Scale Invariant Multi-length Motif Discovery

Yasser Mohammad[1,2] and Toyoaki Nishida[2]

[1] Assiut University, Egypt
[2] Kyoto University, Japan
yasserm@aun.edu.eg, nishida@i.kyoto-u.ac.jp

Abstract. Discovering approximately recurrent motifs (ARMs) in timeseries is an active area of research in data mining. Exact motif discovery was later defined as the problem of efficiently finding the most similar pairs of timeseries subsequences and can be used as a basis for discovering ARMs. The most efficient algorithm for solving this problem is the MK algorithm which was designed to find a single pair of timeseries subsequences with maximum similarity at a known length. Available exact solutions to the problem of finding top K similar subsequence pairs at multiple lengths (which can be the basis of ARM discovery) are not scale invariant. This paper proposes a new algorithm for solving this problem efficiently using scale invariant distance functions and applies it to both real and synthetic dataset.

1 Introduction

Discovering approximately recurrent patterns in timeseries is a basic problem in data mining and provides the basis for solving many real world problems (e.g. gesture discovery [10], any-time nearest neighbor algorithms [12], fluid imitation [8], etc). Consider a robot watching free hand motion of a human operator while operating another actor using gestures [10]. The ability to automatically discover recurring motion patterns allows the robot to learn important gestures related to this domain. Consider an infant listening to the speech around it. The ability to discover recurring speech patterns (words) can be of great value in learning the vocabulary of language. In both in these cases, and in uncountable others, the patterns do not recur exactly in the perceptual space of the learner. These cases motivate our search for an unsupervised algorithm that can discover these kinds of approximately recurring motifs (ARMs) in general time-series. Several algorithms have been proposed for solving this problem [7] [5] [1] [9].

A promising approach to solve the ARM problem is to use an algorithms that finds *exactly* the K timeseries subsequence pairs (called 2-motifs) of maximal similarity then use them as the basis for discovering recurrent patterns which by definition must have maximal similarity between its pairs. The naive algorithm for solving this problem exactly for a time series of length n and motifs of lengths between l_1 and $l_2 = l_1 + l$ has a time complexity of $O\left(n^2 K l\right)$. This quadruple complexity makes it impractical to apply this algorithm except for short timeseries, short motifs, and short motif length ranges.

A. Moonis et al. (Eds.): IEA/AIE 2014, Part II, LNAI 8482, pp. 417–426, 2014.

The simpler problem of exact discovery of the top 2-motif of a given length in a timeseries was defined by Mueen and Keogh [12] in 2009 and an efficient exact solution with amortized linear complexity was proposed (called the MK algorithm). This algorithm reduced the amortized time complexity from quadratic to linear which makes it practical to apply it to moderately long timeseries. The MK algorithm uses the Euclidean distance between zscore normalized subsequences as a dissimilarity measure. The main advantage of this distance function is that it is offset and scale invariant. It was also shown that it can provide a comparable performance to Dynamic Time Wrapping [4].

Mohammand and Nishida [9] proposed MK+ which is an efficient extension of MK to discover top K 2-motifs of a given length using the same distance function and showed that it outperforms iterative application of the MK algorithm. MK+ was further extended in [6] (MK++) to discover top K 2-motifs of a range of lengths but assuming that the distance between two subsequences of the timeseries cannot decreased with increased length. This assumption is true of the Euclidean distance and Euclidean distance between mean-shifted subsequences but is not true for zscore normalized subsequences. This means that MK++ cannot be used to discover scale-invariant 2-motifs which means that it cannot be a basis for scale invariant ARM discovery.

Recently, Mueen proposed MOEN [11] for solving the scale invariant version of the problem tackled in this paper. The main idea of MOEN is to calculate a lower bound on the distance between any two subsequences at length l given this distance at length $l-1$. Using this lower bound, it is possible to efficiently discover 2-motifs at different lengths. MOEN works with zscore normalized subsequences but the proposed algorithm can be applied to a more general class of distance functions.

This paper proposes two solutions to this problem: The first approach is to use a different normalization technique by dividing the mean shifted subsequence by its range (difference between maximum and minimum values) rather than standard deviation and using MK++ with minimal modification. We show that this renders the algorithm approximate but in most cases leads to exactly the same results as the exact algorithm. The second approach called sMD (for scale-invariant Motif Discovery) is to drive an incremental method to calculate any normalized distance function and then to use it to find motifs at all lengths in parallel leading to an exact 2-motif discovery algorithm.

The rest of the paper is organized as follows: Section 2 gives the problem statement. Section 3 describes MK and MK++ that form the basis of the proposed algorithm. Section 4 details the proposed incremental distance calculation method and section 5 gives the details of the proposed algorithm which is evaluated in section 6. The paper is then concluded.

2 Problem Statement

A time series $x(t)$ is an ordered set of T real values. A subsequence $x_{i,j} = [x(i) : x(j)]$ is a continguous part of a time series x. In most cases, the distance between overlapping subsequences is considered to be infinitely high to avoid assuming that two sequences are matching just because they are shifted versions of each other (these are called trivial motifs [3]). In this paper we utilize the following definitions:

Definition 1. *2-Motif:* Given a timeseries x of length T, a motif length L, a maximum internal overlap $0 \geq wMO \geq 1$, maximum between-motifs overlap $0 \geq bWO \geq 1$, and a distance function $D(.,.)$; the top 2-motif is a pair of subsequences s_1, s_2 of length L with minimum distance compared with any other pair of subsequences in the time-series that have an overlap less or equal to wMO, the 2^{nd} 2-motif is the pair of subsequences overlapping the top 2-motif no more than bMO that have the minimum distance compared with any other pair satisfying this overlapping condition. The K^{th} 2-motif is the pair of subsequences overlapping none of the top to the $K - 1^{th}$ 2-motif more than bMO that have the minimum distance compared with any other pair satisfying this overlapping condition.

Using this definition, the problem statement of this paper can be stated as: *Given a time series x, minimum and maximum motif lengths (L_{min} and L_{max}), a maximum allowed within-motif overlap (wMO), and a maximum allowed between-motifs overlap (bMO), find the top K 2-motifs with smallest motif distance among all possible pairs of subsequences.*

3 MK and MK++ Algorithms

The MK algorithm finds the top 2-occurrences motif in a time series. The main idea behind MK algorithm [12] is to use the triangular inequality to prune large distances without the need for calculating them. For metrics $D(.,.)$ (including the Euclidean distance), the triangular inequality can be stated as:

$$D(A, B) - D(C, B) \leq D(A, C) \tag{1}$$

Assume that we have an upper limit on the distance between the two occurrences of the motif we are after (th) and we have the distance between two subsequences A and C and some reference point B. If subtracting the two distances leads a value greater than th, we know that A and C cannot be the motif we are after without ever calculating their distance. By careful selection of the order of distance calculations, MK algorithm can prune away most of the distance calculations required by a brute-force quadratic motif discovery algorithm. The availability of the upper limit on motif distance (th), is also used to stop the calculation of any Euclidean distance once it exceeds this limit. Combining these two factors, 60 folds speedup was reported in [12] compared with the brute-force approach.

The inputs to the algorithm are the time series x, its total length T, motif length L, and the number of reference points N_r .The algorithm starts by selecting a random set of N_r reference points. The algorithm works in two phases: The first phase (called hereafter referencing phase) is used to calculate both the upper limit on best motif distance and a lower limit on distances of all possible pairs. During this phase, distances between the subsequences of length L starting at the N_r reference points and all other $T - L + 1$ points in the time series are calculated resulting in a distance matrix of dimensions $N_r \times (T - L + 1)$. The smallest distance encountered (D_{best}) and the corresponding subsequence locations are updated at every distance calculation.

The final phase of the algorithm (called scanning step hereafter) scans all pairs of subsequences in the order calculated in the referencing phase to ensure pruning most of the calculations. The scan progressed by comparing sequences that are k steps from each other in this ordered list and use the triangular inequality to calculate distances only if needed updating D_{best}. The value of k is increased from 1 to $T - L + 1$. Once a complete pass over the list is done with no update to d_{best}, it is safe to ignore all remaining pairs of subsequences and announce the pair corresponding to D_{best} to be the *exact* motif.

A better approach to discover the top K 2-motifs of a given length was suggested in [9] called MK+ that uses a single scanning rather than K-scanning runs. This approach can also be applied for every length to solve our problem.

Mohammad and Nishida [6] recently proposed an algorithm for solving the multi-length motif discovery problem (by iteratively running a modified version of MK) called MK++. The MK++ algorithm starts by detecting 2-motifs at the shortest length (L_{min}) and progressively finds 2-motifs at higher lengths. The algorithm keeps three lists: D_{bests} representing a sorted list of K best distances encountered so far and L_{bests} representing the 2-occurrence motif corresponding to each member of D_{bests}, and μ_{bests} keeping track of the means of the subsequences in L_{bests}. The *best-so-far* variable of MK is always assigned to the maximum value in D_{bests}. During the referencing phase, the distance between the current reference subsequences and all other subsequences of length L_{min} that do not overlap it with more than $wMO \times L_{min}$ points are calculated. For each of these distances (d) we apply the following rules in order:

Rule 1. If the new pair is overlapping the corresponding $L_{bests}(i)$ pair with more than $wMO \times L$ points, then this i is the index in D_{bests} to be considered

Rule 2. If *Rule1* applies and $D < D_{bests}(i)$, then replace $L_{bests}(i)$ with P.

Rule 3. If *Rule1* does not apply but $D < D_{bests}(i)$, then we search L_{bests} for all pairs $L_{bests}(i)$ for which *Rule1* applies and remove them from the list. After that the new pair P is inserted in the current location of L_{bests} and D in the corresponding location of D_{bests}

The main problem with MK++ is that it assumes that the distance function is nondecreasing which makes it inappropriate for scale-invariant distance functions.

4 Incremental Scale-Invariant Distance Calculation

We utilize the following notation: x_k is the k's element of the timeseries x where x is an ordered list of real numbers of length $L \geq l$. The symbols μ_x^l, σ_x^l,

mx_x^l,mn_x^l stand for the mean, standard deviation, maximum and minimum of x^l. The normalization constant r_x^l is assumed to be a real number calculated from x^l and is used in this paper to achieve scale-invariance by either letting $r_x^l = \sigma_x^l$ (zscore normalization), or $r_x^l = mx_x^l - mn_x^l$ (range normalization). The distance function (between any two timeseries x and y) used in this paper has the general form:

$$D_{xy}^l = \sum\nolimits_{k=0}^{l-1} \left(\frac{x_k - \mu_x^l}{r_x^l} - \frac{y_k - \mu_y^l}{r_y^l} \right)^2 \tag{2}$$

This is an Euclidean distance between two subsequences $\bar{x}$ and $\bar{y}$, where $\bar{z}_k = \left(z_k^l - \mu_z^l\right)/r_z^l$. This means that it satisfies the triangular inequality which allows us to use the speedup strategy described in section 3. Nevertheless, because of the dependence of r_x^l and r_y^l on data and length, it is no longer true that $D_{xy}^{l+1} \geq D_{xy}^l$ and we cannot directly use MK++ [6]. Moreover, once any of these two values change, we can no longer use any catched values of $\bar{x}$ and $\bar{y}$.

We need few more definitions: $\alpha_x^l = r_x^{l-1}/r_x^l$, $\theta_{xy}^l = r_x^l/r_y^l$, $\mathit{\Delta}_k^l = x_k - \theta_{xy}^l y_k$, ${}^n\mu_{xy}^m = \mu_x^m - \theta_{xy}^n \mu_y^m$ and $\mu_{xy}^l = {}^l\mu_{xy}^l$. Notice that it is trivial to prove that the mean of the sequence $\left\langle \mathit{\Delta}_{xy}{}^l \right\rangle$ is equal to μ_{xy}^l.

The first contribution of this paper is a novel incremental formula for calculating scale invariant distances between time-series subsequences which is stated in the following theorem:

Theorem 1. For any two timeseries x and y of lengths $L_x > l$ and $L_y > l$, and using a normalized distance function D_{xy}^l of the form shown in Equation 2, we have:

$$D_{xy}^{l+1} = D_{xy}^l + \frac{1}{\left(r_x^l\right)^2} \left(\begin{array}{l} \left(\left(\alpha_x^l\right)^2 - 1 \right) \displaystyle\sum_{k=0}^{l-1} \left({x_k}^2\right) + 2 \left(\theta_{xy}^l - \left(\alpha_x^l\right)^2 \theta_{xy}^{l+1} \right) \displaystyle\sum_{k=0}^{l-1} x_k y_k \\ + \left(\left(\alpha_x^{l+1} \theta_{xy}^{l+1}\right)^2 - \left(\theta_{xy}^l\right)^2 \right) \displaystyle\sum_{k=0}^{l-1} \left({y_k}^2\right) + l\left(\mu_{xy}^l\right)^2 + \left(\alpha_x^{l+1} \mathit{\Delta}_l^{l+1}\right)^2 \\ - (l+1) \left(\alpha_x^{l+1} \mu_{xy}^{l+1}\right)^2 \end{array} \right)$$

A sketch of the proof for Theorem 1 is:

$$D_{xy}^l = \sum_{k=0}^{l-1} \left(\frac{x_k - \mu_x^l}{r_x^l} - \frac{y_k - \mu_y^l}{r_y^l} \right)^2 = \left(\frac{1}{r_x^l}\right)^2 \sum_{k=0}^{l-1} \left(x_k - \mu_x^l - \theta_x y^l \left(y_k - \mu_y^l\right) \right)^2$$

$$\therefore D_{xy}^l(x, y) = \left(\frac{1}{r_x^l}\right)^2 \sum_{k=0}^{l-1} \left(\mathit{\Delta}_k^l - \mu_{xy}^l\right)^2$$

Notice that this is the form of a variance equation (since the mean of the sequence $\left\langle \mathit{\Delta}_{xy}{}^l \right\rangle$ is equal to μ_{xy}^l) and by simple manipulations we can arrive at the following equation:

$$D_{xy}^l = \left(r_x^l\right)^{-2} \left(-l\left(\mu_{xy}^l\right)^2 + \sum\nolimits_{k=0}^{l-1} \left(\mathit{\Delta}_k^l\right)^2 \right) \tag{3}$$

From Equation 2, it follows that:

$$D_{xy}^{l+1} = \left(r_x^{l+1}\right)^{-2} \left(-\left(l+1\right)\left(\mu_{xy}^{l+1}\right)^2 + \sum\nolimits_{k=0}^{l} \left(\Delta_k^{l+1}\right)^2\right) \tag{4}$$

Subtracting Equations 3 from Equation 4, using the definitions of α and θ given in this section and after some manipulations we get the equation in Theorem 1.

The important point about Theorem 1, is that it shows that by having a running sum of x_k, y_k, $(x_k)^2$, $(y_k)^2$, and $x_k y_k$, we can incrementally calculate the scale invariant distance function for any length l given its value for the previous length $l-1$. This allows us to extend the MK+ algorithm directly to handle *all* motif lengths required in parallel rather than solving the problem for each length serially as was done in MK++.

The form of D_{xy}^{l+1} as a function of D_{xy}^{l} is quite complicated but it can be simplified tremendously if we have another assumption:

Lemma 1. For any two timeseries x and y of lengths $L_x > l$ and $L_y > l$, and using a normalized distance function D_{xy}^{l} of the form shown in Equation 2, and assuming that $r_x^{l+1} = r_x^l$ and $r_y^{l+1} = r_y^l$, we have:

$$D_{xy}^{l+1} = D_{xy}^{l} + \frac{1}{\left(r_x^l\right)^2}\left(\frac{l}{l+1}\right)\left(\mu_{xy}^{l} - \Delta_l^l\right)^2$$

Lemma 1 can be proved by substituting in Theorem 1 noticing that given the assumptions about r_x^l and r_y^l, we have $\Delta_k^{l+1} = \Delta_k^l$ and ${}^{l+1}\mu_{xy}^{l+1} = {}^{l}\mu_{xy}^{l+1}$.

What Lemma 1 shows is that if the normalization constant did not change with increased length, we need only to use the running sum of x_k and y_k for calculating the distance function incrementally and using a much simpler formula. This suggests that the normalization constant should be selected to change as infrequently as possible while keeping the scale invariance nature of the distance function. The most used normalization method to achieve scale invariance is zscore normalization in which $r_x^l = \sigma_x^l$. In this paper we propose using the – less frequently used – range normalization ($r_x^l = mx_x^l - mn_x^l$) because the normalization constants change much less frequently. To support this claim we conducted two experiments. In the first experiment, we generated 100 timeseries pairs of length 1000 each using random walks and calculated the fraction of time in which either r_x^l or r_y^l changed using both zscore and range normalization. The zscore normalization constant changed 15.01% of the time while the range normalization constant changed only 0.092% of the time. In the second experiment, we used 50 timeseries representing the angles of wrist and elbow joints of an actor while generating free gestures as a real world dataset. The zscore normalization constant changed 34.2% of the time while the range normalization constant changed only 0.11% of the time. This suggests that just ignoring the change in the normalization constant would not affect the quality of returned 2-motifs even though it will render the algorithm approximate.

The formulas for incremental evaluation of the distance function given in this section assume that the change in length is a single point. Both formulas can be extended to the case of any difference in the length but proofs are much more involved and due to lack of space will not be presented.

5 Proposed Algorithm

The second contribution of this paper is to use the incremental normalized distance calculation formulas of Theorem 1 and Lemma 1 to extend the MK algorithm to handle the scale-invariant multi-length 2-motif discovery problem stated in section 2.

The first approach – as suggested by Lemma 1– is to use the range normalization and modify the calculation of distances in the D_{bests} list using the formula proposed in Lemma 1. Notice that during the scanning phase, the algorithm will decide to ignore pairs of subsequences based on the distances between them and reference points. We can either keep the exactness of the algorithm by recalculating the distance between the pair and all reference points at every length using the formula given in Theorem 1 or accept an approximate solution (that should not be much worse than the exact one by Lemma 1) and use the distances to reference points from previous lengths as lower bounds. In this paper we choose the second (approximate) method to maximize the speed of the algorithm. This approach is called MK++ for the rest of this paper. We will show that the proposed algorithm is faster than this *approximate* solution while being an exact algorithm.

The second approach is to use the formula in Theorem 1 and run the two phases of the MK algorithm in parallel for all lengths. The algorithm starts similarly to MK+ by calculating the distance between all subsequences of the minimum length and a randomly selected set of reference points. These distances will be used later to find lower bounds during the scanning phase. Based on the variance of the distances associated with reference points, these points are ordered. The subsequences of the timeseries are then ordered according to their distances to the reference point with maximum variance. These steps can be achieved in $O(nlogn)$ operations. The distance function used in these steps (D_{full}) uses Equation 2 for distance calculation but in the same time keeps the five running summation (x_k, y_k, $(x_k)^2$, $(y_k)^2$, and $x_k y_k$) needed for future incremental distance calculations as well as the maximum and minimum of each subsequence. After each distance calculation the structure S^l_{bests} is updated to keep the top K motifs at this length with associated running summations using the same three rules of MK++ (see section 3).

The next step is to calculate the S^l_{bests} list storing distances and running summations for all lengths above the minimum length using the function D_{inc} which utilized Theorem 1 to find the distances at longer lengths. The list is then sorted at every length. Both D_{inc} and D_{full} update the bsf variable which contains the best-so-far distance at all lengths and is used if the run is approximate to further prune out distance calculations during the scanning phase.

The scanning phase is then started in which the subsequences as ordered in the previous phase are taken in order and compared with increasing offset between them. If a complete run at a specific length did not pass the lower-bound test , we can safely ignore all future distance calculations at that length because by the triangular inequality we know that these distances can never be lower than the ones we have in S^l_{bests}. Scanning stops when all lengths are fully scanned.

During scanning we make use of Theorem 1 once more by using an incremental distance calculation to find the distances to reference points and between currently tested subsequences. If we accept approximate results based on Lemma 1, we can speed things up even more by not calculating the distances to reference points during the evaluation of the lower bound and by avoiding this step all-together if the distance at lower length was more than the current maximum distance in S^l_{bests}.

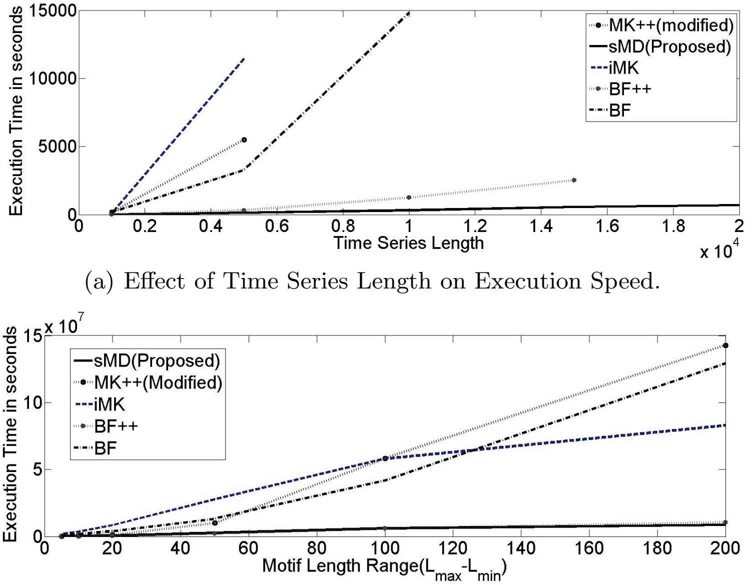
6 Evaluation

We conducted a series of experiments to evaluate the proposed approach to existing state of the art exact motif discovery algorithms. We evaluated MK++ (with the modifications discussed in section 5) and sMD proposed in this paper to the following algorithms: iterative application of the MK algorithm (iMK), the brute force approach of just comparing all possible pairs (using only bsf to prune calculations) at all lengths (called BF from now on), the brute-force algorithm but utilizing the incremental distance calculation proposed in section 4.

In the first experiment, we evaluated the five algorithms for scalability relative to the timeseries length. We used the EEG trace dataset from [12] and applied the algorithm to the first subsequence of length 1000, 5000, 10000, 15000, and 20000points. The motif range was 64 to 256 points and K was 15. Because it is always the case for all of these algorithms that execution time will increase with increased length, we did not evaluate any algorithm for lengths larger than the one at which its execution time exceeded one hour. Fig. 1-a shows the execution time in seconds of the six algorithms as a function of timeseries length. Notice that sMD, and BF++ outperform all other algorithms for even moderate lengths. The fact that the brute-force algorithm is better than iterative MK and even MK++ supports the effectiveness of the proposed incremental distance calculations because both iterative MK and MK++ cannot effectively utilize it.

The second experiment explored the effect of motif length range. We used the same dataset used in the first experiment and a fixed timeseries length of 1000 points. We used a minimum motif length of 50 and varied the maximum motif length from 54 to 249. K was 15 again and we stopped the execution in the same fashion as in the first experiment. Fig. 1-b shows the execution time in seconds of the six algorithms as a function of the motif length range. Again sMD, and BF++ outperform the other three algorithms.

In the final experiment, we tested the application of the proposed algorithm as the basis for ARM discovery by first deleting all 2-motifs of length l that are covered by 2-motifs of a higher length then combine 2-motifs at each length if either of their subsequences are overlapping more than a predefined threshold. We used the CMU Motion Capture Dataset available online from [2]. All the timeseries corresponding to basketball and general exercise and stretching categories were used. The occurrences of each recurring motion pattern in the time



(a) Effect of Time Series Length on Execution Speed.

(b) Effect of Motif Length Range on Execution Speed.

Fig. 1. Comparing scalability of the proposed algorithm (sMD) with other exact motif discovery algorithms. See text for details.

series of the 20 available in this collection were marked by hand to get ground truth information about the locations of different motifs. The total number of frames in the 20 time series was 37788. Timeseries length ranged from 301 to 5357 points each. Before applying motif discovery algorithms, we reduced the dimensionality of each time series using Principal Component Analysis (PCA). To speedup PCA calculations, we used a random set of 500 frames and applied SVD to it then projected the whole time series on the direction of the first Eigen vector.

We applied sMD, MK++, PROJECTIONS [13], and MCFull [7] with a motif length between 100 and 300 to the 20 time series and calculated the accuracy and execution time for each of these five algorithms. The proposed algorithm achieved the highest accuracy (87% compared with 83% for MK++, 74% for MCFull, and 64% for the PROJECTIONS algorithm) and shortest execution time (0.0312 seconds per frame compared with 0.63 seconds for MK++, 3.2 for MCFull, and 10.3 seconds per frame for PROJECTIONS). These results show that the proposed algorithm is applicable to real-world motif discovery.

7 Conclusions

This paper presented an incremental formula for calculating scale invariant distances between timeseries. This formula was then used to design an algorithm for scale invariant multi-length exact motif discovery (called sMD). The proposed algorithm was evaluated against brute-force solution of the problem and two other motif discovery algorithms (MK++ and iterative application of the MK algorithm). The proposed algorithm is an order of magnitude faster than both of them for timeseries of moderate size (10000points).

The work reported in this paper opens several directions of future research. The most obvious direction is parallelizing the scanning phase of the algorithm. Another direction is integrating the proposed incremental distance calculation method, the lower bound used in MOEN and the pruning technique of the MK algorithm to provide even faster exact motif discovery. A third direction of future research is to utilize top-down processing (i.e. from higher to lower motif lengths) in conjunction with the bottom-up processing of the proposed algorithm to guarantee that 2-motifs found at every length are not overlapping those at higher lengths.

References

1. Chiu, B., Keogh, E., Lonardi, S.: Probabilistic discovery of time series motifs. In: ACM KDD 2003, pp. 493–498 (2003)
2. CMU: Cmu motion capture dataset, `http://mocap.cs.cmu.edu`
3. Keogh, E., Lin, J., Fu, A.: Hot sax: Efficiently finding the most unusual time series subsequence. In: IEEE ICDM, pp. 8–16 (November 2005)
4. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Mining and Knowledge Discovery 7(4), 349–371 (2003)
5. Minnen, D., Starner, T., Essa, I., Isbell, C.: Improving activity discovery with automatic neighborhood estimation. In: IJCAI (2007)
6. Mohammad, Y., Nishida, T.: Exact discovery of length-range motifs. In: Nguyen, N.T., Attachoo, B., Trawiński, B., Somboonviwat, K. (eds.) ACIIDS 2014, Part II. LNCS, vol. 8398, pp. 23–32. Springer, Heidelberg (2014)
7. Mohammad, Y., Nishida, T.: Constrained motif discovery in time series. New Generation Computing 27(4), 319–346 (2009)
8. Mohammad, Y., Nishida, T.: Fluid imitation: Discovering what to imitate. International Journal of Social Robotics 4(4), 369–382 (2012)
9. Mohammad, Y., Nishida, T.: Unsupervised discovery of basic human actions from activity recording datasets. In: Proceedings of the IEEE/SICE SII (2012)
10. Mohammad, Y., Nishida, T., Okada, S.: Unsupervised simultaneous learning of gestures, actions and their associations for human-robot interaction. In: Proceedings of IROS 2009, pp. 2537–2544 (2009)
11. Mueen, A.: Enumeration of time series motifs of all lengths. In: 2013 IEEE 13th International Conference on Data Mining (ICDM). IEEE (2013)
12. Mueen, A., Keogh, E.J., Zhu, Q., Cash, S., Westover, M.B.: Exact discovery of time series motifs. In: SDM, pp. 473–484 (2009)
13. Patel, P., Keogh, E., Lin, J., Lonardi, S.: Mining motifs in massive time series databases. In: IEEE International Conference on Data Mining, pp. 370–377 (2002)