A Mining Technique Using N-grams and Motion Transcripts for Body Sensor Network Data Repository

Vitali Loseu*, Hassan Ghasemzadeh*, Latifur R. Khan†, and Roozbeh Jafari*
Embedded Systems and Signal Processing Lab*
Department of Electrical Engineering*
Department of Computer Science†
University of Texas at Dallas, Richardson, TX 75080-3021
{vitali.loseu, h.ghasemzadeh, lkhan, rjafari}@utdallas.edu

ABSTRACT

Recent years have seen a large influx of applications in the field of Body Sensor Networks (BSN). BSN, and in general wearable computers with sensors, can give researchers, users or clinicians access to tremendously valuable information extracted from data that were collected in users' natural environment. With this information, one can monitor the progression of a disease, identify its early onset or simply assess user's wellness. One major obstacle is managing repositories that store large amounts of BSN data. To address this issue, we propose a data mining approach for large BSN data repositories. We represent sensor readings with motion transcripts that maintain structural properties of the signal. To further take advantage of the signal's structure, we define a data mining technique using n-grams. We reduce overwhelmingly large number of n-grams via information gain (IG) feature selection. We report the effectiveness of our approach in terms of the speed of mining while maintaining an acceptable accuracy in terms of precision and recall. We demonstrate that the system can achieve average 99% precision with an average 100% recall on our pilot data with the help of only one transition for each movement.

Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database Applications—*Data mining*

General Terms

Design, Algorithms, Experimentation

Keywords

Body Sensor Networks, String Templates, N-grams, Patricia tree, Data Mining

1. INTRODUCTION

Body Sensor Networks are becoming an increasingly popular field of research for a variety of applications from fall and posture detection [22, 40], and tele-medicine to rehabilitation and sports training [3, 19]. These systems are based on tiny sensor nodes with different sensors selected based on the application. BSN systems are desirable because they provide a relatively inexpensive way to collect realistic and more importantly quantitative data about the subjects without constraints of the lab environment. A problem that has not received much attention is storing and tracking all of the collected data. BSN data is valuable especially in the cases of medical observations. The ability to search and compare BSN observations can potentially shed light on diseases such as Parkinson's Disease [38], which does not have a cure or even a quantitative, objective diagnostic process [10]. The following example demonstrates the usefulness of the idea. A person can be observed for an extended period of time, for example multiple years, with the help of a few sensor nodes. All of the data is simply collected and stored in the data repository. After some time a person is diagnosed with a disease that affects walking. It would be beneficial to analyze old data and extract gait parameters for disease evaluation. A data mining approach would be able to identify movements of interest, in this case walking, so that the raw data of movements can be used to extract the required gait parameters. When comparing the signals we rely on the idea that similar movements have inherently similar structure, while different movements have fundamental differences, which is also an assumption of structural pattern recognition [5]. For structural techniques to be effective it is essential to design an effective data representation approach that simplifies the multidimensional BSN data yet captures the structure of the signal.

The system can extract structural and relational information from the data [31] by representing body states as a set of basic motions called primitives. The idea of motion primitives has been explored in both computer vision and sensor networks. For example, facial expressions can be split into 'atomic' motions [4] and then later appropriately combined to improve smoothness of the $3D$ models. A more advanced study focuses on defining a Human Activity Language (HAL) by splitting movements into primitives based on the video data [13]. A similar idea of representing human movements in terms of string sentences generated by identifying atomic body poses with a multi-view video of the experiment [14]. Fewer works use the notion of motion primitives based on the inertial sensor readings. These approaches normally involve clustering training data and generating class labels for each point of the signal. Supervised

Wireless Health '10, October 5-7, 2010, San Diego, USA

clustering can be used to identify important events of the signal, and look only for those events during the system execution leaving the rest of the input unlabeled [25]. Unsupervised clustering can be used to map each complete action into a motion primitive and then use the acquired primitives for classification [11]. Overall, these approaches suggest that structural pattern recognition excels in recognizing individual primitives and sequences of primitives. This property can potentially be applied to such data mining problems as discovering similar movements, finding similar portions of movements, and finding similar abnormalities in movements.

Based on the idea of structural pattern recognition we present a data mining model for a large BSN data repository. We first handle the problem of the BSN data representation. To achieve it, we define a technique for movement primitive construction for multidimensional BSN data using clustering. We explore ways to preserve the original structure of the signal, even if the movement trials have timing inconsistencies. Unlike other works, we consider multiple clustering techniques for primitive construction using a small and computationally simple feature set. We use the constructed primitives to generate string transcripts to capture the relational information from the signal. We then define a novel data mining model that explores structural and relational properties of the string transcripts via n-grams computation and selection. We use information gain to select the n-grams that can best differentiate between movement, and then define a Patricia tree for data indexing and mining. We verify the quality of our model by applying it to a pilot movement data set. While other works focus on achieving the highest possible accuracy of classification, the key objective of our work is to define a data mining approach that can be applied to a very large data set, which results in decisions that favor speed and simplicity of computation. To the best of our knowledge this has not been applied to multidimensional data like the one collected by BSN sensor nodes.

2. RELATED WORK

During data collection researchers aim to minimize the number of nodes used on a subject to improve system wearbility. This results in a particular choice of sensor types, node count, and node placement. In a practical deployment scenario, a subject's preferences may also cause some changes in the way sensors are placed. For example, a cell phone on the belt of one subject and a sports watch on another can be collecting accelerometer data about walking. These differences seem to deem the information not comparable and possibly not useful. This problem can be resolved with a larger BSN repository, where similarities in portions of the data can link multiple data sets together, thus providing user with more data or even a new perspective of the data with a sensor not immediately available to that user. It is not practical to use the metadata about the experiment, such as movement type or speed, to combine observations, because two variations of the same movement might be performed very differently. In other words the system should be able to recognize not only the movement itself, but also the specific way the movement is performed. For example, when the system is searched for occurrences of limping in the right foot of the subject, it should not return every instance of walking, even though limping is observed during a walking trial. To avoid this, the structure of the compared movement needs to be investigated. Structural pattern recognition can help create an accurate and lightweight approach for data manipulation in the BSN data bank.

An approach for structural data representation and recognition is proposed in [12]. This approach has a major weakness. The comparison evaluation is based on the value of Levenstein distance (or edit distance) [20]. Edit distance calculation assigns the same weight to deletion, insertion, and substitution operation. It is not a problem when the compared strings have similar size. However, BSNs can observe the same movement at different speeds, which may mean that the speed of movement execution can start dominating the edit distance value. It is possible to manually manage the weights of each one of the three edit distance operations, however that would generate a heuristic type of approach [18]. Another way to deal with this issue is to normalize the length of each primitive in motion transcripts [26]. While this approach might work in some specific applications, in general it is very hard to predict how to scale parts of movements depending on the overall execution speed. A possible solution to this problem is to identify significant transitions in the motion transcripts and base the transcript comparison on variations in these transitions. In the field of speech processing, a similar function is often performed by n-gram features. n-grams are substrings of length N, they were first introduced by Shannon [37] as means to analyze vulnerability of ciphers but since have been extensively used in the field of speech and text recognition.

N-grams [8] proved to be useful in structural parameter extraction when used for spoken language recognition [2]. N-grams can be used to capture phoneme, in the case of spoken language, and grammatic constructs, in case of written language, to identify bodies of speech or text. Similarly, n-grams can be used to analyze text summaries [21] or translation quality[8] with respect to co-occurrence statistics. While good at recognizing major structural differences, n-grams can also be used in the case of fine tune spelling error correction [17]. In addition to maintaining structural information of the considered string, n-grams can also significantly reduce the amount of information that needs to be stored and verified. Instead of storing a large body of text, the system can identify important transitions and improve both memory usage as well as execution speed of the search. This idea is applied in malicious code detection software [27, 1]. Instead of keeping a database of viruses, authors extract n-grams that characterize each malicious code, and search files for presence of only those n-grams. These important n-grams can be better organized with a suffix tree [24], which would increase the speed of identifying language constructs [32]. In fact suffix trees are often used to index a large data store in natural language processing and other fields. For example in molecular biology, DNA sequences can be indexed with the help of suffix trees [6]. Authors in [35] discuss an efficient query algorithm on a large compressed body of text using suffix trees. The general effectiveness of the suffix trees is discussed in the work trying to identify local patterns in a event sequence database [16, 7].

At first glance, the above examples have little in common with data collected from BSNs. Suffix tree approaches normally index a unidimensional data set, while BSNs normally have a set of multiple sensors with multiple dimensions of sensing. This problem can be resolved by combining all of the data readings and representing them with unidimensional primitives [12]. While this simple approach seems to

resolve the issue, it fails to recognize that each one of the sensing axis can observe variations such as varying speed and amplitude of the signal. In a text data set the variations are one dimensional, just like the data itself; this is not the case in multidimensional sensor readings of BSNs. Furthermore, it is not clear how variations occurring in multiple sensing dimensions should be handled in the context of a one dimensional primitive. It is possible that different combination of signal variations may hinder the structural consistency of the combined primitive representation.

3. PROBLEM DESCRIPTION

In this paper, we try to address the problem of data mining for the BSN data repository. The system takes raw sensor readings as an input and searches the repository for the signals similar to the input. Due to a potentially large size of the repository, the approach has to be fast yet reliable. As a result, we focus on the speed and simplicity of the approach. When a raw sensor reading of an observation is given as an input, the approach should be able to identify a movement the observation belongs to, so it can be stored in the appropriate place in the repository. It should be able to compare signals of two movements and find possible similarities. Furthermore, it should be able to identify similar portions of the signals, which can be useful if a subject exhibits a consistent abnormality in performing multiple movements. Finally, it should be able to identify movements that contain certain signal instances. For example, identify all of the movements where the torso moves forward.

Table 1: Pilot Application Movements

No.	Description
1	Stand to Sit
2	Sit to Stand
3	Stand to Sit to Stand
4	Kneeling, right leg first
5	Turn counter clockwise 90 degrees
6	Look back clockwise
7	Move forward (1 step)
8	Move to the left (1 step)
9	Move to the right (1 step)
10	Jumping

3.1 Pilot Application

While we are designing an approach for data mining in a large BSN repository, a data set that large is not available to us. Instead, we apply the proposed approach to a classification problem. Classification accuracy is similar to the indexing and searching accuracy, which means that a classification application can assess the precision of the proposed technique. Furthermore, as the approach is designed for a large data set we aim to make it lightweight and as fast as possible. For the experiment we collected data of ten movements from three subjects. The details of the experimental movements can be found in Table 1. Every subject repeated each movement ten times to increase the size of the data set. Each subject was equipped with nine sensor nodes positioned as demonstrated in Figure 1.

3.2 Hardware

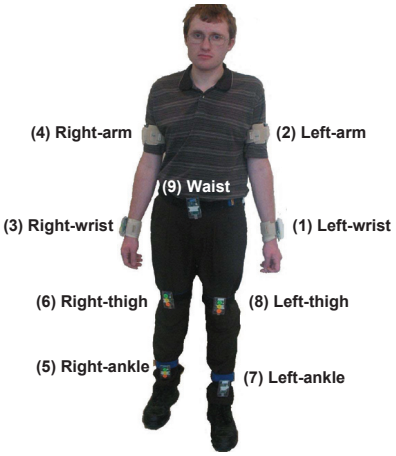

Figure 1: Subject with motes equipped

During the data collection subjects were equipped with several TelosB sensor nodes with custom-designed sensor boards. Each sensor board has a tri-axial accelerometer and a bi-axial gyroscope. Sensors were sampled at $50Hz$, which is a fairly standard data collection frequency for movement applications [33, 28] that satisfies Nyquist criterion [39]. After collecting the data, each node sent its readings to the basestation. In our case, the basestation is a node without a sensor board, which forwards all of the received data to the PC via a USB connection for further processing.

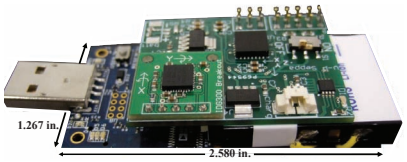

Figure 2: Sensor Node

4. DATA REPRESENTATION

A physical movement can normally be represented as a sequence of shorter motions. Capturing the structure of the movement involves capturing these shorter motions and timing relationships between them. This can be done by identifying motion primitives. This task is not trivial as it has to be done without any prior knowledge of the movement itself but with respect to the observed signal. A common way for unsupervised data grouping is clustering. We follow the idea introduced in [12] and use a clustering technique for primitive generation. We extract features from the signal and cluster the resultant feature set, which means that the clustering outcome is dependent on the perspective that the features can provide. This adds flexibility to the system because different feature sets can characterize the signal from different perspectives.

4.1 Primitive Construction

Before applying a clustering technique, it is necessary to decide what data set the clustering is applied to. One way to handle this issue is to combine all of the sensory axis of

one node and use all of the available to the node data to define primitives. This approach is flawed because, when the multidimensional data is merged into a unidimensional primitive, combining variations of each of the sensing axis could modify the structure of the primitives. Each one of the sensory axis can produce slightly different readings due to a minor alteration in the movement performance. An example of such an alternation can be a slight delay of the movement. This alternation does not modify the structure of the individual sensory axis signals, but, since alternations on all of the axis are independent of each other, aligning them with respect to time can significantly change the structure of the primitives. Figure 3 demonstrates an example where a slight variation in one of the axis' signals, which does not violate the signal structure for that axis, introduces changes to final primitives. In the figure, individual axis of sensing have consistent structure for both trials. If the sensing axis were to be combined they would be combined with respect to time. However, since the primitive transitions are not aligned in the original signals, the time aligned result of axis combining will not have consistent structure for both trials. We also expect to have noise in the inertial data our sensor nodes collect, which will introduce another source of error.

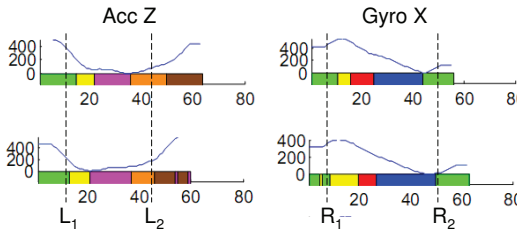

Figure 3: Signal Alignment Issue

To avoid the issue with alignment, we treat the reading of each sensing axis separately. Primitives are created for each one of the axis. In effect, we treat each sensing axis as a separate mote. This approach has an additional benefit by increasing the flexibility of the system. The system does not require all of the sensing axis used in one experiment to be present in other experiments, which means that it does not force users into a particular hardware configuration.

4.1.1 Data Clustering

Clustering is a very effective method of grouping similar data points, and distinguishing between different data points. When trying to cluster BSN data, a clustering approach is normally applied to feature vectors extracted from the original signal. There is a variety of features that can be extracted from inertial data. Different approaches rely on first and second derivatives, signal mean, amplitude, variance, standard deviation, peak detection, morphological features and more. During our study we tried to identify a small and simple data set that would produce good results. Our primitive generation experiment concluded that first and second derivatives are sufficient to describe the structure of our data set. To minimize the effect of the inter-subject differences in features, the system normalizes features with respect to each subject using standard score (or z-score) [30]. While we used these features for our experiment, the proposed approach is independent of the feature selection and only requires that selected features would represent the structure of the input signal.

There is a wide array of clustering techniques that includes hierarchical, partitional, conceptual, and density based approaches. For our analyses we considered two clustering approaches. First, we considered a k-means clustering approach [23]. k-Means is a hierarchical approach that attempts to partition the data in a way that every point is assigned to a cluster with the closest mean, or cluster center. In its spirit this approach is similar to expectation maximization in Gaussian Mixture Models (GMM) [9], which is the second clustering approach that we consider. The approaches are similar, because both of them try to identify the centers of natural clusters of the data. GMM clustering computes the probability that any given point is assigned to every individual cluster and makes an assignment that maximizes its likelihood. We selected the above approaches because of their computational simplicity, and the property that they are trying to identify natural cluster centers of the data set without any prior knowledge of the data. Both approaches start with random cluster centers and re-evaluate them after each round of computation. Once the cluster centers stay constant within a predefined threshold, both algorithms assume to have converged to the natural cluster centers of the data and return the result.

A major problem to consider during unsupervised clustering is the number of clusters k that produces the best results. To find the best solution we varied k from 2 to the length of the shortest observation in the training set, while evaluating parameters of the both k-Means and GMM models. In case of k-Means, we made the decision based on cluster Silhouette [34]. Silhouette is calculated based on the tightness of each cluster and its separation from other clusters. For every point i the silhouette is defined as

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \quad (1)$$

where $a(i)$ is the average distance of point i to all other points in its cluster, $b_j(i)$ is the average distance of point i to all the points in cluster j, and $b(i) = min(b_j(i)), \forall j$.

Silhouette $s(i)$ describes how well the point i is mixed with the similar data points and is separated from the different data points. As a result, the quality of a clustering model with k clusters and d training points can be evaluated as

$$Quality(k) = \frac{\sum_{i:1}^{d} s(i)}{d} \quad (2)$$

The larger the average silhouette value, the better is the model. Therefor, the best value of k can be selected by finding the largest $Quality(k)$ [34].

In case of the GMM we used Expectation Maximization (EM) [29] to find the best mixing parameters for GMM. The mixing parameters, such as the mean and covariance matrices depend on the number of clusters k. Once the GMM parameters are selected there are multiple ways to evaluate the quality of clustering that include log likelihood, Akaike's information criterion (AIC) [41], and Bayesian information criterion (BIC) [36]. Table 2 demonstrates the difference between quality estimation models for a GMM with k clusters, maximum likelihood of the estimated model L, and n points in the training set.

Table 2: GMM Quality estimation

Log Likelihood	AIC	BIC
$ln(L)$	$-ln(L) + 2 * k$	$-ln(L) + k * ln(n)$

Log likelihood just reports the likelihood of the model, AIC and BIC attempt to penalize the system for the number of clusters. We found the penalty of the BIC to be too harsh on our data set, which led to an extremely small number of clusters. As a result we selected AIC as the GMM model evaluation tool. We evaluated the value of AIC based on different values of k and selected k that produced the smallest AIC result.

Once the optimal number of clusters is selected for both k-Means and GMM, it is necessary to evaluate the difference between the clustering approaches, and see which one is more suitable in the context of a data mining approach we are proposing. Figure 4 shows that GMM clusters are not convex, which means that some of the structural information of the data can be lost. For example, it is possible to have a signal where data with both positive and negative first derivative are clustered together, while points with first derivative equal zero are assigned to another class. From the perspective of preserving structure some information is lost. In Section 8 we will demonstrate how this difference in the clustering approach affects the overall data mining accuracy.

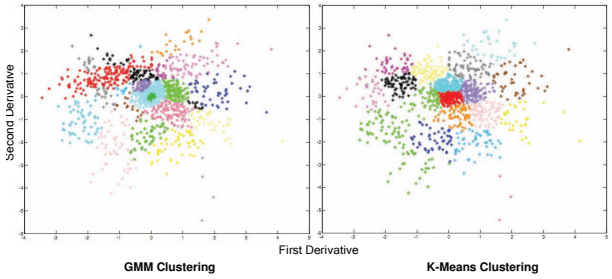

Figure 4: Clustering of the Training Set into 19 components

4.2 Motion Transcripts

Each movement can be described as a series of primitives. When an unlabeled movement comes into a system. The system can extract features from each point, and, based on the clustering technique, assign motion primitives to them. Motion transcripts are sequences of primitives over a certain alphabet assigned to movement trials. Since the data from different motes are not comparable, the system has to make sure to differentiate between individual motes by using a unique alphabet for each one. Figure 5 demonstrates a sample transcript generated by the ankle node for a 'Lie to Sit' movement. Each one of the sensing axis uses a separate alphabet, so while they are displayed with the same color the values of different transcripts are not related.

5. COMPARISON METRIC

Once the BSN data is converted to motion transcripts the system requires an efficient way to classify and search them. In Section 2 we discussed edit distance, a common approach to compare strings. However, edit distance does not perform

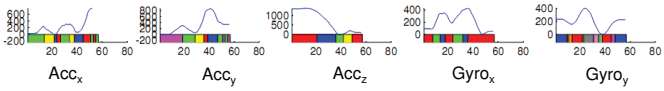

Figure 5: Sample Transcripts

well when the input data has noise and varies in length. Additionally, edit distance calculation is very slow with order of $O(n^2)$, where n is th length of the string. While it may be an acceptable solution for a small application, its speed performance is not at all acceptable for a large data repository potentially containing terabytes of data. To resolve the issue of edit distance, the system can use the idea of n-grams that can track transitions in motion primitives in linear time with respect to the trial length. The goal of the n-grams is to track important transitions between movement primitives in string transcripts. However, the task of identifying n-grams that represent important transitions is not trivial. The difficulty of the task is increased since overlapping n-grams are extracted to improve the quality of the recognition. This means that potentially there is a very large number of n-grams that can be selected from any given transcript. Furthermore, we expect the data to noisy; and it is important to train the system only on the characteristic n-grams and not on the n-grams that represent noise. This task is achieved via n-gram selection.

5.1 N-gram Selection

The objective of this operation is to identify a small number of n-grams that can uniquely characterize the movement of interest, and provide means of distinguishing that movement from others in the repository. There is a variety of ways to select proper n-grams, once n-grams are extracted from all of the training data. Information gain has proven to be effective in the field of natural language processing [42]. Information gain becomes complicated to compute and less effective when each evaluated feature can take a large number of values. However, in our experiment each n-gram has two possible values. A specific n-gram can be present in a motion trial and the value of '1' is assigned to it, or the n-gram can be absent so a value of '0' is assigned. While IG proved effective on our data set, the proposed approach is not dependent on this particular n-gram selection technique and can be modified based on the specific user demands.

IG can assess the effectiveness of a feature by tracking changes in the entropy after consideration of that feature. IG of an feature f on the collection of movements m is defined as

$$Gain(m, f) = H(m) - H(m|f) \quad (3)$$

where $H(m)$ defines entropy of the movement set, and $H(m|f)$ defines conditional entropy of the movement set with respect to feature f. We use a slightly modified approach, because when the system is looking for a target movement all the other movements can be treated the same way. It is possible that a feature might be a good at identifying one movement while being unable to differentiate between the rest of the movements. That feature would have a bad general information gain, however if we compute information gain with respect to each movement, we can identify good features for

each movement. Practically, this means that while computing information gain of a feature with respect to particular movement m_i, the movement set is split into subsets of $\{m_i\}$ and $\{$'not' $m_i\}$ or $\{m - m_i\}$. In this case $H(m_i|f)$ can be different for each m_i and need to be calculated individually. Which means that we can redefine the gain information as

$$Gain(m_i, f) = H(m_i) - H(m_i|f) \tag{4}$$

$H(m_i)$ represents the amount of expected information that set m carries itself with respect to movement m_i and can be defined as

$$\begin{aligned} H(m_i) = &- prob(m_i) * log_2\left[prob(m_i)\right] \\ &- prob(m - m_i) * log_2\left[prob(m - m_i)\right] \end{aligned} \tag{5}$$

Conditional entropy $H(m_i|f)$ defines the expected amount of information the set m carries with respect to feature f and movement m_i.

$$H(m_i|f) = - \sum_{v \in \{0,1\}} \frac{count(f_v)}{total\_count} * \tag{6}$$
$$\left(\frac{pM(f_v)}{count(f_v)} * log_2 \frac{pM(f_v)}{count(f_v)} - \frac{pNM(f_v)}{count(f_v)} * log_2 \frac{pNM(f_v)}{count(f_v)} \right)$$

where $count(f_v)$ represents the number of training trials where f has value of a specific v, $pM(f_v)$ corresponds to the number of trials of the movement m_i where f has value of a specific v, and $pNM(f_v)$ corresponds to the number of trials not of the movement m_i where f has value of a specific v, and $total\_count$ is the total number of training samples.

Once all the n-grams have an IG assigned to them for each movement, we can sort the list of IGs and select t n-grams that have the best IG. This is a very simple approach because it does not consider correlation between features, meaning that some of the features can be redundant. However, even this simple approach can generate good results [27] and is selected for simplicity.

6. CLASSIFIER

Once the set of good n-grams is selected, an approach needs to be defined for fast movement classification and search. This approach also should not rely on the knowledge of the complete structure of the data, and be able to finish classification and search based on partial information. These properties are exhibited by suffix trees [15]; more specifically, we used Patricia tree in our implementation. Patricia trees are used to represent sets of string by splitting them into substrings and assigning substrings to the edges. This idea fits naturally with n-grams that are substrings. Once all of the n-grams are selected for each movement, we combine them and assign the combined set to the edges of a Patricia tree. The paths from the root to all of the leafs corresponds to all the possible permutations of the combined n-gram set. Once the Patricia tree is created, we traverse it for each training trial, and assign movements to the final node of the trial's traversal, which means that at the end of the training, each leaf of the Patricia tree corresponds to a subset of the movements. It may be an empty set or it may contain one or more movements.

7. DATA MINING MODEL

Based on the construct defined earlier we propose a data mining approach. The approach has two distinct parts: *Training* and *Query Processing*.

7.1 Training

During the training phase of the execution the system acquires parameters that can be used during the query processing. The training starts with selecting a portion of the available data trials for training. First and second derivatives are then extracted from each one of the trials for every sensing axis of the motes. Features are then normalized with respect to each subject using standard score (or z-score) [30] in order to remove inter-subject variations of the same movements. Then normalized features are used to define data clusters as described in Section 4.1.1. Once the data clusters are defined, primitives are extracted for the data points in each training trial, and then combined to define motion transcripts as described in Section 4.2. The next step is to extract n-grams from each one of the transcripts generated for the training samples. Since the number of n-grams is very large, the system then selects a small number of t n-grams using information gain as described in Section 5.1. Finally, the system constructs a Patricia tree with selected n-grams on the edges and movement classes on the leafs as described in Section 6. The overall process is demonstrated in Figure 6. The parameters defined during the training are data clusters for each sensing axis of the motes, important n-grams selected with respect to the information gain criteria, and Patricia trees for classification are defined for each mote. Clusters are represented by the cluster center coordinates, while important features selected for each sensing axis of each motes are then combined and stored.

7.2 Query Processing

When a system needs to classify or query for a movement it receives input in the form of the sensor readings. First and second derivatives are extracted from the sensor readings of each of the sensing axis. Based on these features and clusters, defined during the training of the system, each data point of the trial is labeled with a primitive and motion transcripts are created. The system then traverses the transcript of the trial and verifies if it contains important n-grams selected during the training. Using this information the system traverses the Patricia tree defined for this sensing axis during the training and returns the set of movements assigned to the leaf the traversal terminated at. Since all of those operations are defined in terms of individual sensing axis, an approach is required to combine the local decisions. In order to avoid enforcing a certain structure on the hardware, we aim to define a flexible approach that can make a decision with only a subset of sensing axis available. When this idea is combined with the need for speed and simplicity, a simple majority voting approach is applied. This approach performs well in our pilot application, however, the system is not constrained to this approach and can be modified. The flow of the query processing is demonstrated in the Figure 7.

Because the system initially processes each sensing axis individually, it is possible to query only for a subset of axis available in the system. This can be useful when a specific sensor is not available to all users. For example, one user can use a three-dimensional gyroscope, while another may

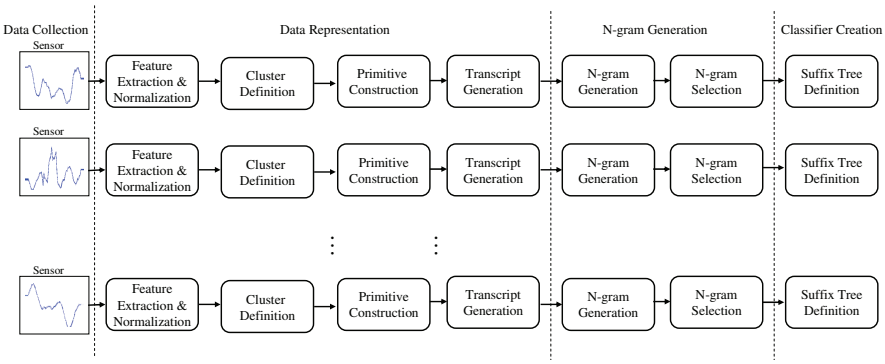

Figure 6: System Training Flow

use only a two-dimensional gyroscope. Additionally, since the system uses a voting scheme it is possible to make classification decisions based on the local view of only a subset of nodes.

7.3 Complexity Analyses

Since the system training can be done off line with no particular time constraints we consider only the complexity of the query processing. Complexity of individual components is described in Table 3, where l is the length of the trial, k is the number of clusters, t is the number of selected features, and m is the number of movements. Note that the total complexity combines complexity of all the sensing axis, however that number is constant (in our case 5) and does not affect the Big-O analyses. The total complexity is $O(l*k+t*m)$. Both k and t are parameters and do not need to be large for good classification results. Which means that the running time of the approach is defined either by the length of each classified trial, or by the number of movements in the system. Simple human movements normally do not exceed $2-3$ seconds, which means that the length of a trial sampled at $50Hz$ would be around $100-150$ samples long. For a repository with a small number of movements the computation would be almost linear with respect to the trial time. For a repository with a large number of movement, exceeding 150, the running time would be almost linear with respect to the number of movements in the repository. It is potentially possible for the repository to contain a very large number of movements, which would make our approach inefficient. The current work does not provide a solution to this problem, however it is going to be a part of our future investigation.

Table 3: Component Complexity

Component	Complexity
Feature Extraction	O(l)
Feature Clustering	O(l*k)
Extracting n-grams	O(l)
Selecting n-grams	O(l)
Tree traversal	O(t*m)
Total	O(l*k + t*m)

8. EXPERIMENTAL RESULTS

To verify the performance of our approach we apply it to a pilot application discussed in Section 3.1. We first compare the results of the system when k-Means clustering is used to the results when GMM clustering is used. We then consider accuracy of the approach with respect to the length of the n-gram n, and number of features selected t. For small numbers of n and t, which would allow us to consider those as constants in the complexity analyses, the approach achieves nearly linear time. Finally, we demonstrate the accuracy trade-off with respect to n and t.

8.1 k-Means or GMM?

In this section we apply our approach to movement transcript generated based on the k-Means and GMM clustering. We compare the two approaches for a 3-gram with the number of selected n-grams varying from 1 to 6 per sensing axis in Table 5, and with {1..6}-gram with only 1 n-gram selected from each sensing axis in Table 4. Both tables indicate that an increase in t or n would increase the precision and recall for both approaches till the over fitting point is reached. It is also clear that the GMM approach outperforms the k-Means approach with respect to varying both n and k, and therefore is a better candidate for our application.

Table 4: 3-gram average performance of k-Means vs GMM with {1..6} n-grams selected

t	GMM		k-Means	
	Precision	Recall	Precision	Recall
1	.93	.97	.75	.86
2	.96	.98	.88	.94
3	.96	.98	.89	.95
4	.95	.96	.90	.94
5	.94	.96	.91	.96
6	.94	.96	.85	.92

8.2 Classification Accuracy

To evaluate classification accuracy of the model we evaluate precision and recall of movement classification using the n-gram size of $n = 3$, and select number of features $t = \{1, 2...5\}$ with GMM clustering model. We tested our model by splitting the data into two parts. Half of the data

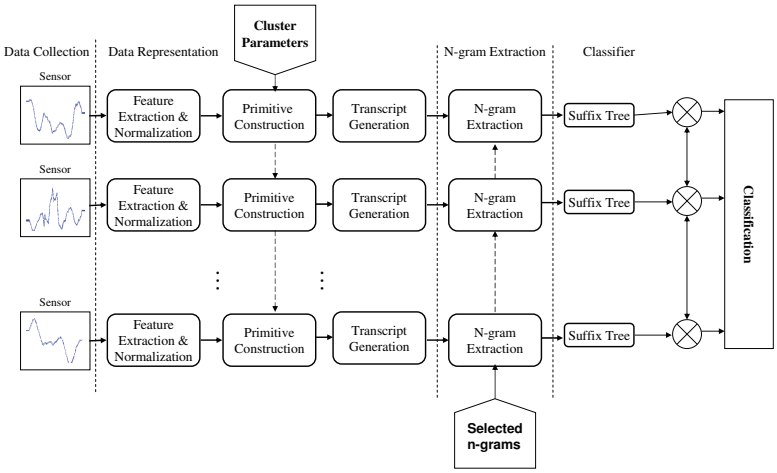

Figure 7: Query Processing Flow

Table 5: {3-7}-gram average performance of k-Means vs GMM with 1 n-gram selected

n	GMM		k-Means	
	Precision	Recall	Precision	Recall
3	.93	.98	.75	.86
4	.99	1	.76	.84
5	.99	1	.83	.89
6	1	1	.89	.93
7	.95	.99	.91	.93

was used to train the system, while the other half was used to test it. The results of the classification are demonstrated in Figure 6 for the precision, and Figure 7 for the recall. These tables confirm that adding more n-grams would improve both average precision and average recall until an over fitting point is reached. Note that individual values for movements sometimes decrease when an additional feature is selected. This is due to the fact that the data set we have has a considerable amount of noise, and while an n-gram improves the overall classification accuracy it may cause confusion in classification of some trials where it appears as noise and not an important transition. Tables 6 7 display the number of n-grams extracted from each sensing axis, so the total number of the n-grams extracted by a sensor node should be multiplied by 5. However, even after that the classification accuracy is fairly high for the number of considered features.

8.3 Parameter Trade-off

The system we define has two parameters that classification accuracy depends on. We can select the length of the substring n and the number of n-grams t selected for classification. As t is increasing so does the accuracy until the over-fitting point is reached. After the over-fitting point is reached the accuracy of the approach will no longer improve with additional features. It is clear that a large n inherently is able to capture more structural information. However, since we use a moving window n-gram extraction, a single erroneous primitive affects more n-grams for large values of n, which means that the over fitting problem or training on the trial but not model specific n-grams should happen

Table 6: Classification Precision

Movement	Features per sensing axis				
	1	2	3	4	5
Stand to Sit	1	1	.92	.92	.75
Sit to Stand	1	1	1	1	1
Stand to Sit to Stand	1	1	1	1	1
Kneeling, right leg first	.67	.75	.83	.83	.92
Turn 90 degrees	1	1	1	.92	.92
Look back clockwise	1	1	1	1	1
Move forward	1	1	1	1	1
Move to the left	.92	1	1	1	1
Move to the right	.75	.83	.83	.83	.83
Jumping	1	1	1	1	1
Average	.93	.96	.96	.95	.94

sooner. We expect the system to converge to the best accuracy faster for large points of n but it also means that the over fitting point will happen faster as well. Table 8 and Table 9 demonstrate precision and recall of accuracy vs the number of n-grams t for different values of n.

From the tables it is clear that higher values of n are desirable before over fitting, which means that n should be determined based on the expected amount of noise in the original signal. For the lower amount of noise a higher value of n would work better, while when the amount of noise is large, low values of n will provide a safer solution with less risk of over fitting. In this example, the quality of the precision is improving from $n = 3$ to $n = 5$, it is fairly stationary from $n = 5$ to $n = 7$, and finally $n = 9$ has decreasing results. The fact that large n-grams take more time to locate in the training trials should also be considered. The system can evaluate multiple possibilities during the training and generate the curves to identify the best operational point from the perspective of the application.

9. FUTURE WORK

In this work we explored possibility of a fast search mechanic on a large data set of BSN data with acceptable accuracy based on a pilot study. While we achieved promising results there are a few issues that have to be addressed be-

Table 7: Classification Recall

Movement	Features per sensing axis				
	1	2	3	4	5
Stand to Sit	1	1	1	1	1
Sit to Stand	1	1	.92	.92	.86
Stand to Sit to Stand	1	1	1	1	.92
Kneeling, right leg first	1	1	1	1	1
Turn 90 degrees	.86	1	.92	1	1
Look back clockwise	.92	.86	1	.92	.92
Move forward	1	1	1	1	1
Move to the left	1	1	1	1	1
Move to the right	1	.91	1	.83	.91
Jumping	1	1	1	1	1
Average	.98	.98	.98	.97	.96

Table 8: Precision with respect to n-gram size, and number of n-grams selected

n	Features per sensing axis				
	1	2	3	4	5
3	.93	.96	.96	.95	.94
5	.99	.97	.96	.98	.97
7	.99	.99	.99	.98	1
9	.96	.94	.94	.95	.93

fore this approach can be practically deployed on the real data. In this approach we assumed that the compared physical axis of the sensors are perfectly aligned, which may not be the case during a real experiment where misplacement is possible. Addressing the misplacement issues is the next step in the development of our model. Additionally, the current n-gram selection approach assumes that all of the possible movements are available during the system training and n-gram selection. Before the observation of new movements can be made available, the system has to retrain itself, which is an acceptable temporary solution. It is desirable to define an n-gram selection approach that can dynamically adjusts with respect to new data without the need for a complete system re-training. Finally, for a database with a very large number of movements, the performance of our approach might start degrading. We would like to consider an approach to group movements together and perhaps first locate the group of movements in question, and then select the movement in that group that best matches the input signal.

10. CONCLUSION

We generated motion primitives based on instantaneous simple features and unsupervised clustering. We showed how the signal primitives can be combined into motion transcripts, which are unidimensional representations of the multidimensional BSN data. Inspired by the techniques of natural language processing, we applied the concept of n-grams retrieval for tracking transitions in the movement transcripts. Due to the large number of n-grams extracted from a movement trial, we apply a simple information gain approach to the features to select k features that provide the most information about each sensing axis. Based on the selected n-grams we build a suffix tree for fast query and identification of movements in the database. We evaluated two different clustering approaches with respect to classification

Table 9: Recall with respect to n-gram size, and number of n-grams selected

n	Features per sensing axis				
	1	2	3	4	5
3	.98	.98	.98	.97	.96
5	1	.97	.98	.99	.98
7	.99	.99	.99	.98	1
9	1	1	.99	.99	.99

quality of our model and demonstrated that a GMM based clustering outperforms k-Means in the context of our pilot application. We demonstrate that the system can achieve average 99% precision with average 100% recall on our pilot data with the help of only 1 characteristic for each moevement transition. We also explored the trade off between the length of the extracted n-grams and the required number of features for the best classification results.

11. REFERENCES

[1] T. Abou-Assaleh, N. Cercone, V. Keselj, and R. Sweidan. Detection of new malicious code using n-grams signatures. In *Proceedings of Second Annual Conference on Privacy, Security and Trust*. Citeseer, 2004.

[2] A. Adami and H. Hermansky. Segmentation of speech for speaker and language recognition. In *Eighth European Conference on Speech Communication and Technology*, 2003.

[3] D. Brunelli, E. Farella, L. Rocchi, M. Dozza, L. Chiari, and L. Benini. Bio-feedback system for rehabilitation based on a wireless body area network. pages 5 pp.–531, March 2006.

[4] T. Bui, D. Heylen, and A. Nijholt. Combination of facial movements on a 3d talking head. In *Computer Graphics International, 2004. Proceedings*, pages 284–290, 2004.

[5] H. Bunke and A. Sanfeliu. *Syntactic and structural pattern recognition: theory and applications*. World Scientific Pub Co Inc, 1990.

[6] S. Burkhardt, A. Crauser, P. Ferragina, H. Lenhof, E. Rivals, and M. Vingron. Q-gram based database searching using a suffix array (QUASAR). In *Proceedings of the third annual international conference on Computational molecular biology*, page 83. ACM, 1999.

[7] S. Chaudhuri. Data mining and database systems: Where is the intersection? *Data Engineering Bulletin*, 21(1):4–8, 1998.

[8] G. Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, page 145. Morgan Kaufmann Publishers Inc., 2002.

[9] H. Friedman and J. Rubin. On some invariant criteria for grouping data. *Journal of the American Statistical Association*, pages 1159–1178, 1967.

[10] D. Gelb, E. Oliver, and S. Gilman. Diagnostic criteria for Parkinson disease. *Archives of Neurology*, 56(1):33, 1999.

[11] H. Ghasemzadeh, J. Barnes, E. Guenterberg, and

R. Jafari. A phonological expression for physical movement monitoring in body sensor networks. In *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008*, pages 58–68, 2008.

[12] H. Ghasemzadeh, V. Loseu, and R. Jafari. Collaborative signal processing for action recognition in body sensor networks: a distributed classification algorithm using motion transcripts. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 244–255. ACM, 2010.

[13] G. Guerra-Filho, C. Fermuller, and Y. Aloimonos. Discovering a language for human activity. In *Proceedings of the AAAI 2005 Fall Symposium on Anticipatory Cognitive Embodied Systems, Washington, DC*, 2005.

[14] G. Guimarães and L. Pereira. Inferring definite-clause grammars to express multivariate time series. *Innovations in Applied Artificial Intelligence*, pages 332–341, 2005.

[15] S. Inenaga, H. Bannai, A. Shinohara, M. Takeda, and S. Arikawa. Discovering best variable length don't care patterns. In *Discovery Science*, pages 169–216. Springer.

[16] X. Jin, L. Wang, Y. Lu, and C. Shi. Indexing and mining of the local patterns in sequence database. *Intelligent Data Engineering and Automated LearningŮIDEAL 2002*, pages 39–52.

[17] K. Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys (CSUR)*, 24(4):439, 1992.

[18] S. Kurtz. Approximate string searching under weighted edit distance. In *Proc. of Third South American Workshop on String Processing*, pages 156–170. Citeseer, 1996.

[19] M. Lapinski, E. Berkson, T. Gill, M. Reinold, and J. Paradiso. A Distributed Wearable, Wireless Sensor System for Evaluating Professional Baseball Pitchers and Batters. In *2009 International Symposium on Wearable Computers*, pages 131–138. IEEE, 2009.

[20] V. Levenshteiti. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics-Doklady*, volume 10, 1966.

[21] C. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of hlt-naacl*, volume 2003, 2003.

[22] B. Lo, J. Wang, and G. Yang. From imaging networks to behavior profiling: Ubiquitous sensing for managed homecare of the elderly. In *Adjunct Proceedings of the 3rd International Conference on Pervasive Computing*. Citeseer, 2005.

[23] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.

[24] U. Manber and G. Myers. Suffix arrays: A new method for on-line string searches. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 319–327. Society for Industrial and Applied Mathematics, 1990.

[25] M. Marin-Perianu, C. Lombriser, O. Amft, P. Havinga, and G. Troster. Distributed activity recognition with fuzzy-enabled wireless sensor networks. *Distributed Computing in Sensor Systems*, pages 296–313.

[26] A. Marzal and E. Vidal. Computation of normalized edit distance and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 926–932, 1993.

[27] M. Masud, L. Khan, and B. Thuraisingham. A scalable multi-level feature extraction technique to detect malicious executables. *Information Systems Frontiers*, 10(1):33–45, 2008.

[28] R. Mayagoitia, A. Nene, and P. Veltink. Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems. *Journal of Biomechanics*, 35(4):537–542, 2002.

[29] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley New York, 1997.

[30] G. Milligan and M. Cooper. A study of standardization of variables in cluster analysis. *Journal of Classification*, 5(2):181–204, 1988.

[31] T. Pavlidis. Structural pattern recognition. 1977.

[32] F. Pereira, Y. Singer, and N. Tishby. Beyond word n-grams. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 95–106, 1995.

[33] N. Ravi, N. Dandekar, P. Mysore, and M. Littman. Activity recognition from accelerometer data. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1541, 2005.

[34] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[35] K. Sadakane. Compressed text databases with efficient query algorithms based on the compressed suffix array. *Algorithms and Computation*, pages 295–321.

[36] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[37] C. Shannon. Communication theory of secrecy systems. *MD Computing*, 15(1):57–64, 1998.

[38] V. Shnayder, B. Chen, K. Lorincz, T. Fulford-Jones, and M. Welsh. Sensor networks for medical care. In *SenSysŠ 05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 314–314. Citeseer, 2005.

[39] N. Stergiou. *Innovative analyses of human movement*. Human Kinetics Publishers, 2004.

[40] A. Volmer, N. Kruger, and R. Orglmeister. Posture and Motion Detection Using Acceleration Data for Context Aware Sensing in Personal Healthcare Systems. In *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany*, pages 71–74. Springer, 2009.

[41] K. Yamaoka, T. Nakagawa, and T. Uno. Application of Akaike's information criterion (AIC) in the evaluation of linear pharmacokinetic equations. *Journal of Pharmacokinetics and Pharmacodynamics*, 6(2):165–175, 1978.

[42] Y. Zhang and S. Vogel. Measuring confidence intervals for the machine translation evaluation metrics. *Proceedings of TMI*, 2004:85–94, 2004.