University of California
Los Angeles

Medical Signal Searching

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Jonathan Scott Woodbridge

2012

Abstract of the Dissertation

Medical Signal Searching

by

Jonathan Scott Woodbridge

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2012

Professor Majid Sarrafzadeh, Chair

This dissertation presents work in medical signal searching. Signal searching (or subsequence matching) is the process of finding similar segments within a database of time series signals. Medical signal searching is extremely challenging given the large, high entropy, multidimensional datasets. However, medical time series signals tend to be cyclical and repetitive and contain a large amount of redundancy. In addition, similar segments in a medical time series signals tend to form tight clusters where the Euclidean distance between segments within the same cluster follow a Gaussian distribution. These properties can be leveraged in many ways to significantly improve search performance in terms of computation, memory, precision, and recall.

This dissertation addresses three main aspects of medical signal searching: extraction of query segments; improvement in the quality of search results; and optimization of indexing structures through the removal of redundancy. These contributions heavily leverage the properties of medical time series signals. Each contribution of this dissertation has been thoroughly tested on a wide variety of medical time series signals including data collected through a clinical study. In addition, a complete system for searching electrocardiogram (ECG) is presented.

The dissertation of Jonathan Scott Woodbridge is approved.

Junghoo Cho

Stott Parker

Alex Bui

Majid Sarrafzadeh, Committee Chair

University of California, Los Angeles

2012

To my wife and daughter . . .

who made this work possible

through their love and support.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to acknowledge Professor Majid Sarrafzadeh for his excellent supervision throughout my graduate career. I would also like to acknowledge Professor Alex Bui for his invaluable input on our work in Medical Signal Searching. Finally, I would like to acknowledge Bobak Mortazavi and Mars Lan for their contributions to this work.

Vita

2004	B.S. (Computer Science), Georgia Institute of Technology.
2004–2008	Senior Engineer, Qualcomm Inc., San Diego, California.
2010	M.S. (Computer Science), UCLA, Los Angeles, California.

Publications

J. Woodbridge, B. Mortazavi, A. Bui, and M. Sarrafzadeh. A Monte Carlo Approach to Biomedical Timeseries Search. IEEE International Conference on Bioinformatics and Biomedicine (IEEE BIBM), October 2012.

J. Woodbridge, B. Mortazavi, A. Bui, and M. Sarrafzadeh. High Performance Biomedical Time Series Indexes Using Salient Segmentation. International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), August 2012.

M. Suh, T. Moin, J. Woodbridge, H. Ghasemsadeh, S. Ahmadi, A. Bui, M. Sarrafzadeh. Dynamic Self-adaptive Remote Health Monitoring System for Diabetics. International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), August 2012.

P. Gad, J. Woodbridge, I. Lavrov, H. Zhong, R. Roy, M. Sarrafzadeh, and V. R. Edgerton Forelimb EMG-Based Trigger to Control an Electronic Spinal Bridge to Enable Hindlimb

Stepping After Complete Spinal Cord Lesion in Rats Journal of NeuroEngineering and Rehabilitation, June 2012, 9:38

M. Suh, J. Woodbridge, A. Bui, L. S. Evangelista, M. Sarrafzadeh. Missing Data Imputation for Remote CHF Patient Monitoring. International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), August 2011.

J.Woodbridge, M. Lan, A. Bui, M. Sarrafzadeh. Salient Segmentation of Medical Time Series Signals. IEEE Conference on Healthcare Informatics, Imaging and Systems Biology (HISB). San Jose, California, July 2011.

M.Suh, C.Chen, J. Woodbridge, M. K. Tu, J. Kim, L. S. Evangelista, M. Sarrafzadeh. A Remote Patient Monitoring System for Congestive Heart Failure. Journal of Medical Systems (JOMS), June 2011.

M. Suh, A. Nahapetian, J. Woodbridge, M. Rofouei, M. Sarrafzadeh. Machine Learning-Based Adaptive Wireless Interval Training Guidance System. Mobile networks and applications (MONET), 2011.

H. Noshadi, S. Ahmadian, H. Hagopian, J. Woodbridge, N. Amini, F. Dabiri, M. Sarrafzadeh, N. Terrafranca. Hermes: Mobile Balance and Instability Assessment System. BioSignals, January 2010.

J. Woodbridge, A. Bui, and M. Sarrafzadeh. Linear Frequency Estimation Technique for Feducing Frequency Based Signals. The 1st Workshop on Light-Weight Signal Processing for Computability Intensive BSN Applications. June 2010, Samos Greece. 54:1-54:4.

CHAPTER 1

Introduction

The advent of remote and wearable medical sensing has created a dire need for efficient medical time series databases. Wearable medical sensing devices provide continuous patient monitoring by various types of sensors, such as accelerometers for activity monitoring; electrocardiogram (ECG) for heart monitoring; and pulse oximeters for blood oxygen saturation monitoring. These devices have the potential to create massive amounts of high dimensional data. For example, there are currently over 3 million people worldwide implanted with a pacemaker [WE02]. If these systems had the ability to gather, store, and transmit an ECG signal, we could expect to receive over 560 terabytes of data per day, just from individuals with pacemakers (assuming a three channel ECG, sampling rate of 360 Hz, and 2 byte ADC). Therefore, medical time series databases must be able to store, index, and mine large high dimensional datasets to enable both data mining as well as analysis of a patient's health.

Signal searching (or subsequence matching) is synonymous with the R-nearest neighbor (R-NN) problem: given a query object u and a collection C of high dimensional objects, find all objects $c \in C$ such that $||u - c||_p < R$. Sequential scan is a naïve solution to R-NN and consists of a computational and memory complexity of $O(n)$. However, medical time series databases have the potential to be extremely large, meaning a sequential scan is not an effective solution to R-NN. Many solutions to R-NN have been proposed in previous works. This dissertation presents work to enhance R-NN methods by improving memory and computational restraints as well as improve both precision and recall.

Case-based reasoning systems are one appealing application of R-NN in biomedical time

series data. In terms of classification, classes are composed of multiple (often Gaussian distributed) sub-groupings [BFG99]. R-NN can be used to find members that belong to a similar sub-grouping as a query segment. Discovery of such sub-grouping can be used for classification as well as discovery of correlations between corresponding patients from within a sub-grouping. These correlations are useful for both diagnosis and tailored treatments. However, as this dissertation shows, previous solutions to R-NN have several weaknesses when searching sufficiently large databases.

Three main improvements to R-NN for medical time series are proposed:

1. Extraction of query segments;
2. Improvement in the quality of search results; and
3. Optimization of index structures.

Extraction of query segments is accomplished using Salient Segmentation. Salient Segmentation is a probabilistic method for locating the most interesting segments from a time series signal. The number of query segments is significantly reduced by ignoring all non-interesting segments. The proposed method reduces redundancy with minimal loss of information. In addition, Salient Segmentation can be used to limit the size of an index. Search performance (in terms of memory and computation) can be improved by only including salient segments in the index. Such a solution yields high performance with limited degradation in the quality of search results.

The quality of search results is improved through a randomized technique. This technique performs a number of randomizations on a query segment, thereby creating a set of query segments. Each of these query segments is used in a nearest neighbor search. These randomizations are done in an intelligent fashion to increase recall with minimal degradation of precision. In addition, this technique adds little in terms of memory and computation.

Indexing structures are optimized using an aggregation framework. This framework aggregates similar segments into a common grouping. Each grouping is assigned a representative segment, and only these representative segments are added to the index (i.e., the search space is composed of only one representative segment from each grouping). As shown later, the number of groupings grows sub-linearly with the size of the signal, yielding a scalable solution to medical time series databases.

This dissertation analyzes the proposed techniques on both the time domain and the feature space. An in depth analysis of the feature space is given for ECG signals. The features space is shown to improve the quality of results (in terms of precision and recall) as well alleviate compatibility issues across different datasets (e.g., sampling rates, noise, hardware difference, etc.).

Figure 1.1 displays the overall flow diagram of the proposed signal search engine. The upper portion of the figure displays the process of populating the database. The database first segments and aggregates a signal when it is received by the system. Segmentation may include Salient Segmentation, sliding window segmentation, or domain specific segmentation. Next, each segment is inserted into a time series index along with any annotations.

The lower portion of the figure displays the process of search. A time series is first segmented by the system. These time series are generally short regions, deemed regions of interest (ROI), selected by the user from a larger time series signal. Next, each extracted segment is randomized to create a set of query segments. These sets of query segments are then used to search the database for nearest neighbors. Annotations and meta data are then returned by the database for analysis.

Methods proposed by this dissertation are tested on publicly available real-world biomedical datasets. In addition, a small clinical trial is performed using an at-home ECG monitoring system on subjects with congestive heart failure (CHF). Data collected by the clinical study was used as query segments into the overall R-NN system proposed by this dissertation to

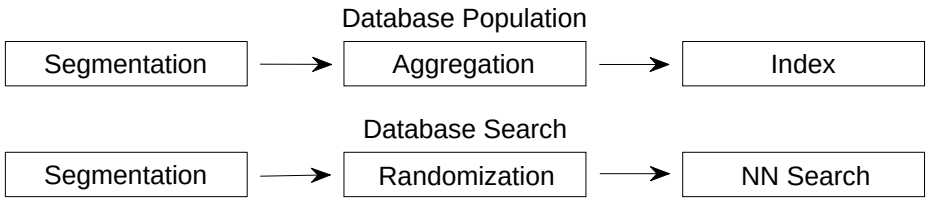

Figure 1.1: A flow diagram of the proposed system. The upper diagram shows the flow of database population. The lower diagram displays the flow of search.

show feasibility of case-based reasoning systems for wearable sensing. The results of this study showed a high correlation between search results (and their respective annotations) to annotations created by medical professionals. These findings suggest that an R-NN system with more data and improved annotations could be used to mine and derive correlations to help tune diagnosis and treatments in the future.

The remainder of this dissertation is organized as follows. Related work is discussed in Chapter 2. Salient Segmentation is introduced in Chapter 3. Aggregated indexing is introduced in Chapter 4 as a means to reduce index sizes. Chapter 5 presents a randomized and efficient technique for improving recall of search results. Chapter 6 provides an in depth analysis of applying the techniques proposed by this dissertation to the feature space. Conclusions and future work is discussed in Chapter 8.

CHAPTER 2

Background

2.1 Definitions

- **Time series signal:** A time series signal $T = \{t_1, t_2, ..., t_n\}$ is composed of an ordered set of n points in the time domain.
- **Segment (subsequence):** A segment (or subsequence) $s_i = \{t_i, t_{i+1}, ..., t_{i+1-m}\}$ is a contiguous set of points in T of length m.
- **Sliding window:** Sliding window segmentation consists of a fixed window of size m that is slid along the time dimension. Each slide is of $D \geq 1$ time points results in one segment being extracted from the time series signal.
- **Match:** Two segments, u and v are considered matching if $||u - v||_p < R$, where R is a predefined radius.

2.2 Distance Measures

This dissertation assumes the L_2 norm as the primary distance measure. However, much of this work is generic and can be applied to any L_p norm. Many authors have suggested the use of elastic measures such as Dynamic Time Warping (DTW) claiming this to be superior to Euclidean distance [SMC][YJF98][KPC04]. However, this superiority has been questioned. Authors in [RK04] show that DTW's ability to compare segments of different

lengths have little to no statistical significance. In addition, authors in [SK09] note that DTW's advantages are generally only in small datasets. As datasets grow, there is a higher probability of finding a match and less of a need to warp. Results were shown heuristically and demonstrated how error rates of both DTW and Euclidean distance converge as the size of datasets grow.

This work uses the L_p norm for three reasons.

1. Dynamic Time Warping has a quadratic computational complexity unlike the linear complexity of L_p norms;

2. Locality Sensitive Hashing (LSH) has a provable sub-linear search complexity unlike indexing methods used for DTW; and

3. Later sections propose the use of normalized domain specific feature vectors when performing nearest neighbor searches. DTW has little meaning in such cases as feature vectors are not based on the time domain.

Finding an optimal distance function becomes more difficult with an increasing number of dimensions as distance norms have the property to converge with an increasing number of dimensions [BGR99]. It has been shown that this convergence is true for any p-norm and is more related to the intrinsic dimension than the embedding dimension [FWV07]. However, [BFG99] suggests that L_p norms are effective when considering small neighborhoods. [BFG99] presented a method to index high dimensional objects under the assumption that, in terms of classification, classes are composed of multiple Gaussian distributed sub-groupings. These sub-groupings are approximated using an optimized Expectation Maximization (EM) algorithm [BFR98]. [BFG99] not only showed nearest neighbor searches to be meaningful in high dimensional space, but that the statistical structure of a dataset can be leveraged to construct a multi-dimensional indexing scheme.

The usefulness of L_p norms was also shown experimentally in [HKK10]. The Euclidean distance between objects in many datasets also has been shown to follow a Gaussian distribution. In addition, the Euclidean distance between objects shows moderate separation between classes. The authors proposed a new distance measure based on shared nearest neighbors. Under this measure of distance, objects with a large number of shared nearest neighbors (in terms of Euclidean distance) are considered more similar to objects that shared few or no nearest neighbors. While better separation was shown with a nearest neighbor comparison, calculating the sets of nearest neighbors has a computational complexity of $O(n^2)$; far too high for large databases.

Authors in [HAK00] have suggested that L_p norms with $p = 1$ or $p = 2$ are the only useful integer norms. They note that the convergence of the distance between nearest neighbors and farthest neighbors from [BGR99] does not necessarily converge to 0 when $p < 3$. Authors show that L_1 norms are the only norm to grow with increasing number of dimensions while L_2 converge to a constant. They also prove that any norm with $p \geq 3$ will converge to 0.

Authors in [AHK01] explore fractional p values ($0 < p < 1$). Authors show that smaller values of p exhibit better discrimination of high dimensional objects. However, their findings rely on the assumption that datasets followed a single distribution with a high concentration of groupings. As noted in [HKK10][BFG99], many datasets do not follow a single distribution.

Assuming high discriminatory power using L_p norms, a match can be defined as $||u - v||_p < R$ where R defines a tight radius (or small neighborhood). Aggregating objects with such a distance measure will result in a large number of precise groupings. As shown later, the number of groupings grows logarithmically with increasing n for many real-world biomedical datasets. Therefore, the search space (or mining space) can be significantly reduced through such an aggregation.

2.3 Segmentation

Generic segmentation exists for many medical domains. These segmentation techniques rely on a known model of the data. For example, ECG is segmented based on the model consisting of a P wave, a QRS complex, a T wave, and a U wave [PT85][KHO02]. However, not all domains have a well known model. One example is activity monitoring using wearable sensors (such as accelerometers, gyroscopes, etc.). The best known techniques for activity monitoring are based on time invariant features and span over several cycles of a repetitive activity (such as walking or running) [BI04][RDM05]. However, these techniques can only distinguish between a limited number of high level activities, such as the difference between walking, running, and climbing stairs. More fine grained detection, such as abnormal gait, requires time dependent features [NAH10]. However, these features are often only meaningful in a constrained environment.

More generic techniques are needed to account for unknown (or partially known) models. Motif discovery is one technique for segmenting time series signals and finding patterns that are repeated throughout the time series. This thesis presents a probabilistic technique in Chapter 3 that only extracts segments that are determined to be locally improbable.

2.3.1 Motif Discovery

Motif discovery finds sets of similar segments in a time series signal [LP02]. Each segment in a time series signal is compared to all other segments. Matches are defined as two segments with a distance less than or equal to a given threshold, thr. The top motif is the segment with the most matches. The k-motif algorithm returns the k segments with the most matches. A simple quadratic algorithm to find the top motif is given in Figure 2.1.

Motif discovery algorithms have three weaknesses. First, the best computational bound for exact motif discovery is quadratic. Second, motif discovery algorithms suffer from redun-

dancy in their output. Third, motif discovery is designed to find the most prevalent patterns and often ignores anomalies.

2.3.1.1 Complexity

Medical time series have the potential to be extremely large, and therefore, can not be segmented in practice by any algorithm with a computational complexity bound that is greater than linear. The classical version of motif discovery calculates all pairwise distances, resulting in a complexity of $O(n^2)$, where n is the length of a time series. Attempts to reduce the average case complexity of exact discovery have been attempted in [MKZ09]. This algorithm uses the triangle inequality to prune the number of distance computations. However, this work is an optimization of classical motif discovery and still holds a complexity of $O(n^2)$.

Approximations to exact motif discovery have also been proposed. Authors in [CKL03] use a series of random projections to create hashes for each segment in a time series. Hashes with the most collisions result in the best motif. This algorithm solves the k-motif algorithm by returning the k hashes with the most collisions. The pairwise distance is calculated between each of these k motifs and all n segments to extract similar segments. The complexity of this algorithm, is therefore, dominated by $O(kn)$.

Authors in [MIE07] proposes a density estimation framework for motif discovery. The algorithm calculates the k-nearest neighbors for each segment. These sets of nearest neighbors are then used to estimate the density of matches. These densities are later used by a Hidden Markov Model (HMM) to locate the best motifs. This algorithm relies on the assumption that the k-nearest neighbors can be calculated in $O(n \log n)$ time, citing the methods defined in [KR05], [Liu06], [Den05]. [KR05] creates an spatial indexing structure based on the dynamic time warping distance between segments to reduce the time complexity. [Den05] presents a kernel density-based clustering algorithm using an underlying kd-tree

(a spatial index) for finding nearest neighbors. However, it has been shown both theoretically and experimentally that spatial indexes perform worse than linear scan with a large enough number of dimensions [WSB98] (resulting in at least a $O(n^2)$ upper bound). [Liu06] uses modifications of the metric-tree [Uhl91] [Omo91] to speed up the k-nearest neighbor search, but admits that the proposed methods approach a $O(n^2)$ bound with a large number of dimensions.

2.3.1.2 Redundancy

Redundancy is the second weakness of motif discovery. Motif results are dominated by similar motifs with different translations. Such redundancy is exacerbated by cyclical and repetitive signals such as medical time series. Redundancy is mitigated by the removal of trivial matches. Two segments $\hat{i} = [i, i + m']$ and $\hat{j} = [j, j + m']$ are trivial matches if $dist(\hat{i}, \hat{j}) \leq thr$ and there is no segment $\hat{i'} = [i', i' + m']$ where $i \leq i' \leq j$ and $dist(\hat{i}, \hat{i'}) > thr$ [LP02].

Removing such trivial matches is too passive of a filter as trivial matches generally result in the removal of very few segments. A more aggressive approach is to set a constant w such that any segment $s_{\hat{i'}}$ is trivial with respect to $s_{\hat{i}}$ if $|\hat{i} - \hat{i'}| \leq w$ [MKZ09]. But a constant w is too rigid of a measure; and a small w does very little to remove redundancy while large windows remove motifs that may not actually be trivial. For example, assume an ECG segment that consists of three heart beats (approximately three seconds in duration). A w of 0.1 seconds would do little to removes redundancy as such a small shift in the center of a segment would yields little difference in the data contained by the signal. On the other hand, if w was set to three seconds, both the first and last heart beat of our example would never be represented in a segments as the center beat. Dropping such segments increase the difficulty in diagnosing certain abnormal heart beats that require context from previous and subsequent beats (such as premature ventricular contractions).

2.3.1.3 Anomalies

Thirdly, motif discovery is designed to return the most prevalent patterns. In medical time series signals, anomalies are extremely important. For instance, the discovery of an abnormal heart beat (such as an arrhythmia) is far more important to a patient's health than the discovery of a normal heart. In addition, a normal heartbeat may appear in many forms. Wearable heart monitors allow patients to gather continuous ECG data over several days to weeks. During this time, patients will participate in different activities resulting in variable heart rates and noise. In this case, an extremely large k would be needed to segment both normal and anomaly heart beats. Depending on the parametrization of the redundancy filter, no k (no matter how large), will retrieve all anomalies.

2.4 Subsequence matching

Subsequence matching, or signal searching, is synonymous with the R-NN problem in high dimensional space. Nearest neighbors are defined as all objects that fall within distance R of a query object. For the purpose of this dissertation, distances are defined by the L_2 norm (Euclidean distance) and objects are represented as points in $\Re^d$. Sequential search is a naïve approach to solving the R-NN problem. This, of course, has an $O(n)$ computational complexity that is far too slow for large databases. Hence, an optimal solution would guarantee sub-linear complexity. In addition, many studies have shown that conventional indexing methods fail with an increasing number of dimensions [BKS90] [BEK98] [KS97] [SK98].

Authors in [FRM94] presented a framework for subsequence matching using spatial access methods (such as R^*-trees, iSAX [SK09], etc.) to index dimensionally reduced objects. Reduction algorithms for this framework must satisfy the property of minimum bounds. The minimum bounds property guarantees that the distance between two reduced objects is less than or equal to the distance between the corresponding original (non-reduced) objects:

procedure $FindTopMotif(A, i, j)$

1: $topMotifCount = 0$
2: $topMotifIndex = 0$
3:
4: **for** $i = 1 \rightarrow length(S) - m + 1$ **do**
5: $count = 0$
6:
7: **for** $i = 1 \rightarrow length(S) - m + 1$ **do**
8: **if** $thr \geq dist(S(i : i + m'), S(j : j + m'))$ **then**
9: $count++$
10: **end if**
11: **end for**
12:
13: **if** $count \geq topMotifCount$ **then**
14: $topMotifCount = count$
15: $topMotifIndex = i$
16: **end if**
17: **end for**
18:
19: Return $topMotifIndex$

Figure 2.1: Top Motif Algorithm

$$dist(\hat{u}, \hat{v}) \leq dist(u, v) \tag{2.1}$$

Where $\hat{u}$ and $\hat{v}$ are the dimensionally reduced counterparts of vectors $u, v \in \Re^d$. The minimum bounds property guarantees that there are no false dismissals. Therefore, the set of all search results is a superset of all true matches. Reduction algorithms satisfying the property of minimum bounds include Piecewise Aggregate Approximation (PAA) [KCP01], Discrete Fourier Transform (DFT) [FRM94], and Chebyshev polynomials [CN04].

The framework proposed in [FRM94] has two main drawbacks. First, query results of dimensionally reduced segments are a superset of the true matches. Therefore, false positives must be filtered from the result set. Filtering is accomplished by finding the true distances between the original non-reduced query segment and its matches. Filtering can be extremely computationally expensive as no theoretical upper bounds are placed on the number of false positives. Second, spatial indexes have been shown both theoretically and experimentally to perform worse than sequential search for data with as few as 10 dimensions [WSB98].

Locality Sensitive Hashing (LSH) [GIM99] was introduced as an alternative to spatial indexing schemes. Unlike the aforementioned spatial indexing schemes, LSH has a proven sub-linear computational complexity. LSH is based on a family of hashing functions that are (r_1, r_2, p_1, p_2)-sensitive, meaning that for any $v, q \in S$:

- if $v \in B(q, r_1)$ then $\Pr_H[h(q) = h(v)] \geq p_1$
- if $v \notin B(q, r_2)$ then $\Pr_H[h(q) = h(v)] \leq p_2$

Where $p_1 > p_2$ and $r_2 > r_1$. The gap between p_1 and p_2 is increased by combining several functions from the same (r_1, r_2, p_1, p_2)-sensitive family.

The experimental implementations in this dissertation uses the LSH scheme based on p-stable distributions defined in [DII04]. The authors propose the following hash function:

$$h_{a,b}(v) = \lfloor \frac{a \cdot v + b}{r} \rfloor \tag{2.2}$$

Where a is a randomized vector following a Gaussian distribution, b is a uniformly randomized number, and r is a predefined constant. Using the properties of the p-stable distribution, the authors show that the probability of collision is calculated as:

$$p(c) = \int_0^r \frac{1}{c} f_p(\frac{t}{c})(1 - \frac{t}{r})\mathrm{d}x \tag{2.3}$$

With c being the distance between two vectors. As can be seen by Equation 2.3, the probability of collision decreases monotonically as c increases.

The aforementioned indexing schemes all suffer from homogeneity. These algorithms are solutions to the R-NN problem and, therefore, produce results that are extremely similar to the query segment. Raising the value of R is one approach to increasing heterogeneity. However, this approach would add instability to the nearest neighbor problem, rendering the solution meaningless [BGR99]. Therefore, the expansion of search results must be bounded to ensure stability.

To note, several works have made improvements to the original LSH algorithm such as *locality sensitive B-tree* (LSB-tree) [TYS09], Multi-Probe LSH [LJW07], and entropy based nearest neighbor search [Pan06]. This dissertation utilizes LSH due to its maturity. However, the proposed approach could be easily adapted to any other method that provides both bounds on the runtime complexity and quality of results.

2.5 Index Reductions

One large issue in time series databases is the sheer size of the index, adding tremendous memory and computational constraints on the system. Several works have explored methods

to reduce the overall index size in textual databases [HLY07][ZS07][BEF06]. In general, these methods break down documents into several fragments. Fragments are indexed in lieu of the entire documents. Identical fragments that occur across multiple documents (or multiple versions of a document) are inserted into an index only once thereby minimizing the size of the index.

For example, [ZS07] uses winnowing [SWA03] to break down documents into 10 to 25 fragments. Fragments (and their respective alignments) are chosen to maximize redundancy, thereby reducing the overall size of the index. Performance of the winnowing algorithm is optimized using hashing such that the comparison of two fragments is based on their respective hashing. Index sizes are reduced by as much as 60% when removing redundancy. However, textual methods such as these rely on exact matching to reduce redundancy. The probability of seeing exact matches in medical time series is extremely low due to environmental (e.g., noise, sensor displacement/placement, etc.) and physiological (e.g., differences in heart rates, weight, age, etc.) factors.

As stated earlier, methods following the model introduced in [FRM94] reduce index sizes by applying generic reductions that follow the property of minimum bounds. While reduction algorithms may significantly reduce the size of the index, they also reduce the overall discriminatory power of distance computations.

PAA and DFT are perhaps the simplest and most studied of these reduction algorithms. In PAA, a reduction of segment X of size m is accomplished by dividing X into M equal size bins. The reduction of X is represented by the average of each of these bins. More formally [KCP01]:

$$\hat{x_i} = \frac{M}{m} \sum_{j=\frac{m}{M}(i-1)+1}^{\frac{m}{M}i} x_j. \tag{2.4}$$

DFT makes the observation that the frequency components of many time series signals

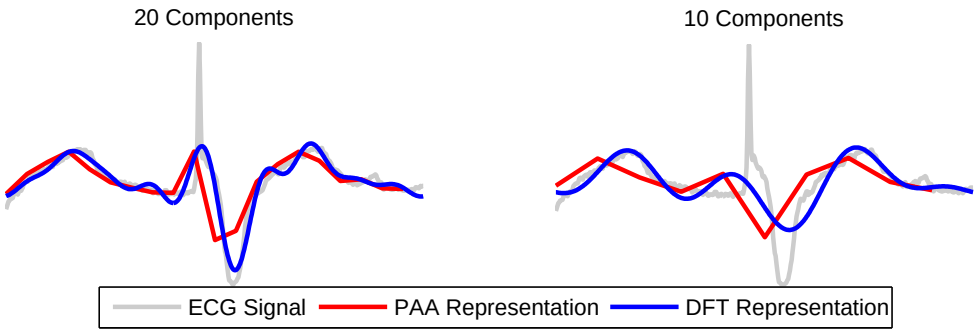

Figure 2.2: The PAA and DFT representations of an ECG signal composed of 500 data points and reduced to 20 and 10 dimensions. Note that a large amount of detail is lost from the segment.

have much of their power in lower frequencies. Hence, a signal can be significantly reduced by excluding all high frequencies with little loss of information.

An example of both reductions is shown in Figure 2.2. A large amount of detail is lost in the reduction, decreasing discriminatory power. This is especially true with biomedical signals that consist of similar repetitions. In addition, the reductions are only linear with respect to the number of dimensions. As the number of objects heavily dominates the number of dimensions, these reductions are not a scalable solution to large index sizes.

2.6 Extracting Domain Specific Features

Much of the previous discussion is in respect to signals in the time domain. However, this work is also applicable to segments converted to the feature domain. Creating feature sets is often more of an art than a science and consists a combination of domain knowledge and statistical properties. ECG is one of the most studied medical time series with extensive studies on domain specifics. Several authors have proposed the usefulness of statistical

properties for classification of ECG signals. Methods include the Hermite Basis Function [LPB00][PCL08] and higher order statistics [PCL08][OL01]. Other works suggest the use of more domain specific features such as properties of various intervals in a heart beat [DOR04][CGK06][MB08]. Authors in [DLF11] provided a full analysis of various types of features and showed that the strongest features are generally related to properties of intervals in ECG signals.

Similar domain specific work has been run on activity recognition using accelerometers and gyroscopes [BI04][RDM05] and properties of GAIT [NAH10][BMP03]. Features are vastly different across domains and require domain experts (such as medical doctors) to derive statistics useful for classification and search.

CHAPTER 3

Salient Segmentation

Salient Segmentation is a preprocessing segmentation technique that extracts the most interesting (or salient) segments from a time series signal [WLS11]. Interestingness, or saliency, is defined as the least probable segments within a region of interest. Time series signals are modeled as Markov chains to calculate the probability of each segment's occurrence ($P(w_i)$). Specifically, saliency is defined as the negative log of each window's probability ($-log(P(w_i))$). Windows with a higher saliency than neighboring windows (local maximums) are considered salient and are extracted. All other high probability segments are ignored. Hence, a user can select large regions of interest and allow Salient Segmentation to extract the most interesting segments from these regions. These segments can be subsequently fed to a signal search database to find other similar segments.

Limiting searches to only salient segments improves data mining and search performance. Previous works have used Motif Discovery as a segmentation technique [KL05]. However, Motif Discovery is highly prone to redundancy. Figure 3.1 presents the top five ECG motifs returned by the motif discovery algorithm in [MKZ09]. Notice that there are only two unique patterns and three redundant patterns. Returning additional motifs only yields a larger number of redundant segments. Medical time series signals are particularly prone to motif redundancy due to their cyclical nature. Hence, finding the true makeup of a medical time series signal using traditional motif discovery algorithms is difficult, at best. Salient segmentation reduces the amount of redundancy in characteristic motifs by more than 85%, yielding a more tangible representation of a medical time series signal. As such, signal search

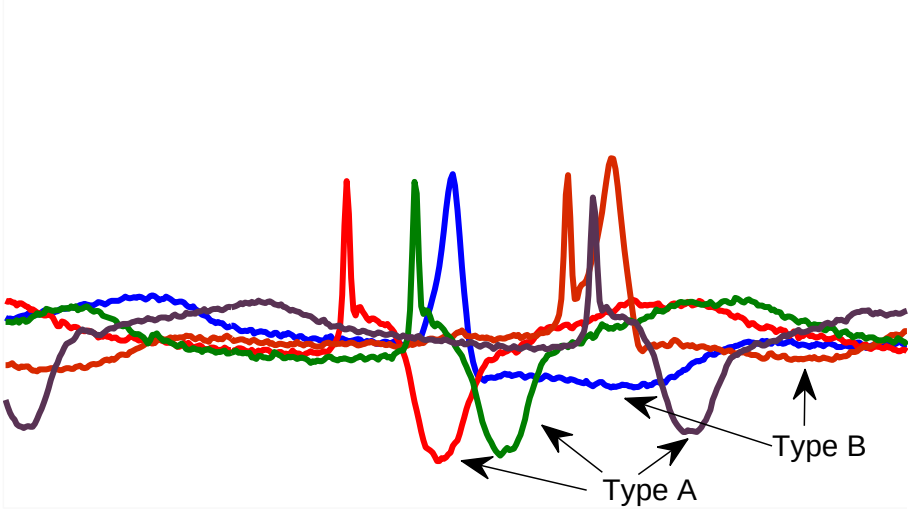

Figure 3.1: Top five motifs from a 2 minute ECG recording returned by the motif discovery algorithm in [MKZ09]. The results consist of only two unique patterns.

is improved by reducing queries to salient segments. A search of the index is initiated by selecting a region of interest (ROI) in a time series signal. An ROI is often very large and contains many segments. However, searching the index for all segments within an ROI can inundate the user with largely redundant results. In addition, an ROI may contain many common and uninteresting segments (such as an accelerometer at rest). Returning matches to such non-salient segments will yield little to no benefit to a user. In preliminary testing, the presented salient segmentation framework decreases the overall number of segments in a time series signal (and ROI) by more than 98%. Limiting searches to only salient segments within an ROI not only reduces the number of returned results, but also improves the quality of those results by ignoring redundant and uninteresting segments (thereby increasing relevancy of results).

Salient segmentation offers the following three properties:

1. All salient patterns are segmented;

2. All salient patterns are segmented consistently (i.e., alignment); and

3. Near linear algorithmic complexity

The first property ensures that all segments that are similar to a salient segment are also labeled salient. The second property ensures alignment and therefore removes redundancy. Salient patterns should only be indexed once unless a translation of that pattern adds significant information. The third property is required for large datasets: a high complexity algorithm would not only drain resources, but introduce a large delay from the time data was received to the time it becomes available for information retrieval tasks. Salient segmentation offers a near linear time algorithm, allowing efficient processing of a time series signal.

3.1 Method

A time series signal is modeled as a Markov chain such that the following property holds.

$$\Pr(T_{n+1}|T_1 = t_1, T_2 = t_2, ..., T_n = t_n) = \Pr(T_{n+1}|T_n = t_n) \tag{3.1}$$

The transition distribution for a time series signal is calculated by taking a histogram of all possible transitions for each state. However, calculating the transition distribution for each state individually is memory inefficient and requires an extremely large signal to ensure an accurate distribution. While a time series database may be significantly large, individual time series may not. Fortunately, the transition distributions for proximal states are quite similar for most time series signals. Using this observation, close states are grouped together such that each grouping has an equal number of samples to estimate their respective transition distributions. Grouping is accomplished by first estimating the distribution of states. Next, the state space is divided into groups such that each group has an equal probability (i.e.,

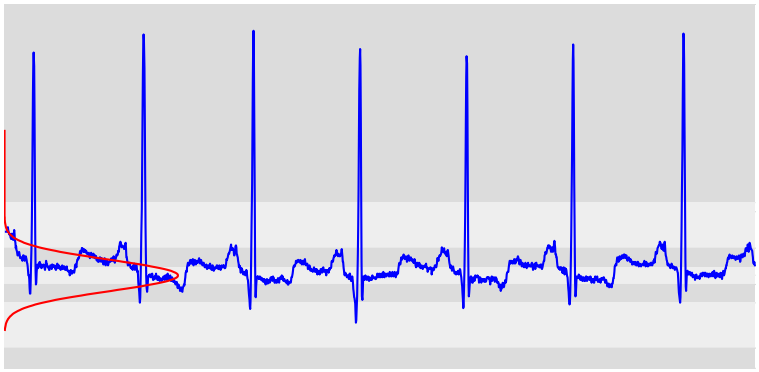

Figure 3.2: The state probability ditribution function is displayed to the left. Each of the seven shaded regions represents a distinct group where each group has an equal probability of occurrence. The transition distribution function is calculated for each distinct group resulting in seven difference transition distributions.

the area under the probability curve for each group is equal). Figure 3.2 gives an example grouping with an ECG signal. Each shaded region represents a distinct grouping (or bin) and has an equal probability of occurrence using the distribution displayed to the left.

Each state T_i uses the transition distribution computed for its corresponding bin Bin_{T_i} and is defined as:

$$P(T_i|T_{i-1} = t_{i-1}) = P(T_i|Bin_{T_{i-1}} = Bin_{t_{i-1}}) \tag{3.2}$$

Note that grouped states are only used for the prior state and not for the transition states. The transition states use the entire range to improve sensitivity.

Each point t_i in a time series corresponds to a window w_i where w_i includes the points $[t_{i-\lfloor m/2 \rfloor}, t_{i+\lfloor m/2 \rfloor}]$. The probability for each window w_i is calculated as:

$$\Pr(w_i) = \prod_{j=i-\lfloor m/2 \rfloor}^{i+\lfloor m/2 \rfloor} \Pr(T_j|Bin_{T_{j-1}} = Bin_{t_{j-1}}) \tag{3.3}$$

With each points corresponding saliency as:

$$Saliency_i = -\log \Pr(w_i) \tag{3.4}$$

A time series saliency function (TSF) is constructed by concatenating each successive point's saliency such that:

$$F_{saliency} = \{Saliency_1, Saliency_2, ..., Saliency_{n-m+1}\} \tag{3.5}$$

Each local maximum i in $F_{saliency}$ is considered salient and its corresponding segment ($[t_{i-\lfloor m'/2 \rfloor}, t_{i+\lfloor m'/2 \rfloor}]$) is inserted into the index. However, saliency functions often contain a significant amount of noise, thus resulting in over segmentation (too many maximums).

A set of linear approximations calculated by the Ramer-Douglass-Peucker (RDP) algorithm [Ram72] [DP73] is used to filter the TSF. The RDP algorithm begins with a linear approximation with endpoints at the first point p_1 and the last point p_n of the TSF. Next, the distance between the linear approximation and each point between the first and last point is calculated. If the point with the largest distance p_i is above a given threshold, thr, the signal is estimated by two linear approximations: $p_1 \rightarrow p_i$ and $p_i \rightarrow p_n$. The algorithm is repeated on both segments and continues until no point is more than thr from its linear approximation. Here, the value of thr is a function of the standard deviation of the estimated TSF:

$$thr = \alpha\sigma \tag{3.6}$$

TSF's with higher standard deviations generally have both a larger amount of noise and disparity between peaks and valleys. Hence, TSF's with a large standard deviation require a more aggressive filter. A larger value of α will result in fewer maximas while a smaller value will result in more maximas. This chapter assumes $\alpha = 1$.

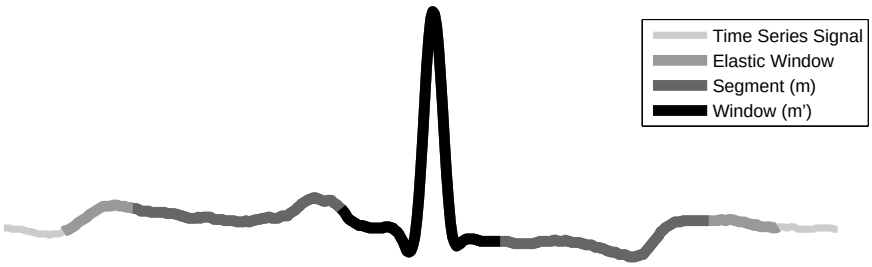

Figure 3.3: Displays the relation of a window, segment, and elastic window.

The RDP approximation of the TSF results in a slight misalignment between segments. This means that the salient points' locations calculated after the RDP approximation has a slight variation from the true locations. Therefore, an additional δ points are added before and after a segment s_i to create an elastic window defined by the range $[i - \lfloor m'/2 \rfloor - \delta, i + \lfloor m'/2 \rfloor + \delta$. When the segment is searched, a window of size m' will compare all windows located between $i - \lfloor m'/2 \rfloor - \delta, i + \lfloor m'/2 \rfloor + \delta$. As $\delta << n$, the elastic window results in only a small decrease in performance.

The relationship between the elastic window, segment and window is shown in Figure 3.3. The salient segmentation algorithm is summarized in Figure 3.4.

3.1.1 Complexity

Quantization of the range and calculation of the transition distribution is done in linear time. The TSF is computed in time $O(mn)$ in Equation 3.5. However, the TSF can be calculated as a sliding window where each slide results in one division and one multiplication yielding an $O(n)$ runtime. The RDP filtering technique has an average runtime of $O(n \log n)$ and has been improved to have an upper bound of $O(n \log n)$ in [HS94]. Finding maximas and extracting segments can both be done in linear time yielding an overall upper bound of

procedure $SalienctSegment(inputSignal, bins, m, m', \alpha, \delta)$

1: $range = QuantizeRange(inputSignal, bins)$
2:
3: $pdf = ComputeTransitionDistribution(inputSignal, range)$
4:
5: $tdf = ComputeTSF(inputSignal, pdf, m)$
6:
7: $filteredTDF = RDP(tdf, \alpha)$
8:
9: $salientPoints = FindMaximums(filteredTDF)$
10:
11: $segments = ExtractSegments(salientPoints, \delta, m')$
12:
13: Return $segments$

Figure 3.4: Salient Segmentation Algorithm

$O(n \log n)$.

3.2 Experimental Setup

Two experiments were conducted: search and motif discovery. The search experiment demonstrates the first two properties for salient segmentation. The motif discovery experiment further proves the second property by demonstrating the removal of redundancy by salient segmentation. The third property was proven in Section 3.1.1.

The datasets used by this chapter are as follows:

1. MIT-BIH Arrhythmia Database (ECG) [MM88]. This dataset contains several 30-

minute segments of two-channel ambulatory ECG recordings. These sample included arrhythmias of varying significance.

2. Gait Dynamics in Neuro-Degenerative Disease Database [HMF97][HLC00]. This dataset contains data gathered from force sensors placed under the foot. Healthy subjects as well as those with Parkinson's disease, Huntington's disease, and amyotrophic lateral sclerosis (ALS) were asked to walk while the data was recorded. Data includes 5-minute segments for each subject.

3. WALK [WLS11]. This dataset contains a series of annotated recordings from a tri-axial accelerometer worn in a subjects pants pocket. Data was recorded while subjects travelled through the interior of a building.

No reduction algorithms or advanced filtering was used in the analysis of salient segmentation. Each segment was inserted into the index with only a moderate low pass filter (non-weighted averaging window). Reduction algorithms and advanced filtering techniques were excluded to remove any biasing of results.

3.2.1 Search

An index was created for each signal in the test data sets using the salient segmentation technique. Each segment in the index was compared to all possible segments in its respective time series signal using a sliding window with $d = 1$. The closest matches from the sliding window were stored as the true closest matches for each segment. Next, each segment in the index was compared to all other segments in the index. The closest matches in the index were compared to the sliding window's closest matches to create precision-recall curves.

Each dataset was run with three elastic windows: 5, 10, and 20 data points. The elastic window was introduced to account for misalignments resulting from the linear approximations. Additional segmentation parameters for each dataset are given in Table 3.1.

Table 3.1: Segmentation Parameters

Dataset	P	m	Filter Size	m'
ECG	8	100	10	600
Gait	4	80	10	600
Walk	2	25	5	300

The parameter m was chosen as the average size of an interesting pattern. For example, a step in the gait data set (heal down to toe up) was approximately 80 data points. Some care must be taken to avoid rounding errors. In the ECG data set, m was chosen to match the approximate size of the QRS complex (100 data points) instead of the entire heart beat (300-360 data points) to avoid rounding errors. The parameter m' was chosen to give context before and after a salient region. This parameter has no affect on the location of salient points and should be chosen based on the user's need.

3.2.2 Motif Discovery

The motif discovery experiment compares results between the motif discovery algorithm in [MKZ09] and a modified motif discovery algorithm using only salient segments. These methods were compared with two metrics: redundancy and coverage. Redundancy measures the percentage of the time series signal that is represented by more than one motif (i.e., the total amount of data points that exists are repeated in two or more motifs). Coverage measures the percentage of the time series signal that is represented by the returned set of motifs. Both motif algorithms were run with increasing k (finding the k closest motifs) until no new motifs were returned. The parameters listed in Table 3.2 were used for the experiment.

Medical time series are cyclic by nature. The parameters chosen to segment the signal were tuned to find individual cycles, such as one heartbeat or one step. The time series signals

Table 3.2: Motif Discovery Parameters

Dataset	m'	R	Reference Points
ECG	300	2	10
Gait	80	2	10
Walk	60	2	10

used for this chapter contain 90-100% activity. Therefore, motif results should contain high coverage as most of the signal contains interesting patterns.

To assess redundancy, m' was reduced from the search experiment. m' is set to the average complete cycle time for each signal. For example, one heartbeat takes approximately 300 samples in the ECG dataset. Two steps in the WALK dataset (left and right) take approximately 60 data points, and one step in the gait dataset takes 80 data points. Only one step is used for the gait dataset as each channel measures only one foot. Reducing m' focuses the comparison on both methods' abilities to isolate individual cycles. A small m' should result in low overlap. Note, m' has no affect on localizing salient points in salient segmentation as shown in Equation 3.3.

The modified motif discovery algorithm finds the two closest segments $\hat{i}$ and $\hat{j}$ in the index such that $dist(\hat{i}, \hat{j}) = \min_{\forall a,b \in I} dist(\hat{a}, \hat{b})$ (where I denotes the set of all segments in the index). Any segment $\hat{i}'$ with $dist(\hat{i}', \hat{i}) \leq 2 \cdot dist(\hat{i}, \hat{j})$ or $dist(\hat{i}', \hat{j}') \leq 2 \cdot dist(\hat{i}, \hat{j})$ are deemed to be in the neighborhood of $\hat{i}$ or $\hat{j}$ respectively and are grouped together with $\hat{i}$ and $\hat{j}$. All segments in the neighborhood of $\hat{i}$ and $\hat{j}$ are removed from the search and the algorithm is repeated until no segments are left. The algorithm was modified from the classical motif discovery algorithm presented in Section 2.3.1 to closely resemble more recent work in [MKZ09]. The algorithm in [MKZ09] utilizes a fixed window w such that any segment $\hat{i}'$ where $|\hat{i}' - \hat{i}| \leq w$ is a trivial match to $\hat{i}$. The experiments for [MKZ09] were repeated for $w = m'$, $\frac{m'}{2}$, and $\frac{m'}{4}$.

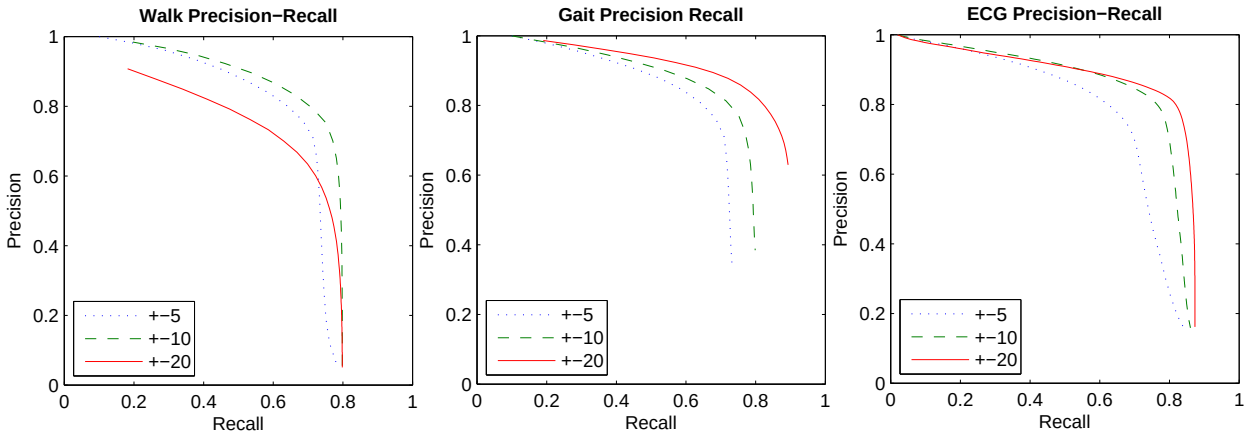

Figure 3.5: Displays precision-recall for the Walk, Gait, and ECG datasets. An elastic window of 5, 10, and 20 data points is displayed for each signal. An increasing elastic window has a small improvement demonstrating salient segmentation's ability to consistently localize salient points.

3.3 Results

3.3.1 Search

Precision-recall curves for the WALK, ECG, and gait datasets are shown in Figure 3.5. Increasing the elastic window size has a small improvement for the gait and ECG datasets. The WALK dataset has a small improvement with a 10-point elastic window and decreased improvement with a 20-point elastic window. There are two sources for the variability in the locations of salient points. First, the RDP approximation adds variability to the location of salient points. Second, similar patterns are not necessarily exact. Therefore, the calculation of saliency may yield slightly different locations of the most salient points in similar patterns. However, both sources of variability are quite small, requiring only small elastic windows to correct alignment.

Precision-recall results will eventually decrease as the elastic window expands. With a constant m', increasing the elastic window increases the probability that a new pattern (not

originally localized by salient segmentation) may be matched. The WALK dataset has a quicker drop-off in performance with respect to the elastic window due to smaller pattern sizes (in terms of data points). The average pattern size (one step) for the WALK dataset is approximately 20-25 data points. An elastic window of 20 or more data points will include an additional pattern (or step) on each side of the isolated segment. These additional patterns cause false positives for segments that lie close to true matches. This phenomenon is shown in Figure 3.5. The WALK data shows a decrease in precision with a large elastic window, but recall is not affected.

The RDP linear approximation of the TSF curve added little variability to search performance. However, close inspection of the TSF curve reveals that the RDP algorithm suppressed a small percentage of peaks (salient points). This suppression resulted in a minor degradation in recall performance. The gait dataset had the simplest time series signals (lowest entropy), with large differentials at the beginning and end of patterns. This causes large peaks in the TSF curve, resulting in the filtering of very few salient points. In contrast, the WALK dataset had the most variable signal (highest entropy) with the smallest differential between the start and end of patterns. The relationship between entropy and recall is shown in the figures with increased recall for lower entropy signals.

The precision-recall results are notable, considering the number of segments suppressed by salient segmentation. Table 3.3 shows the percentage of extracted signal for each time series data set (the elastic window was not included for coverage calculations). Salient segmentation resulted in segmenting below 2% of a sliding window. However, segments within the index spanned near 100% of each time series signal.

3.3.2 Motif Discovery

The top 5 motifs for signals from each of datasets for both salient segmentation and the motif discovery algorithm in [MKZ09] are given in Figure 3.6. Motif discovery with salient

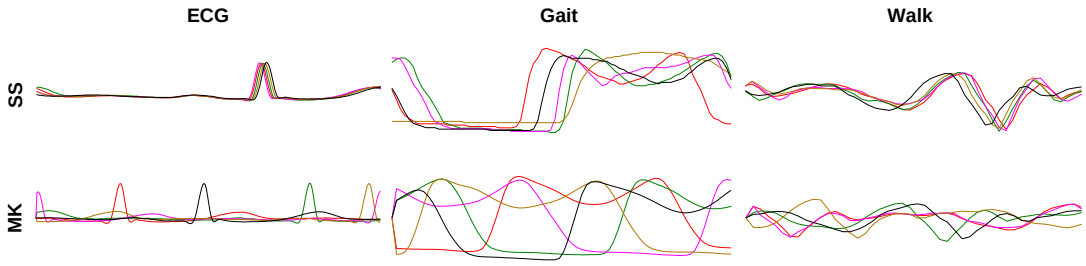

Figure 3.6: Displays the top five motifs returned by the motif discovery algorithm with salient segmentation (SS) and the motif discovery algorithm in [MKZ09](MK). Salient segmentation returns sets of aligned motifs unlike [MKZ09].

segmentation yields aligned motifs, while the comparison algorithm from [MKZ09] yields motifs in an arbitrary alignment. Misaligned results demonstrate a poor representation of a signal's true makeup. By way of illustration, the results for motif discovery with salient segmentation show that all three signals are composed largely of the same pattern. The results from [MKZ09] appear to have five distinct motifs for each signal - but all five patterns are extremely similar and are just returned in different alignments.

Overlap and coverage for all three data sets are shown in Figure 3.7 and Figure 3.8 respectively. The algorithm in [MKZ09] (denoted as MK) shows no overlap when using a window of size m' to filter trivial matches. This is expected as any points lying m' data points from a motif are considered invalid for future rounds of motif discovery, therefore

Table 3.3: Percentage Segmented by Salient Segmentation

Dataset	Segmentation Percentage	Coverage
ECG	.4%	99.9%
Gait	1.8%	96.7%
Walk	1.6%	98.3%

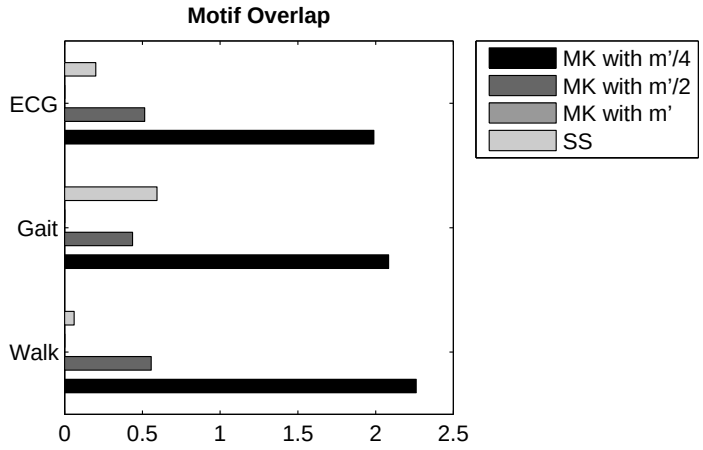

Figure 3.7: Displays the percentage of time series that is represented more than once.

avoiding overlap. Also expected, decreasing m' increases both overlap and coverage. Salient segmentation yields lower overlap than the algorithm in [MKZ09] for the ECG and WALK datasets, and similar overlap for the gait data.

The gait dataset resulted in a significantly higher overlap than the WALK and ECG datasets for salient segmentation. The gait dataset measures pressure from shoes as subjects walked. The on-off pressure is sharp and abrupt, often resulting in two salient points for each step (heel down and toe up). This double segmentation of some steps results in an increased amount of overlap.

Reduced coverage was exhibited for the gait and WALK datasets. Poor coverage had two causes. First, both datasets have small regions of no activity (e.g., such as the subjects standing still). These regions account for 2-10% of these datasets and result in no segmentation within these regions. Therefore, coverage is extremely low in these regions (as expected). Second, the WALK and gait datasets had the highest variability in the length of a pattern cycle. For example, the WALK dataset comprised subjects traversing hallways, and ascending/descending stairways, resulting in variable step lengths. The gait dataset's subject pool consisted of neurologically impaired patients (such as Parkinson's disease and

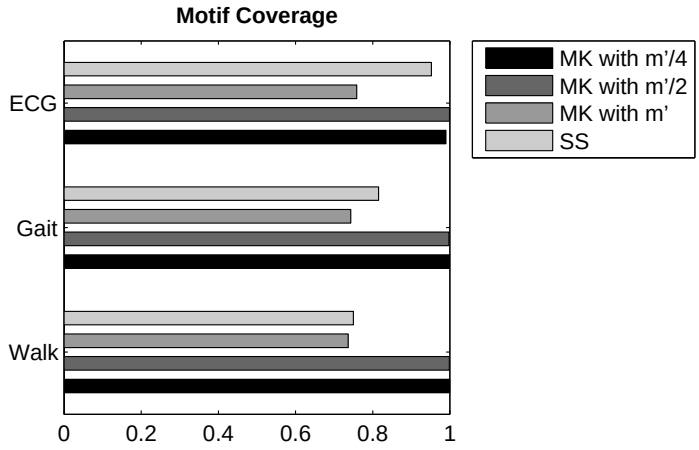

Figure 3.8: Displays the percentage of time series signal represented by the segmentation algorithms.

ALS), resulting in inconsistent gait. Coverage can be improved by decreasing the α parameter for the RDP algorithm (Eq. 3.6) or increasing the size of m'. However, these changes have a trade-off of with redundancy (overlap).

The overlap and coverage results are particularly compelling as no assumptions are made about trivial matches when using salient segmentation. All and only salient segments are matched in the motif discovery algorithm. No assumption is made on the proximity of one segment to another removing the arbitrary measurements proposed in [LP02] [MKZ09]. These results are even more encouraging when considering the alignment offered by salient segmentation. All similar motifs are in alignment vastly improving the quality of motif discovery of algorithms from those in [LP02] [MKZ09].

3.4 Discussion

This chapter presented experiments with a fixed parameter m. This parameter was fixed due to prior knowledge of the test datasets. Real world systems will not have as much a

priori knowledge of an incoming signal. For instance, the ECG dataset consists of heartbeats taken during low activity periods. A more realistic ECG dataset consists of more variable heartbeats; and in general, medical databases must expect a higher variability of incoming signals. Fortunately, salient indices are relatively small and the algorithmic complexity of salient segmentation is low. Therefore, running salient segmentation on several values of m can populate the index with differing levels of granularity to account for variability. As shown with the gait dataset, salient segmentation sometimes suffers from over-segmentation. Such segmentation results in some redundancy over multi-segmented patterns. Some care should be taken when mining for patterns to avoid misleading results from redundancy. However, the redundancy is manageable due to the alignment property offered by the salient segmentation framework. For example, steps in the gait dataset that are segmented twice would exist with two different, but consistent, alignments. When using the algorithm in [MKZ09] similar segments will lie in arbitrary alignments resulting in arbitrary measures to remove redundancy (such as fixed windows for filtering trivial matches).

3.5 Optimizing Time Series Indexes

This section proposes the use of Salient Segmentation to intelligently reduce the size of an LSH index. There are two main contributions to this section. First, this section presents an improved Salient Segmentation technique to that proposed earlier in this chapter. The technique presented previously relies heavily on filtering techniques to extract salient segments. The parametrization of this filtering is domain specific and often arbitrary. The improved Salient Segmentation algorithm requires no filtering, thereby eliminating subjectivity. Second, this section presents the performance improvements of LSH indexes populated with only salient segments over LSH indexes populated with all segments (both salient and non-salient).

The proposed method is evaluated on the three publicly available datasets used previ-

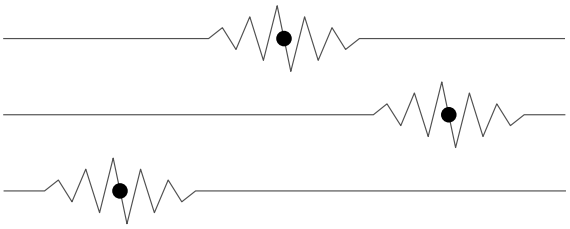

Figure 3.9: Displays the same pattern at three different alignments. The most salient point is marked for each alignment. The TSF defined in (3.5) would label all three alignments with the same saliency resulting in poor localization of the salient point.

ously. Search results of this method are compared to using LSH alone (e.g., populating the index with all segments including both salient and non-salient segments). In all three datasets, the salient index returned near identical results to the complete index and reduced the number of computations by 80% or more.

The new proposed method improves over the previous method in two ways. First, Markov chains are not inherently good at localizing the exact location of a salient point. Fig. 3.9 shows three alignments of the same pattern with the most salient point denoted. The TSF defined in (3.3) labels all three alignments with the same saliency resulting in poor localization of the salient point. Second, (3.3) results in an extremely noisy TSF making it difficult to label local maxima. Both these issues were addressed previously by filtering the TSF. However, the parametrization of the filtering is extremely subjective and domain specific. An improper parametrization can lead to over- or under-segmentation.

This new method address both issues in the previous Salient Segmentation algorithm. First, saliency is calculated by using a range of segment lengths unlike one static length in the previous method. The average of each segment width's saliency centered around point i defines the saliency of i (TSF_i). Second, the saliency function is updated to use entropy. The updated saliency function provides a far smoother TSF resulting in the ability to localize salient points without filtering.

3.6 Improved Method

There are two main components of the experimental implementation: index structure and index population. Index structure utilizes LSH and is the process of indexing segments. Index population is the process of inserting salient segments into the index structure.

3.6.1 Index Structure

The index structure utilizes the LSH scheme based on p-stable distributions defined in [DII04]. The authors propose the following hash function:

$$h_{a,b}(v) = \lfloor \frac{a \cdot v + b}{w} \rfloor, \tag{3.7}$$

where a is a randomized vector following a Gaussian distribution, b is a uniformly randomized vector, and w is a predefined constant. Using the properties of the p-stable distribution, the authors show that the probability of collision is calculated as:

$$p(c) = \int_0^r \frac{1}{c} f_p(\frac{t}{c})(1 - \frac{t}{r})\mathrm{d}x, \tag{3.8}$$

with c being the distance between two vectors. As can be seen by (3.8), the probability of collision decreases monotonically as c increases.

As stated earlier, the gap between p_1 and p_2 is enlarged by combining several functions together. This is accomplished in two ways. First, k hashes are combined to form one parent hash. Objects x and y are matches if all k hashes match between x and y. Second, L parent hashes are created such that objects x and y are matches if at least one of the L parent hashes match between x and y. Therefore, a total of kL hash functions (defined by (3.7)) are created such that each a and b are drawn independently.

The parametrization of the LSH hashing scheme is shown in Table 3.4. A detailed

Table 3.4: LSH Parametrization

Parameter	Value
$\|v\|$	512 (ECG, GAIT)
	64 (WALK)
R	3
w	4
c	4

explanation of the parametrization of LSH is given in [DII04].

3.6.2 Index Population

The index is populated using an improved Salient Segmentation method. This method models a time series as a Markov model such that the probability of each point is defined as follows:

$$p(t_i) = \Pr(T_i | T_{i-1} = t_{i-1}) \tag{3.9}$$

The saliency of a time point t_i is calculated using the entropy of different window sizes centered at t_i:

$$TSF_i = -\frac{1}{|W|} \sum_{w \in W} \sum_{j=i-\frac{w}{2}}^{i+\frac{w}{2}} p(t_i) \log p(t_i), \tag{3.10}$$

where W is the set of window sizes. For best results, window sizes should range from the smallest pattern expected to the largest pattern expected. The experimental window sizes are given in Table 3.5.

Each point t_i that corresponds to a local maximum at TSF_i is extracted as a salient segment defined in the range $[i - \frac{m}{2}, i + \frac{m}{2}]$, where m is the size of the extracted (indexed)

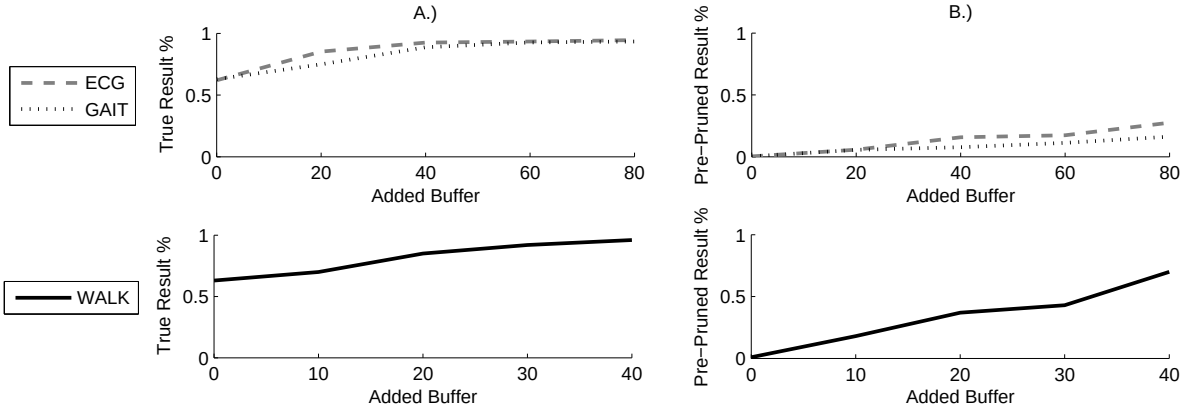

Figure 3.10: A.) Displays the number of true results returned by the salient index over the number of true results returned by the full index with an increasing amount of buffer. B.) Displays the number of LSH results (pre-pruned) of the salient index over the number of LSH results for the full index with an increasing amount of buffer.

segment.

3.7 Results

Two databases were created for the experimental section. Both databases use the LSH indexing structure presented by [DII04]. The first database uses a *salient index* and populates the index with only salient segments. The second database uses a *full index* and populates the index with all segments (i.e., both salient and non-salient segments). As shown in [WLS11],

Table 3.5: Salient Segmentation Parametrization

Dataset	Min	Max	Increment
ECG	25	250	25
GAIT	100	200	10
WALK	20	40	2

Salient Segmentation yields similar alignments for similar patterns. This means that two similar salient patterns may differ slightly in their alignment. Non-elastic distance measures, such as Euclidean distance, can be largely affected by small misalignments. Therefore, recall can be improved dramatically by including a small number of neighboring segments to each salient segment. For example, assuming a salient segment centered at t_i, all segments centered between $t_{i-\frac{p}{2}}$ and $t_{i+\frac{p}{2}}$ are also inserted into the index, where p is defined as the added buffer. During experimentation, p was varied to test its effects.

Fig. 3.10 compares salient indexes (includes only salient segments) to full indexes (includes both salient and non-salient segments). Fig. 3.10 A shows the the number of true results returned by the salient index over the number of true results returned by the full index. An increase in buffer improved the results for all three datasets. However, the increase in buffer also increases the number of pruned results as shown in Fig. 3.10 B. The ECG and GAIT datasets are optimal with a buffer size of 40 (for a salient segment at t_i, index all segments from t_{i-20} to t_{i+20}). For a buffer of 40, the ECG and GAIT datasets retrieve 92% and 89% respectively of the full results and prunes only 15% and 8% respectively than that of the full LSH index. Results for the WALK dataset are best with a buffer of 10-20 with 70%-85% of the full results set with 18%-37% of the number of pruning operations.

The percentage of pruning results increases much faster for the WALK dataset than that of the ECG and GAIT datasets since the segment size is much smaller for WALK. For instance, one cycle (or step) in the WALK dataset consists of approximately 50-60 time points. Therefore, a buffer of size 40 would result in an index that is very close in size to the full index.

The overall performance of the WALK dataset is not as good as the ECG and GAIT datasets due to two reasons. First, accelerometer data is far more diverse than the GAIT and ECG datasets. This means that the measured difference (Euclidean distance) of the accelerometer values between two similar steps is larger on average between two similar heart-

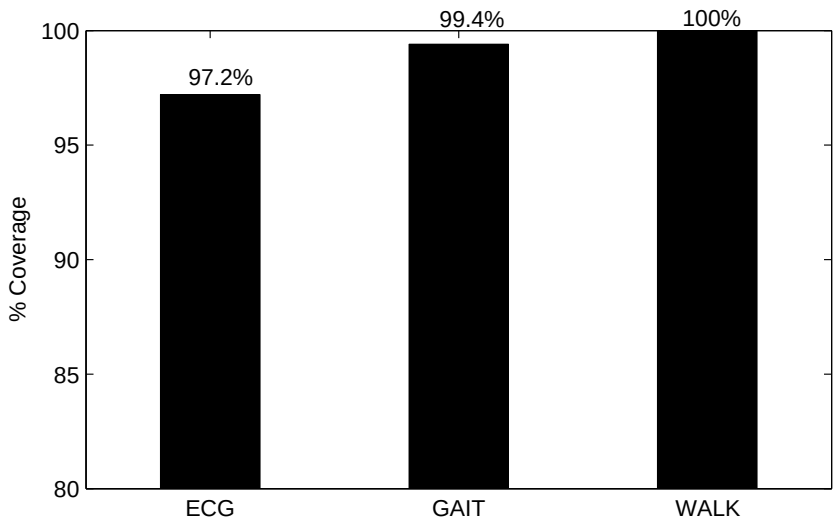

Figure 3.11: Displays the percentage of signal covered by salient segmentation for the ECG, GAIT, and WALK datasets.

beats in ECG or two similar steps in GAIT. Second, the WALK dataset is much smaller than the ECG and GAIT dataset. The full WALK dataset consists of about 0.4M indexed segments versus the ECG and GAIT datasets with 61M and 11.4M indexed segments respectively. Given a larger diversity with a smaller number of potential matches, the experiments must be run with a relatively larger R for WALK (note that all datasets use the same R, but WALK has a much smaller segment size). A smaller R will result in extremely small result sets for both the salient and full indexes. This small R will therefore yields an artificially high performance for the salient index. For example, result sets may be of size 1 where the results include only the search segment.

For the previous experiment, the search segment was randomly selected from the group of salient segments. In order for this experiment to be valid, salient segments should cover a large amount (or all) of their respective time series signal. Fig. 3.11 displays the percentage of signal covered by salient segmentation for the ECG, GAIT, and WALK datasets. All three datasets have coverage of at least 97% and therefore, their respective salient indexes

consist of a large majority of the original time series signals.

CHAPTER 4

Online Aggregation of Biomedical Time-Series Data

Difficulties in searching and mining medical time series data arise from the high dimensionality and the sheer size of the data. Techniques such as k-means clustering and Motif Discovery are not tractable due to high complexities; ($O(n^{dk+1} \log n)$ and $O(dn^2)$ respectively, where d is the number of dimensions, k is the number of clusters, and n is the number of objects). Previous works have reduced the number of dimensions to help mitigate the inherent high runtime complexities of data mining algorithms [FRM94]. However, certain domains, such as medical time series signals, are largely dominated by n ($n >> d$) [WLS11] and therefore, reducing d does very little to improve the overall runtime. In addition, a significant decrease in d may not be possible, as the underlying intrinsic dimensions may also be large. Attempts to decrease d to a smaller space than the intrinsic dimensions will impair the overall discriminatory power of any classifier.

This chapter presents an online aggregation algorithm for constructing efficient time series data indexes. This algorithm takes in a stream of objects and groups them to highly concentrated collections. An aggregated index is composed of an object representative from each collection. Searches and data mining tasks are constrained to only the aggregated index, thereby significantly improving performance with respect to memory and computational complexity. Similarity is defined by the L_2 norm (but any L_p norm can be used). Locality Sensitive Hashing (LSH) [DII04] is used to reduce the overall complexity of the algorithm, allowing it to run online.

The proposed algorithm runs similar to the Online Facility Location problem [Mey01].

An input segment is hashed using LSH. If this hash collides with a facility (e.g., an existing grouping), the segment is assigned to that facility. If no collisions occur, a new facility (or grouping) is created and inserted into the global hash table. The aggregation algorithm is based on the assumption that classes, in terms of classification, are composed of multiple tight sub-groupings where L_p norms have high discriminatory power [BFG99].

Similar indexes have been proposed in textual databases such that redundant information is indexed only once [HLY07][ZS07][BEF06]. Such indexes are often used with versioned data such as indexing the Internet Archive (a collection of over 85 billion versioned web pages over the last decade), version control systems, wikis, data backup solutions, etc. In general, documents are broken down into fragments. Fragments that occur in multiple documents (or versions) are indexed only once. Fragments are chosen in an optimal (aligned) manner such that the number of index fragments is minimized. These systems have been shown to significantly improve search performance by reducing the overall search space.

Index compression is another approach to reducing the size of a textual database. Early algorithms ordered an index by document IDs [ZM06]. Instead of containing each document's ID, these techniques contained the difference between a document's ID and the previous ID (termed d-gap). Such a technique will result in smaller integers contained in the index thereby increasing compression ratios. Later techniques reassigned document IDs such that documents that shared similar terms had similar IDs [SCS03][SOP04]. In such schemes, the index contains many tight clusters with small d-gaps with large d-gaps between clusters. This layout would increase compression ratios over randomly assigned documents IDs as the average d-gap was very small. While not studied by this dissertation, such a technique is complementary to the technique proposed by this chapter. Segment IDs could be reordered in a similar fashion and decrease d-gaps allowing for significant index compression.

Real data is used in the analysis of this algorithm. The proposed algorithm yields logarithmic growth of groupings while keeping sensitivity and specificity above 98%. Search

and clustering performance (in terms of computations) is improved by several orders of magnitude. This algorithm has a low computation and memory complexity, allowing it to run online.

4.1 Method

The proposed algorithm takes in a stream of objects from a biomedical time-series signal. Objects are defined by the segmentation. This chapter assumes two types of segmentation: segmentation by events and segmentation by sliding window. Event-based segmentation are domain specific. Examples include heartbeats of an ECG signal, a cycle in the arterial blood pressure, etc. For sliding window segmentation, segments are extracted by sliding a window along the time dimension, indexing each possible segment. Each slide (or translation) of the window is of D data points where $D \geq 1$ (this chapter sets $D = 1$ for all datasets). A fixed window size is assumed for both types of segmentation.

On an input object u, the following four procedures are run:

1. Find the hash of u (H_u) using LSH
2. Search the global hash table for collisions V
3. Calculate $d_v = ||u - v||_p \forall v \in V$
   (a) If $d_v < R$, assign u to v (on multiple matches, assign randomly)
   (b) Else, create a new grouping g with hash H_u, add u to g, and add g to the global hash table

The LSH implementation using p-stable distributions is used as the hashing algorithm [DII04]. This algorithm uses the following hash function:

$$h_{a,b}(v) = \lfloor \frac{a \cdot v + b}{r} \rfloor \tag{4.1}$$

Where a is a randomized vector following a Gaussian distribution, b is a uniformly randomized vector, and r is a predefined constant. Using the properties of the p-stable distribution, the authors show that the probability of collision is calculated as:

$$p(c) = \int_0^r \frac{1}{c} f_p(\frac{t}{c})(1 - \frac{t}{r})\mathrm{d}x \tag{4.2}$$

With c being the distance between two vectors. As can be seen by Equation 4.2, the probability of collision decreases monotonically as c increases.

Each grouping is represented by its initial object. Upon completion of the aggregation, each grouping's representative is placed into the aggregated index. LSH indexing is used in the experimentation for this chapter. However, this work could be applied to other indexing techniques such as spatial trees. LSH was chosen for analysis due to LSH's sub-linear theoretical complexity.

Searches are constrained to only the aggregated index. A mapping table is kept between a grouping's representative object and all its contents. When a matching representative is found, all objects from its respective grouping are returned.

4.1.1 Computational Complexity

This chapter assumes an optimal hashing algorithm with a sufficiently large table. Therefore, the complexity of the algorithm is dominated by pruning (or distance computations). As shown in [IM98], the number of distance computations per object is bounded by $O(n^{\rho})$, where $\rho = \frac{\ln 1/p_1}{\ln 1/p2}$ and n is the total number of objects. This yields a total runtime complexity of $O(n^{2\rho})$. In practice, the overall runtime is much lower as the search space is significantly reduced (i.e., the search space grows sub-linearly with respect to the number of objects).

Therefore, the actual the number of groupings ($\hat{n}$) at any point in the algorithm is much less than n ($\hat{n} << n$). Therefore, we get a runtime complexity bounded by $O(n\hat{n}^{2\rho})$ where, in general, $\hat{n} << n$.

Memory is bounded by $O(n + nL)$. Notice that k is not included in the memory computation, because LSH creates L different hashes, each composed of k sub-hashes. A match is defined by two segments that match for at least one of the L hashes. A matching of one of the L hash functions is defined as a match of all of the k sub-hashes. Therefore, each of the k sub-hashes can be combined and hashed to a single integer. Hence, the overall storage required to store L LSH hash results is L. As stated earlier, the number of groupings $\hat{n}$ is less than the total number of objects. In practice, the algorithm will be bounded by $O(n + \hat{n}L)$.

4.2 Experimental Setup

Four datasets composed of various types of biomedical signals are used in the analysis of this chapter. The datasets are as follows:

1. MIMIC Database [MM96][GAG00]. This dataset contains multiple channel recordings taken from patients in intensive care units. Electrocardiogram (QRS), arterial blood pressure (ABP), and fingertip plethysmograph (PLE) were used in the analysis.

2. MIT-BIH Arrhythmia Database (MITDB) [MM88][GAG00]. This dataset contains several 30-minute segments of two-channel ambulatory ECG recordings. These sample included arrhythmias of varying significance.

3. Gait Dynamics in Neuro-Degenerative Disease Database [HMF97][HLC00][GAG00]. This dataset contains data gathered from force sensors placed under the foot. Healthy subjects as well as those with Parkinson's disease, Huntington's disease, and amy-

otrophic lateral sclerosis (ALS) were asked to walk while the data was recorded. Data includes 5-minute segments for each subject.

4. WALK [WLS11]. This dataset contains a series of annotated recordings from a tri-axial accelerometer worn in a subject's pants pocket. Data was recorded while subjects travelled through the interior of a building.

The proposed aggregation algorithm is assessed using five experiments. The first experiment aggregates times series signals and displays the growth of groupings with respect to the number of objects with varying values of R. This experiment is run using both event-based segmentation and sliding window segmentation. All three channels of the MIMIC dataset were used for testing event-based segmentation. Segmentation was based on the dataset's annotations. For sliding window segmentation, the MIMIC database along with the GAIT and WALK datasets were used.

The second experiment tested the sensitivity and specificity of the proposed aggregation algorithm. The MITDB database was used for this experiment. The MITDB database is composed of well-annotated two-channel ECG. Time series signals from the MITDB database were aggregated using the proposed algorithm with event based segmentation. Each segment is initially labelled using the dataset's annotations (ground truth labels). Each grouping is labelled using its representative's ground truth label. Sensitivity and specificity are calculated by comparing each segment's true label to that of its grouping label. The MITDB was used in lieu of other datasets as those datasets do not have annotated class labels for the segmented objects.

The third experiment assesses the improvements of clustering when limiting the clustered elements to only grouping representatives. The Fast k-means algorithm [Elk03] is run on both the aggregated and non-aggregated data. Only event-based segmentation of the MIMIC dataset was used. Sliding window segmentation was not assessed as it has been shown to be ineffective for clustering time series signals due to redundancy [KL05]. The number of

iterations is used to demonstrate the performance improvements when using an aggregated signal.

The fourth experiment compares the size of salient indexes [WLS11] to that of aggregated indexes. The MITDB, GAIT, and WALK datasets were used (as these were the same datasets used in [WLS11]).

The fifth, and final experiment, tests the improvement of subsequence search when using aggregated indexes. The index was populated with the MITDB dataset along with two additional datasets. The MIT-BIH Noise Stress Test Database (NSTDB) [GAG00][MMM84] and the MIT-BIH ST Change Database (STDB) [GAG00] were added to increase the overall size of the database. Both a standard LSH index along with an aggregated LSH index were composed. 100 random searches were performed while measuring the respective memory usage, precision and recall.

Each segment is normalized using the standard score normalization function:

$$\frac{X - \mu}{\sigma} \tag{4.3}$$

No filtering was used unless explicitly noted. In general, proper filtering improves the results of machine learning and data mining tasks. However, this chapter forgoes filtering processes to avoid any effects of filtering (both positive and negative). A fixed segment size was used is the experimentation that varied across datasets. Each dataset's parametrization is listed in Table 4.1. As it is assumed that each dataset is cyclical, segment sizes were chosen to encapsulate one cycle. Cycle sizes do vary; however, the proposed algorithm is fairly resilient to the choice of segment size (as shown by the results).

Table 4.1: Dataset Parameters

Dataset	Segment Size	Sampling Rate
MIMIC	128	125Hz
MITDB	512	360Hz
GAIT	512	300Hz
WALK	60	50Hz
NSTDB	512	360Hz
STDB	512	360Hz

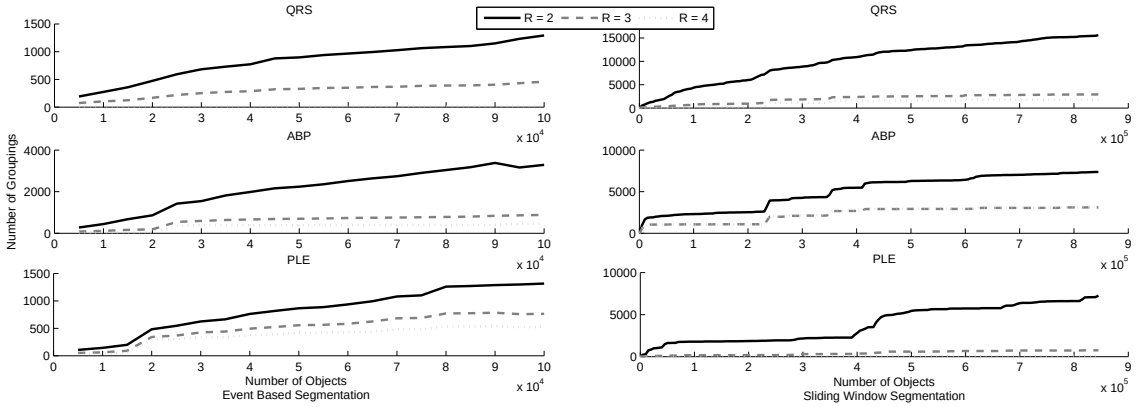

Figure 4.1: Displays the growth of clusters with increasing number of objects for the MIMIC dataset (including QRS, PLE, and ABP). Both event-based segmentation (left) and sliding window segmentation are shown (right). Each type of data shows sub-linear growth with a decreasing rate of growth with increasing R.

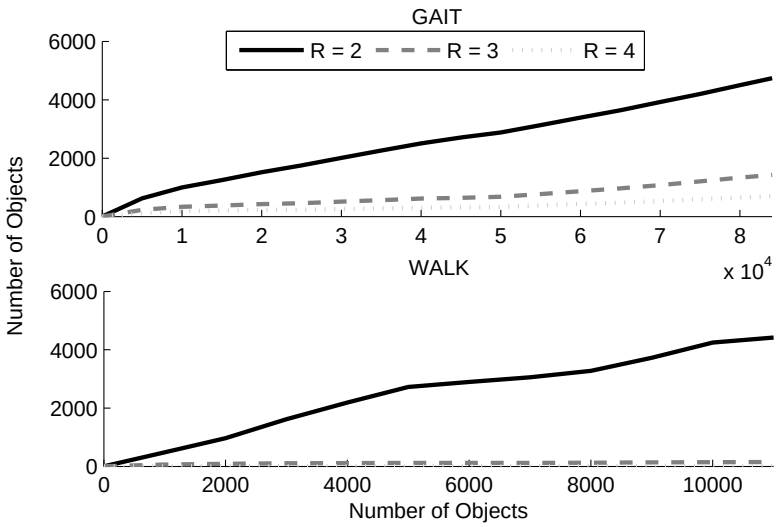

Figure 4.2: Displays the growth of clusters with increasing number of objects for the GAIT and WALK datasets. Only sliding window segmentation is used. Both datasets show sub–linear growth with a decreasing rate of growth with increasing R.

4.3 Results

4.3.1 Sub-linear growth of groupings

Fig. 4.1 and Fig. 4.2 displays the growth of groupings with respect to the total number of segments. Fig. 4.1 displays this growth with the MIMIC dataset (including QRS, PLE, and ABP). The left side of the graph displays the growth for event based segmentation and the right displays results for sliding window segmentation. Fig. 4.2 displays the growth of groupings for the GAIT and WALK dataset with only sliding window segmentation as there are no annotations for segmentation for these two datasets. Three values for the radius were tested: $R = 2, 3$, and 4.

All datasets show sub-linear growth with respect to the total number of objects. Growth is decreased with an increased radius. The rate of growth appears to be similar for both sliding window segmentation and event-based segmentation. Note that sliding window segmentation contains a much larger number of segments than that of its sliding window counterpart.

The effects of the radius R vary across datasets. For example, the WALK dataset shown in Fig. 4.2 reduces the total number of segments by about one third for a radius of $R = 2$. However, a radius of $R = 3$ or $R = 4$ shows excellent sub-linear growth. This is caused by two factors. First, the WALK dataset is much smaller than all other datasets. As more data is collected, there is a high probability of seeing a pattern that was previously seen. Therefore, smaller datasets may not experience as much reduction as larger datasets. Second, the WALK dataset is the most variable dataset tested by this chapter. Accelerometer data is inherently noisy especially when the accelerometer is not affixed to the body (as with the WALK dataset). The results in Fig. 4.2 demonstrates the algorithm's susceptibility to noise. This noise can be improved by increasing R as well as adding filtering.

Fig. 4.3 shows the effect of filtering the data before aggregation for the WALK dataset. A basic high pass filter is used by assigning each time point as the mean of a surrounding window of size 10 (approximately 0.2 seconds). A significant decrease in the cluster growth is observed. The choice of filter can either improve or hurt the results (in terms of sensitivity and specificity). For brevity, a complete analysis of proper filtering techniques is left to future work.

A summary of the reduction of each type of data is given in Fig. 4.4. MITDB, NSTDB, and STDB are excluded as they are all ECG (same as QRS for the MIMIC dataset).

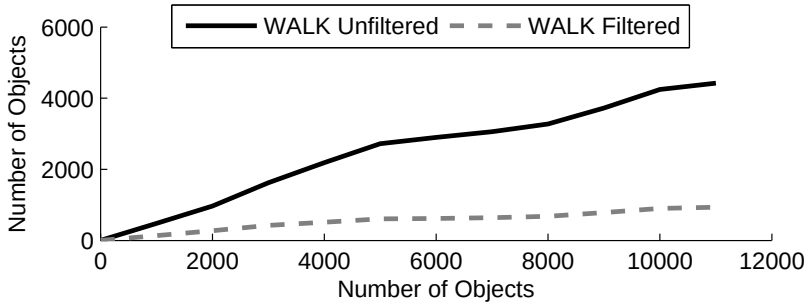

Figure 4.3: Displays the growth of clusters with increasing number of objects for the WALK dataset with a fixed $R = 2$. Results are shown with the WALK dataset filtered and unfiltered. Filtering is shown to slow object growth.

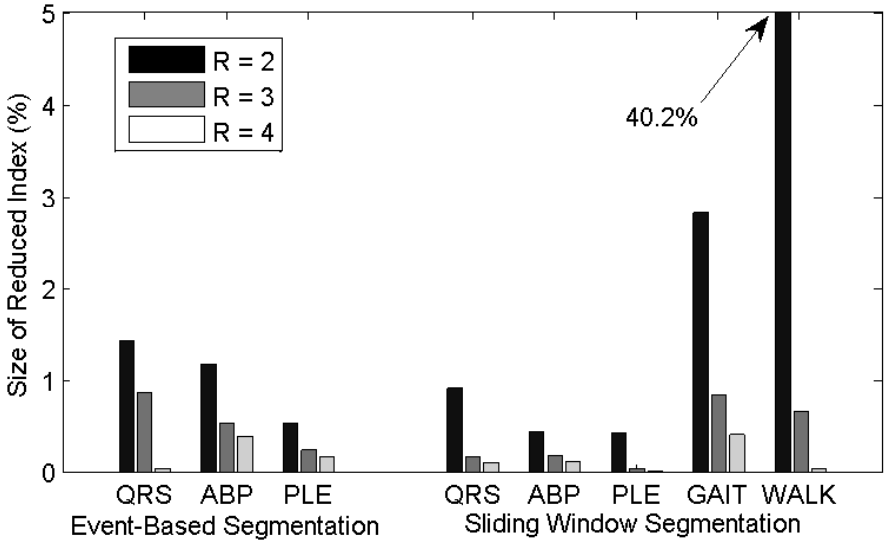

Figure 4.4: Displays the aggregated index sizes for all types of data with $R = 2, 3$ and 4 as compared to original index size.

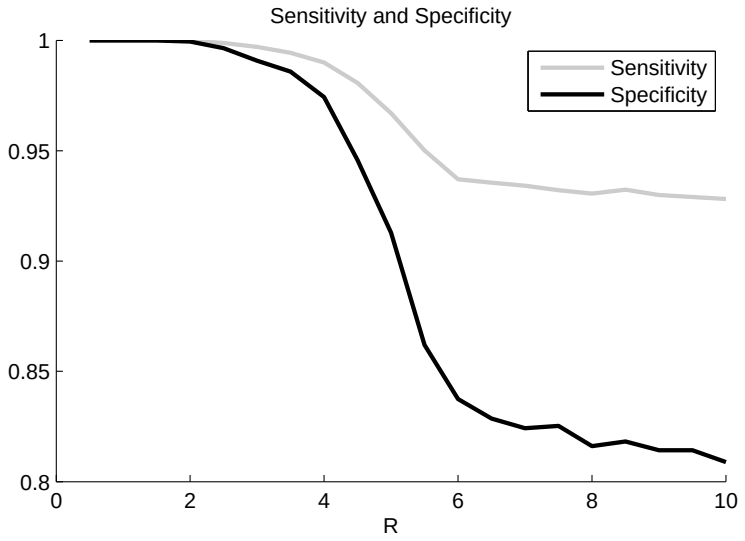

Figure 4.5: Sensitivity and Specificity with varying radius (R) for the MITDB dataset. Both specificity and sensitivity drop significantly with $R > 4$.

4.3.2 Sensitivity and Specificity

Increasing the radius R for the proposed algorithm decreases the growth of groupings. However, this increase in R has a cost. Increasing the radius for an acceptable match increases the probability of incorporating incorrect matches to a grouping. Fig. 4.5 shows the change to sensitivity and specificity with varying values of R. As expected, an increase in R degrades performance of the proposed algorithm. A significant drop off is observed for a radius of $R > 4$. Hence, all other experiments used a value of $R \leq 4$.

The effect of R on the sensitivity and specificity may change depending on the dataset. However, only the MITDB dataset was used in this set of experiments as no other datasets included in this chapter have appropriate annotations for calculating performance measures (i.e., only MITDB contains class labels).

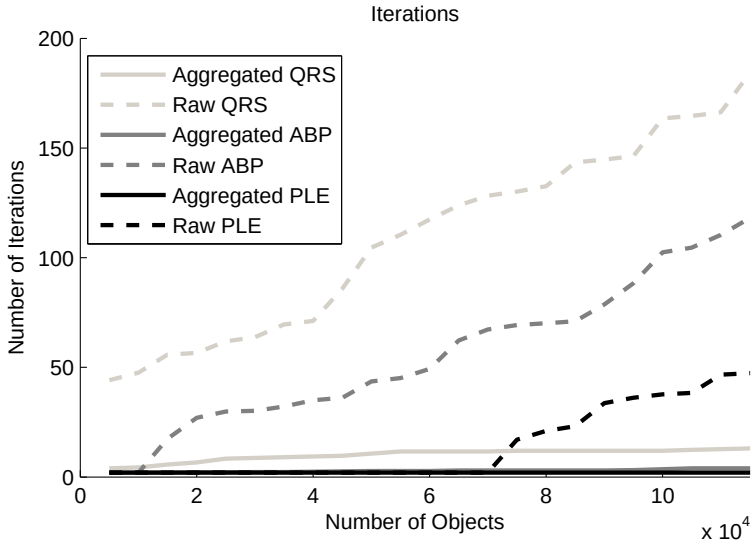

Figure 4.6: Displays the number of iterations when clustering using the fast k-means algorithms [Elk03]. Results for both aggregated and raw data is shown for the MIMIC dataset.

4.3.3 Clustering on aggregated indexes

Fig. 4.6 displays the number of iterations of the fast k-means algorithm in [Elk03] for the MIMIC dataset. The number of iterations is shown as a function of the total number of segments. Only event-based segmentation was used in this experiment as it has been shown that clustering using sliding window segmentation has little to no meaning [KL05].

As shown by Fig. 4.6, the number of cluster iterations for non-aggregated indexes exhibits growth that is greater than linear. When clustering on aggregated indexes, a seemingly linear (or sub-linear) growth is observed. Fig. 4.5 demonstrates that the aggregation results in only a small degradation in sensitivity and specificity with small values of R. Data mining on aggregated indexes will therefore make algorithms more tractable while ensuring a high level of accuracy.

Table 4.2: Sizes of Salient and Aggregated Indexes

Dataset	Salient Segmentation	Aggregated Index		
		$R = 2$	$R = 3$	$R = 4$
MITDB	0.4%	11.1%	1.8%	1.0%
GAIT	1.8%	5.6%	1.7%	0.82%
WALK	1.6%	40.2%	1.3%	1.0%

4.3.4 Comparison of Salient and Aggregated Indexes

Table 4.2 displays the index sizes of Salient Segmentation versus those from the proposed aggregated indexing algorithm. Both algorithms achieve similar results. However, Salient Segmentation is lossy, meaning much of the time series signals are excluded from the database. When using aggregated indexing, only the groupings are added to the index (search space). However, any segment belonging to a grouping can be associated to the grouping through a mapping table. Hence, the database may still contain all segments from the original signal resulting in lossless storage of the signal.

4.3.5 Comparison of LSH on aggregated and non-aggregated indexes

Table 4.3 displays memory and precision for both an LSH index and an LSH aggregated index. Precision and recall are calculated by the annotation labelling. Precision for LSH alone is marginally better than LSH with an aggregated index. However, the average memory for each query is approximately 15 times less for the aggregated index. As the growth of groupings is less than linear with respect to the number objects, memory improvements will increase with larger databases.

The aggregated index has a recall that is slightly better than LSH alone. An improvement in recall is largely due to distant matches (those that have a distance of approximately R to the query segment) that are less likely to have colliding hashes with the query segment.

Table 4.3: Memory, Precision and Recall for Aggregated Index as compared to LSH alone

Non-Aggregated Index		Aggregated Index		
Precision	Memory	Precision	Memory	Recall
0.9523	9424kB	0.9428	691kB	+5%

For example, if a query segment has two distant matches, where both distant matches are extremely close to each other, then it is possible that the query segment will have a matching hash with just one of the distant matches. In addition, the two distant matches will probably have several matching hashes resulting in the two matches being aggregated to the same grouping. Therefore, LSH alone will miss one of the distant matches, while LSH on an aggregated index will likely match to the grouping and retrieve both distant matches.

Precision is slightly lower for the aggregated index as any distant groupings (distance of approximately R to the query segment) may contain objects that are greater than R from the query object. However, the effects of such groupings appear to be minimal.

4.4 Summary

This chapter presented an algorithm for constructing an aggregated medical time series index. The algorithm is based on the observation that medical time series signals are often composed of similar and repetitive cycles. Therefore, the size of the index can be significantly reduced by only including unique patterns. For the purpose of this chapter, two patterns are considered to be the same if the corresponding Euclidean distance is less than R. Euclidean distance has been shown to have high discriminatory power for small values of R [BFG99]. Hence, small neighborhoods with radius R tend to consist of a single class.

The proposed algorithm runs similar to the Online Facility Location problem [Mey01]. An input segment is hashed using LSH. If this hash collides with a facility (e.g., an existing grouping), the segment is assigned to that facility. If no collisions occur, a new facility (or

grouping) is created and inserted into the global hash table. Each grouping is represented by its first resident and only the representative is added to the index.

This chapter showed that the total number of groupings created by the algorithm grows sub-linearly with respect to the total number of objects for many medical time series signals. Aggregated indexes are shown to significantly improve performance of searching and mining medical time series with little degradation in the quality of results.

CHAPTER 5

Monte Carlo Subsequence Matching

5.1 Introduction

This chapter presents a randomized Monte Carlo approach for improving search results. This approach enlarges search results while ensuring precision and yielding higher relative information gain. This method is built upon two assumptions: time series databases are extremely large, and result sets follow a Gaussian distribution. The proposed method consists of two steps. First, a query segment q undergoes m randomizations constructing a set Q of query segments where $|Q| = m$. Next, a R-NN search for each $u \in Q$ is performed using the l_2 norm (Euclidean distance). The Euclidean distance between q and all segments $u \in Q$ follows a Gaussian distribution with a mean μ_Q and standard deviation σ_Q determined by the randomization.

Locality Sensitive Hashing (LSH) [GIM99] is employed as the underlying hash-based nearest neighbor search algorithm. Under an LSH family of hash functions, similar objects have a higher probability of collision (and vice-versa). A search using LSH has a proven sub-linear computational complexity. This is unlike spatial time series databases that have been shown both theoretically and experimentally to perform worse than sequential search for data with as little as ten dimensions [WSB98].

Results from this chapter are shown both theoretically and experimentally. Experiments are run on both synthetic random walk and real-world publicly available datasets. The randomized approach increased the number of search results by several orders of magnitudes

over LSH alone while keeping similar preciseness. Experimental databases contained tens of millions of indexed subsequences showing both correctness and scalability. However, the proposed algorithm is highly parallelizable, potentially allowing for databases of a much larger scale.

5.2 Method

The following solution is founded on two assumptions:

1. Time series databases are extremely large; and
2. Result sets follow a normal distribution.

The first assumption implies that only a subset of true matches is required by a query, and therefore, an accurate sampling is sufficient. The second assumption allows for a Monte Carlo estimation of the true result set. Distances from a search segment s to its result set S are therefore modelled as a normal distribution $N(\mu_S, \sigma_S)$. m randomizations of a query segment s are used as queries into a time series database using Locality Sensitive Hashing (LSH). The randomizations of s are created such that the query results $\hat{S}$ are a Monte Carlo approximation of S and therefore form a normal distribution with μ_S and σ_S

Each time series segment is assumed to be normally distributed ($N(0,1)$). The distance between two time series segments s and $\hat{s}$ is defined as follows:

$$\frac{dist(s,\hat{s})}{2} = \sqrt{\sum_{i=0}^{n}\left(\frac{s(i)-\hat{s}(i)}{2}\right)^2}, \tag{5.1}$$

where 2 is a normalization factor due to the fact that the distribution of the differences between two $N(0,1)$ variables is $N(0,2)$. This results in $dist(s,\hat{s})$ to be distributed as a chi distribution with the following property:

$$\lim_{n\to\infty} \frac{dist(s,\hat{s})/2 - \mu_d}{\sigma_d} \sim N(0,1), \tag{5.2}$$

where μ_d and σ_d are the mean and standard deviation, respectively, of the match population of s. This chapter assumes a large n of at least a several hundred (> 200) allowing the previous property to hold approximately true. We therefore can produce a sampling of matches of s by randomizing each time point in s to create $\hat{s}$ such that:

$$s(i) - \hat{s}(i) \sim N(0, \sigma_r^2), \tag{5.3}$$

where σ_r is the standard deviation of the randomization of s. We define Y to be an independent random variable drawn from $dist(s, \hat{S})$ and X to be an independent random variable drawn from $s(i) - \hat{s}(i)$. Y is therefore distributed as:

$$Y = \sqrt{\sum_{i=0}^{n} X_i^2} = \sigma_r \chi_n \tag{5.4}$$

This yields a mean and standard deviation of:

$$\mu = \sigma_r \mu_{\chi_n} = \sigma_r \sqrt{2} \frac{\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})} \tag{5.5}$$

$$\sigma = \sigma_r \sigma_{\chi_n} = \sigma_r \sqrt{(n - \mu_{\chi_n}^2)}, \tag{5.6}$$

with $\Gamma(x)$ being the gamma function defined as:

$$\Gamma(x) = \int_0^\infty t^x e^{-t} \mathrm{d}t \tag{5.7}$$

A lookup table for μ and σ can be found in Table 5.1 as these computations can suffer from rounding error. For large n, distances between s and elements in $\hat{S}$ are given by the following distribution:

Table 5.1: Lookup table for μ and σ assuming a $\sigma_d = 1$.

n	μ	σ^2
128	11.2916	.4990
256	15.9844	.4995
512	22.6164	.4998
1024	31.9922	.4999
2048	45.2493	.4999

$$Y \sim N(\sigma_r \mu_{\chi_n}, \sigma_r \sigma_{\chi_n}) \tag{5.8}$$

However, the likelihood that a randomized segment exists within a database is extremely low. This is especially true as both the granularity of points and length of time series (n) increase. Therefore, a Monte Carlo approximation is unlikely to yield reasonable results unless the number of samples is infinitely large. Therefore, all neighbors within a distance R to a randomized signal is used in the set of returned results. The probability of finding a segment s with distance in the range of $[x - R, x + R]$ to the search segment can be defined as:

$$f(z) = \frac{1}{2R\sqrt{2\pi\sigma^2}} \int_{x-R}^{x+R} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \, \mathrm{d}x \tag{5.9}$$

There is no closed form solution to Equation 5.9, however, as R approaches 0, $f(z)$ approaches a Gaussian distribution. This chapter assumes an R that is relatively small. Hence, the error introduced by LSH adds a negligible amount to the bounds of the proposed randomization approach. This is heuristically shown in Figure 5.1.

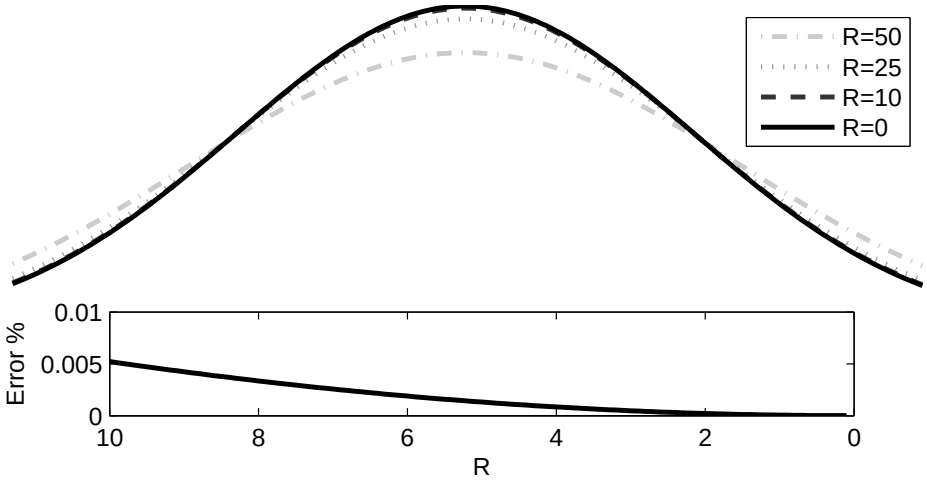

Figure 5.1: Shows the convergence of Equation 5.9 as $R \to 0$.

5.2.1 Complexity

As stated earlier, the algorithmic complexity of LSH is sub-linear and dominated by $O(N^{\rho})$ distance computations (pruning) where $\rho = \frac{\ln 1/p_1}{\ln 1/p_2}$ and N being the size of the database. Assuming a segment size of n, the randomization process is $O(mn)$ where m is the number of randomizations. LSH is run m times yielding a complexity dominated by $O(mN^{\rho})$ (as $O(mN^{\rho}) > O(mn)$). This is still sub-linear as $m << N$.

5.2.2 Trivial Matches

This chapter uses the formal definition of trivial matches proposed in [LP02]. Two segments $\hat{i} = [i, i+m']$ and $\hat{j} = [j, j+m']$ are trivial matches if $dist(\hat{i}, \hat{j}) \leq thr$ and there is no segment $\hat{i}' = [i', i'+m']$ where $i \leq i' \leq j$ and $dist(\hat{i}, \hat{i}') > thr$ [LP02]. Trivial matches add redundancy to the result sets as they represent the same patterns with small translations. This chapter probabilistically filters trivial matches by removing any segments that lie immediately next to or one time point away from a previously extracted segment. More formally, given a segment u_i returned by an LSH search, the segments u_{i-1}, u_{i-2}, u_{i+1}, u_{i+2} will be filtered as

trivial matches (where i is the starting point of segment u). In addition, all trivial matches to previously declared trivial matches will also be removed. The observation is that trivial matches occur in small runs of neighboring segments. The probability of LSH missing one neighboring trivial match (p_2) is reasonably high in practice, but the probability of missing two neighboring segments(p_2^2) is quite low.

5.2.3 Filtering Optimization

LSH returns all segments with colliding hashes. The result set may contain several false positives. These false positives are filtered by taking the Euclidean distance between the query segment and all matches returned by LSH. Any segment with distance greater than R to the query segment is filtered. During experimentation, it was found that a large proportion of time was spent pruning false positives. Each dataset consists of tens of millions of indexed segments, and therefore, even a small percentage of false positives results is a large number of segments. This is especially true when aggregated over several random query segments.

To improve performance, results were pruned using the original query segment and not the randomized query segment. Note, that the original algorithm searches and prunes results for each randomized segment independently. In this optimization, results were kept with probability based on their distances to the original query segment using the distribution $Y \sim N(\sigma_r \mu_{\chi_n}, \sigma_r \sigma_{\chi_n})$.

This pruning technique also improved the results for LSH (deemed LSH with sampling). Therefore, this chapter compares the Monte Carlo approximation with both LSH and LSH with sampling. To note, LSH result sets could be improved by raising the value of R and running LSH with sampling to avoid the overhead added by the Monte Carlo scheme. However, the search runtime using LSH is heavily dominated by the number of distance comparisons during pruning. The Monte Carlo approximation raises the theoretical bound of pruning by a factor of m. On the other hand, raising R exponentially increases the theoretical bound.

With the assumption of uniformly distributed search space, raising R increases the search space by R^d (where d is the number of dimensions).

5.3 Setup

This chapter leveraged four datasets in its assessment of the proposed method. These datasets include:

1. MIT-BIH Arrhythmia Database [MM88] (ECG). This dataset contains several 30-minute segments of two-channel ambulatory ECG recordings. These sample included arrhythmias of varying significance.

2. Gait Dynamics in Neuro-Degenerative Disease Database [HLC00] [HMF97] (GAIT). This dataset contains data gathered from force sensors placed under the foot. Healthy subjects as well as those with Parkinson's disease, Huntington's disease, and amyotrophic lateral sclerosis (ALS) were asked to walk while the data was recorded. Data includes 5-minute segments for each subject.

3. Synthetic Signal (SYN) - This dataset was created using the function $f(x) = \sum_{i=3}^{7} \frac{1}{2^i} * \sin(\pi x 2^{3+i}) + rand(1)$. This is a derivative of the pseudo periodic synthetic time series data set presented in [PLC99]. The randomization inserted by $rand(1)$ is to model noise. This dataset helps to assess both LSH and the Monte Carlo approximation's stability with noisy time series signals.

4. Synthetic Shapes (SHAPE) - This dataset was created to asses precision and recall and is composed of ten different shapes with varying levels of Gaussian noise displayed in Fig. 5.2

Each dataset was indexed and inserted into the the database. A total of 50 segments were randomly chosen as query segments from each dataset. These segments were searched

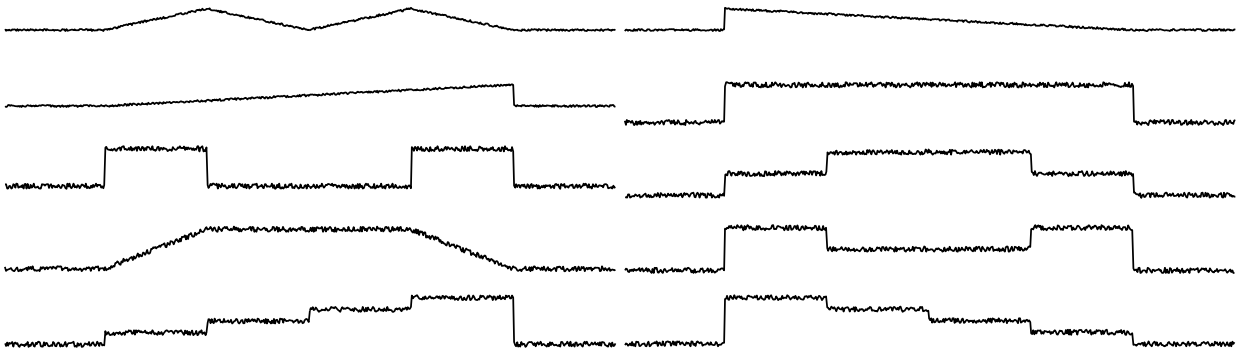

Figure 5.2: Displays the ten shapes (classes) in the synthetic shape dataset with Gaussian noise.

using LSH, LSH with sampling (proposed in Section 5.2.3), and the full Monte Carlo method proposed by this chapter. Both the standard deviation of the randomization (σ_r) and the number of randomizations (m) were varied for the Monte Carlo experiments. The attributes of each dataset are given in Table 5.2. The radius R is relative to the normalized distances (e.g., segments are normalized before finding the Euclidean distance).

The percentage improvements of result sizes of the Monte Carlo approximation over LSH and LSH with sampling are shown. The percentage increase is used as a comparison in lieu of raw counts due to the high variability in the size of result sets. The number of results is highly dependent on the random segment extracted from the time series signal. For example, a common heart beat from the ECG signal will generally match to several thousand segments. An anomaly, on the other hand, may only match to a couple of segments.

Precision and recall was assessed only for the SHAPE dataset with varying amounts of Gaussian noise. The ECG, GAIT, and SYN datasets are not annotated to label specific neighborhoods. In addition, these datasets have tens of millions of instances and manually creating these labels is not feasible. However, the respective means and standard deviations of the Euclidean distance between the result sets and the query segment are given to show

Table 5.2: Dataset Attributes

Dataset	Number of segments	Segment length	R
ECG	70M	512	3
GAIT	15M	512	3
SYN	10M	128	3
SHAPE	1M	600	1.5

correctness. In addition, example result sets are displayed visually for each dataset for further evidence of correctness.

The run time of the Monte Carlo approximation was also evaluated. This experiment consisted of 50 random runs using both LSH and the Monte Carlo approximation. Wall clock time (in seconds) is presented. All experiments were run on an Intel(R) Core(TM) i5 CPU 650 processor clocked at 3.2 GHz [Int11] running Ubuntu 11.10 [Pet11].

The Monte Carlo approximation and LSH were implemented in Python version 2.7 [Pyt09] and utilized a MySQL 5.1.58 database [MyS] for storing, indexing, and retrieving hash values. Raw time series signal were segmented using a sliding window and all unique segments were indexed. An indexed segment in the MySQL database contained a link to the raw time series signal along with its location within the time series. A segment was indexed using its hash values generated by LSH. Segments were retrieved from the file system in batches by sorting segments by their respective host file and location within this file. This is an extremely important optimization as file systems perform poorly under random access.

The implementation of LSH in this chapter is based on disk storage unlike that of [DII04]. [DII04] assumed a database small enough to be stored in memory. However, the databases used by this chapter are extremely large (tens of millions of instances). Hence, the system had to utilize disk based storage.

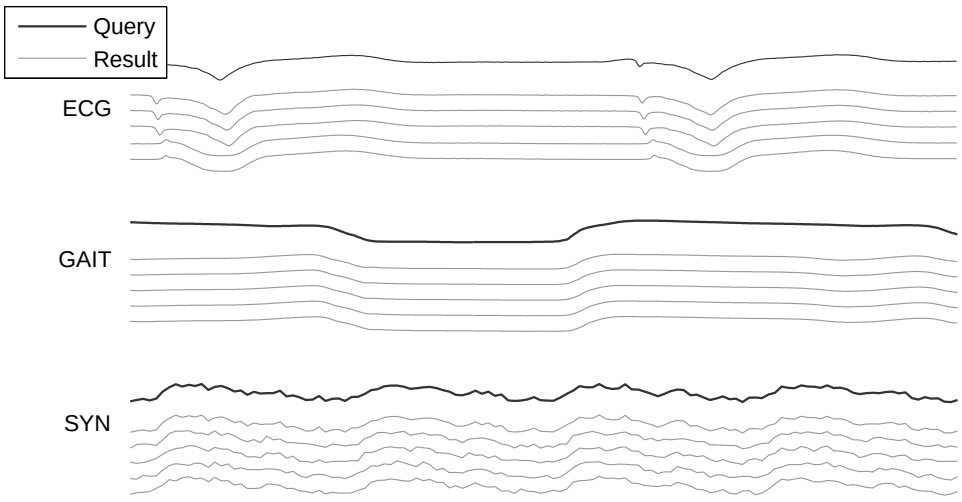

Figure 5.3: Displays three example query results for each dataset using the Monte Carlo approximation. Five segments were randomly displayed from each result set. The greatest variation between segments can be seen in the second column of ECG and the three columns of SYN.

5.4 Results

Example Monte Carlo query results are shown for the ECG, GAIT, and SYN datasets in Fig. 5.3. Five segments are randomly extracted and displayed. Variability can be seen between the query segments and the respective search results. The greatest variation between segments can be seen by the ECG and SYN datasets. This is unlike LSH where each result set contained almost no variability, as shown in Figure 5.4. In fact, LSH consistently retrieves a dataset size of 1 (exact match) for the SYN dataset. The poor performance of LSH for SYN is due to the randomization added during the construction of SYN. This phenomena is important as the randomization of the dataset SYN is similar to the effects of noise. The poor results for SYN demonstrates the degradation of LSH's performance with signal noise. As stated earlier, the value of R can be increased to improve variability to overcome the effects of noise. However, an increase in R increases the algorithmic complexity of LSH

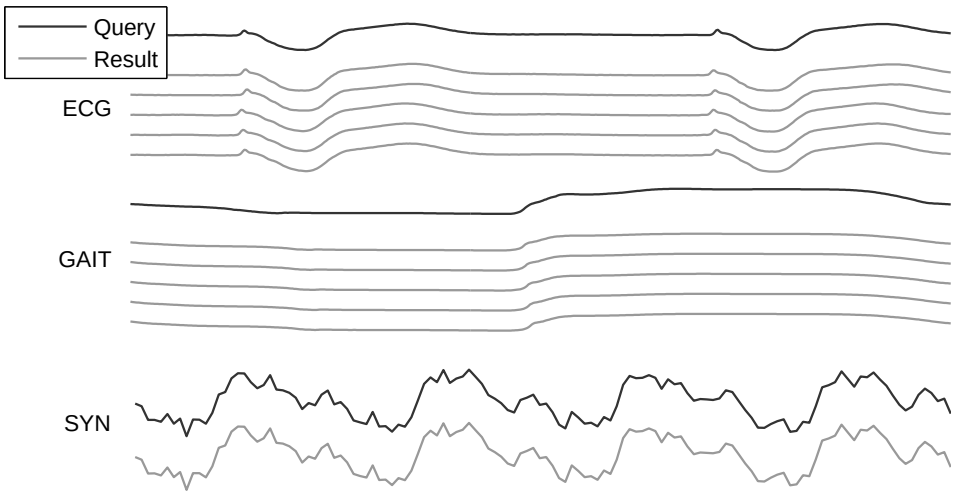

Figure 5.4: Displays example query results for the ECG, GAIT, and SYN datasets using LSH. Five segments were randomly displayed from each result set (when applicable). Almost no variation between segments is observed. The SYN dataset has a result size of one for all queries as shown by the figure.

exponentially (dependent on d).

The percentage increase in the number of results over standard LSH when varying the randomization standard deviation (σ_r) and the number of randomizations (m) is shown in Fig. 5.5. The results for SYN with varying σ_r is shown separately from the ECG and GAIT datasets as the increase in result size is several orders of magnitude larger. The increase over standard LSH ranged from a couple hundred percent to several thousands. This is expected as standard LSH only returns segments that are very similar to the query segment. The larger increase by the GAIT dataset when varying m is caused by the homogeneity of the dataset (i.e., each step in the dataset is largely similar). The randomizations do a good job of expanding the search space. This is extremely important as this same phenomenon would also be seen in other datasets as the number of indexed segments increases.

Increasing σ_r exhibits larger increases for the SYN dataset over standard LSH. A larger

σ_r models a more significant amount of noise. The GAIT and ECG datasets are reasonably clean, and therefore, do not benefit from a larger σ_r. In fact, both the results sets for GAIT and ECG decreased with $\sigma_r > .15$. Larger σ_r cause the search space to extend beyond the neighborhood of the query segment resulting in less matches. The SYN dataset, on the other hand, has a large amount of noise. Hence, a larger σ_r helps improve the size of result sets (as the neighborhood is larger).

The percentage increase over LSH with sampling of result set sizes when varying σ_r and m is shown in Figure 5.6. The ECG and GAIT datasets show an approximate 40% increase when increasing m. However, most of this improvement is seen with less than 50 randomizations. The result sets are filtered such that there are no duplicates or trivial matches. Therefore, adding additional randomizations will cause a large amount of overlap (or duplicates) in the result set yielding smaller gains as m increases. The SYN dataset shows an approximate 100% increase for 250 randomizations. As with an increase in σ_r, an increase in m will improve performance with noisy datasets.

Increasing σ_r shows less of an impact for the SYN dataset when compared to LSH with sampling. This is due to the SYN dataset being composed of the repetition of identical pattern (with the addition of random noise). Therefore, by definition, there is a good probability that these patterns will have colliding hash values. These patterns are filtered with standard LSH (as their distance is greater than R), but LSH with sampling keeps these segments with some probability. The sampling distribution was the same as the Monte Carlo approximations, hence, increasing σ_r also increases the probability that segments with larger distances from the query segment are kept in LSH with sampling.

As stated earlier, the size of a result set is highly dependent upon the respective query segment. A common pattern will yield a large number of results, while an anomaly will yield small sets. Therefore, both Figure 5.5 and Figure 5.6 displayed percentage increases. Figure 5.7 displays raw result set sizes with $m = 25$ and $\sigma_r = 0.1$ for a reference.

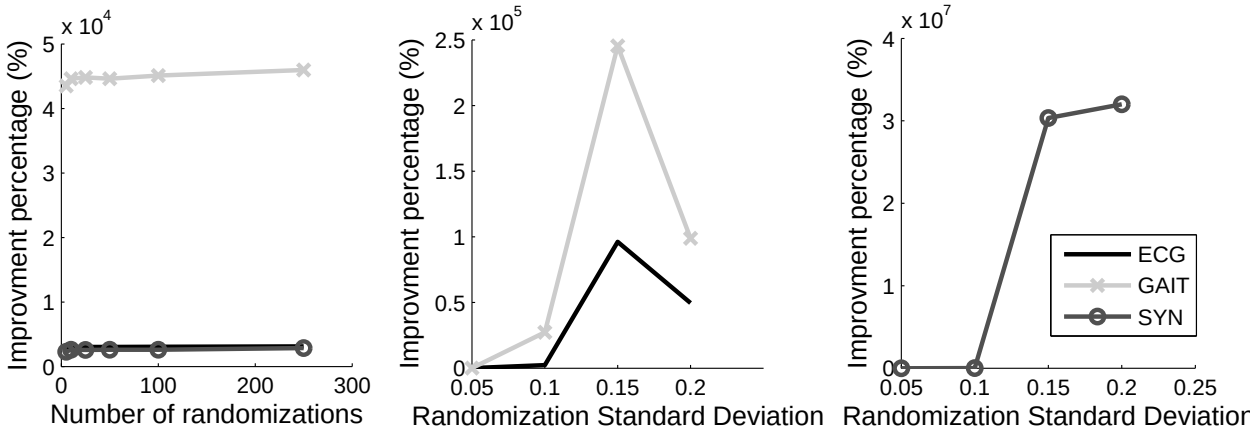

Figure 5.5: Displays the percentage increase in the results sets over standard LSH when varying both m and σ_r. Both a fixed $\sigma_r = .1$ with $m = 5, 10, 25, 100, 250$ and a fixed $m = 25$ with $\sigma_r = .05, .1, .15, .2$ are displayed. The results for SYN with varying σ_r is shown separately from the ECG and GAIT datasets as the increase in result size is several orders of magnitude larger.

The goodness of the result sets for the Monte Carlo approximation was assessed by finding the mean and standard deviation of the Euclidean distance of the query segments to their respective result segments. Fig. 5.8 displays the mean and standard deviation for all three datasets with parameters $\sigma_r = 0.1$ and $m = 5, 10, 25, 100, 250$. The mean and standard deviation converges to the respective theoretical values with the exception of standard deviation for SYN. SYN experiences a slightly higher standard deviation. This is due to the randomization added to the SYN dataset. The randomization of SYN was uniform as opposed to the randomization of query segments, which is based off a Normal distribution, demonstrating the methods robustness to differing distributions. The differing distribution adds a slight disparity to the theoretical and experimental values. However, the standard deviation for the experimental values is still low, and therefore, precise.

The effects of varying σ_r with a fixed m is shown in Figure 5.9. The parameters of the Monte Carlo search are $\sigma_r = .05, .1, .15, .2$ and $m = 25$. The means for all three datasets

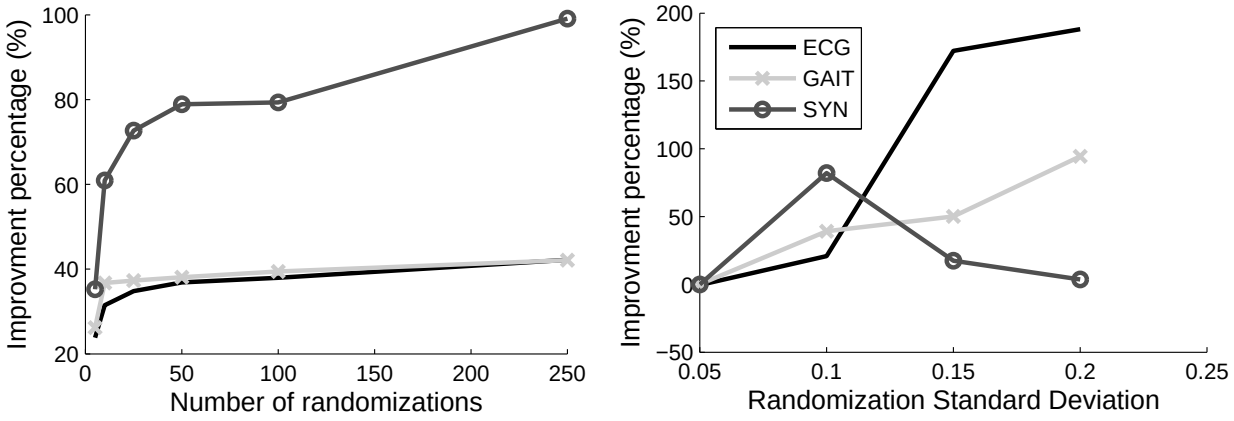

Figure 5.6: Displays the percentage increase in the results sets over LSH with sampling when varying both m and σ_r. Both a fixed $\sigma_r = .1$ with $m = 5, 10, 25, 100, 250$ and a fixed $m = 25$ with $\sigma_r = .05, .1, .15, .2$ are displayed.

were almost identical to their theoretical values. However, the experimental standard deviation showed a slight divergence from the theoretical bound. The ECG and GAIT datasets worsened with a larger σ_r. This can be caused by two factors. First, the database is not large enough, and second, the theoretical standard deviation with large σ_r is bigger than that of the true result set. As stated in [BGR99], good result sets tend to form tight clusters with good separation from the rest of the search space. If the Monte Carlo approximation chooses a σ_r that is too large, then it will search for segments outside of the corresponding true result set cluster. The experimental result set would then contain incorrect matches, consequently raising the standard deviation. The poor results for the SYN dataset is caused by a combination of a large σ_r and the aforementioned effect of randomization.

The experimental values of mean and standard deviation are important to the assessment of the Monte Carlo approximation for two reasons. First, the Monte Carlo approximation should behave similar in practice as it does in theory. Second, it is difficult to find the true mean and standard deviation of a result set until after it has been labelled. Hence, the Monte Carlo approximation must be resilient to a varying mean and standard deviation. The

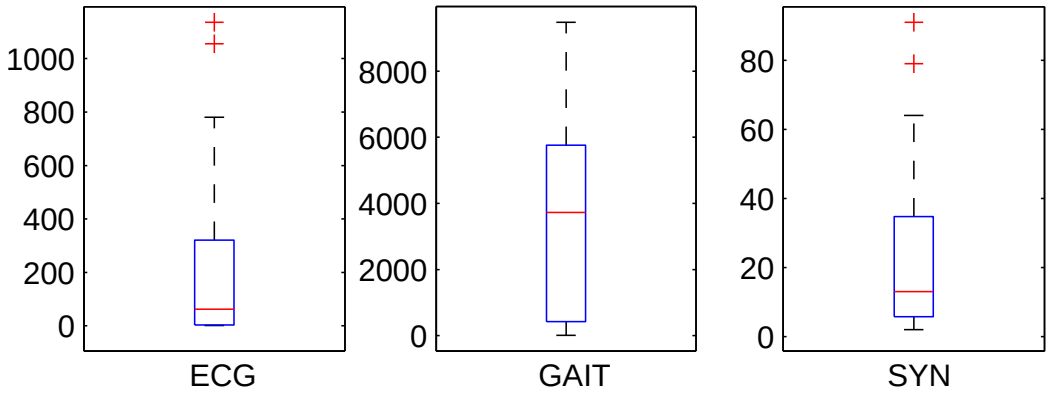

Figure 5.7: Displays the number of results for 50 randomly selected queries with $m = 25$ and $\sigma_r = 0.1$.

experimental standard deviations only showed a slight deviation from the relative theoretical values. Hence, the result sets were still precise.

The runtimes for both standard LSH and the Monte Carlo approximation are given in Fig. 5.10. LSH with sampling was not included in this figure as runtimes for both versions of LSH are near identical. As noted earlier, the computational complexity of LSH is heavily dominated by the number of distance computations. Standard LSH and LSH with sampling only differ in how result sets are pruned, and therefore compute the same number of distance computations.

The implementation of the Monte Carlo approximation was done serially. However, the system could easily be parallelized. The most effective means of improving query times is to parallelize the distance computations. Further improvements could be made by using a distributed database and running the m queries in parallel.

The Monte Carlo approximation is essentially m distinct runs of LSH. Hence, the run times of a Monte Carlo approximation is expected to be m times longer in duration. However, the results in Figure 5.10 are much better. This is due to several optimizations done in the

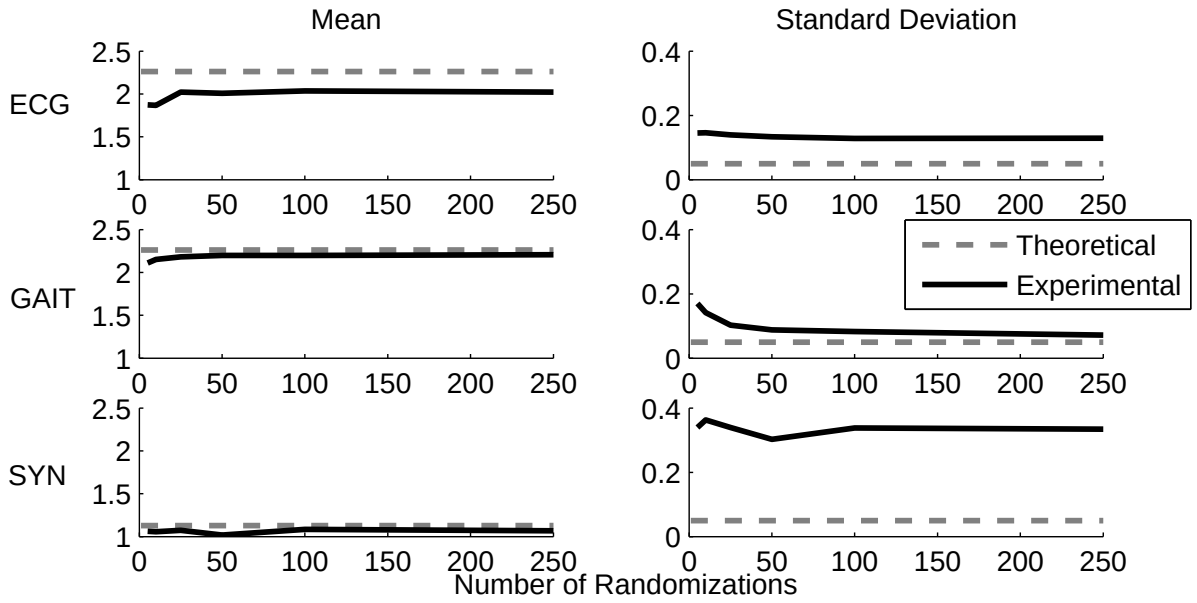

Figure 5.8: Displays the man and standard deviation of the Euclidean distance from query segments to the respective result sets. Queries were run with $\sigma_r = .1$ and $m = 5, 10, 25, 100, 250$. The respective theoretical means and standard deviations are also displayed.

implementation of the method. First, the database relies heavily on caches keeping many hash tables in main memory. In addition, a unique hash is only searched once during a query. For example, if two randomizations produce the same hash function, then that hash function is only searched once. The last major optimization was discussed in Section 5.2.3. When searching on many randomized queries, a large number of duplicate segments are returned in the dataset. Therefore, the computational time can be significantly reduced by only computing one distance computation per distinct segment.

Precision and recall results for the SHAPE dataset is shown in Table 5.3. All tests resulted in a 100% precision score. This is expected as the neighborhoods are known with a radius $R < 1.5$ with high discrimination from other classes. Hence, setting an accurate neighborhood will assure precise results. Recall for LSH with sampling is approximetely

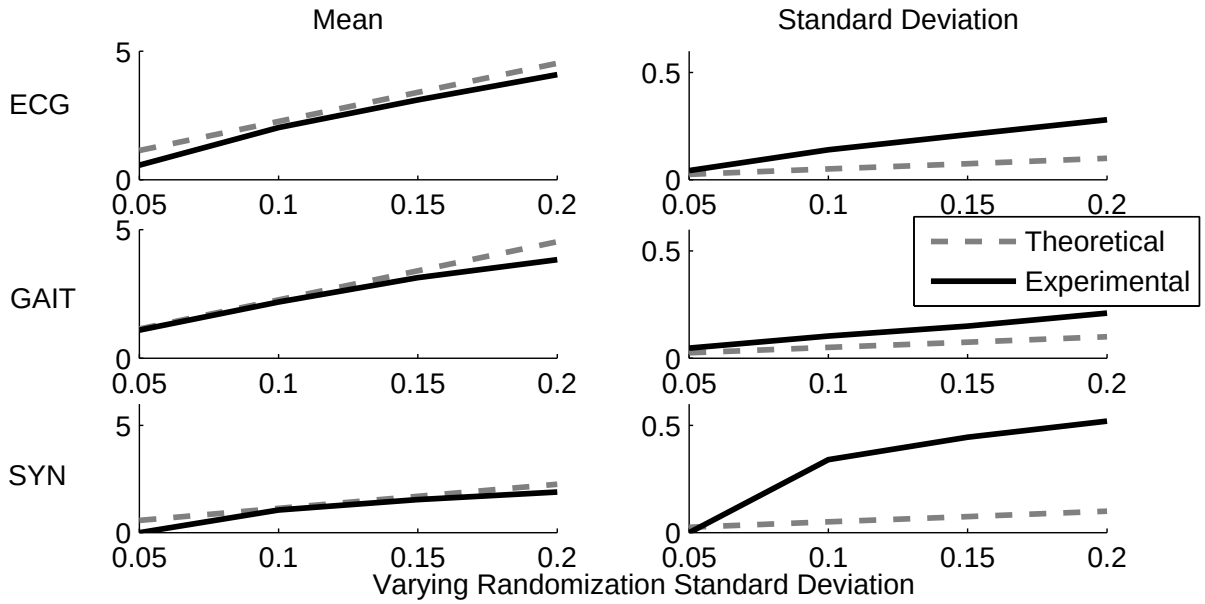

Figure 5.9: Displays the mean and standard deviation of the Euclidean distance from query segments to the respective result sets. Queries were run with $\sigma_r = .05, .1, .15, .2$ and $m = 25$. The respective theoretical means and standard deviations are also displayed.

91% while recall for Monte Carlo search is near 100% with as little as ten randomizations ($m = 10$).

Table 5.3: Precision Recall for LSH with sampling and Monte Carlo Search ($m = 10$)

Method	Precision	Recall
LSH	1.0	0.909089
LSH with sampling	1.0	0.912844
Monte Carlo	1.0	0.999978

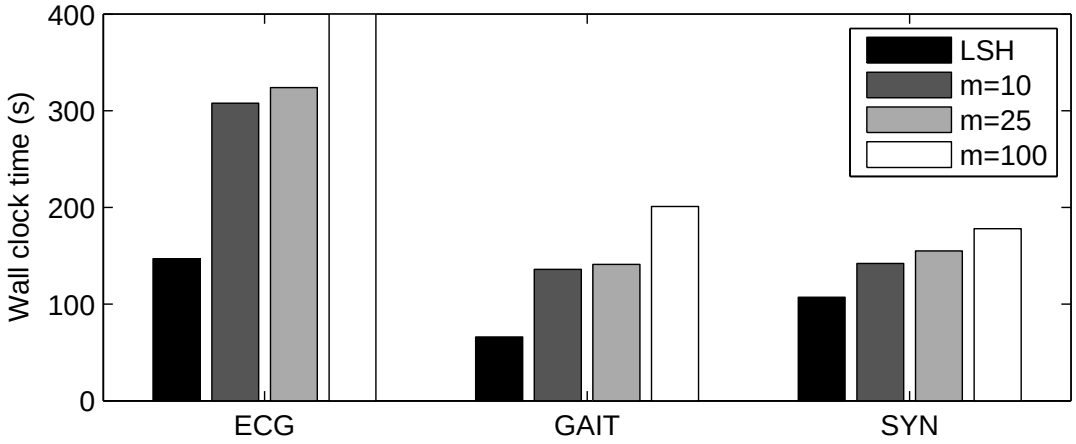

Figure 5.10: Displays the average wall clock time of 50 searches for LSH and the Monte Carlo approximation with $m = 10, 25, 100$. The overhead of the Monte Carlo approximation is minimal (both theoretically and in practice).

5.5 Summary

This chapter presents a Monte Carlo approximation technique for subsequence matching. The number of results for the Monte Carlo approximation are significantly increased over standard Locality Sensitive Hashing (LSH). This technique adds minimal computational complexity and ensures the preciseness of results. The proposed technique takes in a subsequence as input. The subsequence is randomized m times using a normal distribution with standard deviation equal to σ_r. The resulting m randomized subsequences are provided as input to LSH resulting in m distinct R-NN searches. It was theoretically and experimentally shown that the Euclidean distances of a result set to the respective query segment is bounded by a Normal distribution of $Y \sim N(\sigma_r \mu_{\chi_n}, \sigma_r \sigma_{\chi_n})$.

The computational complexity of the Monte Carlo approximation is increased by only a linear factor over LSH (dependent upon the number of randomizations). In practice, however, there was a much smaller decrease in performance. The proposed method ran in

approximately twice the time (wall clock) of LSH with $m = 250$. This method can also be run in parallel to further improve run times.

The Monte Carlo approximation technique was also shown to be resilient under noise, unlike LSH. It was noted that the performance of LSH under noise could be improved by raising the radius in R-NN. However, an increase in the radius results in an exponential increase in the computation complexity (dependent on the number of dimensions d).

CHAPTER 6

Improving Separability of ECG with Domain Features

The success of R-NN indexing schemes, such as LSH, are largely dependent on the their assumed distance measures. Distance measures must provide separability between classes (or sub-classes) in order for R-NN to return meaningful results. This dissertation has focused its attention on the L_2 norm for many reasons listed in Chapter 2. This chapter discusses the inclusion of domain specific features to improve separability of the L_2 norm. This chapter focuses on signals from an electrocardiogram (ECG) as domain specific features of ECG have been significantly studied.

The authors in [DLF11] provide an in-depth survey of features used in classification of heart beats in ECG signals. This chapter has chosen a subset of these features including segmentation intervals [DOR04][MB08] and R-R intervals [DR03][DOR04][TFS05]. These domain specific parameters are well-studied and shown to be very effective in classification of ECG signals.

This chapter utilizes a total of 24 features that are calculated for each heart beat. A detailed description of each of the 24 features is given in Table 6.1. Each heart beat is normalized before calculating morphological features. The MIT-BIH Arrhythmia Database (MITDB) [MM88][GAG00] was used for analysis.

Each feature in Table 6.1 forms a Gaussian distribution and, therefore, can be normalized to a standard normal distribution. The distribution of each feature in Table 6.1 is shown after normalization in Figure 6.1. Each feature shows a near standard normal distribution.

Table 6.1: ECG Feature Descriptions

Feature	Description
*Previous RR	Interval between the R complex of the current heart beat to the R complex of the previous heart beat
Next RR	Interval between the R complex of the current heart beat to the R complex of the following heart beat
Avg. RR Window	Average of the five previous RR intervals and the five following RR intervals
P area	The sum of the absolute value of each time point in the P complex
P max	The maximum value in the P complex
P min	The minimum value in the P complex
*P length	The length of the P complex
QRS area	The sum of the absolute value of each time point in the QRS complex
QRS max	The maximum value in the QRS complex
QRS min	The minimum value in the QRS complex
QRS pos area	The sum of all positive values in the QRS complex
QRS neg area	The absolute value of the sum of all negative values in the QRS complex
QRS std dev	The standard deviation of all points in the QRS complex
QRS skew	The skew of all points in the QRS complex
QRS kurtosis	The kurtosis of all points in the QRS complex
*QRS len	The length of the QRS complex
*QRS QR len	The length of the QR complex
*QRS RS len	The length of the RS complex
T area	The sum of the absolute value of each time point in the T complex
T max	The maximum value in the T complex
T min	The minimum value in the T complex
T len	The length of the T complex
T QT len	The length of the interval between the Q and T complexes
*T ST len	The length of the interval between the Q and T complexes

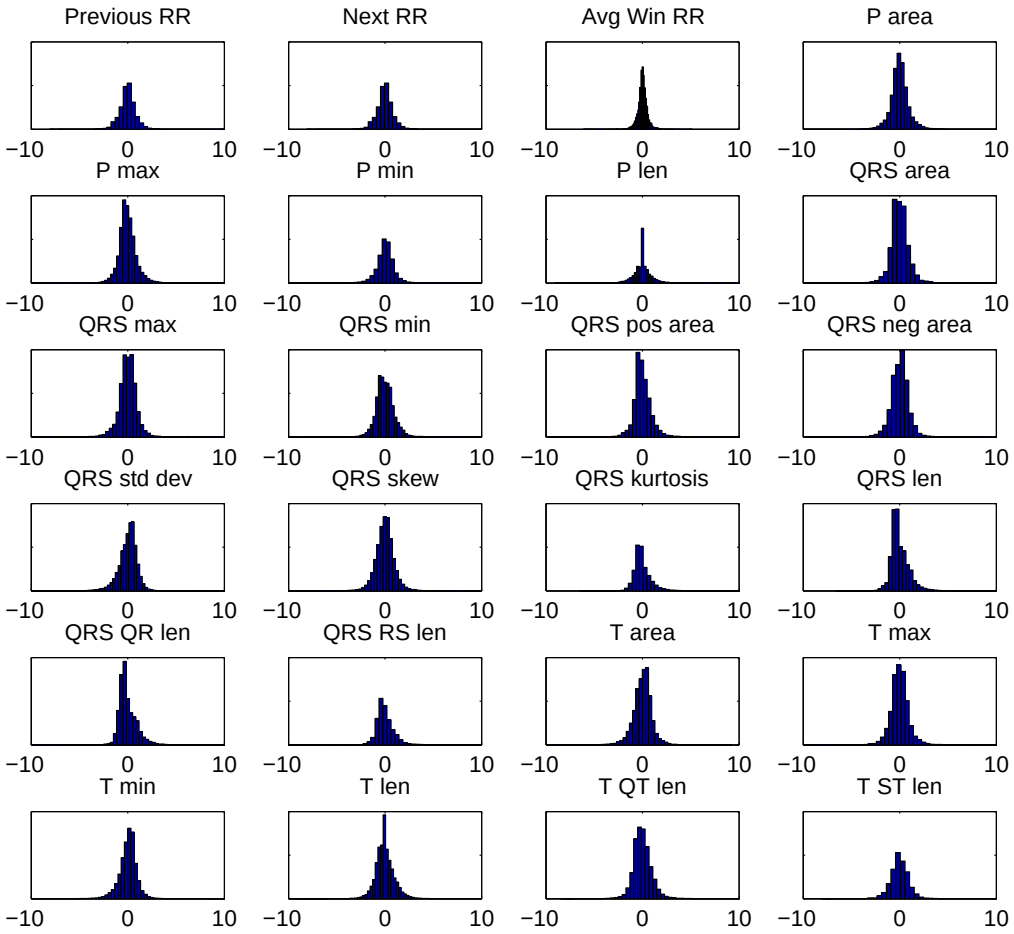

Figure 6.1: Displays the distribution of each feature listed in Table 6.1.

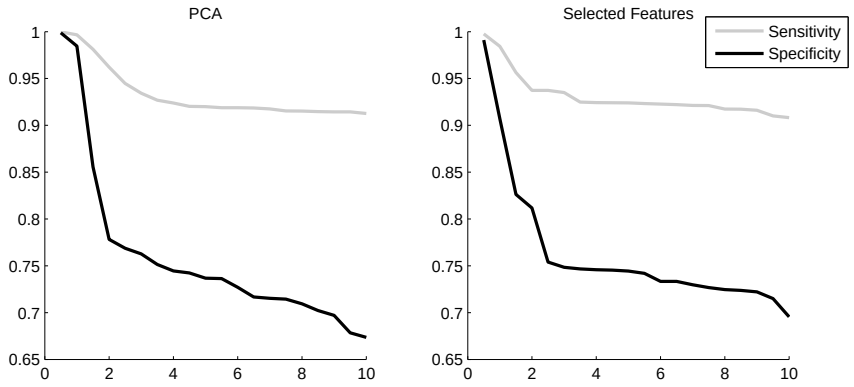

Figure 6.2: Displays sensitivity and specificity when aggregating the PCA reduced and selected features.

Normalization of features causes each feature to have equal weighting when compared using the L_2 norm.

A total of four types (or variations) of feature vectors were assessed in this chapter:

1. **Time Domain:** Segments extracted from the time domain;
2. **Full Feature Vectors:** Vectors composed of the 24 features in Table 6.1;
3. **Principle Component Analysis (PCA) Reduced Vectors:** PCA reduction of the full feature vectors; and
4. **Selected Features:** Vectors composed of the best features from the full feature vectors.

Feature selection was run using correlation-based feature selection [Hal99]. This method is based on the premise that a good subset of features has high correlation to class labels yet low inter-correlation between features. Selected feature vectors were composed of six different features and are denoted in Table 6.1 with an asterisk(*).

Table 6.2: Heart Beat Descriptions

Label	Description	Percent of total database
N	Normal beat	69.9%
V	Premature ventricular contraction	6.4%
R	Right bundle branch block beat	6.4%
L	Left bundle branch block beat	7.6%
/	Paced beat	6.6%
F	Fusion of ventricular and normal beat	1.7%

Table 6.3: Database Parameters

Domain	R	k	L	c	Segment Size
Time Domain	4	4	4	2	512
PCA Reduced Vectors	3	4	4	2	10
Selected Features	1	4	4	2	6

Two experiments were run in this chapter to assess the effectiveness of each feature vector. First, the separability of each type of feature vector was assessed. Separability is calculated by finding the Euclidean distance between all pairs of segments within the same class (intra) and all pairs of segments from different classes (inter).

Second, R-NN searches were performed on the MITDB database to assess precision and recall. Heart beat types were determined by MITDB's annotations and used to calculate precision and recall for each R-NN query. R-NN was performed on all heart beat types that consist of at least 1% of the MITDB database. Any heart beat types that compose less than 1% of the database were ignored as there are not enough instances to draw significant conclusions. A description of each heart beat type that composes at least 1% of the database is described in Table 6.2.

Experiments were run on an LSH database using the aggregation framework in Chapter

4 and the randomization technique in Chapter 5. The full feature vector was omitted for brevity as the PCA presents near equivalent results with a smaller memory footprint (as shown in Section 6.1). The search parameters for each vector type is given in Table 6.3.

The value of R for both the PCA reduced and selected features was chosen by viewing the aggregated sensitivity and specificity plot in Figure 6.2. Similarly, the value of R for the time domain was chosen using the sensitivity and specificity plot in Figure 4.5. Sensitivity and specificity plots were created by aggregating each of the three feature vectors using the aggregation algorithm in Chapter 4. Each feature vector is initially labelled using the dataset's annotations (ground truth labels). Each grouping is labelled using its representative's ground truth label. Sensitivity and specificity are calculated by comparing each feature vector's true label to that of its grouping label. A value of $R = 1$ was chosen for for both the PCA reduced and selected features and a value of $R = 4$ was chosen for the time domain to ensure a minimal number of groupings with high sensitivity and specificity.

6.1 Separability

The separability of the time domain and feature space is shown in Figure 6.3 and Figure 6.4 respectively. The separability of the feature space is shown to be significantly improved over the time domain. In addition, the feature space exhibits a more Gaussian distribution than the time domain.

Principle component analysis (PCA) was performed on the full feature space. Figure 6.5 shows that the top ten components accounts for more than 90% of the variance. Figure 6.6 shows the separability of inter and intra class distances for PCA reduced features. Note that PCA exhibits near identical results to the entire feature space. In addition, PCA consists of only ten principle components resulting in a lower footprint.

While PCA reduces feature vectors by more than 50%, it does not improve separability

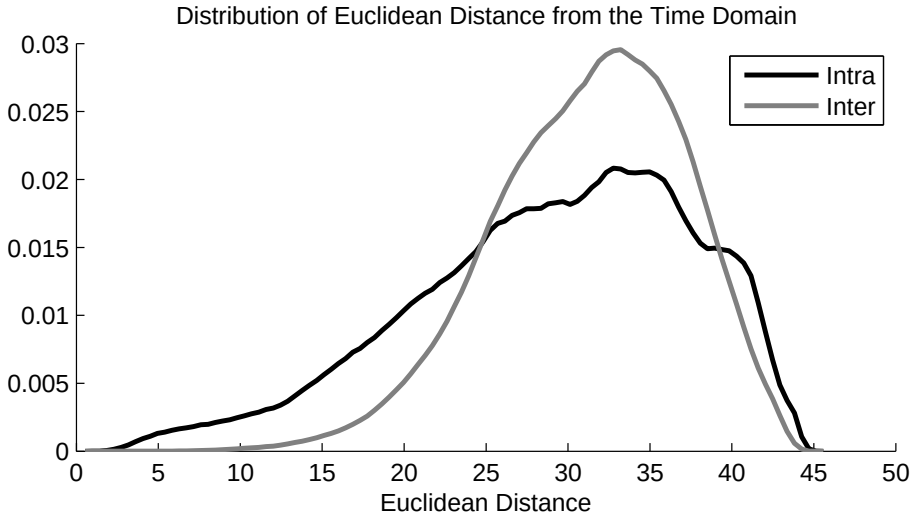

Figure 6.3: Displays the distribution of Euclidean distance for inter and intra class segments in the time domain.

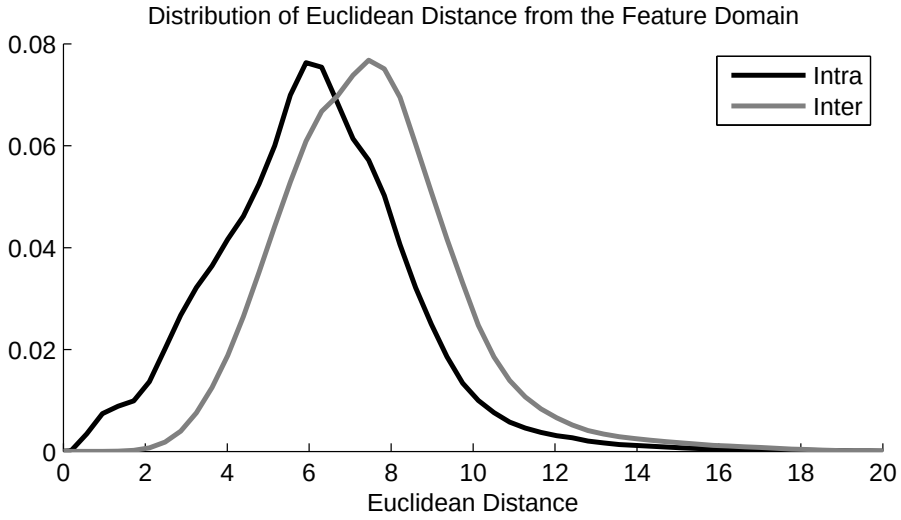

Figure 6.4: Displays the distribution of Euclidean distance for inter and intra class segments in the feature space.

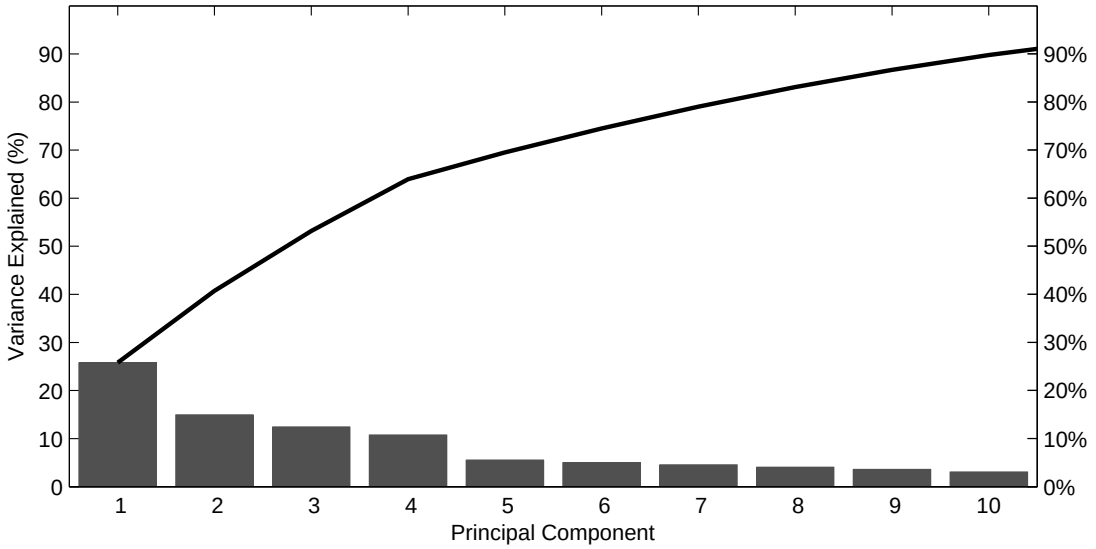

Figure 6.5: Displays the variance explained over the top ten principle components. More than 90% of the variance is explained within the top ten components.

of the feature space. Figure 6.7 shows the separability of using feature selection. Features were reduced to six and resulted in an increase in separability and a significant decrease in dimensionality.

Feature selection exhibits the best separability of all vectors. This phenomena is expected as feature selection allows for the removal of redundancy as well as features that do not significantly contribute to classification. PCA, on the other hand, will reduce redundancy, but does not consider the correlation of each feature to the class labels. The removal of redundant and less correlated features can aid in overcoming the curse of dimensionality in distance comparisons [HKK10]. To note, the removal of less correlated or redundant variables could hurt other classifiers as these variables could be weighted accordingly. For example, a variable could be weighted highly only in a small number of cases where its useful, and otherwise ignored. Such a scheme could be applied to R-NN, however, annotations from MITDB only include a class labels and not sub-class labels. Therefore, an analysis of

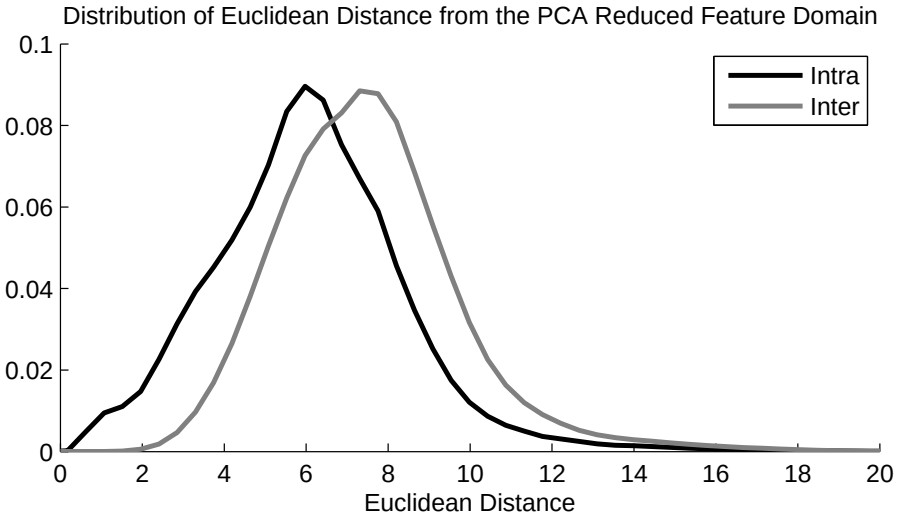

Figure 6.6: Displays the distribution of Euclidean distance for inter and intra class segments in the feature space after PCA reduction.

weighting is left to future work when annotations are sufficient.

Precision-recall and receiver operating characteristic (ROC), as calculated by the separability graphs, is shown in Figure 6.8. The selected feature space shows the best results. The worst results were shown for the time domain with the full feature set and the PCA reduced feature set showing near identical results. As the PCA and full feature set show near identical results, only PCA is considered in the next section.

6.2 R-Nearest Neighbor Search

Figure 6.9 displays precision and recall for all types of heart beats that consist of at least 1% of the MITDB dataset. The average (non-weighted) precision and recall is shown in Figure 6.10. Note that weighting would favor results of normal heart beats yielding an artificially high precision and recall.

Recall is relatively small for all domains. This is due to the fact that R-NN is not

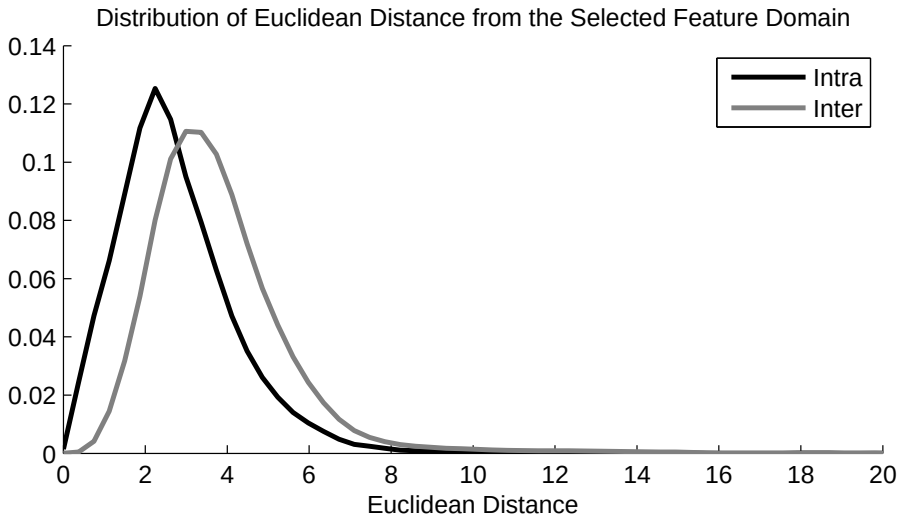

Figure 6.7: Displays the distribution of Euclidean distance for inter and intra class segments in the feature space after running feature selection.

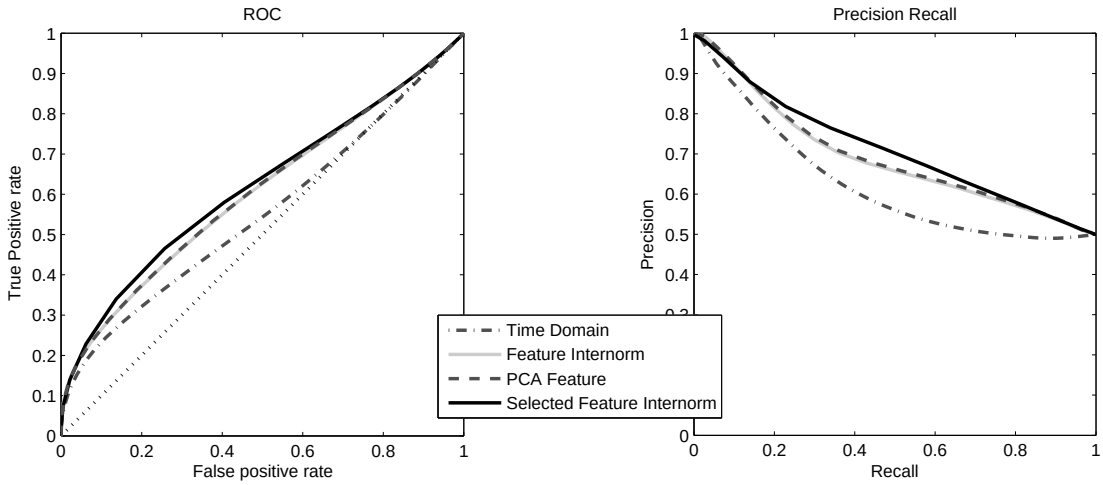

Figure 6.8: Displays ROC curves and Precision Recall curves for ECG segments in the time domain, feature space, selected feature space, and PCA reduced feature space. Note that both the PCA reduced feature space is almost identical to the feature space.

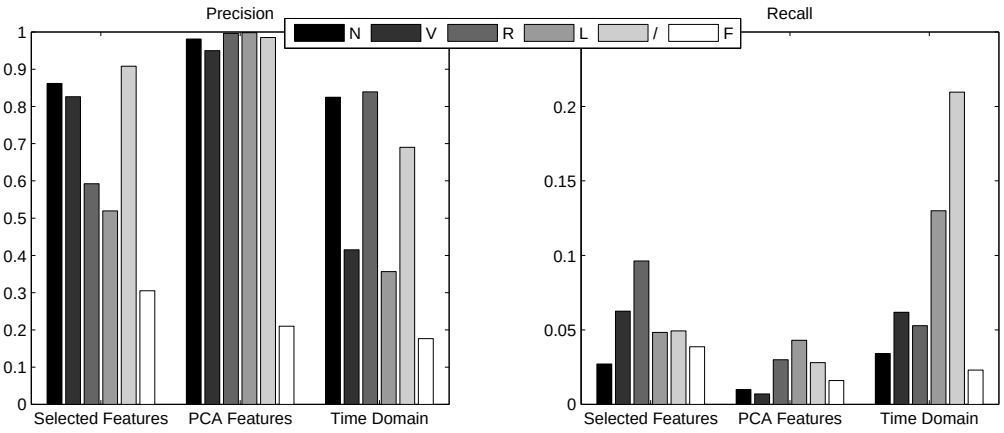

Figure 6.9: Displays precision and recall for all types of heart beats that compose at least 1% of the dataset. Precision and recall is shown for the time domain as well as the PCA reduced features and selected features. Full description of each type of heart beat is given in Table 6.2

designed to find all instances of the same type. Instead, R-NN searches for instances within a subclass. This type of search can be later used in combination with well-annotated data to find trends among sub-classed data.

The PCA reduced features exhibits the best precision and the lowest recall. As shown in Table 6.4, the PCA reduced features also has the largest aggregated index size (in term of objects) and, therefore, it seems an increase in the radius R could increase recall and reduce the index size with a minimal change to precision. The effects of changing from $R = 1$ to $R = 2$ for the PCA reduced features is shown in Figure 6.11. Precision is severely degraded with minimal improvement in recall.

The time domain exhibits the beast overall recall, but the lowest precision. In addition, the aggregated index size is relatively large (only moderately smaller than the PCA reduced feature sets) and the largest memory per query. The selected features exhibit the smallest aggregated index size, the least amount of memory per query with a good precision and

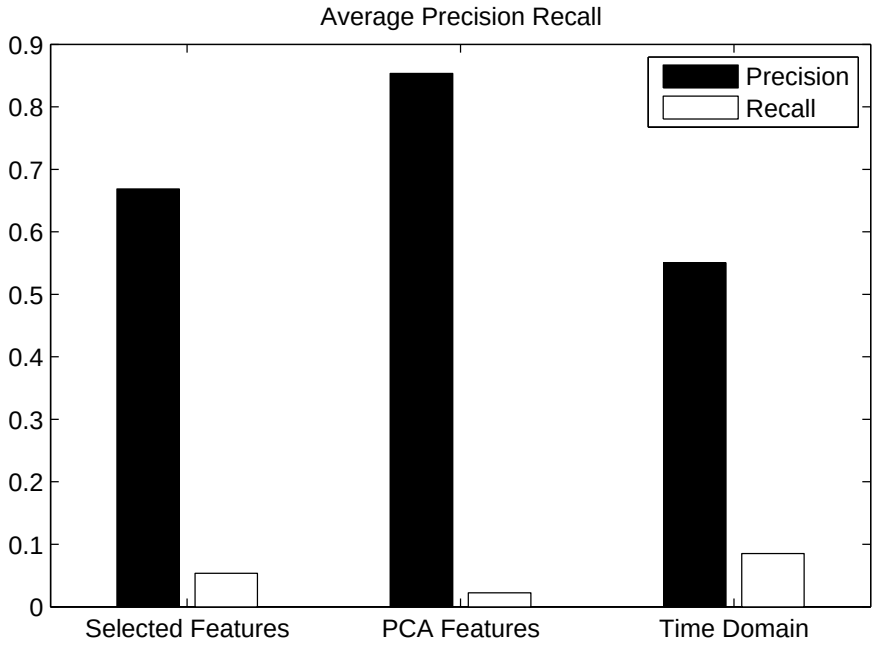

Figure 6.10: Displays the average precision and recall for all types of heart beats that compose at least 1% of the dataset.

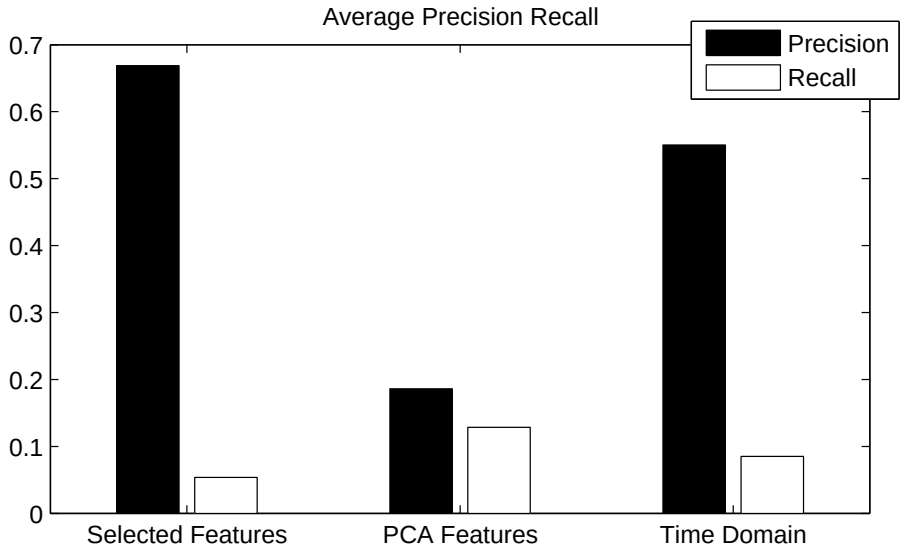

Figure 6.11: Displays the average precision and recall for all types of heart beats that compose at least 1% of the dataset. The selected features and time domain have the same parametrization as 6.10. The PCA reduced feature domain is has $R = 2$.

recall trade off.

A successful search should include diversity in it's result set. Diversity was measured through the number of distinct patients in the result set. MITDB consists of a total of 48 patients. Diversity is given in Table 6.5. The total percentage of signals with each corresponding annotation is given by the last row.

The selected features shows the best overall diversity followed by the time domain. The

Table 6.4: Average Memory Usage

Domain	Memory Per Query	Aggregated Index Size (Instances)	Full Index Size
Selected Features	23.9kB	155.8kB (6,593)	5MB
PCA Features	105.3kB	1515.4kB (37,885)	8MB
Time	5330.5kB	70834.2kB (34,689)	433.4MB

Table 6.5: Search Diversity

Domain	N	V	R	L	/	F	Average
Selected Features	27.8%	18.1%	4.0%	4.6%	8.3%	3.2%	11%
PCA Features	14.0%	3.5%	2.1%	1.3%	1.1%	3.3%	4.22%
Time	25.8%	7.2%	2.7%	5.4%	2.1%	2.1%	7.55%
Total	85.1%	74.5%	10.6%	8.5%	8.5%	40.4%	

least diversity is shown by the PCA reduced feature space. PCA reduces the amount redundancy, but does not account for correlations to class labels. Therefore, PCA will put a large weight on projections with high variability. However, these projections do not necessarily add separation between classes. The selected features, on the other hand, chooses features such that redundancy is reduced while keeping a high correlation to class labels.

Time domain's lack of diversity results from variability in how the signal was collected. Factors such as noise, sensor placement, physiological differences between patients, and misuse can cause variability in biomedical signals. This variability can cause two very similar heart beats to appear very differently in the time domain. Basic normalization was performed to lessen the effect of such variabilities. However, as shown in Table 6.5, the time domain is still brittle to such variability.

Table 6.6 shows the overlap of patients in R-NN searches. Given domain A and B, the percentage of overlap is given as $\frac{A \cap B}{A \cup B}$. The signals returned by the PCA reduced features is a subset of both the time domain and selected features. The diversity for PCA is approximately half of the time domain. The overlap between the two domains is also approximately half, meaning the time domain incorporates most of the diversity as the PCA reduced features. A similar observation is seen for the PCA reduced feature set and the selected feature set.

The overlap between the selected features and the time domain is less than 50%. The time domain shows a diversity of about 75% of the selected features. Therefore, the time

Table 6.6: Result Set Overlap

Intersection	Percentage of Overlap
PCA ∪ Time	49.0
PCA ∪ Selected	35.7
Selected ∪ Time	48.3

domain returns a significant number of signals not returned by the selected features. Hence, it could be beneficial to run the time domain and the selected feature space in parallel to improve diversity.

6.3 Summary

This chapter provided an analysis of domain specific features in R-NN search. Domain specific feature vectors yield higher discriminatory power when comparing classes with the L_2 norm. An in-depth precision/recall analysis was shown using the time domain, PCA reduced feature set, and selected feature set. The selected features showed the best overall results with the lowest memory constraints and the best diversity with a good balance of precision and recall. The PCA reduced feature set exhibited good precision, but had the largest aggregated index size, the lowest recall, and lowest diversity. The time domain and the selected features retrieved result sets that were relatively unique. PCA, on the other hand, produced results that were mostly subsets of the time domain and selected features.

The experiments provided by this chapter are limited by the amount of annotations. Future work would analyze these techniques on datasets with more in depth annotations that include sub-classing (or where sub-classing could be derived). However, such a dataset does not currently exist.

CHAPTER 7

Clinical Trial

Wireless at-home ECG monitoring systems can improve the quality of life of patients with various heart ailments. Authors in [LG06][LGB09] propose a system to monitor ECG signals with a smartphone for early detection of arrhythmias. Authors in [JSC09] use a smartphone based system to monitor and classify if a user is suffering from cardiovascular disease (CVD). Authors in [BJS09] propose the use of ECG devices in a at-home-monitoring system for assessing sleep disorders. These type of systems have the potential to create a large knowledge base of annotated ECG signals. Case-based reasoning (CBR) systems can utilize these knowledge bases with R-NN search to improve both diagnosis and treatment.

Previous chapters introduced various means to improve R-NN search for medical time series signals. These chapters used public datasets that were collected in controlled environments (such as the emergency room). Data collected from wireless at-home monitoring adds variability to ECG signals from various types of noise. For example, patients in a hospital setting (controlled environment) are often stationary while subjects in at-home monitoring systems are generally mobile. Data collected from mobile subjects will contain noise from various physical activities (motion artifacts) that could hurt performance of R-NN searches. This chapter assesses the R-NN techniques presented by this dissertation on data collected from an at-home ECG monitoring clinical trial. The results presented by this dissertation were collected in a pilot study and are preliminary. Data collected from the at-home monitoring system is shown to have similar properties for both aggregation (Chapter 4) and R-NN search (Chapter 6) with limited degradation caused by motion artifacts.

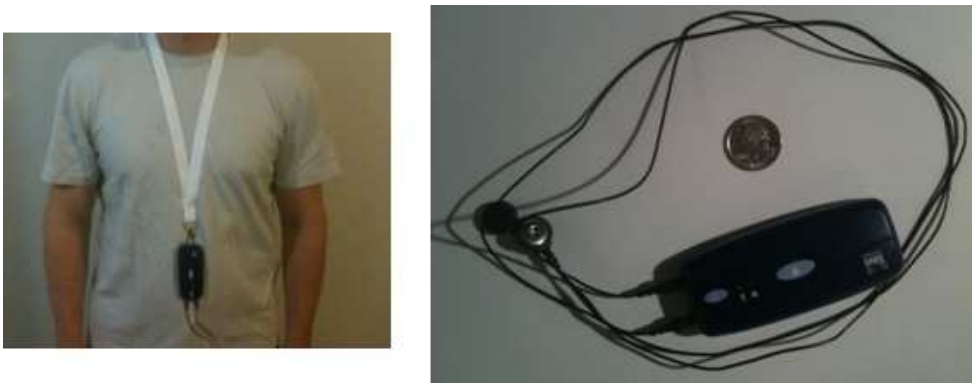

Figure 7.1: Displays the Alive Heart and Activity Monitor by Alive Technologies [Tec12]. The left image shows a subject wearing the device with a lanyard around the neck. The right image shows an up close image of the device with a U.S. quarter for comparison of size.

The clinical trial was run in coordination with the University of California, Los Angeles (UCLA) School of Nursing and was approved through UCLA's Institutional Review Board (IRB). Data was collected from a total of five subjects with congestive heart failure. Subjects were given a wearable ECG device (The Alive Heart and Activity Monitor by Alive Technologies [Tec12] shown in Figure 7.1) that was worn continuously for a period of two days (approximately 48 hours). Data was immediately analyzed by a healthcare professional to locate any heart complications that required immediate medical intervention.

Both aggregation properties and searchability are analyzed in this chapter. In addition, a list of lessons learned is listed for future studies in wearable ECG monitors. Each subject's data was segmented and converted to the the selected feature space as described by Chapter 6. Feature vectors were aggregated using the technique in Chapter 4. The number of groupings was compared to the results in Chapter 4 to show similar rates in reduction. Next, each grouping representative was converted to the feature domain and used as a query

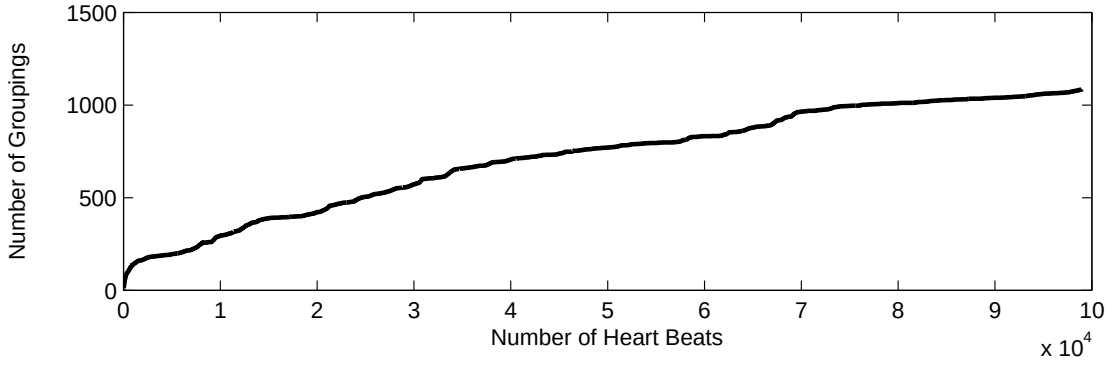

Figure 7.2: Displays the sub-linear growth of groupings for data collected in the clinical trial. Aggregation is accomplished using the algorithm in Chapter 4.

segment for an R-NN search in a database. The search results were used to classify the query segment using majority vote. The results of the queries were analyzed by a domain expert to verify correctness.

The clinical R-NN database is populated with annotated ECG data from the MIT-BIH Arrhythmia Database (MITDB) [MM88][GAG00]. This data was collected from a controlled clinical environment at Boston's Beth Israel Hospital. A description of each significant type of heart beat (comprising at least 1% of the database) is described in Table 6.2.

7.1 Aggregation

The overall reduction in size of data by the aggregation algorithm is shown in Table 7.1. The results are better than those observed in Chapter 4 and Chapter 6. Chapter 4 evaluated the algorithm on datasets of similar size as the clinical trial, however, only the time domain was evaluated. The selected feature domain exhibits better separation (when compared using the L_2 norm) than the time domain, and therefore, will reduce the total number of groupings. Chapter 6 evaluated aggregation on short recordings of ECG signals (approximately 30 minutes). These segments are much shorter than the data from the clinical trial and are

Table 7.1: Reduction by Aggregation

Subject	Total Heart Beats	Total Aggregated Groupings
1	36,391	2303 (6.3%)
2	118,196	1482 (1.3%)
3	98,987	1413 (1.4%)
4	207,008	821 (0.4%)
5	193,412	2468 (1.3%)
Total	653,994	8505 (1.3%)

expected to see less overall reduction in size as the aggregation algorithm exhibits sub-linear growth in groupings (i.e., reductions will increase with the amount of data).

There is a noticeable difference in the number of heart beats. Several subjects had difficulties with the device. Subject 1 had difficulties keeping the electrodes attached to the body due to perspiration and had to continually replace electrodes. Subjects 1-3 all had difficulties with inadvertently turning off the devices, especially while sleeping. Subject 1 also had a noticeably large aggregated size. The larger size was caused by the addition of a large amount of noise caused by electrodes losing connection with the subject. This type of noise is not repetitive (such as ECG signals) and therefore would have poor aggregation properties.

The average growth of groupings versus number of heart beats is shown in Figure 7.2. Note that Subject 1's data was excluded from the graph as it contained a small number of heart beats. A similar sub-linear growth is shown for the clinical trial data as that seen by the datasets used in Chapter 4. There is a quick ramp up period within the first couple thousand instances followed by a levelling off of growth.

Aggregation adds significant value to a home monitoring system for several reasons. First, the amount of work required by medical professionals is significantly reduced for manual

inspection of an ECG signal. In the case of this study, less than 1% of the ECG signal needs inspection. Second, the time required for an R-NN can be reduced. Without aggregation, each heart beat would be used as a query segment into the R-NN. By using aggregation, less than 1% of searches would be required. Only the representative from each grouping would be used as the query segment with results propagating to each feature vector in the respective grouping. Thirdly, as shown in Chapter 6, the search space is significantly reduced improving both run times and memory. ECG signals are aggregated before they are inserted into the index and the search space is restricted to only the representatives of each grouping created by the aggregation.

7.2 R-NN search results

Results for R-NN search is shown in Table 7.2. The first two rows show precision and recall for both abnormal and normal heart beats. Normals show the best precision and recall. This is expected as normals consist of 70% of the database (training set) and offer a larger, more diverse search space (as almost every patient in the training set contributes at least some normal heart beats). Abnormal heart beats show a lower precision and recall. This is largely due to a lack of diversity and instances in the database.

To note, precision in abnormalities is not as important as recall. An R-NN search that denotes a heart beat as abnormal should be verified by a medical professional. For example, in a case-based reasoning system, search results may include detailed annotations (or summary of annotations) that can help a medical professional make a more informed decision about abnormality. In this case, it is better to see a few normals marked as abnormal, than to miss abnormalities.

Premature Ventricular Contraction (PVC) and Right Bundle Branch Block (RBBB) were the two most significant abnormal heart beats seen by the clinical trial. Recall for PVC and

Table 7.2: R-NN Precision and Recall

Type	Precision	Recall
Normal	.986	.983
Abnormal	.733	.762
Premature Ventricular Contraction	.846	.650
Right Bundle Branch Block	.685	.913

precision for RBBB was relatively low. These lower numbers, as well as the overall low precision and recall numbers for abnormal heart beats were caused by motion artifacts as well as a small training size.

Motion artifacts are the most difficult to detect and remove since their spectrum overlaps that of ECG signals [RS02]. These artifacts cause errors when calculating the ECG features listed in Table 6.1. For example, calculating the proper location of the p-wave is extremely important for many features. The p-wave went undetected in 29.2% of all heart beats (using the algorithm in [LJC94]). This is as opposed to the MITDB database where the p-wave went undetected in only 4.5%. Removing results from signals that include more than 20% missing p-waves raises the overall precision and recall of abnormal heart beats to 83.67% and 81.8% respectively.

There are two solutions to the problem of motion artifacts. First, regions present with motion artifacts can be ignored from the aggregation and search. Authors in [KKO07] [OKJ08] [LMM11] have proposed algorithms to detect the presence of motion artifacts. Second, motion artifacts could be removed leaving the original ECG signal intact [RS02] [TBH02]. However, such algorithms have been shown to be ineffective with high amounts of noise [WA07]. Hence, removing motion artifacts from ECG signals is still an open problem.

The training size should be increased to improve the accuracy of the R-NN system. The MITDB database only consisted of 48 patients each with 30 minute recordings. Increasing

the training set is extremely time consuming as the proper annotations need to be made manually. In addition, a much richer annotation set is needed to better assist a case-based reasoning system. The system presented by this dissertation can be used to help build such a training set through use of clinical trials. The initial database can be populated with datasets such as MITDB. As data is collected through clinical trials, it can be fully annotated (including patient history) by health care professionals. This data can be inserted into the database to help in future studies.

7.3 Weaknesses of Clinical Study

Three main weaknesses were observed during the clinical study that should be addressed in future studies. First, only two leads were available by the ECG device that results in one output channel of measured ECG traces. Providing an additional lead gives medical professionals a more complete picture of how the heart is functioning, making diagnosis more certain. Second, three patients noted that the device was difficult to wear at night. An improved device would affix directly to an individual's chest, removing the need to wear a lanyard. Thirdly, three patients had difficulties keeping the electrodes affixed to their skin. Sweating often caused the electrodes to become loose. The study was done in the summer months of the Los Angeles area with subjects that often had lower incomes. Many of the patients did not have air conditioning.

7.4 Summary

This chapter assessed the *R*-NN system proposed by this dissertation on data collected from an at-home ECG monitoring clinical trial. Data was collected from patients with congestive heart failure. Both aggregation were similar to those reported by previous chapters. Precision and recall for *R*-NN was decreased when using clinical trial data. Much of this

degradation was caused by motion artifacts that were not present in data sets used by previous chapters. Motion artifacts hinder segmentation engines causing inaccurate calculation of domain specific features. In addition, the R-NN database would also benefit from a larger database size with a more rich annotation set. No such public dataset currently exists. However, the R-NN system proposed by this dissertation could be used to generate such a dataset.

CHAPTER 8

Conclusion

This dissertation presented a paradigm to improve R-NN searches for the medical time series domain. This dissertation made the following four main contributions:

1. Extraction of query segments (Chapter 3);
2. Improvement in the quality of search results (Chapter 5); and
3. Optimization of index structures (Chapter 4); and
4. An in-depth analysis of feature domains for ECG signals (Chapter 6).

In addition, data was collected from a clinical trial to show the feasibility of the proposed R-NN search paradigm using data collected through an at-home system. The majority of the experiments in this dissertation were performed on an LSH-based database. LSH was chosen due to its proven sub-linear computational and memory complexities. This is unlike other R-NN solutions such as spatial indexes. However, the methods proposed in this dissertation are applicable across any R-NN search engine.

There are two main stages when interacting with an R-NN database: database population and database search. Database population consists of the following three steps:

1. Segmentation (Including feature extraction when applicable);
2. Aggregation; and

3. R-NN Indexing (e.g., LSH, spatial indexing, etc...).

Database search consists of the following three steps:

1. Segmentation (Including feature extraction when applicable);
2. Randomization; and
3. R-NN.

Proper segmentation is largely specific to the domain. Well-known and defined domains, such as ECG, can be segmented in a domain specific fashion. In general, ECG defines segments to be centered around the QRS complex. Other domains, such as wearable accelerometers or gyroscopes, consist of much less defined models. The number of activities performed by an individual wearing such sensors is endless. Therefore, a common domain specific segmentation technique is unlikely to exist unless the environment is severely constrained (e.g., the users only walk or stand). Naïve approaches, such as sliding window segmentation, introduce redundancies that severely hurt the performance of search and data mining tasks [WLS11][KL05]. Therefore, there is limited use of such naïve approaches in this dissertation.

Salient Segmentation is shown to be extremely useful for domains with minimally defined models. Salient Segmentation offers a probabilistic method of finding segments that are unlikely to occur. Its success stems from the following three properties:

1. All salient patterns are segmented;
2. All salient patterns are segmented consistently (i.e., alignment); and
3. Near linear algorithmic complexity

When searching, segments can be represented in any domain (such as time and feature space) that is deemed useful to the search task. As shown in Chapter 6, the feature space can significantly improve precision and recall while reducing the overall memory usage of R-NN.

Aggregation was accomplished using an online algorithm similar to the online clustering algorithm presented in [Mey01]. Segments are first clustered to small tight groupings. Each grouping determines one representative segment and only representative segments are inserted into the index. This algorithm is based on the finding that classes of objects are composed of tight Gaussian distributed sub-classes [BFG99][HKK10]. Hence, each grouping determined by the algorithm tends to be of the same class. The work presented in Chapter 4 show that the number of groupings grows sub-linearly with the total number of objects thereby producing a scalable solution.

Randomization was presented as a means to significantly improve the recall of R-NN searches without decreasing precision. The randomization presented in Chapter 5 was shown to create Gaussian distributed search neighborhoods. The success of this algorithm is based on the finding that sub-classes for tight Gaussian distributed neighborhoods in regards to their Euclidean distances.

This dissertation focused on the use of the L_2 norm for comparing segments. Elastic measures, such as DTW, were ignored as these do not always translate to multiple domains (from time to feature space). In addition, elastic measures exhibit quadratic run times unlike the L_2 norm.

The overall paradigm presented by this dissertation represents a time series database for medical time series signals. Significant improvements in recall were observed with no loss in precision. In addition, memory was significantly reduced by limiting the size of the search space (in a scalable sub-linear fashion). Much of this work was tested on real-world and publicly available datasets. In addition, a small clinical trial was run to collect data from an

at-home monitoring system. These datasets were used to assess the R-NN paradigm with noisy data common with at-home monitoring systems.

8.1 Future Work in Cased-Based Reasoning

The solution presented by this dissertation enables medical time series queries on extremely large databases consisting of millions or billions of rows. Such a database can be used to store time series data collected by wearable at-home monitoring systems [CLC08] [JSC09] [NAH10] [SCW11]. Such systems have the potential to create huge datasets. In addition, these systems are based on a centralized monitoring authority that can provide detailed annotations about the patients.

Medical case-based reasoning (CBR) is a well studied area for medical diagnosis and treatments [BAP02][BKS98]. These systems rely on data mining and machine learning techniques to derive decisions on new cases based on information gathered from a knowledge base of previous cases. Knowledge bases are composed of various types of subjective data (such as monographs, scientific literature, clinical practice guidelines, pathways and cases [BKS98]) and, to a lesser extent, objective data (such as ECG [NFS03] and images [GA96][Per01]). One major drawback of CBR systems is that the knowledge base must contain relevant cases to correctly make decisions on a current case [NS04]. This is especially difficult for measured signals, such as ECG and accelerometers, that exhibit high variability due to noise, sensor displacement, misuse, and other factors that are difficult to control. Therefore, any CBR using medical time series data must be sufficiently large. Wearable computing for at-home monitoring is a means to populate such databases with a sufficient amount of data and respective annotations.

This dissertation (and previous works) often points to sub-classes. However, this is a difficult area to address without proper analysis of sufficiently large and annotated data.

Currently, these datasets do not exist. Future work should generate large public datasets collected from at-home monitoring clinical studies. This data can then be mined (both text and raw signal data) to distinguish sub-classes. These subclasses can then be analyzed for correlations that could improve the ability to diagnose and treat patients. The measured raw signals add a completely objective dimension. In addition, raw signal data can be collected any time and anywhere with wearable devices. Such data will provide a more complete description of the patient unlike data collected in clinical settings. Finally, wearable sensing devices can provide a much more economical solution to collecting data as opposed to collecting data in a clinical setting [PB10].

8.2 Future work in System Scaling

This dissertation presented a scalable solution to medical time series search. However, more work is needed in further scaling the database to enable extremely fast queries on ever increasing database sizes. This dissertation focused on a primarily centralized database. Improvements to query times can be improved through distributed and peer-to-peer databases [OV11] [BCO08]. Future work would investigate algorithms to populate data in an efficient and predominately online manner. In addition, randomization and pruning would need to be modified in order to efficiently run in a distributed manner.

8.3 Future Work in ECG Filtering

Noise due to motion artifacts caused the most significant degradation in precision and recall when performing R-NN on data collected from at-home monitoring systems. Unfortunately, motion artifacts are the most difficult to detect and remove since their spectrum overlaps that of ECG signals [RS02]. Signal databases must account for such noise in ECG signals. Several previous works have focused solely on the detection of noise [KKO07][OKJ08][LMM11].

These works find regions within the ECG signal that are severely degraded by motion artifacts. These regions are then extracted and ignored for future mining and search.

Other works attempt using finite impulse response (FIR) filters to remove noise from motion artifacts leaving the pure ECG signal in tact [RS02][TBH02]. Authors in [WA07] observed that FIR filters require a large amount of parametrization to be effective. They show that many systems do not contain enough data to effectively use a FIR filter under moderate noise conditions. They propose the use of a Laguerre model to remove noise. However, this model is only shown to be effective in photoplethysmograms and not ECG.

Removing motion artifacts will enable an R-NN system to be more efficient as there will be limited loss of important information. This is especially true for systems analyzing data collected from at-home monitoring. Currently, this is still an open problem.

References

[AHK01] C. Aggarwal, A. Hinneburg, and D. Keim. "On the surprising behavior of distance metrics in high dimensional space." *Database TheoryICDT 2001*, pp. 420–434, 2001.

[BAP02] J. Baumeister, M. Atzmueller, and F. Puppe. "Inductive learning for case-based diagnosis with multiple faults." *Advances in case-based reasoning*, pp. 173–216, 2002.

[BCO08] A. Bonifati, P.K. Chrysanthis, A.M. Ouksel, and K.U. Sattler. "Distributed databases and peer-to-peer databases: past and present." *ACM SIGMOD Record*, **37**(1):5–11, 2008.

[BEF06] A. Broder, N. Eiron, M. Fontoura, M. Herscovici, R. Lempel, J. McPherson, R. Qi, and E. Shekita. "Indexing shared content in information retrieval systems." *Advances in Database Technology-EDBT 2006*, pp. 313–330, 2006.

[BEK98] S. Berchtold, B. Ertl, D.A. Keim, H.P. Kriegel, and T. Seidl. "Fast nearest neighbor search in high-dimensional space." In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pp. 209–218. IEEE, 1998.

[BFG99] K.P. Bennett, U. Fayyad, and D. Geiger. "Density-based indexing for approximate nearest-neighbor queries." In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 233–243. ACM, 1999.

[BFR98] P.S. Bradley, U. Fayyad, and C. Reina. "Scaling EM (expectation-maximization) clustering to large databases." *Microsoft Research Report, MSR-TR-98-35*, 1998.

[BGR99] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. "When is "nearest neighbor" meaningful?" *Database Theory-ICDT'99*, pp. 217–235, 1999.

[BI04] L. Bao and S. Intille. "Activity recognition from user-annotated acceleration data." *Pervasive Computing*, pp. 1–17, 2004.

[BJS09] J. Biswas, M. Jayachandran, L. Shue, K. Gopalakrishnan, and P. Yap. "Design and Trial Deployment of a Practical Sleep Activity Pattern Monitoring System." *Ambient Assistive Health and Wellness Management in the Heart of the City*, pp. 190–200, 2009.

[BKS90] N. Beckmann, H.P. Kriegel, R. Schneider, and B. Seeger. *The R*-tree: an efficient and robust access method for points and rectangles*, volume 19. ACM, 1990.

[BKS98] I. Bichindaritz, E. Kansu, and K. Sullivan. "Case-based reasoning in care-partner: Gathering evidence for evidence-based medical practice." *Advances in case-based reasoning*, pp. 334–345, 1998.

[BMP03] A.Y. Benbasat, S.J. Morris, and J.A. Paradiso. "A wireless modular sensor architecture and its application in on-shoe gait analysis." In *Sensors, 2003. Proceedings of IEEE*, volume 2, pp. 1086–1091. IEEE, 2003.

[CGK06] I. Christov, G. Gómez-Herrero, V. Krasteva, I. Jekova, A. Gotchev, and K. Egiazarian. "Comparative study of morphological and time-frequency ECG descriptors for heartbeat classification." *Medical engineering & physics*, **28**(9):876–887, 2006.

[CKL03] B. Chiu, E. Keogh, and S. Lonardi. "Probabilistic discovery of time series motifs." In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 493–498. ACM, 2003.

[CLC08] D.K. Cho, S.H. Lee, A. Chang, T. Massey, C.W. Chang, M.H. Tsai, M. Sarrafzadeh, and M. Gerla. "Opportunistic medical monitoring using bluetooth P2P networks." In *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, pp. 1–6. IEEE, 2008.

[CN04] Y. Cai and R. Ng. "Indexing spatio-temporal trajectories with Chebyshev polynomials." In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pp. 599–610. ACM, 2004.

[Den05] A. Denton. "Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model." In *Data Mining, Fifth IEEE International Conference on*, pp. 8–pp. IEEE, 2005.

[DII04] M. Datar, N. Immorlica, P. Indyk, and V.S. Mirrokni. "Locality-sensitive hashing scheme based on p-stable distributions." In *Proceedings of the twentieth annual symposium on Computational geometry*, pp. 253–262. ACM, 2004.

[DLF11] G. Doquire, G. de Lannoy, D. François, and M. Verleysen. "Feature selection for supervised inter-patient heart beat classification." *Computational Intelligence and Neuroscience*, **2011**(643816):1–9, 2011.

[DOR04] P. De Chazal, M. O'Dwyer, and R.B. Reilly. "Automatic classification of heartbeats using ECG morphology and heartbeat interval features." *Biomedical Engineering, IEEE Transactions on*, **51**(7):1196–1206, 2004.

[DP73] D.H. Douglas and T.K. Peucker. "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature." *Cartographica: The International Journal for Geographic Information and Geovisualization*, **10**(2):112–122, 1973.

[DR03] P. De Chazal and RB Reilly. "Automatic classification of ECG beats using waveform shape and heart beat interval features." In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 2, pp. II–269. Ieee, 2003.

[Elk03] C. Elkan. "Using the triangle inequality to accelerate k-means." In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, volume 20, p. 147, 2003.

[FRM94] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. "Fast subsequence matching in time-series databases." In *Proceedings of the 1994 ACM SIGMOD international conference on Management of data*, SIGMOD '94, pp. 419–429, New York, NY, USA, 1994. ACM.

[FWV07] D. Francois, V. Wertz, and M. Verleysen. "The concentration of fractional distances." *Knowledge and Data Engineering, IEEE Transactions on*, **19**(7):873–886, 2007.

[GA96] M. Grimnes and A. Aamodt. "A two layer case-based reasoning architecture for medical image understanding." *Advances in Case-Based Reasoning*, pp. 164–178, 1996.

[GAG00] A.L. Goldberger, L.A.N. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.K. Peng, and H.E. Stanley. "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals." *Circulation*, **101**(23):e215–e220, 2000.

[GIM99] A. Gionis, P. Indyk, and R. Motwani. "Similarity search in high dimensions via hashing." In *Proceedings of the International Conference on Very Large Data Bases*, pp. 518–529, 1999.

[HAK00] A. Hinneburg, C.C. Aggarwal, and D.A. Keim. "What Is the Nearest Neighbor in High Dimensional Spaces?" In *Proceedings of the 26th International Conference on Very Large Data Bases*, pp. 506–515. Morgan Kaufmann Publishers Inc., 2000.

[Hal99] M.A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[HKK10] M. Houle, H.P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. "Can shared-neighbor distances defeat the curse of dimensionality?" In *Scientific and Statistical Database Management*, pp. 482–500. Springer, 2010.

[HLC00] J.M. Hausdorff, A. Lertratanakul, M.E. Cudkowicz, A.L. Peterson, D. Kaliton, and A.L. Goldberger. "Dynamic markers of altered gait rhythm in amyotrophic lateral sclerosis." *Journal of applied physiology*, **88**(6):2045–2053, 2000.

[HLY07] M. Herscovici, R. Lempel, and S. Yogev. "Efficient indexing of versioned document sequences." *Advances in Information Retrieval*, pp. 76–87, 2007.

[HMF97] J.M. Hausdorff, S.L. Mitchell, R. Firtion, CK Peng, M.E. Cudkowicz, J.Y. Wei, and A.L. Goldberger. "Altered fractal dynamics of gait: reduced stride-interval correlations with aging and Huntingtons disease." *Journal of Applied Physiology*, **82**(1):262, 1997.

[HS94] J. Hershberger and J. Snoeyink. "An O (n log n) implementation of the Douglas-Peucker algorithm for line simplification." In *Proceedings of the tenth annual symposium on Computational geometry*, pp. 383–384. ACM, 1994.

[IM98] P. Indyk and R. Motwani. "Approximate nearest neighbors: towards removing the curse of dimensionality." In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613. ACM, 1998.

[Int11] Intel. "2nd Generation Intel® Core™ Processor Family Desktop Datasheet.", 2011.

[JSC09] Z. Jin, Y. Sun, and A.C. Cheng. "Predicting cardiovascular disease from real-time electrocardiographic monitoring: An adaptive machine learning approach on a cell phone." In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 6889–6892. IEEE, 2009.

[KCP01] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. "Dimensionality reduction for fast similarity search in large time series databases." *Knowledge and information Systems*, **3**(3):263–286, 2001.

[KHO02] B.U. Kohler, C. Hennig, and R. Orglmeister. "The principles of software QRS detection." *Engineering in Medicine and Biology Magazine, IEEE*, **21**(1):42–57, 2002.

[KKO07] Y. Kishimoto, Y. Kutsuna, and K. Oguri. "Detecting motion artifact ECG noise during sleeping by means of a tri-axis accelerometer." In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pp. 2669–2672. IEEE, 2007.

[KL05] E. Keogh and J. Lin. "Clustering of time-series subsequences is meaningless: implications for previous and future research." *Knowledge and Information Systems*, **8**(2):154–177, 2005.

[KPC04] S.W. Kim, S. Park, and W.W. Chu. "Efficient processing of similarity search under time warping in sequence databases: an index-based approach." *Information Systems*, **29**(5):405–420, 2004.

[KR05] E. Keogh and C.A. Ratanamahatana. "Exact indexing of dynamic time warping." *Knowledge and information systems*, **7**(3):358–386, 2005.

[KS97] N. Katayama and S. Satoh. "The SR-tree: An index structure for high-dimensional nearest neighbor queries." In *ACM SIGMOD Record*, volume 26, pp. 369–380. ACM, 1997.

[LG06] P. Leijdekkers and V. Gay. "Personal heart monitoring system using smart phones to detect life threatening arrhythmias." In *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, pp. 157–164. IEEE, 2006.

[LGB09] P. Leijdekkers, V. Gay, and E. Barin. "Trial Results of a Novel Cardiac Rhythm Management System Using Smart Phones and Wireless ECG Sensors." *Ambient Assistive Health and Wellness Management in the Heart of the City*, pp. 32–39, 2009.

[Liu06] T. Liu. "Fast nonparametric machine learning algorithms for high-dimensional massive data and applications." Technical report, DTIC Document, 2006.

[LJC94] P. Laguna, R. Jané, P. Caminal, et al. "Automatic detection of wave boundaries in multilead ECG signals: Validation with the CSE database." *Computers and biomedical research*, **27**(1):45–60, 1994.

[LJW07] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. "Multi-probe LSH: efficient indexing for high-dimensional similarity search." In *Proceedings of the 33rd international conference on Very large data bases*, pp. 950–961. VLDB Endowment, 2007.

[LMM11] J. Lee, D. McManus, S. Merchant, and K. Chon. "Automatic Motion and Noise Artifact Detection in Holter ECG Data using Empirical Mode Decomposition and Statistical Approaches." *Biomedical Engineering, IEEE Transactions on*, (99):1–1, 2011.

[LP02] J.L.E.K.S. Lonardi and P. Patel. "Finding motifs in time series." In *Proc. of the 2nd Workshop on Temporal Data Mining*, pp. 53–68, 2002.

[LPB00] M. Lagerholm, C. Peterson, G. Braccini, L. Edenbrandt, and L. Sornmo. "Clustering ECG complexes using Hermite functions and self-organizing maps." *Biomedical Engineering, IEEE Transactions on*, **47**(7):838–848, 2000.

[MB08] F. Melgani and Y. Bazi. "Classification of electrocardiogram signals with support vector machines and particle swarm optimization." *Information Technology in Biomedicine, IEEE Transactions on*, **12**(5):667–677, 2008.

[Mey01] A. Meyerson. "Online facility location." In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pp. 426–431. IEEE, 2001.

[MIE07] D. Minnen, C.L. Isbell, I. Essa, and T. Starner. "Discovering multivariate motifs using subsequence density estimation and greedy mixture learning." In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 22, p. 615. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

[MKZ09] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover. "Exact discovery of time series motifs." In *Proc. of 2009 SIAM International Conference on Data Mining: SDM*, pp. 1–12, 2009.

[MM88] R. Mark and G. Moody. "MIT-BIH arrhythmia database directory." *Cambridge: Massachusetts Institute of Technology*, 1988.

[MM96] G.B. Moody and R.G. Mark. "A database to support development and evaluation of intelligent intensive care monitoring." In *Computers in Cardiology 1996*, pp. 657–660. IEEE, 1996.

[MMM84] G.B. Moody, W. Muldrow, and R.G. Mark. "A noise stress test for arrhythmia detectors." *Computers in Cardiology*, **11**(3):381–384, 1984.

[MyS] AB MySQL. "Mysql 5.1 reference manual, 2006." *Accessible in URL: http://dev. mysql. com/doc.*

[NAH10] H. Noshadi, S. Ahmadian, H. Hagopian, J. Woodbridge, N. Amini, F. Dabiri, and M. Sarrafzadeh. "HERMES: Mobile balance and instability assessment system.", 2010.

[NFS03] M. Nilsson, P. Funk, and M. Sollenborn. "Complex measurement classification in medical applications using a case-based approach." In *Workshop on CBR in the Health Sciences*, pp. 63–72, 2003.

[NS04] M. Nilsson and M. Sollenborn. "Advancements and Trends in Medical Case-Based Reasoning: An Overview of Systems and System Development." 2004.

[OKJ08] J. Ottenbacher, M. Kirst, L. Jatoba, M. Huflejt, U. Grossmann, and W. Stork. "Reliable motion artifact detection for ECG monitoring systems with dry electrodes." In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pp. 1695–1698. IEEE, 2008.

[OL01] S. Osowski and T.H. Linh. "ECG beat recognition using fuzzy hybrid neural network." *Biomedical Engineering, IEEE Transactions on*, **48**(11):1265–1271, 2001.

[Omo91] S.M. Omohundro. *Bumptrees for efficient function, constraint, and classification learning*. International Computer Science Institute, 1991.

[OV11] M.T. Ozsu and P. Valduriez. *Principles of distributed database systems*. Springer-Verlag New York Inc, 2011.

[Pan06] R. Panigrahy. "Entropy based nearest neighbor search in high dimensions." In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pp. 1186–1195. ACM, 2006.

[PB10] A. Pantelopoulos and N.G. Bourbakis. "A survey on wearable sensor-based systems for health monitoring and prognosis." *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, **40**(1):1–12, 2010.

[PCL08] KS Park, BH Cho, DH Lee, SH Song, JS Lee, YJ Chee, IY Kim, and SI Kim. "Hierarchical support vector machine based heartbeat classification using higher order statistics and hermite basis function." In *Computers in Cardiology, 2008*, pp. 229–232. IEEE, 2008.

[Per01] P. Perner. "Why case-based reasoning is attractive for image interpretation." *Case-Based Reasoning Research and Development*, pp. 27–43, 2001.

[Pet11] R. Petersen. *Ubuntu 11. 10 Desktop: Applications and Administration*. Surfing Turtle Press, 2011.

[PLC99] S. Park, D. Lee, and W.W. Chu. "Fast retrieval of similar subsequences in long sequence databases." In *Knowledge and Data Engineering Exchange, 1999.(KDEX'99) Proceedings. 1999 Workshop on*, pp. 60–67. IEEE, 1999.

[PT85] J. Pan and W.J. Tompkins. "A real-time QRS detection algorithm." *Biomedical Engineering, IEEE Transactions on*, (3):230–236, 1985.

[Pyt09] J. Python. "Python Programming Language." *Python (programming language) 1 CPython 13 Python Software Foundation 15*, p. 1, 2009.

[Ram72] U. Ramer. "An iterative procedure for the polygonal approximation of plane curves." *Computer Graphics and Image Processing*, **1**(3):244–256, 1972.

[RDM05] N. Ravi, N. Dandekar, P. Mysore, and M.L. Littman. "Activity recognition from accelerometer data." In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, p. 1541. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[RK04] C.A. Ratanamahatana and E. Keogh. "Everything you know about dynamic time warping is wrong." In *Third Workshop on Mining Temporal and Sequential Data*, pp. 22–25, 2004.

[RS02] M.A.D. Raya and L.G. Sison. "Adaptive noise cancelling of motion artifact in stress ECG signals using accelerometer." In *[Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society] EMBS/BMES Conference, 2002. Proceedings of the Second Joint*, volume 2, pp. 1756–1757. IEEE, 2002.

[SCS03] W.Y. Shieh, T.F. Chen, J.J.J. Shann, and C.P. Chung. "Inverted file compression through document identifier reassignment." *Information processing & management*, **39**(1):117–131, 2003.

[SCW11] M. Suh, C.A. Chen, J. Woodbridge, M.K. Tu, J.I. Kim, A. Nahapetian, L.S. Evangelista, and M. Sarrafzadeh. "A Remote Patient Monitoring System for Congestive Heart Failure." *Journal of medical systems*, pp. 1–15, 2011.

[SK98] T. Seidl and H.P. Kriegel. "Optimal multi-step k-nearest neighbor search." In *ACM SIGMOD Record*, volume 27, pp. 154–165. ACM, 1998.

[SK09] J. Shieh and E. Keogh. "iSAX: disk-aware mining and indexing of massive time series datasets." *Data Mining and Knowledge Discovery*, **19**(1):24–57, 2009.

[SMC] Y. Shou, N. Mamoulis, and D.W. Cheung. "Efficient Warping of Segmented Time-series." Technical report, HKU CSIS Tech rep, TR-2004-01.

[SOP04] F. Silvestri, S. Orlando, and R. Perego. "Assigning identifiers to documents to enhance the clustering property of fulltext indexes." In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 305–312. ACM, 2004.

[SWA03] S. Schleimer, D.S. Wilkerson, and A. Aiken. "Winnowing: local algorithms for document fingerprinting." In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pp. 76–85. ACM, 2003.

[TBH02] DA Tong, KA Bartels, and KS Honeyager. "Adaptive reduction of motion artifact in the electrocardiogram." In *[Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society] EMBS/BMES Conference, 2002. Proceedings of the Second Joint*, volume 2, pp. 1403–1404. IEEE, 2002.

[Tec12] Alive Technologies. "Alive Technologies Products.", 2012.

[TFS05] MG Tsipouras, DI Fotiadis, and D. Sideris. "An arrhythmia classification system based on the RR-interval signal." *Artificial Intelligence in Medicine*, **33**(3):237–250, 2005.

[TYS09] Y. Tao, K. Yi, C. Sheng, and P. Kalnis. "Quality and efficiency in high dimensional nearest neighbor search." In *Proceedings of the 35th SIGMOD international conference on Management of data*, pp. 563–576. ACM, 2009.

[Uhl91] J.K. Uhlmann. "Satisfying general proximity/similarity queries with metric trees." *Information processing letters*, **40**(4):175–179, 1991.

[WA07] L.B. Wood and H.H. Asada. "Low variance adaptive filter for cancelling motion artifact in wearable photoplethysmogram sensor signals." In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pp. 652–655. IEEE, 2007.

[WE02] M.A. Wood and K.A. Ellenbogen. "Cardiac pacemakers from the patients perspective." *Circulation*, **105**(18):2136–2138, 2002.

[WLS11] J. Woodbridge, M. Lan, M. Sarrafzadeh, and A. Bui. "Salient Segmentation of Medical Time Series Signals." In *Healthcare Informatics, Imaging and Systems Biology (HISB), 2011 First IEEE International Conference on*, pp. 1–8. IEEE, 2011.

[WSB98] R. Weber, H.J. Schek, and S. Blott. "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces." In *Proceedings of the International Conference on Very Large Data Bases*, pp. 194–205. INSTITUTE OF ELECTRICAL & ELECTRONICS ENGINEERS, 1998.

[YJF98] B.K. Yi, HV Jagadish, and C. Faloutsos. "Efficient retrieval of similar time sequences under time warping." In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pp. 201–208. IEEE, 1998.

[ZM06] J. Zobel and A. Moffat. "Inverted files for text search engines." *ACM Computing Surveys (CSUR)*, **38**(2):6, 2006.

[ZS07] J. Zhang and T. Suel. "Efficient search in large textual collections with redundancy." In *Proceedings of the 16th international conference on World Wide Web*, pp. 411–420. ACM, 2007.