A Mining Technique Using *n*-Grams and Motion Transcripts for Body Sensor Network Data Repository

To get efficient use of large amounts of body sensor data, the authors represent human movement data using clustering, and they propose a technique to analyze sensed physiological signals.

By Vitali Loseu, *Student Member IEEE*, Hassan Ghasemzadeh, *Member IEEE*, and Roozbeh Jafari, *Member IEEE*

ABSTRACT | Recent years have witnessed a large influx of applications in the field of cyber-physical systems. An important class of these systems is body sensor networks (BSNs) where lightweight embedded processors and communication systems are tightly coupled with the human body. BSNs can provide researchers, care providers and clinicians access to tremendously valuable information extracted from data that are 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 the large amount of sensing data. To address this issue, we propose a data mining approach inspired by the experience in the areas of text and natural language processing. We represent sensor readings with a sequence of characters, called motion transcripts. Transcripts reduce complexity of the data significantly while maintaining morphological and structural properties of the physiological signals. To further take advantage of the physiological signal's structure, our data mining technique focuses on the characteristic transitions in the signals. These transitions are efficiently captured using the concept of *n*-grams. To facilitate a lightweight and fast mining approach, we reduce the overwhelmingly large number of *n*-grams via information gain (IG) feature selection. We report the effectiveness of the proposed approach in terms of the speed of mining while maintaining an acceptable accuracy in terms of the *F-score* combining both precision and recall.

KEYWORDS | Body sensor networks (BSNs); data mining; *n*-grams; Patricia tree; string templates

I. INTRODUCTION

Body sensor networks (BSNs) are becoming an increasingly popular field of research for a variety of applications ranging from fall and posture detection [1], [2] and telemedicine to rehabilitation and sports training [3], [4]. These systems are composed of lightweight wearable sensors that capture different physiological data from the human body. This physiological data may include an inertial description of human movements, electrocardiograph (ECG) readings of the human heart, electromyography (EMG) readings of the muscle activity, skin conductance level, blood pressure, and many more. Physiological signals of these modalities can be observed by sensors mounted on wearable devices. However, the modern sensing platforms are not perfect. Along with the useful physiological information they also capture noise and other data collection artifacts. Data collection artifacts

Manuscript received March 4, 2011; revised June 22, 2011; accepted June 23, 2011. Date of publication August 22, 2011; date of current version December 21, 2011.

V. Loseu is with the Computer Engineering Department, University of Texas at Dallas, Richardson, TX 75075 USA (e-mail: vitali.loseu@gmail.com).

H. Ghasemzadeh was with the Computer Engineering Department, University of Texas at Dallas, Richardson, TX 75075 USA. He is now with the West Wireless Health Institute, La Jolla, CA 92037 USA (e-mail: hghasemzadeh@gmwhi.org).

R. Jafari is with the Electrical Engineering Department, University of Texas at Dallas, Richardson, TX 75075 USA.

Digital Object Identifier: 10.1109/JPROC.2011.2161238

are abnormalities in the signal that can be introduced by the specific sensor deployment conditions. For example, the type of a strap used to attach a sensor to the body can significantly affect the recorded observation. Additionally, from the high level perspective, similar movements may look the same, and the specific movement execution can introduce variations in the senors observations of those movements. For example, a *sit-to-lie* movement can be performed smoothly, or the subject can throw themselves on the bed and briefly bounce on the mattress. While both movements achieve the same goal, from the inertial sensor perspective they do not look exactly the same.

A. Need for BSN Data Repository

BSN platforms 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 sufficient attention is storing and tracking the collected data. The data collected from these wearable systems are especially valuable 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 [5], which do not have a cure or even a quantitative, objective diagnostic process [6]. Parkinson's disease is a neurological disorder, however many of its symptoms, such a slow automatic movements (for example, blinking), inability to finish some movements, impaired balance while walking, muscle rigidity, and varies tremors, severely affect human movements and can be observed with the help of inertial sensors. The task is aided by the fact that many devices in our daily lives, such as cell phones, already have inertial sensors built-in. Furthermore, the seamless nature of BSN nodes allows their deployment prior to serious health problems to monitor the onset of the condition. The following example demonstrates the usefulness of the idea. A person can be monitored for an extended period of time (e.g., multiple years) with the help of a few sensor nodes. All of the data are simply collected and stored in the data repository. After some time, this person is diagnosed with a disease that involved gait abnormalities. It would be beneficial to analyze old data and extract gait parameters for disease evaluation, examination of disease progress, and treatment. 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.

B. Capturing Signal Structure

During data collection researchers aim to minimize the number of nodes attached to a subject to improve system wearability. 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. The 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 artificial 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 likely to be observed during walking trials. To avoid this, the structure of the compared movement needs to be investigated. 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 the main assumption of the structural pattern recognition [7]. This idea is important, because it suggests that while observations may not match in their entirety, due to data collection artifact and individual subject performance, they still have a significant structural similarity and can be compared by extracting representative signal properties. 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.

C. Design Challenges

BSN sensor nodes are highly constrained in terms of memory, processing resources, and battery lifetime. This means that all of the collected data cannot be stored on the wearable device, communicated wirelessly for an indefinite amount of time, or processed with complicated and possibly slow computational approaches on the device itself. At the same time, they have a potential to produce very large data sets over time. This suggests that the data representation approach needs to significantly reduce the complexity of the data, while maintaining the characteristic structure of the signal. This task is further complicated by the possibility of errors in the signal and intersubject variability in movement performance. This problem can be solved by applying limited processing to the sensor data, as it is being collected, to reduce its size and complexity. This step, however, needs to preserve the structural parameters of the signal. This can be achieved by applying limited processing that exclusively focuses on identifying transitions in the signal that uniquely characterize each movement. For this step to be successful, it is essential for the system to extract the properties of the signal capable of capturing such characteristic transitions. While in other systems, redundancy may be acceptable and even desirable, the resource and time constraints of the BSNs demand that the considered set of signal properties be minimal. That means that some of the machine learning and signal processing techniques may not be suitable for implementation on the sensor nodes.

With these requirements in mind, we present a data mining model for large BSN data repositories. Our approach has two steps. 1) We first propose a solution for the problem of data representation. For this, we define a technique for movement primitive construction from multidimensional physiological signals using clustering algorithms. We explore algorithms that preserve the original structure of the signal, even if human movements have timing inconsistencies. Unlike other works, we consider multiple clustering techniques for primitive construction using a small and computationally simple feature set. We combined the constructed primitives with their timing properties to generate string transcripts to capture the relational information from the signal. 2) We then define a novel data mining model that explores structural and relational properties of the string transcripts. We use information gain (IG) to select the parts of transcripts that can best differentiate between movement, and then define a tree-based classifier 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. While we do not explicitly trade the accuracy for speed and simplicity, it is an important tradeoff we keep in mind during system design decisions.

II. RELATED WORKS

The goal of data mining is identifying relevant objects. The relevancy of an object may be defined by some features of parameters or its similarity to other objects. This task is trivial in a well-structured and indexed database. However, when the data are not trivially structured, defining features is not obvious, or the measure of similarity is not defined for a specific object, this becomes challenging. It can be generally partitioned into phases. *Information retrieval* is the first phase where important, for a given application, information is extracted from a possibly noisy data. *Object summarization* is the second phase where each object of interest is defined in the contest of the relevant information, extracted during the first phase. The first phase combines information theory with the properties of the specific object type. The second phase tries to identify the best way to store and parse the metadata extracted during the first phase to efficient data mining. Before looking into the details of the BSN data mining, it is important to looking into a set of basic machine learning techniques often used in data mining problems.

A. Mining Techniques

The most simple classification rule based on a set of instances is called *1R* or *1-Rule*. In this approach, the system selects one attribute of the collected sensor readings and makes a classification decision based on it. While this is a very simple approach, it tends to work reasonably well for some applications [8]. The rule selection can be described as follows. For each possible attribute, the system can count how often each value of that attribute appears in any given class, and make an attribute-class assignment based on the most appearing value. Calculate the error of all of the attributes based on the cross-validation set, and select the attribute with least error. This algorithm faces two major issues. First, it may not be able to account for the values that are missing in the training set. Second, when an attribute has a large number of values it is prone to overfitting (or detect trends specific to the training data and not the desired observations).

Statistical modeling is a more involved approach to the problem. Instead of selecting only one attribute, the system can select all of the attributes, assuming that they are independent equally likely. Such an approach is known as *naive Bayes* and can perform well when its assumptions hold [9]. The approach has two major problems. First, it assumes that each of the attributes is independent, which is likely not the case for many real problems. This problem is extensively studied [10], [11] where the authors propose a semi-Bayes type of approaches to try and model actual data dependencies, correct data bias, and manage attribute weights. The second problem is the assumption that the attributes are normally distributed, which once again may not hold in many practical applications.

Another way to address the issue of different attributes having different weights is known as divide and conquer. Typically, that suggests creation of a tree-like structure, where each node corresponds to a specific attribute [12]. This way an attribute does not correspond to a whole level of the tree, meaning that at the same level different branches may use different attributes. This type of approach works in a top–down manner, which, at each level of the tree, seeks the best attribute to split the remaining data. The difficulty of this approach lies in selecting proper attributes at proper locations in the tree. It often uses feature selection algorithms such as IG [13], mutual information [14], and utility-based solution such as Bayesian information criterion (BIC) [15]. Shortcomings of these approaches are defined by their respective assumptions. For example, IG tends to work very well when attributes have very few possible values, preferring binary attributes [16]. IG performance decreases as the number of possible values increases due to the nature of the entropy calculations.

Previously described approaches work best with nominal attributes, however the idea can be extended to the numerical attributes as well. The most simple and relatively effective approach is known as linear regression, where the idea is to represent class values as linear combinations of the attributes and their respective weights [17]. The idea is to calculate proper weights during the training process, and apply the classifiers on the validation data. While this approach often works very well, it has a serious drawback. Namely, it assumes that the data can be modeled in a linear fashion, which may not be the case. This problem can be addressed with the help of logistic regression, and then evaluated with log-likelihood maximization [18]. A major problem with this approach is probabilities not adding up to 1 when the logistic regression is applied to multiple classes.

In the instance-based learning, the training trials themselves are used to evaluate unknown samples. It is done with the help of a distance function defined for the data in question. For classification purposes, the system measures the distance from an unknown trial to the training sample and selects the one with the shortest distance. A simple example of this learning time is 1-nearest-neighbor (1-NN) approach. However, it values each attribute equally just like naive Bayes. Additionally, a specific classification can be heavily affected by the outliers that do not represent the class well. These problems can be partially addressed by a *k*-NN-type solution, where instead of finding the nearest sample in the training data, the system looks for a consensus among *k*-NNs [19]. However, *k*-NN approaches are very slow compared to the competition.

Clustering algorithms are applied when there is no predetermined class to be detected, but rather the observed instances are split into natural groups. During the clustering step, the instances are combined together, based on strong resemblance, to form groups that can act as classes during the detection process. There are many approaches for clustering implementation, but they mainly focus on bringing the similar instances together, while separating the dissimilar instances. One of the most commonly used clustering approaches is known as *k*-means [20]. It takes the training instances and the number of desirable clusters as an input, and groups instances together based on their proximity. In the context of the Euclidean distance, the *k*-means approach iteratively minimizes the total squared distance from each instance to the cluster centers. It generally has two weaknesses. First, the best number of clusters is not always obvious, while a bad choice can result in improper grouping. Second, the iterative approach heavily depends on the initial selection of the cluster centers. Different random selections can result in significantly different clustering with no guarantee.

B. Structural Recognition in BSN

We next explore the structural data representation efforts for BSN data. In the context of BSN data, the idea of structural data representation and recognition is explored in [21]. This approach has a major weakness. The comparison evaluation is based on the value of Levenstein distance (or edit distance) [22]. 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 [23]. Another way to deal with this issue is to normalize the length of each primitive in motion transcripts [24]. 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 [25] as means to analyze vulnerability of ciphers but since then, they have been extensively used in the field of speech and text recognition.

n-grams [26] proved to be useful in structural parameter extraction when used for spoken language recognition [27]. *n*-grams can be used to capture phoneme, in the case of spoken language, and grammatical constructs, in case of written language, to identify bodies of speech or text. Similarly, *n*-grams can be used to analyze text summaries [28] or translation quality [26] with respect to co-occurrence statistics. While good at recognizing major structural differences, *n*-grams can also be used in the case of fine grain spelling error correction [29]. 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 the memory usage and the execution speed of the search. These important *n*-grams can be better organized with a suffix tree [30], which would increase the speed of identifying language constructs [31]. In fact, suffix trees are often used to index a large data store in the natural language processing and other fields. For example, in the field of molecular biology, DNA sequences can be indexed with the help of suffix trees [32]. Sadakane [33] discusses 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 [34].

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 [21]. While this simple approach seems to resolve the issue, it fails to recognize that each one of the sensing dimensions (or individual orthogonal sensing axis of sensors such as accelerometer) can observe variations such as changing speed and amplitude of the signal. In a text data set the variations are 1-D, 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 1-D primitive. It is possible that different combination of signal variations may hinder the structural consistency of the combined primitive representation.

III. PROBLEM OVERVIEW

In this section, we first describe the system used throughout this work. We then discuss the desired properties of the solution in the context of the defined system. Finally, we briefly introduce the pilot application used to evaluate our approach.

A. System Operation

The system, in this paper, consists of a set of wearable nodes placed on the human body to collect inertial observations of the human movements, and a computer that maintains the BSN repository and facilitates data organization and mining. In our case, a local computer has been used, however it can be replaced by a remote server, PDA, or even a smart phone without any loss of generality. The wearable nodes are connected to the computer via wireless radios. It is desirable to shift the computer functionality to the wearable nodes, and avoid using the battery for expansive wireless communication. While we keep this goal in mind during the approach design, it is out of the scope of this work.

The system begins operation by sampling the local sensors of the wearable devices. The local sensor data are communicated to the computer for processing. Information, relevant to a given application, is then extracted from the data on the computer side. The extracted information is used to construct a query to the BSN repository that can return observations already stored in the repository that most resemble the observed data. Fig. 1 illustrates the overall signal processing flow. The details of the system execution are described in the following sections and summarized in Section VII.

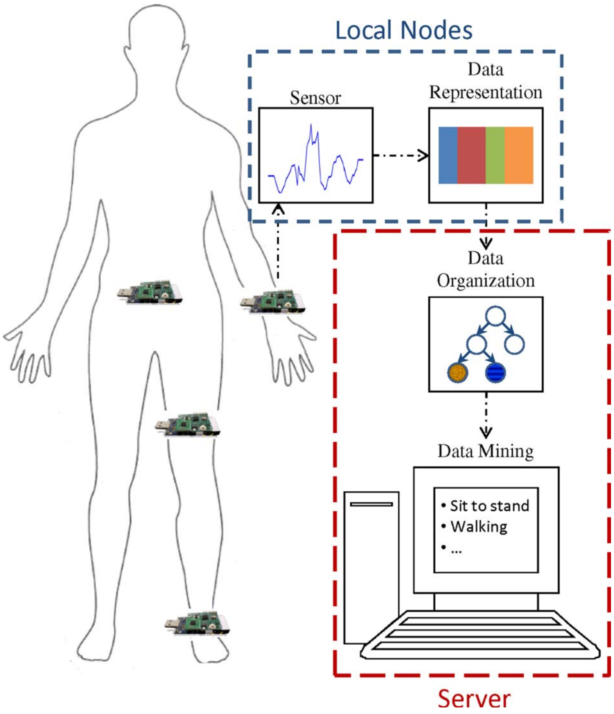


Fig. 1. ***BSN mining system overview.***

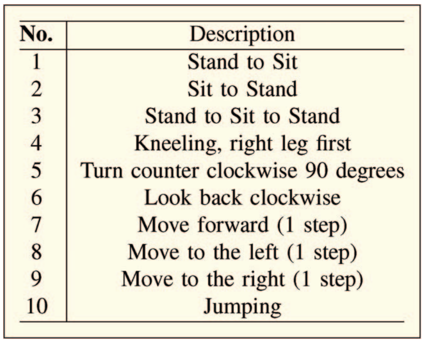
B. Hardware

During the experiment, subjects were equipped with several TelosB sensor nodes with custom-designed sensor boards. Each sensor board has a tri-axial accelerometer (providing *x*-, *y*-, and *z*-axis of acceleration) and a bi-axial gyroscope (providing *x*- and *y*-axis of angular velocity). Sensors were sampled at 50 Hz. This sampling frequency is high enough to provide acceptable resolution of the movements, and has been previously suggested by several other authors for physical movement monitoring applications [35], [36]. Furthermore, it satisfies the Nyquist criterion [37]. After collecting the data, each node sent its readings to the base station. In our case, the base station is a node without a sensor board, which forwards all of the received data to the PC via a USB connection for further processing.

C. Pilot Application

While we are designing an approach for data mining in a large BSN repository, a large data set 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 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

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

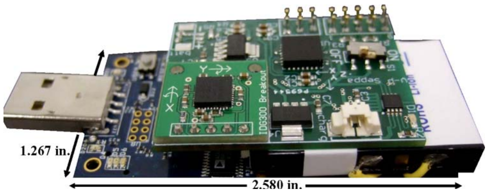


Fig. 2. *TelosB sensor node with a custom sensor board.*

subject repeated each movement ten times to increase the size of the data set. Each subject was equipped with nine sensor nodes positioned on both ankles, both thighs, both wrists and upper arms, and on the belt (as shown in Fig. 2).

D. Desirable Solution Properties

The focus of this study is to address the problem of data mining for the BSN data repositories. The system takes raw sensor readings as an input and performs a computationally efficient search in the repository for the signals similar to the input. Due to a potentially large size of the repository, the approach needs to be fast yet reliable. Consequently, we focus on the speed and simplicity of the approach. Sensor readings can be viewed as observations made by the system. Upon receiving an observation as the input, the search approach should be able to identify a movement to which the observation belongs, so that 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 needs to identify movements that contain certain instances of the signal, for example, identify all the movements where the torso moves forward.

IV. 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. A common way for unsupervised data grouping is clustering. We follow the idea introduced in [21] 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.

A. 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 combined 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 of all of the axis are independent of each other, aligning them with respect to time can significantly change the structure of the combined primitives. Fig. 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. The figure illustrates two trials of the same movement from the perspective of two different sensory axis. While the signals in the second trial have the same structure as the signals in the first trial, as demonstrated by the individual transcripts, their timing is inconsistent with the signals in the first trial. The bottom part of the figure demonstrates the combined transcript generated from two individual transcripts. Large vertical blocks correspond to the parts

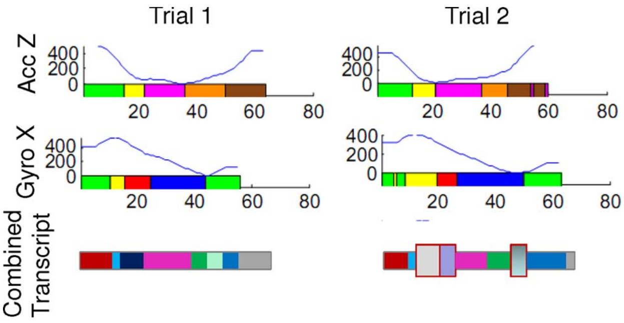


Fig. 3. *Signal alignment issue with respect to time.*

of the combined transcript of the second trial that do not have a corresponding counterpart in the first trial, which suggests that, if the primitive transitions are not aligned in the original signals, the time aligned combination of these signals may not be consistent between both trials. We also expect to have noise in the inertial data our sensor nodes collect, which will introduce another source of error.

To avoid the issue with alignment, we consider 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 classifier. This approach has an additional benefit of 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.

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 feature set that would produce good results. The resultant clusters should be able to identify enough transitions in the signal so that each movement of interest would be characterized with a unique subset of such transitions. 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 intersubject differences in features, the system normalizes features with respect to each subject using standard score (or *z*-score) [38]. While we used these specific 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 [39]. 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 spirit, this approach is similar to expectation maximization in Gaussian mixture models (GMMs) [40], 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 instead of artificially selecting points in the training set as cluster centers. GMM clustering computes the probability that any given point is assigned to every individual cluster and makes an assignment that maximizes its likelihood of such assignment. We selected the above approaches because of their computational simplicity, and the property that they are trying to identify cluster centers of the data set without any prior knowledge of the data. Both approaches start with random cluster centers and reevaluate 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 both k-means and GMM models. In case of k-means, we made the decision based on cluster silhouette [41]. 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

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

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 $b(i) = \min(b_j(i)), \forall j$.

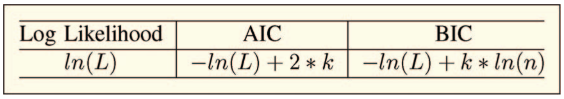
Silhouette $s(i)$ describes how well the point$i$ is mixed with the similar data points and is separated from the different data points. As a result, the quality of a clustering model with k clusters and d training points can be evaluated as

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

The larger the average silhouette value, the better is the model. Therefore, the best value of k can be selected by finding the largest Quality(k) [41].

In case of the GMM, we used expectation–maximization (EM) [42] 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) [43], and BIC [44]. Table 2 demonstrates the difference between quality estimation models for a GMM

Table 2 Quality Estimation of GMM-Based Clustering Using EM



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

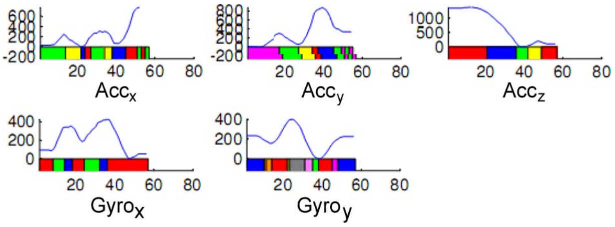


Fig. 4. ***Sample transcripts for three-axis accelerometer and two-axis gyro.***

with k clusters, maximum likelihood of the estimated model L, and n points in the training set.

Log likelihood just reports the likelihood of the model, and 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.

B. Motion Transcripts

Each movement can be described as a series of primitives. When an unlabeled movement needs to be classified, the system can extract features from each of its data points, 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. Fig. 4 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.

V. COMPARISON METRIC

Once the BSN data are converted to motion transcripts the system requires an efficient way to classify and search them. In Section II, we discussed edit distance, a common approach to compare strings. However, edit distance does not perform well when the input data have noise and vary in length. Additionally, the edit distance calculation is very slow with order of $O(n^2)$, where n is the 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 simple. 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.

A. *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. IG has proven to be effective in the field of natural language processing [45]. IG 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 with a value of "0" 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

$$\mathrm{Gain}(m, f) = H(m) - H(m|f) \tag{3}$$

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 good at identifying one movement while being unable to differentiate between the rest of the movements. That feature would have a bad general IG, however if we compute IG with respect to each movement, we can identify good features for each movement. Practically, this means that while computing IG of a feature with respect to particular movement m_i, the movement set is split into subsets of $\{m_i\}$ and $\{\text{'not' } m_i\}$ 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 IG as

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

$H(m_i)$ represents the amount of expected information that set m carries itself with respect to movement m_i.

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.

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 [46] and is selected for simplicity. IG performance can also suffer from movement or subject-specific signal variations. For example, if a subject has a consistent way of performing a movement, which differs from other subjects, the IG may select the subject-specific transitions as characteristic for the whole movement. In reality, these transitions will not be observed from any other subject, and represent an overfitting problem. This problem can be addressed by disqualifying *n*-grams that do not appear in enough training trials before the IG is applied. This step makes selecting a clustering technique that creates a sufficient number of clusters even more important. If the number of primitives is low, selecting only the often appearing *n*-grams, before the IG is applied, is likely to result in almost identical *n*-gram subsets selected for every movement.

VI. 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 [47]; more specifically, we used the 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 correspond to all the possible permutations of the combined *n*-gram set. This idea is illustrated in Fig. 5, where a sample Patricia tree is generated for six movements. The path “*BBB*,” “*AEE*,” and “*EBB*” corresponds a sit-to-stand movements.

Once the Patricia tree is created, each leaf of the Patricia tree corresponds to a subset of the movements. During testing, we use the *n*-grams of the test trial to traverse the tree and return the corresponding movement set. It may be an empty set or it may contain one or more movements. Specifically, if not enough *n*-grams are present to traverse the tree to a leaf, the system returns all movements assigned to the leafs of the subtree rooted at the node where the traversal terminated. For example, in Fig. 5, if the traversal would terminate at Node$_2$, then the set containing {sit-to-stand, stand-to-sit, sit-to-lie-to-sit, bend and grasp} is reported as the answer. If the traversal would terminate at Node$_5$, then only the set containing {sit-to-lie-to-sit, bend and grasp} is reported. Finally, if the traversal terminates at a leaf Node$_1$ 1, then the system reports only *step backward* as the answer.

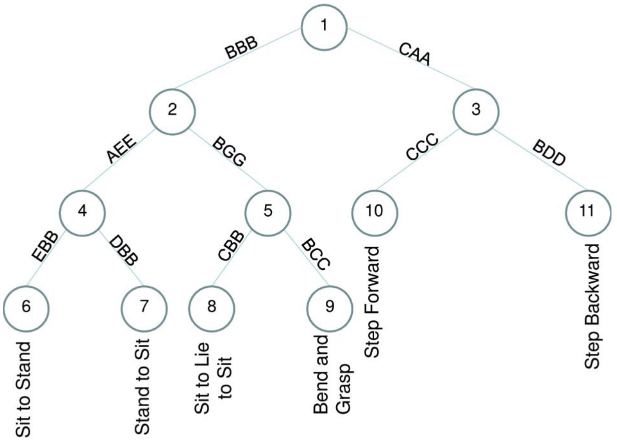


Fig. 5. *A sample Patricia tree for six movements.*

VII. 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*.

A. 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) [38] in order to remove intersubject variations of the same movements. Then, normalized features are used to define data clusters as described in Section IV-A1. 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 IV-B. 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 the IG as described in Section V-A. Finally, the system constructs a Patricia tree with selected *n*-grams on the edges and movement classes on the leafs as described in Section VI. The overall process is demonstrated in Fig. 6. The parameters defined during the training are data

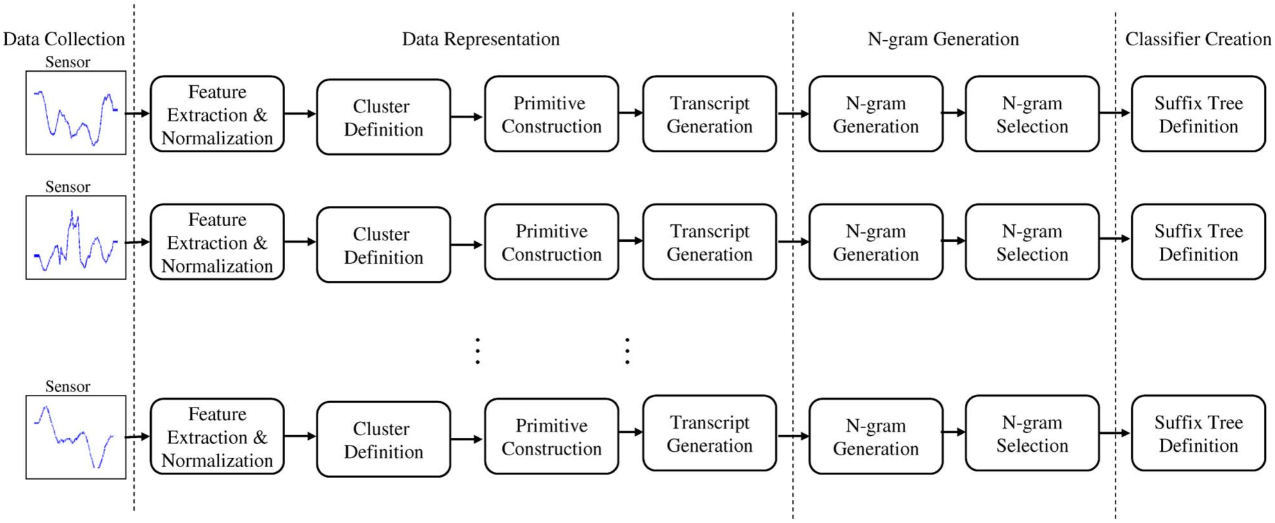


Fig. 6. *System training flow.*

clusters for each sensing axis of the motes, *n*-grams selected with respect to the IG criteria, and Patricia trees for classification. Clusters are represented by the cluster center coordinates, while important features selected for each sensing axis of each motes are then combined and stored.

B. 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. Primitives are combined with respect to timing alignment into motion transcripts. 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. If the traversal terminated at a node *p* that is not a leaf, the algorithm returns all the movements assigned to the leafs of the subtree rooted at node *p*. This allows the system to avoid introducing bias for the axis that observe the same signal for two different movements. Since all of those operations are defined in terms of individual sensing axis, an approach is required to combine the local decisions. In this work, we employ a simple voting scheme, which performs well in the context of our data. However, this method can be improved by treating each sensory axis as an individual classifier. To make a final decision, the individual classifiers can be combined in an intelligent way such as AdaBoost [48]. 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 Fig. 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 3-D gyroscope, while another may use only a 2-D 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.

VIII. EXPERIMENTAL RESULTS

To verify the performance of our approach we apply it to a pilot application discussed in Section III-C. The pilot application can be split into two problems. The first problem is locating the correct place in the repository to store a signal for an unknown trial. To achieve it, we split the available data into two equal sets. The first half of the data is used to train the system, while the second half of the data is used to verify the classification accuracy of the approach. The second problem is creating a representative signal template for each movement, and being able to search the entire repository for the trials consistent with the template. We start by training the system on the entire data set. During the training, the system selects *n*-gram sets, or templates, characteristic to each movement. We then use individual templates to search for relevant trials in the entire repository. While evaluating the first problem, we also compare the results of classification

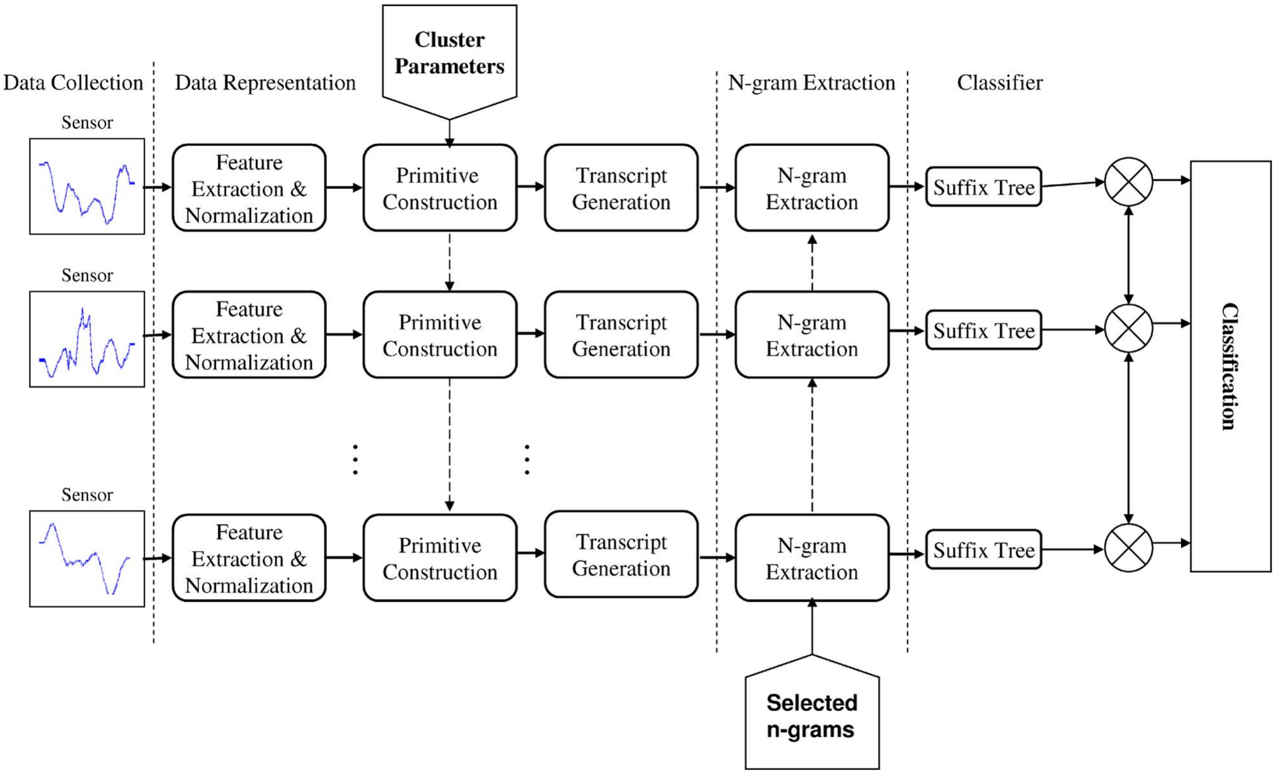


Fig. 7. ***Query processing flow.***

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 tradeoff with respect to n and t. For the second problem, we first evaluate how often each of the templates appears in the repository on average. We then consider the computed templates in the context of individual trials. If a trial contains enough n-grams from a given template, it is accepted as the movement represented by that template. Otherwise the trial is rejected. We report the accuracy of trial classification for each of the movement templates.

A. 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 3, 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 until the overfitting 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.

B. 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 $t = \{1, 2, \ldots, 5\}$ with GMM clustering model. We tested our model by splitting the data into two parts. Half of the data was used to train the system, while the other half was used to test it. The results of the classification are demonstrated in Table 5. The table contains the *F-score* defined as $(2 \times P \times R)/(P + R)$, where P is the classification precision, and R is a classification recall. This table confirms that adding more n-grams would improve both average precision and average recall until an

Table 3 {3–7}-Gram Average Performance of k-Means Versus 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

Table 4 3-Gram Average Performance of *k*-Means Versus 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

overfitting 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. Table 5 displays 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.

C. Parameter Tradeoff

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 overfitting point is reached. After the overfitting 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 overfitting problem or training on the trial but not model specific *n*-grams should happen sooner. We expect the system to converge to the best accuracy faster for large points of *n* but it also means that the overfitting point will happen faster as well. Table 6 demonstrates *F-score* of accuracy versus the number of *n*-grams *t* for different values of *n*.

Table 5 Classification Precision With Respect to the Number of Features Selected

Movement	Features per sensing axis				
	1	2	3	4	5
Stand to Sit	1	1	.958	.958	.857
Sit to Stand	1	1	.958	.958	.925
Stand to Sit to Stand	1	1	1	1	.958
Kneeling, right leg first	.80	.857	.907	.907	.958
Turn 90 degrees	.925	1	.958	.958	.958
Look back clockwise	.958	.925	1	.958	.958
Move forward	1	1	1	1	1
Move to the left	.958	1	1	1	1
Move to the right	.857	.868	.907	.83	.868
Jumping	1	1	1	1	1
Average	.95	.965	.969	.957	.958

Table 6 Precision With Respect to *n*-Gram Size, and Number of *n*-Grams Selected

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

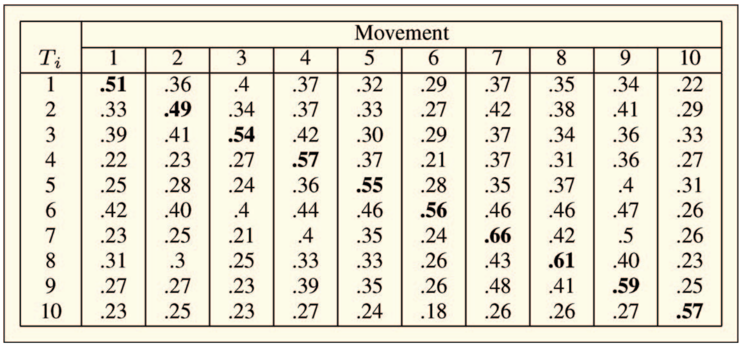
From the tables it is clear that higher values of *n* are desirable before overfitting, which means that *n* should be determined based on the expected amount of noise in the original signal. For the lower amount of noise a higher value of *n* would work better, while when the amount of noise is large, low values of *n* will provide a safer solution with less risk of overfitting. 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.

D. Movement Template Evaluation

Based on the parameters selected in Section VIII-C, we define T_i as combination of sets of 3-grams selected for each sensing axis during the training process for movement M_i. Unlike the previous problem, we use the entire data set to train the templates. Once the training is complete and the templates are generated, we evaluate the average quality of each T_i. It is done by checking how often the *n*-grams of each T_i appear in movement trials of every movement. Intuitively, *n*-grams of T_i should appear more often in M_i than any other movement, in order for the template to be effective. Table 7 demonstrates the results of this evaluation normalized with respect to the size of each template, meaning that on average 51% of T_1 appear in trials of M_1, while only 36% of T_1 appear in trials of M_2.

Two observations can be made based on the results in Table 7. First, it is clear that *n*-grams of T_i appear most often in the M_i itself. This observation is in line with our expectation from a good template. While *n*-grams of the T_i, on average, appear 10% more often in the respective M_i, they also appear in trials of other movements a sizable amount. This result suggests that a closer look on a per-trial basis is required to evaluate the template quality. The intuitive approach to this problem is to search for trials that have the entire template present. However, in a realistic system that contains some noise this solution is

Table 7 Average Template Evaluation Normalized With Respect to the Template Size



T_i	Movement									
	1	2	3	4	5	6	7	8	9	10
1	**.51**	.36	.4	.37	.32	.29	.37	.35	.34	.22
2	.33	**.49**	.34	.37	.33	.27	.42	.38	.41	.29
3	.39	.41	**.54**	.42	.30	.29	.37	.34	.36	.33
4	.22	.23	.27	**.57**	.37	.21	.37	.31	.36	.27
5	.25	.28	.24	.36	**.55**	.28	.35	.37	.4	.31
6	.42	.40	.4	.44	.46	**.56**	.46	.46	.47	.26
7	.23	.25	.21	.4	.35	.24	**.66**	.42	.5	.26
8	.31	.3	.25	.33	.33	.26	.43	**.61**	.40	.23
9	.27	.27	.23	.39	.35	.26	.48	.41	**.59**	.25
10	.23	.25	.23	.27	.24	.18	.26	.26	.27	**.57**

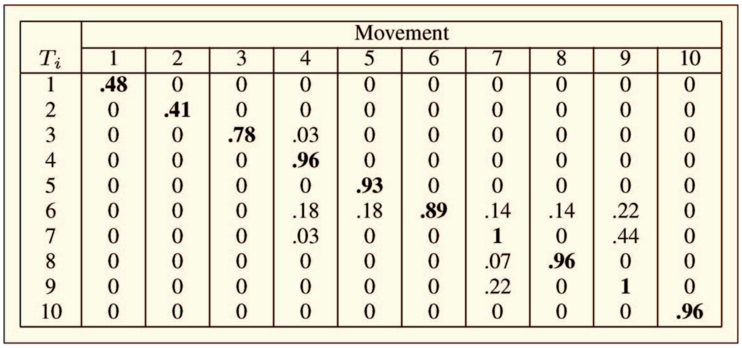
not practical, since it is unlikely that many trials will be perfect. Table 7 confirms this; the highest average values for each T_i are close to 50% and not 100%. We solve this problem by introducing a variable α. α defines the proportion of the number of n-grams from a T_i that need to be present in a trial for it to be classified as M_i. If the α value is too low, it is likely that some trials will be erroneously identified as M_i, increasing the number of false positive errors and decreasing the precision of the template. However, if the α value is too high, some trials of the M_i will not be identified, increasing the amount of false negative error and decreasing the recall of the template. Additionally, lower values of α can speed up the computation, which is desirable in our problem due to a potentially large number of movements in the repository. To achieve a balance between the precision and recall, as well as to promote faster data mining, we select the value of $\alpha = .5$. The value of 50% is also suggested by Table 7, where the average presence of more than 50% n-grams from a template T_i identifies the movement M_i, associated with that template. Table 8 demonstrates the normalized number of trials of each movement identified as M_i by each of the templates T_i, meaning, for example, that template T_1 selects 78% of trials belonging to M_1.

With $\alpha = 0.5$, each T_i correctly identifies substantially more trials of its own movement than any other movement. This low value of α also defines a search speed increase of up to 50% since only 50% of the trials in templates need to be located. However, a some number of false negative and false positive errors are made, which suggests that a static value of α is inappropriate. Templates T_1 and T_2 do well at identifying only trials of their respective movements. However, less than a half of the appropriate trials is identified. This suggests that the value of $\alpha = 0.5$ is too high for these templates. At the same time, T_6 and T_7 have a much better rate of recognizing trials of their respective movements, but they also falsely identify trials of other movements. This suggests that the value of $\alpha = 0.5$ is too low for these templates. Defining a movement-specific value of α_i, based on the training set, for each template, can decrease the amount of errors in the system.

IX. 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 IG approach to the features to select k features that provide the most information about each sensing axis. Based on the

Table 8 Template Evaluation for Individual Trials Normalized With Respect to the Number of Trials of Each Movement



T_i	Movement									
	1	2	3	4	5	6	7	8	9	10
1	**.48**	0	0	0	0	0	0	0	0	0
2	0	**.41**	0	0	0	0	0	0	0	0
3	0	0	**.78**	.03	0	0	0	0	0	0
4	0	0	0	**.96**	0	0	0	0	0	0
5	0	0	0	0	**.93**	0	0	0	0	0
6	0	0	0	.18	.18	**.89**	.14	.14	.22	0
7	0	0	0	.03	0	0	**1**	0	.44	0
8	0	0	0	0	0	0	.07	**.96**	0	0
9	0	0	0	0	0	0	.22	0	**1**	0
10	0	0	0	0	0	0	0	0	0	**.96**

selected *n*-grams we build a suffix tree for fast query and identification of movements in the database. We demonstrate that the system can achieve average *F-score* of 97% on our pilot data with the help of only one characteristic for each movement transition. We also explored the tradeoff between the length of the extracted *n*-grams and the required number of features for the best classification results. ■

REFERENCES

[1] B. Lo, J. Wang, and G. Yang, "From imaging networks to behavior profiling: Ubiquitous sensing for managed homecare of the elderly," in *Proc. 3rd Int. Conf. Pervasive Comput.*, May 2005, pp. 101–104.

[2] A. Volmer, N. Kruger, and R. Orglmeister, "Posture and motion detection using acceleration data for context aware sensing in personal healthcare systems," in *Proc. World Congr. Med. Phys. Biomed. Eng.*, Munich, Germany, 2009, pp. 71–74.

[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," in *Proc. 4th Annu. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mar. 2006, pp. 527–531.

[4] 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 *Proc. Int. Symp. Wearable Comput.*, 2009, pp. 131–138.

[5] V. Shnayder, B. Chen, K. Lorincz, T. Fulford-Jones, and M. Welsh, "Sensor networks for medical care," in *Proc. 3rd Int. Conf. Embedded Netw. Sensor Syst.*, 2005, pp. 314–327.

[6] D. Gelb, E. Oliver, and S. Gilman, "Diagnostic criteria for Parkinson disease," *Archives Neurol.*, vol. 56, no. 1, pp. 33–39, 1999.

[7] H. Bunke and A. Sanfeliu, *Syntactic and Structural Pattern Recognition: Theory and Applications*. Singapore: World Scientific, 1990.

[8] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA: Morgan Kaufmann, 2005.

[9] E. Frank, M. Hall, and B. Pfahringer, "Locally weighted naive Bayes," Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, 2003.

[10] D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," in *Proc. Eur. Conf. Mach. Learn.*, 1998, pp. 4–15.

[11] I. Kononenko, "Semi-naive Bayesian classifier," in *Proc. Eur. Working Session Learn.*, 1991, pp. 206–219.

[12] J. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[13] J. Kent, "Information gain and a general measure of correlation," *Biometrika*, vol. 70, no. 1, pp. 163–173, 1983.

[14] I. Sethi and G. Sarvarayudu, "Hierarchical classifier design using mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-4, no. 4, pp. 441–445, Apr. 1982.

[15] W. Chou and W. Reichl, "Decision tree state tying based on penalized Bayesian information criterion," in *Proc. Int. Conf. Aacoust. Speech Signal Process.*, 1999, pp. 345–348.

[16] M. Rogati and Y. Yang, "High-performing feature selection for text classification," in *Proc. 11th Int. Conf. Inf. Knowl. Manage.*, 2002, pp. 659–661.

[17] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106–2112, Nov. 2010.

[18] A. Genkin, D. Lewis, and D. Madigan, "Large-scale Bayesian logistic regression for text categorization," *Technometrics*, vol. 49, no. 3, pp. 291–304, 2007.

[19] S. Rasheed, D. Stashuk, and M. Kamel, "Adaptive fuzzy k-NN classifier for EMG signal decomposition," *Med. Eng. Phys.*, vol. 28, no. 7, pp. 694–709, 2006.

[20] J. Hartigan and M. Wong, "A k-means clustering algorithm," *J. Roy. Stat. Soc. C*, vol. 28, no. 1, pp. 100–108, 1979.

[21] 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 *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2010, pp. 244–255.

[22] V. Levenshteiti, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics-Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[23] S. Kurtz, "Approximate string searching under weighted edit distance," in *Proc. 3rd South Amer. Workshop String Process.*, 1996, pp. 156–170.

[24] A. Marzal and E. Vidal, "Computation of normalized edit distance and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 926–932, Sep. 1993.

[25] C. Shannon, "Communication theory of secrecy systems," *MD Comput.*, vol. 15, no. 1, pp. 57–64, 1998.

[26] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *Proc. 2nd Int. Conf. Human Lang. Technol. Res.*, 2002, p. 138–145.

[27] A. Adami and H. Hermansky, "Segmentation of speech for speaker and language recognition," in *Proc. 8th Eur. Conf. Speech Commun. Technol.*, 2003, pp. 841–844.

[28] C. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist. Human Lang. Technol.*, 2003, vol. 1, DOI: 10.3115/1073445.1073465.

[29] K. Kukich, "Techniques for automatically correcting words in text," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 377–439, 1992.

[30] U. Manber and G. Myers, "Suffix arrays: A new method for on-line string searches," in *Proc. 1st Annu. ACM/SIAM Symp. Discrete Algorithms*, 1990, pp. 319–327.

[31] F. Pereira, Y. Singer, and N. Tishby, "Beyond word n-grams," in *Proc. 3rd Workshop Very Large Corpora*, 1995, pp. 95–106.

[32] S. Burkhardt, A. Crauser, P. Ferragina, H. Lenhof, E. Rivals, and M. Vingron, "Q-gram based database searching using a suffix array (QUASAR)," in *Proc. 3rd Annu. Int. Conf. Comput. Molecular Biol.*, 1999, pp. 77–83.

[33] K. Sadakane, "Compressed text databases with efficient query algorithms based on the compressed suffix array," *Algorithms and Computation*, ser. Lecture Notes in Computer Science, vol. 1969. Heidelberg: Springer, 2000, pp. 295–321.

[34] X. Jin, L. Wang, Y. Lu, and C. Shi, "Indexing and mining of the local patterns in sequence database," in *Proc. Intelligent Data Eng. Autom. Learn.*, 2002, pp. 39–52.

[35] N. Ravi, N. Dandekar, P. Mysore, and M. Littman, "Activity recognition from accelerometer data," in *Proc. Nat. Conf. Arti. Intell.*, 2005, vol. 20, no. 3, pp. 1541–1546.

[36] R. Mayagoitia, A. Nene, and P. Veltink, "Accelerometer and rate gyroscope measurement of kinematics: An inexpensive alternative to optical motion analysis systems," *J. Biomech.*, vol. 35, no. 4, pp. 537–542, 2002.

[37] N. Stergiou, *Innovative Analyses of Human Movement*. Champaign, IL: Human Kinetics, 2004.

[38] G. Milligan and M. Cooper, "A study of standardization of variables in cluster analysis," *J. Classification*, vol. 5, no. 2, pp. 181–204, 1988.

[39] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probab.*, 1967, vol. 1, pp. 281–297.

[40] H. Friedman and J. Rubin, "On some invariant criteria for grouping data," *J. Amer. Stat. Assoc.*, vol. 62, no. 320, pp. 1159–1178, 1967.

[41] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.

[42] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.

[43] K. Yamaoka, T. Nakagawa, and T. Uno, "Application of Akaike's information criterion (AIC) in the evaluation of linear pharmacokinetic equations," *J. Pharmacokin. Pharmacodyn.*, vol. 6, no. 2, pp. 165–175, 1978.

[44] G. Schwarz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.

[45] Y. Zhang and S. Vogel, "Measuring confidence intervals for the machine translation evaluation metrics," in *Proc. TMI*, 2004, vol. 2004, pp. 85–94.

[46] M. Masud, L. Khan, and B. Thuraisingham, "A scalable multi-level feature extraction technique to detect malicious executables," *Inf. Syst. Frontiers*, vol. 10, no. 1, pp. 33–45, 2008.

[47] S. Inenaga, H. Bannai, A. Shinohara, M. Takeda, and S. Arikawa, "Discovering best variable length don't care patterns *Proc. Int. Conf. Discovery Sci.*, 2002, pp. 169–216.

[48] P. Viola and M. Jones, "Fast and robust classification using asymmetric adaboost and a detector cascade," *Adv. Neural Inf. Process. Syst.*, vol. 2, pp. 1311–1318, 2002.

ABOUT THE AUTHORS

Vitali Loseu (Student Member, IEEE) received the B.S. and M.S. degrees in computer science from the University of Texas at Dallas, Richardson, in 2007 and 2008, respectively, where he is currently working towards the Ph.D. degree in computer engineering in the Embedded Systems and Signal Processing Laboratory (ESSP).

His research interests lie in system optimization for reconfigurable computing.

Hassan Ghasemzadeh (Member, IEEE) received the B.Sc. degree in computer engineering from Sharif University of Technology, Tehran, Iran, in 1998, the M.Sc. degree in computer engineering from University of Tehran, Tehran, Iran, in 2001, and the Ph.D. degree in computer engineering from the University of Texas at Dallas, Richardson, in 2010.

He then joined the West Wireless Health Institute as a Postdoctoral Fellow. His research interests lie in different aspects of embedded systems. He is currently working on collaborative signal processing, reconfigurable computing, and algorithm design for medical embedded systems.

Roozbeh Jafari (Member, IEEE) received the B.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2000, the M.S. degree in electrical engineering from the State University of New York (SUNY) at Buffalo, Buffalo, NY, in 2002, and the M.S. and Ph.D. degrees in computer science from the University of California at Los Angeles, Los Angeles, in 2004 and 2006, respectively.

He spent 2006–2007 in the Electrical Engineering and Computer Science Department, University of California Berkeley, Berkeley, as a Postdoctoral Researcher. He is currently an Assistant Professor in Electrical Engineering at the University of Texas at Dallas, Richardson. His research is primarily in the area of networked embedded system design and reconfigurable computing with emphasis on medical/biological applications, their signal processing and algorithm design. He is the Director of the Embedded Systems and Signal Processing Laboratory (ESSP).