Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data

Robert T. Olszewski

February 2001
CMU-CS-01-108

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Thesis Committee:

Roy Maxion, co-chair
Dan Siewiorek, co-chair
Christos Faloutsos
David Banks, DOT

This research was sponsored by the Defense Advanced Project Research Agency (DARPA) and the Air Force Research Laboratory (AFRL) under grant #F30602-96-1-0349, the National Science Foundation (NSF) under grant #IRI-9224544, and the Air Force Laboratory Graduate Fellowship Program sponsored by the Air Force Office of Scientific Research (AFOSR) and conducted by the Southeastern Center for Electrical Engineering Education (SCEEE).

Keywords: Structural pattern recognition, classification, feature extraction, time series, Fourier transformation, wavelets, semiconductor fabrication, electrocardiography.

Abstract

Pattern recognition encompasses two fundamental tasks: description and classification. Given an object to analyze, a pattern recognition system first generates a description of it (i.e., the pattern) and then classifies the object based on that description (i.e., the recognition). Two general approaches for implementing pattern recognition systems, statistical and structural, employ different techniques for description and classification. Statistical approaches to pattern recognition use decision-theoretic concepts to discriminate among objects belonging to different groups based upon their quantitative features. Structural approaches to pattern recognition use syntactic grammars to discriminate among objects belonging to different groups based upon the arrangement of their morphological (i.e., shape-based or structural) features. Hybrid approaches to pattern recognition combine aspects of both statistical and structural pattern recognition.

Structural pattern recognition systems are difficult to apply to new domains because implementation of both the description and classification tasks requires domain knowledge. Knowledge acquisition techniques necessary to obtain domain knowledge from experts are tedious and often fail to produce a complete and accurate knowledge base. Consequently, applications of structural pattern recognition have been primarily restricted to domains in which the set of useful morphological features has been established in the literature (e.g., speech recognition and character recognition) and the syntactic grammars can be composed by hand (e.g., electrocardiogram diagnosis). To overcome this limitation, a domain-independent approach to structural pattern recognition is needed that is capable of extracting morphological features and performing classification without relying on domain knowledge. A hybrid system that employs a statistical classification technique to perform discrimination based on structural features is a natural solution. While a statistical classifier is inherently domain independent, the domain knowledge necessary to support the description task can be eliminated with a set of generally-useful morphological features. Such a set of morphological features is suggested as the foundation for the development of a suite of structure detectors to perform generalized feature extraction for structural pattern recognition in time-series data.

The ability of the suite of structure detectors to generate features useful for structural pattern recognition is evaluated by comparing the classification accuracies achieved when using the structure detectors versus commonly-used statistical feature extractors. Two real-world databases with markedly different characteristics and established ground truth serve as sources of data for the evaluation. The classification accuracies achieved using the features extracted by the structure detectors were consistently as good as or better than the classification accuracies achieved when using the features generated by the statistical feature extractors, thus demonstrating that the suite of structure detectors effectively performs generalized feature extraction for structural pattern recognition in time-series data.

Acknowledgements

This document is the final result of a course of research that I began quite some time ago. While the journey has been a lengthy one, fraught with twists and turns, it has not been a solitary one. I would like to thank my entire committee—Roy Maxion, Dan Siewiorek, Christos Faloutsos, and David Banks—for their time and patience in shepherding me through the process. I would particularly like to thank David Banks for his consistent and unwavering encouragement that helped me focus on the light at the end of the tunnel.

I would also like to acknowledge the assistance of several other people. Laura Forsyth, Dan Siewiorek's executive assistant, had the unenviable task of arranging and rearranging committee meetings so as to accommodate the overcommitted and ever-changing schedules of the individual committee members. I am grateful for the time taken by Carl Almgren to contribute his domain knowledge and expertise regarding semiconductor fabrication. And, for keeping me level-headed, I would like to thank Pat Loring.

On a more personal level, there are several people that deserve acknowledgement. I am grateful for the support of my parents and sister which has enabled me to pursue and achieve my goals. I would like to thank Dean Behrens for his friendship and for not giving up on me. Orna Grumberg, whether she realizes it or not, provided a welcome relief to the daily grind with her annual visits and indomitable spirit. And finally, I will be eternally indebted to the one person who suffered the most throughout this research. Alon, this is for you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The essential problem of pattern recognition is to identify an object as belonging to a particular group. Assuming that the objects associated with a particular group share common attributes more so than with objects in other groups, the problem of assigning an unlabeled object to a group can be accomplished by determining the attributes of the object (i.e., the pattern) and identifying the group of which those attributes are most representative (i.e., the recognition). If information about the universe of all possible objects and the groups to which they can be assigned is known, then the identification problem is straightforward in that the attributes that best discriminate among groups and the mapping from attributes to groups can both be determined with certainty. When the information about the identification problem is imperfect or incomplete, then the attributes and mapping to use must be inferred from example objects whose group membership is known.

Given the goal of classifying objects based on their attributes, the functionality of an automated pattern recognition system can be divided into two basic tasks: the description task generates attributes of an object using *feature extraction* techniques, and the classification task assigns a group label to the object based on those attributes with a *classifier*. The description and classification tasks work together to determine the most accurate label for each unlabeled object analyzed by the pattern recognition system. This is accomplished with a *training phase* that configures the algorithms used in both the description and classification tasks based on a collection of objects whose labels are known—i.e., a *training set*. During the training phase, a training set is analyzed to determine the attributes and mapping which assigns labels to the objects in the training set with the fewest errors. Once trained, a pattern recognition system assigns a classification to an unlabeled object by applying the mapping to the attributes of that object. A measure of the efficacy of a trained pattern recognition system can be computed by comparing the known labels with the labels assigned by the classification task to the training set: as the agreement between known and assigned labels increases, the accuracy of the pattern recognition system increases. Such a methodology for configuring and evaluating the description and classification tasks of a pattern recognition system is called *supervised learning*. If a training set is not available, then *unsupervised learning* techniques can be used.

The generality of the description and classification architecture in conjunction with the flexibility afforded by the training phase makes automated pattern recognition systems useful for solving a wide range of real-world problems. The objects under analysis in real-world pattern recognition systems typically are data sets containing attributes that were automatically collected and are representative of the physical or behavioral objects to be identified. In electrocardiogram analysis, for example, a data set might contain the electrical activity recorded during one heartbeat by an electrode placed on a patient, and a collection of such data sets would constitute a description of a patient's cardiac behavior over a period of time. Given such a collection of data sets, the goal of a pattern recognition system for electrocardiogram analysis would be to classify each heartbeat (i.e., data set) as being indicative of normal or abnormal cardiac behavior, thereby assisting the physician in diagnosing the condition of a patient's heart. Other real-world applications of automated pattern recognition include the following:

- Industrial Applications
  - Character recognition
  - Process control
  - Signature analysis
  - Speech analysis
- Medical Applications
  - Electroencephalogram analysis
  - Genetic studies
- Government Applications
  - Smog detection and measurement
  - Traffic analysis and control
  - Fingerprint matching
- Military Applications
  - Sonar detection and classification
  - Automatic target recognition

See Friedman and Kandel [29], Fu [30], Jain [47], Nadler [65], and Young and Fu [91] for a discussion of these and other applications of pattern recognition.

1.1. Structural Pattern Recognition

There are two fundamental approaches to implementing a pattern recognition system: statistical and structural. Each approach employs different techniques within the description and classification tasks which constitute a pattern recognition system. Statistical pattern recognition [24][32][47] draws from established concepts in statistical decision theory to discriminate among data from different groups based upon quantitative features of the data. The quantitative nature of statistical pattern recognition, however, makes it difficult to discriminate among groups based on the morphological (i.e., shape-based or structural) subpatterns and their interrelationships embedded within the data. This limitation provided the impetus for the development of a structural approach to pattern recognition.

Structural pattern recognition [31][38][70], sometimes referred to as syntactic pattern recognition due to its origins in formal language theory, relies on syntactic grammars to discriminate among data from different groups based upon the morphological interrelationships (or interconnections) present within the data. Structural pattern recognition systems have proven to be effective for data which contain an inherent, identifiable organization such as image data (which is organized by location within a visual rendering) and time-series data (which is organized by time). The usefulness of structural pattern recognition systems, however, is limited as a consequence of fundamental complications associated with the implementation of the description and classification tasks.

The description task of a structural pattern recognition system is difficult to implement because there is no general solution for extracting structural features, commonly called *primitives*, from data. The lack of a general approach for extracting primitives puts designers of structural pattern recognition systems in an awkward position: feature extractors are necessary to identify primitives in the data, and yet there is no established methodology for deciding which primitives to extract. The result is that feature extractors for structural pattern recognition systems are developed to extract either the simplest and most generic primitives possible or the domain- and application-specific primitives that best support the subsequent classification task. Neither scheme is optimal. Simplistic primitives are domain-independent, but capture a minimum of structural information and postpone deeper interpretation until classification. At the other extreme, domain- and application-specific primitives can be developed with the assistance of a domain expert, but obtaining and formalizing the necessary domain knowledge, called *knowledge acquisition*, can be problematic. To avoid the overhead of knowledge acquisition, existing structural pattern recognition systems rely on morphological features that have been established in the literature as being particularly effective for the domain under analysis.

The classification task of a structural pattern recognition system is difficult to implement because the syntactic grammars embody the precise criteria which discriminate among groups and, therefore, they are by their very nature domain- and application-specific. Grammar inference techniques can be used to construct automatically a grammar from examples, but these methods can fail in the most general cases such as when the target grammar is context free. Consequently, existing structural pattern recognition systems are primarily applied to domains where the syntactic grammars required for classification can be constructed by hand.

1.1.1. Applications to Time-Series Data

Identification problems involving time-series (or waveform) data constitute a subset of pattern recognition applications that is of particular interest because of the large number of domains that involve such data [25][83]. Both statistical and structural approaches can be used for pattern recognition of time-series data: standard statistical techniques have been established for discriminant analysis of time-series data [79], and structural techniques have been shown to be effective in a variety of domains involving time-series data [30]. Structural approaches are particularly appropriate in domains where domain experts classify time-series data sets based on the arrangement of morphological events evident in the waveform—e.g., speech recognition, electrocardiogram diagnosis, seismic activity identification, radar signal detection, and process control.

Structural approaches for pattern recognition in time-series data are typically employed within well-explored domains where the necessary domain knowledge is readily available to guide the implementation of the description and classification tasks. In electrocardiogram diagnosis, for example, the types of morphologies that occur within the waveform and their implications with respect to cardiac behavior are clearly understood. Consequently, the existing body of knowledge within the domain of electrocardiography can serve as a solid foundation for constructing a domain-specific structural pattern recognition system for electrocardiogram diagnosis. Most domains involving time-series data, however, are not nearly as well understood as is electrocardiography. To implement a structural pattern recognition system for poorly-understood domains, knowledge acquisition techniques must be used to assemble the required domain knowledge. To avoid the burden of knowledge acquisition and to enable structural approaches to be applied to unexplored domains where knowledge acquisition techniques would generate an inadequate knowledge base, a domain-independent approach to structural pattern recognition is necessary.

1.1.2. Domain-Independent Structural Pattern Recognition

A domain-independent structural pattern recognition system is one that is capable of acting as a "black box" to extract primitives and perform classification without the need for domain knowledge. Such a system would automatically describe and classify data, thereby eliminating the overhead associated with traditional approaches to structural pattern recognition. A domain-independent structural pattern recognition system for time-series data must incorporate techniques for the description and classification tasks that are not dependent on domain knowledge—i.e., generalized description and generalized classification. Since syntactic grammars are inherently tied to the domain and application, a sensible approach to generalized classification for time-series data is a statistical classifier that performs discrimination based on structural features extracted from the data. Generalized description can be implemented using a foundation of generally-useful morphological features that are effective regardless of the domain.

The field of signal processing offers a suggestion for morphological features that can provide the foundation for generalized description of time-series data. Six fundamental types of modulation commonly used in signal processing systems—constant, straight, exponential, sinusoidal, triangular, and rectangular—entail morphologies deliberately introduced into a continuous medium with the intent of conveying information regardless of the domain or application [7][9][20][62]. Moreover, these six modulation types subsume the small set of domain-independent morphological features commonly extracted by structural pattern recognition systems—straight lines, parabolas, and peaks. A suite of feature extractors which identify morphological features based on these six modulation types, therefore, would constitute a first pass at implementing generalized feature extraction to support domain-independent structural pattern recognition in time-series data.

1.2. Generalized Feature Extraction

The six morphology types suggested by the field of signal processing—constant, straight, exponential, sinusoidal, triangular, and rectangular—serve as the foundation for a set of feature extractors, called structure detectors, for generalized feature extraction in time-series data. Each structure detector fits a unique function to a time-series data set that embodies the morphology type and contains free parameters. A structure detector extracts an instance of its associated morphology type by instantiating values for the free parameters so as to minimize the difference between the raw data and the function; the extracted structure is defined by the function once the free parameters have been fixed. There are six structure detectors, one for each of the six morphology types. The structure detector associated with the rectangular modulation type, however, was generalized to extract instances of trapezoidal modulation so as to increase its descriptive power.

Each structure detector approximates an entire time series with a single structure. Since the values of the free parameters are set so as to minimize the difference between the raw data and the function, the extracted structure must balance the effects of the disparate subregions of the time series by following the general, global trend of the data. Local trends can be captured via the piecewise application of the structure detectors: applying a structure detector to contiguous subregions of a time series such that the union of the subregions is the entire time series and the intersection of the subregions is empty. The application of a structure detector in such a piecewise manner results in an approximation of a time series composed of a sequence of extracted substructures such that each subregion is represented by the same function but with different values instantiated onto the free parameters within each subregion. The local trend in a time series can be better represented by allowing the structure detector used to approximate each subregion to vary among the subregions: the structure detector that most minimizes the difference between the function and the data within a subregion is selected to represent the subregion, regardless of the structure detectors used to represent the other subregions. The piecewise application of these six structure detectors provides the foundation for generalized feature extraction in time-series data.

1.3. Structure Detector Evaluation

The relative efficacy of the structure detectors can be evaluated by comparing the classification accuracies achieved when using the various structure detectors for feature extraction. Additionally, established techniques for extracting features from time-series data can serve as a benchmark: If the classification accuracy achieved with features generated by the structure detectors is at least as high as the accuracies achieved with the other techniques, then it can be concluded that the structure detectors capture characteristics of time-series data suitable for classification at least as well as commonly-used methods. Since the structure detectors subsume general-purpose feature extractors for structural pattern recognition in time-series data (e.g., chain codes, curve fitting), the comparable feature extractors must be drawn from those techniques commonly used for statistical pattern recognition: the identity transformation (i.e., the extracted features are the raw data itself), the Fourier transformation, and the wavelet transformation.

An experiment to evaluate the efficacy of the structure detectors under a range of conditions incorporates several factors: feature extraction method, training set size, composition of training set, and data preprocessing technique. Each experimental factor is allowed to vary over a range of values in order to fully explore the performance of the various feature extractors. For each combination of experimental factors, the experiment proceeds by randomly selecting a training set from the collection of data sets under analysis, configuring the feature extractor with the randomly-selected training set, extracting the attributes from each data set using the trained feature extractor, and classifying the entire collection of data sets based on the extracted features. Two different domains serve as a source of time-series data for the experiment, namely semiconductor fabrication and electrocardiography. The data sets for each domain were inspected by appropriate domain experts, and a label of normal or abnormal was assigned to each data set. The labels assigned by the domain experts are considered to be completely accurate—i.e., the "ground truth."

The classification accuracy achieved by the classifier can be summarized with a pair of percentages: the percent of data sets known to be normal that are classified as normal, and the percent of data sets known to be abnormal that are classified as abnormal. To compensate for the random selection involved in assembling the training set, twenty experimental iterations are performed for each combination of experimental factors and the mean and standard deviation of the percentage pairs over the twenty iterations are computed. Therefore, the overall classification accuracy for each combination of experimental factors is represented by two pairs of values: the mean and standard deviation of the percent of known normal data sets classified as normal, and the mean and standard deviation of the percent of known abnormal data sets classified as abnormal.

The mean classification accuracies achieved by the structure detectors are generally as good as or better than those achieved by the statistical feature extraction methods. Rarely did any of the structure detectors perform poorer overall than the statistical feature extraction methods. The ability of the structure detectors to achieve classification accuracies comparable to the baseline statistical methods demonstrates that the suite of structure detectors effectively performs generalized feature extraction for structural pattern recognition in time-series data.

1.4. Thesis Outline

This thesis explores the topic of structural pattern recognition in time-series data. Specifically, the complications involved in implementing a structural pattern recognition system are examined and shown to act as a barrier to applying structural approaches to new identification problems. A modification to traditional structural pattern recognition that eliminates these problems is proposed, and its efficacy is evaluated.

Chapter 2 explores issues pertaining to the implementation of automated pattern recognition systems. The difficulties associated with implementing a structural pattern recognition system are described and demonstrated to manifest in existing systems designed to analyze time-series data. An architecture for domain-independent structural pattern recognition is suggested as a remedy that would make structural approaches more readily applicable to new domains. The cornerstone of this modified approach to structural pattern recognition is generalized feature extraction. A suite of structure detectors for generalized feature extraction in time-series data is presented in Chapter 3. The implementation of the structure detectors and the methodology employed to identify sequences of primitives in time-series data are explained. Chapter 4 describes the design and outcome of an experiment performed to evaluate the efficacy of the structure detectors for generalized feature extraction. The conclusions of this thesis are discussed in Chapter 5.

Chapter 2

Pattern Recognition

2.1. Introduction

All living organisms must perform different types of identification problems in the course of their existence. For organisms whose main focus is survival, examples of such problems include locating edible food, distinguishing between friend and foe, and seeking shelter that is likely to be free from predators. More complex organisms, such as humans, contend with a wider variety of such problems: locating a desired coin within a handful of change, recognizing relatives among arriving travelers at an airport, and determining the identity of a person by their voice or handwriting. Regardless of the organism, identification problems are resolved by collecting and analyzing sensory information from the environment to discriminate among different populations, groups, or classes. To perform medical diagnosis, for example, a physician assembles information from the environment (i.e., the patient) in order to discriminate among many possible classes (i.e., the conditions, disorders, or diseases that can be diagnosed) based on his knowledge and experience.

2.2. Automated Pattern Recognition Systems

The ability of living organisms to solve identification problems is rooted in their perceptual and cognitive abilities to collect and analyze information from the environment [5][73]. The field of pattern recognition focuses on mechanizing these abilities with the goal of automating the identification process [54][65][91]. In contrast to biological systems, automated pattern recognition systems use algorithms to process data collected either electronically (via monitors or sensors) or transcribed by a human, resulting in an identification of the group of which the data are most representative. Figure 2.1 illustrates the parallel between living organisms and automated pattern recognition systems.

The algorithms used by pattern recognition systems are commonly divided into two tasks, as shown in Figure 2.2. The description task transforms data collected from the environment into *features*—i.e., any value that can be derived from and is representative of the data—which are used in the classification task to arrive at an identification. The description task can involve several different, but interrelated, activities:

- *Preprocessing* is sometimes necessary to modify the data either to correct deficiencies in the data due to limitations of the sensor, or to prepare the data for subsequent activities later in the description task or in the classification task.

- *Feature extraction* is the process of generating features to be used in the classification task. *Elementary features* are explicitly present in the data and can be passed directly to the

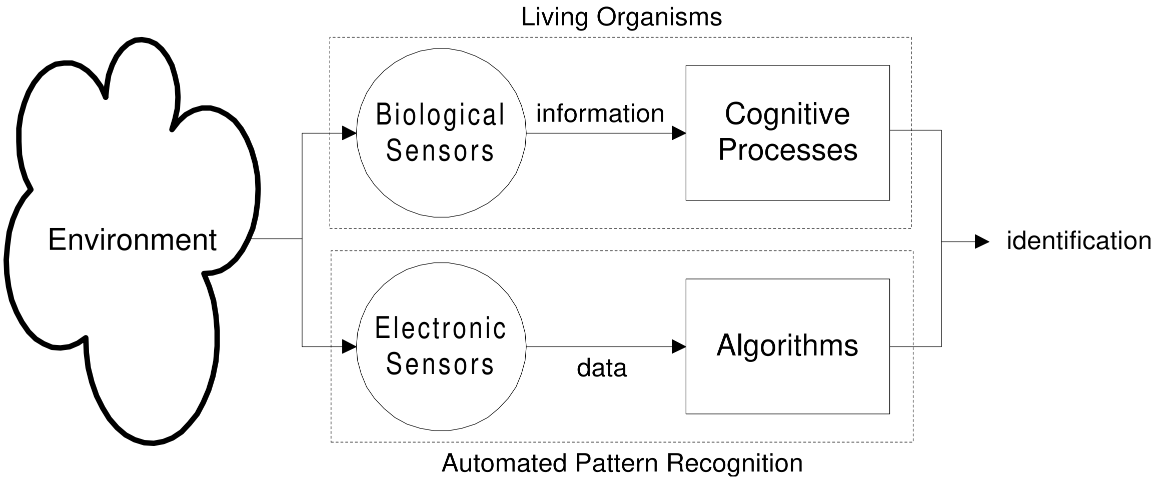


Figure 2.1 The identification problem as solved by living organisms and automated pattern recognition systems. Automated pattern recognition systems deploy electronic sensors and implement algorithms which approximate the functioning of the perceptual and cognitive abilities of living organisms.

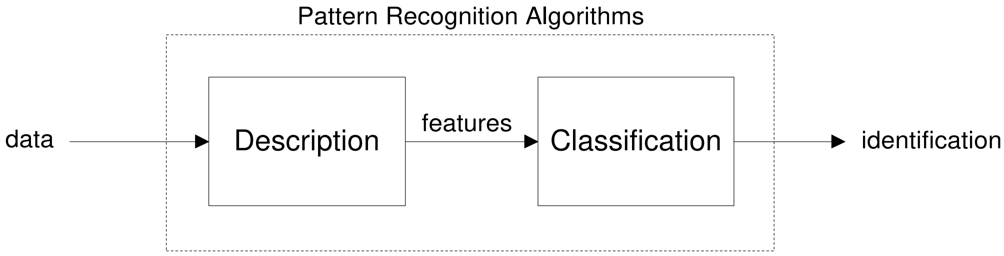


Figure 2.2 The two separate tasks commonly used to divide the algorithms within automated pattern recognition systems. The description task transforms data collected from the environment into features. The classification task arrives at an identification based on the features provided by the description task.

classification task. *Higher-order features* are derived from elementary features and are generated by performing manipulations and/or transformations on the data.

- *Feature selection* reduces the number of features provided to the classification task. Those features which are likely to assist in discrimination are picked out and allowed to be used in the classification task. Features which are not selected are discarded; higher-order features which are determined to be unnecessary for classification can be eliminated from the feature extraction process.

Of these three activities, feature extraction is most critical because the particular features made available for discrimination directly influence the efficacy of the classification task: features which truly discriminate among groups will assist in identification, while the lack of such features can impede the classification task from arriving at an accurate identification. Feature selection, while useful to minimize the feature extraction effort, is often relegated to the classification task so that the usefulness of each feature can be evaluated within the context of the discrimination process.

The end result of the description task is a set of features, commonly called a *feature vector*, which constitutes a representation of the data. The classification task uses a *classifier* to map a feature vector to a group. Such a mapping can be specified by hand or, more commonly, a *training phase* is used to induce the mapping from a collection of feature vectors known to be representative of the various groups among which discrimination is being performed (i.e., the *training set*). Once formulated, the mapping can be used to assign an identification to each unlabeled feature vector subsequently presented to the classifier.

The generality of the description and classification architecture in conjunction with the flexibility afforded by the training phase makes automated pattern recognition systems useful for solving a wide range of real-world problems. Various algorithms can be used for the description and classification tasks to implement a pattern recognition system that is appropriate for a particular domain and application. Different combinations of algorithms have proven to be effective, resulting in two basic approaches to implementing pattern recognition systems.

2.3. Approaches to Pattern Recognition

There are two fundamental approaches to implementing a pattern recognition system: statistical and structural. Each approach employs different techniques to implement the description and classification tasks. Hybrid approaches, sometimes referred to as a unified approach to pattern recognition [35], combine both statistical and structural techniques within a pattern recognition system.

Statistical pattern recognition [24][32][47] draws from established concepts in statistical decision theory to discriminate among data from different groups based upon quantitative features of the data. There are a wide variety of statistical techniques that can be used within the description task for feature extraction, ranging from simple descriptive statistics to complex transformations. Examples of statistical feature extraction techniques include mean and standard deviation computations, frequency count summarizations, Karhunen-Loève transformations, Fourier transformations, wavelet transformations, and Hough transformations. The quantitative features extracted from each object for statistical pattern recognition are organized into a fixed-length feature vector where the meaning associated with each feature is determined by its position within the vector (i.e., the first feature describes a particular characteristic of the data, the second feature describes another characteristic, and so on). The collection of feature vectors generated by the description task are passed to the classification task. Statistical techniques used as classifiers within the classification task include those based on similarity (e.g., template matching, k-nearest neighbor), probability (e.g., Bayes rule), boundaries (e.g., decision trees, neural networks), and clustering (e.g., k-means, hierarchical).

The quantitative nature of statistical pattern recognition makes it difficult to discriminate among groups based on the morphological (i.e., shape-based or structural) subpatterns and their interrelationships embedded within the data. This limitation provided the impetus for the development of a structural approach to pattern recognition that is supported by psychological evidence pertaining to the functioning of human perception and cognition. Object recognition in humans has been demonstrated to involve mental representations of explicit, structure-oriented characteristics of objects [11][48][67][74][75], and human classification decisions have been shown to be made on the basis of the degree of similarity between the extracted features and those of a prototype developed for each group [4][41][72][86][87]. For instance, Biederman [11] proposed the recognition-by-components theory to explain the process of pattern recognition in humans: (1) the object is segmented into separate regions according to edges defined by differences in surface characteristics (e.g., luminance, texture, and color), (2) each segmented region is approximated by a simple geometric shape, and (3) the object is identified based upon the similarity in composition between the geometric representation of the object and the central tendency of each group. This theorized functioning of human perception and cognition serves as the foundation for the structural approach to pattern recognition.

Structural pattern recognition [31][38][70], sometimes referred to as syntactic pattern recognition due to its origins in formal language theory, relies on syntactic grammars to discriminate among data from different groups based upon the morphological interrelationships (or interconnections) present within the data. Structural features, often referred to as *primitives*, represent the subpatterns (or building blocks) and the relationships among them which constitute the data. The semantics associated with each feature are determined by the coding scheme (i.e., the selection of morphologies) used to identify primitives in the data. Feature vectors generated by structural pattern recognition systems contain a variable number of features (one for each primitive extracted from the data) in order to accommodate the presence of superfluous structures which have no impact on classification. Since the interrelationships among the extracted primitives must also be encoded, the feature vector must either include additional features describing the relationships among primitives or take an alternate form, such as a relational graph, that can be parsed by a syntactic grammar.

The emphasis on relationships within data makes a structural approach to pattern recognition most sensible for data which contain an inherent, identifiable organization such as image data (which is organized by location within a visual rendering) and time-series data (which is organized by time); data composed of independent samples of quantitative measurements, such as the Fisher iris data, lack ordering and require a statistical approach.[1] Methodologies used to extract structural features from image data such as morphological image processing techniques [21][33] result in

[1]The Fisher iris data comprise the measurements of the sepal length and width and the petal length and width in centimeters of fifty iris plants for each of three types of iris. These data were originally collected by Anderson [3], but were made famous by Fisher [27]. See Andrews [6] for a more recent discussion of the data.

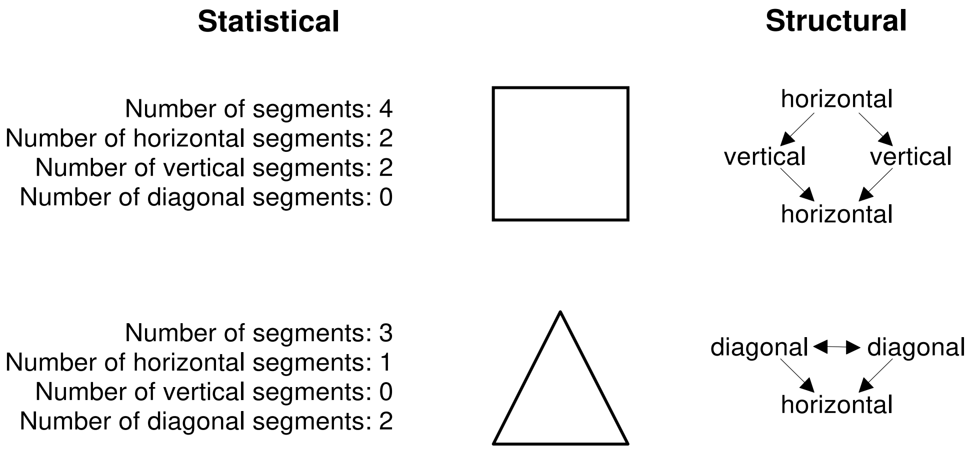


Figure 2.3 The statistical and structural approaches to pattern recognition applied to a common identification problem. The goal is to discriminate between the square and the triangle. A statistical approach extracts quantitative features which are assembled into feature vectors for classification with a decision-theoretic classifier. A structural approach extracts morphological features and their interrelationships, encoding them in relational graphs; classification is performed by parsing the relational graphs with syntactic grammars.

primitives such as edges, curves, and regions; feature extraction techniques for time-series data include chain codes, piecewise-linear regression, and curve fitting which are used to generate primitives that encode sequential, time-ordered relationships. The classification task arrives at an identification using parsing: the extracted structural features are identified as being representative of a particular group if they can be successfully parsed by a syntactic grammar. When discriminating among more than two groups, a syntactic grammar is necessary for each group and the classifier must be extended with an adjudication scheme so as to resolve multiple successful parsings.[2]

Figure 2.3 demonstrates how both approaches can be applied to the same identification problem. The goal is to differentiate between the square and the triangle. A statistical approach extracts quantitative features such as the number of horizontal, vertical, and diagonal segments which are then passed to a decision-theoretic classifier. A structural approach extracts morphological features and their interrelationships within each figure. Using a straight line segment as the elemental morphology, a relational graph is generated and classified by determining the syntactic grammar that can successfully parse the relational graph. In this example, both the statistical and structural approaches would be able to accurately distinguish between the two geometries. In more complex data, however, discriminability is directly influenced by the particular approach employed for pattern recognition because the features extracted represent different characteristics of the data.

[2]The classification task for structural pattern recognition could be more efficiently implemented as a single grammar composed of individual subgrammars where each identifies objects belonging to one particular group; an examination of the resulting parse tree would determine the final identification produced by such a compound grammar. For the purposes of this discussion, a separate grammar is assumed for each group.

	Statistical	Structural
Foundation	Statistical decision theory	Human perception and cognition
Description	Quantitative features	Morphological primitives
	Fixed number of features	Variable number of primitives
	Ignores feature relationships	Captures primitive relationships
	Semantics from feature position	Semantics from primitive encoding
Classification	Statistical classifiers	Parsing with syntactic grammars

Table 2.1 A summary of the differences between statistical and structural approaches to pattern recognition. Due to their divergent theoretical foundations, the two approaches focus on different data characteristics and employ distinctive techniques to implement both the description and classification tasks.

A summary of the differences between statistical and structural approaches to pattern recognition is shown in Table 2.1. The essential dissimilarities are twofold: (1) the description generated by the statistical approach is quantitative, while the structural approach produces a description composed of subpatterns or building blocks; and (2) the statistical approach discriminates based upon numeric differences among features from different groups, while grammars are used by the structural approach to define a "language" encompassing the acceptable configurations of primitives for each group. Hybrid systems can combine the two approaches as a way to compensate for the drawbacks of each approach, while conserving the advantages of each. As a single-level system, structural features can be used with either a statistical or structural classifier. Statistical features cannot be used with a structural classifier because they lack relational information, however statistical information can be associated with structural primitives and used to resolve ambiguities during classification (e.g., as when parsing with attributed grammars) or embedded directly in the classifier itself (e.g., as when parsing with stochastic grammars). Hybrid systems can also combine the two approaches into a multi-level system using a parallel or a hierarchical arrangement.

2.4. Structural Pattern Recognition

Structural approaches, while supported by psychological evidence which suggests that structure-based description and classification parallels that of human perceptual and cognitive processes, have not yet been developed to the fullest potential due to fundamental complications associated with implementing structural pattern recognition systems. Shiavi and Bourne [78, p. 557] summarize the problems of applying structural methods for pattern recognition within the context of analyzing biological waveforms:

> There are obvious problems with the use of [structural techniques]. First, rather deep knowledge about the problem is required in order to successfully identify features and write [grammar] rules. While it is conceptually interesting to consider the possibility of using some automated type of grammatical inference to produce the rules, in practice

> no technique of grammatical inference has proved robust enough to be used with real problems involving biological waveforms. Hence, the writing of rules is incumbent on the designer of the analysis system. Similarly, the selection of features [to extract] must be accomplished essentially by hand since automated techniques usually cannot provide the guidance necessary to make a useful feature selection. Second, the control strategy of typical parsing systems is relatively trivial and cannot deal with very difficult problems. Typical parsing techniques consist of simple repeated application of a list of rules, which is often equivalent to forward chaining, an elementary concept in knowledge-based rule systems. Formation of a robust control strategy for guiding syntactic parsing of strings appears somewhat problematic. However, if rather straightforward amalgamation of constituent elementary tokens in a waveform will suffice to secure an identification or evaluation, then this technique will work rather well.

While Shiavi and Bourne detail several barriers to effectively using a structural approach for pattern recognition, the underlying complication is that both the description and classification tasks must be implemented anew for each unique combination of domain and identification problem which, by the very nature of the techniques used, can require a time-consuming, hand-driven development cycle.

2.4.1. Description

There is no general solution for extracting structural features from data. Pattern recognition texts give scant attention to the topic of primitive selection, most often describing the process as being domain- and application-specific. For example, Friedman [29, p. 243] addresses the issue by saying, "The selection of primitives by which the patterns of interest are going to be described depends upon the type of data and the associated application." Nadler [65, p. 152] seems to support this position when he states, "...features are generally designed by hand, using the experience, intuition, and/or cleverness of the designer."

The lack of a general approach for extracting primitives puts designers of structural pattern recognition systems in an awkward position: feature extractors are necessary to identify primitives in the data, and yet there is no established methodology for deciding which primitives to extract. The result is that feature extractors for structural pattern recognition systems are developed to extract either the simplest and most generic primitives possible, or the domain- and application-specific primitives that best support the subsequent classification task. Some structural pattern recognition systems justify the use of a particular set of feature extractors by claiming that the same set had been used successfully by a previous system developed for a similar application within the same domain; such claims simply shift the burden of feature extractor development onto previously-implemented systems.

Neither of these two philosophies for developing feature extractors for structural pattern recognition is optimal. Simplistic primitives are domain-independent, but capture a minimum of structural information and postpone deeper interpretation until the classification step. At the other extreme, domain- and application-specific primitives can be developed with the assistance of a domain expert, but obtaining and formalizing knowledge from a domain expert, called *knowledge acquisition*, can be problematic.

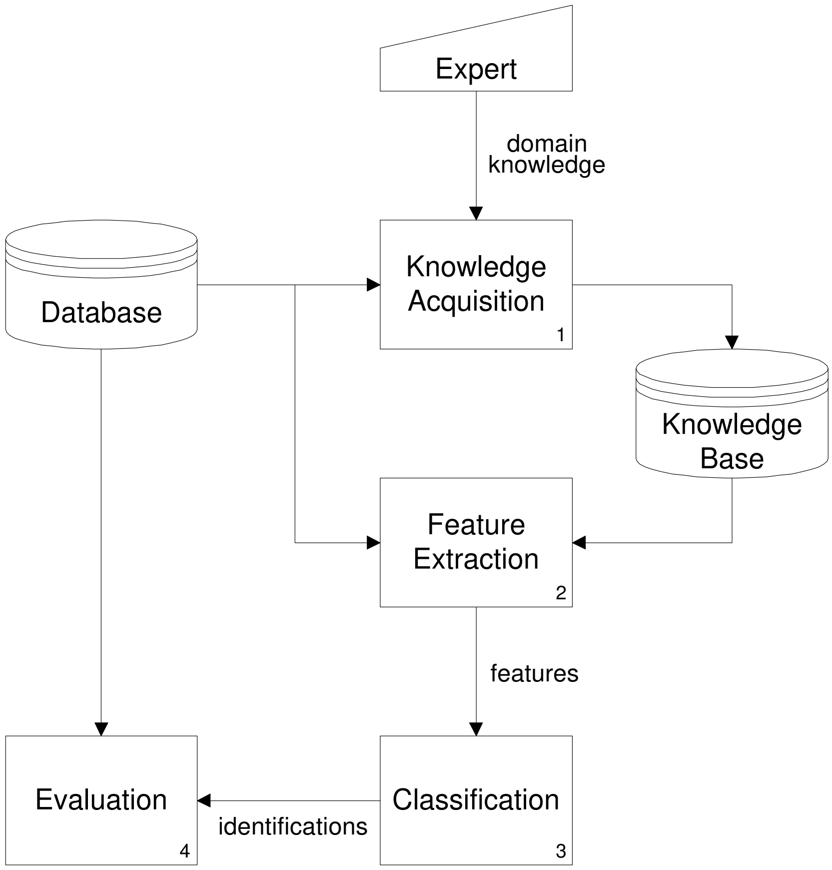


Figure 2.4 The process of knowledge acquisition for developing domain- and application-specific feature extractors for structural pattern recognition. Referring to the numbered activities in the figure, the procedure typically entails several steps: (1) Knowledge acquisition techniques are used to assemble domain knowledge from an expert to produce a knowledge base containing a high-level description of the data. The database contains a collection of labeled data sets and is used to assess the validity of the knowledge base. (2) Domain- and application-specific feature extraction tools are developed using the knowledge base and are applied to the database. (3) Classification is performed with the extracted features. (4) The accuracy of the resulting classification is evaluated. The process iterates until an accurate and robust classification is achieved.

	Knowledge Acquisition Technique			
Domain	Structured Interview	Protocol Analysis	Laddered Grid	Card Sort
Flint Artifacts [16]	33%	11%	33%	33%
Glacier Attributes [15]	33%	16%	35%	23%
Igneous Rocks [14]	28%	8%	28%	30%

Table 2.2 A summary of the efficacy of four knowledge acquisition techniques applied within three different domains to elicit knowledge from experts regarding features useful for classification. For each combination of technique and domain, the average coverage of knowledge elicited from multiple experts is reported.

The process of knowledge acquisition for developing domain- and application-specific feature extractors for structural pattern recognition is shown in Figure 2.4. Referring to the numbered activities in the figure, the procedure typically entails several steps: (1) Knowledge acquisition techniques are used to assemble domain knowledge—information about the database and the environment in which it is generated—from an expert to produce a knowledge base containing a high-level description of the data. The database contains a collection of labeled data sets and is used to assess the validity of the knowledge base. (2) Domain- and application-specific feature extraction tools are developed using the knowledge base and are applied to the database. (3) Classification is performed with the extracted features. (4) The accuracy of the resulting classification is evaluated. If the accuracy of the classification is unacceptable, then the procedure returns to step #1, the domain expert is reconsulted, and the feature extractors are refined. This iterative process continues until the extracted primitives result in an accurate and robust classification.

The success of the knowledge acquisition process hinges on the ability of the domain expert to provide complete, accurate, and consistent domain knowledge. Both manual and computer-based approaches have been developed to elicit domain knowledge from experts. Manual techniques for knowledge acquisition include interviews, task analysis, protocol analysis, card sorting, decision analysis, and graph construction [43]; computer-based techniques consist mainly of automated versions of manual methods and approaches that would be too tedious to perform manually, such as modeling and simulation [12]. Applications of these techniques are most often reported in the literature as case studies, leaving efficacy a matter of anecdote rather than experimental evaluation.

One study, however, has been undertaken to compare rigorously the efficacy of four knowledge acquisition techniques: structured interview (a designed and ordered set of domain-specific questions is answered by the expert), protocol analysis (the expert "thinks aloud" while classifying domain objects), laddered grid (a conceptual graph representing the relationships among domain elements is developed by the expert), and card sort (the expert repeatedly sorts domain objects or cards representing domain elements and describes the rationale behind each unique sorting). Table 2.2 summarizes the efficacy of each of these knowledge acquisition techniques when used to extract features for classification [14][15][16]. Within a domain, each technique was used to elicit knowledge from multiple experts, coverage was computed for each expert (i.e., the percentage of knowledge elicited as compared to a complete knowledge base), and the average coverage for each technique was reported as a measure of its efficacy. As shown in Table 2.2, the overall coverage of elicited knowledge was poor, where the most effective techniques achieved a coverage between 25 and 35 percent (pairwise combinations of these techniques were able to achieve a coverage between 35 and 45 percent). These findings are echoed in a survey of around seventy case studies describing the development of industrial expert systems: almost half reported problems with the quality of knowledge elicited from the expert [18]. This phenomenon, dubbed the *knowledge acquisition bottleneck*, was identified twenty years ago and is still being cited as a major flaw in techniques used for knowledge acquisition [26][37][57].

Using knowledge acquisition techniques to obtain domain knowledge from an expert has two major drawbacks: (1) it is a time-consuming process, and (2) the resulting knowledge base is likely to be incomplete. Developing domain- and application-specific feature extractors, therefore, can be burdensome and will not necessarily produce a robust set of features for classification. As a result, the effort needed to implement the description task for a new domain and application can act as a barrier to using structural approaches to pattern recognition.

2.4.2. Classification

The classifier for a structural pattern recognition system is composed of a set of syntactic grammars, one for each group among which discrimination is being performed, and a parser. The identification generated by the classifier is the group whose associated syntactic grammar successfully parses the primitives extracted from the data. An adjudication scheme is necessary to resolve the situation where there is more than one successful parse.

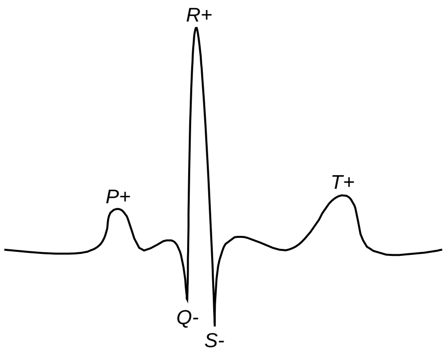
The main difficulty in developing the classifier for a structural pattern recognition system lies in constructing the syntactic grammars. Since the grammars embody the precise criteria which differentiate among the groups, they are by their very nature domain- and application-specific. Complicating matters is the lack of a general solution for extracting structural features from data, causing the primitives used within the grammars to vary among domains, identification problems, and pattern recognition systems. Grammar inference techniques can be used to construct automatically a grammar from examples, but these methods can fail in the most general cases such as when the grammar for a group is context free.

Existing structural pattern recognition systems are typically applied to domains where the grammars required for discrimination can be constructed by hand. For example, structural pattern recognition systems developed for electrocardiogram diagnosis [53][85] routinely use hand-tooled grammars because the domain knowledge is extensive, the primitives are distinct, and the relationships among the primitives are well defined. This approach, however, quickly becomes tedious, unmanageable, and error prone as the complexities of the domain and identification problem increase. Moreover, specification of grammars by hand is impractical for domains that are complex, and infeasible for domains that are poorly understood.

2.5. Applications to Time-Series Data

Identification problems involving time-series (or waveform) data constitute a subset of pattern recognition applications that is of particular interest because of the large number of domains that involve such data [25][83]. Time-series data could be treated as if it were image data by coverting the time-series data into a visual rendering and applying image-based segmentation techniques.

Normal

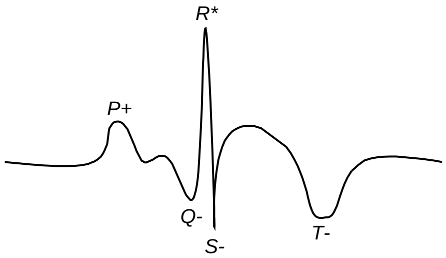


Parse String: P+ Q- R+ S- T+

Grammar:

<NORMAL ECG> → <P><QRS><T>
<P> → <P+>
<QRS> → <Q><R><S>
<Q> → <Q->
<R> → <R+>
<S> → <S->
<T> → <T+>
<P+> : positive wave, expected shape
<Q-> : negative wave, expected shape
<R+> : positive wave, expected shape
<R*> : positive wave, invalid amplitude
<S-> : negative wave, expected shape
<T+> : positive wave, expected shape
<T-> : negative wave, expected shape

Myocardial Infarction



Parse String: P+ Q- R* S- T-

Grammar:

<MYO INF ECG> → <P><QRS><T>
<P> → <P+>
<QRS> → <Q><R><S>
<Q> → <Q->
<R> → <R*>
<S> → <S->
<T> → <T->
<P+> : positive wave, expected shape
<Q-> : negative wave, expected shape
<R+> : positive wave, expected shape
<R*> : positive wave, invalid amplitude
<S-> : negative wave, expected shape
<T+> : positive wave, expected shape
<T-> : negative wave, expected shape

Figure 2.5 An example of structural pattern recognition applied to time-series data for electrocardiogram diagnosis. Each waveform traces the electrical activity recorded during one cardiac cycle (i.e., heartbeat) by a single electrode: the leftmost represents data recorded during a normal heartbeat, and the rightmost represents data recorded during a heartbeat exhibiting behavior indicative of a cardiac condition called myocardial infarction. The primitives extracted from each data set are labeled on the waveforms. Concatenating the primitives by time constitutes a parse string, and a context-free grammar can be constructed to parse, and hence classify, each string of primitives.

Such an approach, however, would ignore the inherent, time-based sequencing and unnecessarily complicate feature extraction by converting one-dimensional time-series data into two-dimensional image data. Feature extractors for structural pattern recognition in time-series data identify time-based subpatterns that manifest across consecutive data points within a time series.

Figure 2.5 illustrates an example of structural pattern recognition applied to time-series data for electrocardiogram diagnosis. Each waveform traces the electrical activity recorded during one cardiac cycle (i.e., heartbeat) by a single electrode: the leftmost represents data recorded during a normal heartbeat, and the rightmost represents data recorded during a heartbeat exhibiting behavior indicative of a cardiac condition called myocardial infarction.[3] Primitives in the data are extracted using a morphological coding scheme that characterizes the peaks in the data according to their shape and location. The extracted primitives are labeled on each waveform. Concatenating the primitives for each data set according to time of appearance constitutes a parse string, and a context-free grammar can be constructed to parse, and hence classify, each string of primitives. An example grammar for each waveform is shown. Notice that the grammar for the normal electrocardiogram can not successfully parse the string of primitives extracted from the waveform exhibiting myocardial infarction and vice versa. Other domains involving time-series data where structural pattern recognition is similarly applied include speech recognition, seismic activity identification, radar signal detection, and process control.

Both statistical and structural approaches can be used for pattern recognition of time-series data: standard statistical techniques have been established for discriminant analysis of time-series data [79], and structural techniques have been shown to be effective in a variety of domains involving time-series data [30]. Examples of structural pattern recognition systems developed for classification of time-series data include the following:

- Stockman and Kanal [84] used a structural approach to develop a waveform parsing system, called WAPSYS, and demonstrated its usefulness at classifying pulse data. Feature extractors were developed with the assistance of medical personnel: curve fitting was used to identify instances of parabolic and straight-line primitives. A context-free grammar called a problem reduction representation (PRR) was developed under a construct-test-accept revision cycle and used for classification. Acknowledging the limitations of WAPSYS, the authors state that "the user must define the structure of his domain in terms of the PRR and primitives and [must] include knowledge about which structures are best recognized first. He may also need to splice problem-specific semantic routines into the system" [84, p. 297].

- Another application of structural approaches to pattern recognition is to monitor sensor data for process control and fault diagnosis. Rengaswamy and Venkatasubramanian [76] describe a system constructed to perform fault diagnosis of a critical process within a refinery. Parabolic and straight-line primitives were selected as features based on earlier work; a grammar capturing the knowledge of a process engineer was used for classification. Love and Simaan [58] designed a structural pattern recognition system to monitor an aluminum strip rolling mill. Visual inspection of the data led to the selection of straight-line primitives, where the slope was used to categorize each extracted primitive into one of four subclasses:

[3]Myocardial infarction, commonly referred to as a heart attack, is the death of an area of heart muscle due to a sudden reduction in blood flow relative to the amount of work the heart is doing at the time.

impulses (i.e., sharp peaks), edges (i.e., sudden increases or decreases), flats (i.e., absence of increases or decreases), and ramps (i.e., linear increases or decreases). Classification was performed using rule-based inference.

- Structural approaches to pattern recognition are commonly used to implement systems for electrocardiogram (ECG) diagnosis [88][81]. Trahanias and Skordalakis [85] developed a structural pattern recognition system for ECG data which extracted instances of straight-line segments, peaks, and parabolas as primitives. The choice of these primitives was rationalized as "a natural one because the complexes are composed of peaks and the segments have the shape of a straight line or a parabola" [85, p. 649]. Attribute grammars developed with domain knowledge were used for classification. Koski, Juhola, and Meriste [53] based the implementation of their structural pattern recognition system for ECG data on techniques successfully used in previous systems: straight-line primitives were extracted and parsed with attributed automata (i.e., finite state machines with scalar variables associated with the states to assist in determining state transitions).

As evidenced by these systems, structural approaches to pattern recognition can be used to solve certain identification problems; however, implementation relies on domain knowledge either provided by an expert or assimilated from the data by the system designer. Some structural pattern recognition systems stray from traditional implementation techniques so as to simplify the development process and/or to generalize the applicability of the system to other identification problems. Examples of such systems include the following:

- The curve fitters used by Stockman and Kanal [84] in WAPSYS allowed different sets of primitives to be defined by varying the constraints placed on the slope, curvature, and length of the extracted parabolic and straight-line features.

- Konstantinov and Yoshida [52] proposed a generic methodology for analyzing time-series data and demonstrated its efficacy at monitoring an amino acid production process. Their system incorporated an expandable library of primitives so that new application-specific entries could be defined. Primitives in the library included standard ones such as straight lines and parabolas, while others described irregular shapes such as "IncreasingConcavely-Convexly" and "StartedToDecrease."

- Keogh and Pazzani [51] developed a structural pattern recognition system for analyzing telemetry data from the space shuttle. Straight-line segments generated by a specialized piecewise-linear segmentation algorithm were used as primitives based upon their success and usefulness for other identification problems. A new statistical classification algorithm, called CTC (Cluster, then Classify), based on distance measurements was used for classification.

- A structural pattern recognition system for the on-line identification of handwritten Chinese characters was implemented by Kuroda, Harada, and Hagiwara [55]. While many character recognition systems analyze images of handwritten text, time-series data composed of the loci of vertical points, horizontal points, and pen pressure were collected as the characters were written. Feature extraction was accomplished using peak segmentation. A regular grammar encoded as a matrix was learned from training data for each class; the set of matrices was then used for classification.

	Structural Features		
	Straight Lines	Parabolas	Peaks
Stockman *et al.* [84]	$\surd$	$\surd$	
Rengaswamy *et al.* [76]	$\surd$	$\surd$	
Love *et al.* [58]	$\surd$		
Trahanias *et al.* [85]	$\surd$	$\surd$	$\surd$
Koski *et al.* [53]	$\surd$		
Konstantinov *et al.* [52]	$\surd$	$\surd$	
Keogh *et al.* [51]	$\surd$		
Kuroda *et al.* [55]			$\surd$

Table 2.3 A summary of the structural features extracted by some structural pattern recognition systems. Each column is associated with a particular primitive type. Each row reports the types of primitives extracted by a system with a check in the appropriate columns.

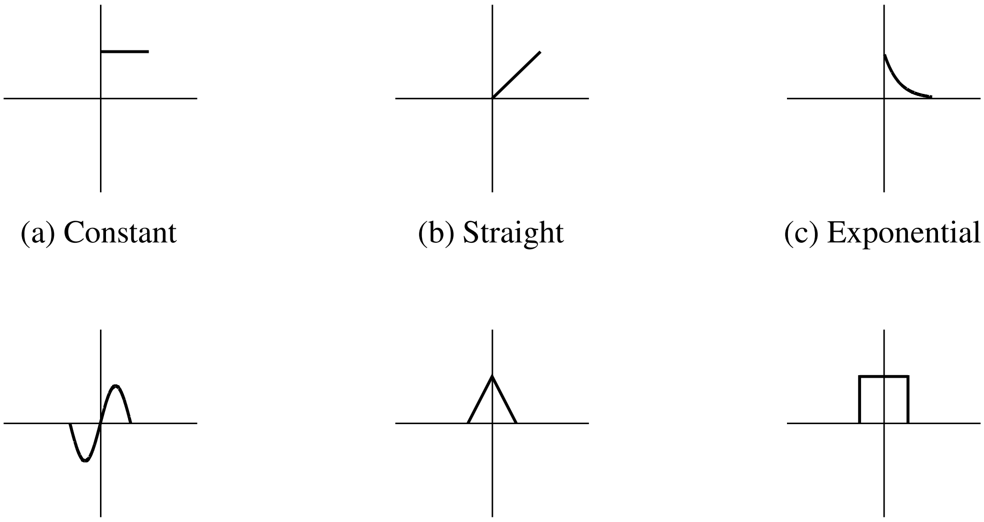
Of these pattern recognitions systems, two trends towards generality are apparent: (1) a broadening of the selection of primitives to permit the inclusion of other structural features that may be effective for other identification problems, and (2) the substitution of syntactic grammars by statistical classification techniques so as to simplify the development of the classifier by eliminating hand-constructed grammars and the need for a separate parser. These trends make structural pattern recognition systems better suited for a wider variety of identification problems, but the continued reliance on domain knowledge to define primitives persists in hindering structural approaches to pattern recognition.

The types of primitives extracted by each of the structural pattern recognition system discussed in this section are summarized in Table 2.3. Notice that almost all systems rely on straight-line primitives, probably due to the ease in which they are identified and their versatility in approximating arbitrary curves. Parabolas and peaks are used less often than straight-line primitives, being included when appropriate for the domain under analysis.

2.6. Domain-Independent Structural Pattern Recognition

A domain-independent structural pattern recognition system is one that is capable of acting as a "black box" to extract primitives and perform classification without the need for domain knowledge. While it may be possible to achieve more accurate results by using domain-dependent techniques, a domain-independent structural pattern recognition system would be advantageous for preliminary data exploration, particularly in complex, poorly-understood domains where knowledge acquisition would be unacceptably lengthy or infeasible. Moreover, a domain-independent structural pattern recognition system could be used to generate a first pass at a set of feature extractors, thereby laying the groundwork for construction of a domain- and application-specific structural pattern recognition system.

A domain-independent structural pattern recognition system for time-series data must incorporate techniques for the description and classification tasks that are not dependent on domain



(a) Constant (b) Straight (c) Exponential

(d) Sinusoidal (e) Triangular (f) Rectangular

Figure 2.6 Types of modulation commonly used by signal processing systems to transmit information via a continuous medium (e.g., an electrical current) include (a) constant, (b) straight, (c) exponential, (d) sinusoidal, (e) triangular, and (f) rectangular.

knowledge. Since syntactic grammars are inherently tied to the domain and application, a natural solution is a hybrid system that employs a statistical classifier to perform discrimination based on structural features extracted from the data. While syntactic grammars are capable of analyzing the variable-length feature vectors generated by structural feature extractors, statistical classifiers require fixed-length feature vectors. This requirement can be satisfied with a separate training phase for the description task so as to determine a fixed number of primitives to be generated by the structural feature extractors. Additionally, the structural features extracted from a time-series data set can be ordered in the associated feature vector according to their linear sequence in the data, thereby encoding the relationships among the primitives.

A popular solution employed by the non-traditional structural pattern recognition systems described in Section 2.5 for generalizing feature extraction is to allow for the inclusion of new primitives. While a dynamic set of primitives can be useful, this solution still requires domain knowledge to define new primitives to add. What is needed is a collection of primitives such that each is generally useful and is easy to extract from time-series data. While a sequence of straight-line segments can be used to approximate any waveform, more complex morphologies identified within a waveform can result in a closer approximation to the waveform, contribute more structural information per extracted feature, reduce the complexity of the classification task, and provide a better foundation for interpretation by humans. As summarized in Table 2.3, straight-line segments, parabolas, and peaks have proven to be useful primitives and, therefore, the description task for time-series data must minimally include feature extractors for these primitives.

The field of signal processing offers a suggestion for additional, more complex primitives for time-series data. Signal processing systems are designed to transmit information via a continuous

	Modulation Types					
	Constant	Straight	Exponential	Sinusoidal	Triangular	Rectangular
Stockman *et al.* [84]	√	√		√		
Rengaswamy *et al.* [76]	√	√		√		
Love *et al.* [58]	√	√				
Trahanias *et al.* [85]	√	√		√	√	
Koski *et al.* [53]	√	√				
Konstantinov *et al.* [52]	√	√		√		
Keogh *et al.* [51]	√	√				
Kuroda *et al.* [55]					√	
Generalized	√	√	√	√	√	√
	Straight Lines			Parabolas	Peaks	
	Structural Features					

Table 2.4 The structural features extracted by the pattern recognition systems listed in Table 2.3 recast as the set of modulation types used in signal processing. The relationship is as follows: the constant and straight modulation types approximate straight-line segments, the sinusoidal modulation type approximates parabolas, and the triangular modulation type approximates peaks. Given this mapping, a check appears in the columns associated with the types of modulation extracted by each system. The last row indicates the modulation types that generalized feature extraction would identify.

medium (e.g., an electrical current) and subsequently reconstruct it. The information is transmitted by modulation of the medium: the transmitter encodes the information as a sequence of modulations, and the receiver decodes the modulations as information. Figure 2.6 shows the six fundamental types of modulation commonly used in signal processing systems: constant, straight, exponential, sinusoidal, triangular, and rectangular [7][9][20][62]. These six modulation types entail morphologies deliberately introduced into time-series data (i.e., the continuous medium) with the intent of conveying information regardless of domain or application. Since the goal of description for domain-independent structural pattern recognition is to extract primitives which represent meaningful structural features in time-series data without relying on domain knowledge, these six modulation types constitute a potentially-useful set of primitives to extract.

The modulation types used in signal processing provide a set of morphologies that can approximate the small set of primitives commonly extracted by structural pattern recognition systems:

straight lines are used outright, peaks can be approximated by a triangle, and parabolas can be approximated by one-half period of a sine curve. Additionally, the modulation types include morphologies not commonly extracted by structural pattern recognition systems. Table 2.4 shows the relationship between the set of modulation types and the structural features listed in Table 2.3. The six modulation types constitute a reasonable foundation for generalized feature extraction for domain-independent structural pattern recognition in time-series data because they constitute a superset of those morphologies extracted by existing structural pattern recognition systems and suggest additional morphologies not already commonly extracted by such systems. The last row in Table 2.4 indicates the modulation types that generalized feature extraction would identify.

2.7. Discussion

A domain-independent structural pattern recognition system is one that is capable of solving an identification problem regardless of domain or application. To achieve this flexibility, techniques that do not require domain knowledge in the development process must be used to implement both the description and classification tasks. An architecture that does not rely on domain knowledge is a hybrid approach that combines feature extractors that identify domain- and application-independent morphological features with a statistical classifier. A set of six morphologies which have proven to be useful for signal processing and subsumes those morphologies extracted by existing structural pattern recognition systems can serve as a foundation for generalized feature extraction in time-series data. In order to realize domain-independent structural pattern recognition, feature extractors which identify these morphologies in time-series data must first be developed.

Chapter 3

Generalized Feature Extraction

3.1. Introduction

Generalized feature extraction relies on a collection of feature extractors that function independently of domain and application. For time-series data, such feature extractors must be able to identify generally-useful structures that emerge from the relationships between consecutive measurement values over time. A suite of suitable feature extractors for time-series data would be characterized by its ability to

- capture fundamental trends and relationships,
- generate accurate approximations,
- represent the extracted structures compactly,
- support subsequent classification, and
- be domain independent.

Section 2.6 discussed both the structural pattern recognition literature and the field of signal processing as sources of structure types commonly identified in time-series data. A preliminary set of feature extractors for generalized feature extraction in time-series data is one that can identify instances of structure types that have support from both fields. The signal processing and the structural pattern recognition literatures suggest that a set of six structure types—constant, straight, exponential, sinusoidal, triangular, and rectangular—would be useful for structural pattern recognition in time-series data. As such, these six structures will serve as the foundation for a set of feature extractors, called structure detectors, for generalized feature extraction in time-series data. Before a formal description of the structure detectors can be offered, however, a more rigorous definition of time-series data is necessary so as to provide the context for describing the structure detectors.

3.2. Time-Series Data

A time-series data set $\boldsymbol{X}$ is an ordered sequence $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}$ where $\boldsymbol{X}_t$ is a vector of observations (i.e., measurements) recorded during epoch t such that the observations in $\boldsymbol{X}_t$ were recorded previous to those in $\boldsymbol{X}_{t+1}$ for all $1 \leq t < n$. Each vector $\boldsymbol{X}_t$ is composed of m measurements $\{d_t^1, d_t^2, \ldots, d_t^m\}$ recorded during epoch t. A transposition of the time-series data set $\boldsymbol{X}$ results in a corresponding data set $\boldsymbol{Y}$ which is a collection of vectors $\{\boldsymbol{Y}^1, \boldsymbol{Y}^2, \ldots, \boldsymbol{Y}^m\}$ where $\boldsymbol{Y}^k$ contains the time-series $\{d_1^k, d_2^k, \ldots, d_n^k\}$ for all $1 \le k \le m$.

Given this definition, a time-series data set has the matrix form

$$\begin{array}{cccccc} & \boldsymbol{Y}^1 & \boldsymbol{Y}^2 & \boldsymbol{Y}^3 & \cdots & \boldsymbol{Y}^m \\ \boldsymbol{X}_1 & d_1^1 & d_1^2 & d_1^3 & \cdots & d_1^m \\ \boldsymbol{X}_2 & d_2^1 & d_2^2 & d_2^3 & \cdots & d_2^m \\ \boldsymbol{X}_3 & d_3^1 & d_3^2 & d_3^3 & \cdots & d_3^m \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{X}_n & d_n^1 & d_n^2 & d_n^3 & \cdots & d_n^m \end{array}$$

where each row comprises the observations for one particular epoch (i.e., row t contains $\boldsymbol{X}_t$) and each column comprises the observations for one particular measurement (i.e., column k contains $\boldsymbol{Y}^k$). Note the difference between $\boldsymbol{X}$ and $\boldsymbol{Y}$: $\boldsymbol{X}$ is an ordered sequence of n vectors each having dimension m, while $\boldsymbol{Y}$ is an unordered collection of m one-dimensional time-series vectors each of length n.

Since each value d_t^k is associated with a particular time t, $\boldsymbol{Y}^k$ can be rewritten in the equivalent functional form

$$Y^k(t) = d_t^k$$

The functional form $Y^k(t)$, therefore, can be used to refer to the time-series vector $\boldsymbol{Y}^k$. Additionally, each structure detector is univariate in nature and can only be applied to a single, one-dimensional time-series vector, thereby making the k superscript unnecessary. Thus, $Y(t)$ will be used to denote any one-dimensional time-series vector or data set.

3.3. Structure Detectors

A structure detector identifies a particular structure in time-series data and generates a new time series containing values which describe the identified structure. That is, the input to the structure detector is a time series of raw data, and the output is another, newly-generated time series that contains the structure extracted from the input time-series data. To perform this transformation, a structure detector implements a function f such that

$$f(Y(t)) = \hat{Y}(t)$$

where $\hat{Y}(t)$ is a time series which constitutes the structure extracted from $Y(t)$. The function f is fitted to $Y(t)$ so that $\hat{Y}(t)$ approximates $Y(t)$. The sum of squared error E is a measure of how closely $\hat{Y}(t)$ approximates $Y(t)$ and is defined as

$$E = \sum_{t=1}^{n} (Y(t) - \hat{Y}(t))^2$$

If $\hat{Y}(t)$ is equal to $Y(t)$ for $1 \leq t \leq n$, then E is equal to zero; as the deviation of $\hat{Y}(t)$ from $Y(t)$ increases, the value of E increases. Since $\hat{Y}(t)$ can differ arbitrarily from $Y(t)$, there is no upper bound to the value of E.

Each structure detector implements a unique function f which extracts one of the modulation types shown in Figure 2.6. Note, however, that the structure detector associated with the rectangular modulation type was generalized to extract trapezoidal modulation so as to increase its descriptive power. The function f is dependent on t as well as one or more free parameters: the function f is fitted to the input time series by setting the values of the free parameters so as to minimize the sum of squared error. Once an instantiation of the free parameters in f has been determined, f is applied to $Y(t)$ to generate $\hat{Y}(t)$.

The functions f implemented by the structure detectors fall into two categories: those that extract instances of linear structures, and those that extract instances of nonlinear structures. Values for the free parameters in the functions f that extract instances of linear structures (i.e., constant and straight) can be computed directly from the input time-series data. Values for the free parameters in the functions f that extract instances of nonlinear structures (i.e., exponential, sinusoidal, triangular, and trapezoidal) are determined by searching the space of all possible combinations of values for an instantiation that minimizes the sum of squared error. Consequently, structure detectors which extract instances of linear structures are implemented differently than structure detectors which extract instances of nonlinear structures.

3.3.1. Linear Structure Detectors

Structure detectors that extract linear structures (i.e., constant and straight) from time-series data use linear regression techniques to compute the values of the free parameters in the functions f. Linear regression is an established statistical method for identifying a linear relationship between two variables [23][42][45]. In this case, it is the linear relationship between t and $Y(t)$ that determines the values of the free parameters in the functions f.

Constant

The constant structure detector extracts a linear relationship between t and $Y(t)$ such that the value of $\hat{Y}(t)$ is invariant with respect to t. The constant structure detector implements the function

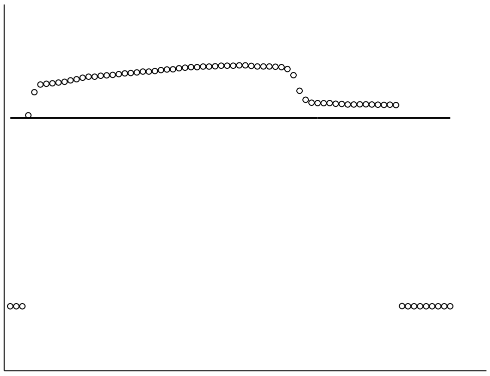
$$f(Y(t)) = a$$

where the value of the free parameter a is computed from $Y(t)$ using the standard linear regression equation

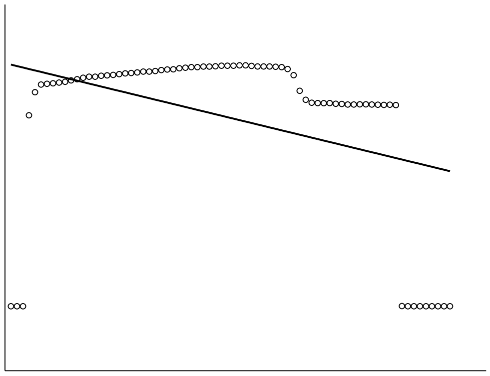
$$a = \frac{1}{n}\sum_{t=1}^{n} Y(t)$$

Computing the value of a in this manner minimizes the sum of squared error and, consequently, generates a $\hat{Y}(t)$ that is the best constant linear approximation of $Y(t)$.

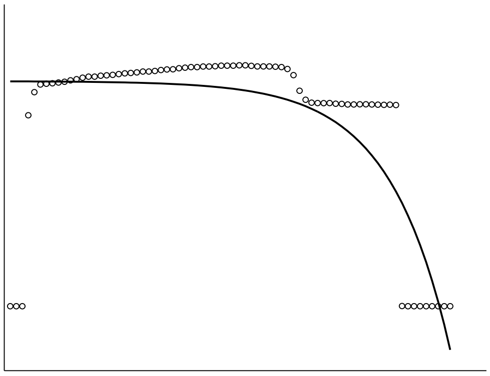
Figure 3.1(a) shows an example of a constant structure extracted from a time-series data set: the hollow bullets represent the input time-series data (i.e., $Y(t)$), and the solid line represents the extracted structure (i.e., $\hat{Y}(t)$). The value of E is equal to the sum of the squared distances between the hollow bullets and the line. By design, $\hat{Y}(t)$ minimizes the value of E and, consequently, no transformation could be performed on the extracted structure to decrease the sum of squared error.



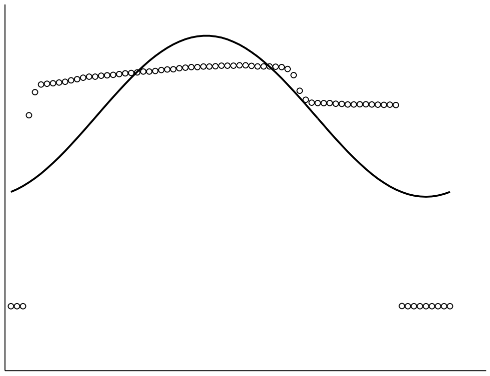
(a) Constant



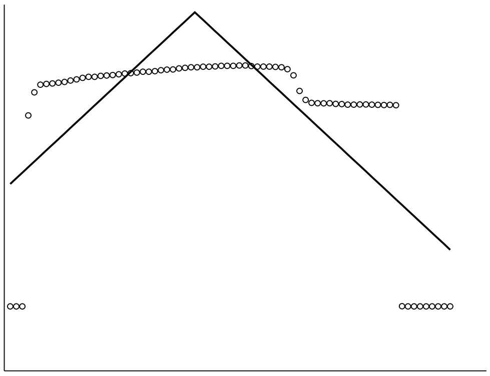
(b) Straight



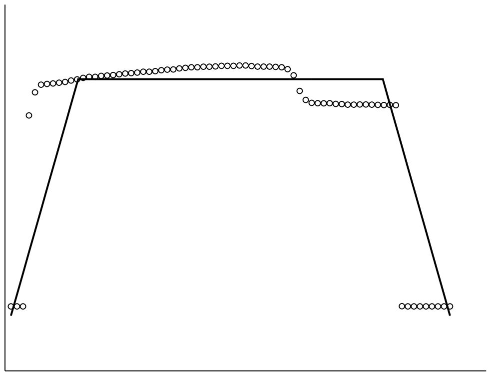
(c) Exponential



(d) Sinusoidal



(e) Triangular



(f) Trapezoidal

Figure 3.1 The six structures fitted to a common data set. Each graph shows the input time-series data plotted with hollow bullets overlaid with a solid line representing one of the six structures. Each structure is fitted to the data so as to minimize the sum of squared error. The structures ordered from best to worst fit (i.e., from smallest to largest sum of squared error) are (f) trapezoidal, (e) triangular, (c) exponential, (d) sinusoidal, (b) straight, and (a) constant.

Straight

The straight structure detector identifies an unconstrained linear relationship between t and $Y(t)$ with the function

$$f(Y(t)) = a + b * t$$

where a and b are free parameters. The values of a and b are computed from $Y(t)$ using the standard linear regression equations

$$b = \frac{\sum_{t=1}^{n}(t - \bar{t}\,)(Y(t) - \bar{y})}{\sum_{t=1}^{n}(t - \bar{t}\,)^2}$$

$$a = \bar{y} - b * \bar{t}$$

where

$$\bar{y} = \frac{1}{n}\sum_{t=1}^{n} Y(t)$$

$$\bar{t} = \frac{1}{n}\sum_{t=1}^{n} t$$

These equations compute the values of a and b so as to minimize the sum of squared error. Therefore, the $\hat{Y}(t)$ generated by the fitted function f will be the best unconstrained linear approximation of $Y(t)$.

Figure 3.1(b) shows an example of a straight structure extracted from a time-series data set: the hollow bullets represent the input time-series data (i.e., $Y(t)$), and the solid line represents the extracted structure (i.e., $\hat{Y}(t)$). The straight structure detector generates $\hat{Y}(t)$ so as to minimize the value of E.

3.3.2. Nonlinear Structure Detectors

Structure detectors that extract nonlinear structures (i.e., exponential, sinusoidal, triangular, and trapezoidal) from time-series data must be implemented differently than linear structure detectors. Linear regression techniques, which serve as the foundation for linear structure detectors, only identify linear relationships among variables and, consequently, can not be used as the basis for nonlinear structure detectors. Since no statistical methodology has been developed to fit an arbitrary nonlinear function f directly from the data, a search must be performed within the space of all possible combinations of values for the free parameters in f to find an instantiation that minimizes the sum of squared error.

While there are many methodologies for optimizing nonlinear functions [46], a commonly-used search strategy that is simple to implement and quick to converge is the simplex method.[1]

[1] The simplex method for function optimization and the simplex method for linear programming both makes use of the geometrical concept of a simplex, however the two algorithms are unrelated and should not be confused.

Simplex search, originally introduced by Spendley, Hext, and Himsworth [82] and generalized by Nelder and Mead [66], is a direct search method in that the search is guided by evaluating the target function with various combinations of values of the free parameters in the function (i.e., no derivative information is used). The Nelder-Mead simplex method moves a geometric shape, called a simplex, through the search space using a set of well-defined transformation operations called reflection, expansion, and contraction [89][90]. Each operation moves one or more of the vertices of the simplex so as to relocate the volume of the simplex closer to the optimal value of the target function; a series of operations is applied to an initial simplex until the simplex has converged on an optimum. No general convergence properties of the original Nelder-Mead simplex search strategy have been proven, but some limited proofs of convergence have been published [56][61].

A generally-available Nelder-Mead simplex search algorithm [71] is used to fit the functions f in the nonlinear structure detectors to the input time-series data. The sum of squared error E is the target function, and the search returns the combination of values for the free parameters in f which minimizes the sum of squared error. There is no guarantee that the simplex search algorithm will fit the function f so that the sum of squared error is the absolute minimum, but suboptimal fits of f have proven to have negligible effects.

Exponential

The exponential structure detector identifies such a relationship between t and $Y(t)$ with the function

$$f(Y(t)) = a * |b|^t + c$$

where a, b, and c are free parameters and control the scale, degree of curvature, and vertical position, respectively, of the extracted exponential structure. The absolute value of b is necessary because a negative value of b would make f discontinuous for incremental integer values of t.

Figure 3.1(c) shows an example of an exponential structure extracted from a time-series data set: the hollow bullets represent the input time-series data (i.e., $Y(t)$), and the solid line represents the extracted structure (i.e., $\hat{Y}(t)$). The exponential structure detector uses simplex search to find values for the free parameters in f so that the $\hat{Y}(t)$ generated by f minimizes the value of E.

Sinusoidal

The sinusoidal structure detector identifies such a relationship between t and $Y(t)$ with the function

$$f(Y(t)) = a * \sin(t + b) + c$$

where a, b, and c are free parameters and control the amplitude, period offset, and vertical position, respectively, of the extracted sinusoidal structure. The sinusoidal structure detector is constrained to fit exactly one period of a sine curve to the input time-series data set. This constraint could be eliminated with an additional free parameter in the function f to control the number of periods in the extracted sinusoidal structure, but at the expense of increasing the complexity of the simplex search.

Figure 3.1(d) shows an example of a sinusoidal structure extracted from a time-series data set: the hollow bullets represent the input time-series data (i.e., $Y(t)$), and the solid line represents the extracted structure (i.e., $\hat{Y}(t)$). The sinusoidal structure detector uses simplex search to find values for the free parameters in f so that the $\hat{Y}(t)$ generated by f minimizes the value of E.

Triangular

The triangular structure detector identifies such a relationship between t and $Y(t)$ with the function

$$f(Y(t)) = \begin{cases} a + b * t & t \leq c \\ (a + 2 * b * c) - (b * t) & t \geq c \end{cases}$$

where a, b, and c are free parameters. The value of c controls the location of the peak: values of t less than or equal to c constitute the leading edge, while values of t greater than or equal to c constitute the trailing edge. Note that the peak itself (i.e., when t is equal to c) is a member of both the leading and trailing edges. The values of a and b control the placement of the line for the leading edge; the trailing edge is constrained to be a line with the negative slope of the leading edge and positioned so as to intersect the leading edge when t is equal to c. These constraints could be eliminated with additional free parameters in the function f, but at the expense of increasing the complexity of the simplex search.

Figure 3.1(e) shows an example of a triangular structure extracted from a time-series data set: the hollow bullets represent the input time-series data (i.e., $Y(t)$), and the solid line represents the extracted structure (i.e., $\hat{Y}(t)$). The triangular structure detector uses simplex search to find values for the free parameters in f so that the $\hat{Y}(t)$ generated by f minimizes the value of E.

Trapezoidal

The trapezoidal structure detector identifies such a relationship between t and $Y(t)$ with the function

$$f(Y(t)) = \begin{cases} a + b * t & t \leq c_{start} \\ a + b * c_{start} & c_{start} \leq t \leq c_{stop} \\ (a + b * c_{start} + b * c_{stop}) - (b * t) & t \geq c_{stop} \end{cases}$$

where a and b are free parameters. A hidden free parameter c controls the length of the horizontal line that forms the top of the trapezoidal structure. The parameters c_{start} and c_{stop} are linear transformations of c where

$$c_{start} = \frac{1}{2}(n - c)$$

$$c_{stop} = n - \frac{1}{2}(n - c) + 1$$

The values of c_{start} and c_{stop} control the onset and offset of the top of the trapezoid structure: values of t less than or equal to c_{start} constitute the leading edge, values of t between c_{start} and c_{stop} inclusive constitute the top of the trapezoidal structure, and values of t greater than or equal to c_{stop} constitute the trailing edge. The values of a and b control the placement of the line for the leading edge. The line that forms the top of the trapezoidal structure is constrained to be a horizontal line that intersects the leading edge when t is equal to c_{start}. The trailing edge is constrained to be a line with the negative slope of the leading edge and positioned so as to intersect the top at c_{stop}. Additionally, the values of c_{start} and c_{stop} are constrained so that the number of epochs (i.e., the number of values of t) which constitute the leading edge is equal to that of the trailing edge. These constraints could be eliminated with additional free parameters in the function f, but at the expense of increasing the complexity of the simplex search.

Figure 3.1(f) shows an example of a trapezoidal structure extracted from a time-series data set: the hollow bullets represent the input time-series data (i.e., $Y(t)$), and the solid line represents the extracted structure (i.e., $\hat{Y}(t)$). The trapezoidal structure detector uses simplex search to find values for the free parameters in f so that the $\hat{Y}(t)$ generated by f minimizes the value of E.

3.4. Piecewise Application of Structure Detectors

Each structure detector fits its function f to an entire time-series data set so as to minimize the sum of squared error between the extracted structure and the time series. The optimal fit of f, therefore, must average the effects of the disparate subregions of the time series, extracting structures that follow the general, global trend of the data. For example, the straight structure shown in Figure 3.1(b) reflects the overall trend of the data, filtering out the visually-salient, local trends. These local trends may provide information useful for subsequent analyses and, as such, the extracted structure must be able to represent them.

Local trends can be captured via the piecewise application of the structure detectors: fitting a function f to contiguous subregions of a time series such that the union of the subregions is the entire time series and the intersection of the subregions is empty. The resulting superstructure consists of p extracted substructures, where the substructures are one of the six structure types described in Section 3.3. Such a superstructure is fit to the time series $Y(t)$ with the function g that is defined as

$$g(Y(t)) = \begin{cases} f(Y(t)) & s_1 \leq t < s_2 \\ f(Y(t)) & s_2 \leq t < s_3 \\ \vdots & \\ f(Y(t)) & s_p \leq t \leq n \end{cases}$$

where the values of s partition the time series $Y(t)$ into subregions of consecutive values of t such that subregion j starts when t is equal to s_j (s_1 is equal to 1), and f is a function fit by one of the six structure detectors applied to each of the p subregions. The sum of squared error for the superstructure, by extension, is simply the sum of the sum of squared errors for all p subregions.

For a given set of values for s, the best-fit structure is extracted from each subregion of the time series by the structure detector implementing the function f selected for g. In order to extract the superstructure with the minimal sum of squared error for the entire time series, the values of s must partition the time series so as to minimize the sum of squared error within each subregion. Consequently, extracting the best-fit superstructure requires an iterative process of two steps: (1) select a set of values for s, and (2) extract from each subregion the structure with the minimum sum of squared error given a particular function f. This process continues until a set of values for s is found that minimizes the sum of squared error for the superstructure. Rather than testing all possible p-partitionings of the time series (which would be intractable for any reasonably-sized data set), dynamic programming [10] is used to reduce the computational effort necessary to find an optimal partitioning which minimizes the sum of squared error for the superstructure. The problem of identifying an optimal set of values for s using dynamic programming can be expressed with the recurrence relation

$$G_\gamma[\alpha, \beta] = \begin{cases} \mathrm{err}(f(Y(\alpha, \ldots, \beta))) & \gamma = 1 \\ \min_{\alpha < \tau \leq \beta - \gamma + 2}\{G_1[\alpha, \tau - 1] + G_{\gamma-1}[\tau, \beta]\} & \gamma > 1 \end{cases}$$

where err computes the sum of squared error of the approximation $f(Y(\alpha, \ldots, \beta))$ and $G_\gamma[\alpha, \beta]$ is the best γ-partitioning of $Y(t)$ for $\alpha \leq t \leq \beta$. Dynamic programming solves each subproblem, namely computing f for each contiguous subsequence of values of t, and composes the results to find the optimal partitioning. The function g, therefore, can be computed as $G_p[1, n]$ and an optimal set of values for s is the sequence of τ values used in the optimal recurrence.

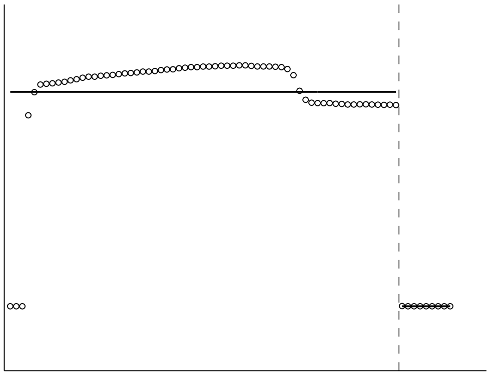
Figure 3.2 shows the piecewise extraction of each of the six structure types to fit two subregions to the same time-series data set shown in Figure 3.1. Each graph plots the input time-series data with hollow bullets overlaid with two solid lines, one for the structure extracted from each subregion, and a vertical dashed line separating the two subregions. For each superstructure, the recurrence relation defined by G is solved by setting p equal to two and f equal to the function associated with the appropriate structure type; the solution to G determines both the values of s which partition the time series into subregions and the free parameters of f which fit the structure within each subregion. For example, Figure 3.2(c) depicts the partitioning and the exponential structure fitted to the time series within each subregion: the first subregion includes the first four epochs (i.e., values of t), the second subregion incorporates the remaining epochs. Since the recurrence relation G is solved for each superstructure separately, the resulting partitionings need not be the same among the piecewise extractions shown in the graphs (e.g., Figures 3.2(a) and 3.2(c)). To avoid fitting the transition between separate structures, the time series is partitioned between epochs. For example, the exponential superstructure shown in Figure 3.2(c) is partitioned between epochs four and five.

The pair of structures extracted for each superstructure tend to better reflect the local trend in the data as compared to the single structure extracted for the same time series as shown in Figure 3.1. Notice that the second subregion for each superstructure (except exponential) has been fitted with a horizontal line. This is possible because each type of structure, both linear and nonlinear, can degrade to a constant structure for particular combinations of values of the free parameters in their corresponding functions f. Figure 3.3 shows the piecewise application of the straight structure detector to fit various numbers of subregions to the same time-series data: as the number of subregions increases, the fit of each superstructure becomes more representative of the local trend embedded within the time series, resulting in a concomitant decrease in the sum of squared error.

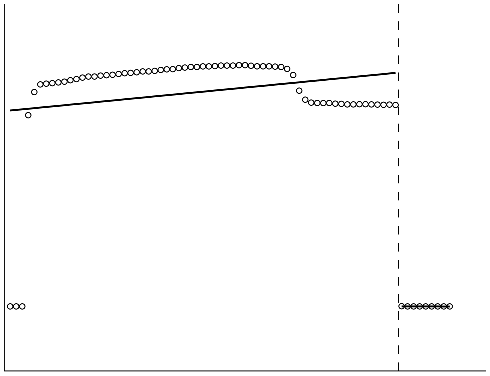
The piecewise application of the structure detectors in the function g extracts the same type of structure from each subregion of the time series. The local trend within each subregion can be better represented if each subregion is fitted with the type of structure that results in the minimum sum of squared error for that subregion, regardless of the structure types fitted to the other subregions. Such a composite superstructure can be fitted using the function h that is defined as

$$h(Y(t)) = \begin{cases} f_1(Y(t)) & s_1 \leq t < s_2 \\ f_2(Y(t)) & s_2 \leq t < s_3 \\ \vdots & \\ f_p(Y(t)) & s_p \leq t \leq n \end{cases}$$

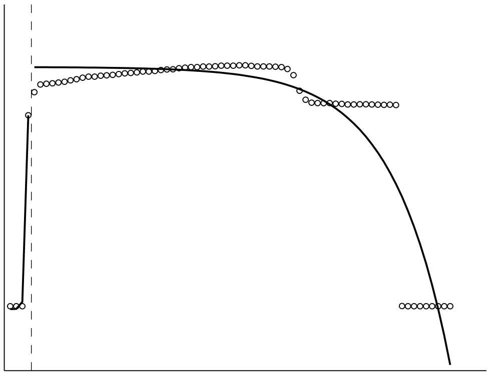
where the values of s partition the time series $Y(t)$ into subregions of consecutive values of t such that subregion j starts when t is equal to s_j (s_1 is equal to 1), and f_j is equivalent to any of the six functions f defined in Section 3.3 and applied to subregion j. As with the function g, the sum of squared error for the composite superstructure is the sum of the sum of squared errors for all p subregions.



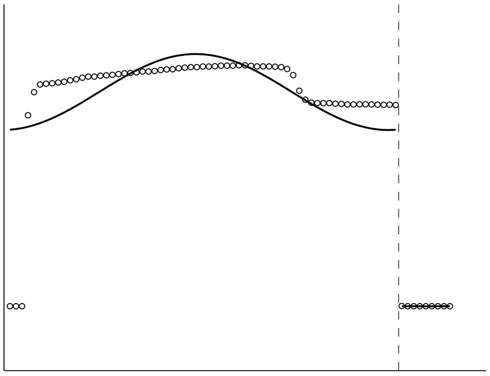
(a) Constant



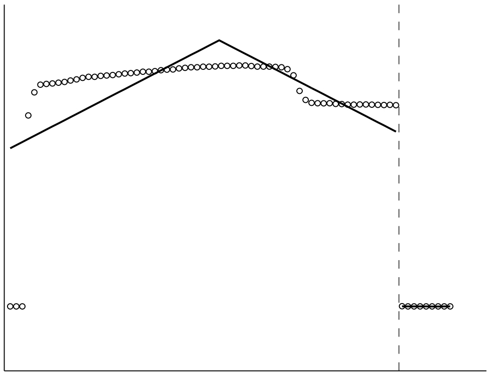
(b) Straight



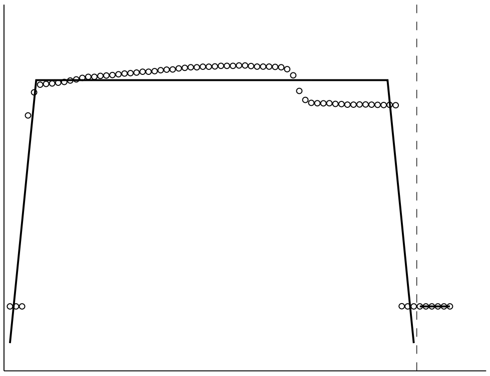
(c) Exponential



(d) Sinusoidal

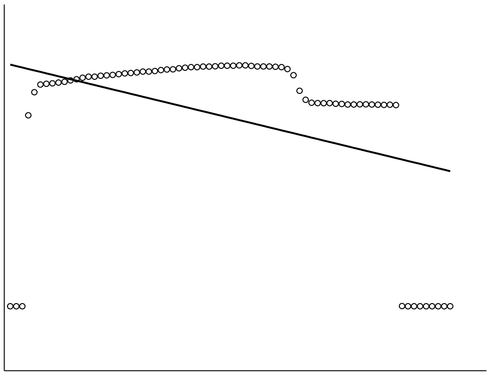


(e) Triangular

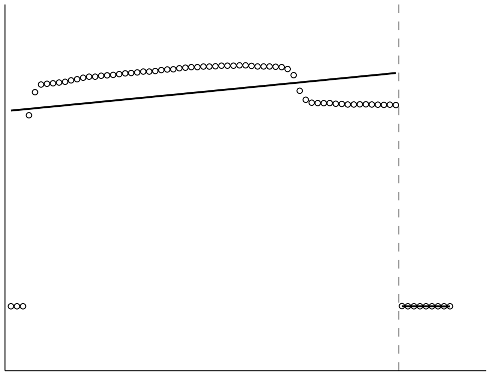


(f) Trapezoidal

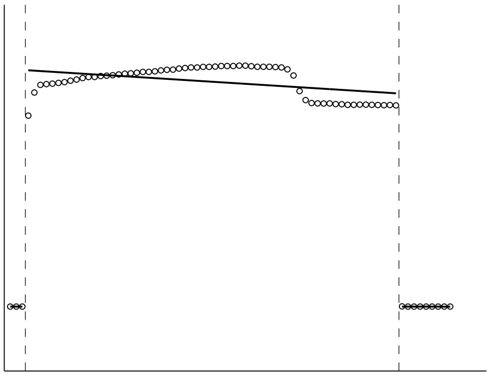
Figure 3.2 The piecewise extraction of the six structures to fit two subregions to a common data set. Each graph shows the input time-series data plotted with hollow bullets overlaid with two solid lines, one for the structure extracted from each subregion, and a vertical dashed line separating the two subregions. The superstructures ordered from smallest to largest sum of squared error are (f) trapezoidal, (c) exponential, (e) triangular, (d) sinusoidal, (b) straight, and (a) constant.



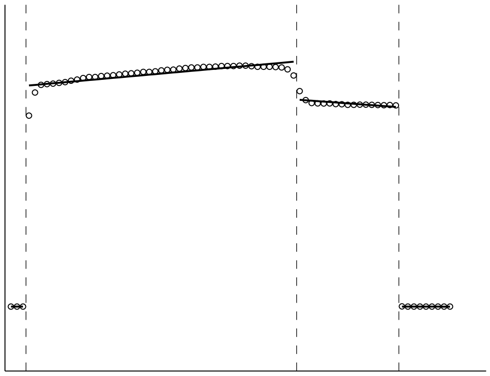
(a) One Subregion



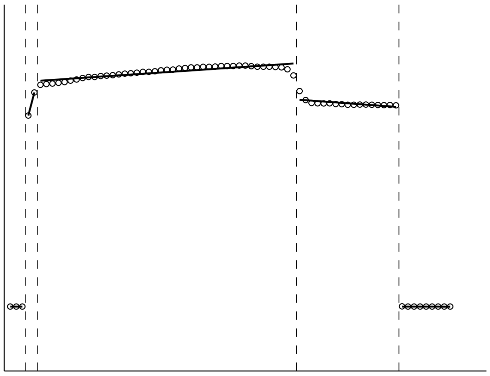
(b) Two Subregions



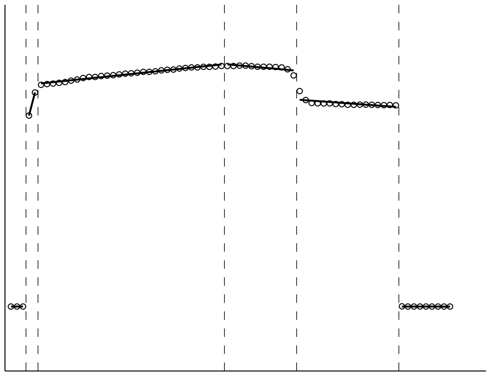
(c) Three Subregions



(d) Four Subregions



(e) Five Subregions



(f) Six Subregions

Figure 3.3 The piecewise extraction of the straight structure to fit various numbers of subregions to a common data set. Each graph shows the input time-series data plotted with hollow bullets overlaid with solid lines representing the straight structure extracted from each subregion. Vertical dashed lines separate the subregions. Increasing the number of subregions results in a concomitant decrease in the sum of squared error.

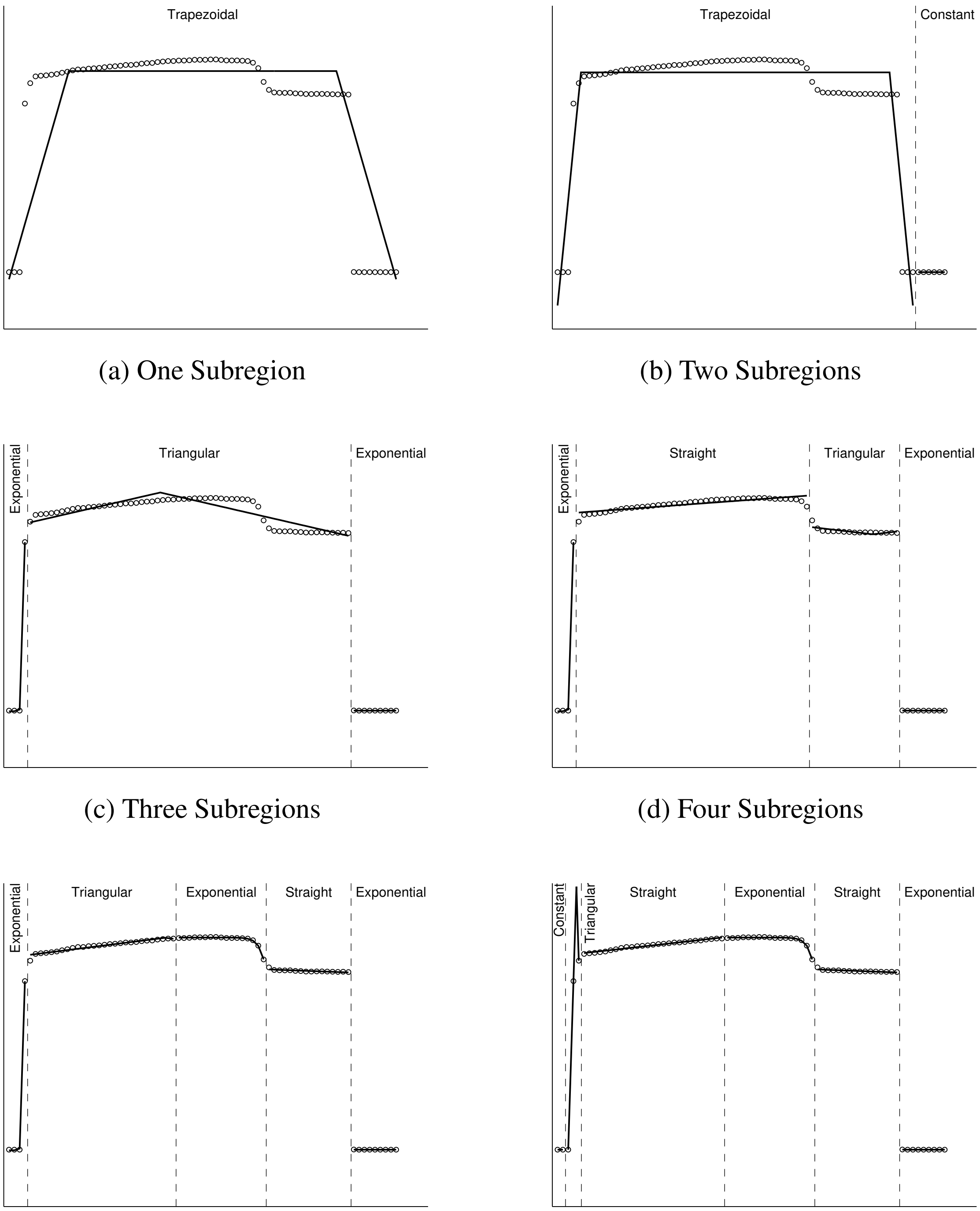
The process of extracting a composite superstructure from a time-series data set requires a dynamic programming approach similar to that used for extracting a homogeneous superstructure. The problem of identifying an optimal set of values for s with dynamic programming for a composite superstructure can be expressed with the recurrence relation

$$H_\gamma[\alpha,\beta] = \begin{cases} \min_f\{\mathrm{err}(f(Y(\alpha,\ldots,\beta)))\} & \gamma = 1 \\ \min_{\alpha<\tau\leq\beta-\gamma+2}\{H_1[\alpha,\tau-1] + H_{\gamma-1}[\tau,\beta]\} & \gamma > 1 \end{cases}$$

The recurrence relation H is the same as G except that the base case of H fits each of the six structure types and selects the one with the smallest sum of squared error to approximate the subregion. In the case of a tie, the less complex structure type is used. Based on the number of free parameters in the functions f and the restrictions placed on assigning values to those free parameters, the structure types in order from least to most complex are constant, straight, exponential, sinusoidal, triangular, and trapezoidal.

Figure 3.4 shows the piecewise extraction of the composite superstructure to fit various numbers of subregions to the same time-series data. For each number of subregions, the recurrence relation defined by H is solved by setting p equal to the number of subregions; the solution to H determines the values of s which partition the time series into subregions, the functions f which best approximate each subregion, and the free parameters for each f which fit the structure within each subregion. The composite superstructure that best fits the time series using one subregion is the trapezoidal structure, as is expected given Figure 3.1. As with the homogeneous superstructure, the fit of each composite superstructure becomes more representative of the local trend as the number of subregions increases, resulting in a concomitant decrease in the sum of squared error. Since the composite superstructure extracts the best-fit structure type from each subregion, the sum of squared error for each composite superstructure is less than or equal to that for the corresponding homogeneous superstructure. Notice that the extracted superstructure with six subregions appears odd: there is a “spike” in the second subregions of the superstructure. This spike is a triangular structure fitted to a subregion of three data points with a sum of squared error equal to zero. The strange appearance of the graph is a consequence of overfitting: too many subregions were used to fit the time series and, as a result, small subregions were fitted with counterintuitive structure types in an effort to minimize the error. To avoid overfitting, a balance between the number of subregions and error must be struck.

Figure 3.5 shows the relationship between the number of subregions used in the piecewise application of the structure detectors to extract a composite superstructure and the resulting sum of squared error: as the number of subregions increases, the sum of squared error decreases rapidly, reaching zero when there are enough subregions so that the structure extracted from each subregion perfectly fits the data in that subregion. (A similar relationship exists when extracting a homogeneous superstructure.) How many subregions are sufficient so that the extracted composite superstructure filters out the noise and other minor perturbations in the data and, at the same time, captures enough of the local trend so as to be of use for subsequent analyses? Such a situation occurs when a weighted sum of the number of subregions and the error is minimized. Model selection [92] using a formal description length metric, such as Akaike’s Information Criterion (AIC) [1] or Bayesian Information Criterion (BIC) [77], could be used to find the optimal number of subregions by balancing the error with a penalty which increases with the number of subregions. Such approaches, however, are grounded in asymptotics and, consequently, may be sensitive to small sample sizes. An alternative approach is to identify the number of subregions such that



(a) One Subregion

(b) Two Subregions

(c) Three Subregions

(d) Four Subregions

(e) Five Subregions

(f) Six Subregions

Figure 3.4 A composite superstructure fitted to a common data set with various numbers of subregions. Each graph shows the input time-series data plotted with hollow bullets overlaid with solid lines representing the structure extracted from each subregion. Vertical dashed lines separate the subregions, and the type of structure fitted to each subregion is indicated. Increasing the number of subregions results in a concomitant decrease in the sum of squared error.

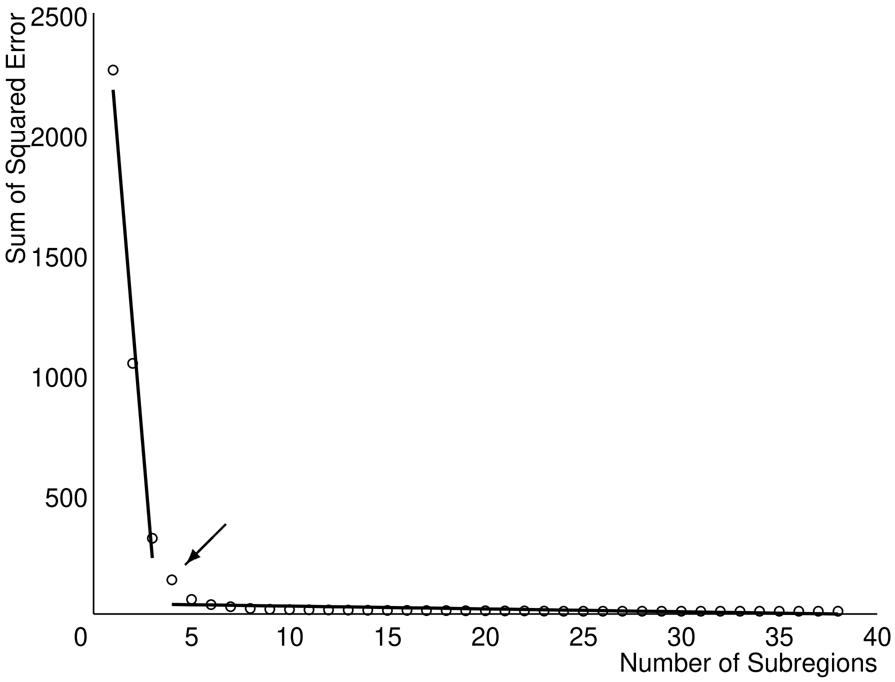


Figure 3.5 The relationship between the number of subregions used in the piecewise application of the structure detectors to extract a composite superstructure and the resulting sum of squared error. The sum of squared error, plotted with hollow bullets, decreases rapidly as the number of subregions increases, reaching zero when there are enough subregions so that the structure extracted from each subregion perfectly fits the data in that subregion. The pair of extracted straight structures, plotted as two solid lines, show how the sum of squared error values are partitioned; the arrow indicates the number of subregions where the reduction in error grows small with respect to the increase in number of subregions.

any increase would result in a comparatively small decrease in error—i.e., the “knee” of the error versus number-of-subregions curve. This approach would produce results similar to those of a description length metric and has an additional benefit of not being asymptotic with respect to the sample size. Moreover, a graph of the error versus number-of-subregions curve can be inspected either to confirm an expected outcome or to discover the cause of an unreasonable outcome.

The knee of the curve lies at the point where the linear relationship between the sum of squared error and the number of subregions changes from mostly vertical (i.e., having consecutively large decreases in error) to mostly horizontal (i.e., having consecutively small decreases in error). The piecewise application of the straight structure can be used to identify the point where this change in the linear relationship between error and the number of subregions occurs: the straight structure detector is used to fit two subregions to the sequence of sum of squared error values ordered by number of subregions, and the knee is located at the point where the two extracted straight structures meet. The piecewise application of the straight structure detector fits two straight structures so as to minimize the sum of squared error for each piece, therefore the curve must be split where the linear relationship between the error and number of subregions changes from mostly vertical to mostly horizontal; the point where this change occurs is the number of subregions which provides diminishing returns and is the optimal number of subregions (i.e., the value of p to use in the recurrence relations G and H in order to specify the number of subregions into which a time series is to be partitioned). The two straight structures extracted from the sum of squared error values is shown in Figure 3.5, and the optimal number of subregions to use to extract a composite superstructure is marked with an arrow (which identifies four subregions as optimal). Returning to Figure 3.4, using four subregions in the extracted composite superstructure results in a fit which identifies the local trends without capturing every small change in the data. Five subregions, by comparison, has a smaller sum of squared error, however the resulting composite superstructure has been fitted to the time series quite accurately and has not identified the abstract, general trends in the data.

The optimal number of subregions to use for a collection of time-series data sets can be identified using a similar procedure: (1) for each number of subregions, generate an approximation for each data set and compute the sum of squared error, summing the individual error values across all the data sets to arrive at the total error; (2) use the straight structure detector to fit two subregions to the sequence of total error values ordered by number of subregions; and (3) identify the optimal number of subregions to be the point where the two extracted straight structures meet. This procedure can be useful, for example, as a training phase to determine a fixed number of subregions for the structure detectors based on a collection of time-series data sets (i.e., a training set) so as to extract the same number of structures from each data set.

3.5. Structure Detector Implementation

The implementation of the structure detectors requires two separate algorithms: the structure detector algorithm identifies the optimal piecewise approximation of a time-series data set using a specified number of subregions, and the structure detector training algorithm determines an optimal number of subregions for the structure detector algorithm based on a collection of time-series data sets. The structure detector algorithm is general in nature and can be used to extract from a time-series data set any of the six homogeneous superstructures (i.e., constant, straight, exponential, sinusoidal, triangular, or trapezoidal) as well as a heterogeneous, or composite, superstructure. The structure detector algorithm extracts the various types of superstructures by using the appropriate function f to approximate each subregion. Similarly, the structure detector training algorithm can determine an optimal number of subregions for the various types of superstructures by invoking the appropriate instantiation of the structure detector algorithm.

3.5.1. Structure Detector Algorithm

The structure detector algorithm solves the recurrence relations specified by G and H in Section 3.4 using Bellman's dynamic programming method [10]. The input to the algorithm is the number of subregions to use, p, and a time-series data set of length n. The output of the algorithm includes the error of approximating the time-series data set with the superstructure, the partitioning of the time series into subregions, the error of approximating the time series within each subregion, and the values of the free parameters of the function f used to approximate each subregion. In the case where a composite superstructure is extracted, the type of structure used to approximate each subregion is also reported by the algorithm.

The dynamic programming methodology used in the structure detector algorithm first examines all the individual components that could be assembled together into a solution, and then searches through the possible ways of assembling the components to find the final solution. Given a value of p and a time-series data set of length n, the structure detector algorithm proceeds as follows:

1. Read the specified time-series data set. Store the time series as a sequence of $Y(t)$ values where $1 \le t \le n$.

2. For each contiguous subset of t, let S contain the $Y(t)$ values within the subset. Compute the sum of squared error resulting from approximating each subset S with a structure, treating S as an indivisible subregion.

   - When extracting one of the six homogeneous superstructures:
     (a) Apply the appropriate function f to S, resulting in an approximation $\hat{S}$.
     (b) Compute the sum of squared error between S and $\hat{S}$.
     (c) Store in table T_1 an entry indexed by S that contains the values of the free parameters used to fit the structure to S and the resulting sum of squared error.
   - When extracting a composite superstructure:
     (a) Apply each of the six functions f to S, producing six approximations $\hat{S}$.
     (b) Compute the sum of squared error between S and each of the six approximations $\hat{S}$.
     (c) Select as the best approximation of S the structure type that results in the minimum of the six sum of squared error values. To resolve the situation where the minimum sum of squared error value is not unique, choose the less complex structure type. Based on the number of free parameters in the functions f and the restrictions placed on assigning values to those free parameters, the structure types in order from least to most complex are constant, straight, exponential, sinusoidal, triangular, and trapezoidal.
     (d) Store in table T_1 an entry indexed by S that contains the structure type selected to approximate the subset S, the values of the free parameters used to fit the structure to S, and the resulting sum of squared error.

3. Compute the minimum sum of squared error of approximating $Y(t)$ within various contiguous subsets of t using q subregions where $1 \le q \le p$. Let i denote the start of the current subset. Given q and for each i such that $1 \le i \le n - q + 1$, let S contain the values of $Y(t)$ where $i \le t \le n$. Compute the minimum sum of squared error of approximating S using q subregions.

   (a) Let j represent the location in S where the values of $Y(t)$ in S will be split into two subregions, S_j^1 and S_j^2. For each j such that $1 \le j \le n - i - q + 2$, the sum of squared error of approximating S_j with q subregions is equal to the total of the sum of squared errors for approximating S_j^1 with one subregion and S_j^2 with $q - 1$ subregions.

- Let S_j^1 contain the subset of $Y(t)$ values where $i \le t \le i+j-1$. The sum of squared error for approximating S_j^1 with one subregion was calculated in step #2 and can be retrieved from table T_1.
- Let S_j^2 contain the subset of $Y(t)$ values where $i+j \le t \le n$. The sum of squared error for approximating S_j^2 with $q-1$ subregions was computed in a previous iteration of step #3 (or in step #2 for the case where $q=2$) and can be retrieved from table T_{q-1}.

(b) Select as the best approximation of S the S_j that results in the minimum sum of squared error.

(c) Store in table T_q an entry indexed by S that contains the selected value of j and the resulting sum of squared error.

4. Report the best approximation to the time-series data set using $q=p$ subregions by backchaining through the tables T and assembling the information for each subregion.

   (a) Let $i=1$. Let S contain the subset of $Y(t)$ values where $i \le t \le n$.

   (b) Retrieve the entry from table T_q indexed by S.

   (c) Use the value of j in the entry to subdivide S into S_1 and S_2.

   - Let S_1 contain the values of $Y(t)$ where $i \le t \le i+j-1$.
   - Let S_2 contain the values of $Y(t)$ where $i+j \le t \le n$.

   (d) Retrieve the entry from table T_1 indexed by S_1, and report its contents to summarize the approximation of the current subregion.

   (e) Let $q=q-1$. Let $i=i+j$. Let $S=S_2$. If $q>1$, then return to step #4(b).

   (f) Retrieve the entry from table T_1 indexed by S, and report its contents to summarize the approximation of the last subregion.

   (g) Report the total of the sum of squared errors across all p subregions as the sum of squared error for fitting the superstructure to the entire time-series data set.

Step #2 computes the sum of squared error for all contiguous subsets within the time-series data sets, thereby investigating all subregions that could possibly appear in the final solution. Step #3 searches through the possible combinations of subregions incrementally, using results generated in step #2 as well as previous iterations of step #3. Once the search is complete, step #4 uses the stored partial results to report the final solution.

The computational complexity of each step of the structure detector algorithm is as follows:

1. Reading the time-series data set is $O(n)$.

2. Searching all contiguous subsets of t is $O(n^2)$.

   - When extracting one of the six homogeneous superstructures:

     (a) Applying the constant function f is $O(1)$, and applying any of the other functions f is $O(n)$.

(b) Computing the sum of squared error for the constant function f is $O(1)$, and computing the sum of squared error for any of the other functions f is $O(n)$.

(c) Storing the approximation information is $O(1)$.

- When extracting a composite superstructure:

  (a) Applying the six functions f is $O(n)$.

  (b) Computing the sum of squared error is $O(n)$.

  (c) Selecting among the structure types is $O(1)$.

  (d) Storing the approximation information is $O(1)$.

3. Examining the values of $q \leq p$ for each subset of t where $i \leq t \leq n$ and $1 \leq i \leq n - q + 1$ is $O(pn)$.

   (a) Moving j through each subset has a complexity of $O(n)$.

   - The sum of squared error for S_j^1 is established via a table lookup of $O(1)$.
   - The sum of squared error for S_j^2 is established via a table lookup of $O(1)$.

   (b) Selecting the best approximation S_j is $O(n)$.

   (c) Storing the approximation information is $O(1)$.

4. Backchaining through the tables requires p table accesses and, thus, has a complexity of $O(p)$.

The computational complexity for each step can be calculated by multiplying the complexities of the individual substeps. Step #1 is $O(n)$. Step #2 is $O(n^2)$ when approximating with the constant superstructure, $O(n^3)$ when approximating with any of the other five homogeneous superstructures, and $O(n^3)$ when approximating with a composite superstructure. Step #3 is $O(pn^2)$. Step #4 is $O(p)$. Combining the computational complexities for each step results in the overall computational complexity for the structural detector algorithm as follows:

- When approximating with the constant superstructure, the algorithm is $O(pn^2)$.

- When approximating with any of the other five homogeneous superstructures, the algorithm is $O(n^3 + pn^2)$.

- When approximating with a composite superstructure, the algorithm is $O(n^3 + pn^2)$.

The computational effort of finding the optimal piecewise approximation of a time-series data set can be lessened by using sampling, smoothing, or multiresolution techniques [71] to reduce the length n of the time series, provided that the resulting morphological description does not compromise the subsequent classification accuracy.

3.5.2. Structure Detector Training Algorithm

The structure detector algorithm requires the number of subregions p as input. In order to determine an optimal value of p that minimizes a weighted sum of the number of subregions and the approximation error, the structure detector training algorithm analyzes the change in error as the number of subregions increases. Moreover, the algorithm performs its analysis within the context of a collection of time-series data sets so as to arrive at a value for p that is appropriate given the variability across the data sets.

The structure detector training algorithm determines a value for p based on a training set comprising a sample of the entire collection of time-series data sets that are to be analyzed. Given a training set R composed of r time-series data sets, R_1 to R_r, the structure detector training algorithm arrives at an optimal value of p as follows:

1. Read each of the time-series data sets R_i where $1 \leq i \leq r$. Let n_i be equal to the length of time-series data set R_i. Let n_{max} be equal to the maximum n_i.
2. For each q such that $1 \leq q \leq \frac{1}{2}n_{max} + 1$, compute the total sum of squared error for approximating each of the time-series data sets in R using q subregions.
   (a) For each time-series data set R_i where $1 \leq i \leq r$, invoke the structure detector algorithm with $p = q$ to approximate the time series R_i with the appropriate type of superstructure. Let m_i be equal to the sum of squared error returned by the structure detector algorithm.
   (b) Let M_q be equal to the sum of the m_i values where $1 \leq i \leq r$.
3. Find the knee in the curve plotted by the sequence of M values. Invoke the structure detector algorithm with $p = 2$ to approximate the sequence of M values with the straight superstructure. Report as the optimal value of p the value of q such that M_q falls at the start of the second subregion.

Once a value of p has been set, it can be used as input to the structure detector algorithm when analyzing each of the time-series data sets within the entire collection.

Assuming that the values of n for the r data sets in R are similar, the computational complexity of each step of the structure detector training algorithm is as follows:

1. Reading the r time-series data sets, each of length n, and calculating n_{max} is $O(rn)$.
2. Examining the values of q is $O(n)$.
   (a) Invoking the structure detector algorithm r times with $p = q$ to fit a constant superstructure is $O(rn^2)$, to fit any of the remaining homogeneous superstructures is $O(rn^3)$, and to fit a composite superstructure is $O(rn^3)$.
   (b) Adding the sum of squared error values has a complexity of $O(r)$.
3. Invoking the structure detector algorithm with $p = 2$ to approximate the M values with a straight superstructure is $O(n^3)$. Reporting the optimal value of p is $O(1)$.

The computational complexity for each step can be calculated as a combination of the individual substeps. Step #1 is $O(rn)$. Step #2 is $O(rn^3)$ when approximating with the constant superstructure, $O(rn^4)$ when approximating with any of the other five homogeneous superstructures, and $O(rn^4)$ when approximating with a composite superstructure. Step #3 is $O(n^3)$. Combining the computational complexities for each step results in the overall computational complexity for the structure detector training algorithm as follows:

- When approximating with the constant superstructure, the algorithm is $O(rn^3)$.
- When approximating with any of the other five homogeneous superstructures, the algorithm is $O(rn^4)$.
- When approximating with a composite superstructure, the algorithm is $O(rn^4)$.

The fact that the structure detector training algorithm has a computational complexity that is dominated by the n^4 term makes the algorithm expensive for any reasonably-sized data sets. A small modification to the structure detector algorithm can simplify the structure detector training algorithm and, by extension, reduce its computational complexity.

3.5.3. Reducing the Computational Complexity

When using the structure detector algorithm to approximate a time-series data set with p subregions, tables T_1 through T_p are produced such that the best approximation using p subregions can be found via backchaining starting with table T_p. Notice, however, that the best approximation of the time-series data set with $p-1$ subregions has also been computed and can be found via backchaining starting with table T_{p-1}. Generally, the best approximation of the time-series data set with each of q subregions can be found via backchaining starting with table T_q where $1 \leq q \leq p$.

Using this observation, the structure detector algorithm can be modified to report the sum of squared error resulting from approximating the time-series data set with q subregions where $1 \leq q \leq p$ by modifying step #4 of the structure detector algorithm to be

4. Report the best approximation to the time-series data set using q subregions where $1 \leq q \leq p$ by backchaining through the tables T and assembling the information for each subregion.

The remainder of the algorithm remains unchanged.

The computational complexity of step #4 of the modified structure detector algorithm is

4. Backchaining through the tables requires p table accesses. Reporting the best approximation for each of q subregions where $1 \leq q \leq p$ is $O(p^2)$.

The computational complexity of the other steps of the algorithm remains unchanged. The overall computational complexity for the modified structure detector algorithm is as follows:

- When approximating with the constant superstructure, the algorithm is $O(p^2 + pn^2)$.
- When approximating with any of the other five homogeneous superstructures, the algorithm is $O(n^3 + p^2 + pn^2)$.

- When approximating with a composite superstructure, the algorithm is $O(n^3 + p^2 + pn^2)$.

If $p \ll n$, the computation complexity of the modified structure detection algorithm is the same as the original algorithm.

The structure detector training algorithm can take advantage of the modified structure detector algorithm with the following changes:

2. Compute the total sum of squared error for approximating the time-series data sets in R using $\frac{1}{2}n_{max} + 1$ subregions.
   (a) For each time-series data set R_i where $1 \leq i \leq r$, invoke the modified structure detector algorithm with $p = \frac{1}{2}n_{max} + 1$ to approximate the time series R_i with the appropriate type of superstructure. Let m_i^q be equal to the sum of squared error returned by the structure detector algorithm for the time-series data set R_i with q subregions where $1 \leq q \leq \frac{1}{2}n_{max} + 1$.
   (b) Let M_q be equal to the sum of the m_i^q values where $1 \leq i \leq r$ for each $1 \leq q \leq \frac{1}{2}n_{max} + 1$.

The other steps of the algorithm remain unchanged.

The computational complexity of this modified structure detector training algorithm undergoes a parallel modification:

2. Examining one value of p is $O(1)$.
   (a) Invoking the modified structure detector algorithm r times with $p = \frac{1}{2}n_{max} + 1$ to fit a constant superstructure is $O(rn^3)$, to fit any of the remaining homogeneous superstructures is $O(rn^3)$, and to fit a composite superstructure is $O(rn^3)$.
   (b) Adding the sum of squared error values has a complexity of $O(rn)$.

The computational complexity of the other steps of the algorithm remains unchanged. The overall computational complexity for the modified structure detector training algorithm is as follows:

- When approximating with the constant superstructure, the algorithm is $O(rn^3)$.
- When approximating with any of the other five homogeneous superstructures, the algorithm is $O(rn^3)$.
- When approximating with a composite superstructure, the algorithm is $O(rn^3)$.

The modified structure detector training algorithm successfully reduces by a factor of n the computational complexity when approximating with any of the other five homogeneous superstructures or a composite superstructure. The complexity when approximating with the constant superstructure remains essentially the same.

Because of the reduction in computational complexity afforded by the modification to both algorithms, the modified version of both the structure detector algorithm and the structure detector training algorithm are the logical choices to implement. Under the assumption that $p \approx O(n)$, the modified structure detector algorithm approximates a time-series data set in $O(n^3)$ and the structure detector training algorithm determines an optimal value of p based on a training set containing r time-series data sets in $O(rn^3)$. For the modified versions of both algorithms, these computational complexities apply regardless of the type of superstructure being extracted from the data.

3.6. Discussion

A structural approach to pattern recognition necessarily requires generating a morphological description of the data under analysis and then performing classification based on that description. A general approach to producing such a morphological description of time-series data is one that finds an optimal piecewise approximation by fitting a sequence of structures selected from a library of domain-independent structure types. The task of generating a morphological description of a time-series data set, therefore, becomes one of identifying a sequence of structures that approximate the data such that the difference between the approximation and the data is minimized.

Dual evidence from the field of signal processing and the structural pattern recognition literature suggests a set of six structure types—constant, straight, exponential, sinusoidal, triangular, and trapezoidal—to serve as the foundation for a library of structure types for generalized feature extraction. A structure detector is used to identify the optimal piecewise-fit of each structure type in time-series data using basic curve-fitting algorithms and search strategies.

The suite of structure detectors and the methodology used to identify the optimal piecewise approximation to time-series data address the characteristics necessary for generalized feature extraction from time-series data as follows:

- The selection of structure types is based on their perceived usefulness in both the field of signal processing and the structural pattern recognition literature. As such, the six structure types capture fundamental trends and relationships that have proven useful in other applications.
- Time-series data can be approximated with arbitrary accuracy, as determined by the number of subregions used in the piecewise application of the structure detectors.
- The optimal piecewise-fit of a time-series data set can be represented by a sequence of structure types, onsets, and values for the free parameters.

Whether the suite of six structure detectors can satisfy the remaining characteristics, namely being able to support a subsequent classification task and being domain independent, can be determined empirically with an experiment to evaluate the efficacy of these structure types for classification of data from various domains.

Chapter 4

Structure Detector Evaluation

4.1. Introduction

The structure detectors described in Chapter 3 were designed to perform generalized feature extraction for structural pattern recognition in time-series data by generating a range of generally-useful descriptive characteristics for classification. Each structure detector acts as a feature extractor capable of recognizing one specific type of primitive; applying a structure detector in a piecewise fashion to a time series results in a sequence of features that represents the original data and can be used as the basis for classification. The resulting classification accuracy depends on how well the extracted features differentiate among objects from different classes: features which better discriminate will result in a higher classification accuracy as compared to features with less discriminatory power. Since each structure detector extracts a unique set of features, selecting among the structure detectors to generate features for classification will affect the subsequent accuracy.

The relative efficacy of the structure detectors can be evaluated by comparing the classification accuracies achieved when using the various structure detectors for feature extraction. Additionally, established techniques used to extract features from time-series data can serve as a benchmark: if the classification accuracies achieved with features generated by the structure detectors are at least as high as the accuracies achieved with the other techniques, then it can be concluded that the structure detectors capture characteristics of time-series data suitable for classification at least as well as commonly-used methods. The most suitable benchmark would comprise domain-independent feature-extraction methods for structural pattern recognition in time-series data—e.g., chain codes and curve fitting. However, such techniques are subsumed by the structure detectors and, consequently, would not constitute a distinct comparison. A benchmark unrelated to the structure detectors can be established with feature-extraction techniques commonly used for statistical pattern recognition.

4.2. Statistical Feature Extractors

Three commonly-used domain-independent statistical methods for feature extraction are the identity transformation, the Fourier transformation, and the wavelet transformation [59][60]. Each of these techniques generates an approximation of a time-series data set composed of a sum of weighted elements drawn from an associated basis (i.e., a set of fundamental, or constituent, waveforms). The weighted elements which constitute an approximation are characteristic of the particular data set under analysis and, therefore, can be used as features for classification. The precise number of weighted elements in the approximation varies among data sets and determines the number of features extracted from the data set. Since statistical classifiers generally require a constant number of features from each data set, a separate training phase for each of the three statistical feature extractors is necessary to determine a fixed number of features to extract. Moreover, each training phase must also transform the time-series data set into a format suitable for analysis by the associated feature extractor, if necessary. To perform these two operations, the training phase for each of the statistical feature extractors will have both a data preparation and a model selection component.

4.2.1. Identity Transformation

The simplest features to use for classification of time-series data are those which require no extraction whatsoever—i.e., the raw data itself. The basis necessary to generate such an approximation comprises a collection of n waveforms, where n is the length of the data set under analysis, such that basis waveform i is equal to one at $t = i$ and equal to zero everywhere else. Since such a basis is essentially a collection of unit vectors, expressing the identity transformation as a sum of weighted basis waveforms would be unnecessarily overcomplicated. Rather, the approximation generated by the identity transformation can be more simply expressed as

$$\hat{Y}(t) = Y(t)$$

where $\hat{Y}(t)$ is a time series which constitutes an approximation to $Y(t)$ such that each value in $\hat{Y}(t)$ is equal to the corresponding value in $Y(t)$. Since $\hat{Y}(t)$ is equivalent to $Y(t)$, the sum of squared error E is always equal to zero. The features generated by the identity transformation for classification comprise the sequence of $\hat{Y}(t)$ values. The extracted features can be arranged into a feature vector having the form

$$\mathcal{A}_1 \; \mathcal{A}_2 \; \mathcal{A}_3 \; \cdots \; \mathcal{A}_n$$

where $\mathcal{A}_i$ is the i^{th} feature such that $\mathcal{A}_i = \hat{Y}(i)$. The feature vector contains a total of n features. See Appendix A for a concrete example. Applying the identity transformation to a data set of length n and assembling the extracted features has a computational complexity of $O(n)$.

The training phase for the identity feature extractor comprises both a data preparation and a model selection component applied to a training set. Since the number of features extracted depends on the length of the data set, the data preparation component must establish a uniform length for all data sets to be analyzed. This can be accomplished by selecting either the shortest or longest data set within the training set. Once established, each data set can be made to conform to the fixed length either by truncation (i.e., terminating the data set at the desired length) or extension (i.e., artificially extending the data set using techniques such as zero padding, extrapolation of observed trends, or wraparound methods). Since there is no tradeoff between the number of basis waveforms used in the approximation and the resulting sum of squared error (because the sum of squared error E is consistently zero), there is no model selection to be performed. Thus, the outcome of the training phase consists solely of the fixed data length. The computational complexity of the training phase is $O(rn)$ when applied to a training set containing r data sets where n is the maximum length of the data sets within the training set.

4.2.2. Fourier Transformation

The Fourier transformation [69][71] generates an approximation to a time-series data set using as a basis the cosine and sine functions having period n and frequency j where $j = \{0, 1, 2, \ldots, \frac{1}{2}n\}$ and n is the length of the data set. Figure 4.1 plots a subset of the basis waveforms used by the Fourier transformation. The sine and cosine functions with frequency j equal to 0, 1, and 2 are shown. When $j = 0$, the sine function is equivalent to the zero function and, consequently, is excluded from the basis.

Given this basis, the Fourier transformation approximates a time series as

$$\hat{Y}(t) = \sum_{j \in B} (a_j \cos jt + b_j \sin jt)$$

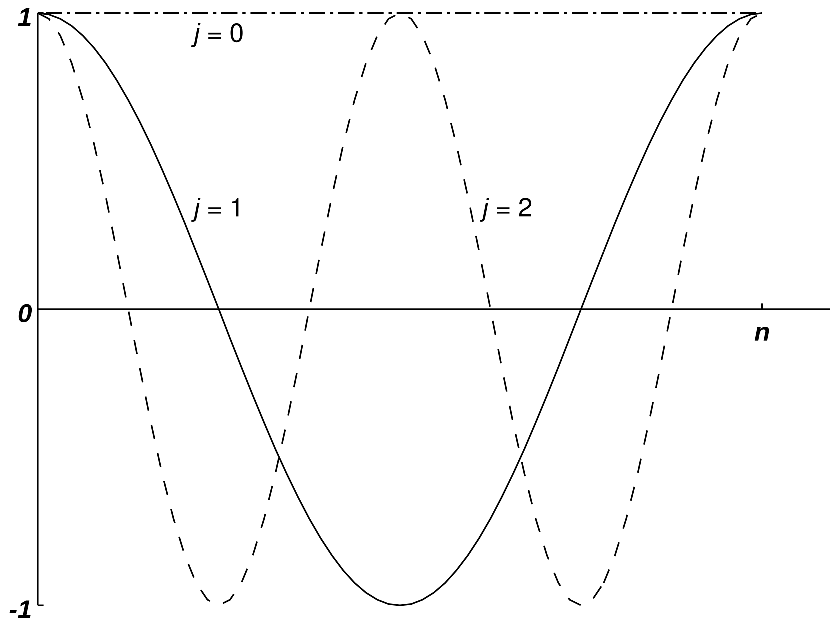
where $\hat{Y}(t)$ is a time series which constitutes an approximation to $Y(t)$, B is a subset of the frequencies within the basis, a_j is the coefficient (or weight) associated with the cosine basis function with frequency j, and b_j is the coefficient (or weight) associated with the sine basis function with frequency j. Notice that for each j in B, both the cosine and sine functions with frequency j are incorporated into the approximation. The sum of squared error E depends on the number of frequencies contained in B: as the number of frequencies in B increases, the sum of squared error E decreases. The features generated by the Fourier transformation for classification comprise the coefficient pairs a_j and b_j for each frequency j in B. The features extracted from a data set can be arranged into a feature vector having the form

$$\mathcal{A}_1 \; \mathcal{B}_1 \; \mathcal{A}_2 \; \mathcal{B}_2 \; \cdots \; \mathcal{A}_{|B|} \; \mathcal{B}_{|B|}$$

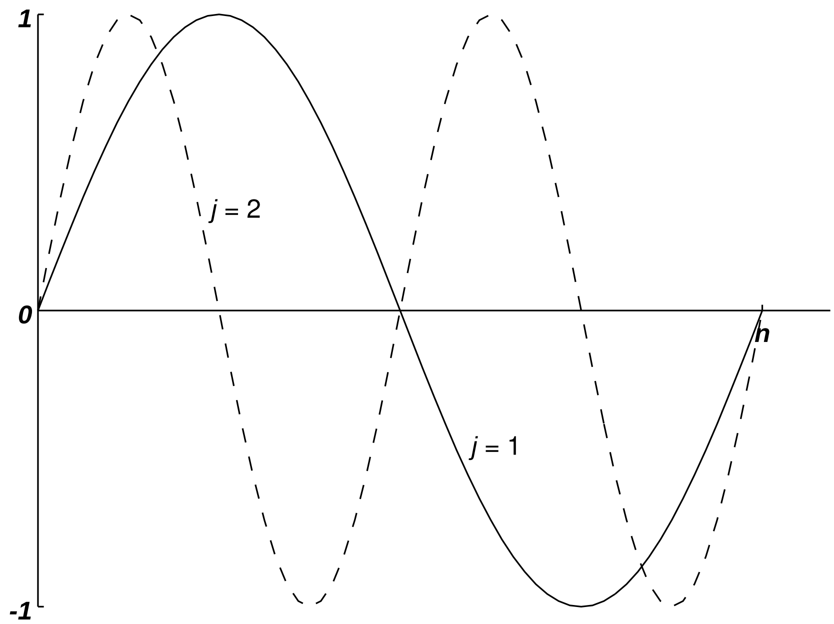
where $\mathcal{A}_i$ and $\mathcal{B}_i$ constitute the i^{th} feature pair for each i such that $1 \leq i \leq |B|$ and $|B|$ is the number of frequencies in B. In order to map feature pair i to frequency j, the frequencies in B are ordered according to their overall contribution towards approximating the data sets within a training set, as determined during the training phase. (The training phase, discussed next, computes the overall contribution of each frequency, c_j, and orders the frequencies according to these values, from highest to lowest.) $\mathcal{A}_1$ and $\mathcal{B}_1$ are equal to a_j and b_j where $j \in B$ such that frequency j contributes the most (i.e., has the largest c_j value) towards approximating the data sets within a training set. $\mathcal{A}_2$ and $\mathcal{B}_2$ are equal to a_j and b_j where $j \in B$ such that frequency j contributes the second most (i.e., has the second largest c_j value) towards approximating the data sets within a training set. And so on. The feature vector contains a total of $2 * |B|$ features. See Appendix A for a concrete example. Applying the Fourier transformation to a data set of length n and assembling the extracted features has a computational complexity of $O(n \log n)$.

The training phase for the Fourier transformation feature extractor comprises both a data preparation and a model selection component applied to a training set. The fast Fourier transformation by Press [71] requires that the time-series data set has a length equal to a power of two.[1] The data preparation component, therefore, must establish a uniform length equal to a power of two for all the data sets to be analyzed. This can be accomplished by selecting either the shortest or the longest data set within the training set: if the shortest data set is selected, then the length is fixed at the

[1]Other, more complex, versions of the fast Fourier transformation can be applied directly to data sets that violate this length requirement.



Cosine Functions



Sine Functions

Figure 4.1 A subset of the cosine and sine functions used as basis waveforms by the Fourier transformation. Each basis waveform has period n and frequency j where $j = \{0, 1, 2, \ldots, \frac{1}{2}n\}$ and n is the length of the data set. The basis waveforms with j equal to 0, 1, and 2 are shown. When $j = 0$, the sine function is equivalent to the zero function and, consequently, is excluded from the basis.

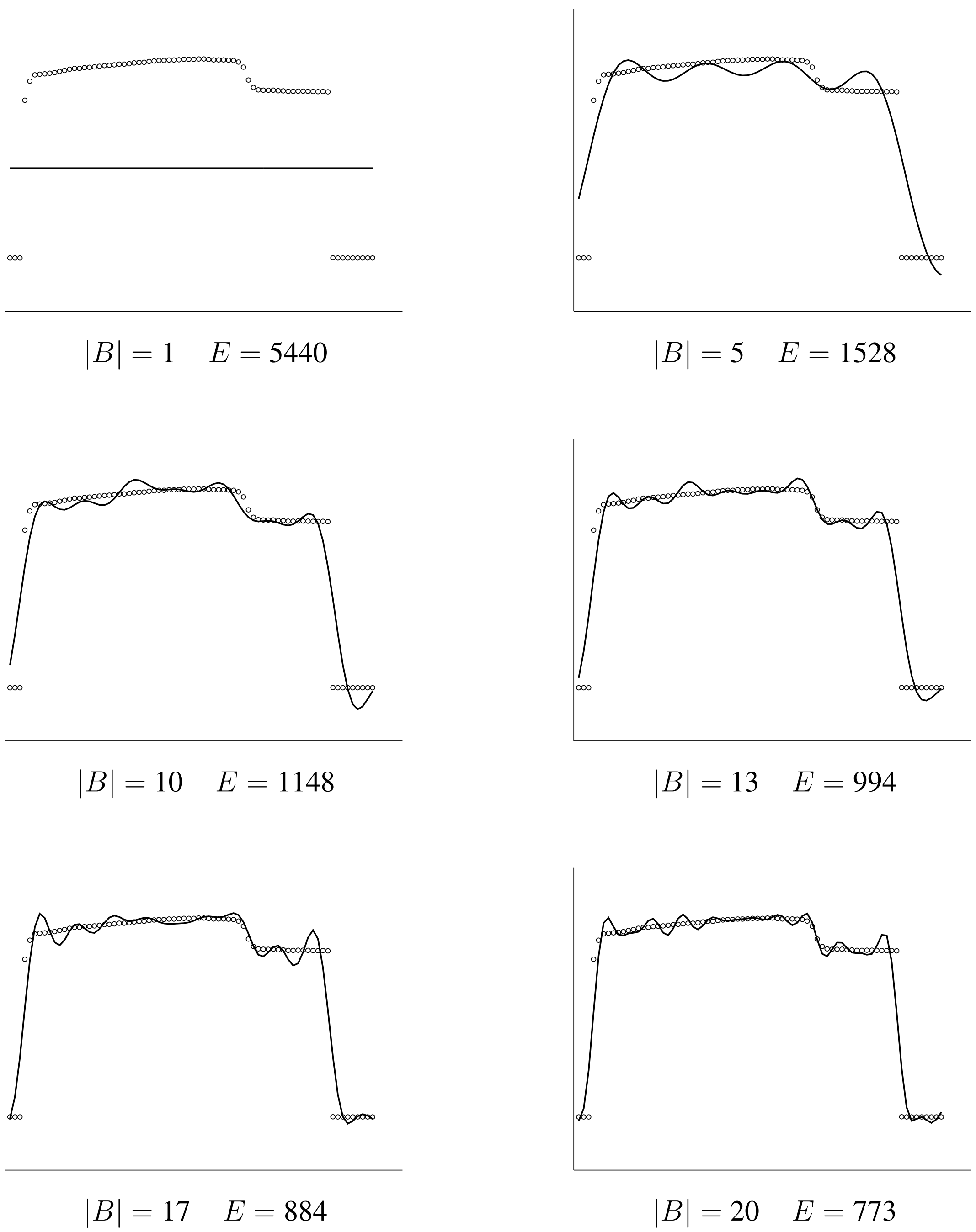


Figure 4.2 The Fourier transformation applied to the same data set for various sizes of B and a constant ordering of the frequencies. Each graph shows the input time-series data plotted with hollow bullets overlaid with a solid line representing the approximation generated by the Fourier transformation with the $|B|$ frequencies that contribute most to the approximation. The resulting sum of squared error E is indicated. Notice that the approximation better represents the data as the number of frequencies in B increases, resulting in a concomitant decrease in the sum of squared error E that diminishes as $|B|$ increases.

power of two less than or equal to the selected data set; if the longest data set is selected, then the length is fixed at the power of two greater than or equal to the longest data set. Once established, each data set can be made to conform to the fixed length either by truncation or extension.

The model selection component of the training phase must assess the tradeoff between the number of frequencies contained in B and the resulting sum of squared error E. Before such an analysis can be performed, the order in which the frequencies will be added to B must be established. For each size of B, the particular frequencies included will be determined by the ordering, thereby guaranteeing that the same set of frequencies is used to approximate each data set and, more importantly, ensuring that the features extracted from each data set measure the same characteristics. The frequencies are ranked according to their overall contribution to the approximation, from most to least, and are added to B in this order. As each frequency is added to B, the sum of squared error E decreases, but the magnitude of the decrease diminishes with each subsequent frequency. Figure 4.2 illustrates the approximation generated by the Fourier transformation for various sizes of B and a constant ordering of the frequencies. Notice that the approximation better represents the data as the size of B increases, resulting in a concomitant decrease in the sum of squared error E. Moreover, the decrease in the sum of squared error diminishes as the size of B increases.

The contribution of frequency j is equal to a combination of the weights assigned to the cosine and sine basis waveforms of frequency j. Let c_j^i be the contribution of frequency j towards the approximation of data set i. The value of c_j^i is computed as

$$c_j^i = \sqrt{(a_j^i)^2 + (b_j^i)^2}$$

where a_j^i and b_j^i are the weights associated with the cosine and sine basis functions, respectively, with frequency j as computed by the Fourier transformation applied to data set i in the training set. Let c_j be the overall contribution of frequency j for the entire training set. The value of c_j is computed as

$$c_j = \sum_i c_j^i$$

where i is a data set in the training set. The c_j values are ordered by magnitude, from largest to smallest. Frequencies are added to B according to the ordered c_j values: the frequency j with the largest c_j value is added first, the frequency j with the second largest c_j value is added second, and so on until the required number of frequencies have been added to B. Once the frequencies have been ordered, the model selection component of the training phase can determine an optimal number of frequencies for B that minimizes a weighted sum of the number of frequencies $|B|$ and the sum of squared error E. A procedure similar to that used in the training phase for the structure detectors (described in Section 3.4) can be employed: (1) compute the sum of squared error for each approximation generated with an incremental number of frequencies in B; (2) order the sum of squared error values according to the number of frequencies contained in B; (3) apply the straight structure detector to approximate the sequence of sum of squared error values with two straight structures; and (4) identify the optimal number of frequencies for B to be the point where the two extracted straight structures meet.

The outcome of the training phase is the fixed data length and the subset of frequencies B to use in approximating the data sets. The subset of frequencies B is determined by the optimal number of frequencies and the ordering established by the c_j values—e.g., if the optimal number of frequencies is five, then the subset of frequencies to use in approximating the data sets contains the five frequencies with the largest c_j values. The computational complexity of the training phase is $O(n^2 + rn \log n)$ when applied to a training set containing r data sets where n is the maximum length of the data sets within the training set.

4.2.3. Wavelet Transformation

The basis used by the Fourier transformation comprises waveforms which span an entire time-series data set, making such a basis appropriate for approximating a time series with pure periodic structure. Many time-series data sets, however, contain local structures which manifest in contiguous subsets of the data—e.g., sharp spikes. The wavelet transformation [19][39][68][71] employs a basis containing waveforms that are localized in space and, therefore, is more suitable for approximating time-series data sets containing regional structures.

The wavelet transformation by Press [71] uses a basis comprising n waveforms where n is the length of the data set under analysis. The basis waveforms are derived from scalings and translations of a mother wavelet ψ. The transformations are ordered according to the degree of localization in the resulting basis waveform such that ψ_j is the j^{th} transformation of ψ where $j = \{1, 2, \ldots, n\}$. Figure 4.3 plots a subset of the basis waveforms derived from the Daubechies 4-coefficient mother wavelet. The basis waveforms with j equal to 1, 3, 6, 10, 25, and 33 are shown. Notice that as j increases, the basis waveforms become more localized.

Given this basis, the wavelet transformation approximates a time series as

$$\hat{Y}(t) = \sum_{j \in B} \phi_j \psi_j(t)$$

where $\hat{Y}(t)$ is a time series which constitutes an approximation to $Y(t)$, B is a subset of the transformations of ψ within the basis, and ϕ_j is the coefficient (or weight) associated with basis waveform ψ_j. The sum of squared error E depends on the number of transformations contained in B: as the number of transformations in B increases, the sum of squared error E decreases. The features generated by the wavelet transformation for classification comprise the coefficients ϕ_j for each transformation j in B. The features extracted from a data set can be arranged into a feature vector having the form

$$\mathcal{A}_1 \; \mathcal{A}_2 \; \cdots \; \mathcal{A}_{|B|}$$

where $\mathcal{A}_i$ constitutes the i^{th} feature for each i such that $1 \leq i \leq |B|$ and $|B|$ is the number of transformations in B. In order to map feature i to transformation j, the transformations in B are ordered according to their overall contribution towards approximating the data sets within a training set, as determined during the training phase. (The training phase, discussed next, computes the overall contribution of each transformation, c_j, and orders the transformations according to these values, from highest to lowest.) $\mathcal{A}_1$ is equal to ϕ_j where $j \in B$ such that transformation j contributes the most (i.e., has the largest c_j value) towards approximating the data sets within a training set. $\mathcal{A}_2$ is equal to ϕ_j where $j \in B$ such that transformation j contributes the second most (i.e., has the second largest c_j value) towards approximating the data sets within a training set. And so on. The feature vector contains a total of $|B|$ features. See Appendix A for a concrete example. Applying

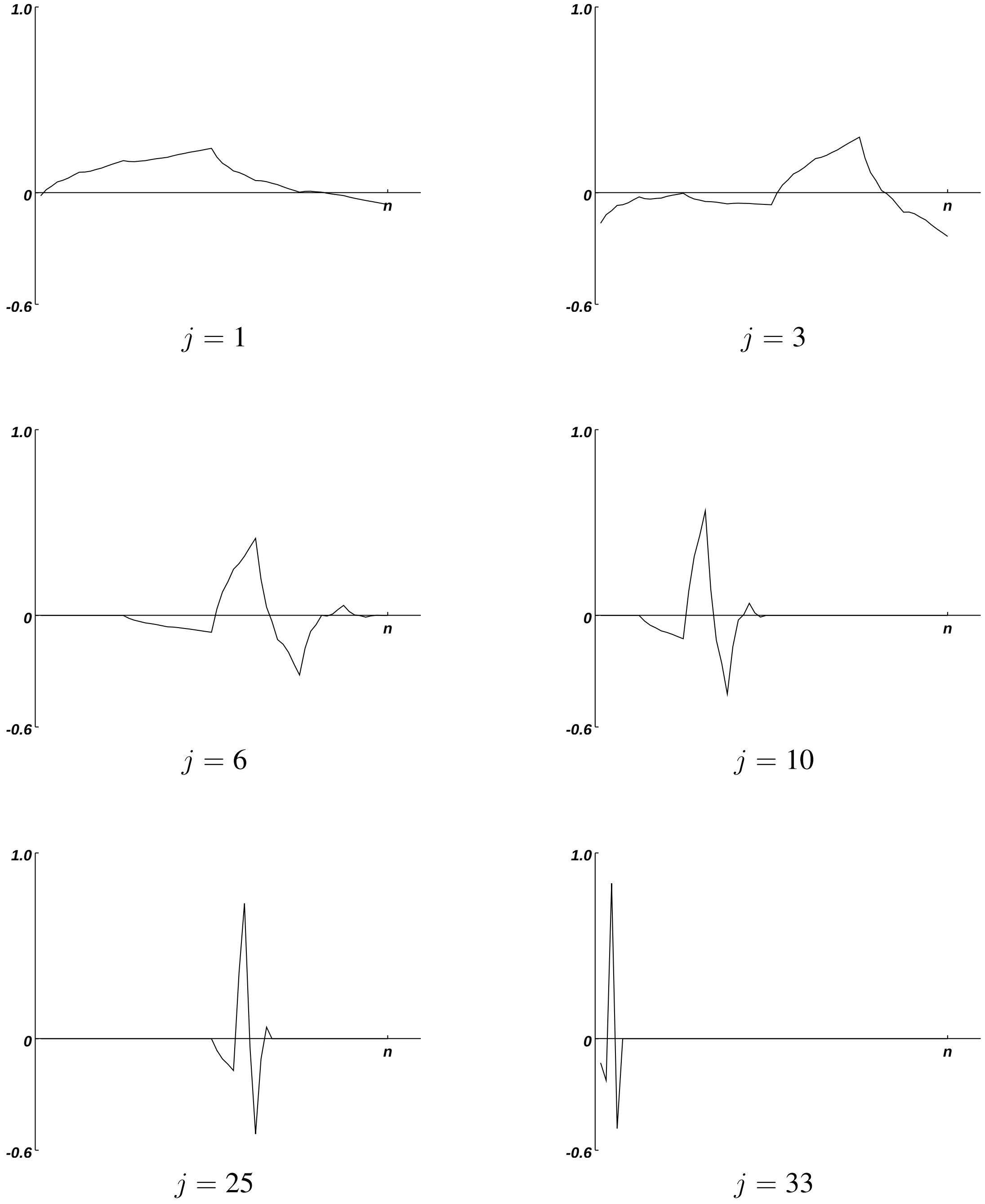


Figure 4.3 A subset of the basis waveforms used by the wavelet transformation derived from the Daubechies 4-coefficient mother wavelet. The basis waveforms are ordered according to their degree of localization and indexed by j where $j = \{0, 1, 2, \ldots, n\}$ and n is the length of the data set. The basis waveforms with j equal to 1, 3, 6, 10, 25, and 33 are shown. Notice that as j increases, the basis waveforms become more localized.

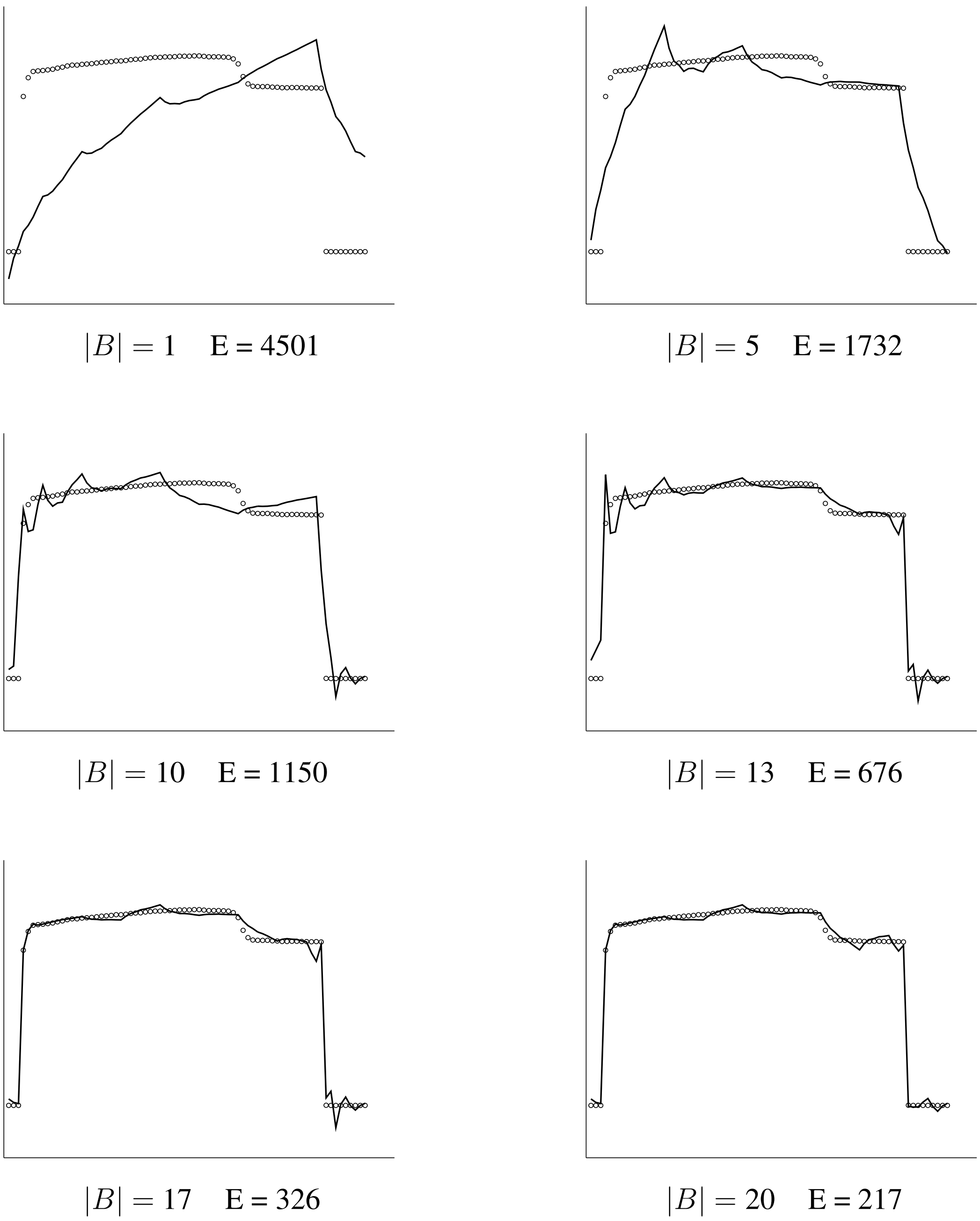


Figure 4.4 The wavelet transformation applied to the same data set for various sizes of B and a constant ordering of the transformations applied to the Daubechies 4-coefficient mother wavelet. Each graph shows the input time-series data plotted with hollow bullets overlaid with a solid line representing the approximation generated by the wavelet transformation with the $|B|$ transformations that contribute most to the approximation. The resulting sum of squared error E is indicated. Notice that the approximation better represents the data as the number of transformations in B increases, resulting in a concomitant decrease in the sum of squared error E that diminishes as $|B|$ increases.

the wavelet transformation to a data set of length n and assembling the extracted features has a computational complexity of $O(n \log n)$.

The training phase for the wavelet transformation feature extractor comprises both a data preparation and a model selection component applied to a training set. The wavelet transformation by Press [71] requires that the time-series data set has a length equal to a power of two. The data preparation component, therefore, must establish a uniform length equal to a power of two for all the data sets to be analyzed. This can be accomplished by selecting either the shortest or the longest data set within the training set: if the shortest data set is selected, then the length is fixed at the power of two less than or equal to the selected data set; if the longest data set is selected, then the length is fixed at the power of two greater than or equal to the longest data set. Once established, each data set can be made to conform to the fixed length either by truncation or extension.

The model selection component of the training phase must assess the tradeoff between the number of transformations contained in B and the resulting sum of squared error E. Before such an analysis can be performed, the order in which the transformations will be added to B must be established. For each size of B, the particular transformations included will be determined by the ordering, thereby guaranteeing that the same set of transformations is used to approximate each data set and, more importantly, ensuring that the features extracted from each data set measure the same characteristics. The transformations are ranked according to their overall contribution to the approximation, from most to least, and are added to B in this order. As each transformation is added to B, the sum of squared error E decreases, but the magnitude of the decrease diminishes with each subsequent transformation. Figure 4.4 illustrates the approximation generated by the wavelet transformation for various sizes of B and a constant ordering of the transformations applied to the Daubechies 4-coefficient mother wavelet. Notice that the approximation better represents the data as the size of B increases, resulting in a concomitant decrease in the sum of squared error E. Moreover, the decrease in the sum of squared error diminishes as the size of B increases.

The contribution of transformation j is equal to the weight assigned to the basis waveform ψ_j. Let c_j^i be the contribution of transformation j towards the approximation of data set i. The value of c_j^i is computed as

$$c_j^i = \phi_j^i$$

where ϕ_j^i is the weight associated with the basis waveform ψ_j as computed by the wavelet transformation applied to data set i in the training set. Let c_j be the overall contribution of transformation j for the entire training set. The value of c_j is computed as

$$c_j = \sum_i c_j^i$$

where i is a data set in the training set. The c_j values are ordered by magnitude, from largest to smallest. Transformations are added to B according to the ordered c_j values: the transformation j with the largest c_j value is added first, the transformation j with the second largest c_j value is added second, and so on until the required number of transformations have been added to B. Once the transformations have been ordered, the model selection component of the training phase can determine an optimal number of transformations for B that minimizes a weighted sum of the number of transformations $|B|$ and the sum of squared error E. A procedure similar to that used in the training phase for the structure detectors (described in Section 3.4) can be employed:

(1) compute the sum of squared error for each approximation generated with an incremental number of transformations in B; (2) order the sum of squared error values according to the number of transformations contained in B; (3) apply the straight structure detector to approximate the sequence of sum of squared error values with two straight structures; and (4) identify the optimal number of transformations for B to be the point where the two extracted straight structures meet.

The outcome of the training phase is the fixed data length and the subset of transformations B to use in approximating the data sets. The subset of transformations B is determined by the optimal number of transformations and the ordering established by the c_j values—e.g., if the optimal number of transformations is five, then the subset of transformations to use in approximating the data sets contains the five transformations with the largest c_j values. The computational complexity of the training phase is $O(n^2 + rn \log n)$ when applied to a training set containing r data sets where n is the maximum length of the data sets within the training set.

4.2.4. Dissimilarity to Structure Detectors

Each of the three statistical feature extractors—identity, Fourier, and wavelet transformations—generates features from a time-series data set using a methodology that differs from the others in terms of the basis used, the constraint placed on the length of the data set, the number of features extracted, and the computational complexity. The training phase for each statistical feature extractor must comprise both a data preparation and a model selection component that are appropriate for the associated feature extractor, thereby resulting in training phases that differ from one another in terms of their functionality and computational complexity. Each of the statistical feature extractors also differs from the structure detectors in terms of these same characteristics.

The structure detectors use a set of generally-useful morphologies as a basis for generating an approximation to a time-series data set (as discussed in Chapter 3). Given this basis, the structure detectors approximate a time series with a homogeneous sequence of structures (i.e., the same morphology type is used to approximate each subregion) as

$$\hat{Y}(t) = g(Y(t))$$

and with a composite, or heterogeneous, sequence of structures (i.e., the morphology type used to approximate each subregion varies among the subregions) as

$$\hat{Y}(t) = h(Y(t))$$

where the functions g and h are defined in Section 3.4. The sum of squared error E depends on the number of subregions p used in the approximation: as the number of subregions increases, the sum of squared error E decreases. The features generated by the structure detectors for classification include, for each subregion, the morphology type used to approximate the subregion, the values of the free parameters of the structure fitted to the subregion, the value of t at the onset of the subregion, the value of t at the offset of the subregion, and the length of the subregion. Additionally, the differences between the last $Y(t)$ value of a subregion and the first $Y(t)$ value of the next subregion for all consecutive pairs of subregions are also used as features. The features extracted from a data set can be arranged into a feature vector having the form

$$\mathcal{A}_1\ \mathcal{B}_1\ \mathcal{A}_2\ \mathcal{B}_2\ \cdots\ \mathcal{A}_{p-1}\ \mathcal{B}_{p-1}\ \mathcal{A}_p$$

where $\mathcal{A}_i$ comprises the features extracted from the i^{th} subregion and $\mathcal{B}_i$ is the difference between the last $Y(t)$ value of subregion i and the first $Y(t)$ value of subregion $i+1$. The extracted features in $\mathcal{A}_i$ have the form

$$\mathcal{C}_i \; \mathcal{D}_i \; \mathcal{E}_i \; \mathcal{F}_i \; \mathcal{G}_i$$

where $\mathcal{C}_i$ is the type of morphology fitted to subregion i, $\mathcal{D}_i$ comprises the free parameters of the structure fitted to subregion i, $\mathcal{E}_i$ is the value of t at the onset of subregion i, $\mathcal{F}_i$ is the value of t at the offset of subregion i, and $\mathcal{G}_i$ is the length of subregion i. The extracted features in $\mathcal{D}_i$ have the form

$$\mathcal{H}_i \; \mathcal{I}_i \; \mathcal{J}_i \; \mathcal{K}_i \; \mathcal{L}_i \; \mathcal{M}_i \; \mathcal{N}_i \; \mathcal{O}_i \; \mathcal{P}_i \; \mathcal{Q}_i \; \mathcal{R}_i \; \mathcal{S}_i \; \mathcal{T}_i \; \mathcal{U}_i \; \mathcal{V}_i$$

where $\mathcal{H}_i$ is equal to the free parameter a of the constant structure type fitted to subregion i; $\mathcal{I}_i$ and $\mathcal{J}_i$ are equal to the free parameters a and b of the straight structure type fitted to subregion i; $\mathcal{K}_i$, $\mathcal{L}_i$, and $\mathcal{M}_i$ are equal to the free parameters a, b, and c of the exponential structure fitted to subregion i; $\mathcal{N}_i$, $\mathcal{O}_i$, and $\mathcal{P}_i$ are equal to the free parameters a, b, and c of the sinusoidal structure fitted to subregion i; $\mathcal{Q}_i$, $\mathcal{R}_i$, and $\mathcal{S}_i$ are equal to the free parameters a, b, and c of the triangular structure fitted to subregion i; and $\mathcal{T}_i$, $\mathcal{U}_i$, and $\mathcal{V}_i$ are equal to the free parameters a, b, and c of the trapezoidal structure fitted to subregion i. Even though only one morphology type is ever used to approximate a subregion, it is necessary to include a placeholder in the feature vector for each free parameter for each possible structure to ensure that the k^{th} feature always has the same meaning (i.e., represents the same characteristic of the data). Without placeholders, the k^{th} feature might represent the free parameter a for a constant structure in one feature vector, and represent the free parameter b for a straight structure in another. Features for structures not fitted to subregion i are set to zero. The consequence of these placeholders is an excessively long feature vector: using the structure detectors to approximate a data set using p subregions results in a feature vector with $20p-1$ features. See Appendix A for a concrete example. Applying the structure detectors to a data set of length n and assembling the extracted features has a computational complexity of $O(n^3)$.

Table 4.1 summarizes the characteristics of the methodologies used by the statistical feature extractors and the structure detectors to extract features from a time-series data set. The identity transformation can be used to extract elementary features from a time series and, consequently, is simple to apply and has a correspondingly small computational complexity. However, the features extracted by the identity transformation ignore abstract behavior that manifests across consecutive data points in a time series. Higher-order features are derived from multiple elementary features and, as such, capture behavior that is not explicitly present in the raw data. Both the Fourier and wavelet transformations extract features which capture such behavior, but each uses a different basis for approximation. The basis waveforms used by the Fourier transformation comprise sine and cosine functions that span the time series, making the technique most effective for extracting global features and least effective for extracting local features that manifest in contiguous subsets of the data. The wavelet transformation employs basis waveforms generated by scaling and translating a mother wavelet, thereby enabling the technique to extract features that are both global and local within a time series. The basis waveforms, however, are derived from a single morphology as defined by the mother wavelet. The structure detectors differ from the statistical feature extractors

	Identity Transformation	Fourier Transformation	Wavelet Transformation	Structure Detectors
Basis for Approximation	unit vectors	sine and cosine waveforms	transformations of a mother wavelet	generally-useful morphologies
Data Length Constraint	none	power of 2	power of 2	none
Number of Features Extracted	n	$2 * \|B\|$	$\|B\|$	$20p - 1$
Computational Complexity	$O(n)$	$O(n \log n)$	$O(n \log n)$	$O(n^3)$
Main Strength	simple	extracts global features	extracts both global and local features	basis comprises various morphologies
Main Weakness	ignores higher-order features	ignores local features	basis derived from a single morphology	computational complexity

Table 4.1 A summary of the characteristics of the methodologies used by the statistical feature extractors and the structure detectors to extract features from a time-series data set of length n. For the Fourier and wavelet transformations, $|B|$ refers to the number of basis waveforms allowed to contribute to the approximation. For the structure detectors, p refers to the number of subregions used in the approximation.

in that the basis comprises various morphologies, but suffers from a computational complexity that exceeds that of the other techniques.

The training phase for the structure detectors comprises both a data preparation and a model selection component applied to a training set. Since the structure detectors can analyze data of any length, the data preparation component performs no function. The model selection component of the training phase must assess the tradeoff between the number of subregions p and the resulting sum of squared error E. The procedure for determining an optimal number of subregions p is discussed in Section 3.4. The outcome of the training phase is the number of subregions p to use in approximating the data sets. The computational complexity of the training phase is $O(rn^3)$ when applied to a training set containing r data sets where n is the maximum length of the data sets within the training set.

Table 4.2 summarizes the characteristics of the training phase associated with each of the statistical feature extractors and the structure detectors applied to a training set containing r data sets with maximum length n. The outcome of the training phase establishes a fixed number of features to be extracted and depends on the associated feature extraction methodology: a uniform data length n' for the identity, Fourier, and wavelet transformations; a subset of basis waveforms

	Identity Transformation	**Fourier Transformation**	**Wavelet Transformation**	**Structure Detectors**
Data Preparation	establish uniform data length	establish uniform data length equal to a power of 2	establish uniform data length equal to a power of 2	none
Model Selection	none	optimize number of basis functions versus error	optimize number of basis functions versus error	optimize number of subregions versus error
Computational Complexity	$O(rn)$	$O(n^2 + rn\log n)$	$O(n^2 + rn\log n)$	$O(rn^3)$
Outcome	n'	n' and B	n' and B	p

Table 4.2 A summary of the characteristics of the training phase associated with each of the statistical feature extractors and the structure detectors applied to a training set containing r data sets with maximum length n. The outcome of the training phase depends on the associated feature extraction methodology: a uniform data length n' for the identity, Fourier, and wavelet transformations; a subset of basis waveforms B for the Fourier and wavelet transformations; and a number of subregions p for the structure detectors.

B for the Fourier and wavelet transformations; and a number of subregions p for the structure detectors. The value of n' determines the length to which each data set will either be truncated or extended before analysis by the identity, Fourier, or wavelet transformations and, therefore, becomes the value of n for the statistical feature extractors in Table 4.1.

Each of the feature extraction techniques—the three statistical feature extractors and the structure detectors—employs a different methodology for extracting features from a time-series data set. A separate training phase associated with each feature extractor determines a fixed number of features to extract based on a training set. A trained feature extractor, therefore, can be used to extract the same number of features from each data set under analysis, generating a fixed-length feature vector representing each data set. A statistical classifier can be used to discriminate among data sets having unique group affiliations based upon their associated fixed-length feature vectors, resulting in a classification accuracy that details the number of data sets correctly classified. The specific features used for discrimination directly influence the classification accuracy: features which truly discriminate among groups will increase accuracy, while the lack of such features will decrease accuracy. Since the dissimilarities among the feature extraction methodologies will cause a different set of features to be extracted by each feature extractor, the classification accuracy achieved when using features extracted by the various methodologies will also differ. In order to evaluate the relative efficacy of each feature extractor, an experiment that compares the classification accuracies achieved when using features extracted by the various methodologies under a range of conditions can be conducted and used to assess the performance of the structure detectors versus the benchmark comprising the statistical feature extractors.

4.3. Experiment Design

The efficacy of the structure detectors for generalized feature extraction can be evaluated by comparing the relative classification accuracies achieved when using features extracted by the structure detectors and the statistical methods under a range of conditions. An experiment to perform such an evaluation requires a database containing a collection of labeled data sets so that the resulting classification accuracy can be computed. For simplicity, the data sets are affiliated with one of two groups: normal or abnormal. To evaluate each feature extractor, the experiment comprises several steps: the training phase associated with the feature extractor is applied to a training set containing a subset of data sets randomly selected from the database, the trained feature extractor is used to generate features from each data set in the database, the features extracted from each data set are assembled into a feature vector, a statistical classifier is used to discriminate among the normal and abnormal data sets based upon their associated feature vectors, and the resulting classification accuracy is reported. Such an experiment includes several factors other than the feature extraction method that may influence the outcome of the training phase. The experimental factors and the levels (or values) for each are as follows:

Feature extraction method. The feature extraction methods include the three statistical methods used as a baseline, the six structure-detector methods applied individually (i.e., generating a homogeneous approximation), and the structure-detector methods applied in combination (i.e., generating a composite, or heterogeneous, approximation). The ten values for this factor are identity, Fourier, wavelet, constant, straight, exponential, sinusoidal, triangular, trapezoidal, and composite.

Training set size. A range of training set sizes is used to determine whether the number of data sets used to train each feature extraction method influences the classification accuracy. The four values for the training set size r are 2, 4, 8, and 16. These values span the sizes for the training set that are of greatest interest: 2 is the minimum number of data sets that should be used for training, and 16 is the maximum number that can be processed in a reasonable amount of time by the training phase associated with the structure detectors due to its computational complexity.

Composition of training set. The distribution of the labels of the data sets in the training set will affect the outcome of the training phase and, by extension, the resulting classification accuracy. Given a database comprising data sets labeled as normal or abnormal, a training set containing only data sets with normal labels will cause the outcome of the training phase to prefer features that better approximate normal data sets, while a training set containing data sets with an equal number of both normal and abnormal labels will cause the outcome of the training phase to compromise and include features that are useful for approximating both groups. To investigate the effect of training set composition on classification accuracy, the training set is randomly selected from the database so as to contain data sets with a mixture of normal and abnormal labels. The composition of the training set w refers to the number of abnormal data sets in the training set. The five values of w are 0, 1, 2, 4, and 8. These values were selected in relation to the values of the training set size r so as to span an interesting range of label distributions. Assuming that the data sets labeled as normal manifest a lesser degree of variability than those labeled as abnormal (since many types of abnormalities are included within the group), it would be more desirable to produce features which better approximate the normal data sets. To prevent the abnormal data sets from constituting a majority of the training set which would cause the training phase to prefer features which better approximate abnormal data sets, the maximum allowable value of w is $\frac{1}{2}r$.

Data preprocessing. This experimental factor specifies the procedure for modifying the data length during the data preprocessing component of the training phase. The data preprocessing component establishes a fixed length for all data sets by selecting either the shortest or the longest data set in the training set, and then using truncation or extension methods to modify each data set to conform to the expected length. When a data set must be extended to meet the required length, zero padding is used because of its simplicity and domain independence. Assuming that selecting the shortest data set primarily requires truncation and selecting the longest data set mainly requires zero padding, the two values of this factor are truncation and padding.

These four factors result in an experiment with a $10 \times 4 \times 5 \times 2$ factorial design. Subtracting the unrealizable combinations (due to the restriction on the composition of the training set and the lack of data preprocessing for the structure detectors) from the 400 possible combinations leaves 182 allowable combinations of experimental factors.

For each combination of experimental factors, the experiment proceeds as follows:

1. Construct the training set. Randomly select r data sets from the database such that w data sets have labels of abnormal and $r - w$ data sets have labels of normal.

2. Train the feature extraction method. Perform the training phase associated with the feature extraction method (as discussed in Section 4.2). For the identity, Fourier, and wavelet transformations, a uniform data length is established using the technique specified by the value of the data preprocessing factor. For the wavelet transformation, the basis derived from the Daubechies 4-coefficient mother wavelet is used.

3. Extract features from the data sets. Apply the trained feature extraction method to each data set in the database as specified by the parameters determined by the outcome of the training phase.

4. Assemble the features extracted from each data set into a feature vector. The form of the feature vector for each feature extraction method is as discussed in Section 4.2. Each feature vector has the same known group label as its associated data set.

5. Classify the collection of feature vectors. Use a statistical classifier to assign a group label to each feature vector.

6. Evaluate the classification accuracy. Compare the assigned and known group labels for each feature vector, and compute the overall classification accuracy across all feature vectors.

A flowchart of the experiment procedure is shown in Figure 4.5. The numbered activities in the figure refer to the individual steps in the experiment (as described above). For a particular combination of experimental factors, the end result of performing the experiment procedure is a measure of the resulting classification accuracy.

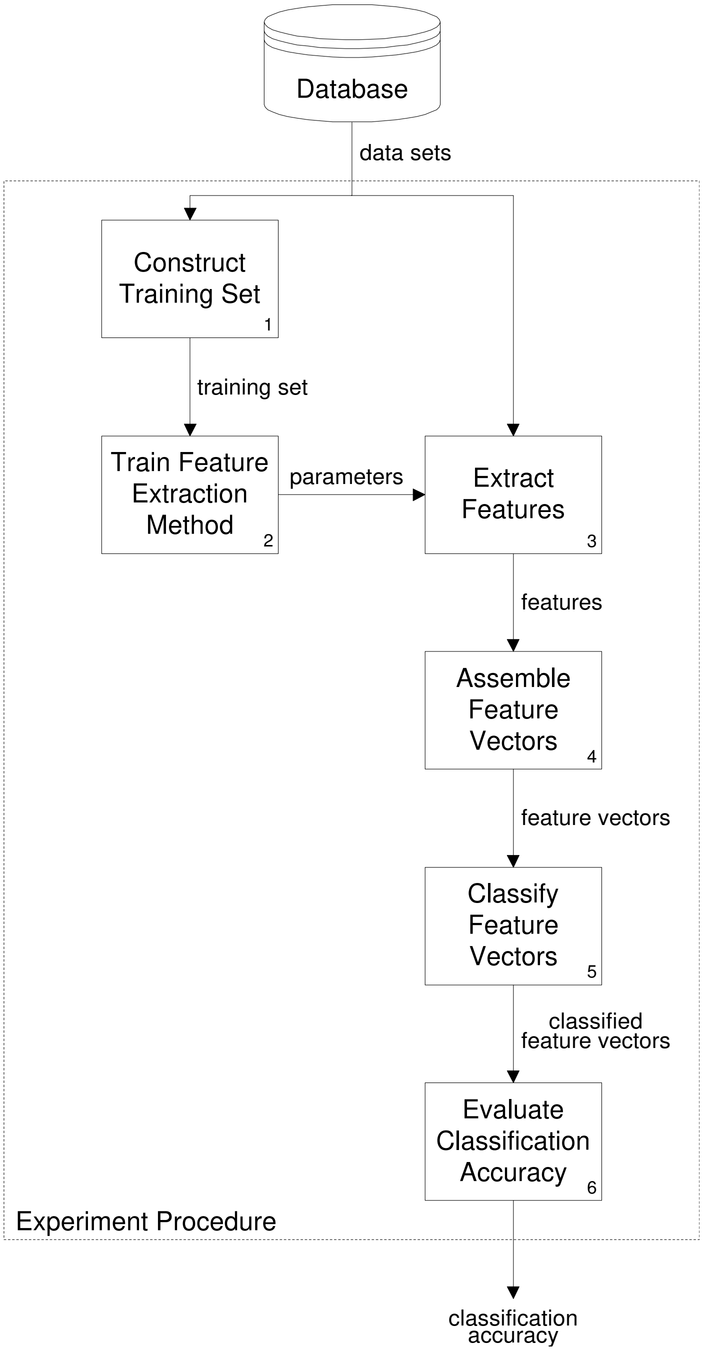


Figure 4.5 A flowchart of the experiment procedure. Referring to the numbered activities in the figure, the experiment procedure comprises several steps: (1) a training set is assembled, (2) the feature extraction method is trained, (3) features are extracted as specified by the parameters determined by the outcome of the training phase, (4) the features extracted from each data set are assembled into a feature vector, (5) classification is performed on the feature vectors, and (6) the classification accuracy is evaluated.

	Predicted	
Known	Normal	Abnormal
Normal	a	b
Abnormal	c	d

Figure 4.6 A confusion matrix generated by CART reporting the classification results. The confusion matrix is a two-by-two grid consisting of one column for each of the known classes (i.e., normal and abnormal), and one row for each of the predicted classes (i.e., normal and abnormal). The values a, b, c, and d are the number of feature vectors—and, by extension, the number of data sets—falling into each category. The value a is the number of data sets known to be normal that were classified as normal. The value d is the number of data sets known to be abnormal that were classified as abnormal.

4.3.1. Classification Accuracy

Since the purpose of the experiment is to assess the effect of the experimental factors on classification accuracy as opposed to evaluating the efficacy of various classifiers using a particular set of features, only one classifier is necessary for the experiment. The decision-tree classifier CART [13] was selected for the experiment over other techniques (e.g., linear discriminant analysis) because CART uses the available data judiciously, generates a classification tree that is easy to interpret as decision rules, has a demonstrated resistance to the curse of dimensionality for regression [8] (which should hold for classification as well), and makes no assumptions about the feature vectors within each group (e.g., linear discriminant analysis discriminates best when the features vectors within each group are normally distributed with the same covariance matrix). CART uses a sequence of univariate (i.e., one-feature) partitionings to subdivide the collection of feature vectors into clusters such that all feature vectors within the same cluster are predicted to belong to the same class (in this case, normal or abnormal). The default parameters were used to run CART (e.g., use ten-fold cross-validation, split nodes based on the gini diversity index) for the experiment.

The classification results generated by CART are reported as a confusion matrix having the form shown in Figure 4.6. The confusion matrix is a two-by-two grid consisting of one column for each of the known classes (i.e., normal and abnormal), and one row for each of the predicted classes (i.e., normal and abnormal). The values a, b, c, and d are the number of feature vectors—and, by extension, the number of data sets—falling into each category. The value a is the number of data sets known to be normal that were classified as normal. The value d is the number of data sets known to be abnormal that were classified as abnormal. The classification accuracy can be summarized with a pair of percentages: the percent of data sets known to be normal that were classified as normal ($\frac{a}{a+b} * 100$), and the percent of data sets known to be abnormal that were classified as abnormal ($\frac{d}{c+d} * 100$).

To compensate for the effect on the classification accuracy due to the random selection of the training set, twenty iterations of the experiment procedure are performed for each combination of experimental factors—requiring a total of $182 \times 20 = 3640$ experimental iterations per database—and the mean and standard deviation of the percentage pairs over the twenty iterations are computed. The overall classification accuracy for each combination of experimental factors is reported as

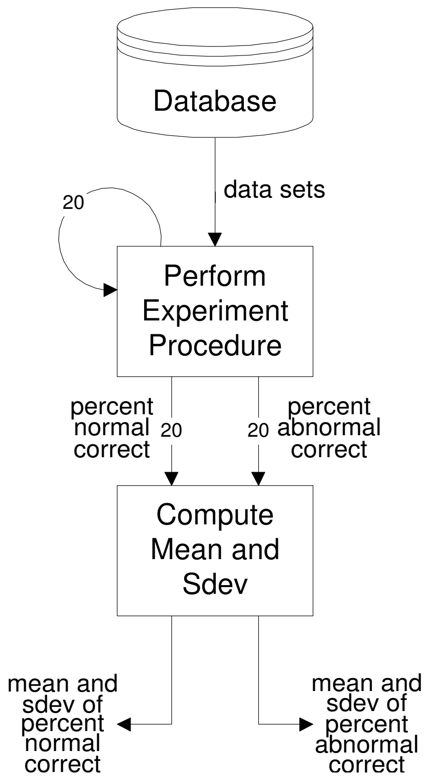


Figure 4.7 The methodology used to compute the overall classification accuracy for each combination of experimental factors. To compensate for the effect on the classification accuracy due to the random selection of the training set, the experiment procedure is repeated twenty times for each combination of experimental factors. The mean and standard deviation of the percent of normal data sets correctly classified and the percent of abnormal data sets correctly classified represent the overall classification accuracy.

two pairs of values: the mean and standard deviation of the percent of known normal data sets classified as normal, and the mean and standard deviation of the percent of known abnormal data sets classified as abnormal. Figure 4.7 sketches the methodology used to compute the overall classification accuracy for each combination of experimental factors.

4.3.2. Databases

Two different domains serve as a source of time-series data sets for the experiment, namely semiconductor microelectronics fabrication and electrocardiography. A collection of in-line process-control measurements recorded from various sensors during the processing of silicon wafers for semiconductor fabrication constitute the wafer database; each data set in the wafer database contains the measurements recorded by one sensor during the processing of one wafer by one tool. The electrocardiogram (ECG) database contains measurements of cardiac electrical activity as recorded from electrodes at various locations on the body; each data set in the ECG database contains the

	Normal Data Sets	Abnormal Data Sets
Wafer Database	1067	127
ECG Database	133	67

Table 4.3 The number of normal and abnormal data sets contained in the wafer and ECG databases. There are a total of 1194 data sets in the wafer database, and a total of 200 data sets in the ECG database.

measurements recorded by one electrode during one heartbeat. The data sets contained in each database were analyzed by appropriate domain experts, and a label of normal or abnormal was assigned to each data set. Of the 1194 data sets in the wafer database, 1067 data sets were identified as normal and 127 data sets were identified as abnormal. The ECG database contains 200 data sets where 133 were identified as normal and 67 were identified as abnormal. Table 4.3 summarizes the contents of both databases.

Semiconductor microelectronics are manufactured using a complex process during which layers of various materials are applied to a silicon wafer and selectively removed to define circuit elements on the wafer [2][80]. This procedure is called etching. The most common method used for etching starts by applying a mask using photolithography: a photosensitive resin, called positive photoresist, is coated onto the surface of the wafer and selectively exposed to ultraviolet (UV) light under a mask detailing the desired circuit pattern. A chemical solution designed to remove the portion of the photoresist exposed to the UV light is then applied, leaving behind only the photoresist that protects the areas needed to construct the circuit elements (i.e., the non-circuit areas are left unprotected). After the photolithography process, the wafer is placed in a vacuum chamber and exposed to a reactive plasma (i.e., an energized gas) designed to remove the regions of the wafer which are unprotected by the photoresist (i.e., the non-circuit areas). When the unprotected areas have been completely removed, the gasses in the plasma chamber change in stoichiometry (i.e., chemical properties and composition) because of a decrease in both the reaction byproducts and the consumption of the reactant gasses. This change can be sensed by monitoring the light emission from the plasma, acting as an indicator for terminating exposure of the wafer to the plasma. This is commonly called the etch endpoint.

The process of manufacturing semiconductor microelectronics involves over 250 processing steps, any of which can result in degraded performance and reliability, reduced yield, or even scrappage of microelectronics if the manufacturing tools operate outside of allowable tolerances [2]. Very complex semiconductor manufacturing tools may have up to 450 in-line sensors which can be monitored during the etch process for quality control. Each tool has its own special set of parameters which best reflect the state of the manufacturing process, the wafer, and the tool. In the case of the tool used for this study, six parameters have been identified by domain experts as being critical for monitoring purposes: radio frequency forward power, radio frequency reflected power, chamber pressure, 405 nanometer (nm) emission, 520 nanometer (nm) emission, and direct current bias. The first two parameters are measures of electrical power applied to the plasma, the third parameter measures the pressure within the etch chamber, the fourth and fifth parameters measure the intensity of two different wavelengths (i.e., colors) of light emitted by the plasma, and

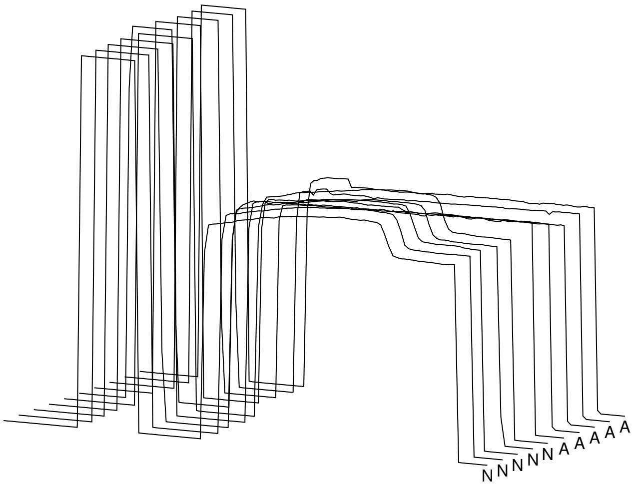


Figure 4.8 Examples of normal and abnormal data sets for the 405 nm parameter in the wafer database. The waveforms labeled with “N” are graphical representations of five normal data sets, and the waveforms labeled with “A” are graphical representations of five abnormal data sets. These data sets were recorded simultaneously with those shown in Figure 4.9.

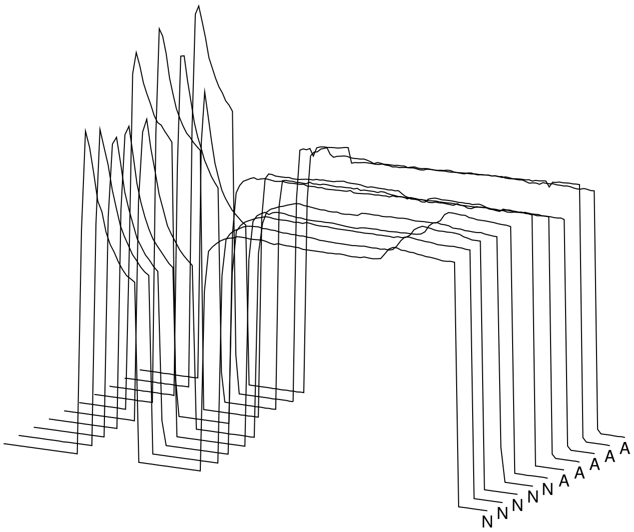


Figure 4.9 Examples of normal and abnormal data sets for the 520 nm parameter in the wafer database. The waveforms labeled with “N” are graphical representations of five normal data sets, and the waveforms labeled with “A” are graphical representations of five abnormal data sets. These data sets were recorded simultaneously with those shown in Figure 4.8.

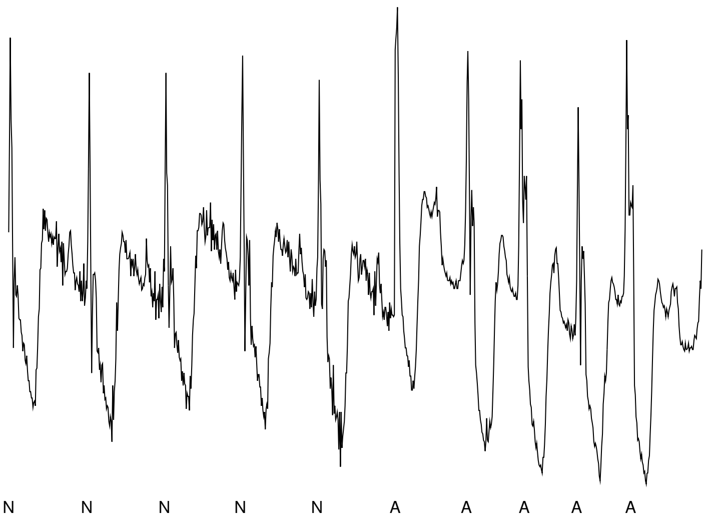


Figure 4.10 Examples of normal and abnormal data sets for the lead 0 parameter in the ECG database. The graphical representation of five normal and five abnormal data sets are shown concatenated into one continuous waveform. A label appears at the onset of each data set within the waveform: an "N" indicates the start of a normal data set, and an "A" indicates the start of an abnormal data set. These data sets were recorded simultaneously with those shown in Figure 4.11.

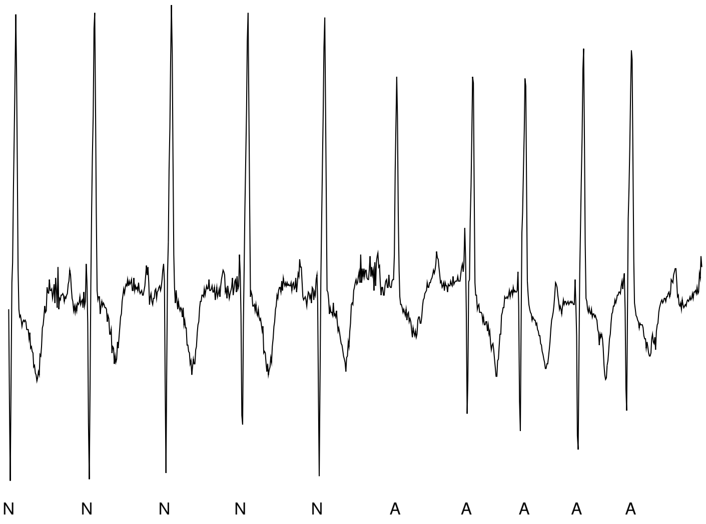


Figure 4.11 Examples of normal and abnormal data sets for the lead 1 parameter in the ECG database. The graphical representation of five normal and five abnormal data sets are shown concatenated into one continuous waveform. A label appears at the onset of each data set within the waveform: an "N" indicates the start of a normal data set, and an "A" indicates the start of an abnormal data set. These data sets were recorded simultaneously with those shown in Figure 4.10.

the sixth parameter measures the direct current electrical potential difference within the tool. Each data set contains the measurement values from one sensor for one wafer. Previous experiments [22] have shown that the most accurate classification of wafer data sets can be accomplished using the 405 nm and 520 nm parameters. Figures 4.8 and 4.9 show graphical representations of normal and abnormal data sets for the 405 nm and 520 nm parameters, respectively, which were collected simultaneously and are representative of the same sequence of wafers. The collection of data sets for each of these two parameters constitute two individual subsets from the wafer database, and will be used separately as sources of data for the experiment.

Electrocardiography [17][36] is a diagnostic procedure that monitors the electrical activity of the heart with the intention of diagnosing cardiac pathologies. An electrocardiogram (ECG) is generated by placing one or more electrodes at standardized locations on the body, and recording the electrical potential difference observed at that site during each heartbeat; a complete ECG utilizes twelve electrodes, but fewer are often used for simpler diagnostic procedures. Electrocardiographic monitoring is often performed in a resting state, but can also be performed under ambulatory conditions using a special recording device called a Holter monitor [44][50]. Long-term monitoring of cardiac behavior is possible with a Holter ECG recorder, but at the expense of data that is noisier than that which can be collected under resting conditions. The ECG database for the experiment was drawn from the Supraventricular Arrhythmia Database (SVDB) [40] which is available on the MIT-BIH Arrhythmia Database CD-ROM [64] and the affiliated on-line repository PhysioNet [34]. The SVDB contains 78 Holter ECG recordings where each electrocardiogram was recorded from a single patient for a duration of approximately thirty minutes. One ECG was selected at random and the portions of the recording representative of heartbeats with the most prevalent abnormality—supraventricular premature beat—were extracted along with a random sample of the portions of the recording representative of normal heartbeats. Two leads, called lead 0 and lead 1, were used to record this ECG, and each data set contains the measurement values from one lead for one heartbeat.[2] Figures 4.10 and 4.11 show graphical representations of normal and abnormal data sets for the lead 0 and lead 1 parameters, respectively, which were collected simultaneously and are representative of the same sequence of heartbeats. The collection of data sets for each of these two parameters constitute two individual subsets from the ECG database, and will be used separately as sources of data for the experiment.

Table 4.4 lists some elementary characteristics of the normal and abnormal data sets contained both in the wafer database for the 405 nm and 520 nm parameters and in the ECG database for the lead 0 and lead 1 parameters. The statistics describe the ranges of measurement values contained in the data sets as well as the lengths of the data sets. Several observations are apparent from the table: (1) the wafer data sets tend to contain longer data sets; (2) the wafer data sets contain measurement values with a much larger range than those in the ECG data sets; (3) there is only a small difference in the ranges of measurement values between the lead 0 and lead 1 parameters, but the corresponding difference in the wafer database is very large; (4) there is little difference

[2]The correspondence between lead 0 and lead 1 in the database and the standardized leads used in electrocardiography is not recorded in the documentation for the Supraventricular Arrhythmia Database. However, the laboratory in which these data were collected regularly used a modified lead II (which is roughly parallel to limb lead II but with the electrodes on the chest) to record lead 0 in the database, and chest lead V_1 to record lead 1 in the database [63]. So, it would not be unreasonable to assume that this correspondence holds for the data used in this research. See Constant [17] or Goldman [36] for additional information about the standardized leads used in electrocardiography.

Wafer Database		Range Min	Range Max	Range Mean	Length Min	Length Max	Length Mean
405 nm	Normal	2045	2049	2046	114	152	136
	Abnormal	2045	2047	2046	104	198	144
520 nm	Normal	755	1185	1011	114	152	136
	Abnormal	778	1408	1069	104	198	144

ECG Database		Range Min	Range Max	Range Mean	Length Min	Length Max	Length Mean
Lead 0	Normal	360	720	490	61	116	94
	Abnormal	302	719	548	39	152	81
Lead 1	Normal	392	676	542	61	116	94
	Abnormal	112	692	462	39	152	81

Table 4.4 Characteristics of the normal and abnormal data sets for each parameter in the wafer and ECG databases. The minimum, maximum, and mean range of the measurement values contained in the data sets are given. The minimum, maximum, and mean length of the data sets are also given.

in terms of lengths between the lead 0 and lead 1 parameters as well as between the 405 nm and 520 nm parameters; and (5) comparing the normal and abnormal data sets within each parameter suggests that the differences in lengths may assist in discrimination, but the differences in ranges is overall less useful.

Besides being generated by completely different physical processes, the characteristics listed in Table 4.4 suggest that the wafer and ECG databases contain data that are dissimilar. However, these statistics do not address differences in the structural composition of the data. To evaluate the structural differences between each pair of parameters, an approach based on chain codes [28][49] can be employed. Chain codes were originally introduced by Freeman [28] as a methodology for encoding arbitrary curves as a sequence of symbols: a grid is used to digitize a continuous curve into a series of discrete points, and a label is assigned to each pair of successive points based on the spatial relationship between them. Figure 4.12 shows an example application of chain codes using a typical grid and labeling scheme that differentiates among eight spatial relationships between successive digitized points.

A modified version of Freeman's chain codes can be used to analyze the differences in the structural composition between each pair of parameters in the wafer and ECG databases. The modified chain code methodology is applied to an individual data set as follows: compute the magnitude of the slope between each consecutive pair of values, normalize the slopes so that the values range between zero and one, assign a label to each slope based upon its normalized value, and total the number of normalized slope values assigned with each unique label. Since the normalized slope values can vary arbitrarily between zero and one, the encoding scheme must assign symbols based on continuous ranges of normalized slope values. The strategy for assigning labels to slopes for the modified chain code methodology is shown in Table 4.5. A log-like scale was used to assign labels so as to elucidate the distribution of slopes; uniform ranges would have

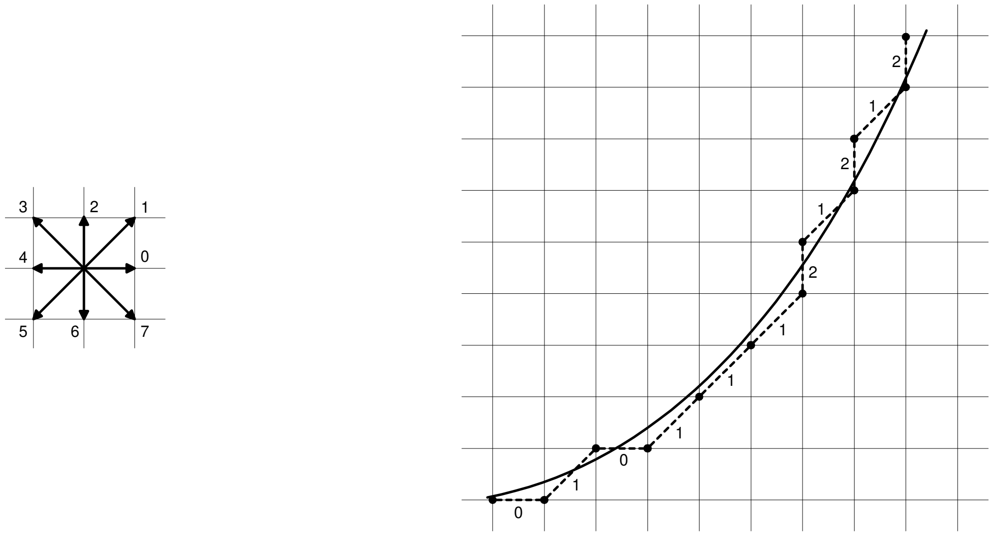


Figure 4.12 An example application of chain codes. A typical encoding scheme is shown on the left: each of the eight different spatial relationships between successive digitized points is assigned a unique symbol. This encoding scheme is applied to the curve on the right: the continuous curve (plotted with a solid line) is digitized (plotted with solid bullets connected by a dashed line), and each successive pair of points is assigned a label according to the encoding scheme. The chain code representation of this curve is the sequence 01011121212.

Label	Assignment Rule
A	$slope \leq 0.0001$
B	$0.0001 < slope \leq 0.0005$
C	$0.0005 < slope \leq 0.0020$
D	$0.0020 < slope \leq 0.0085$
E	$0.0085 < slope \leq 0.0340$
F	$0.0340 < slope \leq 0.1365$
G	$0.1365 < slope \leq 0.5460$
H	$0.5460 < slope \leq 1.0000$

Table 4.5 The encoding scheme used for the modified chain code methodology to assign labels to slope values. Each label has an associated assignment rule expressed as a range of values. A slope is assigned the label whose assignment rule includes the normalized slope value.

resulted in a similar outcome, with the exception that the smaller slopes with different labels under the log-like scale would have all been assigned the same label. To compute the distribution of labels for each parameter, apply the modified chain code methodology to each data set for the parameter, add the total number of normalized slope values assigned with each unique label for all data sets, and compute the percent of the total number of normalized slope values assigned with each unique label. The distribution of the percent of normalized slope values across the labels indicates how frequently and radically the measurement values change over time and, moreover, provides a basis for comparing the structural composition among parameters.

Figure 4.13 plots the distribution for each parameter in the two databases: the percent of slope values assigned with each of the eight labels is shown, where the shaded bars represent the distribution for the parameters in the wafer database and the patterned bars represent the distribution for the parameters in the ECG database. The distributions for the two parameters from the wafer database are skewed to the left, indicating that a majority of the slope values are small. Therefore, it can be concluded that the wafer database comprises data sets containing values that do not change often or abruptly over time. The distributions for the two parameters from the ECG database are skewed to the right, indicating that a majority of the slope values are large. Therefore, it can be concluded that the ECG database comprises data sets containing values that consistently undergo large shifts over time. This disparity between the distributions demonstrates that the two databases contain data that are markedly different from each other.

4.3.3. Computational Effort

There are 182 combinations of experimental factors, each of which is repeated 20 times, resulting in a total of 3640 iterations of the experiment procedure per parameter. For the four parameters contained in the wafer and ECG databases, a grand total of 14,560 iterations of the experiment procedure are necessary. It is possible, however, to reduce the amount of computational effort required to perform the experiment. Such a reduction can be achieved because the steps in the experiment after training the feature extractor are deterministic and, therefore, the classification accuracies for two experiment iterations must necessarily be the same if the outcomes of their training phases are identical, the same feature extractor is used, and the same data preprocessing technique is employed. Consequently, all iterations of the experiment procedure for a specific parameter having the same values for the feature extractor method and data preprocessing factors can be clustered according to the outcome of the training phase. Once clustered, only one iteration from each cluster is pursued to completion, and the resulting classification accuracy is transferred to the other iterations of the experiment procedure within the cluster. Using this approach, all iterations of the experiment procedure for a parameter must be performed at least through the feature extractor training, but only one iteration of the experiment procedure per cluster needs to be completed.

The experiment was performed using this clustering scheme. Table 4.6 details the number of iterations of the experiment procedure that were completed under the clustering scheme broken down by parameter and feature extractor. The total number of iterations per parameter that would have been completed without clustering is 560 for each statistical feature extractor and 280 for each structure detector. Notice that the number of iterations completed is very low for the structure detectors and consistently high for the wavelet transformation. This disparity is due to the fact that the outcomes of their associated training phases differ in the degree to which they enable

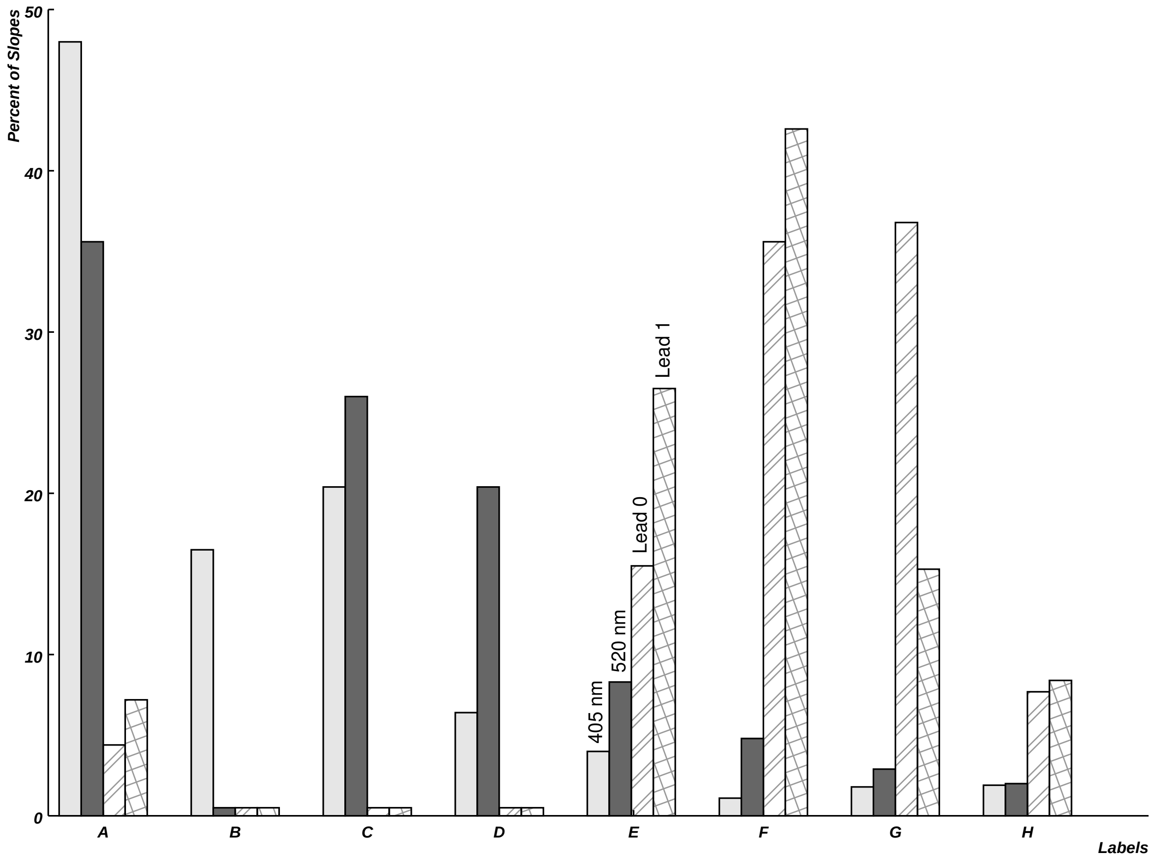


Figure 4.13 The distribution of the percent of normalized slope values across the labels for each of the four parameters contained in the wafer and ECG databases. The distributions for the wafer database parameters are represented by the shaded bars: the lightly-shaded bar shows the distribution for the 405 nm parameter, and the darkly-shaded bar shows the distribution for the 520 nm parameter. The distributions for the ECG database parameters are represented by the patterned bars: the diagonally-filled bar shows the distributions for the lead 0 parameter, and the crosshatch-filled bar shows the distributions for the lead 1 parameter.

	405 nm	520 nm	Lead 0	Lead 1
Identity	48	48	65	65
Fourier	28	29	208	83
Wavelet	464	405	303	288
Constant	2	2	4	5
Straight	2	2	4	3
Exponential	1	2	3	3
Sinusoidal	2	2	3	2
Triangular	2	2	4	2
Trapezoidal	1	2	4	2
Composite	1	1	4	2

Table 4.6 The number of iterations of the experiment procedure that were completed under the clustering scheme broken down by parameter and feature extractor. The total number of iterations per parameter that would have been completed without clustering is 560 for each statistical feature extractor and 280 for each structure detector.

clustering. The outcome of the training phase for the structure detectors consists solely of the number of subregions. The value for the number of subregions varied little among the iterations, thereby allowing for a high degree of clustering. The outcome of the training phase for the wavelet transformation consists of a uniform data length and a subset of transformations B. The subset of transformations B varied greatly among the iterations, thereby preventing a high degree of clustering (since iterations can not be clustered if the contents of B differs). Overall, only 2,103 of the possible 14,560 iterations of the experiment procedure were completed, thereby reducing the computational effort necessary to perform the experiment by reducing the number of iterations completed to about 14% of the total possible number of iterations.

4.4. Experiment Results

The experiment outlined in Section 4.3 was performed separately for each of the two parameters in both the wafer and ECG databases. The experiment results for the wafer and ECG databases are presented separately in Sections 4.4.1 and 4.4.2, respectively. A discussion of the combined results appears in Section 4.5.

4.4.1. Wafer Database

Tables 4.7 and 4.9 report the classification accuracies for the normal and abnormal data sets, respectively, of the 405 nm parameter using the statistical feature extractors and broken down by combinations of the experimental factors. Tables 4.8 and 4.10 report the parallel results for the 520 nm parameter. The classification accuracy for each combination of experimental factors is reported as the mean and standard deviation of the percents of the data sets correctly classified across twenty iterations of the experiment procedure. Several trends are apparent:

		Padding			Truncation		
r	w	Identity	Fourier	Wavelet	Identity	Fourier	Wavelet
2	0	99.7 (0.0)	99.6 (0.1)	99.9 (0.1)	99.7 (0.0)	99.8 (0.7)	99.3 (0.9)
2	1	99.7 (0.0)	99.6 (0.1)	99.9 (0.0)	99.7 (0.0)	100 (0.0)	99.5 (0.2)
4	0	99.7 (0.0)	99.6 (0.1)	99.9 (0.0)	99.7 (0.0)	99.6 (1.1)	99.1 (1.1)
4	1	99.7 (0.0)	99.6 (0.1)	99.9 (0.0)	99.7 (0.1)	99.7 (0.9)	99.2 (1.0)
4	2	99.7 (0.0)	99.6 (0.1)	99.9 (0.1)	99.7 (0.1)	99.5 (1.1)	98.9 (1.3)
8	0	99.7 (0.0)	99.5 (0.0)	99.9 (0.0)	99.7 (0.1)	99.0 (1.4)	98.7 (1.4)
8	1	99.7 (0.0)	99.5 (0.0)	99.9 (0.0)	99.7 (0.1)	99.0 (1.4)	98.8 (1.3)
8	2	99.7 (0.0)	99.5 (0.0)	99.9 (0.0)	99.7 (0.1)	99.1 (1.4)	99.2 (0.8)
8	4	99.7 (0.0)	99.6 (0.2)	99.9 (0.0)	99.7 (0.1)	99.6 (0.9)	99.2 (0.8)
16	0	99.7 (0.0)	99.6 (0.0)	99.9 (0.0)	99.6 (0.1)	98.3 (1.5)	97.5 (1.5)
16	1	99.7 (0.0)	99.6 (0.0)	99.9 (0.0)	99.6 (0.1)	98.0 (1.4)	97.4 (1.5)
16	2	99.7 (0.0)	99.5 (0.0)	99.9 (0.0)	99.6 (0.1)	98.1 (1.4)	97.2 (1.5)
16	4	99.7 (0.0)	99.5 (0.0)	99.9 (0.0)	99.6 (0.1)	98.1 (1.4)	97.5 (1.4)
16	8	99.7 (0.0)	99.6 (0.1)	99.9 (0.0)	99.6 (0.1)	99.0 (1.4)	98.7 (1.3)

Table 4.7 Normal, 405 nm, statistical extractors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the normal data sets correctly classified are reported.

		Padding			Truncation		
r	w	Identity	Fourier	Wavelet	Identity	Fourier	Wavelet
2	0	99.1 (0.2)	99.1 (0.2)	99.6 (0.3)	98.9 (0.3)	98.5 (0.3)	98.6 (0.4)
2	1	99.1 (0.1)	99.2 (0.3)	99.4 (0.5)	99.0 (0.2)	98.2 (0.2)	98.6 (0.1)
4	0	99.1 (0.1)	99.1 (0.1)	99.7 (0.3)	98.8 (0.3)	98.6 (0.4)	98.6 (0.4)
4	1	99.0 (0.1)	99.1 (0.1)	99.7 (0.2)	98.7 (0.4)	98.5 (0.5)	98.7 (0.5)
4	2	99.0 (0.1)	99.1 (0.1)	99.6 (0.3)	98.8 (0.5)	98.6 (0.6)	98.5 (0.2)
8	0	99.1 (0.1)	99.1 (0.1)	99.7 (0.2)	98.6 (0.2)	98.9 (0.6)	98.5 (0.4)
8	1	99.0 (0.1)	99.1 (0.1)	99.7 (0.3)	98.6 (0.2)	98.9 (0.7)	98.5 (0.3)
8	2	99.0 (0.1)	99.1 (0.1)	99.7 (0.3)	98.6 (0.4)	98.8 (0.6)	98.7 (0.7)
8	4	99.0 (0.0)	99.1 (0.1)	99.7 (0.1)	98.6 (0.4)	98.6 (0.6)	99.4 (0.8)
16	0	99.0 (0.1)	99.1 (0.1)	99.8 (0.0)	98.6 (0.2)	99.3 (0.6)	98.3 (0.2)
16	1	99.0 (0.1)	99.1 (0.1)	99.8 (0.2)	98.6 (0.2)	99.4 (0.6)	98.3 (0.2)
16	2	99.0 (0.1)	99.1 (0.1)	99.8 (0.0)	98.6 (0.3)	99.4 (0.6)	98.3 (0.3)
16	4	99.0 (0.0)	99.1 (0.2)	99.8 (0.0)	98.5 (0.3)	99.4 (0.6)	98.3 (0.3)
16	8	99.0 (0.0)	99.1 (0.1)	99.8 (0.1)	98.4 (0.4)	98.9 (0.8)	98.4 (0.2)

Table 4.8 Normal, 520 nm, statistical extractors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the normal data sets correctly classified are reported.

		Padding			Truncation		
r	w	Identity	Fourier	Wavelet	Identity	Fourier	Wavelet
2	0	90.6 (0.0)	77.5 (5.6)	84.1 (1.7)	90.6 (0.3)	72.7 (16.1)	87.0 (3.0)
2	1	90.6 (0.0)	76.7 (6.2)	83.7 (0.7)	90.6 (0.0)	78.0 (0.0)	88.7 (2.5)
4	0	90.6 (0.0)	78.6 (5.5)	83.2 (0.8)	90.6 (0.3)	70.9 (17.3)	86.1 (4.3)
4	1	90.6 (0.0)	77.3 (6.1)	83.5 (1.2)	90.2 (1.6)	69.3 (21.8)	86.8 (5.2)
4	2	90.6 (0.0)	78.1 (5.9)	83.7 (1.2)	89.8 (2.2)	68.3 (19.9)	87.9 (4.0)
8	0	90.6 (0.0)	80.0 (4.6)	82.8 (0.5)	90.2 (1.4)	61.2 (23.5)	76.4 (17.3)
8	1	90.6 (0.0)	80.0 (4.6)	83.0 (0.6)	90.2 (1.4)	61.2 (23.5)	76.5 (18.1)
8	2	90.6 (0.0)	81.9 (0.0)	83.0 (0.6)	89.8 (2.2)	63.8 (22.2)	72.9 (23.1)
8	4	90.6 (0.0)	79.4 (5.2)	83.4 (0.8)	89.9 (2.1)	66.7 (23.8)	83.3 (14.7)
16	0	90.6 (0.0)	79.2 (5.1)	82.8 (0.4)	90.0 (1.7)	39.7 (26.1)	74.0 (12.2)
16	1	90.6 (0.0)	79.4 (5.2)	82.8 (0.4)	90.0 (1.7)	41.0 (22.9)	74.8 (12.2)
16	2	90.6 (0.0)	81.3 (2.8)	82.8 (0.5)	89.7 (2.1)	39.4 (24.6)	77.7 (5.7)
16	4	90.6 (0.0)	81.3 (2.8)	82.9 (0.6)	89.6 (2.3)	39.4 (24.6)	73.5 (12.8)
16	8	90.6 (0.0)	81.3 (2.8)	83.3 (0.8)	88.7 (3.0)	57.5 (26.5)	77.2 (18.0)

Table 4.9 Abnormal, 405 nm, statistical extractors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the abnormal data sets correctly classified are reported.

		Padding			Truncation		
r	w	Identity	Fourier	Wavelet	Identity	Fourier	Wavelet
2	0	77.1 (2.2)	76.9 (5.0)	73.9 (1.8)	76.7 (4.2)	59.5 (12.9)	68.7 (18.9)
2	1	78.7 (1.6)	76.2 (5.1)	75.2 (3.6)	76.9 (2.3)	68.2 (5.7)	72.6 (8.9)
4	0	77.7 (2.0)	77.0 (4.6)	73.9 (2.5)	76.5 (4.7)	56.0 (15.4)	69.5 (19.8)
4	1	79.5 (1.4)	78.3 (3.6)	73.8 (2.7)	75.9 (4.8)	58.2 (16.9)	65.6 (24.9)
4	2	79.7 (1.3)	78.9 (3.4)	74.0 (3.2)	75.6 (4.4)	56.1 (21.7)	70.3 (11.6)
8	0	78.6 (1.6)	77.6 (4.2)	73.7 (2.6)	74.0 (6.3)	46.1 (22.8)	66.8 (20.3)
8	1	79.5 (1.4)	78.2 (3.8)	73.8 (3.1)	74.3 (6.3)	45.9 (23.0)	70.0 (13.1)
8	2	80.0 (1.0)	78.3 (3.8)	73.4 (2.6)	75.5 (4.7)	48.5 (21.7)	55.2 (30.5)
8	4	80.2 (0.7)	79.5 (2.3)	73.6 (1.0)	76.5 (4.1)	55.9 (19.4)	23.2 (33.2)
16	0	79.2 (1.5)	78.1 (3.8)	72.8 (1.0)	69.3 (6.5)	28.5 (20.8)	61.3 (12.5)
16	1	79.8 (1.2)	78.7 (3.3)	73.2 (2.8)	69.6 (6.4)	27.6 (21.1)	61.3 (12.5)
16	2	80.0 (1.0)	78.8 (3.3)	72.6 (1.0)	69.5 (6.6)	26.4 (21.6)	61.5 (12.7)
16	4	80.3 (0.0)	78.5 (3.9)	72.8 (1.0)	69.6 (6.5)	27.1 (21.4)	59.8 (10.7)
16	8	80.3 (0.0)	79.1 (3.3)	73.3 (0.9)	73.7 (6.1)	45.8 (26.9)	65.0 (10.7)

Table 4.10 Abnormal, 520 nm, statistical extractors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the abnormal data sets correctly classified are reported.

- The classification accuracies for the normal data sets of both parameters are uniformly around 99% regardless of the values of the experimental factors.

- When the experimental factors are held constant except for data preprocessing, the pairwise differences are small between the classification accuracies when using padding versus truncation for the normal data sets of each parameter. In most cases, the means are slightly higher and the standard deviations are slightly lower when using padding.

- The classification accuracies for the abnormal data sets of each parameter vary with the values of the feature extraction method and the data preprocessing experimental factors. The training set size and the training set composition experimental factors have less of an effect on the classification accuracies, with the notable exception of when the training set size is 16 and the Fourier transformation is used as the feature extraction method.

- When the experimental factors are held constant except for data preprocessing, the pairwise relationships between the classification accuracies when using padding versus truncation do not consistently recommend one technique over the other for the abnormal data sets of each parameter, but the pairwise differences can be large in those instances where padding outperforms truncation. Moreover, the standard deviations when using truncation are always larger than the corresponding standard deviations when using padding.

The most interesting observation is that the identity transformation, which implements the simplest feature extraction method, performs as well as or better than the other statistical techniques across all combinations of the remaining experimental factors for the normal and abnormal data sets of each parameter. This outcome suggests that there is little value in using a more complex statistical technique to extract features from these data.

Table 4.11 reports the classification accuracies for the normal and abnormal data sets of the 405 nm and 520 nm parameters averaged across the various values of the training set size and the training set composition experimental factors for the statistical feature extractors. For each combination of the feature extraction method and the data preprocessing experimental factors, the mean and standard deviation of the percents of the data sets correctly classified are reported. When the experimental factors are held constant except for data preprocessing, the pairwise differences between the classification accuracies when using padding versus truncation are all statistically significant (based on a one-tailed t-test with $p < .001$). Since padding consistently results in a superior classification accuracy, its use as a data preprocessing technique for these data is preferable to truncation.

Tables 4.12 and 4.14 report the classification accuracies for the normal and abnormal data sets, respectively, of the 405 nm parameter using the structure detectors and broken down by combinations of the experimental factors. Tables 4.13 and 4.15 report the parallel results for the 520 nm parameter. The classification accuracy for each combination of experimental factors is reported as the mean and standard deviation of the percents of the data sets correctly classified across twenty iterations of the experiment procedure. Several trends are apparent:

- The classification accuracies are generally unaffected by the training set size and the training set composition experimental factors for the normal and abnormal data sets of each parameter. The standard deviations are most often zero.

405 nm	Normal		Abnormal	
	Padding	Truncation	Padding	Truncation
Identity	99.7 (0.0)	99.7 (0.1)	90.6 (0.0)	90.0 (1.8)
Fourier	99.6 (0.1)	99.1 (1.3)	79.4 (4.9)	59.1 (25.2)
Wavelet	99.9 (0.0)	98.6 (1.4)	83.2 (0.9)	80.2 (13.7)

520 nm	Normal		Abnormal	
	Padding	Truncation	Padding	Truncation
Identity	99.0 (0.1)	98.7 (0.3)	79.3 (1.6)	73.8 (6.0)
Fourier	99.1 (0.1)	98.8 (0.7)	78.2 (3.9)	46.4 (23.7)
Wavelet	99.7 (0.3)	98.5 (0.5)	73.6 (2.4)	62.2 (21.7)

Table 4.11 Classification accuracies for the normal and abnormal data sets of the 405 nm and 520 nm parameters averaged across the various values of the training set size and the training set composition experimental factors for the statistical feature extractors. The means (and standard deviations) of the percents of the data sets correctly classified are reported. When the experimental factors are held constant except for data preprocessing, the pairwise differences between the classification accuracies when using padding versus truncation are all statistically significant (based on a one-tailed t-test with $p < .001$).

- For each combination of experimental factors, the classification accuracies for the normal data sets of the 405 nm parameter are approximately equivalent to the corresponding classification accuracies for the normal data sets of the 520 nm parameter.

- For each combination of experimental factors, the classification accuracies for the abnormal data sets of the 405 nm parameter are superior to the corresponding classification accuracies for the abnormal data sets of the 520 nm parameter, with the exception of the triangular structure detector.

- For each combination of experimental factors, the classification accuracies for the normal data sets of each parameter are superior to the corresponding classification accuracies for the abnormal data sets.

Notice that the classification accuracies when using the composite structure detector, which combines the heterogeneous structure detectors to extract features from a data set, are rarely better than when using each heterogeneous structure detector individually (except for the normal data sets of the 405 nm parameter). The underlying cause for this counterintuitive result is discussed in Section 4.5.

The large number of standard deviations equal to zero indicates that the percent of the data sets correctly classified is often the same across the twenty iterations of the experiment procedure for each combination of experimental factors. Similar behavior can be observed in the classification accuracies reported when the identity transformation is used as the feature extraction method in Tables 4.7 and 4.9. The structure detectors and the identity transformation are prone to such

r	w	Constant	Straight	Exponential	Sinusoidal	Triangular	Trapezoidal	Composite
2	0	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.0)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
2	1	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.7 (0.1)	99.2 (0.2)	99.3 (0.0)	100 (0.0)
4	0	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.1)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
4	1	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.7 (0.1)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
4	2	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.1)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
8	0	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.0)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
8	1	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.0)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
8	2	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.0)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
8	4	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.0)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
16	0	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.0)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
16	1	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.0)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
16	2	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.0)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
16	4	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.0)	99.1 (0.0)	99.3 (0.0)	100 (0.0)
16	8	99.4 (0.0)	99.7 (0.0)	99.9 (0.0)	99.6 (0.0)	99.1 (0.0)	99.3 (0.0)	100 (0.0)

Table 4.12 Normal, 405 nm, structure detectors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the normal data sets correctly classified are reported.

r	w	Constant	Straight	Exponential	Sinusoidal	Triangular	Trapezoidal	Composite
2	0	99.4 (0.0)	99.3 (0.2)	98.9 (0.1)	99.1 (0.0)	99.7 (0.2)	98.1 (0.0)	99.3 (0.0)
2	1	99.4 (0.0)	99.3 (0.2)	98.9 (0.2)	99.0 (0.1)	99.5 (0.4)	98.2 (0.3)	99.3 (0.0)
4	0	99.4 (0.0)	99.3 (0.0)	98.9 (0.0)	99.1 (0.0)	99.7 (0.0)	98.1 (0.0)	99.3 (0.0)
4	1	99.4 (0.0)	99.4 (0.2)	98.9 (0.0)	99.0 (0.1)	99.7 (0.0)	98.1 (0.0)	99.3 (0.0)
4	2	99.4 (0.0)	99.4 (0.3)	98.9 (0.0)	99.0 (0.1)	99.7 (0.2)	98.1 (0.0)	99.3 (0.0)
8	0	99.4 (0.0)	99.3 (0.0)	98.9 (0.0)	99.1 (0.0)	99.7 (0.0)	98.1 (0.0)	99.3 (0.0)
8	1	99.4 (0.0)	99.3 (0.1)	98.9 (0.0)	99.1 (0.0)	99.7 (0.0)	98.1 (0.0)	99.3 (0.0)
8	2	99.4 (0.0)	99.3 (0.2)	98.9 (0.0)	99.1 (0.0)	99.7 (0.0)	98.1 (0.0)	99.3 (0.0)
8	4	99.4 (0.0)	99.4 (0.3)	98.9 (0.0)	99.1 (0.0)	99.7 (0.0)	98.1 (0.0)	99.3 (0.0)
16	0	99.4 (0.0)	99.3 (0.0)	98.9 (0.0)	99.1 (0.0)	99.7 (0.0)	98.1 (0.0)	99.3 (0.0)
16	1	99.4 (0.0)	99.3 (0.0)	98.9 (0.0)	99.1 (0.0)	99.7 (0.0)	98.1 (0.0)	99.3 (0.0)
16	2	99.4 (0.0)	99.3 (0.0)	98.9 (0.0)	99.1 (0.0)	99.7 (0.0)	98.1 (0.0)	99.3 (0.0)
16	4	99.4 (0.0)	99.3 (0.2)	98.9 (0.0)	99.1 (0.0)	99.7 (0.0)	98.1 (0.0)	99.3 (0.0)
16	8	99.4 (0.0)	99.4 (0.2)	98.9 (0.0)	99.1 (0.0)	99.7 (0.0)	98.1 (0.0)	99.3 (0.0)

Table 4.13 Normal, 520 nm, structure detectors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the normal data sets correctly classified are reported.

r	w	Constant	Straight	Exponential	Sinusoidal	Triangular	Trapezoidal	Composite
2	0	91.3 (0.0)	85.8 (0.0)	92.1 (0.0)	87.4 (0.0)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
2	1	90.6 (1.6)	86.7 (2.0)	92.1 (0.0)	87.9 (1.2)	83.7 (5.9)	92.1 (0.0)	88.2 (0.0)
4	0	91.1 (0.9)	85.8 (0.0)	92.1 (0.0)	87.6 (0.7)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
4	1	90.9 (1.2)	85.8 (0.0)	92.1 (0.0)	87.7 (1.0)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
4	2	91.1 (0.9)	85.8 (0.0)	92.1 (0.0)	87.6 (0.7)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
8	0	91.3 (0.0)	85.8 (0.0)	92.1 (0.0)	87.4 (0.0)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
8	1	91.3 (0.0)	85.8 (0.0)	92.1 (0.0)	87.4 (0.0)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
8	2	91.3 (0.0)	85.8 (0.0)	92.1 (0.0)	87.4 (0.0)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
8	4	91.3 (0.0)	85.8 (0.0)	92.1 (0.0)	87.4 (0.0)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
16	0	91.3 (0.0)	85.8 (0.0)	92.1 (0.0)	87.4 (0.0)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
16	1	91.3 (0.0)	85.8 (0.0)	92.1 (0.0)	87.4 (0.0)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
16	2	91.3 (0.0)	85.8 (0.0)	92.1 (0.0)	87.4 (0.0)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
16	4	91.3 (0.0)	85.8 (0.0)	92.1 (0.0)	87.4 (0.0)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)
16	8	91.3 (0.0)	85.8 (0.0)	92.1 (0.0)	87.4 (0.0)	80.3 (0.0)	92.1 (0.0)	88.2 (0.0)

Table 4.14 Abnormal, 405 nm, structure detectors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the abnormal data sets correctly classified are reported.

r	w	Constant	Straight	Exponential	Sinusoidal	Triangular	Trapezoidal	Composite
2	0	81.9 (0.0)	85.7 (0.2)	83.5 (0.4)	81.9 (0.0)	82.9 (1.1)	74.8 (0.0)	78.7 (0.0)
2	1	82.0 (0.2)	85.7 (0.3)	83.6 (0.5)	83.4 (3.8)	83.6 (1.9)	75.4 (1.7)	78.7 (0.0)
4	0	81.9 (0.0)	85.8 (0.0)	83.5 (0.0)	81.9 (0.0)	82.7 (0.0)	74.8 (0.0)	78.7 (0.0)
4	1	81.9 (0.0)	85.6 (0.3)	83.5 (0.0)	82.9 (3.2)	82.7 (0.0)	74.8 (0.0)	78.7 (0.0)
4	2	81.9 (0.0)	85.6 (0.4)	83.5 (0.0)	82.9 (3.2)	82.9 (1.1)	74.8 (0.0)	78.7 (0.0)
8	0	81.9 (0.0)	85.8 (0.0)	83.5 (0.0)	81.9 (0.0)	82.7 (0.0)	74.8 (0.0)	78.7 (0.0)
8	1	81.9 (0.0)	85.8 (0.2)	83.5 (0.0)	81.9 (0.0)	82.7 (0.0)	74.8 (0.0)	78.7 (0.0)
8	2	81.9 (0.0)	85.7 (0.3)	83.5 (0.0)	81.9 (0.0)	82.7 (0.0)	74.8 (0.0)	78.7 (0.0)
8	4	81.9 (0.0)	85.6 (0.4)	83.5 (0.0)	81.9 (0.0)	82.7 (0.0)	74.8 (0.0)	78.7 (0.0)
16	0	81.9 (0.0)	85.8 (0.0)	83.5 (0.0)	81.9 (0.0)	82.7 (0.0)	74.8 (0.0)	78.7 (0.0)
16	1	81.9 (0.0)	85.8 (0.0)	83.5 (0.0)	81.9 (0.0)	82.7 (0.0)	74.8 (0.0)	78.7 (0.0)
16	2	81.9 (0.0)	85.8 (0.0)	83.5 (0.0)	81.9 (0.0)	82.7 (0.0)	74.8 (0.0)	78.7 (0.0)
16	4	81.9 (0.0)	85.7 (0.2)	83.5 (0.0)	81.9 (0.0)	82.7 (0.0)	74.8 (0.0)	78.7 (0.0)
16	8	81.9 (0.0)	85.6 (0.3)	83.5 (0.0)	81.9 (0.0)	82.7 (0.0)	74.8 (0.0)	78.7 (0.0)

Table 4.15 Abnormal, 520 nm, structure detectors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the abnormal data sets correctly classified are reported.

	Normal		Abnormal	
	405 nm	520 nm	405 nm	520 nm
Constant	99.4 (0.0)	99.4 (0.0)	91.2 (0.7)	81.9 (0.1)
Straight	99.7 (0.0)	99.3 (0.2)	85.9 (0.6)	85.7 (0.3)
Exponential	99.9 (0.0)	98.9 (0.1)	92.1 (0.0)	83.5 (0.2)
Sinusoidal	99.6 (0.0)	99.1 (0.0)	87.5 (0.5)	82.1 (1.6)
Triangular	99.1 (0.1)	99.7 (0.1)	80.6 (1.8)	82.8 (0.7)
Trapezoidal	99.3 (0.0)	98.1 (0.1)	92.1 (0.0)	74.8 (0.5)
Composite	100 (0.0)	99.3 (0.0)	88.2 (0.0)	78.7 (0.0)

Table 4.16 Classification accuracies for the normal and abnormal data sets of the 405 nm and 520 nm parameters averaged across the various levels of the training set size and the training set composition experimental factors for the structure detectors. The means (and standard deviations) of the percents of the data sets correctly classified are reported.

Normal		Abnormal	
405 nm	520 nm	405 nm	520 nm
Composite	Triangular	Exponential	Straight
Exponential	Constant	Trapezoidal	Exponential
Straight	Composite	Constant	Triangular
Sinusoidal	Straight	Composite	Sinusoidal
Constant	Sinusoidal	Sinusoidal	Constant
Trapezoidal	Exponential	Straight	Composite
Triangular	Trapezoidal	Triangular	Trapezoidal

Table 4.17 The structure detectors ordered by classification accuracy for the normal and abnormal data sets of the 405 nm and 520 nm parameters. The structure detectors are listed from most (top) to least (bottom) accurate.

results because each is associated with a training phase with an outcome comprising a single numeric value, thus increasing the likelihood that separate training phases will produce the same outcome. By comparison, the outcomes of the training phases associated with the Fourier and wavelet transformations include a subset of basis waveforms that can vary widely among data sets, thereby reducing the likelihood that separate training phases will produce the same outcome.

Table 4.16 reports the classification accuracies for the normal and abnormal data sets of the 405 nm and 520 nm parameters averaged across the various levels of the training set size and the training set composition experimental factors for the structure detectors. Clearly, the structure detectors classify the normal data sets more accurately than the abnormal data sets, regardless of parameter. The classification accuracies of the abnormal data sets are slightly better for the 405 nm parameter than for the 520 nm parameter. Table 4.17 lists the structure detectors sorted by the classification accuracies for the normal and abnormal data sets of each parameter. The exponential structure detector performs well for both the normal and abnormal data sets of the 405 nanometer parameter;

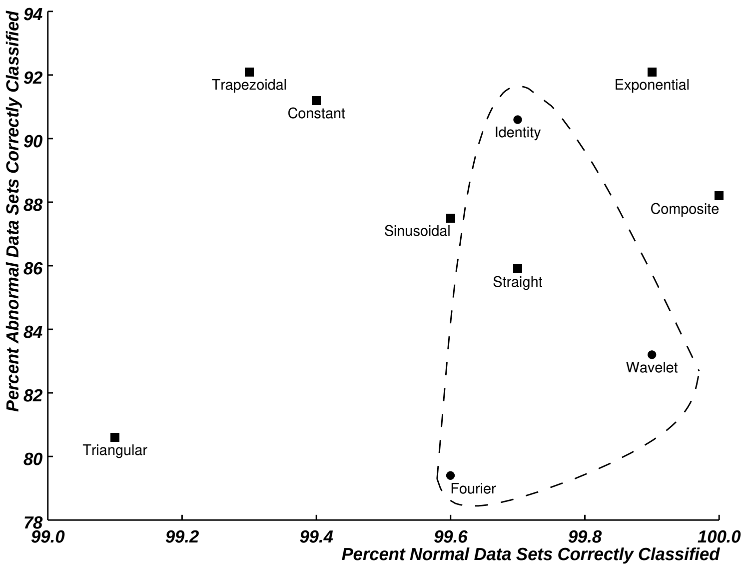


Figure 4.14 The relative classification accuracies for all feature extraction methods for the 405 nm parameter. Each feature extraction method is plotted using the mean percent of normal and abnormal data sets correctly classified as coordinates. For the statistical feature extraction methods, the results with padding are used and are enclosed with a dashed line.

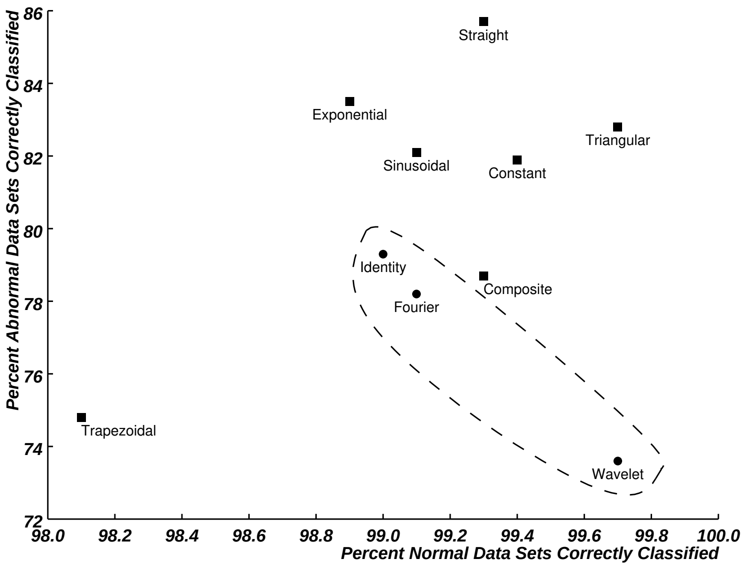


Figure 4.15 The relative classification accuracies for all feature extraction methods for the 520 nm parameter. Each feature extraction method is plotted using the mean percent of normal and abnormal data sets correctly classified as coordinates. For the statistical feature extraction methods, the results with padding are used and are enclosed with a dashed line.

the triangular structure detector performs best for both the normal and abnormal data sets of the 520 nanometer parameter. No one structure detector, however, stands out as being superior to the others.

To compare the efficacy of the structure detectors with the statistical feature extraction methods, the overall ability to correctly classify both the normal and abnormal data sets of each parameter must be examined. Figures 4.14 and 4.15 illustrate the relative classification accuracies of the ten feature extraction methods (note that the scales of the x and y axes on each graph are not uniform and differ between the two graphs). Each feature extraction method is plotted using the mean percent of normal and abnormal data sets correctly classified as coordinates: the results in Table 4.11 are used to plot the statistical feature extraction methods (the results with padding are used), and the results in Table 4.16 are used to plot the structure detectors. A dashed line encloses the three statistical feature extraction methods; the structure detectors with better classification accuracies are those which are plotted further towards the upper-right of the graph relative to the dashed area. For the 405 nm parameter, the exponential structure detector is the best overall feature extraction method; the triangular structure detector performs poorly. For the 520 nm parameter, the straight structure detector is the best overall feature extraction method (the triangular structure detector could be considered to be superior if correctly classifying normal data sets far outweighs correctly classifying abnormal data sets); the triangular, sinusoidal, and constant structure detectors perform acceptably well over the statistical feature extraction methods, while the trapezoidal structure detector performs particularly poorly.

4.4.2. ECG Database

Tables 4.18 and 4.20 report the classification accuracies for the normal and abnormal data sets, respectively, of the lead 0 parameter using the statistical feature extractors and broken down by combinations of the experimental factors. Tables 4.19 and 4.21 report the parallel results for the lead 1 parameter. The classification accuracy for each combination of experimental factors is reported as the mean and standard deviation of the percents of the data sets correctly classified across twenty iterations of the experiment procedure. While there are no trends which hold for all three statistical feature extractors, observations about the classification accuracies can be made on an individual basis:

- When the experimental factors are held constant except for data preprocessing, the classification accuracies are higher when using truncation as compared to padding with the identity and Fourier feature extraction methods for the normal data sets of each parameter. The classification accuracies are higher when using padding as compared to truncation with the wavelet feature extraction method for the normal data sets of the lead 0 parameter; the classification accuracies depend on the levels of the training set size and the training set composition factors for the normal data sets of the lead 1 parameter.

- The classification accuracies with the identity and wavelet feature extraction methods are generally better for normal data sets of the lead 1 parameter versus the lead 0 parameter across all combinations of the remaining experimental factors. The classification accuracies with the Fourier feature extraction method are better for the normal data sets of the lead 0 parameter versus the lead 1 parameter for all combinations of the remaining experimental factors.

		Padding			Truncation		
r	w	Identity	Fourier	Wavelet	Identity	Fourier	Wavelet
2	0	78.2 (0.0)	84.0 (3.9)	86.1 (2.4)	78.2 (0.0)	91.8 (2.2)	85.5 (5.8)
2	1	78.2 (0.0)	84.3 (3.5)	88.4 (2.1)	78.3 (2.2)	90.3 (2.1)	82.6 (6.8)
4	0	78.2 (0.0)	84.4 (2.8)	87.0 (3.3)	78.2 (0.0)	91.7 (1.5)	82.0 (6.8)
4	1	78.2 (0.0)	83.7 (2.3)	86.7 (4.2)	79.4 (3.5)	90.8 (1.8)	80.2 (5.9)
4	2	78.2 (0.0)	83.5 (3.0)	86.9 (4.2)	80.3 (4.2)	90.6 (1.9)	83.3 (6.2)
8	0	78.2 (0.0)	84.7 (2.3)	86.8 (1.9)	77.8 (0.7)	91.8 (1.9)	83.3 (5.7)
8	1	78.2 (0.0)	84.7 (1.4)	87.7 (1.8)	80.1 (4.3)	90.8 (2.0)	81.5 (5.9)
8	2	78.2 (0.0)	84.1 (2.8)	87.3 (1.9)	81.2 (5.1)	90.1 (1.9)	81.6 (4.8)
8	4	78.2 (0.0)	84.4 (2.4)	87.4 (2.0)	82.0 (5.3)	89.1 (1.1)	83.8 (5.2)
16	0	78.2 (0.0)	85.2 (0.4)	86.4 (1.6)	77.4 (0.8)	90.8 (2.1)	83.0 (5.6)
16	1	78.2 (0.0)	84.9 (2.5)	86.5 (1.7)	80.7 (5.0)	89.6 (1.9)	85.1 (2.7)
16	2	78.2 (0.0)	85.2 (1.1)	87.6 (2.3)	82.1 (5.3)	89.4 (1.7)	84.9 (2.8)
16	4	78.2 (0.0)	84.6 (0.8)	87.3 (1.9)	82.6 (5.2)	89.2 (1.4)	84.3 (4.7)
16	8	78.2 (0.0)	84.1 (2.0)	87.0 (3.1)	85.1 (4.3)	88.7 (0.0)	85.3 (3.7)

Table 4.18 Normal, lead 0, statistical extractors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the normal data sets correctly classified are reported.

		Padding			Truncation		
r	w	Identity	Fourier	Wavelet	Identity	Fourier	Wavelet
2	0	84.4 (0.8)	81.5 (1.3)	90.3 (2.4)	85.0 (1.5)	86.7 (0.7)	93.2 (0.0)
2	1	84.3 (0.5)	81.7 (1.6)	88.6 (2.8)	86.4 (2.3)	85.8 (1.9)	88.0 (7.8)
4	0	84.2 (0.0)	81.4 (1.2)	89.1 (3.4)	86.1 (1.9)	86.5 (0.8)	93.2 (0.0)
4	1	84.2 (0.0)	82.2 (2.1)	88.6 (2.8)	87.5 (1.7)	86.1 (2.0)	88.7 (8.0)
4	2	84.4 (0.8)	82.8 (2.5)	87.9 (3.3)	86.9 (4.4)	85.3 (2.5)	86.8 (8.8)
8	0	84.2 (0.0)	80.8 (0.6)	88.2 (3.1)	87.6 (1.8)	86.2 (2.0)	90.5 (6.6)
8	1	84.2 (0.0)	81.3 (1.5)	87.2 (2.5)	87.2 (2.5)	85.0 (2.4)	86.9 (8.8)
8	2	84.2 (0.0)	81.7 (2.0)	87.0 (4.6)	86.8 (3.3)	83.9 (2.7)	82.4 (9.1)
8	4	84.2 (0.0)	82.7 (2.6)	87.5 (3.5)	85.1 (6.2)	82.9 (2.0)	77.8 (6.3)
16	0	84.2 (0.0)	80.9 (0.6)	90.1 (1.9)	88.9 (0.8)	84.9 (2.5)	86.0 (9.1)
16	1	84.2 (0.0)	81.3 (1.3)	90.3 (1.7)	85.8 (5.2)	82.9 (1.9)	78.8 (7.4)
16	2	84.2 (0.0)	81.5 (1.9)	90.3 (1.7)	84.7 (6.1)	82.7 (1.9)	77.9 (6.6)
16	4	84.2 (0.0)	82.2 (2.7)	88.8 (2.4)	84.1 (6.9)	82.7 (2.0)	77.9 (6.6)
16	8	84.2 (0.0)	83.2 (3.0)	87.8 (2.2)	83.1 (7.3)	81.9 (0.2)	75.2 (0.0)

Table 4.19 Normal, lead 1, statistical extractors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the normal data sets correctly classified are reported.

		Padding			Truncation		
r	w	Identity	Fourier	Wavelet	Identity	Fourier	Wavelet
2	0	76.1 (0.0)	74.9 (4.4)	72.2 (2.8)	76.1 (0.0)	76.3 (1.0)	72.8 (6.2)
2	1	76.1 (0.0)	73.3 (5.2)	73.2 (3.2)	76.0 (2.5)	77.7 (3.6)	75.7 (6.9)
4	0	76.1 (0.0)	74.0 (3.9)	71.3 (3.7)	76.1 (0.0)	76.1 (0.0)	76.5 (7.2)
4	1	76.1 (0.0)	75.6 (5.7)	73.7 (3.4)	74.8 (4.0)	78.0 (3.3)	78.4 (6.0)
4	2	76.1 (0.0)	76.6 (6.4)	73.5 (3.6)	73.7 (4.8)	78.7 (3.7)	74.9 (6.1)
8	0	76.1 (0.0)	72.9 (2.8)	72.1 (2.8)	76.5 (0.7)	77.2 (2.7)	74.3 (6.5)
8	1	76.1 (0.0)	74.3 (4.5)	72.3 (2.9)	73.9 (4.9)	78.7 (3.7)	76.3 (6.7)
8	2	76.1 (0.0)	75.2 (4.1)	72.1 (3.4)	72.5 (5.8)	80.6 (3.8)	75.8 (5.0)
8	4	76.1 (0.0)	74.6 (6.5)	71.6 (4.0)	71.6 (6.1)	82.5 (2.7)	74.6 (4.2)
16	0	76.1 (0.0)	73.1 (0.0)	72.0 (2.8)	76.9 (0.8)	79.1 (3.8)	74.9 (6.4)
16	1	76.1 (0.0)	74.1 (3.6)	71.6 (2.6)	73.1 (5.6)	82.1 (3.1)	72.7 (2.4)
16	2	76.1 (0.0)	73.6 (4.1)	72.1 (3.5)	71.5 (6.0)	82.5 (2.7)	73.1 (2.0)
16	4	76.1 (0.0)	74.5 (6.4)	71.6 (3.3)	70.9 (5.9)	82.5 (2.7)	74.0 (3.7)
16	8	76.1 (0.0)	76.6 (7.3)	73.3 (4.1)	68.1 (4.9)	83.6 (0.0)	73.0 (1.8)

Table 4.20 Abnormal, lead 0, statistical extractors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the abnormal data sets correctly classified are reported.

		Padding			Truncation		
r	w	Identity	Fourier	Wavelet	Identity	Fourier	Wavelet
2	0	60.9 (1.3)	72.2 (2.7)	62.9 (4.4)	60.0 (2.5)	74.8 (1.5)	70.1 (0.0)
2	1	60.8 (1.7)	70.6 (3.6)	62.8 (3.7)	58.8 (3.9)	74.0 (1.8)	71.3 (3.9)
4	0	61.2 (0.0)	71.9 (2.1)	63.9 (4.3)	58.2 (3.1)	74.3 (1.1)	70.1 (0.0)
4	1	61.2 (0.0)	70.1 (4.7)	63.8 (4.0)	57.2 (4.3)	73.7 (1.1)	71.6 (2.7)
4	2	60.9 (1.3)	68.9 (5.9)	65.1 (4.9)	58.7 (7.7)	73.4 (1.2)	71.9 (3.4)
8	0	61.2 (0.0)	71.8 (2.0)	63.7 (4.2)	56.4 (2.5)	74.5 (1.4)	71.0 (2.2)
8	1	61.2 (0.0)	70.8 (3.2)	65.7 (3.7)	59.4 (6.5)	74.0 (1.2)	72.2 (2.9)
8	2	61.2 (0.0)	70.2 (4.2)	65.4 (6.3)	61.1 (7.7)	73.9 (1.1)	73.7 (3.0)
8	4	61.2 (0.0)	68.7 (5.7)	64.6 (3.7)	63.8 (9.7)	73.4 (1.0)	75.4 (1.8)
16	0	61.2 (0.0)	71.6 (0.3)	63.1 (3.9)	55.2 (0.0)	74.4 (1.4)	72.5 (3.0)
16	1	61.2 (0.0)	71.0 (2.7)	63.3 (4.1)	61.2 (8.9)	73.6 (1.0)	74.9 (2.5)
16	2	61.2 (0.0)	70.4 (3.7)	63.0 (4.0)	63.8 (9.7)	73.4 (0.9)	75.2 (2.2)
16	4	61.2 (0.0)	68.7 (5.0)	64.0 (3.6)	64.9 (10.2)	73.4 (0.7)	75.2 (2.2)
16	8	61.2 (0.0)	66.9 (5.8)	64.7 (3.4)	68.6 (8.9)	73.2 (0.3)	76.1 (0.0)

Table 4.21 Abnormal, lead 1, statistical extractors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the abnormal data sets correctly classified are reported.

	Normal		Abnormal	
Lead 0	Padding	Truncation	Padding	Truncation
Identity	78.2 (0.0)	80.2 (4.3)	76.1 (0.0)	73.7 (4.9)
Fourier	84.4 (2.4)	90.3 (2.0)	74.5 (5.0)	79.7 (3.8)
Wavelet	87.1 (2.6)	83.3 (5.4)	72.3 (3.3)	74.8 (5.5)

	Normal		Abnormal	
Lead 1	Padding	Truncation	Padding	Truncation
Identity	84.2 (0.3)	86.1 (4.4)	61.1 (0.7)	60.5 (7.6)
Fourier	81.8 (2.0)	84.5 (2.5)	70.3 (4.2)	73.8 (1.3)
Wavelet	88.7 (3.0)	84.5 (9.0)	64.0 (4.2)	73.0 (3.1)

Table 4.22 Classification accuracies for the normal and abnormal data sets of the lead 0 and lead 1 parameters averaged across the various levels of the training set size and the training set composition experimental factors for the statistical feature extractors. The means (and standard deviations) of the percents of the data sets correctly classified are reported. When the experimental factors are held constant except for data preprocessing, the pairwise differences between the classification accuracies when using padding versus truncation are all statistically significant (based on a one-tailed t-test with $p < .001$), except with the identity feature extractor for the abnormal data sets of the lead 1 parameter.

	Normal		Abnormal	
	Lead 0	Lead 1	Lead 0	Lead 1
Identity	Truncation	Truncation	Padding	$*$
Fourier	Truncation	Truncation	Truncation	Truncation
Wavelet	Padding	Padding	Truncation	Truncation

Table 4.23 The more effective data preprocessing technique for the normal and abnormal data sets of the lead 0 and lead 1 parameters averaged across the various levels of the training set size and the training set composition experimental factors for the statistical feature extractors. The data preprocessing technique resulting in a statistically significant increase in classification accuracy over the other is reported. There is no statistical difference between the two data preprocessing techniques with the identity feature extractor for the abnormal data sets of the lead 1 parameter.

- When the experimental factors are held constant except for data preprocessing, the classification accuracies are higher using truncation as compared to padding with the Fourier and wavelet feature extraction methods for the abnormal data sets of each parameter. The classification accuracies are higher using padding as compared to truncation with the identity feature extraction method for the abnormal data sets of the lead 0 parameter; the classification accuracies are dependent on the levels of the training set size and the training set composition factors for the abnormal data sets of the lead 1 parameter.

- The classification accuracies are generally better with the identity and Fourier feature extraction methods for the abnormal data sets of the lead 0 parameter versus the abnormal data sets of the lead 1 parameter for all combinations of the remaining experimental factors. The classification accuracies are generally better with the wavelet feature extraction method for the abnormal data sets of the lead 0 parameter versus the abnormal data sets of the lead 1 parameter when using padding for all combinations of the remaining experimental factors; the classification accuracies depend on the values of the training set size and training set composition factors for the abnormal data sets of both parameters.

The most interesting observation is that the classification accuracies with the identity and Fourier feature extraction methods for the normal and abnormal data sets of each parameter are not influenced by the values of the remaining experimental factors. In contrast, the classification accuracies with the wavelet feature extraction method for the normal and abnormal data sets of both parameters are affected by the values of the remaining experimental factors.

Table 4.22 reports the classification accuracies for the normal and abnormal data sets of the lead 0 and lead 1 parameters averaged across the various levels of the training set size and the training set composition experimental factors for the statistical feature extractors. For each combination of the feature extraction method and the data preprocessing experimental factors, the mean and standard deviation of the percents of the data sets correctly classified are reported. When the experimental factors are held constant except for data preprocessing, the pairwise differences between the classification accuracies when using padding versus truncation are all statistically significant (based on a one-tailed t-test with $p < .001$), except with the identity feature extractor for the abnormal data sets of the lead 1 parameter. Table 4.23 reports the more effective data preprocessing technique for the normal and abnormal data sets of the lead 0 and lead 1 parameters averaged across the various levels of the training set size and the training set composition experimental factors for the statistical feature extractors. The data preprocessing technique resulting in a statistically significant increase in classification accuracy over the other is reported. For the normal data sets of each parameter, the identity and Fourier feature extraction methods result in a higher classification accuracy when using truncation as compared to padding; the wavelet feature extraction method results in a higher classification accuracy when using padding as compared to truncation. For the abnormal data sets, the wavelet and Fourier feature extractors result in a higher classification accuracy when using truncation as compared to padding; the identity feature extraction method results in a higher classification accuracy when using padding as compared to truncation.

Tables 4.24 and 4.26 report the classification accuracies for the normal and abnormal data sets, respectively, of the lead 0 parameter using the structure detectors and broken down by combinations of the experimental factors. Tables 4.25 and 4.27 report the parallel results for the lead 1 parameter. The classification accuracy for each combination of experimental factors is reported as the mean

r	w	Constant	Straight	Exponential	Sinusoidal	Triangular	Trapezoidal	Composite
2	0	85.2 (5.0)	88.0 (0.0)	87.0 (1.1)	90.3 (2.6)	87.4 (1.2)	83.0 (4.3)	87.1 (1.2)
2	1	83.6 (3.7)	87.9 (0.5)	87.2 (0.8)	90.2 (2.6)	86.9 (1.8)	82.6 (3.8)	86.5 (2.0)
4	0	86.5 (4.6)	88.0 (0.0)	87.1 (0.8)	90.9 (2.2)	87.8 (0.7)	82.1 (3.9)	87.2 (0.9)
4	1	85.3 (5.0)	88.0 (0.0)	86.9 (1.1)	91.0 (1.8)	87.7 (0.9)	82.6 (4.2)	87.1 (1.2)
4	2	84.9 (4.6)	87.7 (1.2)	87.3 (1.1)	91.0 (1.8)	87.5 (1.1)	84.3 (4.6)	86.6 (1.2)
8	0	86.1 (1.8)	88.0 (0.0)	86.9 (0.7)	91.5 (0.8)	88.0 (0.0)	81.1 (3.3)	87.2 (0.0)
8	1	86.1 (1.8)	88.0 (0.0)	86.8 (0.6)	91.7 (0.0)	88.0 (0.0)	82.0 (4.0)	87.2 (0.0)
8	2	85.7 (2.7)	88.0 (0.0)	87.0 (0.7)	91.7 (0.0)	87.7 (0.9)	81.5 (3.7)	87.1 (0.7)
8	4	85.7 (4.6)	88.0 (0.0)	87.2 (0.8)	91.5 (0.8)	87.5 (1.1)	82.4 (4.2)	86.9 (0.9)
16	0	85.7 (0.0)	88.0 (0.0)	86.8 (0.7)	91.7 (0.0)	88.0 (0.0)	82.0 (4.0)	87.2 (0.0)
16	1	86.1 (1.8)	88.0 (0.0)	86.8 (0.7)	91.7 (0.0)	88.0 (0.0)	82.0 (4.0)	87.2 (0.0)
16	2	87.4 (3.4)	88.0 (0.0)	86.8 (0.7)	91.7 (0.0)	88.0 (0.0)	81.1 (3.3)	87.2 (0.0)
16	4	87.8 (3.7)	88.0 (0.0)	86.8 (0.7)	91.7 (0.0)	88.0 (0.0)	82.0 (4.0)	87.2 (0.0)
16	8	87.0 (3.0)	88.0 (0.0)	86.8 (0.6)	91.7 (0.0)	87.4 (1.2)	79.7 (0.0)	87.1 (0.7)

Table 4.24 Normal, lead 0, structure detectors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the normal data sets correctly classified are reported.

r	w	Constant	Straight	Exponential	Sinusoidal	Triangular	Trapezoidal	Composite
2	0	88.2 (0.9)	84.9 (3.5)	78.3 (1.9)	91.0 (3.1)	77.4 (0.0)	78.9 (0.3)	80.1 (3.7)
2	1	88.4 (1.6)	85.2 (3.4)	80.1 (3.4)	91.4 (2.8)	79.7 (4.6)	78.9 (0.0)	80.1 (3.7)
4	0	88.4 (0.7)	84.2 (3.5)	79.2 (2.3)	91.7 (2.3)	77.4 (0.0)	78.9 (0.0)	78.9 (3.1)
4	1	88.2 (0.9)	85.3 (3.4)	79.5 (2.3)	91.4 (2.8)	77.4 (0.0)	78.9 (0.0)	79.7 (3.5)
4	2	88.4 (1.5)	85.2 (3.4)	80.2 (2.9)	91.0 (3.1)	77.4 (0.0)	78.9 (0.0)	80.1 (3.7)
8	0	88.5 (0.6)	85.9 (3.2)	78.6 (2.0)	91.7 (2.3)	77.4 (0.0)	78.9 (0.0)	78.9 (3.1)
8	1	88.3 (0.8)	85.6 (3.3)	78.1 (1.7)	91.7 (2.3)	77.4 (0.0)	78.9 (0.0)	79.7 (3.5)
8	2	88.2 (0.8)	84.2 (3.5)	79.7 (2.3)	92.5 (0.0)	77.4 (0.0)	78.9 (0.0)	79.3 (3.3)
8	4	88.0 (1.0)	85.6 (3.3)	79.7 (2.3)	91.4 (2.8)	77.4 (0.0)	78.9 (0.0)	80.5 (3.8)
16	0	88.6 (0.3)	84.9 (3.5)	78.8 (2.1)	92.5 (0.0)	77.4 (0.0)	78.9 (0.0)	78.6 (2.8)
16	1	88.6 (0.5)	84.2 (3.5)	78.6 (2.0)	92.5 (0.0)	77.4 (0.0)	78.9 (0.0)	78.2 (2.3)
16	2	88.3 (0.7)	84.9 (3.5)	79.5 (2.3)	92.5 (0.0)	77.4 (0.0)	78.9 (0.0)	78.9 (3.1)
16	4	88.2 (0.8)	84.2 (3.5)	79.5 (2.3)	92.5 (0.0)	77.4 (0.0)	78.9 (0.0)	80.1 (3.7)
16	8	87.9 (0.8)	86.3 (3.0)	79.7 (2.3)	92.5 (0.0)	77.4 (0.0)	78.9 (0.0)	82.3 (3.7)

Table 4.25 Normal, lead 1, structure detectors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the normal data sets correctly classified are reported.

r	w	Constant	Straight	Exponential	Sinusoidal	Triangular	Trapezoidal	Composite
2	0	76.0 (9.1)	67.5 (8.8)	73.8 (2.2)	79.1 (2.7)	81.9 (1.6)	81.3 (1.9)	73.3 (3.5)
2	1	78.7 (7.3)	72.1 (9.2)	73.9 (2.3)	78.8 (2.8)	82.3 (1.0)	81.4 (3.7)	72.5 (2.0)
4	0	73.6 (8.1)	68.4 (9.0)	73.4 (2.3)	79.7 (2.2)	81.7 (1.4)	81.6 (1.8)	72.5 (2.6)
4	1	75.9 (9.1)	71.0 (9.1)	73.6 (2.2)	79.7 (2.2)	81.8 (1.5)	81.5 (1.8)	73.3 (3.5)
4	2	76.6 (8.5)	73.0 (8.9)	74.7 (2.1)	79.7 (2.2)	81.9 (1.6)	80.7 (1.8)	72.8 (2.5)
8	0	74.0 (3.0)	67.5 (8.8)	73.0 (2.1)	80.3 (1.3)	82.1 (0.0)	81.6 (1.1)	71.6 (0.0)
8	1	74.0 (3.0)	66.6 (8.4)	72.5 (1.8)	80.6 (0.0)	82.1 (0.0)	81.3 (1.3)	71.6 (0.0)
8	2	74.8 (4.9)	68.4 (9.0)	73.2 (2.2)	80.6 (0.0)	82.2 (0.5)	81.5 (1.2)	71.9 (1.3)
8	4	75.1 (8.4)	71.9 (9.0)	73.9 (2.3)	80.3 (1.3)	82.3 (0.5)	81.2 (1.4)	72.2 (1.8)
16	0	74.6 (0.0)	70.1 (9.2)	72.8 (2.0)	80.6 (0.0)	82.1 (0.0)	81.3 (1.3)	71.6 (0.0)
16	1	74.0 (3.0)	70.1 (9.2)	72.8 (2.0)	80.6 (0.0)	82.1 (0.0)	81.3 (1.3)	71.6 (0.0)
16	2	71.9 (5.5)	69.3 (9.1)	72.8 (2.0)	80.6 (0.0)	82.1 (0.0)	81.6 (1.1)	71.6 (0.0)
16	4	71.3 (6.0)	67.5 (8.8)	72.8 (2.0)	80.6 (0.0)	82.1 (0.0)	81.3 (1.3)	71.6 (0.0)
16	8	72.6 (4.9)	67.5 (8.8)	72.5 (1.8)	80.6 (0.0)	82.4 (0.6)	82.1 (0.0)	71.9 (1.3)

Table 4.26 Abnormal, lead 0, structure detectors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the abnormal data sets correctly classified are reported.

r	w	Constant	Straight	Exponential	Sinusoidal	Triangular	Trapezoidal	Composite
2	0	75.7 (3.6)	76.0 (1.5)	87.5 (7.4)	78.8 (0.6)	91.0 (0.0)	85.3 (1.0)	82.8 (7.3)
2	1	74.6 (3.3)	75.5 (2.2)	83.3 (8.8)	78.9 (0.5)	87.8 (6.7)	85.1 (0.0)	82.8 (7.3)
4	0	75.7 (3.9)	76.3 (1.5)	83.9 (9.0)	79.0 (0.5)	91.0 (0.0)	85.1 (0.0)	85.1 (6.1)
4	1	75.1 (4.1)	75.8 (1.5)	83.0 (9.1)	78.9 (0.5)	91.0 (0.0)	85.1 (0.0)	83.6 (7.0)
4	2	74.6 (4.0)	75.5 (2.2)	81.3 (9.0)	78.8 (0.6)	91.0 (0.0)	85.1 (0.0)	82.8 (7.3)
8	0	76.8 (2.6)	75.5 (1.4)	86.6 (8.0)	79.0 (0.5)	91.0 (0.0)	85.1 (0.0)	85.1 (6.1)
8	1	75.2 (4.3)	75.7 (1.5)	88.4 (6.6)	79.0 (0.5)	91.0 (0.0)	85.1 (0.0)	83.6 (7.0)
8	2	74.9 (4.4)	76.3 (1.5)	82.1 (9.2)	79.1 (0.0)	91.0 (0.0)	85.1 (0.0)	84.3 (6.6)
8	4	74.7 (3.9)	75.7 (1.5)	82.1 (9.2)	78.9 (0.5)	91.0 (0.0)	85.1 (0.0)	82.1 (7.5)
16	0	77.1 (2.3)	76.0 (1.5)	85.7 (8.4)	79.1 (0.0)	91.0 (0.0)	85.1 (0.0)	85.8 (5.5)
16	1	76.6 (3.2)	76.3 (1.5)	86.6 (8.0)	79.1 (0.0)	91.0 (0.0)	85.1 (0.0)	86.6 (4.6)
16	2	75.0 (4.6)	76.0 (1.5)	83.0 (9.1)	79.1 (0.0)	91.0 (0.0)	85.1 (0.0)	85.1 (6.1)
16	4	74.2 (4.9)	76.3 (1.5)	83.0 (9.1)	79.1 (0.0)	91.0 (0.0)	85.1 (0.0)	82.8 (7.3)
16	8	72.6 (5.2)	75.4 (1.3)	82.1 (9.2)	79.1 (0.0)	91.0 (0.0)	85.1 (0.0)	78.4 (7.3)

Table 4.27 Abnormal, lead 1, structure detectors: The classification accuracies are broken down by combinations of the experimental factors where r is the training set size and w is the training set composition. The means (and standard deviations) of the percents of the abnormal data sets correctly classified are reported.

	Normal		Abnormal	
	Lead 0	Lead 1	Lead 0	Lead 1
Constant	85.9 (3.6)	88.3 (0.9)	74.5 (6.5)	75.2 (4.0)
Straight	87.9 (0.3)	85.1 (3.4)	69.3 (9.0)	75.9 (1.6)
Exponential	87.0 (0.8)	79.2 (2.4)	73.3 (2.1)	84.2 (8.7)
Sinusoidal	91.3 (1.4)	91.9 (2.1)	80.1 (1.6)	79.0 (0.4)
Triangular	87.7 (0.9)	77.6 (1.3)	82.1 (0.9)	90.8 (2.0)
Trapezoidal	82.0 (3.9)	78.9 (0.1)	81.4 (1.7)	85.1 (0.3)
Composite	87.1 (0.9)	79.7 (3.4)	72.2 (1.9)	83.6 (6.8)

Table 4.28 Classification accuracies for the normal and abnormal data sets of the lead 0 and lead 1 parameters averaged across the various levels of the training set size and the training set composition experimental factors for the structure detectors. The means (and standard deviations) of the percents of the data sets correctly classified are reported.

Normal		Abnormal	
Lead 0	Lead 1	Lead 0	Lead 1
Sinusoidal	Sinusoidal	Triangular	Triangular
Straight	Constant	Trapezoidal	Trapezoidal
Triangular	Straight	Sinusoidal	Exponential
Composite	Composite	Constant	Composite
Exponential	Exponential	Exponential	Sinusoidal
Constant	Trapezoidal	Composite	Straight
Trapezoidal	Triangular	Straight	Constant

Table 4.29 The structure detectors ordered by classification accuracy for the normal and abnormal data sets of the lead 0 and lead 1 parameters. The structure detectors are listed from most (top) to least (bottom) accurate.

and standard deviation of the percents of the data sets correctly classified across twenty iterations of the experiment procedure. Several trends are apparent:

- While the classification accuracies fluctuate across the combinations of the experimental factors for the normal and abnormal data sets of each parameter, the effect of the training set size and the training set composition is minimal.
- The classification accuracies with each structure detector (except constant and sinusoidal) are better for the normal data sets of the lead 0 parameter than for the abnormal data sets of the lead 1 parameter for all combinations of the remaining experimental factors. The accuracies with each structure detector (except constant and sinusoidal) are better for the abnormal data sets of the lead 1 parameter than for the abnormal data sets of the lead 0 parameter for all combinations of the remaining experimental factors.
- The classification accuracies for the normal data sets of lead 0 parameter are better than the classification accuracies for the abnormal data sets of the lead 0 parameter across the experimental factors. The differences between the classification accuracies for the normal and abnormal data sets of lead 1 depend on the feature extraction method.

Notice that the classification accuracies when using the composite structure detector, which combines the heterogeneous structure detectors to extract features from a data set, are never the best and yet rarely the worst as compared to the classification accuracies when using each heterogeneous structure detector individually. The underlying cause for this counterintuitive result is discussed in Section 4.5.

Table 4.28 reports the classification accuracies for the normal and abnormal data sets of the lead 0 and lead 1 parameters averaged across the various levels of the training set size and the training set composition experimental factors for the structure detectors. The classification accuracies for the normal and abnormal data sets for each parameter are dependent on the feature extraction method. Table 4.29 lists the structure detectors sorted by the classification accuracies for the normal and abnormal data sets of each parameter. The sinusoidal structure detector is best for normal data sets regardless of parameter, and the triangular structure detector is best for abnormal data sets regardless of parameter. No one structure detector, however, stands out as being consistently superior to the others.

To compare the efficacy of the structure detectors versus the statistical feature extraction methods, the overall ability to correctly classify both the normal and abnormal data sets of each parameter must be examined. Figures 4.16 and 4.17 illustrate the relative classification accuracies of the ten feature extraction methods (note that the scales of the x and y axes on each graph are not uniform and differ between the two graphs). Each feature extraction method is plotted using the mean percent of the normal and abnormal data sets correctly classified as coordinates: the results in Table 4.22 are used to plot the statistical feature extraction methods (the results with both padding and truncation are used), and the results in Table 4.28 are used to plot the structure detectors. A dashed line encloses the three statistical feature extraction methods using both types of data preprocessing; the structure detectors with better classification accuracies are those which are plotted further towards the upper-right of the graph relative to the dashed area. For the lead 0 parameter, the sinusoidal structure detector performs slightly better overall than the statistical feature extraction methods. The triangular and trapezoidal structure detectors are superior to the statistical feature extraction

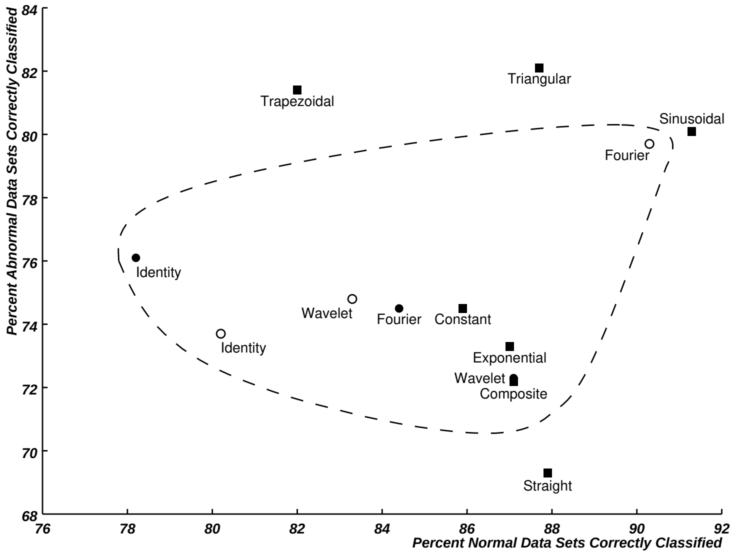


Figure 4.16 The relative classification accuracies for all feature extraction methods for the lead 0 parameter. Each feature extraction method is plotted using the mean percent of the normal and abnormal data sets correctly classified as coordinates. A dashed line encloses the three statistical feature extractors for both types of data preprocessing: padding (plotted with solid bullets) and truncation (plotted with hollow bullets).

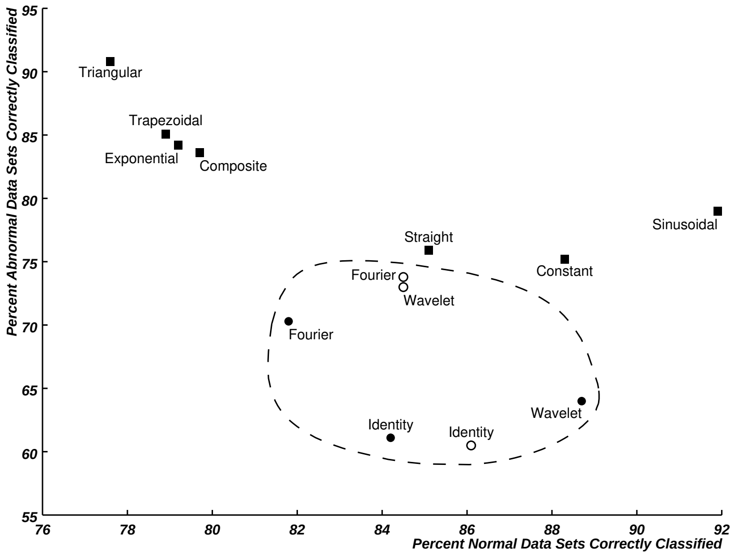


Figure 4.17 The relative classification accuracies for all feature extraction methods for the lead 1 parameter. Each feature extraction method is plotted using the mean percent of the normal and abnormal data sets correctly classified as coordinates. A dashed line encloses the three statistical feature extractors for both types of data preprocessing: padding (plotted with solid bullets) and truncation (plotted with hollow bullets).

methods in terms of their classification accuracies for the abnormal data sets, while the straight structure detector performs notably poorly. For the lead 1 parameter, the constant and sinusoidal structure detectors perform better overall than the statistical feature extraction methods. Moreover, the remaining structure detectors all are superior to the statistical feature extraction methods in terms of the classification accuracies for the abnormal data sets.

4.5. Discussion

Two demonstrably different domains, namely semiconductor fabrication and electrocardiography, were used to assess the efficacy of the structure detectors as general-purpose feature extractors for structural pattern recognition. The classification accuracies achieved using features extracted by the structure detectors under a range of conditions influencing the outcome of the associated training phase were compared to the corresponding accuracies achieved by commonly-used statistical methods. The overall conclusions of the experiment are as follows:

- The classification accuracies achieved by the structure detectors are generally as good as or better than those achieved by the statistical feature extraction methods. Rarely did any of the structure detectors perform poorer overall than the statistical feature extraction methods.

- The effect on classification accuracy by the padding and truncation data preprocessing techniques is dependent on the feature extraction method and the characteristics of the data—sometimes padding results in better classification accuracy than truncation, and sometimes the reverse is true. Common wisdom recommends zero padding over truncation for data preprocessing, but these results to not bear this out. Alternative data preprocessing techniques (e.g., extrapolation) may avoid the drawbacks associated with both zero padding and truncation.

- While the training set size and the training set composition influence the classification accuracy for all feature extraction methods, the effect is inconsistent. It should be noted, however, that acceptable classification accuracies can be achieved with small training set sizes, regardless of the training set composition.

No one structure detector consistently produced the best classification accuracies. The inability of the composite structure detector to fill this role is counterintuitive: the composite structure detector incorporates each of the heterogeneous structure detectors and, therefore, should be able to perform at least as well as the best heterogeneous structure detector. The failure of the composite structure detector to consistently outperform the heterogeneous structure detectors is rooted in the methodology employed to select among the six morphology types: the structure that has the smallest sum of squared error is used to represent each subregion. The composite structure detector, therefore, extracts features based upon how closely the data is represented rather than how well the features discriminate among groups. Additionally, the outcome of the training phase associated with the structure detectors only determines the number of subregions, thus allowing the sequence of morphology types used by the composite structure detector to vary among the data sets. Using different morphology types to represent the same subregion across data sets with the same group label introduces variability that can confuse the classifier. Modifying the training phase associated with the composite structure detector to determine a fixed sequence of morphology types may improve the overall performance of the composite structure detector.

The ability of the structure detectors to achieve classification accuracies as good as or better than the baseline statistical methods in the experiment demonstrates that the structure detectors are worthwhile feature extractors that perform well on data with disparate characteristics. No one structure detector performed well across all experimental conditions, raising the question of how best to proceed when confronted with a new, unexplored data set. A simplistic approach would be to run all structure detectors and select the one that results in the most accurate classification. A more rigorous approach involves creating a taxonomy of time-series data based on characteristics of data drawn from different sources and identifying the efficacy of each structure detector within various subregions of the taxonomy. Given such a mapping from a set of data characteristics to a structure detector, the characteristics of a new data set can be used to select the structure detector that is likely to be most effective. A third, more interactive, approach would allow a user to select structures to fit to subregions of the data, evaluate the resulting classification, and iterate until a satisfactory result is obtained.

Chapter 5

Conclusions

Structural pattern recognition can be a powerful analysis tool within domains where a description composed of morphological subpatterns and their interrelationships is paramount to accurate classification decisions. A structural pattern recognition system typically includes feature extractors to identify instances of morphological characteristics of the data which, in turn, are used as the basis for classification using syntactic grammars. The domain knowledge necessary to guide feature extractor and grammar development is gathered using knowledge acquisition techniques. However, such techniques are time consuming, inexact, and do not always produce a complete knowledge base of the domain. Consequently, structural approaches to pattern recognition are difficult to apply to unexplored or poorly-understood domains, thus limiting them to domains where the feature types and the syntactic grammars have either become established in the literature or are obvious upon inspection of the data. Eliminating the effort necessary to implement feature extraction and classification for structural pattern recognition systems will widen the applicability of structural approaches to complex, poorly-understood domains. This can be accomplished using domain-independent techniques for feature extraction and classification.

A domain-independent structural pattern recognition system is one that is capable of extracting features and performing classification without the need for domain knowledge. Such a system can be implemented using a hybrid approach that incorporates structural features with statistical techniques for classification: the structural features retain the morphological information necessary for discrimination, while the statistical classifier avoids the need to develop syntactic grammars that are inherently domain- and application-specific. The solution to making feature extraction domain-independent is to employ generalized feature extraction to identify instances of morphologies which have proven to be useful across domains.

To address the problem of generalized feature extraction within domains involving time-series data, a suite of structure detectors based on structural features commonly cited as useful in the pattern recognition literature and used in signal processing was developed. Structure detectors were implemented to approximate a time-series data set with one of six morphologies—constant, straight, exponential, sinusoidal, triangular, and trapezoidal. A methodology for applying these structure detectors to a time-series data set in a piecewise fashion was developed, producing either a homogeneous or heterogeneous sequence of structures that together best approximate the entire time series. The efficacy of these structure detectors to generate morphological features suitable for classification was assessed against three standard statistical techniques for feature extraction—the identity, Fourier, and wavelet transformations—using two databases having markedly different characteristics. The classification accuracies achieved when using the structure detectors were at least as good as (and often superior to) the classification accuracies achieved when using the statistical feature extractors.

The ability of the structure detectors to generate morphological features that result in classification accuracies better than the baseline established by commonly-used statistical techniques demonstrates that the morphologies identified by the suite of structure detectors constitute a useful set of structural feature types. Moreover, the classification accuracies achieved on the two disparate databases illustrate that the suite of structure detectors is capable of extracting features from data with various characteristics. Certainly it is possible to produce better classification accuracies with domain- and application-specific feature extractors developed with the assistance of a domain expert, but this is burdensome for well-understood domains and impossible for domains that are poorly understood. What this suite of structure detectors offers is a starting point for extracting features which have been shown to be generally effective for classification, providing a springboard for domain exploration and the subsequent refinement of these structure detectors with the goal of producing a structural pattern recognition system targeted to a particular domain and application.

5.1. Contributions

Several contributions within the field of pattern recognition have been made in the course of this research. Those contributions include the following:

- A suite of structure detectors was designed and implemented, as described in Chapter 3, to extract structural features in time-series data based on morphologies suggested by both the pattern recognition literature and the field of signal processing.
- An evaluation was performed to compare the classification accuracies achieved when using the structure detectors to extract features versus commonly-used statistical feature extraction methods, as described in Chapter 4. The classification accuracies achieved by the structure detectors were at least as good as those achieved by the baseline methods.
- Empirical evidence that refutes the generally-accepted heuristic to use zero padding for data preprocessing was produced by the experiment performed to evaluate the structure detectors. The evidence suggests that zero padding does not consistently result in superior classification accuracy, as discussed in Section 4.5.
- A technique for characterizing the aggregate structural composition of time-series data using chain codes was proposed and used to demonstrate the differences between databases, as described in Section 4.3.

5.2. Future Work

The development and evaluation of the suite of structure detectors for generalized feature extraction in time-series data serves as a solid foundation for continuing research within the fields of pattern recognition and knowledge acquisition. Topics of investigation include the following:

- The suite of structure detectors can be used as a cue generator to suggest morphologies that may have significance within a domain. In this way, the structure detectors can assist an expert to recall implicit domain knowledge and provide a foundation for expressing that knowledge.

- The structure detectors can be used to produce a baseline morphological description from which improvements can be made iteratively until converging on an acceptable set of features to extract. Using such an approach for knowledge acquisition could reduce the overall effort associated with assembling domain knowledge.

- Solving tedious, expert-level problems could be accomplished by novices by focusing their attention on the discriminatory features. The suite of structure detectors could be used to achieve this goal by eliminating noise from time-series data, thus clarifying the underlying morphology of the waveform and elucidating for the novice the previously-obscured features. Using such an approach could enable novices to discriminate among classes to a level comparable to that of experts.

- Evaluating the generality of feature extractors necessarily requires that they be applied to a variety of databases which span the range of data characteristics. Chain codes is one approach that can be employed to describe time-series data in terms of the frequency and degree to which the data change over time. A more inclusive methodology is required to capture other pertinent aspects of the data—such as the duration and magnitude of morphological events—to arrive at a more descriptive taxonomy of time-series data. Once completed, such a taxonomy can be used to evaluate the generality of feature extractors by analyzing databases that represent various combinations of data characteristics.

- Only one classifier was used in the experiment to evaluate the efficacy of the structure detectors. A natural question is whether the results reported here are applicable when other classifiers are used. To address this issue, additional experiments can be performed with other classifiers (essentially introducing the selection of the classifier as an additional experimental factor). For example, the efficacy of neural networks across the various combinations of values for the experimental factors would be worthwhile to investigate.

- Classification was performed using the features generated by an individual structure detector. Generating features using combinations of structure detectors could result in increased classification accuracies. Experiments can be performed to determine which combinations are most effective and how best to unify their results (e.g., perform classification using all of the features extracted by multiple structure detectors, or weight the outcomes of multiple classifications performed using features extracted by individual structure detectors).

- Classification was performed using feature vectors comprising features extracted from only one parameter at a time. Classification accuracy could be improved by extracting features from all parameters and combining them into the same feature vector. For example, the data sets in the wafer database were classified based on the features extracted from either the 405 nm parameter or the 520 nm parameter; concatenating the individual feature vectors for each parameter into one large feature vector could improve classification accuracy.

- The structure detectors can be used to compress time-series data. The degree of compression is determined by the number of subregions used by the structure detectors to approximate the raw data: the more subregions, the higher the accuracy but at the expense of less compression. The efficacy of the structure detectors for data compression can be evaluated using various data sets to determine the degree and accuracy of the resulting compression.

- The suite of structure detectors was limited to univariate features (i.e., morphologies that manifest in the consecutive data points of a single time-series data set). In some situations, multivariate features (i.e., morphologies that manifest in the relationship between consecutive data points across time-series data sets) would be appropriate and would require structure detectors specifically designed to extract features comprising multiple time-series data sets. Alternatively, a technique for combining the output of univariate structure detectors into multivariate features could also be effective.

- The composite superstructure allows the particular sequence of morphologies fitted to a collection of time-series data sets to change, selecting the sequence that minimizes the sum of squared error for each data set. This contributes variability to the features and may result in a reduction in classification accuracy. To eliminate this problem, the composite superstructure can be modified so as to approximate a collection of time-series data sets using a fixed sequence of morphologies. The particular sequence of morphologies to extract can be determined using a modified version of the structure detector training algorithm that analyzes a training set to determine both the optimal number of subregions and the type of structure to fit to each subregion.

Appendix A

Feature Vector Examples

A statistical classifier can be used to discriminate among time-series data sets based upon a collection of associated feature vectors such that each vector contains features extracted from one of the time-series data sets under analysis and each data set is represented by a vector within the collection. For a particular feature extraction method, each feature vector is expected to be formated so that the i^{th} feature in each vector describes the same characteristic in each data set. To satisfy this requirement, each of the statistical feature extraction methods—identity, Fourier, and wavelet transformations—and the structure detectors have a specific format into which the extracted features are arranged, as described in Section 4.2. To generate a feature vector that represents a time-series data set for classification, the feature extraction method is applied to the data set, an approximation to the data is generated, and the individual components of the approximation are arranged into a feature vector that is formated according to the particular feature extraction method employed.

For example, each of the feature extraction methods was applied to the same time-series data set having a length n equal to ten. Figure A.1 shows the approximations to the data set generated by the identity, Fourier, and wavelet transformations as well as the structure detectors when used to produce a composite, or heterogeneous, approximation. Each graph shows the input time-series data plotted with hollow bullets overlaid with a solid line representing the approximation. For the identity transformation, the data set was neither truncated nor padded, remaining at its length of ten. For the Fourier and wavelet transformations, the data set was padded to a length of sixteen (i.e., six values of t were added, each associated with a $Y(t)$ value of zero), and the top two frequencies or transformations were allowed to contribute to the approximations. For the structure detectors, two subregions were used and are separated by a vertical dashed line: the first subregion was approximated with an exponential structure, and the second with a constant structure.

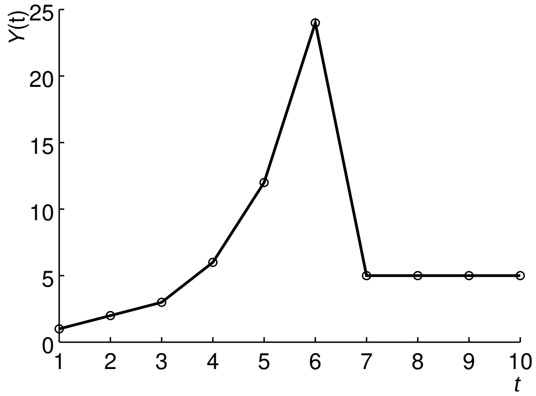
The feature vector used to represent this data set depends on the feature extraction method. The approximation $\hat{Y}(t)$ generated by the identity transformation shown in Figure A.1(a) is

$$\hat{Y}(t) = Y(t)$$

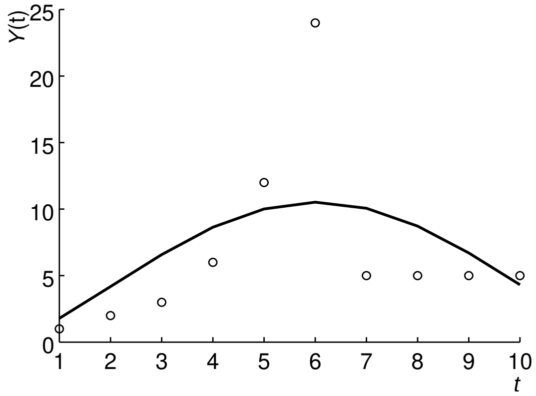
where $Y(t)$ is a value in the original data set such that $1 \leq t \leq n$. The feature vector assembled from the components of this approximation is

$$1.0\ \ 2.0\ \ 3.0\ \ 6.0\ \ 12.0\ \ 24.0\ \ 5.0\ \ 5.0\ \ 5.0\ \ 5.0$$

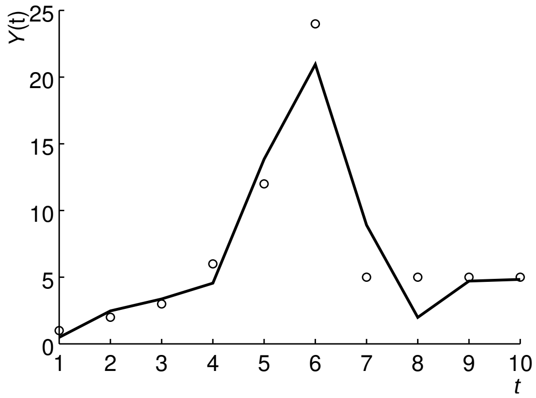
where element i of the feature vector (from left to right) is equal to $\hat{Y}(i)$ such that $1 \leq i \leq 10$. Note that since the length of the original data set is ten, the resulting feature vector contains a total of ten elements.



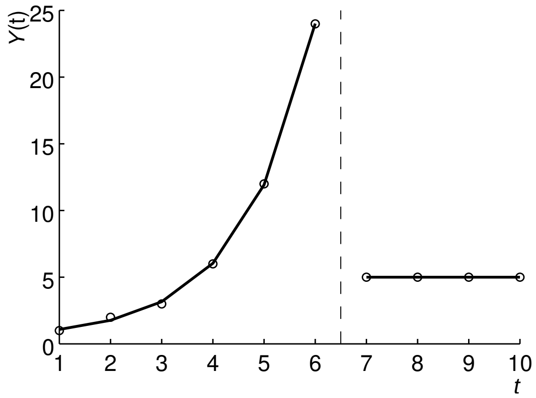
(a) Identity Transformation



(b) Fourier Transformation



(c) Wavelet Transformation



(d) Structure Detectors

Figure A.1 The approximations to a common data set generated by the identity, Fourier, and wavelet transformations as well as the structure detectors when used to produce a composite, or heterogeneous, approximation. Each graph shows the input time-series data plotted with hollow bullets overlaid with a solid line representing the approximation. For the identity transformation, the data set was neither truncated nor padded. For the Fourier and wavelet transformations, the data set was padded, and the top two frequencies or transformations were allowed to contribute to the approximations. For the structure detectors, two subregions were used and are separated by a vertical dashed line: the first subregion was approximated with an exponential structure, and the second with a constant structure.

The approximation $\hat{Y}(t)$ generated by the Fourier transformation shown in Figure A.1(b) is

$$\hat{Y}(t) = \sum_{j \in B} (a_j \cos jt + b_j \sin jt)$$

where the subset of frequencies B is $\{0,1\}$, a_0 is equal to 4.3, b_0 is equal to 0.0, a_1 is equal to -2.5, and b_1 is equal to 5.8. The feature vector assembled from the components of this approximation is

$$4.3 \quad 0.0 \quad -2.5 \quad 5.8$$

where the first and second elements of the feature vector (from left to right) are equal to a_0 and b_0, respectively, and the third and fourth elements are equal to a_1 and b_1, respectively. Note that since only the top two frequencies were allowed to contribute to the approximation, the resulting feature vector contains two pairs of values and has a total of four elements.

The approximation $\hat{Y}(t)$ generated by the wavelet transformation shown in Figure A.1(c) is

$$\hat{Y}(t) = \sum_{j \in B} \phi_j \psi_j(t)$$

where the subset of transformations B is $\{1,5\}$, ϕ_1 is equal to 22.9, and ϕ_5 is equal to 17.0. The feature vector assembled from the components of this approximation is

$$22.9 \quad 17.0$$

where the first element of the feature vector (from left to right) is equal to ϕ_1, and the second element is equal to ϕ_5. Note that since only the top two transformations were allowed to contribute to the approximation, the resulting feature vector contains a total of two elements.

The approximation $\hat{Y}(t)$ generated by the structure detectors shown in Figure A.1(d) is

$$\hat{Y}(t) = h(Y(t))$$

where the function h is defined in Section 3.4. The first subregion includes the first six values of t, with an onset at $t = 1$ and an offset at $t = 6$, and is approximated with an exponential structure with the function

$$f(Y(t)) = a * |b|^t + c$$

where a is equal to 0.6, b is equal to 36.3, and c is equal to 0.4. The second subregion includes the last four values of t, with an onset at $t = 7$ and an offset at $t = 10$, and is approximated with a constant structure with the function

$$f(Y(t)) = a$$

where a is equal to 5.0. The difference between the subregions (i.e., between $Y(6)$ and $Y(7)$) is -19.0. The feature vector assembled from the components of this approximation is

3 0.0 0.0 0.0 0.6 36.3 0.4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1 6 6 –19.0
1 5.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 7 10 4

where the feature vector has been split across two lines due to its length. The first nineteen elements (from left to right, top to bottom) describe the approximation of the first subregion, the twentieth element describes the difference between the two subregions, and the final nineteen elements describe the approximation of the second subregion. The first nineteen elements are set as follows:

1 The structure type used to approximate the subregion. Set to 3.
2 The value of the free parameter a in the constant structure. Set to 0.0.
3 The value of the free parameter a in the straight structure. Set to 0.0.
4 The value of the free parameter b in the straight structure. Set to 0.0.
5 The value of the free parameter a in the exponential structure. Set to 0.6.
6 The value of the free parameter b in the exponential structure. Set to 36.3.
7 The value of the free parameter c in the exponential structure. Set to 0.4.
8 The value of the free parameter a in the sinusoidal structure. Set to 0.0.
9 The value of the free parameter b in the sinusoidal structure. Set to 0.0.
10 The value of the free parameter c in the sinusoidal structure. Set to 0.0.
11 The value of the free parameter a in the triangular structure. Set to 0.0.
12 The value of the free parameter b in the triangular structure. Set to 0.0.
13 The value of the free parameter c in the triangular structure. Set to 0.0.
14 The value of the free parameter a in the trapezoidal structure. Set to 0.0.
15 The value of the free parameter b in the trapezoidal structure. Set to 0.0.
16 The value of the free parameter c in the trapezoidal structure. Set to 0.0.
17 The onset of the subregion. Set to 1.
18 The offset of the subregion. Set to 6.
19 The length of the subregion. Set to 6.

The twentieth element is set to the difference between the subregions, which is -19.0. The final nineteen elements are set as follows:

21 The structure type used to approximate the subregion. Set to 1.
22 The value of the free parameter a in the constant structure. Set to 5.0.
23 The value of the free parameter a in the straight structure. Set to 0.0.
24 The value of the free parameter b in the straight structure. Set to 0.0.
25 The value of the free parameter a in the exponential structure. Set to 0.0.
26 The value of the free parameter b in the exponential structure. Set to 0.0.
27 The value of the free parameter c in the exponential structure. Set to 0.0.
28 The value of the free parameter a in the sinusoidal structure. Set to 0.0.
29 The value of the free parameter b in the sinusoidal structure. Set to 0.0.
30 The value of the free parameter c in the sinusoidal structure. Set to 0.0.
31 The value of the free parameter a in the triangular structure. Set to 0.0.
32 The value of the free parameter b in the triangular structure. Set to 0.0.
33 The value of the free parameter c in the triangular structure. Set to 0.0.
34 The value of the free parameter a in the trapezoidal structure. Set to 0.0.
35 The value of the free parameter b in the trapezoidal structure. Set to 0.0.
36 The value of the free parameter c in the trapezoidal structure. Set to 0.0.
37 The onset of the subregion. Set to 7.
38 The offset of the subregion. Set to 10.
39 The length of the subregion. Set to 4.

The numeric value which indicates the structure type used to approximate each subregion is arbitrary: the values from one to six are assigned, in order, to the constant, straight, exponential, sinusoidal, triangular, and trapezoidal structures. Note that since two subregions were used, two groups of nineteen elements plus an additional element to describe the difference between the subregions were used, resulting in a feature vector having a total of thirty-nine elements. Had the structure detectors been used to generate a heterogeneous approximation using two subregions, the resulting feature vector would have been formated exactly the same and would have had the same length, however the particular values in the feature vector would have been different.

A collection of feature vectors for classification is generated by applying a feature extraction method to each data set under analysis and arranging the extracted features into a fixed-length feature vector associated with each data set. For instance, if the wavelet transformation is applied as described above, then each data set is represented by a feature vector that contains exactly two elements. To apply the Fourier transformation, a separate collection of feature vectors would be generated where, in this case, each vector would contain exactly four elements. Classification could then be performed separately on each individual collection of feature vectors.

Bibliography

[1] H. Akaike. Information Theory and an Extension of the Maximum Likelihood Principle. In B. N. Petrov and F. Csáki, editors, *Second International Symposium on Information Theory*, pages 267–281. Akadémiai Kiadó, Budapest, 1973.

[2] Carl Almgren, February 2001. Personal communication.

[3] E. Anderson. The Irises of the Gaspé Peninsula. *Bulletin of the American Iris Society*, 59:2–5, 1935.

[4] John R. Anderson. The Adaptive Nature of Human Categorization. *Psychological Review*, 98(3):409 – 429, July 1991.

[5] John R. Anderson. *Cognitive Psychology and Its Implications*. Worth Publishers, New York, fifth edition, 2000.

[6] D. F. Andrews and A. M. Herzberg. *Data: A Collection of Problems from Many Fields for the Student and Research Worker*. Springer-Verlag, New York, 1985.

[7] Andreas Antoniou. *Digital Filters: Analysis and Design*. McGraw-Hill, New York, 1979.

[8] David L. Banks, Robert T. Olszewski, and Roy A. Maxion. Comparing Methods for Multivariate Nonparametric Regression. Technical Report CMU-CS-99-102, Carnegie Mellon University, January 1999.

[9] Andrew Bateman and Warren Yates. *Digital Signal Processing Design*. Computer Science Press, New York, 1989.

[10] Richard E. Bellman and Stuart E. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1962.

[11] Irving Biederman. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 94(2):115–147, April 1987.

[12] John H. Boose. A Survey of Knowledge Acquisition Techniques and Tools. *Knowledge Acquisition*, 1(1):3–37, March 1989.

[13] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, California, 1984.

[14] A. M. Burton, N. R. Shadbolt, A. P. Hedgecock, and G. Rugg. A Formal Evaluation of Knowledge Elicitation Techniques for Expert Systems: Domain 1. In D. S. Moralee, editor, *Research and Development in Expert Systems IV*, pages 136 –145. Cambridge University Press, Cambridge, England, 1988.

[15] A. M. Burton, N. R. Shadbolt, G. Rugg, and A. P. Hedgecock. Knowledge Elicitation Techniques in Classification Domains. In Yves Kodratoff, editor, *Proceedings of the 8th European Conference on Artificial Intelligence*, pages 85–90. Pitman Publishing, London, 1988.

[16] A. M. Burton, N. R. Shadbolt, G. Rugg, and A. P. Hedgecock. The Efficacy of Knowledge Elicitation Techniques: A Comparison Across Domains and Levels of Expertise. *Knowledge Acquisition*, 2(2):167–178, June 1990.

[17] Jules Constant. *Essentials of Learning Electrocardiography*. The Parthenon Publishing Group, New York, 1997.

[18] J. Cullen and A. Bryman. The Knowledge Acquisition Bottleneck: Time for Reassessment? *Expert Systems*, 5(3):216–225, August 1988.

[19] Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.

[20] Frederic de Coulon. *Signal Theory and Processing*. Artech House, Dedham, Massachusetts, 1986.

[21] Edward R. Dougherty. *An Introduction to Morphological Image Processing*. SPIE Optical Engineering Press, Bellingham, Washington, 1992.

[22] George Dougherty and Philip Lee. Automatic Classification of Good and Faulty Semiconductor Wafers. Unpublished manuscript prepared for the Statistical Practice course at Carnegie Mellon University, April 1999. A copy is available upon request via email to `Robert.Olszewski@cs.cmu.edu`.

[23] Norman R. Draper and Harry Smith. *Applied Regression Analysis*. Wiley, New York, third edition, 1998.

[24] Richard O. Duda, Peter E. Hart, and David E. Stork. *Pattern Classification*. Wiley, New York, second edition, 2001.

[25] Jean-Michel Durocher, D. Robert Hay, and Roger W. Y. Chan. Applications of Waveform and Image Pattern Recognition. In R. A. Vaughan, editor, *Pattern Recognition and Image Processing in Physics*, pages 247–256. Adam Hilger, Bristol, England, 1991.

[26] Edward A. Feigenbaum. Themes and Case Studies of Knowledge Engineering. In Donald Michie, editor, *Expert Systems in the Micro-Electronic Age*, pages 3–25. Edinburgh University Press, Edinburgh, 1979.

[27] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7:179–188, 1936.

[28] Herbert Freeman. On the Encoding of Arbitrary Geometric Configurations. *IRE Transactions on Electronic Computers*, EC-10(2):260–268, June 1961.

[29] Menahem Friedman and Abraham Kandel. *Introduction to Pattern Recognition: Statistical, Structural, Neural, and Fuzzy Logic Approaches*. World Scientific, Singapore, 1999.

[30] K. S. Fu, editor. *Syntactic Pattern Recognition, Applications*. Springer-Verlag, Berlin, 1977.

[31] K. S. Fu. *Syntactic Pattern Recognition and Applications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[32] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, second edition, 1990.

[33] Charles R. Giardina and Edward R. Dougherty. *Morphological Methods in Image and Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[34] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*, 101(23):e215–e220, June 2000. Appears in the *Circulation* Electronic Pages and can be accessed at `http://circ.ahajournals.org/cgi/content/full/101/23/e215`. The resources are available at `http://www.physionet.org`.

[35] Lev Goldfarb. A Unified Approach to Pattern Recognition. *Pattern Recognition*, 17(5):575–582, 1984.

[36] Mervin J. Goldman. *Principles of Clinical Electrocardiography*. Lange Medical Publications, Los Altos, California, ninth edition, 1976.

[37] Avelino J. Gonzalez and Douglas D. Dankel. *The Engineering of Knowledge-Based Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[38] Rafael C. Gonzalez and Michael G. Thomason. *Syntactic Pattern Recognition: An Introduction*. Addison-Wesley, Reading, Massachusetts, 1978.

[39] Amara Graps. An Introduction to Wavelets. *IEEE Computational Science & Engineering*, 2(2):50–61, Summer 1995.

[40] Scott D. Greenwald. *Improved Detection and Classification of Arrhythmias in Noise-Corrupted Electrocardiograms Using Contextual Information*. PhD thesis, Harvard-MIT Division of Health Sciences and Technology, Massachusetts Institute of Technology, 1990.

[41] Barbara Hayes-Roth and Frederick Hayes-Roth. Concept Learning and the Recognition and Classification of Exemplars. *Journal of Verbal Learning and Verbal Behavior*, 16(3):321–338, June 1977.

[42] Ronald R. Hocking. Linear Regression. In Samuel Kotz, Norman L. Johnson, and Campbell B. Read, editors, *Encyclopedia of Statistical Sciences*, volume 5, pages 58–64. Wiley, New York, 1985.

[43] Robert R. Hoffman, Nigel R. Shadbolt, A. Mike Burton, and Gary Klein. Eliciting Knowledge from Experts: A Methodological Analysis. *Organizational Behavior and Human Decision Processes*, 62(2):129 –158, May 1995.

[44] Susan L. Horner. *Ambulatory Electrocardiography: Applications and Techniques*. J. B. Lippincott, Philadelphia, Pennsylvania, 1983.

[45] Robert Hultquist. Regression Coefficients. In Samuel Kotz, Norman L. Johnson, and Campbell B. Read, editors, *Encyclopedia of Statistical Sciences*, volume 7, pages 683 – 687. Wiley, New York, 1986.

[46] S. L. S. Jacoby, J. S. Kowalik, and J. T. Pizzo. *Iterative Methods for Nonlinear Optimization Problems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1972.

[47] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4 –37, January 2000.

[48] Martin Jüttner, Terry Caelli, and Ingo Rentschler. Evidence-Based Pattern Classification: A Structural Approach to Human Perceptual Learning and Generalization. *Journal of Mathematical Psychology*, 41(3):244 –259, September 1997.

[49] Toru Kaneko and Masashi Okudaira. Encoding of Arbitrary Curves Based on the Chain Code Representation. *IEEE Transactions on Communications*, COM-33(7):697–707, July 1985.

[50] Harold L. Kennedy. *Ambulatory Electrocardiography Including Holter Recording Technology*. Lea & Febiger, Philadelphia, Pennsylvania, 1981.

[51] Eamonn J. Keogh and Michael J. Pazzani. An Enhanced Representation of Time Series which Allows Fast and Accurate Classification, Clustering and Relevance Feedback. In Rakesh Agrawal, Paul Stolorz, and Gregory Piatetsky-Shapiro, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 239–243. AAAI Press, Menlo Park, California, 1998.

[52] Konstantin B. Konstantinov and Toshiomi Yoshida. Real-Time Qualitative Analysis of the Temporal Shapes of (Bio)process Variables. *AIChE Journal*, 38(11):1703–1715, November 1992.

[53] Antti Koski, Martti Juhola, and Merik Meriste. Syntactic Recognition of ECG Signals by Attributed Finite Automata. *Pattern Recognition*, 28(12):1927–1940, December 1995.

[54] P. R. Krishnaiah and L. N. Kanal, editors. *Classification, Pattern Recognition, and Reduction of Dimensionality*, volume 2 of *Handbook of Statistics*. North-Holland, Amsterdam, 1982.

[55] Kazuhiro Kuroda, Ken Harada, and Masafumi Hagiwara. Large Scale On-Line Handwritten Chinese Character Recognition Using Improved Syntactic Pattern Recognition. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 5, pages 4530 – 4535. IEEE, New York, 1997.

[56] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright. Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998.

[57] Jay Liebowitz, editor. *The Handbook of Applied Expert Systems*. CRC Press, Boca Raton, Florida, 1998.

[58] P. L. Love and M. Simaan. Automatic Recognition of Primitive Changes in Manufacturing Process Signals. *Pattern Recognition*, 21(4):333–342, 1988.

[59] Y. Mallet, D. Coomans, and O. de Vel. Recent Developments in Discriminant Analysis on High Dimensional Spectral Data. *Chemometrics and Intelligent Laboratory Systems*, 35(2):157–173, December 1996.

[60] Yvette Mallet, Danny Coomans, Jerry Kautsky, and Olivier de Vel. Classification Using Adaptive Wavelets for Feature Extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1058–1066, October 1997.

[61] K. I. M. Mckinnon. Convergence of the Nelder-Mead Simplex Method to a Nonstationary Point. *SIAM Journal on Optimization*, 9(1):148–158, 1998.

[62] Nirode Mohanty. *Signal Processing: Signals, Filtering, and Detection*. Van Nostrand Reinhold, New York, 1987.

[63] George B. Moody, March 2001. Personal communication.

[64] George B. Moody and Roger G. Mark. The MIT-BIH Arrhythmia Database on CD-ROM and Software for Use with It. In *Computers in Cardiology*, pages 185–188. IEEE Computer Society Press, Los Alamitos, California, 1991. Information about the CD-ROM is available at `http://ecg.mit.edu/`.

[65] Morton Nadler and Eric P. Smith. *Pattern Recognition Engineering*. Wiley, New York, 1993.

[66] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7:308–313, 1965.

[67] David Noton. A Theory of Visual Pattern Perception. *IEEE Transactions on Systems Science and Cybernetics*, SSC-6(4):349–357, October 1970.

[68] R. Todd Ogden. *Essential Wavelets for Statistical Applications and Data Analysis*. Birkhäuser, Boston, 1997.

[69] Alan V. Oppenheim and Alan S. Willsky. *Signals & Systems*. Prentice-Hall, Upper Saddle River, New Jersey, second edition, 1997.

[70] T. Pavlidis. *Structural Pattern Recognition*. Springer-Verlag, Berlin, 1977.

[71] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, England, second edition, 1992.

[72] Stephen K. Reed. Pattern Recognition and Categorization. *Cognitive Psychology*, 3(3):382–407, July 1972.

[73] Stephen K. Reed. *Psychological Processes in Pattern Recognition*. Academic Press, New York, 1973.

[74] Stephen K. Reed. Structural Descriptions and the Limitations of Visual Images. *Memory & Cognition*, 2(2):329–336, April 1974.

[75] Stephen K. Reed and Jeffrey A. Johnsen. Detection of Parts in Patterns and Images. *Memory & Cognition*, 3(5):569–575, September 1975.

[76] Raghunathan Rengaswamy and Venkat Venkatasubramanian. A Syntactic Pattern-Recognition Approach for Process Monitoring and Fault Diagnosis. *Engineering Applications of Artificial Intelligence*, 8(1):35–51, 1995.

[77] Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, March 1978.

[78] Richard G. Shiavi and John R. Bourne. Methods of Biological Signal Processing. In Tzay Y. Young and King-Sun Fu, editors, *Handbook of Pattern Recognition and Image Processing*, chapter 22, pages 545–568. Academic Press, Orlando, Florida, 1986.

[79] R. H. Shumway. Discriminant Analysis for Time Series. In P. R. Krishnaiah and L. N. Kanal, editors, *Classification, Pattern Recognition, and Reduction of Dimensionality*, volume 2 of *Handbook of Statistics*, chapter 1, pages 1–46. North-Holland, Amsterdam, 1982.

[80] Jasprit Singh. *Semiconductor Devices: Basic Principles*. Wiley, New York, 2001.

[81] E. Skordalakis. Syntactic ECG Processing: A Review. *Pattern Recognition*, 19(4):305–313, 1986.

[82] W. Spendley, G. R. Hext, and F. R. Himsworth. Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation. *Technometrics*, 4(4):441–461, November 1962.

[83] George C. Stockman. Waveform Parsing Systems. In P. R. Krishnaiah and L. N. Kanal, editors, *Classification, Pattern Recognition, and Reduction of Dimensionality*, volume 2 of *Handbook of Statistics*, chapter 24, pages 527–548. North-Holland, Amsterdam, 1982.

[84] George C. Stockman and Laveen N. Kanal. Problem Reduction Representation for the Linguistic Analysis of Waveforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(3):287–298, May 1983.

[85] Panagiotis Trahanias and Emmanuel Skordalakis. Syntactic Pattern Recognition of the ECG. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):648–657, July 1990.

[86] Amos Tversky. Features of Similarity. *Psychological Review*, 84(4):327–352, July 1977.

[87] Barbara Tversky and Kathleen Hemenway. Objects, Parts, and Categories. *Journal of Experimental Psychology: General*, 113(2):169–193, June 1984.

[88] Jan H. van Bemmel. Recognition of Electrocardiographic Patterns. In P. R. Krishnaiah and L. N. Kanal, editors, *Classification, Pattern Recognition, and Reduction of Dimensionality*, volume 2 of *Handbook of Statistics*, chapter 23, pages 501–526. North-Holland, Amsterdam, 1982.

[89] Frederick H. Walters, Lloyd R. Parker, Jr., Stephen L. Morgan, and Stanley N. Deming. *Sequential Simplex Optimization*. CRC Press, Boca Raton, Florida, 1991.

[90] M. H. Wright. Direct Search Methods: Once Scorned, Now Respectable. In D. F. Griffiths and G. A. Watson, editors, *Numerical Analysis 1995*, pages 191–208. Addison Wesley Longman, Essex, England, 1996.

[91] Tzay Y. Young and King-Sun Fu, editors. *Handbook of Pattern Recognition and Image Processing*. Academic Press, Orlando, Florida, 1986.

[92] Walter Zucchini. An Introduction to Model Selection. *Journal of Mathematical Psychology*, 44(1):41–61, March 2000.