Discovering Deformable Motifs in Continuous Time Series data

Suchi Saria*
Computer Science Department
Stanford University, CA
ssaria@cs.stanford.edu

Andrew Duchi*
Computer Science Department
Stanford University, CA
aduchi@stanford.edu

Daphne Koller
Computer Science Department
Stanford University, CA
koller@cs.stanford.edu

Abstract

Continuous time series data often comprise or contain repeated *motifs* — patterns that have similar shape, and yet exhibit nontrivial variability. Identifying these motifs, even in the presence of variation, is an important subtask in both unsupervised knowledge discovery and constructing useful features for discriminative tasks. This paper addresses this task using a probabilistic framework that models generation of data as switching between a random walk state and states that generate motifs. A motif is generated from a continuous shape template that can undergo non-linear transformations such as temporal warping and additive noise. We propose an unsupervised algorithm that simultaneously discovers both the set of canonical shape templates and a template-specific model of variability manifested in the data. Experimental results on three real-world data sets demonstrate that our model is able to recover templates in data where repeated instances show large variability. The recovered templates provide higher classification accuracy and coverage when compared to those from alternatives such as random projection based methods and simpler generative models that do not model variability. Moreover, in analyzing physiological signals from infants in the ICU, we discover both known signatures as well as novel physiomarkers.

1 Introduction and Background

Continuous-valued time series data are collected in multiple domains, including surveillance, pose tracking, ICU patient monitoring, and finance. These time series often contain *motifs* — segments that repeat within and across different series. For example, in trajectories of people at an airport, we might see repeated motifs in a person checking in at the ticket counter, stopping to buy food, etc. In pose tracking, we might see characteristic patterns such as bending down, sitting, kicking, etc. And in physiologic signals, recognizable shapes such as bradycardia and apnea are known to precede severe complications such as infection. Discovering these repeated segments can provide primitives that are useful for domain understanding and as higher-level, meaningful features that can be used to segment time series or discriminate among time series data from different groups.

In many domains, different instances of the same motif can be structurally similar but vary greatly in terms of pointwise distance [Höppner, 2002]. For example, the temporal position profile of the body in a front kick can vary greatly, depending on how quickly the leg is raised, the extent to which it is raised and then how quickly it is brought back to position. Yet, these profiles are structurally similar, and different from that of a round-house kick. Bradycardia and apnea are also known to manifest significant variation in both amplitude and temporal duration. Our goal is to deal with the unsupervised discovery of these *deformable* motifs in continuous time series data.

Much work has been done on the problem of motif detection in continuous time series data. One very popular and successful approach is the work of Keogh and colleagues (e.g., [Mueen *et al.*, 2009]), in which a motif is defined via a pair of windows of the same length that are closely matched in terms of Euclidean distance. Such pairs are identified via a sliding window approach followed by random projections to identify highly similar pairs that have not been previously identified. However, this method is not geared towards finding motifs that can exhibit significant deformation. Another line of work tries to find regions of high density in the space of all subsequences via clustering; see Oates [2002]; Denton [2005] and more recently Minnen et. al. [2007]. These works define a motif as a vector of means and variances over the length of the window, a representation that also is not geared to capturing deformable motifs. Of these methods, only the work of Minnen et. al. [2007] addresses deformation, using dynamic time warping to measure warped distance. However, motifs often exhibit structured transformations, where the warp changes gradually over time. As we show in our results, encoding this bias greatly improves performance. The work of Listgarten et. al. [2005]; Kim et. al. [2006] focus on developing a probabilistic model for aligning sequences that exhibit variability. However, these methods rely on having a segmentation of the time series into corresponding motifs. This assumption allows them to impose relatively few constraints on the model, rendering them

[*] equal contribution

highly under-constrained in our unsupervised setting.

This paper proposes a method, which we call CSTM (Continuous Shape Template Model), that is specifically targeted to the task of unsupervised discovery of deformable motifs in continuous time series data. CSTM seeks to explain the entire data in terms of repeated, warped motifs interspersed with non-repeating segments. In particular, we define a hidden, segmental Markov model in which each state either generates a motif or samples from a non-repeated random walk (NRW). The individual motifs are represented by smooth continuous functions that are subject to non-linear warp and scale transformations. Our warp model is inspired by Listgarten et. al. [2005], but utilizes a significantly more constrained version, more suited to our task. We learn both the motifs and their allowed warps in an unsupervised way from unsegmented time series data. We demonstrate the applicability of CSTM to three distinct real-world domains and show that it achieves considerably better performance than previous methods, which were not tailored to this task.

2 Generative Model

The CSTM model assumes that the observed time series is generated by switching between a state that generates non-repeating segments and states that generate repeating (structurally similar) segments or *motifs*. Motifs are generated as samples from a shape template that can undergo non-linear transformations such as shrinkage, amplification or local shifts. The transformations applied at each observed time t for a sequence are tracked via latent states, the distribution over which is inferred. Simultaneously, the canonical shape template and the likelihood of possible transformations for each template are learned from the data. The random-walk state generates trajectory data without long-term memory. Thus, these segments lack repetitive structural patterns. Below, we describe more formally the components of the CSTM generative model. In Table 1, we summarize the notation used for each component of the model.

A Canonical Shape Template (CST)

Each shape template, indexed by k, is represented as a continuous function $s^k(l)$ where $l \in (0, L^k]$ and L^k is the length of the kth template. Although the choice of function class for s^k is flexible, a parameterization that encodes the property of motifs expected to be present in given data will yield better results. In many domains, motifs appear as smooth functions. A possible representation might be an L_2-regularized Markovian model. However, these penalize smooth functions with higher curvature more than those with lower curvature, a bias not always justified. A promising alternative is piecewise Bézier splines [Gallier, 1999]. Shape templates of varying complexity are intuitively represented by using fewer or more pieces. For our purpose, it suffices to present the mathematics for the case of piecewise third order Bézier curves over two dimensions, where the first dimension is the time t and the second diemsnion is the signal value.

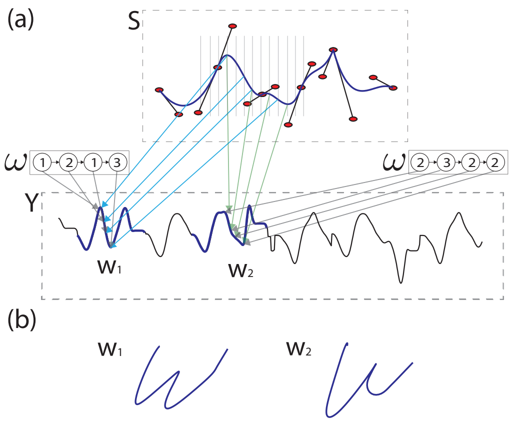

Figure 1: a) The template S shows the canonical shape for the pen-tip velocity along the x-dimension and a piecewise Bézier fit to the signal. The generation of two different transformed versions of the template are shown; for simplicity, we assume only a temporal warp is used and ω tracks the warp at each time, b) The resulting character 'w' generated by integrating velocities along both the x and y dimension.

A third order Bezier curve is parameterized by four points $p_i \in \mathcal{R}^2$ for $i \in 0, \cdots, 3$. Control points p_0 and p_3 are the start and end of each curve piece in the template and shared between adjacent pieces (see figure 1). The intermediate control points p_1 and p_2 control the tangent directions of the curve at the start and end, as well as the interpolation shape. Between end points, $\tau \in [0, 1]$ controls the position on the curve, and each piece of the curve is interpolated as

$$f(\tau) = \sum_{i=0}^{3} \binom{3}{i} (1-\tau)^{3-i} \tau^i p_i \qquad (1)$$

For higher-dimensional signals in $\mathcal{R}^n$, $p_i \in \mathcal{R}^{n+1}$. Although only $\mathcal{C}^0$ continuity is imposed here, it is possible to impose arbitrary continuity within this framework of piecewise Bezier curves if such additional bias is relevant.

Shape Transformation Model

Motifs are generated by non-uniform sampling and scaling of s^k. Temporal warp can be introduced by moving slowly or quickly through s^k. The allowable temporal warps are specified as an ordered set $\{w_1, \ldots, w_n\}$ of time increments that determines the rate at which we advance through s^k. A template-specific warp transition matrix π^k_ω specifies the probability of transitions between warp states. To generate a series $y_1, \cdots, y_T$, let $\omega_t \in \{w_1, \ldots, w_n\}$ be the random variable tracking the warp and ρ_t be the position within the template s^k at time t. Then, y_{t+1} would be generated from the value $s^k(\rho_{t+1})$ where $\rho_{t+1} = \rho_t + \omega_{t+1}$ and $\omega_{t+1} \sim \pi^k_\omega(\omega_t)$. (For all our experiments, the allowable warps are $\{1, 2, 3\}\delta t$ where δt is the sampling rate; this posits that the longest sequence from s^k is at most three times the shortest sequence sampled from it.)

We also want to model scale deformations. Analogously, the set of allowable scaling coefficients are maintained as the set $\{c_1, \ldots, c_n\}$. Let $\phi_{t+1} \in \{c_1, \ldots, c_n\}$ be the sampled scale value at time $t+1$, sampled from the scale transition matrix π^k_ϕ. Thus, the observation y_{t+1} would be generated around the value $\phi_{t+1}s^k(\rho_{t+1})$, a scaled version of the value of the motif at ρ_{t+1}, where $\phi_{t+1} \sim \pi^k_\phi(\phi_t)$. Finally, an additive noise value $\nu_{t+1} \sim \mathcal{N}(0, \dot{\sigma})$ models small shifts. The parameter $\dot{\sigma}$ is shared across all templates.

In summary, putting together all three possible deformations, we have that $y_{t+1} = \nu_{t+1} + \phi_{t+1}s^k(\rho_{t+1})$. We use $z_t = \{\rho_t, \phi_t, \nu_t\}$ to represent the values of all transformations at time t.

In many natural domains, motion models are often smooth due to inertia. For example, while kicking, as the person gets tired, he may decrease the pace at which he raises his leg. But, the decrease in his pace is likely to be smooth rather than transitioning between a very fast and very slow pace from one time step to another. One simple way to capture this bias is by constraining the scale and warp transition matrices to be band diagonal. Specifically, $\phi_\omega(w, w') = 0$ if $|w - w'| > b$ where $2b + 1$ is the size of the band. (We set $b = 1$ for all our experiments.) Experimentally, we observe that in the absence of such a prior, the model is able to align random walk sequences to motif sequences by switching arbitrarily between transformation states, leading to noisy templates and poor performance.

Non-repeating Random Walk (NRW)

We use the NRW model to capture data not generated from the templates (see also Denton [2005]). If this data has different noise characteristics, our task becomes simpler as the noise characteristics can help disambiguate between motif-generated segments and NRW segments. The generation of smooth series can be modeled using an autoregressive process. We use an $AR(1)$ process for our experiments where $y_t = \mathcal{N}(y_{t-1}, \sigma)$. We refer to the NRW model as the 0th template.

Template Transitions

Transitions between generating NRW data and motifs from the CSTs are modeled via a transition matrix, $\mathcal{T}$ of size $(K + 1) \times (K + 1)$ where the number of CSTs is K. The random variable κ_t tracks the template for an observed series. Transitions into and out of templates are only allowed at the start and end of the template, respectively. Thus, when the position within the template is at the end i.e., $\rho_{t-1} = L^{\kappa_{t-1}}$, we have that $\kappa_t \sim \mathcal{T}(\kappa_{t-1})$, otherwise $\kappa_t = \kappa_{t-1}$. For $\mathcal{T}$, we fix the self-transition parameter for the NRW state as λ, a prespecified input. Different settings of λ allows control over the proportion of data assigned to motifs versus NRW. As λ increases, more of the data is explained by the NRW state and as a result, the recovered templates have lower variance.[1]

Below, we summarize the generative process at each time t:

$$\kappa_t \sim \mathcal{T}(\kappa_{t-1}, \rho_{t-1}) \quad (2)$$

$$\omega_t \sim \pi^{\kappa_t}_\omega(\omega_{t-1}) \quad (3)$$

$$\rho_t = \begin{cases} \min(\rho_{t-1} + \omega_t, L^{\kappa_t}), & \text{if } \kappa_t = \kappa_{t-1} \quad (4) \\ 1, & \text{if } \kappa_t \neq \kappa_{t-1} \quad (5) \end{cases}$$

Symbol	Description
y_t	Observation at time t
κ_t	Index of the template used at time t
ρ_t	Position within the template at time t
ω_t	Temporal warp applied at time t
ϕ_t	Scale tranformation applied at time t
ν_t	Additive noise at time t
z_t	Vector $\{\rho_t, \phi_t, \nu_t\}$ of transformations at time t
s^k	kth template, length of template is L^k
π^k_ω	Warp transition matrix for kth template
π^k_ϕ	Scale transition matrix for kth template
$\mathcal{T}$	Transition matrix for transitions between templates and NRW

Table 1: Notation for the generative process of CSTM.

$$\phi_t \sim \pi^{\kappa_t}_\phi(\phi_t) \quad (6)$$

$$\nu_t \sim \mathcal{N}(0, \dot{\sigma}) \quad (7)$$

$$y_t = \nu_t + \phi_t s^{\kappa_t}(\rho_t) \quad (8)$$

3 Learning the model

The canonical shape templates, their template-specific transformation models, the NRW model, the template transition matrix and the latent states $(\kappa_{1:T}, z_{1:T})$ for the observed series are all inferred from the data using hard EM. Coordinate ascent is used to update model parameters in the M-step. In the E-step, given the model parameters, Viterbi is used for inferring the latent trace.

3.1 E-step

Given $y_{1:T}$ and the model $\mathcal{M}$ from the previous iteration, in the E-step, we compute assignments to the latent variables $\{\kappa_{1:T}, z_{1:T}\}$ using approximate Viterbi decoding (we use $t_1 : t_2$ as shorthand for the sequences of time indices $t_1, t_1 + 1, \ldots, t_2$):

$$\{\kappa^*_{1:T}, z^*_{1:T}\} = \text{argmax}_{\kappa_{1:T}, z_{1:T}} P(\kappa_{1:T}, z_{1:T} | y_{1:T}, \mathcal{M})$$

During the forward phase within Viterbi, at each time, we prune to maintain only the top B belief states[2]. For our experiments, we maintain $K \times 20$ states. This does not degrade performance as most transitions are highly unlikely[3].

3.2 M-step

Given the data $y_{1:T}$ and the latent trace $\{\kappa_{1:T}, z_{1:T}\}$, the model parameters are optimized by taking the gradient of the penalized complete data log-likelihood w.r.t. each parameter.

[1] Learning λ while simultaneously learning the remaining parameters leads to degenerate results where all points end up in the NRW state with learned $\lambda = 1$.

[2] Although we use pruning to speed up Viterbi, exact inference is also feasible. The cost of exact inference in this model is $\mathcal{O}(max(T * W^2 * D^2 * K, T * K^2))$ where T is the length of the series, W and D are dimensions of the warp and scale transformation matrices respectively and K is the number of templates.

[3] Pruning within segmental Hidden Markov Models has been used extensively for speech recognition. To the best of our knowledge, no theoretical guarantees exist for these pruning schemes but in practice they have been shown to perform well.

Below, we discuss the penalty for each component and the corresponding update equations.

Updating π_ω, π_ϕ and $\mathcal{T}$

A Dirichlet prior, conjugate to the multinomial distribution, is used for each row of the transition matrices as penalty $\mathcal{P}_\omega$ and $\mathcal{P}_\phi$. In both cases, the prior matrix is constrained to be band-diagonal. As a result, the posterior matrices are also band-diagonal. The update is the straightforward MAP estimate for multinomials with a Dirichlet prior, so we omit details for lack of space. For all our experiments, we set a weak prior favoring shorter warps: Dir$(7, 3, 0)$, Dir$(4, 5, 1)$ and Dir$(0, 7, 3)$ for each of the rows of π_ω and π_ϕ given our setting of allowable warps. As always, the effect of the prior decreases with larger amounts of data. In our experiments, we found the recovered templates to be insensitive to the setting for a reasonable range[4].

The template transition matrix $\mathcal{T}$ is updated similarly. A single hyperparameter $\dot\eta$ is used to control the strength of the prior. We set $\dot\eta = n/(K^2 L)$, where n is the total amount of observed data, L is the anticipated template length used in the initializations, and K is the pre-set number of templates. This is equivalent to assuming that the prior has the same strength as the data and is distributed uniformly across all shape templates. Let $\mathcal{I}(E)$ be the indicator function for the event E. To update the transitions out of the NRW state,

$$\mathcal{T}_{0,k} = (1-\lambda)\frac{\dot\eta + \sum_{t=2}^{T}\mathcal{I}(\kappa_{t-1}=0)\mathcal{I}(\kappa_t=k)}{\dot\eta K + \sum_{t=2}^{T}\mathcal{I}(\kappa_{t-1}=0)\sum_{k'=1}^{K}\mathcal{I}(\kappa_t=k')}$$

Transitions are only allowed at the end of each template. Thus, to update transitions between shape templates,

$$\mathcal{T}_{k,k'} \propto \dot\eta + \sum_{t=2}^{T}\mathcal{I}(\kappa_{t-1}=k)\mathcal{I}(\kappa_t=k')\mathcal{I}(\rho_{t-1}=L^k)$$

Fitting Shape Templates

Given the scaled and aligned segments of all observed time series assigned to any given shape template s^k, the smooth piecewise function can be fitted independently for each shape template. Thus, collecting terms from the log-likelihood relevant for fitting each template, we get:

$$\mathcal{L}_k = -\mathcal{P}_{s^k} - \sum_{t=1}^{T}\mathcal{I}(\kappa_t=k)\frac{(y_t - \phi_t s^k(\rho_t))^2}{\dot\sigma^2} \qquad (9)$$

where $\mathcal{P}_{s^k}$ is a regularization for the kth shape template. A natural regularization for controlling model complexity is the BIC penalty [Schwarz, 1978] specified as $0.5\log(N)\nu_k$, where ν_k is the number of Bézier pieces used and N is the number of samples assigned to the template.[5]

Piecewise Bézier curve fitting to chains has been studied extensively. $\mathcal{L}_k$ is not differentiable and non-convex; a series of hill-climbing moves are typically used to get to an optimum. We employ a variant which has been commercially deployed for large and diverse image collections [Diebel, 2008].

Updating σ and $\dot\sigma$

Given the assignments of the data to the NRW and the template states, and the fitted template functions, the variances σ and $\dot\sigma$ are computed easily.

3.3 Escaping local maxima

EM is known to get stuck in local maxima, having too many clusters in one part of the space and too few in another [Ueda *et al.*, 1998]. Split and merge steps can help escape these configurations by: a) splitting clusters that have high variance due to the assignment of a mixture of series, and b) merging similar clusters. At each such step, for each existing template k, 2-means clustering is run on the aligned segmentations. Let $k1$ and $k2$ be the indices representing the two new clusters created from splitting the kth cluster. Then, the split score L_k^{split} for each cluster is L_{k1} + L_{k2} - L_k where L_i defines the observation likelihood of the data in cluster i. The merge score for two template clusters $L_{k'k''}^{merge}$ is computed by defining the likelihood score based on the center of the new cluster (indexed by $k'k''$) inferred from all time series assigned to both clusters k' and k'' being merged. Thus, $L_{k'k''}^{merge} = L_{k'k''} - L_{k'} - L_{k''}$. A split-merge step with candidate clusters (k, k', k'') is accepted if $L_k^{split} + L_{k'k''}^{merge} > 0$.[6]

3.4 Peak-based initialization

The choice of window length is not always obvious, especially in domains where motifs show considerable warp. An alternative approach is to describe the desired motifs in terms of their structural complexity — the number of distinct peaks in the motif. Given such a specification, we first characterize a segment s in the continuous time series by the set of its extremal points [Fink and Gandhi, 2010] — their heights $f_1^s, \cdots, f_M^s$ and positions $t_1^s, \ldots, t_M^s$. We can find segments of the desired structure using a simple sliding window approach, in which we extract segments s that contain the given number of extremal points (we only consider windows in which the boundary points are extremal points). We now define $\delta_m^s = f_m^s - f_{m-1}^s$ (taking $f_0^s = 0$), that is, the height difference between two consecutive peaks. The *peak profile* $\delta_1^s, \ldots, \delta_M^s$ is a warp invariant signature for the window: two windows that have the same structure but undergo only temporal warp have the same peak profile. Multidimensional signals are handled by concatenating the peak profile of each dimension. We now define the distance between two segments with the same number of peaks as the weighted sum of the L_2 distance of their peak profile and the L_2 distance of the times at which the peaks occur:

$$d(s, s') = \sum_{m=1}^{M}\|\delta_m^s - \delta_m^{s'}\|_2 + \eta\sum_{m=1}^{M}\|t_m^s - t_m^{s'}\|_2 \qquad (10)$$

[4] If the motifs exhibit large unstructured warp, the prior over the rows of the warp matrices can be initialized as a symmetric Dirichlet distribution. However, as seen in our experiments, we posit that in natural domains, having a structured prior improves recovery.

[5] A modified BIC penalty of $\gamma(0.5\log(N)\nu_k)$ can be used if further tuning is desired. Higher values of γ lead to smoother curves.

[6] In order to avoid curve fitting exhaustively to all candidate pairs for the merge move, we propose plausible pairs based on the distance between their template means, and then evaluate them using the correct objective.

The parameter η controls the extent to which temporal warp is considered in the similarity metric (for example, $\eta = 0$ defines an entirely warp-invariant distance); we use $\eta = 1$. Using the metric d, we cluster segments (using e.g., kmeans) and select the top K most compact clusters as an initialization for CSTM. Compactness is evaluated as the distance between all segments in the cluster to a single segment in the cluster, minimized over the choice of this segment.

4 Experiments and Results

We evaluated the performance of our CSTM model on four different datasets. We compare on both classification accuracy and coverage, comparing to the widely-used random-projection-based methods of Mueen et. al. [2009]; Chiu et. al. [2003]. We also compare against variants of our model to elucidate the importance of novel bias our model imposes over prior work. We give a brief overview of our experimental setup before describing our results.

4.1 Experimental Overview

Datasets and Metric

The *Character* data is a collection of x and y-pen tip velocities generated by writing characters on a tablet [Keogh and Folias, 2002]. We concatenated the individual series to form a set of labeled unsegmented data for motif discovery.

The *Kinect Exercise* data was created using Microsoft Kinect. The data features six leg exercises such as front kick, rotation, and knee high, interspersed with other miscellaneous activity as the subject relaxes. The dataset was collected in two different settings. We extract the three dimensional coordinates of the ankle.

A *Simulated* dataset of seven hand-drawn curves was used to evaluate how model performance degrades under different amounts of non-repeating segments. With these templates and a random intialization of our model, we generated four different datasets where the proportions of non-repeating segments were 10%, 25%, 50% and 80%.

On heart rate data collected from infants in the Edinburgh *NICU* [Williams *et al.*, 2005], our goal is to find known and novel clinical physiomarkers. This dataset is not fully labeled, but provides labeled examples of bradycardia. Our work was primarily motivated by settings such as this where simple clustering fails due to the amount of warp and non-repeating segments in the data.

On the first three datasets which are fully labeled, we evaluate the quality of our recovered templates using classification. We treat the discovered motifs as the feature basis and their relative proportions within a segment as the feature vector for that segment. Thus, for each true motif class (e.g., a character or action) a mean feature vector is computed from the training set. On the test set, each true motif is assigned a label based on distance between its feature vector and the mean feature vector for each class. Classification performance on the test set are reported. This way of measuring accuracy is less sensitive to the number of templates used.

Baseline Methods

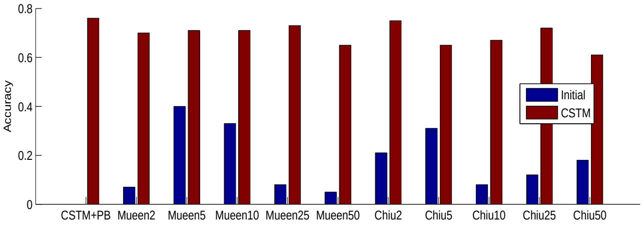

Figure 2: Comparison of our model with an initialization using the peak-based method, and initializations from Mueen and Chiu with different settings for R.

Mueen [Mueen *et al.*, 2009], repeatedly finds the closest matched pair from the set of all candidate windows. To avoid finding similar sequences at consecutive iterations, we use a sliding window to remove windows within distance d times the distances between the closest pair and iterate. We refer to this method as Mueen$d$. We also experiment with **Chiu** [Chiu *et al.*, 2003], a method widely used for motif-discovery. Unlike Mueen, Chiu selects a motif at each iteration based on its frequency in the data. For both methods, to extract matches to existing motifs on a test sequence, at each point, we compute the distance to all motifs at all shifts and label the point with the closest matched motif.

Since prior works [Minnen *et al.*, 2007] have extensively used dynamic time warping for computing similarity between warped subsequences, we define the variant **CSTM-DTW** where each row of the warp matrix is set to be the uniform distribution. **CSTM-NW** allows no warps. We also define the variant **CSTM-MC** which represents the motif as a simple template encoded as a mean vector (one for each point), as done in majority of prior works [Oates, 2002; Minnen *et al.*, 2007; Denton, 2005].

4.2 Results

Our method, similar to prior work, requires an input of the template length and the number of templates K. When characterizing the motif length is unintuitive, peak based initialization can be used to define the initial templates based on complexity of the desired motifs. In addition, our method requires a setting of the NRW self-transition parameter: λ controls the tightness of the recovered templates and can be incrementally increased (or decreased) to tune to desiderata. A non-trivial partition of the data is obtained when $\lambda < \frac{\sigma}{\dot{\sigma}} * 1/w$ where w is the maximum allowed warp.[7] In all our experiments, we set $\lambda = 0.5$, a value which respects this constraint. We subsample the data at the Nyquist frequency [Nyquist, 1928][8].

Character Data. On the character dataset, for different settings of the parameters, number of clusters and the distance d, we computed classification accuracies for Mueen and Chiu. The window length is easy to infer for this data even without knowledge of the actual labels; we set it to be 15 (in the subsampled version). We experiment with different initializations for CSTM: using the motifs derived by the meth-

[7] Essentially, this inequality is derived by comparing the likelihood of NRW to the template state for a given data and eliminating terms that are approximately equal.

[8] Intuitively, this is the highest frequency at which there is still information in the signal.

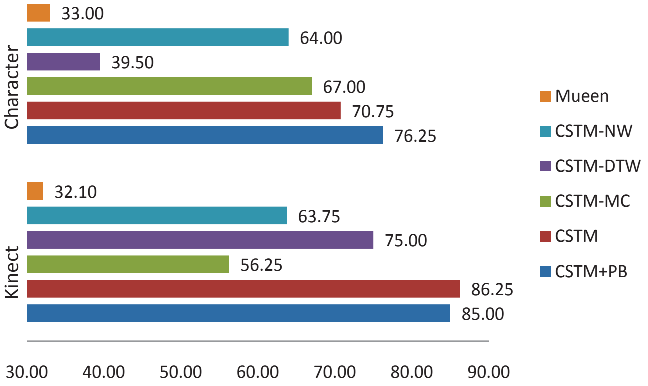

Figure 3: Accuracy on Character (top) and Kinect (bottom) for CSTM and its variants. Two different initializations for CSTM are compared: Mueen10 and peak-based.

ods of Mueen and Chiu, and those derived using the peak-based initialization. Figure 2 shows the classification accuracies for these different initializations. The performance of a random classifier for this dataset is 4.8%. Our method consistently dominates Mueen and Chiu by a large amount and yields average and best case performance of 68.75% and 76.25% over all initializations. Our method is also relatively insensitive to the choice of initialization. Our best performance is achieved by initializing with the peak-based method (**CSTM+PB**) which requires no knowledge of the length of the template. Moreover, for those parameter settings where Mueen does relatively well, our model achieves significant gain by fitting warped versions of a motif to the same template. In contrast, Mueen and Chiu must expend additional templates for each warped version of a pattern, fragmenting the true data clusters and filling some templates with redundant information, thereby preventing other character patterns from being learned. Increasing the distance parameter for Mueen can capture more warped characters within the same template; however, many characters in this dataset are remarkably similar and performance suffers from their misclassification as d increases.

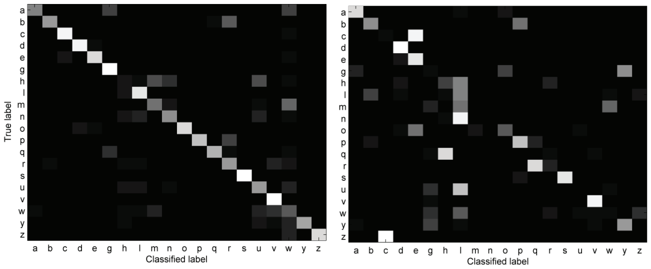

Figure 4: Confusion matrix showing performance of CSTM (left) and CSTM-DTW (right) on the character data.

In the next batch of experiments, we focused on a single initialization. Since Mueen performed better than Chiu, and is relatively more stable, we consider the Mueen10 initialization, and compare CSTM against its variants with no warp, with uniform warp, and without a template prior. In figure 3a, we see that performance degrades in all cases. A qualitative examination of the results shows that, while the no-warp version fails to align warped motifs, the DTW model aligns too freely resulting in convergence to poor optimum. The confusion matrix for CSTM-DTW in figure 4 shows that many more characters are confused when contrasted with the confusion matrix for CSTM in figure 4. Where CSTM misses, we see that it fails in intuitive ways, with many of the misclassifications occurring between similar looking letters, or letters that have similar parts; for example, we see that h is confused with m and n, p with r and w with v.

Kinect Data. Next, we tested the performance of our method on the Kinect Exercise dataset. To evaluate Mueen on this dataset, we tested Mueen with parameter settings taken from the cross product of template lengths of 5, 10, 15, or 20, distance thresholds of 2, 5, 10, 25, or 50, and a number of templates of 5, 10, 15, or 20. A mean accuracy of 20% was achieved over these 80 different parameter settings; accuracies over 50% were achieved only on 7 of the 80, and the best accuracy was 62%. Using Mueen10 as an initialization (with 10 clusters and window length 10, as above), we evaluate CSTM and its variants. CSTM achieves performance of over 86%, compared to the 32% achieved by Mueen10 directly. CSTM with a peak-based initialization (using either 5 or 7 peaks) produced very similar results, showing again the relative robustness of CSTMs to initialization. Comparing to different variants of CSTM, we see that the lack of bias in the template representation in this dataset lowers performance dramatically to 56.25%. We note that the templates here are relatively short, so, unlike the character data, the drop in performance due to unstructured warp is relatively smaller.

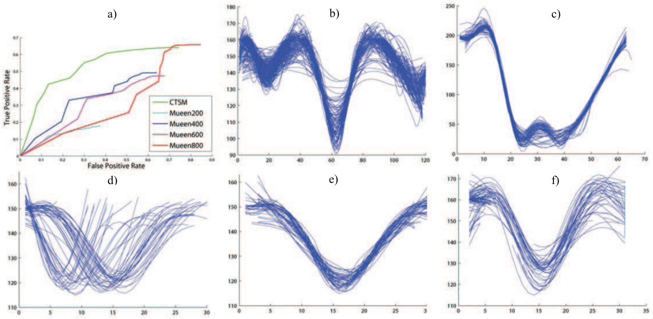

Figure 5: a) ROC curve for bradycardia, b-c) novel physiomarkers rocovered by CSTM, d) an example bradycardia cluster recovered by CSTM, e) aligned version of cluster in d, and f) an example bradycardia cluster from Mueen)

Synthetic Data. To evaluate how our model performs as the proportion of non-repeating segments increases, we evaluate the different variants of CSTM and Mueen10 on simulated data of hand-drawn curves. CSTM performance is 78% even at the 80% random walk level, and performs considerably better than Mueen10, whose performance is around 50%. Moreover, CSTM's performance ($74.5\% - 90\%$) is consistently higher than its less-structured variants ($62\% - 72\%$).

NICU Data. On the NICU data, we compute the ROC curve for identifying bradycardia (true positive and false positive measures are computed as each new cluster is added up to a total of 20 clusters). We perform a single run with peak based clustering using $3 - 7$ peaks and multiple runs for Mueen with different settings for d (see figure 5a). The ROC curve from CSTM dominates those from Mueen with significantly higher true positive rates at lower false positive rates. In 5b and 5c, we show examples of novel clusters not previously known (and potentially clinically significant). In 5d and 5f, we show clusters containing bradycardia signals generated by CSTM and Mueen respectively. The former is able to capture highly variable versions of bradycardia while those in the latter are fairly homogeneous.

5 Discussion and Conclusion

We have presented a new model for unsupervised discovery of deformable motifs in continuous time series data. Our probabilistic model seeks to explain the entire series and identify repeating and non-repeating segments. This approach allows us to model and learn important representational biases regarding the nature of deformable motifs. We demonstrate the importance of these design choices on multiple real-world domains, and show that our approach performs consistently better compared to prior works.

Our work can be extended in several ways. Our warp-invariant signatures can be used for a forward lookup within beam pruning to significantly speed up inference when K, the number of templates is large. Our current implementation requires fixing this number of clusters. However, our approach can easily be adapted to incremental data exploration, where additional templates can be introduced at a given iteration to refine existing templates or discover new templates. A Bayesian nonparametric prior is another approach that could be used to systematically control the number of classes based on model complexity. A different extension could build a hierarchy of motifs, where larger motifs are comprised of multiple occurrences of smaller motifs, thereby possibly providing an understanding of the data at different time scales. More broadly, this work can serve as a basis for building non-parametric priors over deformable multivariate curves.

6 Acknowledgements

The authors are very grateful to Christian Plagemann and Hendrik Dahlkamp for their generous help in collecting the Kinect 3D dataset. The authors would also like to thank Vladimir Jojic and Cristina Pop for feedback regarding the manuscript.

References

[Chiu *et al.*, 2003] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *KDD*, 2003.

[Denton, 2005] A. Denton. Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model. In *ICDM*, 2005.

[Diebel, 2008] J. Diebel. Bayesian Image Vectorization: the probabilistic inversion of vector image rasterization. Phd thesis, Computer Science Department, Stanford University, 2008.

[Fink and Gandhi, 2010] E. Fink and H. Gandhi. Compression of time series by extracting major extrema. In *Journal of Experimental and Theoretical AI*, 2010.

[Gallier, 1999] J. Gallier. *Curves and Surfaces in Geometric Modeling: Theory and Algorithms*. Morgan Kaufmann, 1999.

[Höppner, 2002] F. Höppner. Knowledge discovery from sequential data. 2002.

[Keogh and Folias, 2002] E. Keogh and T. Folias. UCR time series data mining archive. 2002.

[Kim *et al.*, 2006] S. Kim, P. Smyth, and S. Luther. Modeling waveform shapes with random effects segmental hidden Markov models. In *J. Mach. Learn. Res.* 2006.

[Listgarten *et al.*, 2005] J. Listgarten, R. Neal, S. Roweis, and A. Emili. Multiple alignment of continuous time series. In *NIPS*, 2005.

[Minnen *et al.*, 2007] D. Minnen, C. L. Isbell, I. Essa, and T. Starner. Discovering multivariate motifs using subsequence density estimation and greedy mixture learning. In *AAAI*, 2007.

[Mueen *et al.*, 2009] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover. Exact discovery of time series motifs. In *SDM*, 2009.

[Nyquist, 1928] H. Nyquist. Certain topics in telegraph transmission theory. 1928.

[Oates, 2002] T. Oates. PERUSE:an unsupervised algorithm for finding recurring patterns in time series. In *ICDM*, 2002.

[Schwarz, 1978] G. Schwarz. Estimating the dimension of a model. In *Annals of Statistics*. 1978.

[Ueda *et al.*, 1998] N. Ueda, R. Nakano, Z. Ghahramani, and G. Hinton. Split and merge EM algorithm for improving Gaussian mixture density estimates. In *NIPS*, 1998.

[Williams *et al.*, 2005] C. Williams, J. Quinn, and N. McIntosh. Factorial switching Kalman filters for condition monitoring in neonatal intensive care. In *NIPS*, 2005.