions about the quality of the similarity scores at hand. The interpretation is this: if the matrix is divided into four quadrants, the upper left and upper right corner quadrants are intra-class similarities between networks for the first
Table 5.6: Cross validation accuracies for (dis)similarity-based classification of graph edit distances after transformed into similarities through four different ($S_1$, $S_2$, $S_3$, $S_4$) linear and non-linear transformation functions as listed in Equation 3.13. Highest and significantly improved statistics are displayed in bold.

and second class respectively; the upper right and lower left quadrants are symmetric and represents inter-class similarities. The diagonal of each figure shows the self similarities for each instance and therefore has the highest color value.

We apply standard SVM, spectrum modification and Indefinite SVM algorithms on the test datasets. As can be seen in Table 5.6, three out of four datasets with trivial linear transformation $S_1$ perform better than nonlinear ones. In the first dataset a nonlinear transformation of similarities performed better. However, the differences per classifier were not statistically significant. If the effect of the classifier is not included, the linear transformation performs significantly better than nonlinear ones.

Compared to previous graph-based classification methods listed in Table 5.5, one or
Figure 5.8: Similarity matrix representation for each sample dataset, similarities between instances acquired from (a) Cognition dataset, (b) Atlantic Ocean hurricanes, (c) Indian Ocean hurricanes, and (d) Graz-BCI experiment data.

More (linear or nonlinear) similarity-based classification algorithms perform better than spectral feature embedding or MDS methods.

Clip, Flip, Shift or Indefinite Kernels?
When the effect of each spectrum modifications on the test datasets is analyzed, it can be seen that each spectrum operation performs differently at each dataset. To understand the behavior of those operations, the spectra of the graphs can be investigated. For datasets where the negative eigenvalues of the similarity matrix are large, the variations in accuracies are also high. The spectrum modifications working on the negative eigenvalues change the original similarity matrix $S$ significantly.

The question which spectrum method should be chosen therefore depends on two factors. The first is the characteristics of the embedded graphs; the second is the accuracies over validation sets. If negative eigenvectors are small, it is possible that any of
the spectrum modifications will work well with the data. A clip operation, as suggested by Gupta et al. [17], could be the preferred method since its effect is similar to noise removal. For larger negative eigenvalues, clipping may cause the loss of important information from the data. Shifting, which works well in our test datasets, can be employed after a validation step over the training data.

5.5 Comparison of Feature and Similarity-based Approaches

In Figure 5.9 we compare the classification accuracies for the approaches investigated throughout the current chapter. We show the results of best of the (linear and nonlinear) SVM accuracies listed in Table 5.1 to Table 5.6 to compare the different approaches. In Figure 5.9, the trend across all datasets can be seen in terms of average accuracies. The tabular embedding of spatiotemporal dataset in combination with standard classification algorithms on raw data performs worst among all, and generic (filter-based) feature selection accuracies improve the raw data classification accuracies slightly. Domain specific feature selection algorithms such as climate indices, and CSP perform better than
When we shift gears towards network-based approaches, the features generated via spectral graph embedding techniques improve the accuracies over feature-based approaches. Multidimensional scaling of the graph edit distances performs better than graph embedding techniques. Our proposed graph edit distance-based network classification on the other hand outperforms both the spectral embedding and MDS approaches. The highest accuracies were achieved using weighted network edges as features.

In Figure 5.10, the statistical analysis results for Kruskal Wallis $H$-test is presented. In terms of statistical significance, the similarity-based classification of network instances and correlation/coherence features extracted from the networks performed significantly better over generic feature-based approaches and tabularly embedded raw data.

To complete the analysis, we compare our results with the accuracies published by Scherer et al. in their recent study [97] on public Graz-BCI mental state classification task dataset. Their highest classification accuracy in the same dataset (hand vs word association task) is less than $\leq 70\%$ compared to our highest accuracy 73% obtained from network-based methods proves the validity of our approach in mental and cognitive state classification problems.
Chapter 6

Conclusions

Data-driven models overcome the limitations of alternative approaches to understanding complex systems, such as simulation, which has been popular in many of the sciences. Using data-driven techniques, we are able to make inferences about a system under consideration, create statistical models to make predictions about future behavior, and interpret conditions causing events to happen by analyzing available (historical) data with limited knowledge of underlying processes. In this work, using the models we generate from the data, we have examined different ways of predicting the behavior of observable events emerging from complex physical systems in the domains of climate and the human brain.

To apply data-driven techniques in a given domain, we start by analyzing the characteristics, the challenges and the behavior of the data acquired from the domain. Scientific datasets, climate and brain data in particular, bring challenges that needs to be handled carefully if a data-driven model is to be successful. One challenge is the typical nature of scientific datasets being spatiotemporal: strong spatial and temporal autocorrelations often exist in the data. Another challenge is that the number of training samples can be limited or difficult to obtain, and therefore problems are usually underdetermined: there are more features than available samples. In this research, we have analyzed two factors that could enable more accurate models: (1) the value of the information that different predictors and data representations bring and (2) the effect of different type of classifiers and decision functions.

For the first factor, different predictors and data representations were investigated and their results systematically compared. An important part of any machine learn-
ing project is to find most informative feature sets to represent a specific type of data. Our analysis has shown that some domain-specific features perform significantly better than features selected using statistical methods, mainly because they naturally solve the autocorrelation issues by filtering spatial and/or temporal components, therefore representing the data in a more refined way. Additionally, different data representations, such as network-based approaches, can bring new sources of information into the equation. A network-based approach allows us to compare one system as whole with others, by representing the systems as structural entities instead of features. Our results show that network-based approaches are valid and potentially better than feature-based approaches for the analysis of climate and functional brain datasets.

For the second factor, the effect of classifiers, when we analyze the distribution of data, we find that some predictive variables distribute the data more heterogeneously between different classes and allow (model-based) classifiers to develop more generalizable linear and nonlinear decision boundaries. However, when these predictive variables cannot separate the data from different classes well enough, as we saw in the example graphs in Chapter 5, the local classifiers overfit the data to boost their training accuracies but most likely at the cost of losing generality. On the other hand, global classifiers working on these variables produce decision boundaries with high training errors. For under-determined problems, if the predictors are not informative enough, it is likely that a trade-off between the fitness of the model and validation (testing) accuracies will be faced. To overcome this problem, it would be worthwhile to investigate ensemble methods (discussed in Section 6.5 but outside the scope of this research) that could help to create more generalizable decision boundaries using classifier ensembles.

In the remaining of this chapter, we present the contributions of our work from computer science, cognitive science and climate science perspectives separately. Then we discuss the limitations of the study and possible future research directions.

## 6.1 Contributions to Computer Science

As stated above, scientific datasets bring challenges that needs to be handled carefully if a data-driven model is to be successful. One challenge is the nature of spatiotemporal scientific datasets; there is no consensus in the machine learning community about how to handle spatiotemporal datasets, especially from the classification point of view. In
In this respect, one of the main contributions of the current work is to draw a unified framework for spatiotemporal data analysis in a tree-like structure. We also implement techniques that each branch of this method tree represents, and we have systematically compared their accuracies on benchmark datasets. Chapter 5 discusses their advantages and disadvantages on example problems. In the future, more domains or datasets sharing the same underlying principles can be added to the application areas of the techniques presented, or the method tree can be extended with new approaches which can be used to analyze spatiotemporal systems.

In addition to feature-based approaches, in this dissertation we have introduced similarity-based classification for spatiotemporal systems. For this purpose, we propose two novel alternatives: direct similarities between time series data over spatial locations across training instances, and (dis)similarities based on graph edit distances over the network structures associated with each instance. The first alternative, namely direct similarities across two spatiotemporal systems, did not provide us with a good metric for classification, and requires more work.

In network-based classification, on the other hand, the proposed graph-edit distance-based (dis)similarity classification method outperforms the eigenvalue decomposition of spatial graphs and other graph centric methods. When we compare different approaches using a public EEG-based cognitive data (i.e. Graz-BCI), we find that the proposed method perform better than the classification accuracies reported on the original paper of the corresponding study. In comparison to traditional feature-based approaches, network-based approaches are promising alternatives not only because they provide higher classification accuracies but also because the networks are more robust structures built over the course of time in comparison to dynamic and spurious features represented by single data points or as vectors.

In our network classification approach, we mainly use the structural classification techniques based on graph matchings. However, existing graph matching algorithms such as standard graph edit distances are effective in the context they are created for such as computer vision applications, and do not satisfy the requirements of spatial network classification. To fill the gap, we present a novel graph matching algorithm for spatial graphs. This approach is computationally efficient and can be implemented to larger graphs due to its polynomial time complexity. The proposed matching function also serves as a similarity score between networks, and it is a more suitable metric compared
to existing graph kernels produced for specific applications such as molecular and chemical structure classifications.

6.2 Contributions to Cognitive Science

Cognitive systems such as brain computer interfaces are challenging architectures due to the nature of the problem, because well known characteristics of EEG data is being non-stationary and noisy. Therefore, fitting a decision function on EEG data requires carefully designed prepossessing and feature selection steps to allow noise-free and robust features in the target model. In this study, we present the idea of using network-based classifiers to EEG data instead of feature-based approaches that suffer in performance due to the non-stationary nature of the data. The results show that the proposed method help us to achieve better classification accuracies (i.e. \( \geq 90\% \)) in comparison to existing feature-based techniques on a difficult inter-subject classification problem (i.e. cognition dataset) where a subject’s EEG pattern is classified using a model built upon the data acquired from other system users.

Another important observation deals with a common paradigm in functional brain experiments. In cognitive science studies, current practice is to ask subjects to visualize certain actions such as imaginary motor movements or mental arithmetic tasks, in order to generate differentiable patterns in their recorded brain signals to build a successful machine learning model. An example is the experiments that produced the Graz-BCI dataset, where the goal is to find out which pairwise imaginary actions create the most distinct patterns for binary classification. However, the system gives no explicit assistance to subjects to trigger those imaginary action sequences, which influences the quality and separability of the data. Fluctuations in classification accuracies or spurious changes in values of the selected features during task execution can easily be caused by subjective reasons not related to the goals of the experiment. In other words, they can simply be noise from the classification point of view. That would eventually affect the success of the system built upon such an experimental paradigm. In contrast, the experiments designed for the 2D-3D mental transformation task (i.e. the cognition dataset in this dissertation), the imaginary actions are supported by visual cues and feedback loops allowing the quality of the data to be better in comparison to experiments based on imaginary action paradigms. By looking at the highest classification accuracies achieved,
we can state that design of the experiment is an important factor for success in inferring some cognitive processes in studies of this kind, and accuracies are much higher when there is a some kind of feedback loop or association between the problem asked and the task being executed.

6.3 Contributions to Climate Science

Forecasting hurricane behaviors such as characteristics, seasonal counts, and impact levels are important open questions in climate research; hurricanes can lead to major disasters, especially when they make a landfall. To better understand and predict the behavior of regional hurricanes, it is important to consider the factors influencing the dynamics of their tracks. Hurricane tracks have been studied in the literature in terms of clustering of their patterns into distinct groups, then from association mining point of view, attempts are made to identify climate variables correlated to different clusters of hurricanes. In this study, we present the idea of forecasting hurricane end trajectories as a classification problem. By looking at the historical data, we attempt to predict the outcome of a hurricane originating from a certain location, specifically whether it will hit land or remain offshore.

In this process, instead of feature-based approaches, we present the idea of network-based classification, since networks can capture spatial and temporal changes and better utilize the available data. The results indicate that when classifying end trajectories of hurricane tracks, network-based approaches are valid alternatives to feature-based methods. Application areas can be extended to predicting other types of climatological (extreme) events in the future.

6.4 Limitations

In this research, we have explored different avenues to build accurate data-driven models. Another factor, however as important as statistical foundations, is the quality of the data itself. For instance, in cognitive science studies, current practice is to ask subjects to visualize certain actions such as imaginary motor movements to generate distinct brain patterns and success of the cognitive architectures depends on the quality of the data obtained from such kind of experimental process. One important research area for
cognitive studies and BCI-like systems should be exploring better experimental design strategies in order to produce more reliable datasets that would allow more stable machine learning models, in return leading to more usable, robust systems. Therefore, the signal quality is a limiting factor for success of the classifiers especially in one of the benchmark datasets (i.e. Graz-BCI). Similarly, in climate science, the data we use is *NCEP-NCAR 50-years reanalysis data*. It is one of the most widely used and reliable data currently available, however still a product of model simulations, and to some extent it is assimilated rather than observational data [63]. Hence, as in cognitive science research, in climate, data itself might be the limiting factor for the success of the generated model.

In terms of applicability of the proposed methods to the problems from other disciplines, it is important to note that the proposed techniques are developed for two distinct domains climate and brain; these domains share similar underlying characteristics such as the data being collected from a set of sensor locations over a period of time. For that reason, the data instances (i.e. spatiotemporal processes with a labelled outcome) share the same spatial infrastructure in both domains. Therefore, for each application the similarities between data instances can be computed directly using the spatial information as part of the equation. Spatiotemporal datasets, on the other hand, can emerge in many different forms and characteristics and may or may not share the same physical constraints that climate and brain datasets have in common. That could be a limiting factor to apply the same ideas into different domains directly. By implementing some additional intermediate steps, the feature and (dis)similarity-based approaches proposed in this study may still be applied to different domains, because despite their differences, they would still be fitted under the umbrella of spatiotemporal data classification framework.

In this study, we apply spatial graph edit distances into (dis)similarity-based classification framework and utilize learning algorithms including kernel classifiers with spectrum modifications, indefinite SVMs and neighborhood-based approaches since they are promising alternatives for the definitions and constraints of the example problems. However, there are other alternative (dis)similarity-based learning approaches (briefly covered in the background and at the end of method sections) such as similarities based on graph kernels and dissimilarity space embeddings based on the graph edit distances on prototype sets. Such approaches are outside of the scope of the current work, mainly because in the former care we do not have graph kernels specifically defined for the spatial networks, and in the latter case, for underdetermined problems, searching for a prototype
set may not be feasible due to the limited size of the sample space. We also do not go into a hybrid learning approach that could merge feature and network-based approaches within a single classification framework; this is discussed in the following section.

6.5 Future Work

In the beginning of this work, different feature sets and data representations were analyzed for the purpose of classification of spatiotemporal data. Results obtained from one set of features were compared with others. As an additional step, hybrid models that merge features coming from different sources within a single model can be explored in the future. For example, in a fraud detection framework, the goal is to classify online credit card transactions to find out whether they are legitimate or fraudulent [115]. For this purpose, a set of available signatures would be defined as features in real time such as frequency and amount of transactions. Another source of information is the social interactions between people who committed the fraud, represented as social or transactional networks. Features extracted from these networks and the features from the transactional data could be combined under a common logistic regression model and a final decision made using information from both sources.

We could also use a similar hybrid classification technique such as features from network-based approaches and features from generic or domain-specific selection strategies; these would be merged within a single model. Which network features are most informative for the respective domains is another research problem that needs to be investigated.

To improve model accuracies, another possible addition to current work could be utilizing ensemble techniques [71]. For instance, to improve the stability of a classifier and to reduce the variations due to sampling, bagging techniques can be applied. In bagging, randomly sampled training instances form a sample set, and for each sample set a classifier is trained, forming a pool of classifiers in the end. Decisions are made as a result of a voting scheme where each classifier in the pool has an equal or weighted effect on the output. Another option is to use boosting. In boosting, misclassified instances during training phase receive higher weights, and an iterative process attempts to classify those instances correctly. In each iteration we create a weak classifier that learns only some part of the decision space to create a strong “boosted” classifier at the end.
In terms of application areas, in climate science, the current work can be extended to the classification of hurricane tracks from different regions on earth such as predicting Eastern and Western Pacific hurricanes. Additionally, other climatological events such as occurrence of draughts or precipitation extremes can be analyzed using network-based approaches. For cognitive science, the proposed methods can be applied in cognition related studies and other BCI applications. However, since the goal of a BCI application is usually to control external devices in real time, they may require faster response times, and networks formed in longer time periods may not be suitable for that purpose. Still, different network construction approaches requiring shorter time intervals to develop can be explored. The same classification framework may be applied on these types of networks as well. Finally, open classification problems from other fields of science sharing the same spatiotemporal characteristics and physical constraints can be explored and potentially solved using the similarity and network-based approaches presented here. The observed results can be compared with the results of existing techniques in the corresponding application areas to further validate the applicability of the proposed methods in different domains.
BIBLIOGRAPHY


[74] Terrence A McCloskey, Thomas A Bianchette, and Kam-Biu Liu. Track patterns of landfalling and coastal tropical cyclones in the atlantic basin, their relationship with the north atlantic oscillation (nao), and the potential effect of global warming. 2013.


