Detecting Subdimensional Motifs: An Efficient Algorithm for Generalized Multivariate Pattern Discovery

David Minnen, Charles Isbell, Irfan Essa, and Thad Starner
Georgia Institute of Technology
College of Computing, School of Interactive Computing
Atlanta, GA 30332-0760 USA
dminn,isbell,irfan,thad@cc.gatech.edu

Abstract

Discovering recurring patterns in time series data is a fundamental problem for temporal data mining. This paper addresses the problem of locating subdimensional motifs in real-valued, multivariate time series, which requires the simultaneous discovery of sets of recurring patterns along with the corresponding relevant dimensions. While many approaches to motif discovery have been developed, most are restricted to categorical data, univariate time series, or multivariate data in which the temporal patterns span all of the dimensions. In this paper, we present an expected linear-time algorithm that addresses a generalization of multivariate pattern discovery in which each motif may span only a subset of the dimensions. To validate our algorithm, we discuss its theoretical properties and empirically evaluate it using several data sets including synthetic data and motion capture data collected by an on-body inertial sensor.

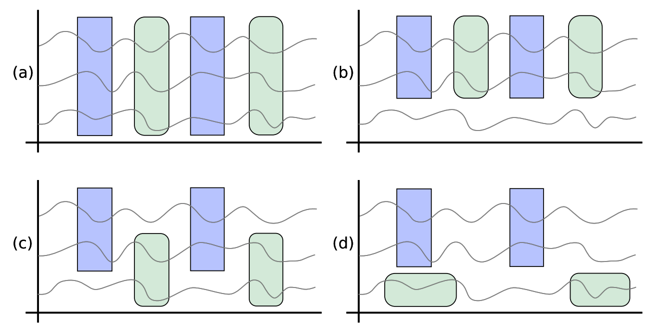

Figure 1. Extending the idea of a univariate motif to multivariate data can take several form: (a) every motif spans all of the dimensions, (b) each motif spans the same subset of the dimensions, (c) each motif spans a (potentially unique) subset of the dimension, but motifs never temporally overlap, and (d) motifs may temporally overlap if they span different dimensions.

1. Introduction

A central problem in temporal data mining is the unsupervised discovery of recurring patterns in time series data. This paper focuses on the case of detecting such unknown patterns, often called *motifs*, in multivariate, real-valued data. Many methods have been developed for motif discovery in categorical data and univariate, real-valued time series [6, 1, 4, 3, 12], but relatively little work has looked at multivariate data sets. Multidimensional time series are very common, however, and arise directly from multi-sensor systems and indirectly due to descriptive features extracted from univariate signals.

The existing research that does address the problem of multivariate motif discovery typically focuses on locating patterns that span all of the dimensions in the data [10, 11, 2, 7, 9, 8]. While this generalization from the univariate case represents important progress and may fit the properties of a particular data set quite well, we are interested in addressing a broader form of multivariate pattern detection, which we call *subdimensional* motif discovery. Figure 1 depicts four categories of multidimensional motifs in order of increasing generality. The case typically addressed is shown in Figure 1a where the motifs span all three of the dimensions. Alternate subdimensional formulations include Figure 1b where some dimensions are irrelevant to all of the motifs, Figure 1c where the relevancy of each dimension is determined independently for each motif, and finally Figure 1d which allows the recurring patterns to temporally overlap in different dimensions. In this paper, we present an algorithm that can efficiently and accurately locate previously unknown patterns in multivariate time series up to the generality represented by Figure 1c. Note that our algorithm also naturally handles the more restrictive problems depicted in Figure 1a, which we call "all-dimensional" motif discovery, and Figure 1b, as well as the univariate case.

Subdimensional motifs arise in many circumstances including distributed sensor systems, multimedia mining, on-body sensor analysis, and motion capture data. The key benefit of subdimensional motif discovery is that such methods can find patterns that would remain hidden to typical multivariate algorithms. The ability to automatically detect the relevance of each dimension on a per-motif basis allows great flexibility and provides data mining practitioners with the freedom to include additional features, indicators, or sensors without requiring them to be a part of the pattern. Subdimensional discovery also provides robustness to noisy or otherwise uninformative sensor channels.

2. Discovering Subdimensional Motifs

Our approach to subdimensional motif discovery extends the framework developed by Chiu *et al.* [3], which has also been adapted by several other researchers to address variations on the basic motif discovery problem [9, 11, 12]. In this section, we provide a brief review of the existing algorithmic framework and present our enhancements that allow efficient subdimensional motif discovery.

Our algorithm searches for pairs of similar, fixed-length subsequences and uses these *motif seeds* to detect other occurrences of the same motif. The search is made efficient by first discretizing each subsequence and then using random projection to find similar strings in linear time. Once a potential motif seed is found, the algorithm determines the relevance of each dimension for that motif and then (optionally) estimates the motif's neighborhood size and searches for additional occurrences. See Algorithm 1 for a more detailed overview.

Algorithm 1 Subdimensional Motif Discovery

Input: Time series data (S), subsequence length (w), word length (m), maximum number of random projection iterations (max_{rp}), threshold for dimension relevance ($thresh_{rel}$), and a distance measure ($D(\cdot,\cdot)$)

Output: Set of discovered motifs including occurrence locations and relevant dimensions

1. Collect all subsequences, s_i, of length w from the time series S: $s_i = \langle S_i, .., S_{i+w-1}\rangle : 1 \leq i \leq |S| - w + 1$
2. Compute $\hat{p}(D) \approx p(D(s_{i,d}, s_{j,d}))$, an estimate of the distribution over the distance between all non-trivial matches for each dimension, d, by random sampling
3. Search for values of α (alphabet size) and c (projection dimensionality) that lead to a sparse collision matrix
4. Compute the SAX word of length m and alphabet size α for each dimension of each subsequence
5. Build the collision matrix using random projection over the SAX words; number of iterations $= min(\binom{m}{c}, max_{rp})$
6. Enumerate the motifs based on the collision matrix
   (a) Find the best collision matrix entry $(\dot{x}^1, \dot{x}^2)$
      i. Find the largest entry in the collision matrix and extract the set of all collisions with this value: $X = \{(x_1^1, x_1^2), (x_2^1, x_2^2), ..., (x_{|X|}^1, x_{|X|}^2)\}$
      ii. Compute the distance between the subsequences x_j^1 and x_j^2 in each collision, $1 \leq j \leq |X|$, and dimension, d: $dist_{j,d} = D(s_{x_j^1,d}, s_{x_j^2,d})$
      iii. Determine which dimensions are relevant: $rel(d) = \mathrm{I}(\int_{-\infty}^{dist_{j,d}} \hat{p}(D)) < thresh_{rel})$
      iv. Select the collision with smallest average distance per relevant dimension: $(x_j^1, x_j^2) : j = arg\min_j(\frac{\sum_d dist_{j,d} \cdot rel(d)}{\sum_d rel(d)})$
   (b) Estimate the neighborhood radius, R, using only the relevant dimensions
   (c) Locate all other occurrences of this motif: $min(D(s_{\dot{x}^1}, s_i), D(s_{\dot{x}^2}, s_i)) \leq R$
   (d) Remove subsequences that would constitute trivial matches with the occurrences of this motif

2.1. Local Discretization

We adopt the method of symbolic aggregate approximation (SAX) as a means for very efficient local discretization of time series subsequences [5]. SAX is a local quantization method that first computes a piecewise aggregate approximation (PAA) of the normalized window data and then replaces each PAA segment with a symbol. The SAX algorithm assigns a symbol to each segment by consulting a table of precomputed breakpoints that divide the data range into equiprobable regions assuming an underlying Gaussian distribution.

2.2. Random Projection

Random projection provides a mechanism for locating approximately equal subsequences in linear time. After extracting the subsequences and converting them to SAX words, the algorithm proceeds through several iterations of random projection. Each iteration selects a subset of the word positions and projects each word by removing the remaining positions. This is essentially axis-aligned projection for fixed-length strings (see Figure 2a and 2b).

In order to detect similar words, a *collision matrix* is maintained. If there are T subsequences, then the collision matrix has size T x T and stores the number of iter-

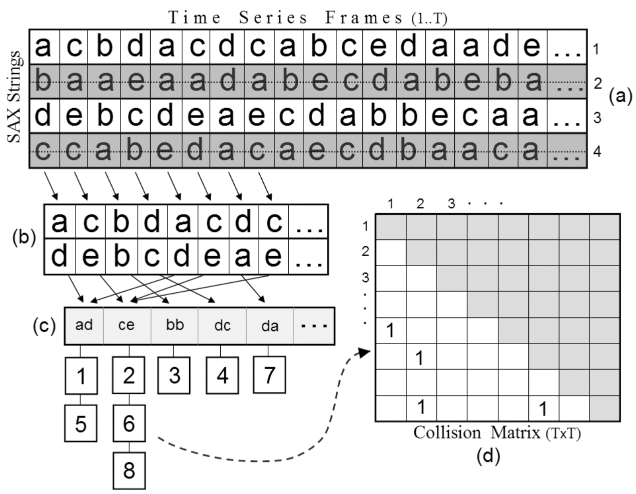

Figure 2. (a,b) For each iteration of random projection, a subset of string positions are selected (here, positions one and three). (c) The selected symbols are hashed, and (d) equivalent projections are tallied in a collision matrix.

ations in which each pair of subsequences were equivalent after discretization and random projection. The matrix is updated after each iteration by hashing the projected words and then incrementing the matrix entry for each equivalent pair (see Figure 2c and 2d). Finally, after the last iteration of random projection, the entries in the collision matrix represent the relative degree of similarity between subsequences. These values provide a means for focusing computational resources by only analyzing those entries that are large relative to the expected hit rate for random strings [3].

The total time complexity of the random projection algorithm is linear in the number of strings (T), the number of iterations (I), the length of each projected word (c), and the number of collisions ($C = \sum_{i=1}^{I} C_i$). The complexity is dominated by the collisions since C_i grows quadratically with the number of equivalent projected words, which can rise as high as T in the worst case. Specifically, $C_i = \sum_{h \in H} \binom{N_h}{2} = \sum_{h \in H} \frac{1}{2} N_h \cdot (N_h - 1)$, where H is the set of all projected strings and N_h equals the number of strings that project to $h \in H$. In the case where a large proportion of the subsequences have the same projection, h^*, we have $N_{h^*} = O(T)$ and thus $C_i = O(T^2)$, which is infeasible in terms of both time and space for large data sets.

In order to avoid quadratic complexity, our algorithm searches for parameters that ensure a sufficiently wide projection distribution. Using a SAX alphabet of size α and projection dimensionality of c, there will be α^c possible projected strings, and, given that the SAX algorithm seeks equiprobable symbols, the distribution should be close to uniform except where actual recurring patterns create a bias.

At run time, the algorithm dynamically adjusts α and c to control the number of hits. Starting with $\alpha = 3$ and c set to the length of the original word (*i.e.,* no projection), the value of α is increased if the collision matrix becomes too dense, while c is reduced if too few matches are found. Furthermore, the collision matrix uses a sparse matrix data structure to ensure that the storage requirements scale with the number of collisions rather than with the full size of the matrix.

When dealing with univariate data, applying the random projection algorithm is straightforward. For multivariate data, however, each dimension leads to its own SAX word, and so a method for combined projection is required. To address the all-dimensional motif discovery problem (Figure 1a), researchers have simply concatenated the projections of the words from each dimension and then hashed the resulting string [9]. To discover subdimensional motifs, our algorithm instead increments the collision matrix for each dimension that matches. This change can be understood as a switch from a logical AND policy in the all-dimensional case (*i.e., all* dimensions must match to qualify as a collision) to a logical OR policy (*i.e.,* a collision occurs if *any* of the dimensions match. The algorithm increments the relevant entry once for each matching dimension to account for the additional support that multiple similar dimensions provides.

2.3. Locating Relevant Dimensions

The random projection algorithm, as described in the previous section, does not provide information about which dimensions are relevant for a particular motif. Although we could modify the algorithm to maintain separate collision matrices for each dimension, initial experiments showed that this approach led to inaccurate relevance estimation. Instead, we use the collision matrix to help locate motif seeds and then determine the relevant dimensions by analyzing the original, real-valued data.

When motifs are defined by a fixed, user-specified neighborhood radius, dimension relevance is easily determined by locating those dimensions which do not cause the distance between the seeds to exceed the given radius. Specifically, we can sort the dimensions by increasing distance and then incrementally add dimensions until the seed distance grows too large.

In the case when the neighborhood radius must be estimated, a more involved approach is required. Here we estimate the distribution over distances between random subsequences for each dimension by sampling from the data set. Then, given the distribution and a seed to analyze, we can evaluate the probability that a value smaller than the seed distance would arise randomly by calculating the corresponding value of the cumulative distribution function. If

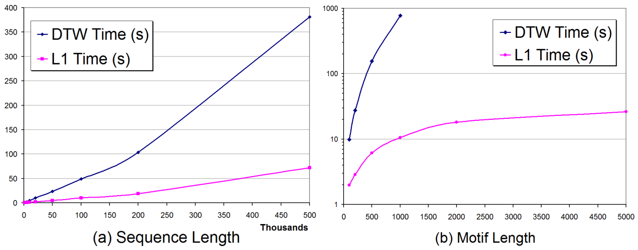

Figure 3. Graphs showing how the subdimensional discovery algorithm scales with (a) increasing time series length and (b) increasing motif length

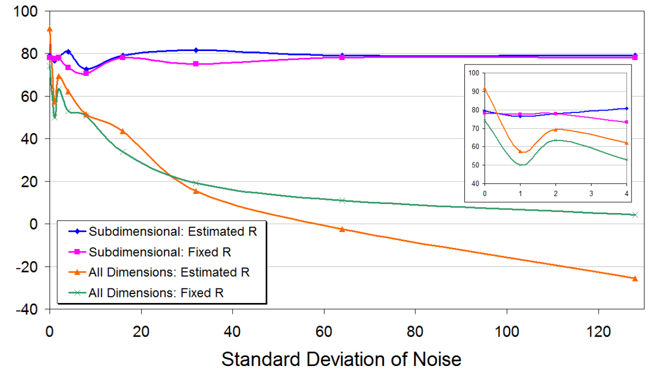

Figure 4. Event-based accuracy for both the subdimensional and all-dimensional discovery algorithms.

this value is large, then we deem the dimension irrelevant because it is likely to arise at random, while if it small, it likely indicates an interesting similarity.

In the experiments presented in this paper, we model the distances with a Gaussian distribution and require the seed distance to be smaller than 80% of the expected distances (*i.e.,* $cdf(dist_d) \leq 0.2$). It is straightforward, however, to use a more expressive model, such as a nonparametric kernel density estimate or gamma distribution, for more accurate relevance decisions.

3. Experimental Evaluation

We evaluated our algorithm by running experiments using planted motifs as well as non-synthetic data captured by on-body inertial sensors. Our experiments demonstrate the efficacy of the algorithm as well as its scaling properties as the length of the time series data and the number of dimensions increases. We also investigate the effect of different distance metrics and provide a comparison with other multivariate discovery algorithms.

3.1. Planted Motifs

As an initial verification that our subdimensional motif discovery algorithm is able to locate motifs amongst irrelevant sensor channels, we performed a planted motif experiment using synthetic data. For this problem, a random time series is generated and then one or more artificial motifs are inserted. The discovery system, which has no knowledge of the pattern, must then locate the planted motifs.

For the case of a single planted motif, Figure 3 shows how our algorithm scales as the length of the time series (T) increases (Figure 3a) and as the length of the motif (M) increases (Figure 3b). The algorithm is able to accurately locate the motif in all cases, and, importantly, it correctly identifies the irrelevant dimension. From Figure 3a, we see that the time required to locate the planted motif scales linearly with the length of the time series. As the motif length increases, however, the behavior changes. When the L_1 distance metric is used, the algorithm still scales linearly, but when the dynamic time warping (DTW) distance measure is used, however, the time scales quadratically. This is not surprising since DTW is quadratic in M even when warping constraints are used (in all of the experiments, we used a 10% Sakoe-Chiba band). In typical cases of motif discovery, however, $M \ll T$, and so linear dependence on T still dominates the overall run time.

3.2. Distracting Noise Channels

In this section, we investigate the ability of our subdimensional motif discovery algorithm to detect multivariate motifs in real sensor data despite the presence of distracting noise dimensions. We evaluated robustness in two cases by: (1) adding increasingly large amounts of noise to a single distracting noise dimension and (2) adding additional irrelevant dimensions each with a moderate amount of noise.

The non-synthetic data set was captured during an exercise regime made up of six different dumbbell exercises. A three-axis accelerometer and gyroscope mounted on the subject's wrist were used to record each exercise. The data set consists of 20,711 frames over 32 sequences and contains roughly 144 occurrences of each exercise. This data set was previously used to evaluate an all-dimensional motif discovery algorithms [9] and so we use that method as a basis for comparison.

Figure 4 shows the results of the first experiment in which a single dimension of noise was added to the six dimensional exercise data. The graph shows the accuracy of the discovered motifs relative to the known exercises. The evaluation framework matched discovered motifs to known exercises and calculated the score by determining those motif occurrences that correctly overlapped the real instances (C), along with all insertion (I), deletion (D),

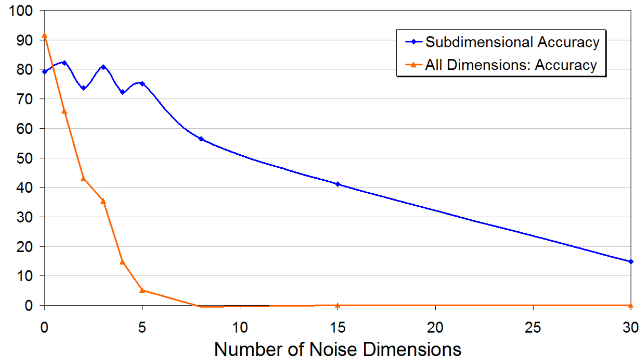

Figure 5. Event-based accuracy for both the subdimensional and all-dimensional discovery algorithms using automatic neighborhood estimation.

and substitution errors (S). Accuracy was then calculated as $acc = \frac{C-I-S}{N}$ where $N = C + D + S$, the total number of real occurrences.

From Figure 4 we see that with no noise, both subdimensional algorithms achieve roughly 80% accuracy while the fixed radius all-dimensional algorithm performs slightly worse (74.2%) and the automatic radius estimation version performs somewhat better at 91.7%. As the scale of the noise in the extra dimension increases, however, the accuracy of both all-dimensional systems quickly falls, while accuracy of the subdimensional algorithms remains relatively unchanged. Note that this behavior is expected as the all-dimensional algorithms try to locate motifs that include the (overwhelming) noise dimension, while the subdimensional algorithms simply detect its irrelevance and only search for motifs that span the six remaining dimensions that contain valid sensor data.

In the second experiment, instead of increasing the scale of the noise, we increased the number of dimensions with moderate noise (equivalent to a standard deviation of four in Figure 4, which is close to the average signal level of the real data). Figure 6 shows three discovered occurrences of the “twist curl” exercise along with the three noise dimensions that the algorithm identified as irrelevant. The effect that additional noise dimension have on accuracy is shown in Figure 5. From the graph, we see that the performance of both the all-dimensional and subdimensional algorithms decrease with extra noise dimensions but the all-dimensional algorithm decreases much more rapidly. Ideally, the subdimensional algorithm would detect all of the additional noise dimensions as irrelevant and performance would stay level as it did in Figure 4. We believe that performance drops because the algorithm discovers incidental patterns in the random dimensions which are counted as errors by the evaluation framework. This phenomenon makes sense because

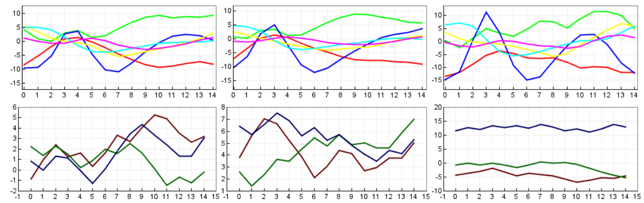

Figure 6. Three discovered occurrences of the twist curl exercise. The top row shows the (correct) relevant dimensions corresponding to the real sensor data while the bottom row shows the irrelevant (noise) dimensions.

the probability of an unintentional pattern arising increases as the number of noise dimensions increases.

4. Related Work

Many data mining researchers have developed methods for motif discovery in real-valued, univariate data. Lin *et al.* [6] use a hashing algorithm (later introduced as symbolic aggregate approximation [5]) and an efficient lower-bound calculation to search for motifs. Chiu *et al.* [3] use the same local discretization procedure and random projection based on Buhler and Tompa’s research [1] to find candidate motifs in noisy data. Yankov *et al.* [12] extend this approach by using a uniform scaling distance metric rather than Euclidean distance, which allows the algorithm to detect patterns with different lengths and different temporal scaling rates. Denton’s approach [4] avoids discretization and frames subsequence clustering in terms of kernel density estimation. Her method relies on the assumption of a random-walk noise model to separate motifs from background clutter.

Other researchers have developed discovery algorithms that detect multivariate (all-dimensional) patterns. Minnen *et al.* [9] extend Chiu’s approach by supporting multivariate time series and automatically estimating the neighborhood size of each motif. In earlier work, the same researchers used a global discretization method based on vector quantization and then analyzed the resulting string using a suffix tree [7]. Tanaka *et al.* [11] also extend Chiu’s work, but rather than analyzing the multivariate data directly, they use principal component analysis to project the signal down to one dimension and then apply a univariate discovery algorithm.

While the above methods discretize the multivariate time series to allow efficient motif discovery, other research has investigated methods that do not require such discretization. Oates developed the PERUSE algorithm to find recurring patterns in multivariate sensor data collected by robots [10]. PERUSE is one of the few algorithms that can handle non-uniformly sampled data and variable-length motifs, but it suffers from some computational drawbacks and stability issues when estimating motif models. Catalano *et al*. introduced a very efficient algorithm for locating variable length patterns in multivariate data using random sampling, which allows it to run in linear time and constant memory [2]. Minnen *et al*. framed motif discovery in terms of density estimation and greedy mixture learning [8]. They estimated density via k-nearest neighbor search using a dual-tree algorithm to achieve an expected linear run time and then used hidden Markov models to locate motif occurrences.

5. Future Work

We are currently investigating several research directions that can improve our subdimensional motif discovery algorithm. For instance, we are interested in using discovered motifs as the primitives within a broader temporal knowledge discovery system. With such a system, we might discover that certain motifs provide good predictors for other motifs or for trends in the data. Similarly, learned temporal relationships may support the detection of poorly formed motif occurrences that are predicted by the higher-level model, or they may help identify anomalies where a predicted motif is missing.

Generalizing our algorithm to allow for the discovery of variable-length motifs is another important enhancement. We are exploring methods that will allow small temporal variations between motifs and motif occurrences. For instance, methods for temporally extending discovered motifs or combining overlapping motifs may be applicable [10, 7]. Similarly, we are working to develop a robust method for estimating the time scale of motifs and allowing for the discovery of motifs at multiple time scales.

Finally, we are also exploring the design of an interactive discovery system. While we consider the automated operation of our algorithm to be a strength, an interactive system may be better suited to allow domain researchers to guide the discovery process and resolve ambiguities encountered by the discovery system.

6. Conclusion

We have described several generalizations of the multivariate motif discovery problem and have presented a subdimensional discovery algorithm that can efficiently detect recurring patterns that may exist in only a subset of the dimensions of a data set. The key insight that allows linear-time discovery of subdimensional motifs is that we can apply random projection independently to each dimension only if the size of the collision matrix is monitored and the relevant parameters (α and c from Section 2.2) are dynamically updated to limit the density of the matrix. The accuracy and efficiency of our algorithm was empirically demonstrated with planted motifs as well as non-synthetic sensor data. We are currently working with other researchers to apply our method to additional domains including multi-sensor EEG recordings, motion capture data for automatic activity analysis and gesture discovery, entomological data, econometric time series, distributed sensor systems deployed in homes and offices, and speech analysis.

References

[1] J. Buhler and M. Tompa. Finding motifs using random projections. In *International Conference on Computational Biology*, pages 69–76, 2001.

[2] J. Catalano, T. Armstrong, and T. Oates. Discovering patterns in real-valued time series. In *Proc. of the Tenth European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Septembar 2006.

[3] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic disovery of time series motifs. In *Conf. on Knowledge Discovery in Data*, pages 493–498, 2003.

[4] A. Denton. Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, November 2005.

[5] 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.

[6] J. Lin, E. Keogh, S. Lonardi, and P. Patel. Finding motifs in time series. In *Proc. of the Second Workshop on Temporal Data Mining*, Edmonton, Alberta, Canada, July 2002.

[7] D. Minnen, T. Starner, I. Essa, and C. Isbell. Discovering characteristic actions from on-body sensor data. In *Int. Symp. on Wearable Computers*, pages 11–18, Oct. 2006.

[8] D. Minnen, T. Starner, I. Essa, and C. Isbell. Discovering multivariate motifs using subsequence density estimation. In *AAAI Conf. on Artificial Intelligence*, 2007.

[9] D. Minnen, T. Starner, I. Essa, and C. Isbell. Improving activity discovery with automatic neighborhood estimation. In *Int. Joint Conf. on Artificial Intelligence*, 2007.

[10] T. Oates. PERUSE: An unsupervised algorithm for finding recurring patterns in time series. In *Int. Conf. on Data Mining*, pages 330–337, 2002.

[11] Y. Tanaka, K. Iwamoto, and K. Uehara. Discovery of time-series motif from multi-dimensional data based on mdl principle. *Machine Learning*, 58(2-3):269–300, 2005.

[12] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan. Detecting motifs under uniform scaling. In *Int. Conf. on Knowledge Discovery and Data Mining*, August 2007.