Maximum Margin Temporal Clustering

Minh Hoai **Fernando De la Torre**

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract

Temporal Clustering (TC) refers to the factorization of multiple time series into a set of non-overlapping segments that belong to k temporal clusters. Existing methods based on extensions of generative models such as k-means or Switching Linear Dynamical Systems (SLDS) often lead to intractable inference and lack a mechanism for feature selection, critical when dealing with high dimensional data. To overcome these limitations, this paper proposes Maximum Margin Temporal Clustering (MMTC). MMTC simultaneously determines the start and the end of each segment, while learning a multi-class Support Vector Machine (SVM) to discriminate among temporal clusters. MMTC extends Maximum Margin Clustering in two ways: first, it incorporates the notion of TC, and second, it introduces additional constraints to achieve better balance between clusters. Experiments on clustering human actions and bee dancing motions illustrate the benefits of our approach compared to state-of-the-art methods.

1 Introduction

Time series data are prevalent in every field from physics and robotics to finance and biology. Factorizing time series into temporally coherent segments is often useful in its own right as a self-exploratory technique or as a subroutine in more complex data-mining algorithms. In particular, Temporal Clustering (TC) has been applied to learning taxonomies of facial behavior (Zhou et al., 2010), speaker diarization (Fox et al., 2009), discovering motion primitives (Guerra-Filho and Aloimonos, 2006; Vecchio et al., 2003; Zhou et al., 2008), and clustering human actions in video (Turaga et al., 2009).

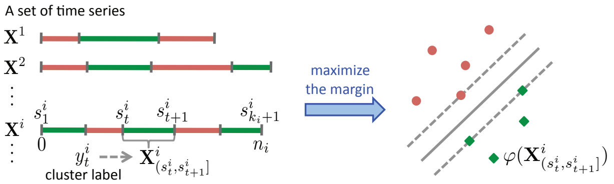

Figure 1: Temporal clustering: time series are partitioned into segments (by finding a set of change points $s_1^i, \cdots, s_{k_i+1}^i$) and similar segments are grouped into classes (i.e., assigning cluster labels $y_1^i, \cdots, y_{k_i}^i$ to the segments). The objective is to maximize the margin for the separation between clusters. Though this figure only illustrates the case of two classes, our method is multi-class.

It is important to notice that the TC problem addressed in this paper is different from clustering time series. Clustering time series, e.g., Liao (2005), refers to the problem of grouping time series that have been pre-segmented. TC refers to the factorization of multiple time series into a set of non-overlapping segments that belong to one of k temporal clusters (see Fig. 1). Also, recall that TC is an unsupervised problem, and it is different from the supervised and weakly-supervised temporal segmentation problems (e.g., Hoai et al. (2011); Nguyen et al. (2009); Oh et al. (2008); Shi et al. (2008)). Models such as segmental SLDS (Oh et al., 2008) or semi-Markov model (Shi et al., 2008) are trained in a supervised manner using manually annotated data. This paper explores an unsupervised approach to temporal segmentation and clustering.

TC is a relatively unexplored problem. Few unsupervised approaches exist, and all of them are based on generative models such as extensions of Dynamic Bayesian Networks (DBNs) (Fox et al., 2009), k-means (Robards and Sunehag, 2009) or combining spectral clustering methods with Dynamic Time

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX. Copyright 2012 by the authors.

Warping (Zhou et al., 2010, 2008). Existing TC algorithms have several issues. First, algorithm based on k-means clustering are only optimal for spherical clusters. Second, generative approaches lack a mechanism for feature selection, that is specially critical when clustering high-dimensional noisy data. Third, variations of switching DBNs typically lead to intractable inference.

To partially address the aforementioned problems, this paper proposes Maximum Margin Temporal Clustering (MMTC), a novel learning framework that simultaneously performs temporal segmentation and learns a multi-class SVM for separating temporal clusters. MMTC is based on Maximum Margin Clustering (MMC) (Xu et al., 2004) and extends it to cluster time series segments. Fig. 1 illustrates the key idea of our method: divide each time series into a set of disjoint segments such that each segment belongs to a cluster. MMTC maximizes the cluster separability using the SVM score as the measure of separability. We demonstrate our approach on several publicly available datasets and show that our unsupervised method consistently matches and often surpasses the performance of state-of-the-art methods for TC.

2 Related Work

This section describes related work in MMC, TC, and temporal segmentation.

MMC (Xu et al., 2004) extends the theory of SVMs to unsupervised learning, and it has shown promising results. Since its introduction, it has been extended in many ways, for instance, changing the loss function (e.g., Gieseke et al. (2009)), incorporating additional constraints such as pairwise links (Hu et al., 2008) and manifold smoothness (Wang et al., 2009), or adding a feature weighting mechanism (Zhao et al., 2009). Though many extensions of MMC have been proposed, none can be directly applied to the task of TC. The most relevant work that used MMC for time series segmentation was proposed by Estevan et al. (2007). They determined the boundaries between phonemes in a speech signal by examining the cluster assignment provided by applying MMC on the frames inside a short-time window. However, their only goal was to detect the boundaries of speech signals while we seek both segmentation and a discriminative model for each temporal cluster.

The literature on segmentation and clustering of time series falls in several categories. A popular strategy is to use DBNs such as SLDS (Fox et al., 2009; Oh et al., 2008). This approach is generative, often requires labeled training data, and typically leads to intractable inference. Some recent works (Do and Artières, 2009; Sha and Saul, 2007) explore the combination of large margin training and HMMs for the sequence labeling problem, but they assume that the temporal segmentation is provided. Change point detection such as Xuan and Murphy (2007) and Harchaoui et al. (2009) is another popular time series segmentation technique; it works by performing a sequence of change-point analysis in a sliding window along the time dimension. This, unlike our proposed method, only detects local boundaries and does not provide a global model for temporal events. Time series segmentation has also been implicitly studied from the perspective of analyzing periodicity of cyclic events, e.g., Cutler and Davis (2000); Pogalin et al. (2008). Cyclic motion analysis, however, only extracts segments of repetitive motion; consequently a substantial portion of a signal might neither be segmented nor modeled. Segmentation in video can be done by clustering frames and grouping those that are assigned to the same cluster to form a segment, as in Zelnik-Manor and Irani (2006). This approach performs segmentation as a subsequent step of clustering; it lacks a mechanism to incorporate the dynamics of temporal events in the clustering process. There has been substantial amount of work on subsequence time series clustering (see Keogh and Lin (2005) for a survey). Algorithms in this category, e.g. Robards and Sunehag (2009), often use k-means for clustering because of programming convenience. However, k-means clustering has several drawbacks; it is only optimal for spherical clusters and lacks a mechanism for feature selection. To partially solve this problem Zhou et al. (2010, 2008) proposed Aligned Cluster Analysis, an extension of spectral clustering method to cluster time series. ACA combines spectral clustering and dynamic time warping in a principal manner, and it provides a natural embedding method for time series. However, ACA does not provide a feature weighting mechanism. In contrast, we propose to incorporate discriminative clustering, which has been shown to have advantages over generative models (Bach and Harchaoui, 2009; De la Torre and Kanade, 2006; Zhao et al., 2008), into the TC problem.

3 MMC Revisited

This section reviews the formulation of MMC Zhao et al. (2008) and points out a major limitation of cluster degeneration. To address this issue, we propose to replace the current balancing constraint by another that better regulates cluster sizes.

3.1 Multi-class MMC

MMC (Xu et al., 2004) is a discriminative clustering algorithm that seeks a binary partition of the data to maximize the classification margin of SVMs. Xu and Schuurmans (2005); Zhao et al. (2008) further extended MMC for the multi-class case. Given a set of data points $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^d$, multi-class MMC simultaneously finds the maximum margin hyperplanes $\mathbf{w}_1, \cdots, \mathbf{w}_m \in \mathbb{R}^d$ and the best cluster labels $y_1, \cdots, y_n \in \{1, \cdots, m\}$ by optimizing:

$$\underset{\mathbf{w}_j, y_i, \xi_i \geq 0}{\text{minimize}} \ \frac{1}{2m} \sum_{j=1}^{m} ||\mathbf{w}_j||^2 + C \sum_{i=1}^{n} \xi_i, \tag{1}$$

$$\text{s.t. } \forall i : \mathbf{w}_{y_i}^T \mathbf{x}_i - \mathbf{w}_y^T \mathbf{x}_i \geq 1 - \xi_i \ \forall y \neq y_i, \tag{2}$$

$$\forall j, j' : -\lambda \leq (\mathbf{w}_j - \mathbf{w}_{j'})^T \sum_{i=1}^{n} \mathbf{x}_i \leq \lambda. \tag{3}$$

Here $\mathbf{w}_y^T \mathbf{x}_i$ is the confidence score for assigning data point $\mathbf{x}_i$ to cluster y. Constraint (2) requires $\mathbf{x}_i$ to belong to cluster y_i with relatively high confidence, higher than that of any other cluster by a margin. $\{\xi_i\}$ are slack variables which allow for penalized constraint violation, and C is the parameter controlling the trade-off between having a larger margin and having less constraint violation. Constraint (3) is added aiming to attain the balance between clusters.

The above MMC formulation has an inherent problem of a discriminative clustering method which is cluster degeneration, i.e., many clusters are empty. MMC requires every pair of clusters to be well separated by a margin. Thus every pair of clusters leads to a constraint on the maximum size of the margin. As a result, MMC favors a model with fewer number of clusters because less effort for separation is required. In the extreme case, MMC would create a single cluster if Constraint (3) is not used, and therefore Constraint (3) is added to balance the cluster sizes. Here λ is a tunable parameter of the balancing constraint. In practice, however, it only works well if the allowable number of clusters is two, $m = 2$. For $m > 2$, cluster degeneration still occurs often. Furthermore, Constraint (3) is not translation invariant. If the data is centralized at the origin, i.e., $\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i = \mathbf{0}$, the constraint has no effect and becomes redundant. In the next subsection we propose a modification to the MMC formulation to address this issue.

3.2 Membership Requirement MMC

This section proposes Membership Requirement Maximum Margin Clustering (MRMMC), a modification to the MMC formulation to address the issue of cluster degeneration:

$$\underset{\substack{\mathbf{w}_j, y_i \\ \xi_i \geq 0, \beta_j \geq 0}}{\text{minimize}} \ \frac{1}{2m} \sum_{j=1}^{m} ||\mathbf{w}_j||^2 + C \sum_{i=1}^{n} \xi_i + C_2 \sum_{j=1}^{m} \beta_j, \tag{4}$$

$$\text{s.t. } \forall i : \mathbf{w}_{y_i}^T \mathbf{x}_i - \mathbf{w}_y^T \mathbf{x}_i \geq 1 - \xi_i \ \forall y \neq y_i, \tag{5}$$

$$\forall j : \exists\ l \text{ different indexes } i\text{‘}s :$$

$$\mathbf{w}_j^T \mathbf{x}_i - \mathbf{w}_{j'}^T \mathbf{x}_i \geq 1 - \beta_j \ \forall j' \neq j. \tag{6}$$

The difference between MRMMC and the original MMC formulation lies at Constraint (6). In the essence, this is a soft constraint for requiring each cluster to have at least l members; β_j's are slack variables that allow for penalized constraint violation. This new formulation has several advantages over the original one, as will be shown in the experimental section.

We propose to optimize the above problem using block coordinate descent, which alternates between two steps: i) fixing $\{\mathbf{w}_j\}$, optimizes Eq. 4 over $\{y_i\}$, $\{\xi_i\}$, $\{\beta_j\}$, and the l members $\mathbf{x}_i$'s for each cluster j; ii) fixing $\{y_i\}$ and the l members $\mathbf{x}_i$'s for each cluster j, optimizes Eq. 4 over $\{\mathbf{w}_j\}$, $\{\xi_i\}$, and $\{\beta_j\}$. This optimization algorithm is simple to implement and monotonically decreases the energy and converges to a critical point. It is effective when combining with multiple restarts, as will be shown in the experiment section.

4 Maximum Margin Temporal Clustering

This section describes MMTC, a maximum margin approach for unsupervised time series clustering.

4.1 Joint Segmentation and Clustering

Given a collection of time series $\mathbf{X}^1, \cdots, \mathbf{X}^n$, MMTC divides each time series into a set of disjoint segments that maximizes the multi-class SVM margin. In other words, MMTC divides time series $\mathbf{X}^i \in \mathbb{R}^{d \times n_i}$ into k_i segments by finding a set of change points $s_1^i < \cdots < s_{k_i+1}^i$ (as shown in Fig. 1) and the associated cluster labels $y_1^i, \cdots, y_{k_i}^i \in \{1, \cdots, m\}$ that lead to maximum cluster separation:

$$\underset{\substack{\mathbf{w}_j, k_i, s_t^i, y_t^i \\ \xi_t^i \geq 0, \beta_j \geq 0}}{\text{minimize}} \ \frac{1}{2m} \sum_{j=1}^{m} ||\mathbf{w}_j||^2 + C \sum_{i=1}^{n} \sum_{t=1}^{k_i} \xi_t^i + C_2 \sum_{j=1}^{m} \beta_j, \tag{7}$$

$$\text{s.t. } \forall i, t : s_{t+1}^i - s_t^i \leq l_{max}, s_1^i = 0, s_{k_i+1}^i = n_i, \tag{8}$$

$$\forall i, t : (\mathbf{w}_{y_t^i} - \mathbf{w}_y)^T \varphi(\mathbf{X}^i_{(s_t^i, s_{t+1}^i]}) \geq 1 - \xi_t^i \ \forall y \neq y_t^i, \tag{9}$$

$$\forall j : \exists\ l \text{ segments, i.e., index pairs } (i, t) :$$

$$(\mathbf{w}_j^T - \mathbf{w}_{j'}^T) \varphi(\mathbf{X}^i_{(s_t^i, s_{t+1}^i]}) \geq 1 - \beta_j \ \forall j' \neq j. \tag{10}$$

Here $\mathbf{X}^i_{(s_t^i, s_{t+1}^i]}$ denotes the segment of time series $\mathbf{X}^i$ taken from frame $s_t^i + 1$ to frame s_{t+1}^i inclusive.

$\varphi(\cdot)$ denotes the feature computation function, and $\mathbf{w}_y^T \varphi(\mathbf{X}^i_{(s^i_t, s^i_{t+1}]})$ is the confidence score for assigning segment $\mathbf{X}^i_{(s^i_t, s^i_{t+1}]}$ to cluster y. Constraint (8) requires each time series to be divided into segments with the lengths are bounded by l_{max}, an application specific value. Constraint (9) requires segment $\mathbf{X}^i_{(s^i_t, s^i_{t+1}]}$ to belong to cluster y^i_t with relatively high confidence, higher than that of any other cluster by a margin. $\{\xi^i_t\}$ are slack variables which allow for penalized constraint violation, and C is the parameter controlling the trade-off between a larger margin and less constraint violation. Constraint (10) requires each cluster to have at least l members; this is also a soft constraint as slack variables $\{\beta_j\}$ are used.

Following Altun et al. (2003), we consider an additive feature mapping:

$$\varphi(\mathbf{X}^i_{(s^i_t, s^i_{t+1}]}) = \sum_{p=s^i_t+1}^{s^i_{t+1}} \varphi(\mathbf{X}^i_p). \tag{11}$$

This type of segment-level feature mappings subsumes both HMM and Bag-of-Words approaches. Given Eq. (11), the left hand side of Constraint (10) is:

$$\begin{aligned}(\mathbf{w}_j^T - \mathbf{w}_{j'}^T)\varphi(\mathbf{X}^i_{(s^i_t, s^i_{t+1}]}) &= \\ (\mathbf{w}_j^T - \mathbf{w}_{j'}^T) &\underset{p\in(s^i_t, s^i_{t+1}]}{\text{mean}} \{\varphi(\mathbf{X}^i_p)\}(s^i_{t+1} - s^i_t).\end{aligned} \tag{12}$$

For tractable segmentation and labeling inference during the learning stage, we approximate the mean of $\{\varphi(\mathbf{X}^i_p)\}$ by a particular instance $\varphi(\mathbf{X}^i_q)$ and $s^i_{t+1} - s^i_t$ by $l_{max}/2$. This approximation is only necessary in the learning stage, in which the balancing constraint, Constraint (10), is enforced. Constraint (10) is then approximated by:

$$\begin{aligned}&\forall j : \exists\ l' \text{ index pairs } (i,q) : \\ &\quad (\mathbf{w}_j^T - \mathbf{w}_{j'}^T)\varphi(\mathbf{X}^i_q)\frac{l_{max}}{2} \geq 1 - \beta_j\ \forall j' \neq j.\end{aligned} \tag{13}$$

Roughly speaking, Constraint (10) requires each cluster to have at least l segments, while Constraint (13) requires each cluster to have at least l' frames, with $l' = \frac{l_{max}}{2} l$. Both constraints regulate the cluster sizes by putting requirements on the cluster parameters $\mathbf{w}_j$. However, the latter does not depend on the segmentation.

4.2 Optimization

The above problem can be solved using block coordinate descent that alternates between the following two procedures:

(A) Given the current segmentation, update the clustering model, i.e., fixing $\{k_i\}$ and $\{s^i_t\}$, optimizing (7) w.r.t. $\{\mathbf{w}_j\}$, $\{y^i_t\}$, $\{\xi^i_t\}$, and $\{\beta_j\}$.

(B) Given the current clustering model, update the segmentation and cluster labels, i.e., fixing $\{\mathbf{w}_j\}$, optimizing (7) w.r.t. $\{k_i\}$, $\{s^i_t\}$, $\{y^i_t\}$, and $\{\xi^i_t\}$.

Note that $\{y^i_t\}$ and $\{\xi^i_t\}$ are optimized in both procedures. Procedure (A) performs MMC on a defined set of temporal segments. Procedure (B) updates the segmentation and cluster labels while fixing the weight vectors of the clustering model.

Procedure (B) is separable; each time series $\mathbf{X}^i$ can be updated independently of the others, efficiently using dynamic programming. We now describe this dynamic programming algorithm. For brevity, we drop the superscript i and consider updating the segmentation-labeling of time series $\mathbf{X}$. We need to find the change points $\{s_t\}$ and the cluster labels $\{y_t\}$ that minimize Eq. (7). Because $\sum_{j=1}^m ||\mathbf{w}_j||^2$ is fixed and Eq. (10) is replaced by Eq. (13), which is independent of the segmentation, optimizing Eq. (7) is equivalent to:

$$\begin{aligned}&\underset{k,s_t,y_t,\xi_t\geq 0}{\text{minimize}} \sum_{t=1}^k \xi_t \qquad\qquad (14)\\ &\text{s.t. } \forall t : 1 \leq s_{t+1} - s_t \leq l_{max}, s_1 = 0, s_{k+1} = len(\mathbf{X}), \\ &\quad \forall t : (\mathbf{w}_{y_t} - \mathbf{w}_y)^T \varphi(\mathbf{X}_{(s_t, s_{t+1}]}) \geq 1 - \xi_t \forall y \neq y_t.\end{aligned}$$

This can be solved using dynamic programming, which makes two passes over the time series $\mathbf{X}$. In the forward pass, at frame u ($1 \leq u \leq len(\mathbf{X})$), it computes the best objective value for segmenting and labeling truncated time series $\mathbf{X}_{(0,u]}$ (ignoring frames from $u+1$ onward), i.e.,

$$\begin{aligned}&f(u) = \min_{k,s_t,y_t,\xi_t\geq 0} \sum_{t=1}^k \xi_t, \qquad\qquad (15)\\ &\text{s.t. } \forall t : 1 \leq s_{t+1} - s_t \leq l_{max},\ s_1 = 0, s_{k+1} = u, \\ &\quad \forall t : (\mathbf{w}_{y_t} - \mathbf{w}_y)^T \varphi(\mathbf{X}_{(s_t, s_{t+1}]}) \geq 1 - \xi_t\ \forall y \neq y_t.\end{aligned}$$

The forward pass computes $f(u)$ and $\mathcal{L}(u)$ for $u = 1, \cdots, len(\mathbf{X})$ using the recursive formulas:

$$\begin{aligned}f(u) &= \min_{1\leq l\leq l_{max}} \{\xi(u,l) + f(u-l)\}, \\ \mathcal{L}(u) &= \underset{1\leq l\leq l_{max}}{\text{argmin}} \{\xi(u,l) + f(u-l)\}.\end{aligned}$$

Here $\xi(u,l)$ denotes the slack value of segment $\mathbf{X}_{(u-l,u]}$:

$$\begin{aligned}&\xi(u,l) = \max\{0, 1 - (\mathbf{w}_{\hat{y}} - \mathbf{w}_{\tilde{y}})^T \varphi(\mathbf{X}_{(u-l,u]})\}, \\ &\text{where} \quad \hat{y} = \underset{y}{\text{argmax}}\, \mathbf{w}_y^T \varphi(\mathbf{X}_{(u-l,u]}), \text{and} \\ &\qquad\qquad \tilde{y} = \underset{y\neq\hat{y}}{\text{argmax}}\, \mathbf{w}_y^T \varphi(\mathbf{X}_{(u-l,u]}).\end{aligned}$$

The backward pass of the algorithm finds the best segmentation for $\mathbf{X}$, starting with $s_{k+1} = len(\mathbf{X})$ and using the back-recursive formula:

$$s_t = s_{t+1} - \mathcal{L}(s_{t+1}).$$

Once the optimal segmentation has been determined, the optimal assignment of cluster labels can be found using:

$$y_t = \underset{y}{\operatorname{argmax}}\, \mathbf{w}_y^T \varphi(\mathbf{X}_{(s_t, s_{t+1}]}).$$

The total complexity for this dynamic programming algorithm is $\mathbf{O}(ml_{max}len(\mathbf{X}))$, which is linear in the length of the time series.

5 Experiments

This section describes two sets of experiments. In the first set of experiments, we compare the performance of MRMMC against MMC and k-means to illustrate the problem of unbalanced clusters. In the second set of experiments we compare the performance of MMTC to state-of-the-art algorithms for the TC problem on several time series datasets.

Our method has several parameters, and we found our algorithm robust to the selection of these parameters. We set up the slack parameters C and C_2 to 1 in our experiments. For the experiments in Sec. 5.1, we set $l = \frac{n}{3m}$ where n is the number of training samples and m is the number of classes. Similarly, for experiments in Sec. 5.2, we set $l' = \frac{\sum n_i}{3m}$ where $\sum n_i$ is the total lengths of all sequences and m is the number of classes.

5.1 Clustering Performance of MRMMC

We validated the performance of MRMMC on publicly available datasets from the UCI repository[1]. This repository contains many datasets, but not many of them have more than several classes and contain no categorical or missing attributes. We selected the datasets that were used in the experiments of Zhao et al. (2008) and added several ones to create a collection of datasets with diversified numbers of classes. In particular, we used Wine, Glass, Segmentation, Digits, and Letters. We compared our method against the MMC formulation of Zhao et al. (2008) and k-means.

In our experiments, we set the number of clusters equal to the true number of classes. To measure clustering accuracy, we followed the strategy used by Xu et al. (2004); Zhao et al. (2008), where we first took a set of labeled data, removed the labels and ran the clustering algorithms. We then found the best one-to-one association between the resulting clusters and the ground truth clusters. Finally, we reported the percentage of correct assignment. This is referred as *purity* in information theoretic measures (Meila, 2007; Tuytelaars et al., 2009). Initialization was done similarly for all methods. For each method and dataset, we first ran the algorithm with 10 random initializations on 1/10 of the dataset. We used the output of the run with lowest energy to initialize the final run of the algorithm on the full dataset. Table 1 displays the experimental results. As can be seen, our method consistently outperforms other clustering algorithms. The MMC formulation by Zhao et al. (2008) yields similar results to ours when the number of classes is two or three. However, when the number of classes is higher, MMC performance is significantly worse than ours; this is due to the problem of cluster degeneration.

Table 1: Clustering accuracies (%) of k-means (KM), MMC (Zhao et al., 2008), and MRMMC on UCI datasets. For each dataset, results within 1% of the maximum value are printed in bold. The second column lists the numbers of classes.

Dataset	m	KM	MMC	MRMMC
Digit 3,8	2	94.7	**96.6**	**96.6**
Digit 1,7	2	**100**	**100**	**100**
Wine	3	**95.8**	**95.6**	**96.3**
Digit 1,2,7,9	4	87.4	**90.4**	**90.5**
Digit 0,6,8,9	4	94.8	94.5	**97.6**
Glass	6	43.5	46.1	**48.8**
Segmentation	7	59.0	40.0	**66.1**
Digit 0-9	10	79.2	36.5	**85.1**
Letter a-j	10	**42.6**	28.6	**43.0**
Letter a-z	26	27.3	10.9	**33.8**

5.2 Temporal Clustering Experiments

This section describes experimental results on several time series datasets. In all experiments we measured the joint segmentation-clustering performance as follows. We ran our algorithm to obtain a segmentation and cluster labels. Each frame was then associated with a particular cluster, and we found the best cluster-to-class association between the resulting clusters and the ground truth classes. The overall frame-level accuracy was calculated as the percentage of agreement. For comparison, we implemented kMSeg (Robards and Sunehag, 2009) a generative counterpart of MMTC in which MRMMC is replaced by k-means.

5.2.1 Weizmann Dataset

The Weizmann dataset contains 90 video sequences of 9 people, each performing 10 actions. Each video sequence in this dataset consists of a single action. For

[1] http://archive.ics.uci.edu/ml/

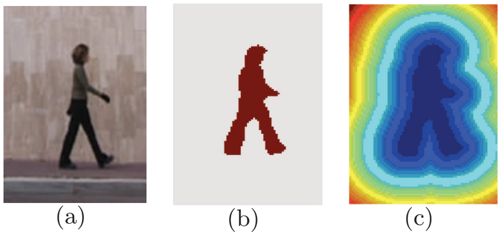

Figure 2: Weizmann dataset – frame-level features. (a): original frame, (b): binary mask, (c): Euclidean distance transform for frame-level features.

segmentation-clustering evaluation, we performed experiments on longer video sequences which were created by concatenating single-action sequences. We extracted binary masks (Fig. 2(b)) and computed Euclidean distance transform (Fig. 2(c)) for frame-level features (i.e., $\mathbf{X}_p$) as proposed by Gorelick et al. (2007). Following the success of the Bag-of-Words approach for document classification (Blei et al., 2003) and object recognition (Sivic et al., 2005), we built a dictionary of temporal words with 100 clusters using k-means, and $\varphi(\mathbf{X}_p)$ was the 100-dimensional binary indicator vector of the cluster that $\mathbf{X}_p$ was assigned to. Thus the representation of a segment was the histogram of temporal words in the segment.

Fig. 3 plots the frame-level accuracies as a function of the number of classes. We computed the frame-level accuracy for m classes ($2 \leq m \leq 10$) as follows. We randomly chose m classes out of 10 actions and concatenated video sequences of those actions (with random ordering) to form a long video sequence. We ran MMTC and kMSeg and reported the frame level accuracies as explained at the beginning of Sec. 5.2. We repeated the experiment with 30 runs; the mean and standard error curves are plotted in Fig. 3. As can be seen, MMTC outperformed kMSeg. In this experiment, the desired number of clusters was set to the true number of classes.

The above experiment assumed the true number of classes was known, but this might not be the case in reality. For sensitivity analysis, we performed an experiment where we fixed the number of true classes but varied the desired number of clusters. For this experiment, the evaluation criterion given at the beginning of Sec. 5.2 could not be applied because there was no one-to-one mapping between the resulting clusters and the ground truth classes. We instead used different performance criteria which were based on two principles: i) two frames that belong to the same class should be assigned to the same cluster; and ii) two frames that belong to different classes should be assigned to different clusters. Formally speaking, consider all pairs of same-class video frames, let p_1 be the percentage of pairs of which both frames were assigned to the same cluster. Consider all pairs of different-class video frames, let p_2 be the percentage of pairs of which two frames were assigned to different clusters. Let p_3 be the average of these two values $p_3 = (p_1 + p_2)/2$, which summarizes the clustering performance. These criteria are referred as pair-counting measures (Meila, 2007). Fig. 4 plots these values; the true number of classes is 10 while the desired number of clusters varies from 2 to 15. As the number of clusters increases, p_1 decreases while p_2 increases. However, the summarized value p_3 is not so sensitive to the desired number of clusters. Alternatively, the optimal number of clusters could be learned with cross-validation.

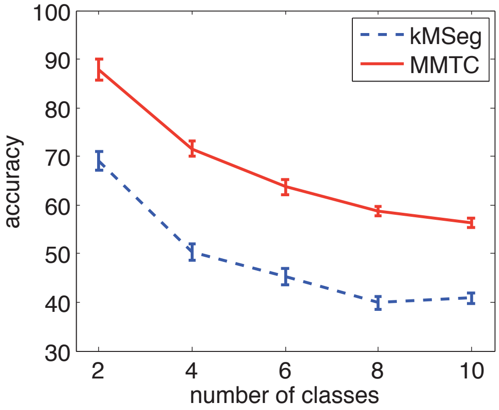

Figure 3: Segmentation-clustering accuracy as a function of the number of classes. MMTC outperforms kMSeg.

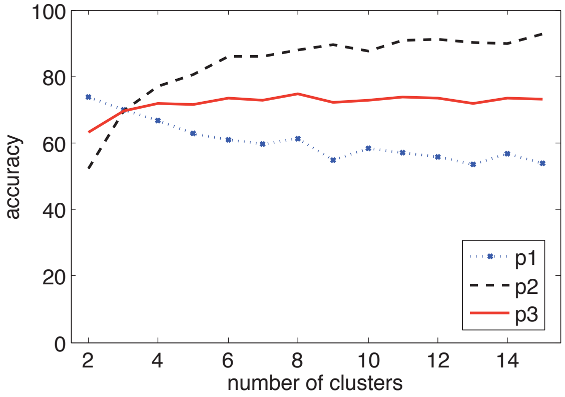

Figure 4: Sensitivity analysis – accuracy values when the desired number of clusters varies around 10, the true number of classes.

5.2.2 Honeybee Dance Dataset

The honeybee dataset (Oh et al., 2008) contains video sequences of honeybees that communicate the loca-

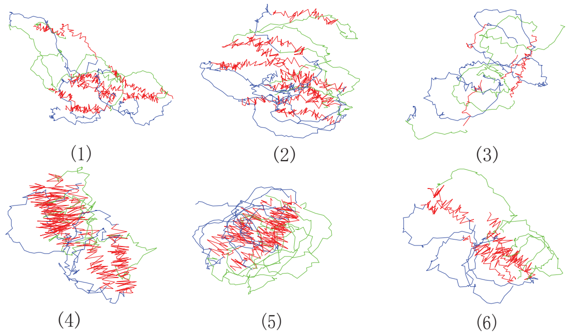

Figure 5: Trajectories of dancing bees. Each dance trajectory is the output of a vision-based tracker. The segments are color coded; red, green, and blue correspond to waggle, right-turn, and left-turn, respectively. This figure is best seen in color.

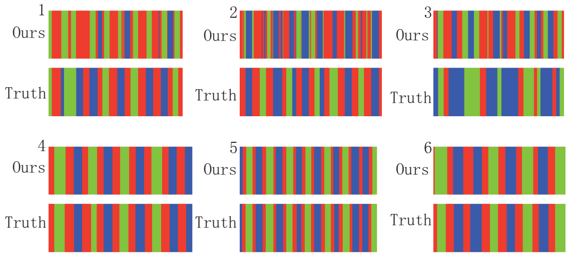

Figure 6: MMTC results versus human-labeled ground truth. Segments are color coded; red, green, blue correspond to waggle, right-turn, left-turn, respectively. This figure is best seen in color.

tion and distance to a food source through a dance that takes place within a hive. The dance can be decomposed into three different movement patterns: waggle, right-turn, and left-turn. During the waggle dance, the bee moves roughly in a straight line while rapidly shaking its body from left to right; the duration and orientation of this phase correspond to the distance and the orientation to the food source. At the endpoint of a waggle dance, the bee turns in a clockwise or counterclockwise direction to form a turning dance. The dataset consists of six video sequences with lengths 1058, 1125, 1054, 757, 609, and 814 frames, respectively. The bees were visually tracked, and their locations and head angles were recorded. The 2D trajectories of the bees in six sequences are shown in Fig. 5. The frame-level feature vector was $[v_x, v_y, \sin(v\theta), \cos(v\theta)]$, where (v_x, v_y) was the velocity vector and $v\theta$ was the angular velocity of the bee's head angle.

Following Altun et al. (2003) and inspired by HMMs, we propose to use two types of features, interactions between the observation vectors and the set of predefined states as well as the transition between states of neighboring frames: $\varphi(\mathbf{X}_p) = \begin{bmatrix} \phi^{obs}(\mathbf{X}_p) \\ \phi^{int}(\mathbf{X}_p) \end{bmatrix}$. Here $\phi^{obs}(\mathbf{X}_p)$ and $\phi^{int}(\mathbf{X}_p)$ are the observation and interaction feature vectors, respectively. These feature vectors are computed as follows. First we build a dictionary of temporal words by clustering the raw feature vectors from the time series in the dataset. Let $\mathbf{c}_1, \cdots, \mathbf{c}_r$ denote the set of clustering centroids. We consider $\phi^{obs}(\mathbf{X}_p)$ as a $r \times 1$ vector with the i^{th} entry is $\phi^{obs}_i(\mathbf{X}_p) = \mu \exp(-\gamma ||\mathbf{X}_p - \mathbf{c}_i||^2)$. Intuitively, the i^{th} entry of observation vector is the pseudo-probability that $\mathbf{X}_p$ belongs to state i, which is proportional to how close $\mathbf{X}_p$ to the cluster centroid $\mathbf{c}_i$. The scale factor μ is chosen such that the sum of the entries of $\phi^{obs}(\mathbf{X}_p)$ is one. The interaction feature vector $\phi^{int}(\mathbf{X}_p)$ is defined as a $r^2 \times 1$ vector, with:

$$\phi^{int}_{(u-1)r+v}(\mathbf{X}_p) = \phi^{obs}_u(\mathbf{X}_p)\phi^{obs}_v(\mathbf{X}_{p-1}) \ \forall u, v = 1, \cdots, r.$$

The above quantity is the pseudo-probability for transitioning from state v to state u at time p. The interaction feature vector depends on both the observation vectors of the frame $\mathbf{X}_p$ and the preceding frame $\mathbf{X}_{p-1}$. In our experiment, we set $r = 15$.

Tab. 2 displays the experimental results of MMTC, kMSeg, and NPSLDS (Fox et al., 2009). NPSLDS is a non-parametric method combining hierarchical Dirichlet process prior and a SLDS. The reported numbers in Tab. 2 are frame-level accuracy (%) measuring the joint segmentation-clustering performance as described at the beginning of Sec. 5.2. For MMTC and kMSeg, we show both the averages and standard errors of the results over 20 runs. For each honeybee sequence, results within 1% of the maximum value are printed in bold. MMTC achieves the best or close to the best performance on five out of six sequences, and it has the highest overall accuracy. For some sequences, the results of our method are very close to those of the best supervised method (Oh et al., 2008) which are 75.9, 92.4, 83.1, 93.4, 90.4, and 91.0. Fig. 6 displays side-by-side comparison of the prediction result and the human-labeled ground truth. In this experiment, the desired number of clusters was set to 3. The coordinate descent optimization algorithm of MMTC required 34 iterations on average (for convergence). Notably, the results for the first three sequences are worse than those for the other sequences. This not only happens for MMTC but also for all other methods, including the supervised one (Oh et al., 2008). On a close examination of the honeybee videos, we find that the human annotation for the first three sequences are noisy.

Table 2: Joint segmentation-clustering accuracy (%) on the honeybee dataset. NPSLDS results were published by Fox et al. (2009). MMTC and kMSeg results are averaged over 20 runs; the standard errors are also shown. Results within 1% of the maximum values are displayed in bold. Our method achieves the best or close to the best result on five out of six sequences, and it has the highest average accuracy.

Sequence	1	2	3	4	5	6	Mean
NPSLDS	45.0	42.7	47.3	88.1	**92.5**	**88.2**	67.3
kMSeg	**51.5 ± .01**	50.1 ± .15	46.7 ± .12	**91.0 ± .07**	**91.7 ± .07**	84.7 ± 2.27	69.3 ± .45
MMTC	**51.0 ± .56**	**66.6 ± 2.39**	**48.3 ± .25**	**91.6 ± .16**	91.2 ± .02	**88.8** ±.07	**72.9 ± .57**

6 Conclusions

This paper proposes MMTC, a novel framework for simultaneous segmentation and clustering of time series. Clustering is performed robustly using temporal extensions of MMC for learning discriminative patterns whereas the inference over the segments is done efficiently with dynamic programming. Experiments on several real datasets in the context of human activity and honeybee dancing showed that our discriminative clustering often led to segmentation-clustering accuracy superior to the performance obtained with generative methods. Although the results presented in the paper excelled state-of-the-art algorithms, several open research problems that need to be addressed in future work. First, currently, the number of clusters is assumed to be known. In order to automatically select the optimal number of clusters, criteria similar to Akaike Information Criterion or Minimum Description Length could be added to the MMTC formulation. Second, MMTC is susceptible to local minima, and although random initialization with multiple restarts has worked well, better initialization strategies or convex approximations to the problem will be worth exploring in future work. Finally, traditional algorithms for supervised temporal segmentation heavily rely on label data. Accurately labeling ground truth data to recognize activities or event is time consuming. More importantly, the labeling process is often subjective and difficult to standardize across coders; for example, it is unclear how to consistently determine the start and the end of a particular action. In this context, TC algorithms that can discover temporal patterns in an unsupervised fashion could improve speed and reliability of manual coding. We will explore this use in future applications.

Acknowledgements

This work was supported by the National Science Foundation RI-1116583 and partially supported by the National Science Foundation under Grant CPS-0931999. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. The authors would like to thank Zhen-Zhong Lan for helping with the experiments.

The first author, Minh Hoai, would like to dedicate his share of this work to Professor Nguyen Duy Tien on the occasion of his 70^{th} birthday, with highest regards and gratefulness.

References

Altun, Y., Tsochantaridis, I., and Hofmann, T. (2003). Hidden Markov support vector machines. In *International Conference on Machine Learning*.

Bach, F. R. and Harchaoui, Z. (2009). DIFFRAC: a discriminative and flexible framework for clustering. In *Neural Information Processing Systems*.

Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(4–5):993–1022.

Cutler, R. and Davis, L. (2000). Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796.

De la Torre, F. and Kanade, T. (2006). Discriminative cluster analysis. In *International Conference on Machine Learning*.

Do, T.-M.-T. and Artières, T. (2009). Large margin training for hidden markov models with partially observed states. In *International Conference on Machine Learning*.

Estevan, Y. P., Wan, V., and Scharenborg, O. (2007). Finding maximum margin segments in speech. In *Acoustics, Speech and Signal Processing*.

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2009). Nonparametric Bayesian learning of switching linear dynamical systems. In *Advances in Neural Information Processing Systems*.

Gieseke, F., Pahikkala, T., and Kramer, O. (2009). Fast evolutionary maximum margin clustering. In *International Conference on Machine Learning*.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. (2007). Actions as space-time shapes.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(12):2247–2253.

Guerra-Filho, G. and Aloimonos, Y. (2006). Understanding visuo-motor primitives for motion synthesis and analysis. *Computer Animation and Virtual Worlds*, 17(3-4).

Harchaoui, Z., Bach, F., and Moulines, E. (2009). Kernel change-point analysis. In *Neural Information Processing Systems*.

Hoai, M., Lan, Z.-Z., and De la Torre, F. (2011). Joint segmentation and classification of human actions in video. In *Proceedings of IEEE Conference of Computer Vision and Pattern Recognition*.

Hu, Y., Wang, J., Yu, N., and Hua, X.-S. (2008). Maximum margin clustering with pairwise constraints. In *International Conference on Data Mining*.

Keogh, E. and Lin, J. (2005). Clustering of time series subsequences is meaningless: Implications for previous and future research. *Knowledge and Information Systems*, 8:154–177.

Liao, T. W. (2005). Clustering of time series data – a survey. *Pattern Recognition*, 38(11):1857–1874.

Meila, M. (2007). Comparing clusterings – an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

Nguyen, M. H., Torresani, L., De la Torre, F., and Rother, C. (2009). Weakly supervised discriminative localization and classification: a joint learning process. In *Proceedings of International Conference on Computer Vision*.

Oh, S. M., Rehg, J. M., Balch, T., and Dellaert, F. (2008). Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *International Journal of Computer Vision*, 77(1–3):103–124.

Pogalin, E., Smeulders, A., and Thean, A. (2008). Visual quasi-periodicity. In *Computer Vision and Pattern Recognition*.

Robards, M. and Sunehag, P. (2009). Semi-Markov kMeans clustering and activity recognition from body-worn sensors. In *International Conference on Data Mining*.

Sha, F. and Saul, L. K. (2007). Large margin hidden markov models for automatic speech recognition. In *Advances in Neural Information Processing Systems*.

Shi, Q., Wang, L., Cheng, L., and Smola, A. (2008). Discriminative human action segmentation and recognition using semi-Markov model. In *Computer Vision and Pattern Recognition*.

Sivic, J., Russell, B., Efros, A., Zisserman, A., and Freeman, W. (2005). Discovering objects and their location in images. In *Proceedings of International Conference on Computer Vision*.

Turaga, P., Veeraraghavan, A., and Chellappa, R. (2009). Unsupervised view and rate invariant clustering of video sequences. *Computer Vision and Image Understanding*, 113(2).

Tuytelaars, T., Lampert, C. H., Blaschko, M. B., and Buntine, W. (2009). Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88(2):284–302.

Vecchio, D. D., Murray, R. M., and Perona, P. (2003). Decomposition of human motion into dynamics-based primitives with application to drawing tasks. *Automatica*, 39(12).

Wang, F., Wang, X., and Li, T. (2009). Maximum margin clustering on data manifolds. In *International Conference on Data Mining*.

Xu, L., Neufeld, J., Larson, B., and Schuurmans, D. (2004). Maximum margin clustering. In *Advances in Neural Information Processing Systems*.

Xu, L. and Schuurmans, D. (2005). Unsupervised and semi-supervised multi-class support vector machines. In *AAAI Conference on Artificial Intelligence*.

Xuan, X. and Murphy, K. (2007). Modeling changing dependency structure in multivariate time series. In *International Conference on Machine Learning*.

Zelnik-Manor, L. and Irani, M. (2006). Statistical analysis of dynamic actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1530–1535.

Zhang, J., Marszalek, M., Lazebnik, S., and Schmid, C. (2001). Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73(2):213–238.

Zhao, B., Kwok, J., Wang, F., and Zhang, C. (2009). Unsupervised maximum margin feature selection with manifold regularization. In *Computer Vision and Pattern Recognition*.

Zhao, B., Wang, F., and Zhang, C. (2008). Efficient multiclass maximum margin clustering. In *International Conference on Machine Learning*.

Zhou, F., De la Torre, F., and Cohn, J. F. (2010). Unsupervised discovery of facial events. In *Computer Vision and Pattern Recognition*.

Zhou, F., De la Torre, F., and Hodgins, J. K. (2008). Aligned cluster analysis for temporal segmentation of human motion. In *IEEE Conference on Automatic Face and Gestures Recognition*.