Spectral Clustering and Embedding with Hidden Markov Models

Tony Jebara, Yingbo Song, and Kapil Thadani

Department of Computer Science, Columbia University, New York NY 10027, USA
{jebara,yingbo,kapil}@cs.columbia.edu

Abstract. Clustering has recently enjoyed progress via spectral methods which group data using only pairwise affinities and avoid parametric assumptions. While spectral clustering of vector inputs is straightforward, extensions to structured data or time-series data remain less explored. This paper proposes a clustering method for time-series data that couples non-parametric spectral clustering with parametric hidden Markov models (HMMs). HMMs add some beneficial structural and parametric assumptions such as Markov properties and hidden state variables which are useful for clustering. This article shows that using probabilistic pairwise kernel estimates between parametric models provides improved experimental results for unsupervised clustering and visualization of real and synthetic datasets. Results are compared with a fully parametric baseline method (a mixture of hidden Markov models) and a non-parametric baseline method (spectral clustering with non-parametric time-series kernels).

1 Introduction

This paper explores unsupervised learning in the time-series domain using a combination of parametric and non-parametric methods. Some parametric assumptions, such as Markov assumptions and hidden state assumptions, are quite useful for time-series data. However, it is also advantageous to remain non-parametric and agnostic about the overall shape that a collection of time-series data forms. This paper provides surprising empirical evidence that a semi-parametric [1,2] method can outperform both fully parametric methods of describing multiple time-series observations and fully non-parametric methods. These improvements include better clustering performance as well as better embedding and visualization over existing state-of-the-art time-series techniques.

There are a variety of parametric and non-parametric algorithms for discovering clusters within a dataset; however, the application of these techniques for clustering sequential data such as time-series data poses a number of additional challenges. Time-series data has inherent structure which may be disregarded by a fully non-parametric method. Additionally, a clustering approach for time-series data must be capable of detecting similar hidden properties or behavior between sequences of different lengths with no obvious alignment principle across temporal observations.

Alternatively, standard parametric clustering methods are popular but can make excessively strict assumptions about the overall distribution of a dataset of time-series samples. Expectation Maximization (EM), for example, estimates a fully parametric mixture model by iteratively adjusting the parameters to maximize likelihood [3,4,5]. In the time-series domain, a mixture of hidden Markov models (HMMs) can be used for clustering [5]. This is sensible since the underlying Markov process assumption is useful for a time series. However, the parametric *mixture* of HMMs may make invalid assumptions about the shape of the overall distribution of the collection of time-series exemplars. Just as a mixture of Gaussians assumes a radial shape for each cluster, many fully parametric time series models make assumptions about the shape of the variation across the many time series in a dataset. This is only sensible if data is organized into radial or symmetric clusters. In practice, though, clusters (of points or of time series) might actually be smoothly varying in a non-radially distributed manner.

Recent graph-theoretic approaches to clustering [6,7,8], on the other hand, do not make assumptions about the underlying distribution of the data. Data samples are treated as nodes in a weighted graph, where the edge weight between any two nodes is given by a similarity metric or kernel function. A k-way clustering is represented as a series of cuts which remove edges from the graph, dividing the graph into a set of k disjoint subgraphs. This approach clusters data even when the underlying parametric form is unknown (as long as there are enough samples) and far from radial or spherical. Unfortunately, recovering the optimal cuts is an NP-complete problem as was shown by Shi and Malik [8] who proposed the Normalized Cut (NCut) criterion. Spectral clustering is a relaxation of NCut into an linear system which uses eigenvectors of the graph Laplacian to cluster the data [7].

This article takes a semi-parametric approach to combine the complementary advantages of both methods. It applies recent work in spectral clustering [7] to the task of clustering time-series data. However, each time series is individually modeled using HMMs. This assumes HMM structure for each time-series datum on its own yet assumes no underlying structure in the overall distribution of the time-series data. The sequences are clustered only according to their individual pairwise proximity in HMM parameter space. It should be noted that though HMMs are used in this paper, the approach is applicable to clustering other time-series datasets under different parametric assumptions such as linear dynamical systems.

2 HMMs and Kernels

2.1 Hidden Markov Models

Assume a dataset of $n = 1 \dots N$ time-series sequences where each datum $\mathbf{x}_n$ is an ordered sequence of $t = 1, \dots, T_n$ vectors $x_{n,t}$ in some Euclidean space. A natural parametric model for representing a single time series or sequence $\mathbf{x}_n$ is the hidden Markov model (HMM), whose likelihood is denoted $p(\mathbf{x}_n|\theta_n)$. Note that the model θ_n is different for each sequence. This replaces the common *iid* (independent identically distributed) assumption on the dataset with a weaker *id* (independently distributed) assumption. More specifically, for $p(\mathbf{x}_n|\theta_n)$, consider a first-order stationary HMM with Gaussian emissions. The probability model of a sequence is $p(\mathbf{x}|\theta)$ where $\mathbf{x} = \{x_1, \ldots, x_T\}$ is a sequence of length T where each observation vector is $x_t \in \Re^d$. In addition, an HMM has a hidden state at each time point $\mathbf{q} = \{q_1, \ldots, q_T\}$ where each state takes on a discrete value $q_t = \{1, \ldots, M\}$. The likelihood of the HMM factorizes as follows:

$$p(\mathbf{x}|\theta) = \sum_{q_0,\ldots,q_T} p(x_0|q_0)p(q_0) \prod_{t=1}^{T} p(x_t|q_t)p(q_t|q_{t-1}) \tag{1}$$

The HMM is specified by the parameters: $\theta = (\pi, \alpha, \mu, \Sigma)$

1. The initial state probability distribution $\pi_i = p(q_0 = i)$, $i = 1 \ldots M$.
2. The state transition probability distribution given by a matrix $\alpha \in \Re^{M\times M}$ where $\alpha_{ij} = p(q_t = j|q_{t-1} = i)$.
3. The emission density $p(x_t|q_t = i) = \mathcal{N}(x_t|\mu_i, \Sigma_i)$, for $i = 1 \ldots M$, where $\mu_i \in \Re^d$ and $\Sigma_i \in \Re^{d\times d}$ are the mean and covariance of the Gaussian in state i. Take $\mu = \{\mu_1, \ldots, \mu_M\}$ and $\Sigma = \{\Sigma_1, \ldots, \Sigma_M\}$ for short.

Estimating the parameters of an HMM for a single sequence is typically done via EM. The E-step uses a forward-backward pass or junction tree algorithm (JTA) to obtain posterior marginals over the hidden states given the observations: $\gamma_t(i) = p(q_t = S_i|\mathbf{x}_n, \hat{\theta}_n)$ and $\xi_t(i,j) = p(q_t = S_i, q_{t+1} = S_j|\mathbf{x}_n, \hat{\theta}_n)$. The M-step updates the parameters θ using these E-step marginals as follows: $\hat{\pi}_i = \gamma_1(i)$

$$\hat{\alpha}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1}\sum_{j=1}^{M} \xi_t(i,j)} \quad \hat{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_t(i)x_t}{\sum_{t=1}^{T} \gamma_t(i)} \quad \hat{\Sigma}_i = \frac{\sum_{t=1}^{T} \gamma_t(i)(x_t - \mu_i)(x_t - \mu_i)^T}{\sum_{t=1}^{T} \gamma_t(i)}.$$

2.2 Probability Product Kernels

A natural choice of kernel between HMMs is the probability product kernel (PPK) described in [9] since it computes an affinity between distributions. The generalized inner product is found by integrating a product of the distributions of pairs of data sequences over the space of all potential observable sequences $\mathcal{X}$: $\mathcal{K}(p(\mathbf{x}|\theta), p(\mathbf{x}|\theta')) = \int p^\beta(\mathbf{x}|\theta)p^\beta(\mathbf{x}|\theta')d\mathbf{x}$. When $\beta = 1/2$, the PPK becomes the classic Bhattacharyya affinity metric between two probability distributions. The Bhattacharyya affinity is favored over other probabilistic divergences and affinities such as Kullback-Leibler (KL) divergence because it is symmetric and positive semi-definite (it is a Mercer kernel). In addition, it is computable in closed form for a variety of distributions including HMMs while the KL between two HMMs cannot be exactly recovered efficiently.

This section discusses the computation of the PPK between two HMMs $p(\mathbf{x}|\theta)$ and $p(\mathbf{x}|\theta')$. For brevity, we denote $p(\mathbf{x}|\theta)$ as p and $p'(\mathbf{x}|\theta')$ as p' where $\mathbf{x}$ represents a sequence of emissions x_t for $t = 1...T$. While the brute force evaluation

Table 1. The probability product kernel

Probability product kernel $\mathcal{K}(\theta, \theta')$:
$\mathcal{K}(\theta, \theta') = \sum_{q_T} \sum_{q'_T} \Psi(q_T, q'_T)$
$\prod_{t=1}^{T} \sum_{q_{t-1}} \sum_{q'_{t-1}} p(q_t|q_{t-1})^\beta p'(q'_t|q'_{t-1})^\beta \Psi(q_{t-1}, q'_{t-1}) p(q_0)^\beta p'(q'_0)^\beta$
Elementary kernel $\Psi(\cdot)$:
$\Psi(q_t = i, q'_t = j) = \int_{x_t} p(x_t|q_t = i)^\beta p'(x_t|q'_t = j)^\beta dx_t$

An efficient iterative method to calculate the PPK $\tilde{\mathcal{K}}(\theta, \theta')$:

```
$\Phi(q_0, q'_0) = p(q_0)^\beta p'(q'_0)^\beta$
for $t = 1 \ldots T$
    $\Phi(q_t, q'_t) = \sum_{q_{t-1}} \sum_{q'_{t-1}} p(q_t|q_{t-1})^\beta p(q'_t|q'_{t-1})^\beta \Psi(q_{t-1}, q'_{t-1}) \Phi(q_{t-1}, q'_{t-1})$
end
$\tilde{\mathcal{K}}(\theta, \theta') = \sum_{q_T} \sum_{q'_T} \Phi(q_T, q'_T) \Psi(q_T, q'_T)$
```

of the integral over $\mathbf{x}$ is expensive, an exact efficient formula is possible. Effectively, the kernel takes advantage of the factorization of the HMMs to set up an efficient iterative formula.

Initially, an elementary kernel $\Psi(\theta, \theta') = \int_{x_t} p^\beta(x_t|\theta) p^\beta(x_t|\theta') dx_t$ is computed; this is the Bhattacharyya affinity between the emissions models for p and p' integrated over the space of all emissions. For the exponential family of distributions this integral can be calculated in closed form. For HMMs with Gaussian emissions, this integral is proportional to:

$$\Psi(i, j) = \frac{|\Sigma^\dagger|^{1/2}}{|\Sigma_i|^{\beta/2} |\Sigma_j|^{\beta/2} \exp(-\frac{\beta}{2}(\mu_i^T \Sigma_i^{-1} \mu_i + \mu_j{}^T \Sigma_j{}^{-1} \mu_j - \mu^{\dagger T} \Sigma^\dagger \mu^\dagger)}$$

where $\Sigma^\dagger = (\Sigma_i^{-1} + \Sigma_j{}^{-1})^{-1}$ and $\mu^\dagger = \Sigma_i^{-1} \mu_i + \Sigma_j{}^{-1} \mu_j$. Given the elementary kernel, the kernel between two HMMs is solved in $\mathcal{O}(TM^2)$ operations using the formula in Table 1 (further details can be found in [9]). Given a kernel affinity between two HMMs, a non-parametric relationship between time series sequences emerges. It is now straightforward to apply non-parametric clustering and embedding methods which will be described in section 4. The next section, however, first describes a more typical fully parametric clustering setup using a mixture of HMM models to couple the sequences and parameters θ_n. This has the undesirable effect of coupling pairs of sequences by making global parametric assumptions on the whole dataset instead of only on pairs of sequences.

3 Clustering as a Mixture of HMMs

Parametric approaches to clustering of time-series data using HMMs assume that each observation sequence $\mathbf{x}_n$ is generated from a mixture of K components and use different clustering formulations in order to estimate this mixture. Two such techniques for estimating a mixture of K HMMs are the hard-clustering k-means approach and the soft-clustering EM approach.

A k-means approach is used in [3,4] to assign sequences to clusters in each iteration and use only the sequences assigned to a cluster for re-estimation of its HMM parameters. Each sequence can only be assigned to *one* cluster per iteration and it can run into problems when there is no good separation between the processes that generated the data approximated by the HMM parameters. A soft-clustering approach that overcomes these problems is described in [5]; here each sequence has a prior probability $p(z = k)$ of being generated by the k'th HMM. This reduces to a search for the set of parameters $\{\theta_1, \ldots, \theta_K, p(z)\}$ where $z \in \{1, \ldots K\}$ that maximize the likelihood function $\prod_{n=1}^{N} \sum_{k=1}^{K} p(z = k) p(\mathbf{x}_n|\theta_k)$. The E-step is similar to the E-step in the HMM-training algorithm run separately for each of the K HMMs. The posterior likelihood $\tau_n^k = \frac{p(z=k)p(\mathbf{x}_n|\theta_k)}{\sum_{k=1}^{K} p(z=k)p(\mathbf{x}_n|\theta_k)}$ is estimated as the probability that sequence $\mathbf{x}_n$ was generated by HMM k. The M-step incorporates the posterior τ_n^k of each sequence n into its contribution to the updated parameters for the kth HMM.

$$\hat{\alpha}_{ij} = \frac{\sum_{n=1}^{N} \tau_n^k \sum_{t=1}^{T_n-1} \xi_{n,t}^k(i,j)}{\sum_{n=1}^{N} \tau_n^k \sum_{t=1}^{T_n-1} \sum_{j=1}^{M} \xi_{n,t}^k(i,j)} \qquad \hat{\mu}_i = \frac{\sum_{n=1}^{N} \tau_n^k \sum_{t=1}^{T_n} \gamma_{n,t}^k(i) x_{n,t}}{\sum_{n=1}^{N} \tau_n^k \sum_{t=1}^{T_n} \gamma_{n,t}^k(i)}$$

$$\hat{\pi}_i = \frac{\sum_{n=1}^{N} \tau_n^k \gamma_{n,1}^k(i)}{N} \qquad \hat{\Sigma}_i = \frac{\sum_{n=1}^{N} \tau_n^k \sum_{t=1}^{T_n} \gamma_{n,t}^k(i)(x_{n,t} - \mu_i)(x_{n,t} - \mu_i)^T}{\sum_{n=1}^{N} \tau_n^k \sum_{t=1}^{T_n} \gamma_{n,t}^k(i)}.$$

The responsibility terms τ_n^k also provide the priors $p(z = k)$ for the next iteration and determine the final clustering assignment at convergence.

In summary, this method makes strict parametric assumptions about the underlying distribution of all the sequences in the dataset and attempts to find a parameter setting that maximizes the posterior probabilities given such models for the underlying distributions. However, these parametric assumptions aren't always intuitive and, as our experiments show, often negatively affect clustering performance as opposed to the non-parametric methods that are being investigated in this article.

4 Spectral Clustering of HMMs

The spectral approach to HMM clustering involves estimating an HMM model for each sequence using the approach outlined in section 2.1. The PPK is then computed between all pairs of HMMs to generate a Gram matrix which is used for spectral clustering. This approach leverages both parametric and non-parametric techniques in the clustering process; parametric HMMs make some assumptions about the structure of the individual sequences (such as Markov assumptions) but the spectral clustering approach makes no assumptions about the overall distribution of the sequences (for instance, *i.i.d* assumptions). Empirically, this approach (Spectral Clustering of Probability Product Kernels or SC-PPK) achieves a noticeable improvement in clustering accuracy over fully parametric models such as mixtures of HMMs or naive pairwise likelihood comparisons.

Listed below are the steps of our proposed algorithm. It is a time-series analogue of the Ng-Weiss algorithm presented in [7].

The SC-PPK algorithm

1. Fit an HMM to each of the $n = 1 \dots N$ time-series sequences to retrieve models $\theta_1 ... \theta_N$.
2. Calculate the Gram matrix $A \in \mathbb{R}^{N \times N}$ where $A_{m,n} = \mathcal{K}(\theta_m, \theta_n)$ for all pairs of models using the probability product kernel (default setting: $\beta = 1/2$, T=10).
3. Define $D \in \mathbb{R}^{N \times N}$ to be the diagonal matrix where $D_{m,m} = \sum_n A_{m,n}$ and construct the Laplacian matrix: $L = D^{-1/2} A D^{-1/2}$.
4. Find the K largest eigenvectors of L and form matrix $X \in \mathbb{R}^{N \times K}$ by stacking the eigenvectors in columns. Renormalize the rows of matrix X to have unit length.
5. Cluster the N rows of X into K clusters via k-means or any other algorithm that attempts to minimize distortion.
6. The cluster labels for the N rows are used to label the corresponding N HMM models.

Another well-known approach to clustering time-series data is Dynamic-Time-Warping, introduced in [3]. This method was surpassed in performance by the spectral clustering method proposed by Yin and Yang [10] which uses a direct comparison of HMM likelihoods as the kernel affinity. The SC-PPK method outperforms Yin and Yang's method due to the fact that it doesn't calculate the affinities based on a pair of time-series samples but integrates over the entire space of all possible samples given the HMM models. Thus, the SC-PPK approach recovers a stronger and more representative affinity score between HMM models. Furthermore, since the PPK computes the integration by solving a closed form kernel using an efficient iterative method, it achieves a significant gain in speed over the kernel used in [10].

5 Experiments

This section details the experiments that were conducted to compare semi-parametric spectral approaches and fully parametric approaches to clustering of time-series data. k-Means and EM versions of a mixture of HMMs approach were used to represent the parametric setting. The two spectral clustering algorithms investigated were Yin and Yang's algorithm[10], which computes a likelihood-based kernel between pairs of sequences, and the SC-PPK algorithm which computes a kernel over the HMM model parameters. Note that there are parameters which can be adjusted for both of the spectral clustering methods: the σ fall-off ratio for the Yin-Yang kernel and the mixing proportion T for the SC-PPK method. In the following experiments, the default settings were used for both methods, i.e. $\sigma = 1$ and $T = 10$. Stability results for these kernels are shown in Fig 1.

5.1 Datasets

The evaluations were run over a variety of real-world and synthesized datasets which are described below.

MOCAP: The Motion Capture dataset (available from Carnegie Mellon University[1]) consists of time-series data representing human locomotion and actions. The sequences consist of 123-dimensional vectors representing 41 body markers tracked spatially through time for various physical activities. For these experiments, simple activities that were likely to be hard to cluster were considere; tests compared either similar activities or the same activity for different subjects. Table 2 contains the results of the evaluation over these real-world datasets.
Rotated MOCAP: A synthesized dataset was generated by rotating two MOCAP sequences 5° at a time through 360 degrees. The seed sequences used were a walking sequence from WALK(#7) and a running sequence from RUN(#9). This dataset, which provides us with clusters of sequences that lie on a regular manifold, is used for comparing clustering in Table 3, comparing running times in Table 6 and embedding in Fig 2. An additional dataset SWERVING was generated by rotating the walking and running sequence left and right periodically to make their movements seem zig-zagged.
Arabic handwriting: This dataset consists of 2-dimensional time-series sequences which represent written word-parts of Arabic characters extracted from the dataset used in [11]. Table 4 shows the results of clustering pairs of similar word-parts (identified by their Unicode characters) and Fig 3(a) shows an MDS embedding of three symbols.
Australian sign language: This dataset (from the University of California-Irvine[2]) consists of multiple sign-language gestures, each represented by 27 instances of 22-dimensional time-series sequences. Semantically-related expressions such as *write* and *draw* or antonyms such as *give* and *take* were assumed to have similar real-world symbols and formed the basis of all experiments with this dataset. Table 5 shows the average accuracy over the clustering of 18 such pairs of sequences while varying the number of HMM states used and Fig 3(b) shows an embedding of three gestures.

5.2 Results

The tables in this section show the results of experiments over these various datasets. Experiments were restricted to pairwise clustering over datasets of similar size. The standard accuracy metric is used for comparison of results. The results reported were averaged over multiple folds - five folds for the spectral clustering algorithms and ten folds for the fully parametric techniques since these algorithms converge to local maxima and have highly variable performance).

In general, the kernel-based methods outperformed the parametric methods and the PPK performed favorably compared to Yin and Yang's kernel.

[1] http://mocap.cs.cmu.edu/
[2] http://www.cse.unsw.edu.au/w̃aleed/tml/data

Table 2. Clustering accuracy with 2-state HMMs on MOCAP data. The numbers in the parentheses identify the index of the subject performing that particular specified action.

Dataset	k-Means	EM	Yin-Yang	SC-PPK
simple walk vs run set	**100%**	**100%**	**100%**	**100%**
run(#9) vs run/jog(#35)	57%	52%	76%	**100%**
walk(#7) vs walk(#8)	59%	60%	**68%**	**68%**
walk(#7) vs run/jog(#35)	71%	69%	71%	**95%**
jump(#13) vs jump forward(#13)	50%	50%	75%	**87%**
jump(#13,#16) vs jump forward(#13,#16)	50%	50%	60%	**66%**

Table 3. Clustering accuracy with 2-state HMMs on synthesized MOCAP dataset. A single pair of walking and running time-series samples were used, subsequent samples were generated by rotating the seed pair 5° at a time to generate 72 unique pairs.

rotation limit	step size	k-Means	EM	Yin-Yang	SC-PPK
30°	5°	**100%**	**100%**	92%	**100%**
60°	5°	54%	71%	63%	**100%**
90°	5°	50%	50%	75%	**100%**
180°	5°	51%	51%	71%	**100%**
360°	5°	50%	50%	60%	**100%**
360°	10°	50%	50%	50%	**100%**
360°	15°	50%	50%	50%	**100%**
360°	30°	54%	50%	50%	**100%**
Swerving	–	50%	50%	85%	**100%**

Table 4. Clustering accuracy with 2-state HMMs on Arabic handwriting dataset

Dataset	k-Means	EM	YY	SC-PPK
U0641 vs U0643	68%	64%	86%	**97%**
U0645 vs U0647	70%	66%	86%	**100%**
U062D vs U062F	78%	80%	93%	**95%**
U0621 vs U062D	66%	65%	86%	**93%**
U0628 vs U0631	71%	70%	94%	**100%**
U0635 vs U0644	74%	76%	95%	**100%**
U0621 vs U0647	71%	66%	96%	**98%**

Improvements in accuracy well as better runtime performance were seen in both quantitative clustering accuracy and qualitative embedding performance. In addition, experiments were conducted to investigate the stability of the SC-PPK method over the T parameter. Fig 1 shows stability comparisons for both spectral clustering kernels over the respective parameters. The accuracies were averaged over five-fold testing. It was noted that a useful setting for T usually lay within the interval of 5 to 25. In practice, cross validation would be useful for recovering the optimal setting.

Table 5. Clustering accuracy on Australian sign language dataset; 18 semantically related pairs of signs compared. Top: 5 representative pairs are shown for a range of SC-PPK clustering accuracies. Bottom: averages for the 18 pairs over different number of states.

Sample pairs (2-state)	k-Means	EM	Yin-Yang	SC-PPK
'hot' vs 'cold'	64%	98%	96%	**100%**
'eat' vs 'drink'	52%	50%	52%	**93%**
'happy' vs 'sad'	73%	54%	50%	**87%**
'spend' vs 'cost'	58%	53%	52%	**80%**
'yes' vs 'no'	51%	51%	52%	**59%**
# of hidden states	k-Means	EM	Yin-Yang	SC-PPK
2	64.5%	72.1%	69.3%	**76.1%**
3	64.4%	73.3%	**75.5%**	**75.5%**
4	65.4%	**74.9%**	74.7%	74.3%

Table 6. Average runtimes in seconds on a 3-Ghz machine with emissions $x_t \in \mathbb{R}^{123\times 1}$, includes HMM training time

# of time-series samples	k-Means	EM	Yin-Yang	SC-PPK
5	20.6s	23.5s	4.1s	**3.6s**
10	47.1s	59.7s	11.1s	**6.1s**
25	68.29s	185.0s	49.9s	**15.4s**
50	111.6s	212.9s	171.8s	**30.1s**
75	178.3s	455.6s	382.8s	**48.6s**
100	295.9s	723.1s	650.2s	**64.5s**

5.3 Runtime Advantages

Unlike EM-HMM, which needs to calculate posteriors over $N \times k$ HMM-sequence pairs and maximize over k HMMs at *every* iteration until convergence, SC-PPK requires a single HMM to be trained once for each sequence in the dataset. The SC-PPK method calculates the integration between two HMM parameters in closed form directly without needing to evaluate the likelihood of the individual time-series samples, resulting in a dramatic reduction of in total runtime. In practice, we noticed around two orders of magnitude improvement in clustering speed over EM, as shown in Table 6. These runtimes include HMM training times as well as clustering times.

6 Visualization of HMM Parameters

Visualization and manifold learning is another important component of unsupervised learning. Starting from a set of high dimensional data points $\mathcal{X}$ with $x_i \in \mathbb{R}^N$, embedding and visualization methods recover a set of corresponding low dimensional datapoints $\mathcal{Y}$ (typically with typically $y_i \in \mathbb{R}^2$) such that the

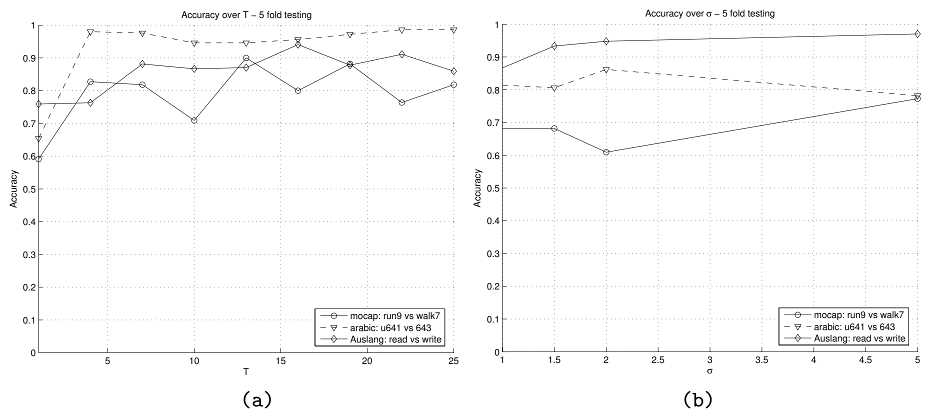

Fig. 1. Kernel stability over their respective parameters (a) SC-PPK - T (b) YY kernel - σ. Accuracies averaged over five runs.

distance between y_i and y_j is similar to the distance between x_i and x_j for all $i, j = 1...N$. These techniques permit visualization of high dimensional datasets and can help confirm clustering patterns. Classic visualization and embedding techniques include multi-dimensional scaling (MDS) [12] as well as recent contenders such as semi-definite embedding (SDE) [13].

To further analyze the usefulness of the PPK at capturing an accurate representation of the kernel affinity between two sequences, embedding using MDS was applied to the datasets. This corresponds to training an HMM over each of the time-series sequences and recovering the $\mathbb{R}^{N\times N}$ Gram matrix where $A_{m,n} = \mathcal{K}(\theta_m, \theta_n)$ – identical to steps 1 and 2 of the SC-PPK algorithm. From the Gram matrix, the dissimilarity matrix $D \in \mathbb{R}^{N\times N}$ is given by $D_{i,j} = 1/A_{i,j}$. This matrix is then used as the input for the standard MDS algorithm as in [12]. MDS is chosen because of its simplicity although more sophisticated methods such as SDE are equally viable. Fig 2 shows the embedding for the rotated data using a rotation step size of 10° under the Yin and Yang kernel (a) and the PPK (b). From the figure, we can see that the Yin and Yang kernel captures some of the periodic structure of the dataset in the embedding but it is only locally useful and does not adequately capture the expected global circular structure in the rotation. Conversely, the result from the PPK method is much clearer. The PPK integrates over the sample space providing a less brittle description of each time series. Thus the kernel affinity captures an accurate representation of the the distance between HMM parameters with respect to all of the data samples, forming a perfectly circular global embedding of the 360° rotated MOCAP dataset. Fig 3(a) shows PPK-based embeddings for three classes from the Arabic handwriting dataset. The method recovered an accurated embedding as the three classes are separated from one-another. Similarly, Fig 3(b) shows the embeddings for three classes from the Australian sign language dataset.

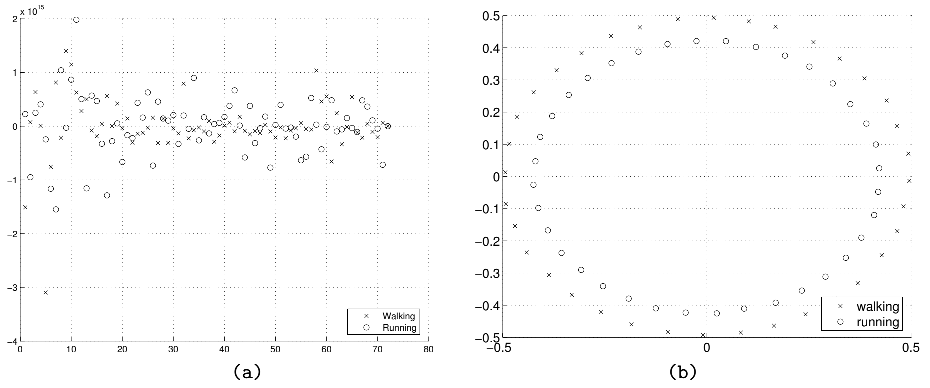

Fig. 2. MDS embedding of HMM parameters using the synthesized 360° rotated MOCAP dataset with (a) the Yin-Yang kernel and (b) the probability product kernel

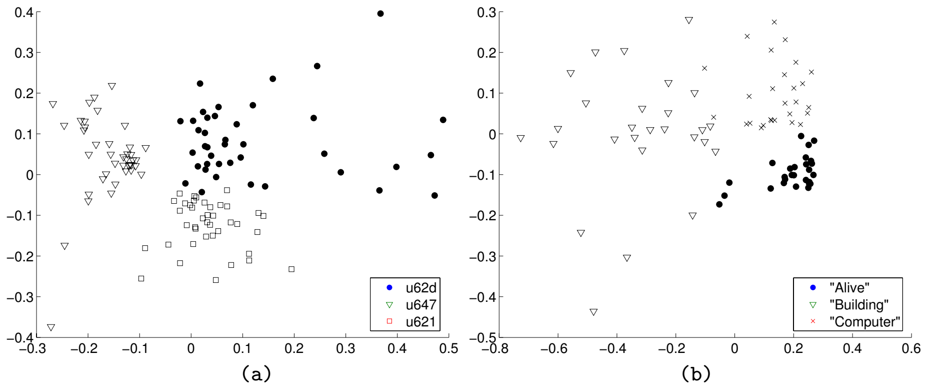

Fig. 3. MDS embedding of HMM parameters (a) Arabic handwriting dataset and (b) Australian sign Language dataset

7 Conclusions

This paper presented a semi-parametric approach for clustering time-series data that exploits some parametric knowledge about the data including its Markov properties and the presence of latent states, and at the same time utilizes non-parametric principles and remains agnostic about the shape of the clusters a multitude of time series can form. This works surprisingly well for time-series data in experiments on real-world data. By avoiding parametric assumptions about the underlying distributions or the variations across the time series, improved clustering accuracy is possible.

The method combines both a non-parametric spectral clustering approach using the probability product kernel with a fully parametric maximum likelihood estimation approach for each singleton time series. We showed that spectral clustering with a probability product kernel method provides improvements in clustering accuracy over fully parametric mixture modeling as well as spectral clustering with non-probabilistic and non-parametric affinity measures. Furthermore, embedding and visualization of time series data is also improved. Finally, the proposed method has computational efficiency benefits over prior approaches.

For future work, we are investigating spectral clustering with the probability product kernel for other generalized graphical models and parametric distributions. In addition, we are investigating a general formalism and generalized cost functions for semi-parametric estimation that unify both parametric and non-parametric criteria and leverage the complementary advantages of both approaches.

References

1. Dey, D.: Practical Nonparametric & Semiparametric Bayesian Statistics, vol. 133. Springer, Heidelberg (1998)
2. Hoti, F., Holstrom, L.: A semiparametric density estimation approach to pattern classification. Pattern Recognition (2004)
3. Tim Oates, L.F., Cohen, P.R.: Clustering time series with hidden Markov models and dynamic time warping. In: IJCAI-99 Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning (1998)
4. Smyth, P.: Clustering sequences with hidden Markov models. In: Advances in Neural Information Processing Systems, pp. 648–654 (1997)
5. Alon, J., Sclaroff, S., Kollios, G., Pavlovic, V.: Discovering clusters in motion time-series data. IEEE Computer Vision and Pattern Recognition (2003)
6. Shental, N., Zomet, A., Hertz, T., Weiss, Y.: Pairwise clustering and graphical models. In: Advances in Neural Information Processing Systems (2003)
7. Ng, A., Jordan, M., Weiss, Y.: On Spectral Clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems (2001)
8. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)
9. Jebara, T., Kondor, R., Howard, A.: Probability product kernels. Journal of Machine Learning Research 5, 819–844 (2004)
10. Yin, J., Yang, Q.: Integrating hidden Markov models and spectral analysis for sensory timeseries clustering. In: Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05), IEEE Computer Society Press, Los Alamitos (2005)
11. Biadsy, F., El-Sana, J., Habash, N.: Online arabic handwriting recognition using hidden Markov models. In: The 10th International Workshop on Frontiers of Handwriting Recognition (2006)
12. Kruskal, J.B., Wish, M.: Multidimensional Scaling. In: Quantitative Application in the Social Sciences, Sage University Paper (1978)
13. Weinberger, K.Q., Saul, L.K.: Unsupervised learning of image manifolds by semidefinite programming. International Journal of Computer Vision 70(1), 77–90 (2006)