A NEW REPRESENTATION METHOD FOR MULTIVARIATE TIME SERIES CLASSIFICATION PROBLEM USING INTERVAL MEANS AND POLAR HISTOGRAM DENSITIES

by

Nurettin Dorukhan Sergin

B.S., Industrial Engineering, Boğaziçi University, 2015

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in M.S. in Industrial Engineering

Boğaziçi University

2017

ACKNOWLEDGEMENTS

While this thesis is a product of a yearlong hard work and devotion, my efforts are dwarfed by the invaluable contributions of many respectable individuals. Thus, I would like to dedicate the following words to them which will still not be enough express my indebtedness.

I would like to start with Dr. Mustafa Gökçe Baydoğan, my former undergraduate dissertation supervisor and current master's thesis supervisor for simply making all of these possible. He has always patiently appreciated my headstrong approach and found ingenious ways to turn it into useful ideas and efforts. It is almost miraculous that he had joined Boğaziçi University right before I graduated and I am more than proud to be following his footsteps to the other side of the globe now.

My sincere gratitudes goes to Dr. Cemal Deniz Yenigün and Dr. Gönenç Yücel for kindly accepting to be in the dissertation committee of this thesis and their valuable comments. More importantly, both of these two extremely talented academics had been a major influence on my decision to pursue a Ph.D degree and helped me grow as a scientist with their brilliant guidance.

This thesis could not have been completed without the endless support of my mother and my father who selflessly alleviated many barriers in front of me, without even letting me notice them. My indebtedness naturally extends to my sister who is the ultimate source of joy in my life and my guiding spirit who put me back on track whenever I needed to.

Last but not least, I would like to thank my dearest friends for supporting me all the way during this thesis year even if this meant that we will be continents apart at the end of it. I apologize them for every time I could not be with them, sometimes just because I had to spent hours working on a minor issue about the problem.

This research was partially supported by the Bogazici University Research Fund (BAP) under grant 14A03SUP6.

CMU MOCAP S16, KickvsPunch and WalkvsRun datasets used in this project was obtained from mocap.cs.cmu.edu. The Graphics Lab Motion Capture Database was created with funding from NSF EIA-0196217.

ABSTRACT

A NEW REPRESENTATION METHOD FOR MULTIVARIATE TIME SERIES CLASSIFICATION PROBLEM USING INTERVAL MEANS AND POLAR HISTOGRAM DENSITIES

Multivariate time series (MTS) classification is an instance of common time series data mining tasks and is ubiquitously found in many domains such as medicine, finance or human-computer interaction. Traditionally, the research community has approached the problem by extending the well-established methods available in the univariate time series (UTS) classification literature. In this work, a new feature based method is developed specifically for MTS classification and aims to capture not only features of individual univariate series—as the extension methods do— but also the interaction between them. The method utilizes simple interval statistics as the base feature and polar histogram densities to represent 2-way interactions. The feature vectors are processed with a random forest classifier for its ability to handle high-dimensionality. The results are reported for benchmark datasets of various types and from a range of domains. The method provides a satisfyingly accurate and scalable solution to the problem. The 2-way interaction information significantly increases the accuracy in most of the cases while the extraction phase of this information dominates the computation time. The method is comparable to the state-of-the-art methods in the literature even though there is a significant room for improvement.

ÖZET

ÇOKDEĞİŞKENLİ ZAMAN DİZİSİ SINIFLANDIRMA PROBLEMİ İÇİN ARALIK ORTALAMALARI VE KUTUPSAL HİSTOGRAM YOĞUNLUKLARI KULLANILAN YENİ BİR TEMSİL METODU

Çokdeğişkenli zaman dizisi sınıflandırma problemi zaman dizilerinde veri madenciliğinde sık görülen problemlerden biridir ve finans, tıp, insan-bilgisayar etkileşimi gibi bir çok alanda karşılaşılır. Geleneksel olarak bu problem tek değişkenli zaman dizisi sınıflandırma metotlarının çokdeğişkenliye uyarlamalarıyla çözülür. Bu çalışmada, çokdeğişkenli dizilerin sınıflandırılması için özel olarak geliştirilmiş öznitelik temelli bir metot sunulmaktadır. Bu metot sadece tek değişkenli müstakil dizilerin kendi özniteliklerinen değil, aynı zamanda değişkenler arası ikili etklişimin bilgisinden de yararlanır. Metot iki temel özniteliğin kaynaşımından oluşur: aralık ortalamaları ve kutupsal histogram yoğunlukları. Etiketli öznitelik vektörleri, yüksek boyutlu değişkenler üzerinde iyi çalışan rassal ormanlar aracılığıyla sınıflandırılır. Metodun literatürde sıklıkla kurllanılan ölçüt veri setleri üzerindeki performansı raporlanmıştır. Buna göre metot tatmin edici ölçülerde hatasız sonuçlar vermektedir ve uygulanabilirlik açısından bakıldığında gerektiği şekilde ölçeklendirilebilmektedir. Ayrıca değişkenler arası ikili etkileşim özniteliklerinin doğruluk oranlarını vakaların çoğunda arttırdığı görülmüştür. Buna ek olarak toplam hesaplama sürelerinin en büyük kısmı yine bu özniteliklerin hesaplanmasına harcanmaktadır. Bütün bunlar ışığında metot, geliştirilmesi gereken pek çok yanı olmasına karşın en modern metotlarla mukayese edilebilir seviyededir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT v
ÖZET vi
LIST OF FIGURES ix
LIST OF TABLES xii
LIST OF SYMBOLS xiii
LIST OF ACRONYMS/ABBREVIATIONS xiv
1. INTRODUCTION 1
2. PROBLEM STATEMENT AND RELATED WORK 6
2.1. Definitions 6
2.2. Problem Statement 7
2.3. Related Work 7
3. THE INTERVAL MEANS AND POLAR HISTOGRAM OF DIFFERENCES METHOD 11
3.1. Background 11
3.1.1. Polar Histogram 11
3.1.2. Random Forest Classification 14
3.2. The Method 14
3.3. A Toy Example of Feature Extraction Phase 20
3.4. The Intuition Behind The Feature Design 24
4. EXPERIMENTS AND RESULTS 31
4.1. Benchmark Datasets 31
4.2. Experimental Setup 33
4.3. Results 34
4.4. Sensitivity Analysis 37
4.5. Computational Complexity 38
5. CONCLUSION 45
5.1. An Overview of the Thesis 45
5.2. Directions for Future Study 45

5.2.1. Handling High-dimensional Multivariate Time Series 46
5.2.2. Different Bin Boundaries and Resolution for PHD Features . . . 46
5.2.3. Visualisation Strategy for Interpretablity 46
5.2.4. Customized Classification Algorithms per Application 47
5.2.5. Additional Features 47
REFERENCES 48

LIST OF FIGURES

Figure 2.1. Flowchart illustrating the weak classification of MTS problem. . . 8

Figure 3.1. An example wind rose plot created with synthetic data. The relative frequencies can easily be recognized with this kind of visualisation. 12

Figure 3.2. Atan2 function with respect to the sign of y and x. 13

Figure 3.3. 8 polar histogram bins which the output of four-quadrant inverse tangent function is sorted out to. 17

Figure 3.4. The pseudocode for Interval Means and Polar Histogram Densities Algorithm. 19

Figure 3.5. The plotted MTS instance with interval cut points shown with dashed black lines. Each coloured line denotes the time-ordered accelerometer readings for a specific axis. 21

Figure 3.6. Polar scatter plot for the whole angular values of X-Y pair. The colour of the point represent the 2-norm of the difference pair where colder colours represent higher 2-norm values. 25

Figure 3.7. Plotted time series for the synthetic example. The synthetic example has four different classes. 26

Figure 3.8. Heatmap for the extracted IM features for the series shown in Figure 3.7. The parameter λ is taken as 4. 27

Figure 3.9. Plotted time series for the multivariate synthetic example without the noise. 28

Figure 3.10. Heatmap for the extracted IM features of the series shown in Figure 3.9. The parameter β is taken as 0. 29

Figure 3.11. Plotted time series for the multivariate toy example with the noise added. 29

Figure 3.12. Heatmap for the extracted IM features of the series shown in Figure 3.11. The parameter β is taken as 0. 30

Figure 3.13. Heatmap for the extracted IM features of the series shown in Figure 3.11. The parameter β is taken as 0.35. 30

Figure 4.1. K-Fold Cross Validation Strategy. 35

Figure 4.2. Error rates comparison between only IM features version and full IM-PHD features version. 38

Figure 4.3. Sensitivity analysis on a 5x5 parameter grid for Character Trajectories dataset. 42

Figure 4.4. Sensitivity analysis on a 5x5 parameter grid for Uwave Gesture Library dataset. 43

Figure 4.5. Median prediction latency per instance with varying number of dimensions. 43

Figure 4.6. Median prediction latency per instance with varying fraction of original series length. 44

Figure 4.7. Median prediction latency per instance with varying fraction of original dataset size. 44

Figure 4.8. Median prediction latency per instance with varying number of intervals. 44

LIST OF TABLES

Table 3.1. A downsampled MTS instance from UWave Gesture Library Dataset. The horizontal bar in the middle denotes the cut points from where the instance is split into four, almost equal length intervals. 22

Table 3.2. Binning process for the X-Y pair when $\beta = 0.15$. 23

Table 3.3. Densities as a result of the binning process outlined in Table 3.2. . 24

Table 4.1. Properties of benchmark datasets used for performance evaluation. 32

Table 4.2. Results with only IM features. Tested on the predefined train-test split. Reported statistics are of 10 replications. 36

Table 4.3. Results with full IM-PHD features. Tested on the predefined train-test split. Reported statistics are of 10 replications. 37

Table 4.4. Results with full IM-PHD features. Tested with fivefold CV. . . . 39

Table 4.5. Results with full IM-PHD features. Tested with tenfold CV. 40

Table 4.6. Comparison of test error results against gRSF, LPS and SMTS. . . 41

LIST OF SYMBOLS

$\mathcal{C}$	The set of 2-combinations of multiple sources
$\mathcal{D}$	Dataset of training multivariate time series objects
$\mathcal{F}$	The set of IM-PHD features obtained from a dataset
$\mathcal{L}$	The set of labels
$\mathbf{M}$	The number of different sources of a multivariate time series object
$\mathbf{N}$	The number of instances in the training set
$\mathbf{T}$	The length of a multivariate time series object
$x_m^n(t)$	An observed variable of the n^{th} instance, from source m, at time point t
β	Radius cut parameter
λ	Number of intervals parameter

LIST OF ACRONYMS/ABBREVIATIONS

CV	Cross Validation
DTW	Dynamic Time Warping
IM	Interval Means
IM-PHD	Interval Means and Polar Histogram Densities
KNN	K Nearest Neighbours
MTS	Multivariate Time Series
PHD	Polar Histogram Densities
RF	Random Forest
SVM	Support Vector Machine
TSDM	Time Series Data Mining
UTS	Univariate Time Series

1. INTRODUCTION

Time is an intrinsic aspect of most of the data collection processes. Almost every data gathering and measuring action has an implicit temporal dimension. Whether that information is made explicit or not is the decision of the collector. Yet in many cases, the temporal dimension brings a rich source of insight to the question that is being answered through the use of those data. It is therefore not a surprise that temporal database applications are ubiquitous in many domains such as finance, medicine, manufacturing, transportation, meteorology, human-computer interaction and science [1]. In the light of the recent developments in the network-connected devices and database management tools, temporal data are gaining importance in many other domains as well (e.g. social media monitoring applications).

An important class of temporal data are time series data [2]. It is the common temporal data model for statistical analysis [3]. A time series represents a regularly spaced, time-ordered measurements of a collection of values [4]. Traditionally, time series are studied mainly for forecasting and retrospective analysis. Recently, the data mining paradigm has spread its influence on time series data that led to what is now called time series data mining (TSDM). The interest for TSDM has grown exponentially in the last two decades. This can be observed through the increasing number of published works [3–9] which provides an elaborate overview of a large body of scientific research and applications made so far in the field.

TSDM consists of many well-defined tasks that involve the discovery of frequent patterns and hidden knowledge in time series databases. An elaborate, but not exhaustive, list of such tasks can be listed as follows:

- Representation to reduce dimensionality
- Similarity measure for whole sequence and/or subsequence matching
- Indexing
- Strong and weak classification

- Clustering
- Anomaly and rare event detection
- Segmentation

This thesis aims to propose and discuss a novel method— the Interval Means and Polar Histogram Densities (IM-PHD) method— to deal with one of these tasks, namely, weak classification of multivariate time series. For ease of reading, the term classification will be used to refer to weak classification since strong classification is completely out of scope of this study.

The definition of the MTS classification problem implies an isolated setting where an individual collection of regularly spaced, multi sourced and time-ordered data object can be semantically associated with one and only one label from a fixed set of labels. A simple example is where an accelerometer collects regular triaxial acceleration data from the gestures of a user with a device (e.g. a smartphone). For further simplification, let us bound the case such that the user draws either one of two kinds of shapes in the air with that device: squares and circles. This setting allows us to label each separate collection of measurements either as circles or squares. The classification problem arises when a new MTS is obtained with an unknown label. A classification method is developed to accurately and swiftly label the incoming measurement. In the data mining and machine learning paradigm, the set of rules to achieve this is learned through previously labelled examples of gestures. This specific example is from a domain of problems known as gesture recognition and has serious applications in the area of human-computer interaction [10]. Problems that are analogous to gesture recognition from an algorithmic point of view, can be found in many areas such as medicine, finance or meteorology.

Before setting the stage for MTS classification, it is worth briefly mentioning the current state-of-art in univariate time-series (UTS) classification. Oftentimes, MTS classification algorithms are inspired by, if not an extension of, their UTS counterparts. UTS classification research is deemed to be rich and well-established [11, 12]. A wide array of algorithms have been developed for this purpose. Xing *et al.* evaluates UTS

classification algorithms in three main categories:

- Feature based classification: This type of algorithms are focused on a feature extraction and selection phase which converts a high-dimensional time series object into a set of features that is applicable to regular classification methods. These features can be of several kind, such as summary statistics of global or local intervals [13] or bag-of-words like features [14]. It is desirable for these features to be interpretable or easy to visualize.
- Distance based classification: The aim here is to define a distance function to quantify the (dis)similarity between a pair of sequences. A well defined distance function is applicable to KNN or SVM classifiers. The simplest and most well-known distance function is probably the Euclidean distance. However, Euclidean distance performs worse as there are more variations in between the members of the same class in terms of phase differences and distortions. Also, Euclidean distance is inapplicable to time series of different lengths. This led to the rise of DTW [15], a highly popular choice to overcome previously mentioned problems.
- Model based classification: The idea of representing a time series object with a generative model leads to the model-based classification methods. Arguably the most popular choice is the Hidden Markov Model which also proved to be a successful choice especially for problems in the domain of speech recognition. The logic behind such algorithms is to match an unlabelled instance with the class of the model that produces the highest likelihood with that model.

In principle, most of the univariate time-series (UTS) classification algorithms can be generalized to MTS classification as long as the other assumptions hold and other limitations permit. This is simply done by vectorizing the MTS by concatenating its individual source series or by deploying a voting mechanism based on different sources. However, this approach is prone to fail. The principal reason is that an MTS is more than the sum of its individual time-series such that the interaction among different attributes of an MTS often plays an important role [16]. The IM-PHD method is a feature based method that is specifically designed for MTS classification and aims to account for this issue. As the name implies, there are actually two separate types of

features fused into one single feature vector. Interval Means (IM) type features aim to capture time ordered level information while Polar Histogram Densities (PHD) type features intend to summarise the nature of 2-way interactions between the dimensions of the MTS. Random forests are chosen as the algorithm to conclude the process while the feature vector is also applicable to other well-known supervised learners such as support vector machines or neural networks.

IM-PHD method is designed with many important specifications kept in mind that determine the quality of an MTS classification method. The predictive power of that method is defined by how well the predictive model is able to accurately label the unlabelled objects. While predictive power is arguably the most important aspect, the ideal algorithm should also be computationally inexpensive. It should work with minimal memory and processing resources so that it is applicable also in practice. The algorithms that provide the best predictive performance in small datasets may not be scalable for larger datasets which makes them obsolete for real-life purposes. Given that exploding amounts of data is one of the defining characteristics of emerging industries, computational tractability becomes even more important as time progresses.

For the case of time series classification, handling time series objects with different lengths is also an important issue. Especially real time series databases consists of a variety of time series object. A simple example would be the speech recognition databases where each speaker might produce vocal sequences of different lengths that corresponds to the same word or letter (i.e. the same class, in terms of problem jargon). This problem may be overcome by upsampling or downsampling methods. However, such preprocessing steps might lead to information loss or distortion of the information at hand, in addition to the added computational complexity.

Handling missing values— which naturally arises in real datasets— is also an important quality of an algorithm. An algorithm without this feature may face losing a certain amount of data and the information contained within that. MTS data may contain categorical values as well as numerical ones, depending on the source. Such mixed type MTS can only be classified by algorithms that are able to handle both

categorical and numerical values.

IM-PHD is a satisfyingly accurate and tractable algorithm which can handle missing values as well as series with different lengths. The results are reported on various benchmark datasets and reported. However, the current version cannot handle categorical variables. Overall, the method positions itself as an out-of-box, general purpose, scalable method for MTS classification problems.

The following parts of the thesis is organised as follows. Common notation is defined, the problem is formally stated and relevant literature is discussed in Chapter 2. Chapter 3 introduces the reader to the details of the feature extraction phase. Chapter 4 provides the results of the experiments to asses the performance of the method. Chapter 5 lays the ground for future research and concludes the thesis.

2. PROBLEM STATEMENT AND RELATED WORK

2.1. Definitions

Chapter 2 is mainly dedicated to formally state the problem for which the new method is developed for and to extensively discuss the current state of the research in this field. In order for this to be done in an easily communicable way, some terms and definitions must be given.

Definition 2.1.1. An observed variable, $x_m^n(t)$, is almost the same as how it is defined in statistics, except for the additional information about the time point it was observed at. This additional information —denoted by the subscript— allows distinguishing the variables that are of the same source but observed at different points in time. Accordingly, m denotes the source, t denotes the time point and n denotes the instance. When there is only one source considered, the notation reduces to x_t^n for convenience. When there are multiple sources considered, the notation reduces to X_t^n where X is a vector of M variables, $X = [x_1, x_2, \ldots, x_{M-1}, x_M]'$.

Definition 2.1.2. A univariate time series (UTS) instance, $x^n = (x_1^n, x_2^n, \ldots, x_{T-1}^n, x_T^n)$, is an instance of a sequence of T observations from a single source, observed at T different time points.

Definition 2.1.3. A time series interval, $x[a, b] = (x_a, x_{a+1} \ldots, x_{b-1}, x_b)$ is a subsection of a UTS x from start point a to end point b, where the observed variables are contiguous.

Definition 2.1.4. A multivariate time series (MTS), $X^n = (X_1^n, X_2^n, \ldots, X_{T-1}^n, X_T^n)$, is an instance of a sequence of T observations from multiple sources, observed at T different time points. The vector of observations that is denoted by the same temporal subscript is assumed to be observed at the same point in time.

Definition 2.1.5. A multivariate time series dataset, $\mathcal{D} = \{X^1, X^2, \ldots, X^{N-1}, X^N\}$ is a set of N MTS instances. An MTS object may be associated with a label y^n, for $n = (1, 2, \ldots, N-1, N)$ and $l^n \in \mathcal{L} = \{1, 2, \ldots, L-1, L\}$ where L is the number of distinct classes defined for that dataset.

2.2. Problem Statement

The classification of MTS problem can simply be defined as developing a function or a set of rules that maps a given unlabelled MTS instance to one of the class labels from a predefined set.

$$g : \mathcal{D} \rightarrow \mathcal{L}$$

In the machine learning and data mining paradigm, classification is considered to be an instance of a broader class of tasks called supervised learning [17]. Under that paradigm, the function g is learned through the guidance of a set of labelled observations. For the special case of MTS classification, a set of labelled MTS objects are used to learn the function g. A generic process is summarised in the flowchart illustrated in Figure 2.1, for the special case of feature based methods. The process can be described in two parts. The red route shows the learning phase while the green route denotes the predicting phase. Labelled MTS objects are reduced to feature vectors and patterns are learned in the learning phase. The labels of the incoming objects are predicted through their feature vectors and previously trained predictive model.

2.3. Related Work

In one of the earliest work on extending the classification problems to MTS, Kadous attempts to provide a general system for classification of MTS [18]. He conceptualises metafeatures for that end which are basically user defined metrics that are deemed to possess a discriminatory power for that particular problem. The fact that the metafeatures must be defined by the user hinders the ease of use of the method as an "out-of-box" method. Also, these features do not necessarily represent the interaction among different attributes. Geurts and Wehenkel employ ideas from image processing to be used in MTS classification [19]. The advantage of that algorithm is that it is non-parametric.

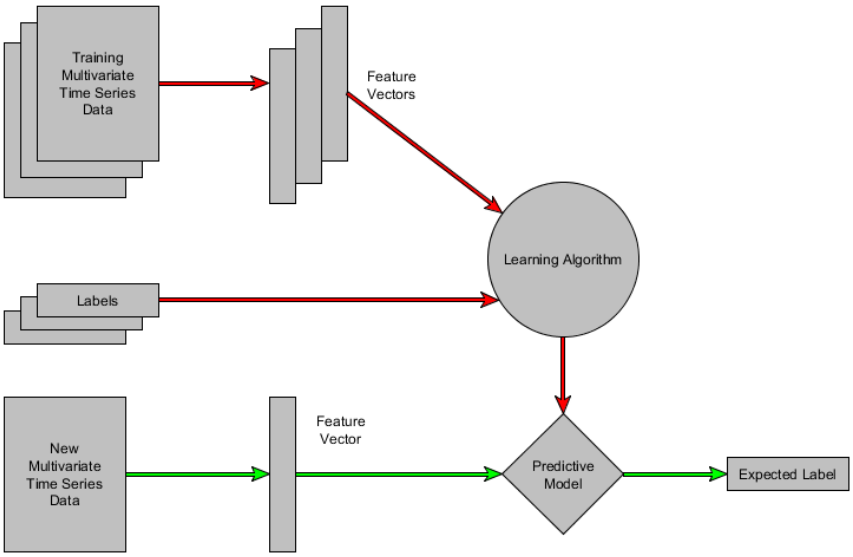


Figure 2.1. Flowchart illustrating the weak classification of MTS problem.

DTW, a popular choice for data mining tasks in UTS, is considered by some researchers for MTS as well. Chaovalitwongse and Palados develop an SVM classifier with a DTW based kernel [20]. Akl and Valaee combine DTW with affinity propagation [21]. Both papers demonstrate the performance of the proposed methods on only a single dataset. This casts doubt on the overall effectiveness of the approaches. Orsenigo and Vercellis propose a two-phase algorithm utilizing warping diştances and discrete SVM experimented on several benchmark datasets [22]. The major drawback of DTW based methods is that they are not able capture the relationship among the attributes of MTS.

Li *et al.* addresses the problem by handling an MTS object as a matrix where columns denote different attributes and rows denote time-ordered observations [23]. Such a matrix is reduced to eigenvectors by using singular value decomposition (SVD) which are then processed by an SVM classifier. Weng and Shen builds their method on Li's approach by incorporating locality preserving projections [24]. The method allows for carrying out experiments on databases with MTS objects of different lengths. SVD is also considered by Spiegel *et al.* in the segmentation phase of the method [25]. According the that paper, SVD exposes the correlation structure among the attributes,

which is helpful for effective segmentation. The second phase involves agglomerative hierarchical clustering for grouping the segmented series.

Motivated by tornado prediction problem from the domain of meteorology, McGovern *et al.* proposes a prediction algorithm based on multi-attribute temporal motif discovery [26]. The main drawback of this approach is that it can only be utilized for binary classification problems.

Weng and Shen's work is one of the earliest work where benchmark datasets are used for experimentation [24]. As outlined by Keogh and Kasetty, data bias is an important issue in comparing methods in the literature and therefore the use of benchmark datasets are important [9]. A good approach should be able to perform well on a heterogeneous mix of datasets. The UCR Time Series Classification Archive provides a comprehensive, centralised repository for UTS classification. Such a centralised repository is currently absent for MTS classification. However as of today, a wide variety of high quality datasets are available for the use of the scientific community. It is observed that in the recent papers in the field of MTS classification, these benchmark datasets are more widely adopted.

Bag-of-words (BoW) model is a widely used representation technique in the domain of natural language processing. Inspired by the applications in that domain, the MTS classification problem can be solved by BoW approaches. Ordóñez *et al.* proposes two variations of a BoW based approach where a preprocessing step is required to convert individual numerical series into symbolic ones [27]. The famous Symbolic Aggregate Approximation (SAX) method [28] is used for the preprocessing step. The SAX method is data dependent but it ignores the information contained in the class labels. Baydogan and Runger employs a more sophisticated BoW based approach, conceptualising a codebook achieved through a tree-based supervised learner [16]. The terminal nodes of the forest expose attribute regions where class distributions are the most homogeneous. The symbol frequencies are then used as feature vectors for another supervised classifier.

Time series shapelets have recently been introduced as a new structure for TSDM tasks including classification [29]. Ultra fast shapelets (UFS) [30] and generalized random shapelet forests (gRSF) [31] are two approaches that exploits shapelets to classify multivariate time series. While shapelets are highly interpretable and powerful approaches for assessing similarity between time series objects, they often require extremely long training time. These two approaches strive to overcome this by careful selection of shapelet candidates. According to the reported results, gRSF clearly outperforms UFS both in terms of accuracy and training time.

Learned pattern similarity (LPS) is a UTS similarity measure that can be easily extended to the multivariate case [32]. LPS exploits local autocorrelation structures in time series data with the help of an ensemble of regression trees. KNN with 1 neighbour (i.e. 1-NN) is used combined with the similarity measure to conclude the classification procedure.

During the comparison of the proposed method in this thesis with other methods; SMTS, LPS and gSRF are considered to be state-of-the-art approaches as they are the best among other in terms of self-reported results and that their implementation is publicly available.

3. THE INTERVAL MEANS AND POLAR HISTOGRAM OF DIFFERENCES METHOD

In this chapter the reader will be introduced to the proposed MTS classification method called Interval Means and Polar Histogram Densities (IM-PHD). The method consists of two phases: the feature extraction phase and the classification phase. For the former, polar histograms are used while for the latter, random forests are utilized. Accordingly, the chapter will begin with a background section where these two concepts are reviewed for the reader. The formal, detailed description of the method will be given in the later section. The chapter will be concluded with a toy example for further clarification of the calculations and discussions on some synthetic time series for justification of the design.

3.1. Background

3.1.1. Polar Histogram

Consider a sample of wind direction observations taken within a certain time period. Such a variable is naturally represented with an angular measure such as degrees or radians. How far away the wind has originated from the observation point is not deemed to be important. A wind rose plot is a perfect choice for visualising the distribution of the sample. An example wind rose plot is shown in Figure 3.1. Each polar bin represents a frequency.

Next, consider the navigation of a flock of birds. The measurement of the coordinates for the location of the flock contains information about the motion. If the observations are made regularly in time, the first difference of the resulting time series yields information about the latitudinal and longitudinal pace of the flock. However, a biologist might be interested in the direction of the navigation rather than the pace. In that case, the scientist converts coordinates from Cartesian to polar. Assume that

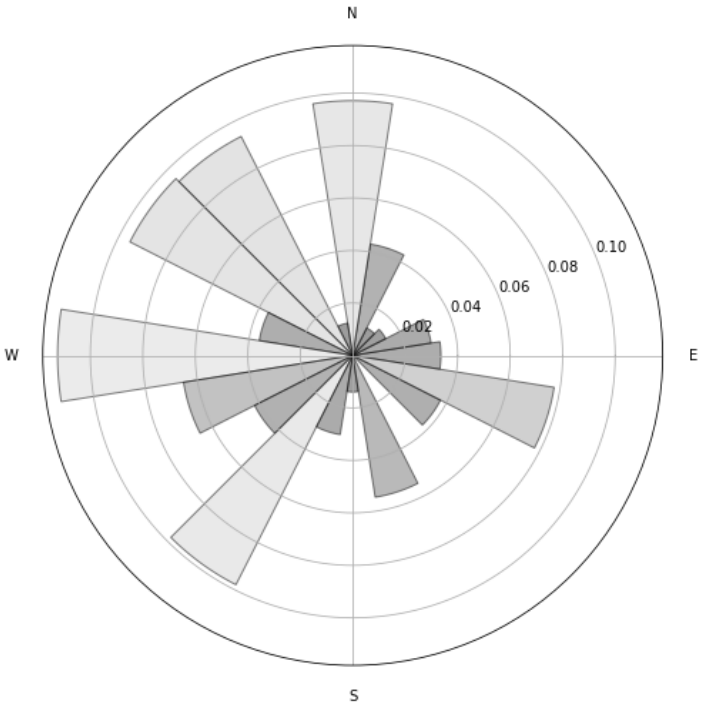


Figure 3.1. An example wind rose plot created with synthetic data. The relative frequencies can easily be recognized with this kind of visualisation.

x is the longitude and y is the latitude. The direction φ can be found with the `atan2` function. This function is also known as four-quadrant inverse tangent.

$$\varphi = \text{atan2}(y, x) \tag{3.1}$$

The reason why `atan2` is used instead of the classical arctangent function is that the `atan2` function can handle situations where the input is undefined for arctangent and `atan2` utilizes the sign information fully. The resulting angle of this function lies in the $[-\pi, \pi]$ range, in terms of radians.

Lastly, consider any two dimensional MTS. Given that both dimensions are individually normalized and brought to a common scale, the first difference of this series

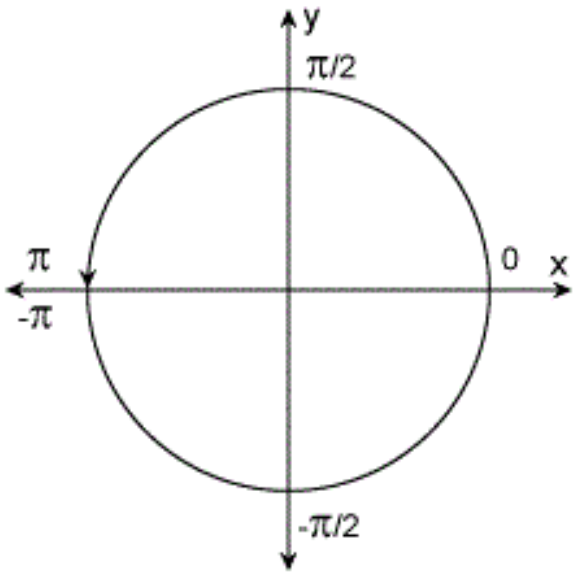


Figure 3.2. Atan2 function with respect to the sign of y and x.

would yield a sample of input ready to be processed with the `atan2` function. The resulting angular data can be represented in what is called a polar diagram, where the data are sorted out into equally spaced bins in the $[-\pi, \pi]$ range. However, π and $-\pi$ are not necessarily the endpoints of the bins as the scale is not ordinal because it can be wrapped around. The relative frequencies will expose information about the relationship between the two attributes, in terms of co-movement. Such information can easily be visualised by a polar histogram plot similar to Figure 3.1. Moreover, the relative frequencies can be used as a vectorized, numerical source of information. For the proposed method in this work, 2-combinations of the attribute set of the difference series of a normalized MTS are considered as the input for the `atan2` function. From each combination, a vector of relative frequencies are taken and then concatenated to form the second part of the full feature vector. This part includes valuable information about the relationship between the attributes, which will later be an important input for the random forest classifier.

3.1.2. Random Forest Classification

Random forests (RFs) have been one of the most popular choice for classification tasks since their introduction in 2001 [33]. In line with the notation in [34], an RF classifier is an ensemble of B decision trees $\{T_b, b = 1, 2, \ldots, B-1, B\}$, each grown over a bootstrap sample taken from N observations and in each tree grown, m variables are considered at random from among p possible variables. The majority vote is taken as the output of the algorithm for each predicted instance. This nature of RFs makes them especially useful for multi-class, high dimensional problems. IM-PHD, while in principle reduces the dimension of an MTS object, still produces high dimensional vectors. In addition, many classification problems that IM-PHD was developed for involves a large set of class labels.

A second reason why RFs were chosen as the to-go algorithm for classification is the option to use out-of-bag sample for parameter optimization. Out-of-bag (OOB) error can replace cross validation [34], eliminating the need for sacrificing valuable training data. In many application areas where MTS classification problem arises, data collection is usually expensive. Eliminating the need for cross validation proves to be even more important in such an experimental setting.

3.2. The Method

The IM-PHD method requires a training set of N standardized MTS instances $\mathcal{D} = \{X^n, i : 1, 2, \ldots, N\}$ with their labels $\mathcal{L} = \{l^n, i : 1, 2, \ldots, N\}$ and two user defined parameters: number of intervals λ and radius threshold β. The method has two main phases: the feature extraction phase and the classification phase. The former takes each MTS in the training set, fuses interval-wise level information and co-movement information into a single feature vector. The latter takes the output of the former as a set of labelled feature vectors and fits an RF classifier. The reader is requested to refer to Algorithm 15.1 in [34] for details of the RF classifier training phase. This section is mostly focused on the details of the feature extraction phase.

Let an MTS instance X be represented as a $T \times M$ matrix where T denotes the length of the standardized series and M denotes the number of attributes.

$$X = \begin{bmatrix} x_{11} & \dots & x_{1M} \\ \vdots & \ddots & \vdots \\ x_{T1} & \dots & x_{TM} \end{bmatrix} \tag{3.2}$$

For any given attribute indexed with m, a UTS can be defined.

$$x_m = \begin{bmatrix} x_m(1) & \dots & x_m(T) \end{bmatrix}' \tag{3.3}$$

Since the MTS is standardized, the mean and standard deviation of the values are fixed to 0 and 1 respectively.

$$\bar{x}_m = 0 \tag{3.4}$$

$$\sigma_{x_m} = 1 \tag{3.5}$$

The first part of the feature extraction phase splits each UTS associated with each attribute into λ number of intervals of same length. However, this is not always possible (e.g. when T is not divisible by λ), therefore the algorithm tries make the lengths as balanced as possible. Then, the mean of all the intervals are obtained and concatenated into a vector. Let $\mathcal{R} = \{r_i, i : 1, 2, \dots, \lambda - 1\}$ be the set of points that effectively splits the whole range of time points $1 \dots T$ into λ quasi-equal length intervals. Then the interval means feature vector (IM) for a given MTS object X is defined as follows:

$$IM = \begin{bmatrix} \bar{x}_1[1, r_1] & \dots & \bar{x}_1[r_{\lambda-2}, r_{\lambda-1}] & \dots & \bar{x}_M[1, r_1] & \dots & \bar{x}_M[r_{\lambda-1}, r_\lambda] \end{bmatrix}' \tag{3.6}$$

The second part of the feature extraction phase begins with a simple preprocessing step. Let Q be the first difference series of X.

$$Q = \begin{bmatrix} x_{21} - x_{11} & \dots & x_{2M} - x_{1M} \\ \vdots & \ddots & \vdots \\ x_{T1} - x_{(T-1)1} & \dots & x_{TM} - x_{(T-1)M} \end{bmatrix} = \begin{bmatrix} q_{11} & \dots & q_{1M} \\ \vdots & \ddots & \vdots \\ q_{(T-1)1} & \dots & q_{(T-1)M} \end{bmatrix} \tag{3.7}$$

Let $\mathcal{C} = \{(1,2),(1,3),\dots,(M-1,M)\}$ be the set containing 2-tuples of all 2-combinations of the set of attributes $\mathcal{M} = \{1,\dots,M\}$. For each element of $\mathcal{C}$, a $(T-1) \times 2$ sub-matrix $Q_{(m_1,m_2)}$ can be obtained from Q.

$$Q_{(m_1,m_2)} = \begin{bmatrix} q_{1m_1} & q_{1m_2} \\ \vdots & \vdots \\ q_{(T-1)m_1} & q_{(T-1)m_2} \end{bmatrix} \tag{3.8}$$

Finally, let $V_{(m_1,m_2)}$ be the row-wise implementation of the `atan2` function on $Q_{(m_1,m_2)}$.

$$V_{(m_1,m_2)} = \begin{bmatrix} \texttt{atan2}(q_{1m_1}, q_{1m_2}) \\ \vdots \\ \texttt{atan2}(q_{(T-1)m_1}, q_{(T-1)m_2}) \end{bmatrix} \tag{3.9}$$

Note that each element of $V_{(m_1,m_2)}$ lies in the in the $[-\pi,\pi]$ range. These elements will be sorted out into 8 equal-width of bins in the $[-\pi,\pi]$ range, where bin edges are chosen as $(\frac{\pi}{8}, \frac{3\pi}{8}, \frac{5\pi}{8}, \frac{7\pi}{8}, \frac{-7\pi}{8}, \frac{-5\pi}{8}, \frac{-3\pi}{8}, \frac{-\pi}{8})$. For a visual representation of these bins, please refer to Figure 3.3. The frequencies at each possible angular range will be denoted as $P = \{p_i, i : 1,\dots,8\}$. Before the binning process, the second parameter —radius threshold β, comes in. The radius cut parameter tries to filter out the noise in the co-movement information by discarding (q_{tm_1}, q_{tm_2}) pairs which have radii less then β when they are projected to polar coordinates. The reader is directed to Section 3.4

for further details on why this strategy is chosen. The radius counterpart is trivially found by the 2-norm of the vector. Thus, a paired difference point is considered in the binning process only if it satisfies the following condition:

$$\|(q_{tm_1}, q_{tm_2})\| > \beta \tag{3.10}$$

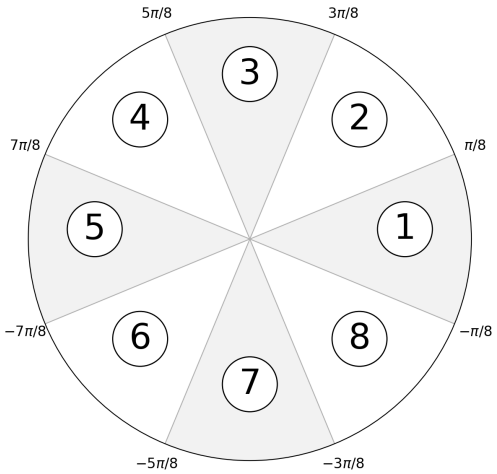


Figure 3.3. 8 polar histogram bins which the output of four-quadrant inverse tangent function is sorted out to.

Note that the least possible meaningful value for β is 0. If the 2-norm of a paired difference is 0, which implies that both of the values are equal to 0, the inverse tangent function is undefined. Such a threshold would automatically solve that problem.

The final step for is to obtain the relative frequency vector $PHD_{(m_1,m_2)}$ by dividing each element by the sum of the absolute frequency vector.

$$PHD_{(m_1,m_2)} = \begin{bmatrix} p_1/\sum p_i \\ \vdots \\ p_8/\sum p_i \end{bmatrix} \tag{3.11}$$

The bin range choice illustrated in Figure 3.3 is not arbitrary. Evaluating Figure 3.3 and Figure 3.2 together, the semantic behind the bin choice is revealed as follows:

- Bin 1: X increases significantly, insignificant change in Y
- Bin 2: Both X and Y increases significantly
- Bin 3: Y increases significantly, insignificant change in X
- Bin 4: Y increases significantly, X decreases significantly
- Bin 5: X decreases signigicantly, insignificant change in Y
- Bin 6: Both X and Y decreases significantly
- Bin 7: Y decreases significantly, insignificant change in X
- Bin 8: X increases significantly, Y decreases significantly

A complete feature vector $IM - PHD$ would be the concatenation of IM and all the $PHD_{(m_1,m_2)}$ associated with each combination. It is easy to infer that for each X, a $IM - PHD$ vector is of length $M(\lambda + 4M(M-1))$, if an exhaustive list of all combinations are desired. An important option considered for the algorithm is that the user is allowed to manually define the set of 2-combinations, either to discard the ones that are deemed to be irrelevant to the problem or to investigate the effect of the ones that are deemed to be important for the problem. In any case, the feature is independent of the time series length T, which makes the method applicable to MTS databases containing time series of different lengths.

$$IM - PHD = \begin{bmatrix} IM & PHD_{(1,2)} & PHD_{(1,3)} & \dots & PHD_{(M-1,M)} \end{bmatrix} \tag{3.12}$$

If the feature extraction process is mapped onto the whole training set $\mathcal{Z}$, a set of feature vectors $\mathcal{F}$ is obtained. Consequently, this set becomes exploitable by classical supervised learners, including the RF classifier. A concise outline of the method can be found in Figure 3.4.

$$\mathcal{F} = \{IM - PHD^n, n : 1, 2, \ldots, N\} \tag{3.13}$$

Input: $\mathcal{D} = \{X^n, n : 1, 2, \ldots, N\}$: a training set of standardized MTS objects, $\mathcal{L} = \{l^n, i : 1, 2, \ldots, N\}$: a label set, β: radius threshold, λ: number of intervals, $\mathcal{C}$: a set of 2-combinations (optional).

Output: RF: A trained random forest classifier

for $n = 1$ to N **do**

Obtain IM^n from X^n as defined in Eq. 3.6

Obtain the difference series Q^n as defined in Eq. 3.7

if $\mathcal{C}$ is passed by the user **then**

Obtain PHD^n from Q^n using all 2-combinations in $\mathcal{C}$ as defined in Eq. 3.8,3.9 and 3.11

else

Obtain PHD^n from Q^n using all possible 2-combinations of $\{1, 2, \ldots, M\}$, as defined in Eq. 3.8,3.9 and 3.11

end if

Concatenate IM^n and PHD^n to obtain $IM - PHD^n$ and add it to features set $\mathcal{F}$

end for

Train a random forest classifier RF using $\mathcal{F}$ as training features and $\mathcal{L}$ as training labels

return RF

Figure 3.4. The pseudocode for Interval Means and Polar Histogram Densities Algorithm.

3.3. A Toy Example of Feature Extraction Phase

As seen in the previous section, the algorithm is fairly simple to understand and does not utilize sophisticated computations to extract features. Even so, it is beneficial for the reader to be exposed to a demonstration. For this purpose, an MTS instance from one of the datasets available in the UCR Time Series Classification Archive [35], the UWave Gesture Library, is obtained. The UWave Gesture Library consists of recordings of a set of gestures with a three-axis accelerometer. An instance is taken from the gesture class which denotes a counter-clockwise circular move. The original length of the MTS is 315 so it was downsampled to a length of 45 for easier demonstration. Like all the other datasets in the UCR Archive, the instances in the UWave Gesture Library is already standardized, therefore that step is skipped in this demonstration.

The next step is to extract interval features. The number of intervals λ is taken as 4. The MTS is split into balanced length intervals as shown in Figure 3.5. The cut points are 11, 22 and 33. For each axis reading, the mean is computed for each four interval. Table 3.1 provides interval means for each axis at the bottom of each block. These summary statistics are then concatenated to form the IM part of the feature vector of length 12 IM =(-0.47, -0.82, 0.70, -0.24, -0.20, -1.51, 1.11, 1.15, 0.47, -0.47, -0.14, 0.29).

Before proceeding with the PHD part of the feature, numerical first difference is calculated for the MTS instance. Then `atan2` function is applied separately to each 2-combinations where the paired difference 2-norm is greater than $\beta = 0.15$. Radial cut-off lines shows in Figure 3.6 show how some of the difference pairs are discarded with respect to the value of β. Table 3.2 demonstrates the binning process only for the X-Y pair. Observe how some of the paired differences are discarded based on their 2-norm value and how the rest is sorted out to 8 different bins.

After the sorting process is done, pair is normalized with the total count of occurrences. The density columns in Table 3.3 are then concatenated to form the PHD

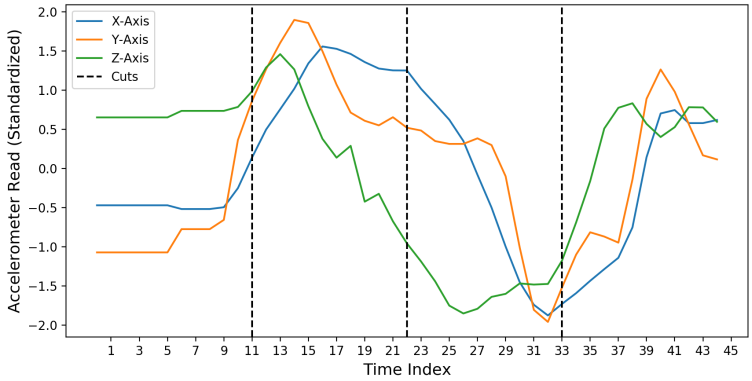


Figure 3.5. The plotted MTS instance with interval cut points shown with dashed black lines. Each coloured line denotes the time-ordered accelerometer readings for a specific axis.

part of the feature. Lastly, IM and PHD features are concatenated into a single vector of length 36 and the feature extraction phase is finished.

Table 3.1. A downsampled MTS instance from UWave Gesture Library Dataset. The horizontal bar in the middle denotes the cut points from where the instance is split into four, almost equal length intervals.

Index	**X-Axis**	**Y-Axis**	**Z-Axis**	**Index**	**X-Axis**	**Y-Axis**	**Z-Axis**
1	-0.48	-1.08	0.66	23	1.24	0.52	-0.96
2	-0.48	-1.08	0.66	24	1.00	0.48	-1.19
3	-0.48	-1.08	0.66	25	0.81	0.35	-1.44
4	-0.48	-1.08	0.66	26	0.61	0.31	-1.75
5	-0.48	-1.08	0.66	27	0.34	0.31	-1.85
6	-0.48	-1.08	0.66	28	-0.09	0.38	-1.79
7	-0.53	-0.78	0.74	29	-0.51	0.30	-1.64
8	-0.53	-0.78	0.74	30	-1.01	-0.10	-1.6
9	-0.53	-0.78	0.74	31	-1.46	-1.02	-1.47
10	-0.5	-0.66	0.74	32	-1.75	-1.82	-1.48
11	-0.26	0.36	0.79	33	-1.88	-1.97	-1.48
Mean	-0.47	-0.82	0.70	Mean	-0.24	-0.20	-1.51
12	0.13	0.86	0.99	34	-1.74	-1.53	-1.18
13	0.48	1.27	1.29	35	-1.60	-1.11	-0.69
14	0.74	1.61	1.47	36	-1.44	-0.82	-0.16
15	1.00	1.9	1.27	37	-1.29	-0.88	0.51
16	1.33	1.86	0.8	38	-1.15	-0.96	0.78
17	1.54	1.5	0.38	39	-0.76	-0.15	0.84
18	1.51	1.07	0.14	40	0.13	0.89	0.57
19	1.45	0.71	0.29	41	0.69	1.27	0.4
20	1.34	0.61	-0.42	42	0.73	0.98	0.53
21	1.26	0.55	-0.32	43	0.57	0.56	0.79
22	1.24	0.65	-0.67	44	0.57	0.16	0.78
				45	0.61	0.11	0.6
Mean	1.11	1.15	0.47	Mean	-0.47	-0.14	0.29

Table 3.2. Binning process for the X-Y pair when $\beta = 0.15$.

Index	**X-Diff**	**Y-Diff**	**2-Norm**	**Bin**	**Index**	**X-Diff**	**Y-Diff**	**2-Norm**	**Bin**	**Index**	**X-Diff**	**Y-Diff**	**2-Norm**	**Bin**
1	0.00	0.00	0.00	-	16	0.22	-0.36	0.42	8	31	-0.29	-0.80	0.85	7
2	0.00	0.00	0.00	-	17	-0.03	-0.43	0.43	7	32	-0.14	-0.15	0.21	6
3	0.00	0.00	0.00	-	18	-0.06	-0.36	0.36	7	33	0.15	0.44	0.46	3
4	0.00	0.00	0.00	-	19	-0.10	-0.10	0.15	-	34	0.14	0.42	0.44	3
5	0.00	0.00	0.00	-	20	-0.08	-0.06	0.10	-	35	0.16	0.29	0.33	2
6	-0.04	0.29	0.30	3	21	-0.02	0.10	0.11	-	36	0.15	-0.05	0.16	1
7	0.00	0.00	0.00	-	22	0.00	-0.13	0.13	-	37	0.14	-0.08	0.16	8
8	0.00	0.00	0.00	-	23	-0.23	-0.04	0.24	5	38	0.39	0.81	0.89	2
9	0.02	0.11	0.12	-	24	-0.19	-0.13	0.24	6	39	0.90	1.03	1.37	2
10	0.24	1.02	1.04	3	25	-0.20	-0.04	0.20	5	40	0.56	0.37	0.67	2
11	0.38	0.49	0.63	2	26	-0.27	0.00	0.27	5	41	0.04	-0.28	0.29	7
12	0.35	0.40	0.54	2	27	-0.43	0.07	0.44	5	42	-0.17	-0.42	0.45	7
13	0.26	0.33	0.42	2	28	-0.42	-0.09	0.43	5	43	0.00	-0.40	0.40	7
14	0.26	0.29	0.39	2	29	-0.50	-0.40	0.64	6	44	0.04	-0.05	0.07	-
15	0.32	-0.39	0.33	1	30	-0.45	-0.91	1.02	6					

Table 3.3. Densities as a result of the binning process outlined in Table 3.2.

	X-Y Pair		**Y-Z Pair**		**X-Z Pair**	
Bin	**Count**	**Density**	**Count**	**Density**	**Count**	**Density**
1	2	0.06	4	0.13	5	0.15
2	8	0.26	5	0.16	5	0.15
3	4	0.13	3	0.1	2	0.06
4	0	0	2	0.06	4	0.12
5	5	0.16	6	0.19	5	0.15
6	4	0.13	3	0.1	4	0.12
7	6	0.19	5	0.16	6	0.18
8	2	0.06	3	0.1	2	0.06

3.4. The Intuition Behind The Feature Design

What kind of information can discriminate one class from another in a time series dataset? Even for the univariate case, the number of possibilities approach infinity. To further simplify the discussion, let us assume that a numerical time series is composed of some arbitrary number and types of Gaussian pulse superimposed on a constant series of value 0. A pattern can be extracted for each class from the following possible properties:

- How many pulses are there? A class may be characterized by 3 pulses while for another class, it typically happens once.
- Does a pulse happen at a certain interval? For example, for a certain class, one and only one pulse happens in the first half of the time horizon while for another class it happens in the other half.
- Is the pulse negative or positive?
- Is it a sharp or a blunt pulse?

The list is obviously not exhaustive. Even if it was, the combinations of these properties would still yield infinitely many different higher-order patterns. An instance

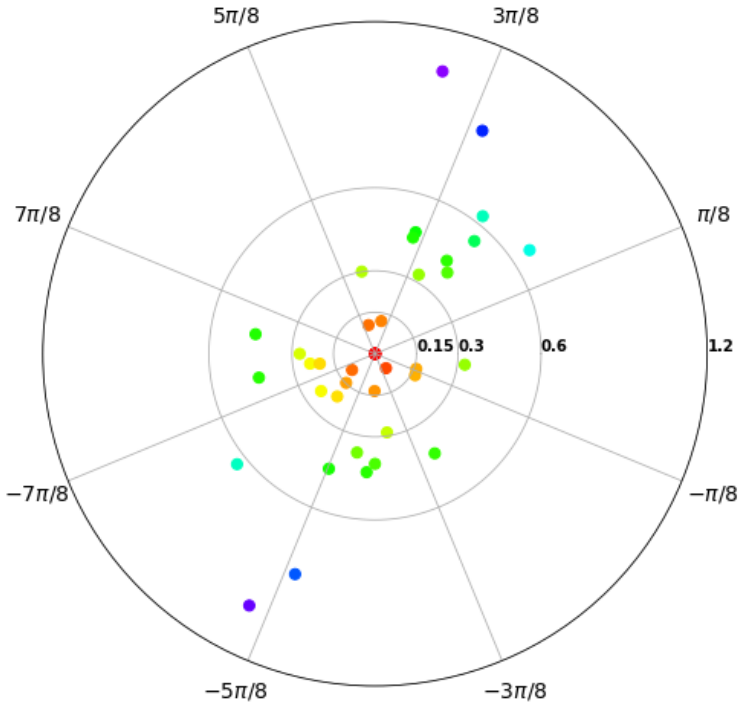
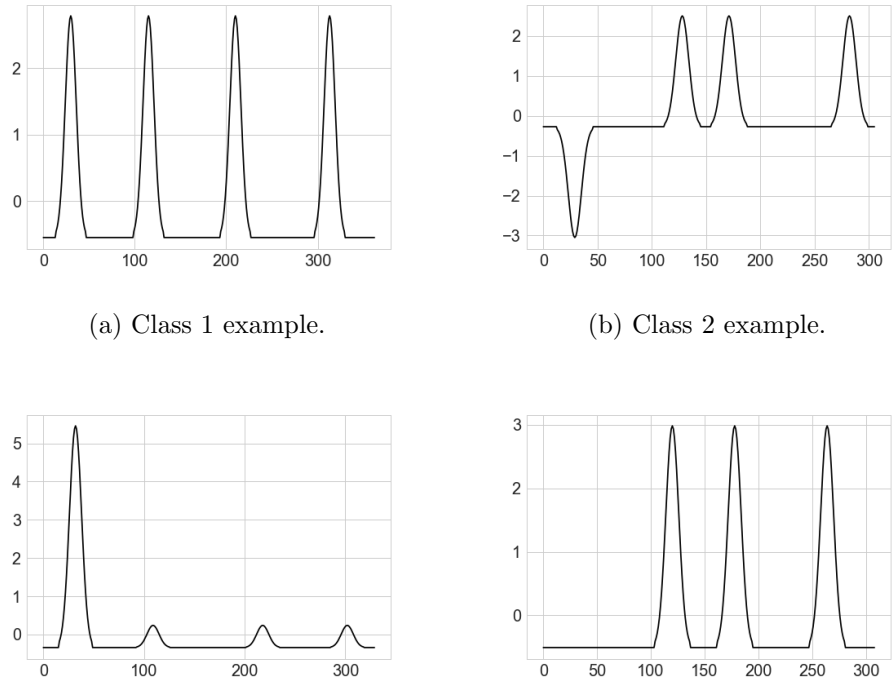


Figure 3.6. Polar scatter plot for the whole angular values of X-Y pair. The colour of the point represent the 2-norm of the difference pair where colder colours represent higher 2-norm values.

would be a class characterized by a sharp positive pulse in the first quarter and a blunt negative pulse in the third quarter of the whole time horizon. Given the complexity of such an extremely simplified case, a "one algorithm fits all" type method would be overly ambitious. The intuition behind the IM-PHD algorithm is can be expressed in two stages. At the first stage, the IM features captures information with respect to the level of the series and the temporal information attached to it. An example of a synthetic dataset based on the aforementioned pulse behaviours is presented at this point. Figure 3.7 shows examples from four theoretical classes. Each class has three pulses in the last three quarters of the full horizon. The first quarter is the decisive part. One of the classes has a blunt positive pulse, one has a blunt negative pulse, one has a sharp positive pulse and the last does not have any pulse. Note that the series

are all of different lengths and standardized. The standardization is the reason why the last three pulses were dwarfed with respect to the first one in the third class.



(a) Class 1 example.

(b) Class 2 example.

(c) Class 3 example.

(d) Class 4 example.

Figure 3.7. Plotted time series for the synthetic example. The synthetic example has four different classes.

The IM properties with $\lambda = 4$ is shown in Figure 3.8 in heatmap style. It is clear that it is fairly easy for any classifier algorithm to distinguish these classes. While this example may be overlooked as ridiculously simple, it will be shown in Section 4.3 that IM features alone provides a strong basis for classification. Especially when there is a satisfyingly large number of training examples, it is easy for the algorithm to match similar cases from the same class. IM features are able to partly overcome the weakness that Euclidean distance is exposed to: distortion in time axis. In addition, it allows handling of time series of different length and also missing values by ignoring them. Visualising is convenient and commenting on the features is simple. Lastly, for the multivariate case, the combined predictive performance of the features of different dimensions is a bonus.

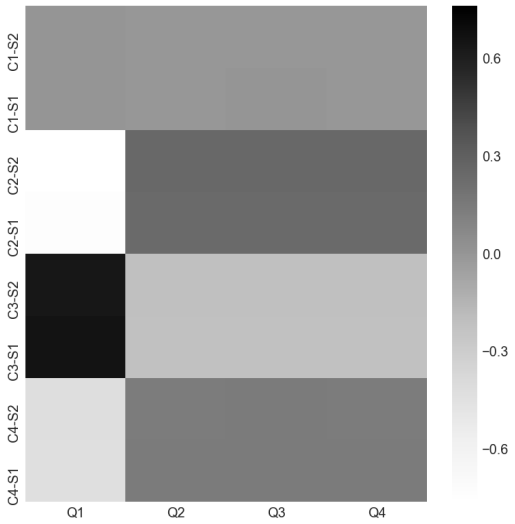
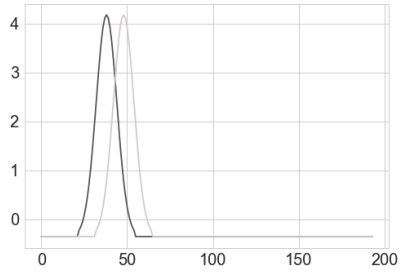


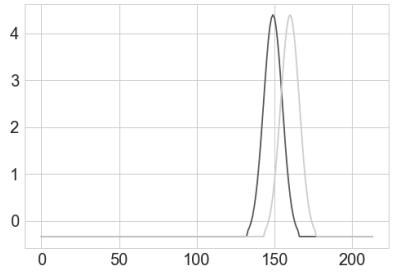
Figure 3.8. Heatmap for the extracted IM features for the series shown in Figure 3.7. The parameter λ is taken as 4.

Now consider a new example where time series are 2-dimensional. For simplicity, each attribute is characterized by a single pulse. Assume that in either one of the attributes, the peak of the pulse in one dimension triggers the start of the pulse in the other dimension. What distinguishes one class from another is whether the first attribute has triggered the second or vice versa. An example of such a scenario is materialized in Figure 3.9. One can easily infer that IM features alone cannot solve this problem. It would mistake Class 1 samples of each class as the same, just as it would do for the Class 2 samples. As mentioned before, PHD features aim to resolve such issues using the relationship between attributes. Figure 3.10 shows the features extracted for the whole 8 samples. It is trivial to see that any proper classifier would be able to solve this case using either one of the first, thirds, fourth, fifth, seventh or eight bin. Each of these bins represent a dominant co-movement pattern for either of the classes, as described in Section 3.2.

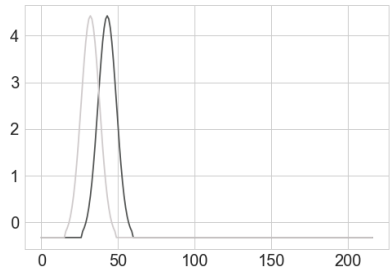
The example illustrated in Figure 3.9 consists of noiseless signals. Although most data collection processes post-process the signals to filter out the noise, it is still an integral part of many real-life applications. Radius threshold parameter β is



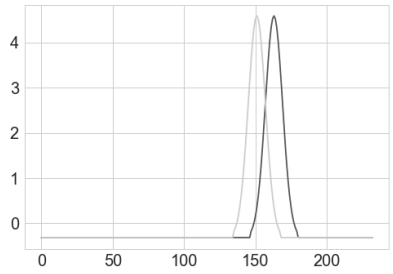
(a) Class 1 example 1.



(b) Class 1 example 2.



(c) Class 2 example 1.



(d) Class 2 example2 .

Figure 3.9. Plotted time series for the multivariate synthetic example without the noise.

introduced as a precaution in such a case. Let us add some small noise to the noiseless. The series are illustrated in Figure 3.11. The corresponding PHD features are mapped in Figure 3.12 when $\beta = 0$. Even in such an easy scenario, the noise in the dataset greatly hinders the discriminatory information in the feature. A radius threshold of 0.35 recovers important information, as in Figure 3.13.

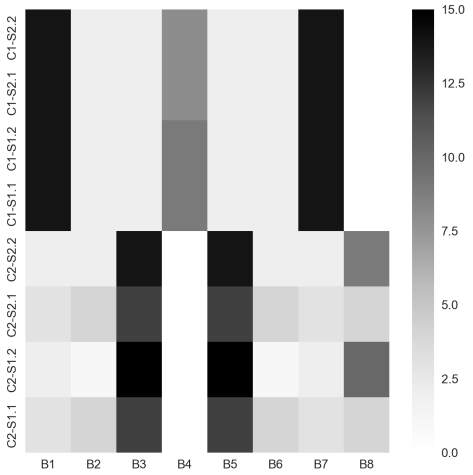
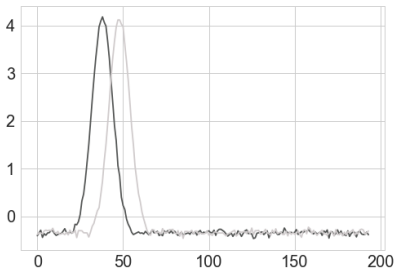
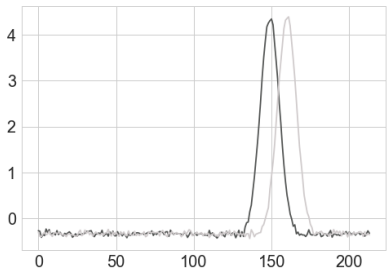


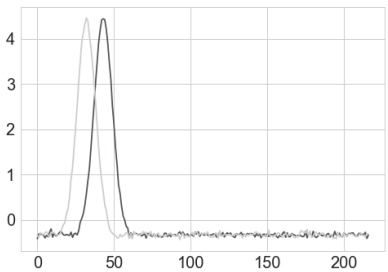
Figure 3.10. Heatmap for the extracted IM features of the series shown in Figure 3.9. The parameter β is taken as 0.



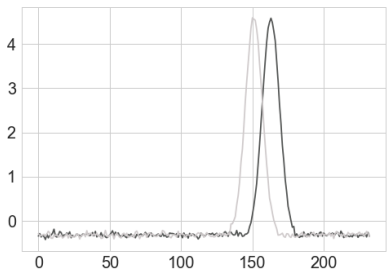
(a) Class 1 example 1.



(b) Class 1 example 2.



(c) Class 2 example 1.



(d) Class 2 example2 .

Figure 3.11. Plotted time series for the multivariate toy example with the noise added.

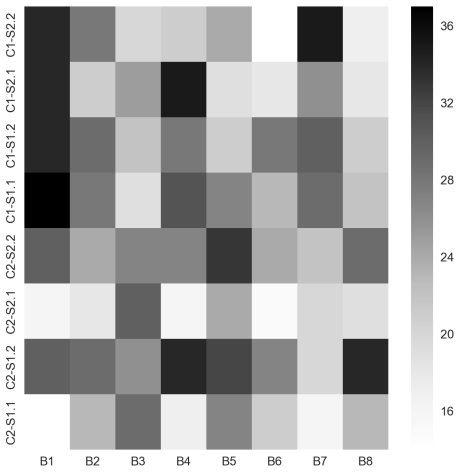


Figure 3.12. Heatmap for the extracted IM features of the series shown in Figure 3.11. The parameter β is taken as 0.

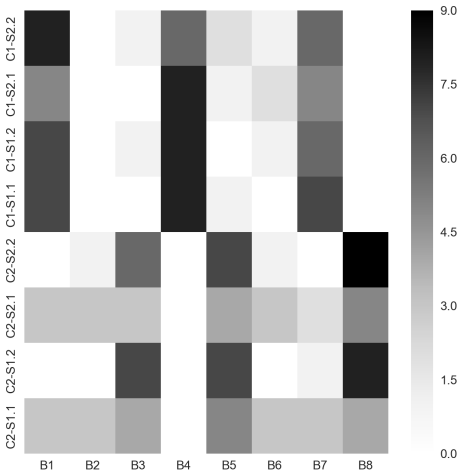


Figure 3.13. Heatmap for the extracted IM features of the series shown in Figure 3.11. The parameter β is taken as 0.35.

4. EXPERIMENTS AND RESULTS

4.1. Benchmark Datasets

An objective evaluation of the proposed method is of uttermost importance in order for it to be comparable and credible. Keogh and Kasetty underscore the importance of avoiding data bias, advising the scientific community to test their method on a wide spectrum of datasets [9]. In line with this idea, a rich collection of 14 datasets were gathered for the purpose of unbiased assessment of the effectiveness of the proposed method. Table 4.1 provides an overview of the benchmark datasets as well as their original sources. The datasets were acquired from Baydogan's personal website [36] where they are already scaled and split into two as training and test sets. The following list identifies the criteria used for the dataset selection process:

- The collection should represent different applications areas where MTS classification problem may arise. It currently includes the following domains: medical condition classification, sign language and gesture recognition, motion capture classification, image outline classification, classification of sensor readings from a manufacturing process.
- The collection should include datasets where varying time series lengths are present in order to demonstrate the applicability of the method on such datasets. Currently there are 11 datasets with varying series lengths.
- The collection should include cases with different number of attributes. The maximum number of attributes found in any dataset is 62 while the minimum is 2.
- The collection should include cases with different number of classes. The maximum number of classes found in any dataset is 95 while the minimum is 2.

Table 4.1. Properties of benchmark datasets used for performance evaluation.

Dataset Name	**Dimension**	**Length**	**Class Count**	**Train Size**	**Test Size**	**Original Source**
Arabic Digits	13	4-93	10	6600	2200	[37]
AUSLAN	22	45-136	95	1140	1425	[18]
Character Trajectories	3	109-205	20	300	2558	[37]
CMU MOCAP S16	62	127-580	2	29	29	[38]
Digits Shape	2	30-98	4	24	16	[39]
ECG	2	39-152	2	100	100	[40]
Japanese Vowels	12	7-29	9	270	370	[37]
KickvsPunch	62	274-841	2	16	10	[38]
Libras	2	45	15	180	180	[37]
PenDigits	2	8	10	300	10692	[37]
Shapes	2	52-98	3	18	12	[39]
Uwave Gesture Library	3	315	8	896	3582	[35]
Wafer	6	104-198	2	298	896	[40]
WalkvsRun	62	128-1918	2	28	16	[38]

4.2. Experimental Setup

Researchers refer to varying experimental setups when it comes to reporting the performance of their proposed methods. For example [23] uses threefold CV, [20] uses leave-one-out CV whereas in [16, 22] a mix of fivefold and tenfold CV is used. In several studies [30–32] error rates are based on the test-train splits defined by the provider of the dataset. Moreover, if the software implementation of the method is not disclosed along with the dataset folder used for producing the results, it almost becomes impossible to have a fair comparison between the approaches.

As for the performance metric, unless the method is specifically developed for binary classification problems, the primary choice is the accuracy score. The accuracy score is simply the number of correctly identified labels over the number of predictions made in total. From a qualitative point-of-view, the accuracy score describes how often a predictor correctly identifies the true label of an incoming instance. In order to provide as much comparability as possible, the results for all datasets listed in Section 4.1 are reported with three different experimental setups: average accuracy over fivefold CV, average accuracy over tenfold CV and accuracy over the predefined train-test splits. For CV methods, stratified sampling is used

The advantage of using CV over predefined train-test split is that the results are more biased towards the specific split for the latter compared to the former. On the other hand, CV splits involves randomness, leading to less than completely fair comparisons. CV methods are found to be pessimistically biased [41]. In order to restrain the pessimism, stratified sampling method is used for CV setups. In other words, the distribution of class frequencies in both sets are kept as close as possible. Lastly, it should be noted that there is a variance-bias trade-off between using fivefold and tenfold CV. As the number of folds increases, the bias decreases while variance increases.

In an orthodox classification scheme, nested CV strategy should be employed to cover parameter optimization and testing in order to avoid over-fitting [42]. However,

such a strategy is not necessary for the proposed method. Out-of-bag errors of the random forest classifier is used for parameter optimization while the CV is carried out on a single layer. An outline of the K-Fold CV strategy for both parameter optimization and reporting accuracy is provided in Figure 4.1. It is trivial to see that for predefined test-train split, the procedure is almost the same except that only one and non-random split is used.

The same 5 by 5 parameter grid is used for each single experimentation. The λ parameter is chosen from the following values: 7, 10, 13, 16 and 19. The β parameter is chosen from among the following values: 0, 0.025, 0.05, 0.075, 0.1.

Lastly, a brief information about the development environment should be mentioned. The algorithm is implemented with Python version 3.5.2 [43] and using packages NumPy version 1.11.3, scikit-learn version 0.18.1. The figures are obtained using packages Matplotlib version 2.0.0 and Seaborn version 0.7.1. The random forest classifier implementation of scikit-learn is used with default hyper-parameters except for the number of decision trees, which is increased from 10 to 100. A working version of the code and the dataset used ca be found in author's personal website [44].

4.3. Results

This section presents the results for the various experimental setups mentioned in Section 4.2. Before going into the full results however, it is worth assessing the added value the PHD features. Table 4.2 shows how does the classification perform when only IM features are used. Note that the results are for the 10 times replicated trial on the predefined test-train split. As mentioned in Section 3.4, despite being trivial, IM features alone provide a strong basis. However, PHD features visibly increase the performance in some of the datasets. Datasets that fall into the lower part of Figure 4.2 denote the ones where PHD features helped increase predictive power for the same experimental setup. PHD features are particularly effective for Libras, UWave Gesture Library, KickvsPunch and CMU MOCAP S16 datasets. The common characteristic of these datasets is that they are all motion classification problems. This result

Input: $\mathcal{D}$: a set of N labelled MTS instances, K: number of folds, $\mathcal{H}$: a grid consisting of value pairs for interval count λ and bin count β

Extract IM and PHD features for the whole set $\mathcal{D}$ using each value available in H to construct the complete feature set $\mathcal{F}$

Randomly split $\mathcal{F}$ into K mutually exclusive partitions $\mathcal{F}_k$, $k = 1, \ldots, K$, keeping the class distributions as balanced as possible

for $k = 1$ to K **do**

Retain $\mathcal{F}_k$ as the test set. Use $\mathcal{Z} = \mathcal{D} \setminus \mathcal{F}_k$ as the training set.

Train a random forest classifier for all possible value pairs in the grid H, using $\mathcal{Z}$, obtain OOB errors.

Choose $h^* \in \mathcal{H}$, the value pair with the lowest OOB error. Let RF^* be the classifier model associated with that pair.

Calculate test error e_k, predicting the labels of $\mathcal{F}_k$ using RF^*

end for

return Report the average error for the K-Fold $\bar{e} = \sum_{k=1}^{K}$

Figure 4.1. K-Fold Cross Validation Strategy.

might be a hint to the claim that PHD features are more informative for a certain type of applications. However, such a claim requires deeper analysis for a complete justification.

Table 4.3, 4.5 and 4.4 shows the results for predefined train-test split, tenfold CV and fivefold CV respectively. Although the main indicator of predictive performance is taken as the mean of the replications, it is worth reporting the minimum, median and maximum as well in order to give an idea about the best and the worst case scenario along with the variability in predictive power. It is visible in the tables that the classification scheme provides a robust performance with respect to the randomness in the solution space.

Finally, Table 4.6 overviews the comparison of IM-PHD against other state-of-the-art MTS classification methods. The strength of IM-PHD comes from its universality.

Table 4.2. Results with only IM features. Tested on the predefined train-test split. Reported statistics are of 10 replications.

Dataset Name	**Min**	**Mean**	**Median**	**Max**
Arabic Digits	0.040	0.044	0.044	0.049
AUSLAN	0.040	0.045	0.045	0.051
Character Trajectories	0.024	0.030	0.030	0.036
CMU MOCAP S16	0.034	0.034	0.034	0.034
Digits Shape	0.000	0.000	0.000	0.000
ECG	0.130	0.153	0.150	0.180
Japanese Vowels	0.035	0.041	0.042	0.046
KickvsPunch	0.100	0.140	0.100	0.200
Libras	0.217	0.238	0.242	0.256
PenDigits	0.074	0.081	0.079	0.091
Shapes	0.000	0.000	0.000	0.000
Uwave Gesture Library	0.053	0.056	0.056	0.058
Wafer	0.015	0.021	0.017	0.031
WalkvsRun	0.000	0.000	0.000	0.000

It does not rank first in most of the datasets (except for the ones where there is a tie for rank 1), but it usually ranks second or third. Even for the single case of Libras dataset where it ranks the last, the error rate is not severe. Therefore the algorithm positions itself as a to-go method where the user cannot detect a more suitable match for the application domain at hand. It is inferred from Section 3.4 that the method is application independent and the foundations are purely based on the statistical nature of multivariate time series.

Table 4.3. Results with full IM-PHD features. Tested on the predefined train-test split. Reported statistics are of 10 replications.

Dataset Name	**Min**	**Mean**	**Median**	**Max**
Arabic Digits	0.033	0.036	0.035	0.039
AUSLAN	0.042	0.049	0.047	0.061
Character Trajectories	0.021	0.025	0.024	0.031
CMU MOCAP S16	0.000	0.000	0.000	0.000
Digits Shape	0.000	0.000	0.000	0.000
ECG	0.130	0.154	0.150	0.180
Japanese Vowels	0.035	0.039	0.038	0.046
KickvsPunch	0.000	0.000	0.000	0.000
Libras	0.122	0.139	0.139	0.150
PenDigits	0.073	0.077	0.077	0.080
Shapes	0.000	0.000	0.000	0.000
Uwave Gesture Library	0.032	0.035	0.035	0.036
Wafer	0.012	0.016	0.014	0.031
WalkvsRun	0.000	0.000	0.000	0.000

4.4. Sensitivity Analysis

In real life applications of classification schemes, it is crucial that the experimenter is able to infer how her/his model will behave when fresh data arrive. A good model should hint how the predictive performance would behave for the incoming data over a set of its parameters. Two of the benchmark datasets were chosen in order to asses this attribute of IM-PHD features: UWave Gesture Library and Character Trajectories. For each dataset, tenfold CV is applied by ten different replications where for each replication, out-of-bag errors on training split and test errors on test split are recorded separately for each parameter pair. The average of these error measurements are illustrated in Figure 4.3 and 4.4. Note once again that out-of-bag error rate is used

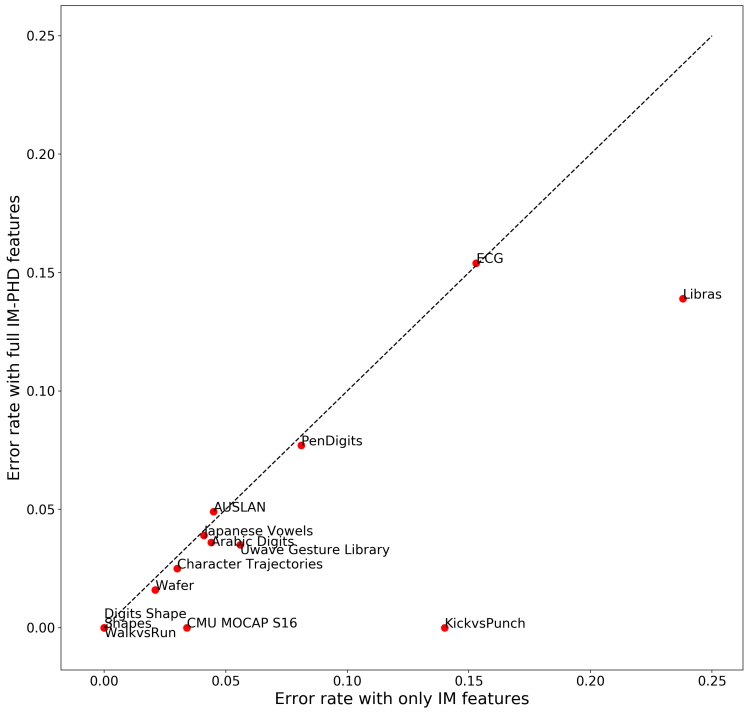


Figure 4.2. Error rates comparison between only IM features version and full IM-PHD features version.

for parameter optimization phase of the method, therefore they must be compared with the test error rate. It is clearly visible from the figures that out-of-bag error infers the true behaviour of the test error against the parameter grid with a high accuracy.

4.5. Computational Complexity

As mentioned in Section 2.2, an ideal classification should be computationally tractable in order to provide feasible production in real-life applications. The important metric here to track is the unit prediction latency. This is because the training phase is usually completed off-line for model building and updates so that what matters for the production mode is the time to classify. Prediction process consists of four phases: differencing, IM features extraction, PHD features extraction, tree traversal

Table 4.4. Results with full IM-PHD features. Tested with fivefold CV.

Dataset Name	**Min**	**Mean**	**Median**	**Max**
Arabic Digits	0.016	0.043	0.036	0.091
AUSLAN	0.011	0.030	0.017	0.089
Character Trajectories	0.005	0.020	0.019	0.040
CMU MOCAP S16	0.000	0.015	0.000	0.077
Digits Shape	0.000	0.000	0.000	0.000
ECG	0.103	0.164	0.171	0.275
Japanese Vowels	0.023	0.034	0.024	0.054
KickvsPunch	0.000	0.040	0.000	0.200
Libras	0.067	0.144	0.133	0.253
PenDigits	0.006	0.009	0.009	0.015
Shapes	0.000	0.000	0.000	0.000
Uwave Gesture Library	0.017	0.023	0.022	0.028
Wafer	0.000	0.011	0.013	0.017
WalkvsRun	0.000	0.020	0.000	0.100

for classification. In this section, the behaviour of prediction latency is discussed with respect to altering conditions in the problem. The algorithm is implemented with Python version 3.5.2 [43] in a 64-bit Windows 10 system with 8GB RAM, i5-3230M CPU @ 2.60GHz. The durations reported here are obtained on a single core.

The Wafer dataset [40] is used for each experiment as it provides a good mix of high-dimensional, sufficiently long time series with a satisfying number of testing instances. The original dataset consists of 1194 instances (298 training and 896 test) of 6-dimensional multivariate time series objects with length varying between 104 and 198. Each MTS in this dataset is reduced to length 104 prior to the experiments. The computation time is reported in seconds per instance for varying number of dimensions, database size and time series length as well as varying number of intervals λ for the feature extraction phase. The radius cut parameter β is not considered in this section

Table 4.5. Results with full IM-PHD features. Tested with tenfold CV.

Dataset Name	**Min**	**Mean**	**Median**	**Max**
Arabic Digits	0.008	0.040	0.035	0.113
AUSLAN	0.000	0.022	0.013	0.063
Character Trajectories	0.003	0.015	0.009	0.052
CMU MOCAP S16	0.000	0.017	0.000	0.167
Digits Shape	0.000	0.000	0.000	0.000
ECG	0.000	0.159	0.174	0.400
Japanese Vowels	0.000	0.036	0.032	0.078
KickvsPunch	0.000	0.000	0.000	0.000
Libras	0.033	0.129	0.122	0.233
PenDigits	0.004	0.008	0.009	0.015
Shapes	0.000	0.000	0.000	0.000
Uwave Gesture Library	0.011	0.019	0.019	0.031
Wafer	0.000	0.011	0.013	0.025
WalkvsRun	0.000	0.017	0.000	0.167

as it is trivial to infer that it will not have a significant effect on the duration. The reader should also note that the standardization step is ignored in this discussion. The experimental setup is designed so that for each possible condition:

- The Wafer dataset is modified accordingly in order to represent the varying condition,
- The experiment is replicated 10 times and the median is reported,
- Other variables are kept at its original level in order to keep ceteris paribus. Except for the last experiment, λ is taken as 7 and β is taken as 0.

One of the most important observations made from the results is that the PHD features extraction phase clearly dominates other components in the prediction process

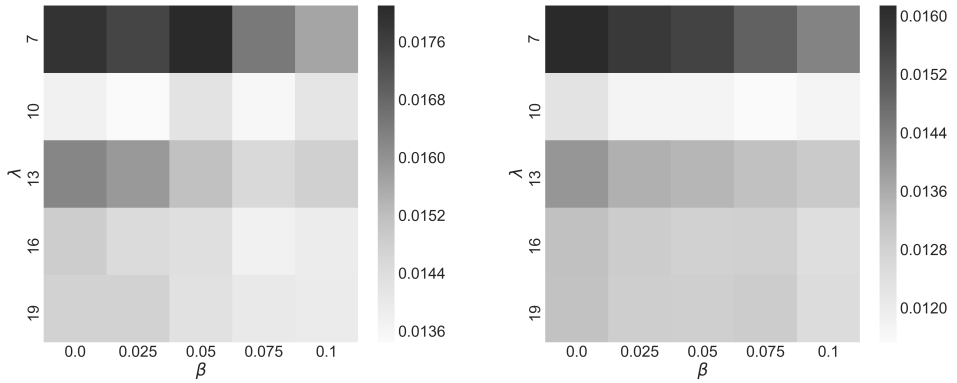
Table 4.6. Comparison of test error results against gRSF, LPS and SMTS.

Dataset Name	**IM-PHD**	**gRSF**	**LPS**	**SMTS**	**Rank**
Arabic Digits	0.036	0.025	0.029	0.036	3
AUSLAN	0.049	0.045	0.246	0.053	2
Character Trajectories	0.025	0.006	0.035	0.008	3
CMU MOCAP S16	0.000	0.000	0.000	0.003	1
Digits Shape	0.000	-	0.000	-	1
ECG	0.154	0.120	0.180	0.182	2
Japanese Vowels	0.039	0.200	0.049	0.031	2
KickvsPunch	0.000	0.000	0.100	0.150	1
Libras	0.139	0.089	0.097	0.091	4
PenDigits	0.077	0.068	0.069	0.083	3
Shapes	0.000	-	0.000	-	1
Uwave Gesture Library	0.035	0.071	0.020	0.059	2
Wafer	0.016	0.008	0.038	0.035	2
WalkvsRun	0.000	0.000	0.000	0.000	1

in terms of elapsed time. Accordingly, the behaviour of PHD features extraction phase is the same as the behaviour of total elapsed time against varying conditions.

The number of dimensions in a dataset is the Achilles' heel of the algorithm. As the number of dimensions increase, the PHD features extraction phase is prolonged quadratically (Figure 4.5) since the number of two-way interactions increase as such. In other words, the worst-case time complexity of PHD features extraction is $O(M^2)$. Still, the complexity is polynomial time and therefore considered as tractable.

In the next experiment, the length of the series is contracted with respect to the original length of 104 by a certain fraction. As the series length increases, the number of data points to be processed increases as well. It is also worthwhile to note that



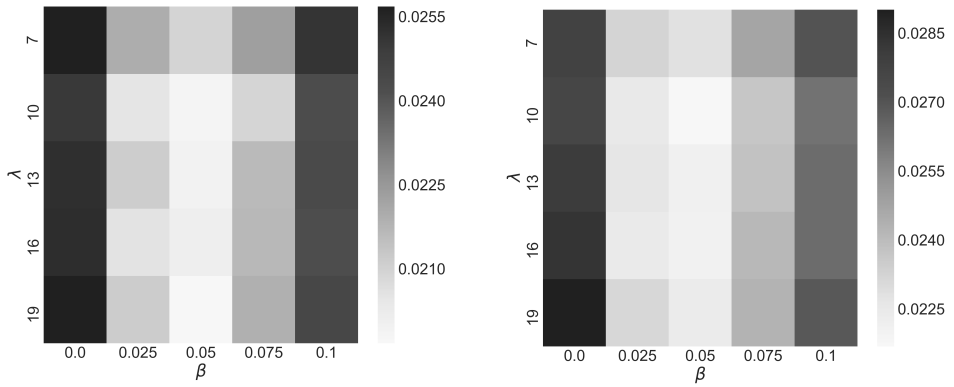
(a) Test error. (b) OOB error.

Figure 4.3. Sensitivity analysis on a 5x5 parameter grid for Character Trajectories dataset.

this increase happens linearly. This why the linear increasing trend in Figure 4.6 is expected.

A similar but oppositely directed behaviour is observable for increasing size of the MTS database. The original dataset had 896 test instances. Certain fractions of samples are acquired from the original dataset and the prediction is applied to that sample. The results are illustrated in Figure 4.7. Per unit prediction latency becomes shorter for increasing database size probably because of decreasing per unit overhead time costs.

Finally, the number of intervals λ linearly increases the IM feature extraction phase, as seen in Figure 4.8. Yet, the IM extraction phase is dwarfed by the PHD extraction phase and therefore it does not require special consideration.



(a) Test error. (b) OOB error.

Figure 4.4. Sensitivity analysis on a 5x5 parameter grid for Uwave Gesture Library dataset.

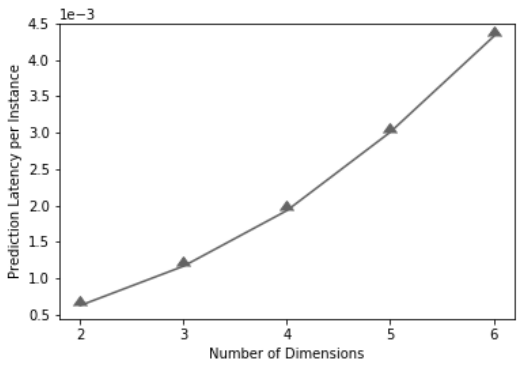


Figure 4.5. Median prediction latency per instance with varying number of dimensions.

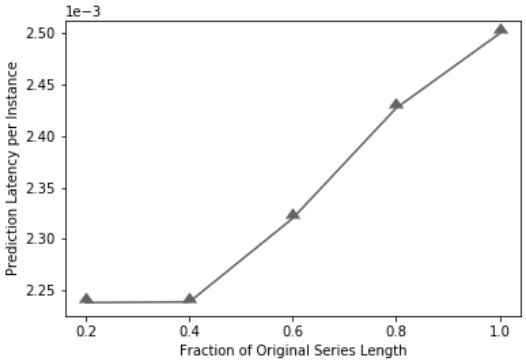


Figure 4.6. Median prediction latency per instance with varying fraction of original series length.

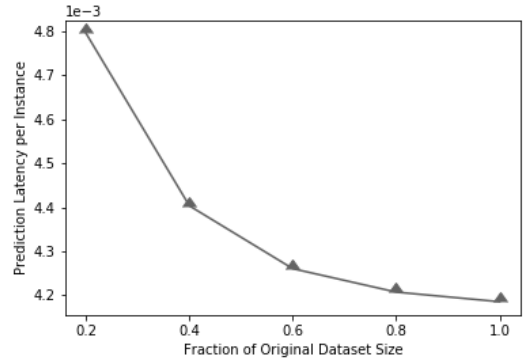


Figure 4.7. Median prediction latency per instance with varying fraction of original dataset size.

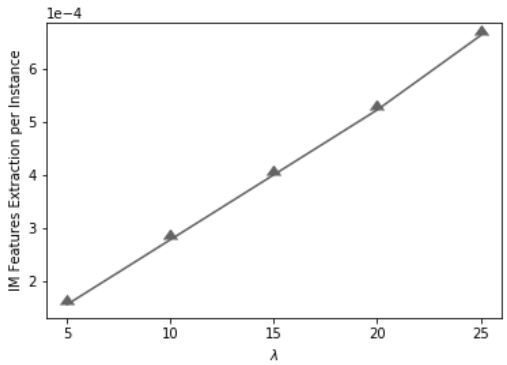


Figure 4.8. Median prediction latency per instance with varying number of intervals.

5. CONCLUSION

5.1. An Overview of the Thesis

The aim of this thesis was to propose a general-purpose, scalable, feature-based MTS classification method with interpretable features effectively representing individual and inter-dimensional characteristics. In order to do so, two types of features were designed and fused together to form the single feature vector that reduces the dimensionality of the MTS and brings it to a mathematical structure that is exploitable by standard classification algorithms.

The feature extraction algorithm requires tuning of only two parameters which is an acceptable number. One of the parameters control the resolution at which level information is obtained from each dimension while the other acts as a filtering threshold to avoid noise in co-movement information.

Random forests were used as the sole classification option for their effective handling of high-dimensional feature vectors and the option to use out-of-bag errors for parameter optimization.

The results on benchmark datasets have shown that the features are effective all-round, regardless of the application domain. It is also shown that predictive time complexity of the model is polynomial and therefore tractable.

5.2. Directions for Future Study

Due to time and resource limitations of this study as well as to keep the scope of the thesis at a manageable level, a number of possible research directions that would improve the method are discarded. This section will briefly mention these directions.

5.2.1. Handling High-dimensional Multivariate Time Series

As the number of dimensions increase in the problem, the IM-PHD method faces the challenge of the curse of dimensionality. The number of features in the feature vector increases polynomially with increasing number of dimensions. The chances of providing irrelevant and noisy features increases with such behaviour, introducing sparsity in the data with a fixed number of samples and leading to the problem of overfitting. There might be several alternatives to reduce the dimensionality of the IM-PHD vectors. The simplest example would be choosing a random subset of the possible 2-combinations of dimensions in order to keep the cardinality of the set at a desired level. This is of course, a very naive approach that does not take relevancy into account. Domain expertise could be incorporated at this stage in order to choose a smart subset of the possible 2-way interactions, for example by eliminating relationships that are known to be irrelevant to the problem. As a more automated alternative, a smart subset may be obtained in a data-driven way with the use of sparse inverse covariance estimation methods [45]. An undirected graph of relationships would yield a subset of interactions for each class. These subsets would then be united to form the data-driven smart subset of interactions.

5.2.2. Different Bin Boundaries and Resolution for PHD Features

The boundaries and the count of bins are fixed for ease of interpretability. However, it is possible that different bin boundaries and polar resolutions might be more effective for different applications. In a future study, this could be brought as an option. Yet this requires careful experimentally backed justification for which it was discarded from the scope of this study.

5.2.3. Visualisation Strategy for Interpretablity

Although the IM-PHD feature extraction significantly reduces the dimensionality of a given MTS and the all of the individual features have their simple explanation, the vector itself is still quite hard to interpret just by looking at the numbers. Patterns

in the dataset should be visualised through factorial plotting schemes and appropriate plot styles must be chosen to deliver important high-level semantic information about the problem at hand.

5.2.4. Customized Classification Algorithms per Application

Much better accuracy results or much shorter prediction latencies could have been achieved with customized classifiers for each dataset. What is meant by a customized classifier can be a hyper-parameter optimized random forest classifier or a completely different algorithm such as neural networks or gradient boosting. This was considered to be out of scope of this study as the focus is kept rather on the feature extraction phase. Consequently, identical random forest classifiers were used for each single experiment mentioned in this study. However, it is intended to provide a technical report that contains information about how the predictive performance and computational complexity changes with respect to differing types of classification algorithms.

5.2.5. Additional Features

Although IM-PHD features provide a good basis for high-accuracy predictions, there are still uncovered characteristics of MTS datasets which are not captured with neither IM nor PHD features. Most notable example would be auto-correlative and cross-correlative information, something that was completely ignored by this study. Modifications to the existing algorithm that would bring additional informative representation without increasing the dimensionality too much, would be an important contribution to the method.

REFERENCES

1. Jensen, C. S. and R. T. Snodgrass, "Temporal data management", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 1, pp. 36–44, 1999.

2. Baydogan, M. G., *Modeling time series data for supervised learning*, Ph.D. Thesis, Arizona State University, 2012.

3. Mörchen, F., *Time series knowledge mining*, Ph.D. Thesis, Philipps-Universität Marburg, 2006.

4. Esling, P. and C. Agon, "Time-series data mining", *ACM Computing Surveys (CSUR)*, Vol. 45, No. 1, p. 12, 2012.

5. Last, M., A. Kandel and H. Bunke, *Data mining in time series databases*, Vol. 57, World scientific, 2004.

6. Mitsa, T., *Temporal data mining*, CRC Press, 2010.

7. Lin, W., M. A. Orgun and G. J. Williams, "An overview of temporal data mining", *Proceedings of the 1st Australian data mining workshop*, pp. 83–90, 2002.

8. Fu, T.-c., "A review on time series data mining", *Engineering Applications of Artificial Intelligence*, Vol. 24, No. 1, pp. 164–181, 2011.

9. Keogh, E. and S. Kasetty, "On the need for time series data mining benchmarks: a survey and empirical demonstration", *Data Mining and knowledge discovery*, Vol. 7, No. 4, pp. 349–371, 2003.

10. Mitra, S. and T. Acharya, "Gesture recognition: A survey", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 37, No. 3, pp. 311–324, 2007.

11. Bagnall, A., J. Lines, A. Bostrom, J. Large and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances", *Data Mining and Knowledge Discovery*, pp. 1–55, 2016.

12. Xing, Z., J. Pei and E. Keogh, "A brief survey on sequence classification", *ACM Sigkdd Explorations Newsletter*, Vol. 12, No. 1, pp. 40–48, 2010.

13. Deng, H., G. Runger, E. Tuv and M. Vladimir, "A time series forest for classification and feature extraction", *Information Sciences*, Vol. 239, pp. 142–153, 2013.

14. Bailly, A., S. Malinowski, R. Tavenard, T. Guyet and L. Chapel, "Bag-of-Temporal-SIFT-Words for time series classification", *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, 2015.

15. Berndt, D. J. and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series", *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, pp. 359–370, 1994.

16. Baydogan, M. G. and G. Runger, "Learning a symbolic representation for multivariate time series classification", *Data Mining and Knowledge Discovery*, Vol. 29, No. 2, pp. 400–422, 2015.

17. Alpaydin, E., *Introduction to machine learning*, MIT press, 2014.

18. Kadous, M. W., *Temporal classification: Extending the classification paradigm to multivariate time series*, Ph.D. Thesis, The University of New South Wales, 2002.

19. Geurts, P. and L. Wehenkel, "Segment and combine approach for non-parametric time-series classification", *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 478–485, Springer, 2005.

20. Chaovalitwongse, W. and P. Pardalos, "On the time series support vector machine using dynamic time warping kernel for brain activity classification", *Cybernetics*

and Systems Analysis, Vol. 44, No. 1, pp. 125–138, 2008.

21. Akl, A. and S. Valaee, "Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing", *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 2270–2273, IEEE, 2010.

22. Orsenigo, C. and C. Vercellis, "Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification", *Pattern Recognition*, Vol. 43, No. 11, pp. 3787–3794, 2010.

23. Li, C., L. Khan and B. Prabhakaran, "Real-time classification of variable length multi-attribute motions", *Knowledge and Information Systems*, Vol. 10, No. 2, pp. 163–183, 2006.

24. Weng, X. and J. Shen, "Classification of multivariate time series using locality preserving projections", *Knowledge-Based Systems*, Vol. 21, No. 7, pp. 581–587, 2008.

25. Spiegel, S., J. Gaebler, A. Lommatzsch, E. De Luca and S. Albayrak, "Pattern recognition and classification for multivariate time series", *Proceedings of the fifth international workshop on knowledge discovery from sensor data*, pp. 34–42, ACM, 2011.

26. McGovern, A., D. H. Rosendahl, R. A. Brown and K. K. Droegemeier, "Identifying predictive multi-dimensional time series motifs: an application to severe weather prediction", *Data Mining and Knowledge Discovery*, Vol. 22, No. 1, pp. 232–258, 2011.

27. Ordonez, P., T. Armstrong, T. Oates and J. Fackler, "Using modified multivariate bag-of-words models to classify physiological data", *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pp. 534–539, IEEE, 2011.

28. Lin, J., E. Keogh, W. Li and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series", *Data Mining and knowledge discovery*, Vol. 15, No. 2, p. 107, 2007.

29. Ye, L. and E. Keogh, "Time series shapelets: a novel technique that allows accurate, interpretable and fast classification", *Data mining and knowledge discovery*, Vol. 22, No. 1, pp. 149–182, 2011.

30. Wistuba, M., J. Grabocka and L. Schmidt-Thieme, "Ultra-Fast Shapelets for Time Series Classification", *CoRR*, Vol. abs/1503.05018, 2015.

31. Karlsson, I., P. Papapetrou and H. Boström, "Generalized random shapelet forests", *Data Mining and Knowledge Discovery*, Vol. 30, No. 5, pp. 1053–1085, 2016.

32. Baydogan, M. G. and G. Runger, "Time series representation and similarity based on local autopatterns", *Data Mining and Knowledge Discovery*, Vol. 30, No. 2, pp. 476–509, 2016.

33. Breiman, L., "Random forests", *Machine learning*, Vol. 45, No. 1, pp. 5–32, 2001.

34. Friedman, J., T. Hastie and R. Tibshirani, *The elements of statistical learning*, Vol. 1, Springer series in statistics New York, 2001.

35. Chen, Y., E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen and G. Batista, *The UCR Time Series Classification Archive*, 2015, `www.cs.ucr.edu/ eamonn/time_series_data/`, accessed at April 2017.

36. Baydogan, M. G., *Files-Data Set—Mustafa Baydogan*, 2014, `http://www.mustafabaydogan.com/files/viewcategory/20-data-sets.html`, accessed at May 2017.

37. Lichman, M., *UCI Machine Learning Repository*, 2013,

"http://archive.ics.uci.edu/ml", accessed at May 2017.

38. Hodgins, J. K., *CMU Motion Capture Library*, 2017, http://mocap.cs.cmu.edu/, accessed at May 2017.

39. Sübakan, Y. C., B. Kurt, A. T. Cemgil and B. Sankur, "Probabilistic sequence clustering with spectral learning", *Digital Signal Processing*, Vol. 29, pp. 1–19, 2014.

40. Olszewski, R. T., *Bobski's World*, 2001, http://www.cs.cmu.edu/ bobski/, accessed at May 2017.

41. Kohavi, R., "A study of cross-validation and bootstrap for accuracy estimation and model selection", *IJCAI'95: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, pp. 1137–1145, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995.

42. Cawley, G. C. and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation", *Journal of Machine Learning Research*, Vol. 11, No. Jul, pp. 2079–2107, 2010.

43. Python Core Team, *Python: A dynamic, open source programming language, version 3.5*, 2016, https://www.python.org/, accessed at February 2017.

44. Sergin, N. D., *Research—Dorukhan Sergin*, 2017, https://www.dorukhansergin.org/research/, accessed at July 2017.

45. Friedman, J., T. Hastie and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso", *Biostatistics*, Vol. 9, No. 3, pp. 432–441, 2008.