

# Discovering Characteristic Actions from On-Body Sensor Data

David Minnen, Thad Starner, Irfan Essa, and Charles Isbell  
College of Computing  
Georgia Institute of Technology  
Atlanta, GA 30332 USA  
{dminn,thad,irfan,isbell}@cc.gatech.edu

## Abstract

We present an approach to activity discovery, the unsupervised identification and modeling of human actions embedded in a larger sensor stream. Activity discovery can be seen as the inverse of the activity recognition problem. Rather than learn models from hand-labeled sequences, we attempt to discover motifs, sets of similar subsequences within the raw sensor stream, without the benefit of labels or manual segmentation. These motifs are statistically unlikely and thus typically correspond to important or characteristic actions within the activity.

The problem of activity discovery differs from typical motif discovery, such as locating protein binding sites, because of the nature of time series data representing human activity. For example, in activity data, motifs will tend to be sparsely distributed, vary in length, and may only exhibit intra-motif similarity after appropriate time warping. In this paper, we motivate the activity discovery problem and present our approach for efficient discovery of meaningful actions from sensor data representing human activity. We empirically evaluate the approach on an exercise data set captured by a wrist-mounted, three-axis inertial sensor. Our algorithm successfully discovers motifs that correspond to the real exercises with a recall rate of 96.3% and overall accuracy of 86.7% over six exercises and 864 occurrences.

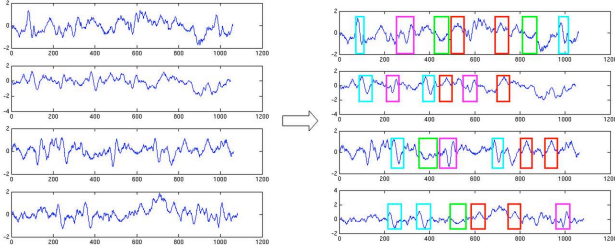
## 1 Introduction

Activity discovery describes the problem of identifying, locating, and modeling meaningful actions embedded in time series data representing different activities. We are specifically interested in discovering meaningful actions in human activity data captured by on-body sensors, though similar time series representing the behavior of other creatures (e.g., bees, ants, or robots) is also applicable. Researchers analyzing activity data have typically investigated

activity recognition. They decide *a priori* which activity components are important and either manually construct detectors or label training data for use with machine learning methods that estimate model parameters appropriate for recognition. Our goal is to invert this problem and to *discover* the important activity components through unsupervised analysis of the captured data.

Our motivation for addressing the discovery problem is three-fold: (1) in many situations, we may think that we know the important actions associated with an activity, and an unsupervised discovery system can either provide validation or suggest alternatives; (2) in cases where we do not know the low-level components of an activity, such a system can help identify them; and (3) we may not be able to identify or describe actions that are easily detected by the system's sensors, whereas discovered actions are necessarily identifiable within the given modality. Thus, the possible goals of an activity discovery system include validating our knowledge assumptions, generating new knowledge, and interpreting data from the perspective of new sensor modalities.

The benefits of an automated discovery system are extensive. Traditional arguments for unsupervised learning, such as reducing cost by precluding the need to manually label data, aiding adaptation to non-stationary patterns, and providing early exploratory tools, certainly apply [7]. In addition, we envision a range of applications specific to activity discovery. By learning typical patterns of behavior automatically, perceptual systems become less cumbersome to deploy and potentially more effective due to natural user and environment-specific adaptation. The ability to discover typical activities could enhance health monitoring systems by providing a characterization of typical healthy behavior specific to the observed location and to the worn sensors, which may be difficult for a human observer to fully specify. In cooperative environments such as homes and offices, an activity discovery system could also learn important routines and behaviors, thus enabling it to provide the models necessary to detect contextual information



**Figure 1. An illustration of the sparse motif discovery problem in one dimension. Rectangles correspond to motif occurrences and each color represents a different motif.**

for intelligent interfaces such as memory aids [19] and execution monitors [6].

We approach the activity discovery problem as one of sparse motif discovery in multivariate time series. In this context, a motif is a collection of relatively short subsequences that exhibit high intra-motif similarity yet are distinguishable from other subsequences (see Figure 1). Motif discovery, then, is the unsupervised identification, modeling, and localization of each motif and its occurrences in the time series. Specifically, no knowledge of the number of motifs, their shape (*i.e.*, model parameters, prototypical members, or representative examples), their locations in the time series, or the number of occurrences of each motif is available to the system. In the general case, which is typical for activity data, motifs will only be sparsely distributed. This means that much of the time series data is not part of any motif, which contrasts with densely distributed motifs and has important consequences for the practical efficiency of activity discovery algorithms. Furthermore, a realistic activity discovery system must operate efficiently on multivariate data for the simple reason that researchers have yet to find a single sensor or a derived feature that captures the range of human actions in any but the most trivial activities. Finally, discovery in time series, rather than in non-temporal sequences as in bioinformatics, introduces additional modeling complexities that must be considered by the discovery algorithm. For example, the occurrences of a single motif may have different lengths and may exhibit intra-motif similarity only after appropriate time warping.

The contribution of our work is to introduce and motivate activity discovery as an important method for learning about human activity from on-body sensor data and to present an approach that efficiently discovers actions in sparse, multivariate sensor streams. In the remainder of this paper, we situate our work relative to existing research, provide a detailed overview of our algorithm, and present empirical results that validate the approach. Finally, we outline several

improvements to our algorithm that we are currently investigating and discuss our plans for extending the approach.

## 2 Related Work

Much research has been done in the field of activity recognition. One focus has been on detecting gestures and relatively low-level activities such as running, walking, sitting, shaking hands, vacuuming, *etc.* For example, Westeyn *et al.* used hidden Markov models (HMMs) to recognize mimicked autistic self-stimulatory behaviors using three on-body accelerometers [22]. Bao and Intille compared several classification methods and on-body sensor positions to recognize 20 different activities collected under “semi-naturalistic” conditions [2]. Other researchers have focused on recognizing American Sign Language (ASL) [17, 21, 3]. Huynh and Schiele analyzed sensor readings from a multimodal, on-body sensor package in order to empirically evaluate different features and window lengths [10]. They found that discriminative features vary across activities but that FFT coefficients aggregated in exponential bands may be generally applicable. Lester *et al.* continued along this path and used boosted decision stumps to automatically select features and then used HMMs to capture temporal regularities [13]. Space constraints preclude a more complete review, but see Bao and Intille’s review for additional references in this area [2].

Previous work in motif discovery more closely relates to our research. In bioinformatics, for example, systems such as MEME were developed to discover motifs in DNA and protein sequences [1]. Many other specialized systems have been developed since then, though few are applicable to time series analysis since they were designed to work with categorical sequences (see [11] for a brief review). Recently, Jensen *et al.* generalized motif discovery over both categorical and continuous data and across arbitrary similarity metrics [11]. This represents a major improvement, but their approach has two significant issues. First, it requires a pairwise comparison between all subsequences of a given length. This operation is quadratic in the length of the time series, making it unable to scale to even medium-length data sets. Second, although they do support variable length motif occurrences, this is only achieved via a post-process. In sequence data this is probably not a major problem, but in time series, two very similar subsequences may appear quite different if only fixed-length segments are compared (*e.g.*, consider comparing ABCD and AABBCDD with a window length of four).

Within the data mining community, an efficient, probabilistic algorithm for motif discovery using locality-sensitive hashing was developed [4]. This approach only discovers fixed-length motifs in univariate data however. It gains its efficiency by first searching for similar pairs of subsequences using a discrete representation of the origi-

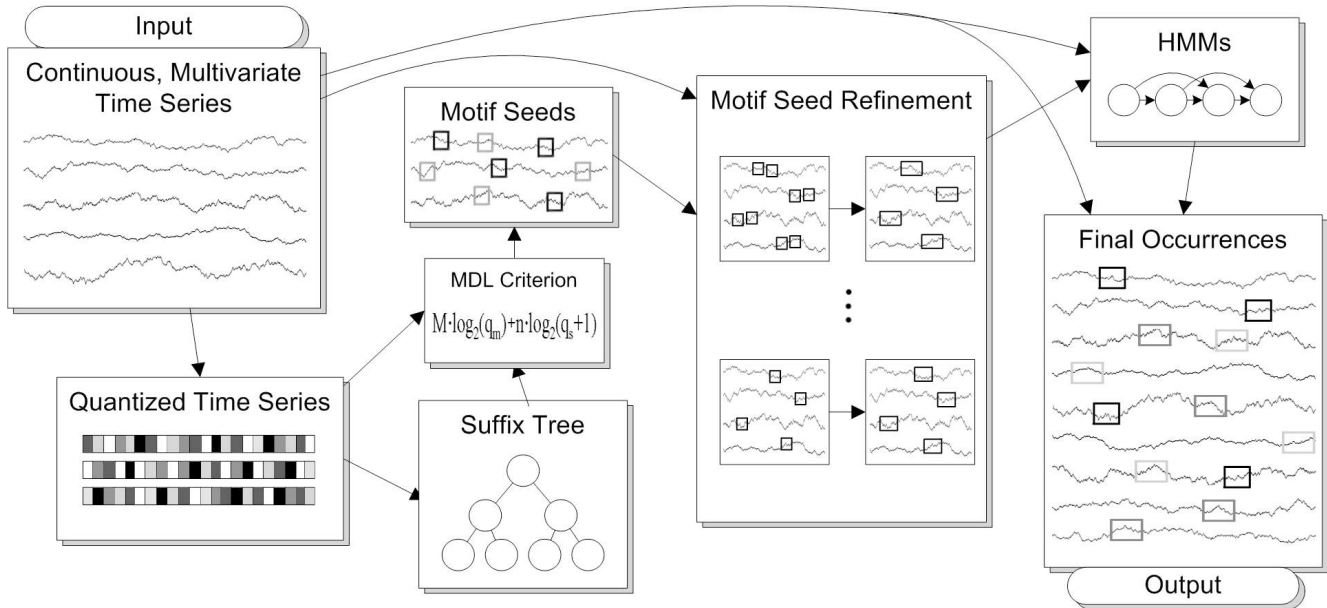


Figure 2. Algorithm Overview

nal, real-valued data. Local quantization is performed by symbolic aggregate approximation (SAX) [14], and then the most similar pair is used as a seed to find the remaining motif occurrences. Although this approach can be extended to operate on multivariate time series [12], the assumption that motifs have a fixed length along with the requirement that the user specify a threshold for intra-motif similarity, makes it difficult to use for activity discovery.

Tanaka and Uehara generalized the approach of Chiu *et al.* to work with multivariate time series and to allow variable length motifs [18]. Their solution, however, is simply to apply a univariate algorithm to the first principal component of the time series. Unfortunately, the first principal component will often not retain enough information about the original multivariate sequence to allow differentiation between different actions. Their use of a minimum description length (MDL) criterion to select variable-length motifs, however, appears to be quite powerful, and a similar technique is used in our approach (see Section 3.1).

The PERUSE algorithm discovers motifs directly in multivariate time series and allows non-linear time warping and variable-length motifs [15]. Impressively, it was able to learn many of the repeated words when applied to speech data and was also able to learn “episodes” from robot sensor data. The algorithm was developed for data with a dense motif structure, an assumption that holds for both the speech and robot sensor series. However, although the algorithm is applicable to sparse data, it scales quite poorly due to the need to exhaustively search for exemplars. In practice, sampling methods can speed up the exemplar search, but

this benefit decreases as the sparsity increases, eventually returning to an effectively linear search.

Although less closely related than the aforementioned motif discovery algorithms, much research in the computer vision and multi-modal pattern recognition communities focuses on data with dense structure in what can broadly be called scene detection. A particularly nice example is the work of Xie *et al.* in which the parameters of a hierarchical hidden Markov model (HHMM) are estimated in an unsupervised manner to learn different segments of soccer and baseball games [23]. The games are analyzed based on both video and audio features, and the hierarchical model naturally accounts for both low-level and high-level structure. In earlier work, Clarkson and Pentland explicitly used a two-layer HMM to model ambulatory audio and video data captured using on-body sensors [5]. Low-level transitions between the states of each HMM captured local information, while transitions between HMMs captured scene boundaries.

### 3 Algorithmic Details

Our approach to activity discovery is based on the formulation set forth by Oates [15], which frames the problem as a parallel estimation of hidden location probabilities and motif model parameters. The hidden variables,  $z_{S,t}$ , specify the likelihood of a motif occurrence ending at a particular time,  $t$ , in a sequence,  $S$ . Expectation-maximization (EM) is used to iterate between calculating the expected value of the  $z_{S,t}$  given the current model parameters and then estimating the optimal model parameters given the current oc-

currence location probabilities.

A key issue of any iterative algorithm, including EM, is initialization. Since no information about the motif occurrence locations is known *a priori*, initializing  $z_{s,t}$  is not possible. Instead, the PERUSE algorithm initializes the model parameters for the current motif from an exemplar. Since no true exemplars (*i.e.*, actual occurrences of the motif) are given, the algorithm uses a fixed-length sliding window to try all possible exemplars and then selects the one that leads to the highest overall occurrence probability as a real motif (see [15] for additional details).

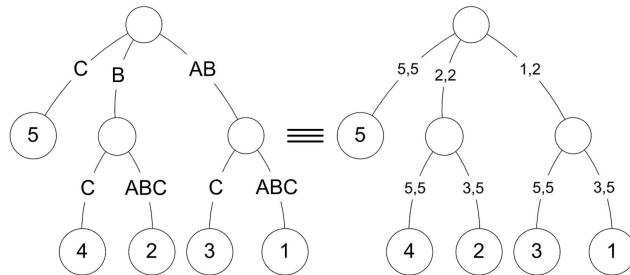
Although this is an intuitive, principled formulation, two problems do exist. First, the initial model must be learned from a single example. Although PERUSE mediates this issue somewhat by only searching for a small number of occurrences based on this model (and then re-estimating the model parameters based on these additional examples), no initial information about the allowable variance in either the observations or temporal warping can be learned from the single exemplar. In addition, if the motifs are sparse, a huge amount of time could be wasted searching for motifs based on false exemplars. Although these should eventually be discarded in favor of real motifs, the time is still lost.

To address these issues, our approach to activity discovery seeks to combine the efficiency benefits of searching quantized time series (as in Chiu *et al.*'s work [4]) with the powerful modeling abilities of probabilistic temporal models in a continuous domain (as in PERUSE). The algorithm proceeds in three main phases (see Figure 2 for a diagrammatic overview). In the first phase, it generates a set of seed motifs by greedily selecting the best motif, removing its occurrences, and then iterating until the next best motif fails to meet an information-theoretic criterion described in detail below. The seed motifs are then refined using information from the continuous time series, and, finally, a HMM is trained for each seed motif and corresponding occurrences are detected via a modified Viterbi alignment procedure.

### 3.1 Seed Motif Identification

In the initial phase of the approach, the multivariate data is quantized by fitting a mixture of Gaussian distributions to the frames of the time series. Thus, if the original data set consists of eight series with 1,000 frames each, then the mixture model would be fit to a data set of 8,000 points. Each mixture component is assigned a unique symbol, and each frame of the time series is mapped to the symbol corresponding to the closest mixture component. In the example above, this would lead to eight strings of length 1,000.

A generalized suffix tree is then built from the quantized sequences. A suffix tree is a linear space tree structure in which every path from root to leaf is a valid suffix (see Figure 3) [9]. As an example, if the string ABCD is used as



**Figure 3. A suffix tree for the string ABABC. Node annotations show the index of the suffix in the original string, while edge annotations show the substring either explicitly (left) or as an index range (right).**

input, then the suffix tree will hold the strings ABCD, BCD, CD, and D. Importantly, every subsequence of the original string is the *prefix* of a *suffix*, and so every subsequence is stored in the suffix tree starting at the root node.

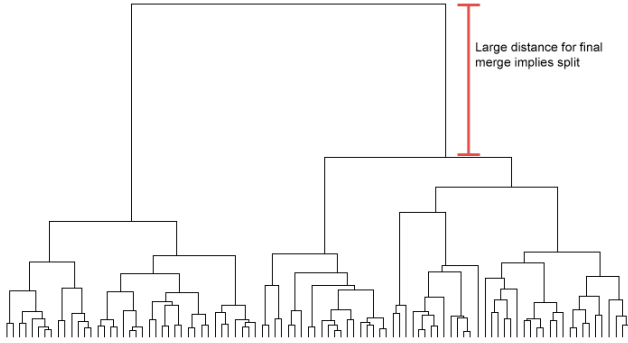
Ukkonen devised an algorithm to build a suffix tree in linear time [20], and his work was later extended to include linear time construction of generalized suffix trees which store multiple strings within a single tree [9]. Generalized suffix trees and the linear time construction algorithm provide a method to efficiently represent all of the quantized data and then rapidly search for common subsequences.

In order to rank the potential motifs and determine when to stop searching, each motif is scored according to an information-theoretic criterion similar to that used by Tanaka and Uehara. The criterion computes the change in description length of the original data sequence if the motif were encoded separately and every occurrence were replaced with a new symbol. This criterion serves to balance the number of occurrences of a particular motif with the complexity of each occurrence, which is necessary for two reasons. First, the motifs found should be *maximal*, which means that they should be as long as possible while maintaining intra-motif similarity. Second, neither very long, complex, but rare occurrences nor short, simple, frequent motifs are particularly useful in general.

The MDL criterion balances occurrence frequency with motif complexity and represents the number of bits needed to encode the motif plus the number of bits needed to mark each occurrence. The formula for the criterion is:

$$M \cdot \log_2(q_m) + n_m \cdot \log_2(q_s + 1)$$

where  $M$  is the total number of frames in all occurrences,  $q_m$  is the number of unique symbols in the motif,  $n_m$  is the number of occurrences, and  $q_s$  is the total number of unique symbols in the quantized data.



**Figure 4. Dendrogram showing a motif that should be split**

The motif with the largest score (corresponding to the largest reduction in total description length) is selected as the next best and is added to the list of seed motifs. If this score is below a user-specified threshold, however, seed motif discovery terminates.

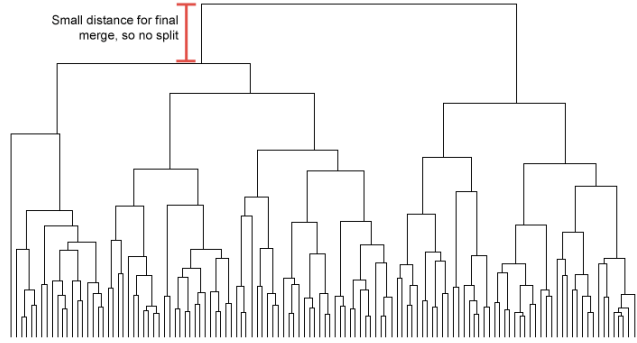
### 3.2 Seed Motif Refinement

Once the full set of seed motifs is identified, they are refined to account for errors introduced by the quantization and by the user-specified query length. Four different kinds of refinement are performed: splitting, merging, affix detection, and temporal extension.

The split refinement step accounts for motifs that appear similar in the quantized data, but are clearly different in the continuous domain. For each motif, the occurrences are analyzed using agglomerative clustering with the farthest-neighbor rule for comparing clusters. The resulting merge tree is then tested to see if it supports splitting the set into two new clusters. The test is performed by comparing the last merge (from two down to one cluster) to the second to last merge. If the corresponding distances are sufficiently large, then the two clusters are very different, and so a split is performed (see Figures 4 & 5).

After all seed motifs have been tested for splitting, the resulting set undergoes merge refinement. This step addresses a common problem with quantized data that occurs when data exists very close to a quantization boundary. In this case, two values that are similar in the continuous domain may be assigned different symbols. As a simple 1D example, consider the values 0.99 and 1.01 split by a quantization boundary at 1.0. Like splitting, merging proceeds via agglomerative clustering, but now the clustering is performed over the seed motifs rather than within their occurrences. If two seed motifs are determined to be very similar, then they are merged.

Next, affix detection looks for pairs of seed motifs in



**Figure 5. A dendrogram showing a motif that should not be split. Note how the final merge accounts for a relatively small amount of the total distance.**

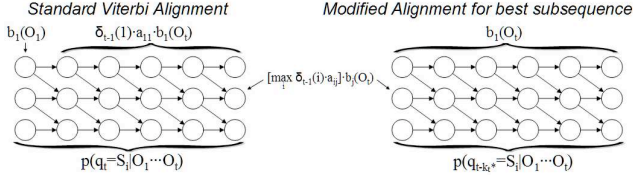
which the occurrences of one motif always closely follow the occurrences of another. This can happen if the user-specified query length is short relative to the actual motif. In such cases, a single motif may be identified multiple times, each time corresponding to a different portion of the full motif (prefix, middle, suffix, *etc.*). Affix detection seeks to detect this situation and fix it by combining the motifs.

The final refinement step deals with temporal extension. Extension is a common method for adapting the motif length to fit the data after part of the motif has been detected. A similar method is used in both PERUSE and the generalized discovery algorithm of Jensen *et al.* For each seed motif, the parameters of a left-right HMM are estimated from the motif occurrences and the set of variances in the observation distributions are extracted. The variance of the frame that precedes each occurrence is then compared to the mean of this set, and if it is comparable (or smaller), then the motif is extended to include the preceding frame. An identical procedure attempts to extend the motif forward in time, and in both cases, temporal extension continues until the variance of the next frame is too large to maintain intra-motif similarity.

### 3.3 Motif Modeling and Occurrence Detection

The final phase of the algorithm takes the refined seed motifs, builds a probabilistic model for each set, and uses the models to find all of the occurrences in the original time series. We use left-right HMMs as models due to their history of good performance for speech and gesture recognition and because a simple modification of the Viterbi alignment algorithm provides an efficient motif detection method.

The goal of this phase is to iteratively find the sub-



**Figure 6. Modified Viterbi alignment**

sequence with the maximum probability given each of the motif models. Unfortunately, there are  $O(T^2)$  subsequences, where  $T$  is the length of the longest time series, which makes scoring each subsequence impractical even for medium-length data sets. We can adapt the Viterbi alignment algorithm, however, to compute all of the needed probabilities with a single pass over the data rather than with a single pass over each subsequence.

Viterbi alignment builds a trellis that stores the maximum probability of being in each hidden state of the model for each frame, assuming that we start in the first frame and follow the optimal hidden state path:

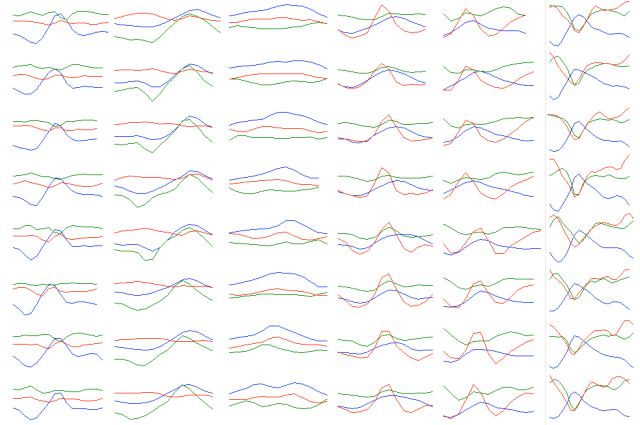
$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} p(q_t = S_i | O_1 \dots O_t)$$

where  $q_t = S_i$  is the probability of being in state  $S_i$  at time  $t$  and  $O_t$  is the observation at time  $t$  (see [16] for a review of the Viterbi algorithm and relevant notation). In such a trellis, the top row stores the probability of being in the first state at each time step, while the bottom row stores the probability of being in the last state. Each column represents one frame of data. Typically, only the first column is initialized and then a dynamic programming algorithm is used to fill in the rest of the trellis.

In our approach, however, the first column is initialized in the standard way, but we also initialize the top row as if the model started in the current frame:  $p(q_t = S_1 | O_t) = b_1(O_t)$  rather than  $p(q_t = S_1 | O_1 \dots O_t) = \delta_{t-1}(1) \cdot a_{11} \cdot b_1(O_t)$  (see Figure 6), where  $a_{ij}$  is the transition probability from state  $i$  to state  $j$ . The rest of the trellis is then computed using the standard method.

This modification changes the semantics of the nodes in the trellis. Now, each node stores the maximum probability of being in a hidden state without presuming a fixed starting time. Instead, the dynamic programming ensures that the probability is maximal given *any* previous starting time, and the exact time can be easily calculated using standard state sequence backtracking. This means that we can locate the best subsequence in linear time by simply scanning the bottom row for a maximum and then tracing backward through the trellis to find the optimal start time.

Note that this modification only works because we use a constrained HMM topology. Specifically, the modification requires that all start states be true sources so that even the



**Figure 7. Motifs discovered in the exercise data (only accelerometer readings shown).**

self-transition of a start state are zero (*i.e.*, if  $S_j$  is a start state, then  $\forall i : a_{ij} = 0$ ). There are no other restrictions imposed by the optimization, and this requirement is easily met by a left-right model without loss in representational power.

We can enumerate the motif occurrences in descending order of likelihood using the above procedure. All that remains is to determine when the next best occurrence is invalid. A simple approach would be to require the user to specify a minimum likelihood for valid matches. This is infeasible, however, because the absolute likelihood depends greatly on the motif (*e.g.*, variance in the observations and duration), thus requiring too much *a priori* knowledge. This approach would also likely require the user to specify a different threshold for each motif, an impossible task since the number of motifs is not generally known before the discovery process begins.

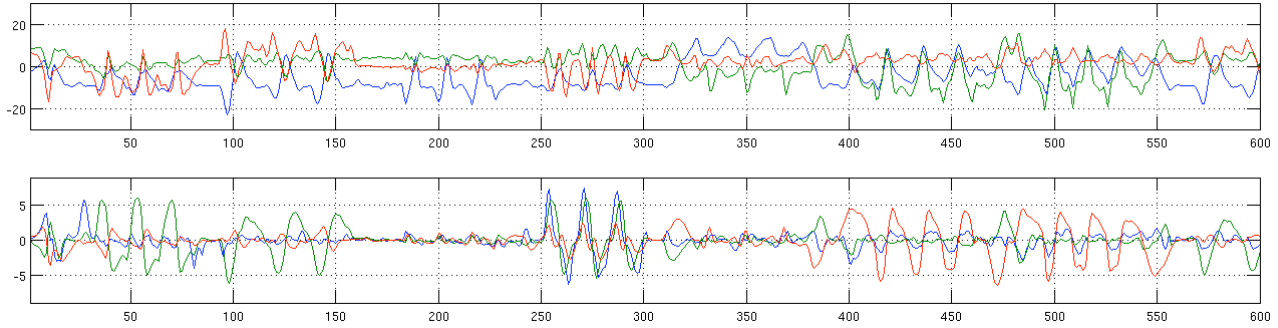
To circumvent these problems, we designed the criterion based on the likelihood of the occurrence relative to the distribution of likelihoods over the seed occurrences. Specifically, each occurrence is scored according to

$$\text{score}(\ell) = (1 + e^{\frac{-(\ell - \mu)}{\sigma}})^{-1}$$

where  $\ell$  is the likelihood of the occurrence, and  $\mu$  and  $\sigma$  are the mean and standard deviation of the likelihoods of the seed occurrences used to train the model. Since the scoring function is a sigmoid normalized by the standard deviation of the seed likelihoods, a single stopping threshold is applicable to all of the motifs.

## 4 Experimental Results

In order to empirically validate our approach to activity discovery, we captured data from a mock exercise routine



**Figure 8. Raw accelerometer (top) and gyroscope data (bottom) representing 600 frames (48 seconds) of data from one of the exercise sequences.**

composed of six different dumbbell exercises. An XSens MT9 inertial motion sensor was attached to the subject’s wrist by fitting it into a pouch sewn to the back of a thin glove. We sampled the MT9 at 100Hz and recorded three-axis accelerometer and gyroscope readings. In total, we captured approximately 27.5 minutes of data over 32 sequences. For the experiment, we down-sampled the data to 12.5Hz leading to 20,711 frames. The data contains six different exercises and 864 total repetitions (144 occurrences of each exercise). Each frame is composed of the raw accelerometer and gyroscope readings leading to a six-dimensional feature vector. The data was quantized using a mixture of ten 6D Gaussians. Figure 8 shows the raw data for one of the sequences.

Our algorithm correctly determines that there are six motifs and successfully locates 832 of the 864 of the occurrences (96.3% recall). It also locates 51 false occurrences (insertion errors) in addition to missing 32 real occurrences (deletion errors). There are no substitution errors, probably due to the high variation between the different exercises. This leads to an overall accuracy of 86.7% and precision of 88.4%. Figure 7 shows the accelerometer readings for several occurrences of each discovered motif.

Clearly, this is an evaluation of motif discovery in an artificial setting in that strong assumptions about the “correct” motifs are made. Such assumptions are unavoidable, however, if an objective and detailed measure of performance is desired. In real applications, the discovered motifs would be evaluated in other ways depending on the goals of the user. For instance, if the motifs represent an intermediate representation between the low-level sensor readings and some higher-level activity model, then the performance of a higher-level classifier based on the discovered motifs would be a better indication of their value.

Alternatively, the purpose of the discovery system could be to help the user form a better understanding of a domain

as perceived by a new sensor. In this case, subjective evaluation based on the semantic interpretability of the motifs is reasonable. This determination could be made by visualizing the occurrences of each motif in the context of the original sensor or by inspecting the readings of a different sensor, like a time-synchronized video or audio recording, that is more natural for the user to interpret.

## 5 Future Work

In light of the encouraging results on the exercise data set, we are currently validating our approach by evaluating it on additional data sets taken from other domains and with different sensors. One such domain is American Sign Language for which we have a data set consisting of 500 sentences composed from a 40 word vocabulary captured by a head-mounted video camera. Here, the time series are made up of 16-dimensional feature vectors describing the blobs corresponding to the signer’s left and right hand. Other data sets include English speech, Kung Fu forms captured by on-body accelerometers, and another ASL data set captured with a glove that measures finger posture.

Another major concern with our current system involves the various user-specified parameters including the seed motif stopping threshold, merge, split, and temporal extension criteria, and initial query length. Although we have made a great effort to ensure that the parameters are as stable and as independent of the particular data as possible (*e.g.*, with regard to sampling rate, model complexity, occurrence density, *etc.*), we still need to demonstrate this empirically and evaluate the stability of the parameters.

We are also interested in learning high-level structure over the discovered actions and in devising a way of incorporating such knowledge to improve the motif discovery results. This could serve as a way to inject user knowledge to help guide the system or to improve the existing unsuper-

vised system by inducing activity structure and then incorporating high- to low-level feedback.

Finally, we would like to develop an interactive discovery system based on our approach. Ideally, the system would detect potential errors or ambiguities and present the user with a graphical display in which they could easily provide guiding information. For example, the system could present sample images or video corresponding to several borderline seed motifs and the user could click on those that do not belong. Alternatively, it may be easy for the user to manually specify which seed motifs need to be split or merged once presented with example occurrences, thus avoiding a lengthy calculation and error-prone threshold. Although much work is needed to detect appropriate interaction points, select useful examples, and incorporate the user's feedback, we are encouraged by the recent success of other interactive machine learning research [8] and think that similar benefits are possible within activity and general motif discovery.

## 6 Conclusions

We have presented an approach for activity discovery that allows for variable-length motifs, supports dynamic time warping, and which is efficient even when the motifs are sparsely distributed. Our approach has been evaluated on a real data set containing over 27 minutes of sensor readings from an on-body accelerometer and gyroscope. It is able to correctly identify the number of motifs and to locate 96.3% of the occurrences with an overall accuracy of 86.7%. Based on these results, we are currently validating our approach with data from other domains and with different sensor modalities. Finally, we are interested in applying our technique to data sets in which the "correct" motifs are unknown and then using input from domain experts to interpret and verify the discovered patterns.

## References

- [1] T. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Second International Conference on Intelligent Systems for Molecular Biology*, pages 28–36. AAAI Press, 1994.
- [2] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive*, volume LNCS 3001, pages 1–17. Berlin Heidelberg: Springer-Verlag, 2004.
- [3] H. Brashear, T. Starner, P. Lukowicz, and H. Junker. Using multiple sensors for mobile sign language recognition. In *IEEE International Symposium on Wearable Computers*, October 2003.
- [4] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *Conf. on Knowledge Discovery in Data*, pages 493–498, 2003.
- [5] B. Clarkson and A. Pentland. Unsupervised clustering of ambulatory audio and video. In *ICASSP*, 1999.
- [6] D. Colbry, B. Peintner, and M. Pollack. Execution monitoring with quantitative temporal bayesian networks. In *6th International Conference on AI Planning and Scheduling*, April 2002.
- [7] R. Duda, P. Hart, and D. Stork. *Pattern Classification*, 2nd ed. John Wiley & Sons, Inc., 2001.
- [8] J. A. Fails and D. R. Olsen, Jr. Interactive machine learning. In *8th International Conference on Intelligent User Interfaces*, pages 39–45. ACM, 2003.
- [9] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
- [10] T. Huynh and B. Schiele. Analyzing features for activity recognition. In *Smart Objects and Ambient Intelligence*, October 2005.
- [11] K. Jensen, M. P. Styczynski, I. Rigoutsos, and G. Stephanopoulos. A generic motif discovery algorithm for sequential data. *Bioinformatics*, 22(1):21–28, 2006.
- [12] E. Keogh. Personal communication, 2006.
- [13] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hananford. A hybrid discriminative/generative approach for modeling human activities. In *Nineteenth International Joint Conference on Artificial Intelligence*, pages 766–772, July 30 - August 5 2005.
- [14] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, June 2003.
- [15] T. Oates. PERUSE: An unsupervised algorithm for finding recurring patterns in time series. In *Int. Conf. on Data Mining*, pages 330–337, 2002.
- [16] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Readings in speech recognition*, pages 267–296, 1990.
- [17] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *PAMI*, 20(12):1371–1375, December 1998.
- [18] Y. Tanaka and K. Uehara. Discover motifs in multi-dimensional time-series using the principal component analysis and the MDL principle. In *International Conference on Machine Learning and Data Mining*, pages 252–265, 2003.
- [19] Q. Tran, G. Calcaterra, and E. Mynatt. Cook's collage: Memory aid display for cooking. In *HOIT*, page to appear, April 2005.
- [20] E. Ukkonen. Constructing suffix-trees on-line in linear time. *Algorithms*, 1(92):484–492, 1992.
- [21] C. Vogler and D. Metaxas. ASL recognition based on a coupling between hmms and 3d motion analysis. In *ICCV98*, pages 363–369, 1998.
- [22] T. Westeyn, K. Vadas, X. Bian, T. Starner, and G. D. Abowd. Recognizing mimicked autistic self-stimulatory behaviors using hmms. In *IEEE International Symposium on Wearable Computers*, pages 164–169, October 2005.
- [23] L. Xie, S.-F. Change, A. Divakaran, and H. Sun. *Unsupervised Mining of Statistical Temporal Structures in Video*, chapter 10. Kluwer Academic Publishers, 2003.