2014 IEEE International Conference on Data Mining

Early Classification of Ongoing Observation

Kang Li Sheng Li
Department of Electrical and Computer Engineering
College of Engineering
Northeastern University
Boston, USA
Email: li.ka@husky.neu.edu, shengli@ece.neu.edu

Yun Fu
Department of Electrical and Computer Engineering,
College of Computer and Information Science (Affiliated)
Northeastern University
Boston, USA
Email: yunfu@ece.neu.edu

***Abstract*—This work focuses on early classification of ongoing observation of the object, which is beneficial for a number of applications that require time-critical decision making. We propose an approach for discovering two key aspects of multivariate time series (m.t.s.) observation, (1) *Temporal Dynamics* and (2) *Sequential Cues*. The key idea is that m.t.s. observation can be represented as an instantiation of a Multivariate Marked Point-Process (Multi-MPP). Each variable characterizes the *temporal dynamics* of a particular feature event of an object, where both timing and strength information of that feature event are preserved. To make this model computationally practical, we introduce the Multilevel-Discretized Marked Point-Process (MD-MPP) model which can ensure a good piece-wise stationary property both in the time-domain and mark-space while preserving dynamics as much as possible. Based on this model, another important temporal patterns of early classification, *sequential cues* among variables, becomes formalizable. We construct a probabilistic suffix tree to represent sequential patterns among features in terms of Variable order Markov Model (VMM). The effectiveness of our approach is evaluated on three experimental scenarios. Our method achieves superior performance for early classification of ongoing m.t.s. observation data.**

***Keywords*-Early Classification; Time Series; Temporal Dynamics; Sequential Cue;**

I. Introduction

In the fields of data mining and machine learning, many problems involve classifying Multivariate Time Series (m.t.s.) observation data. The classical supervised learning task is to construct a classifier from training m.t.s. samples that can correctly predict the classes of new samples after they are fully observed. However, in many cases, a quick decision without waiting to the end of the observation is desired.

Early classification of ongoing observation of the object is highly valuable in a large variety of time-critical applications. For instance, it can be of tremendous help by identifying the illness at the early stages before the full-blown symptoms erupt. In human-computer interaction, people's intention can be predicted by early recognizing human actions or hand gestures captured by sensors or cameras, which may greatly reduce the system response time and provide a more natural experience of communication. In many real-time required systems, such as closed-captioning, early recognition would be an effective way to avoid the sense of delay. Another interesting application would be in social media monitoring, such as predicting president election or passage of law bills, where waiting for a fully observed data is pointless.

Though the problem of early classification arises in a wide variety of applications, it is quite a new topic for the domain of statistical learning. Existing works are either focusing on Univariate Time Series (u.t.s.) [1], [2] or from application perspectives by tuning on traditional time series classification models [3]. The disadvantages of previous work are three folds. First, many approaches assume that the time series observations from the same class will always have equal durations, which reduced the problem into a significantly simplified one (simple distance measuring between samples). In terms of early classification task, the equal length assumption also implicitly means that we can exactly tell how much an ongoing time series has progressed and when it will be finished. But in most of the real world applications, this assumption cannot hold. Secondly, an important factor, temporal correlations among variables of m.t.s., are not fully considered, which can be quite informative for identifying the object class at early stage of observation. For instance, in human action recognition, a particular action is a combined motion of multiple joints with temporal order. Thirdly, all previous work [1], [2], [3] are extensions of traditional distance based approach, which are computational too demanding. However, in many cases, the practical merit of early classification lies in a quick and accurate recognition.

In this paper, we propose a novel approach to early classify multivariate time series (m.t.s.) data by modeling two types of time pattern: (1) **Temporal Dynamics** and (2) **Sequential Cue**, shown in Fig. 1.

Our key idea is that m.t.s. observation can be represented as an instantiation of a Multivariate Marked Point-Process (Multi-MPP). Each dimension of Multi-MPP characterizes the *temporal dynamics* of a particular property of the object, where both timing and strength information are kept. Since a full parameter estimation of Multi-MPP can easily become impractical with the increase of the number of time instants, in this paper, we introduce Multilevel-Discretized Marked Point-Process Model (MD-MPP), which is a class of Multi-MPP that can ensure a good piece-wise stationary property both in time-domain and mark-space while keep dynamics as much as possible. Based on this representation, another important temporal pattern of early classification, *sequential*

1550-4786/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDM.2014.100

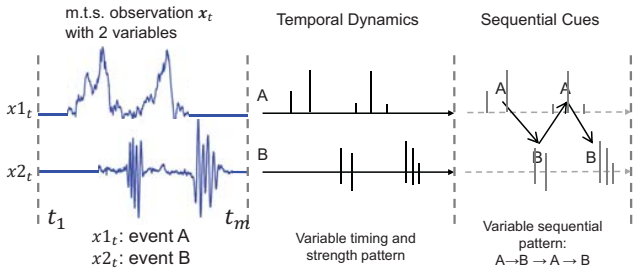


Fig. 1. Temporal Dynamics and Sequential Cues. Each dimension of x_t is considered as an event, whose timing and strength information are characterized by MPP. The temporal order of firing patterns among events contains important sequential cue to early recognize the class of ongoing observation. For example, assuming $A \rightarrow B$ pattern is discriminative for this class, then we can make classification decision when we observed only half of the m.t.s. data.

cue, becomes formalizable. We construct a probabilistic suffix tree (PST) to represent sequential patterns among variables (feature dimensions) in terms of variable order Markov dependencies. We use MD-MPP+TD to denote this extended version of our approach, in which *temporal dynamics* and *sequential cue* are integrated. In order to test the efficacy of our method, comprehensive evaluations are performed on three real world datasets. The proposed algorithms achieve superior performance for early classification of m.t.s. data.

II. Related Work

In general, there are three categories of works that are mostly related to ours.

Classification of time series has attracted great interest from the data mining community. The dynamic and continuous nature of time series makes it an interesting research problem (see reviews [4], [5], and recent work [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]). Though one-nearest neighbor (1NN) with Dynamic Time Warping (DTW) [16], [17], [7], [9] still stays state-of-the-art for most time series problems, its high computational cost drives researchers for lighter solutions of classification. In recent years, much effort has been dedicated to finding compact and effective features such as segmented-based features (SBFs) [6], time series shapelets [7], or a generic set of features [10]. All the works mentioned above are focusing on u.t.s.. A natural generalization of time series classification is to deal with m.t.s., but only a few work looked into this direction. And most of them are trying to design a special type of feature for multivariate case, such as temporal logic of trends between variables [18] or domain expert defined substructures [19]. However the m.t.s. problem of recognizing simultaneous movement of a collection of time series (or stochastic processes) which are correlated to each other has received much less attention. In our work, temporal correlations among variables are modeled effectively through Variable order Markov Model (VMM), which encodes the learned sequential patterns as important cues for early classification.

Early classification of time series. While there is a vast amount of literature on classification of time series, early classification of ongoing time series is ignored until quite recently [1], [3], [2], [20]. The unique and non-trivial challenge here is that either features or distance metrics formulated in previous work for classification of time series might not be robust, when whole time series is not available. Additionally, early classification always makes stricter demands on time efficiency, because the algorithm will lose its merit, if it unintentionally forces us to wait till the end of time series. To the best of our knowledge, the work of [1] first explicitly proposed a solution of early classification of time series to the community, though similar concepts have been raised in other two works [21], [22]. They developed ECTS (Early Classification on Time Series) algorithm, which is an extension of 1NN classification method. ECTS evaluates neighbors both in full observation and prefixes of time series. But their algorithm is only limited to u.t.s. data and assuming that all time series samples have the same length. Following the spirit of the classic work in [7] on discovering interpretable time series shapelets, [3] and [2] extend it to the early classification scenarios. However all three methods are distance based approaches, the inherent efficiency problem is not considered for earliness. Sometime things might even get worse with necessary sliding widow search for shapelets along a long time series [3]. Another important aspect of early classification, Sequential Cue, is also missing in the discussion.

Point process model. As a special type of stochastic process, point process has gained a lot of attention recently in the statistical learning community because of its powerful capability on modeling and analyzing rich dynamical phenomena [23], [24], [25], [26]. Adopting a point process representation of random events in time opens up pattern recognition to a large class of statistical models that have seen wide applications in many fields. Gunawardana et al. [24] propose a variant of MPP model, called Piecewise-Constant Conditional Intensity Model (PCIM) for learning temporal dependencies in event streams. Their algorithm is evaluated on two real world applications: modeling supercomputer event logs and forecasting future interests of Web search users. Although rich temporal structure information is encoded, they did not consider any classification possibility from that point. Prabhakar et al. [26] used MPP as a representation for visual events, and try to identify temporal patterns of human interactions by applying pairwise test for Granger causality. They are from an interpretation point of view, rather than a recognition point of view. Also, Kim et al. [25] investigated the problem of Web image prediction by developing a predictive framework based on MPP. They focus on predicting future event rather than early classification of m.t.s.. Jansen and Niyogi [23] applied point process model in the context of speech recognition, especially for obstruent super-segment decoding. A general framework for other domains is not considered.

III. Preliminaries

A. Notation and Problem Definition

For better illustration, Table I summarizes the abbreviations and notations used throughout the paper. We begin by defining the key terms in the paper. We use lowercase letters to

TABLE I
SYMBOL AND ABBREAVATION

Abbr.	Description
u.t.s	univariate time series
m.t.s	multivariate time series
MPP	marked point process
MD-MPP	multilevel-discretized marked point-process
1NN	1 nearest neighbor
DTW	dynamic time warping
PST	probabilistic suffix tree
VMM	variable order Markov model

Symbol	Description
X	Observation of time series with full length
X', Y'	ongoing time series
$\mathbf{X}^d$	set of d-dimensional m.t.s.
$\mathbf{D}$	time series training data set
$\mathbf{T}$	time (index set)
$\mathbf{C}$	set of class labels
$\|X\|$	length of time series
$\mathcal{F}$	classifier
$\widetilde{\mathbf{N}}$	multivariate point-process
$\ddot{\widetilde{\mathbf{N}}}$	multivariate marked point-process
S	number of segments by factoring time line
$\mathbf{\Lambda}$	trained MD-MPP model
$\mathbf{E}$	set of events
$\overline{\mathbf{D}}_\Lambda$	set of sampled discrete event streams from model Λ
$\overline{\mathbf{D}}_{Y'}$	set of sampled discrete event streams from testing Y'
$\bar{a}_i$	discrete event stream

represent scalar values, lowercase bold letters to represent vectors. We use uppercase letters to represent time series, uppercase bold letters to represent sets.

Definition 1: *Multivariate Time Series*: A multivariate time series $X = \{\mathbf{x}_t : t \in \mathbf{T}\}$ is an ordered set of real-valued observations, where $\mathbf{T}$ is the index set consist of all possible time stamps. If $\mathbf{x}_t \in \mathbb{R}^d$, where $d > 1$, for instance $\mathbf{x}_t = \langle x_t^1, x_t^2, ..., x_t^d \rangle$, X is called a d-dimensional m.t.s..

In this paper, observations $\mathbf{x}_t$ are always arranged by temporal order with equal time intervals.

Definition 2: *Classification of Multivariate Time Series*: A m.t.s. $X = \{\mathbf{x}_t : t \in \mathbf{T}\}$ may globally carry a class label. Given $\mathbf{C}$ as a set of class labels, and a training set $\mathbf{D} = \{\langle X_i, C_i \rangle : C_i \in \mathbf{C}, i = 1, ..., n\}$, the task of classification of m.t.s. is to learn a classifier, which is a function $\mathcal{F} : \mathbf{X}^d \to \mathbf{C}$, where $\mathbf{X}^d$ is the set of d-dimensional m.t.s..

We use $|X|$ to represent the *length* of time series, namely $X = \{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, ..., \mathbf{x}_{t_{|X|}}\}$. By default, X is considered as the full-length observed time series, while a corresponding *ongoing time series* of X is denoted as $X' = \{\mathbf{x}'_{t_1}, \mathbf{x}'_{t_2}, ..., \mathbf{x}'_{t_{|X'|}}\}$, where $\mathbf{x}'_{t_i} = \mathbf{x}_{t_i}$ for $i = 1, ..., |X'|$, and $t_{|X'|} < t_{|X|}$. The ratio $p = |X'|/|X|$ is called the *progress level* of X'. It's obvious that the progress level of full-length observed time series is always 1. We use X'_p to indicate an ongoing time series with progress level p.

Definition 3: *Early Classification of Multivariate Time Series*: Given training set $\mathbf{D} = \{\langle X_j, C_j \rangle : C_j \in \mathbf{C}, j = 1, ..., n\}$ with n m.t.s. samples, the task of early classification of m.t.s. is to learn a classifier, which is a function $\mathcal{F} : \mathbf{X}' \to \mathbf{C}$, where $\mathbf{X}'$ is the set of ongoing m.t.s..

Specifically, we can do classification along the progress of time series, and predict the class label at different progress levels of X, generating a bunch of decisions, $\{\mathcal{F}(X'_{p_1}), \mathcal{F}(X'_{p_2}), ..., \mathcal{F}(X'_1)\}$. In this paper we use 5% of full duration as an interval of generating a new prediction result, which results in 20 rounds of classification for different progress levels. Intuitively, the prediction accuracy should go up with the increasing progress level, since we observed more information. But, interestingly, through our evaluations at later sections, we found that, sometimes, it is quite contradictory to our common sense. The reason is that observations at different segments of time series may have different discriminativeness for classification task, and how the discriminative segments distribute along the timeline really depends on the data.

B. Multivariate Marked Point-process

In probability theory, *stochastic process* is sequence of random variables indexed by a totally ordered set $\mathbf{T}$ ("time"). *Point process* is a special type of stochastic process which is frequently used as models for firing pattern of random events in time. Specifically, the process counts the number of events and record the time that these events occur in a given observation time interval.

Definition 4: A d-dimensional *multivariate point-process*is described by $\widetilde{\mathbf{N}} = \langle N^1, N^2, ..., N^d \rangle$, where $N^i = \{t_1^i, t_2^i, ..., t_m^i\}$ is a univariate point-process, and t_k^i indicates the time stamps on which a particular "event" or "property"[1] x_i has been detected. $N^i(t)$ is the total number of observed event x_i in the interval $(0, t]$, for instance, $N^i(t_k^i) = k$. Then, $N^i(t+\Delta t) - N^i(t)$ represents the number of detections in the small region Δt. Similarly, $\widetilde{\mathbf{N}}(t) = \langle N^1(t), N^2(t), ..., N^d(t) \rangle$,

By letting $\Delta t \to 0$, we can have the *intensity function* $\mathbf{\Lambda}(t) = \{\lambda^i(t)\}$, which indicates the expected occurrence rate of the event x^i at time t: $\lambda^i(t) = \lim_{\Delta t \to 0} N^i(t+\Delta t) - N^i(t)$ [27]. This is the key to identify a point process.

In many real world applications, the time landmarks of events arise not as the only object of study but as a component of a more complex model, where each landmark is associated with other random elements $M^i = \{x_1^i, x_2^i, ..., x_m^i\}$, called marks, containing further information about the events. Each (t_k^i, x_k^i) is a marked point, and the sequence $\{(t_k^i, x_k^i)\}$ of marked points is referred to as a *marked point processes*.

Definition 5: A d-dimensional *multivariate marked point process* is described as following:

$$\ddot{\widetilde{\mathbf{N}}} = \langle \{N^1, M^1\}, \{N^2, M^2\}, ..., \{N^d, M^d\} \rangle \qquad (1)$$

where $\{N^i, M^i\} = \{(t_k^i, x_k^i)\}$ on $\mathbb{R}^+ \times \mathbb{R}$ is a univariate marked point process.

IV. METHODOLOGY

In this section, we describe the proposed early classification approach for m.t.s. data. Our basic idea is to construct early classification function $\mathcal{F}(Y')$ by using the knowledge learned from a *temporal dynamics* model $\Pr(Y'|\mathbf{\Lambda})$ (Section IV-A) and a *sequential cue* model $\Pr(Y'|\mathbf{\Phi})$ (IV-B). We use MD-MPP to

[1]In this paper, the concepts "variable", "property" or "event" are interchangeably used to refer to a certain dimension of m.t.s.

denote the first model, and MD-MPP+TD to denote the second model. Giving an ongoing m.t.s. Y' in a domain application with $|\mathbf{C}|$ classes, the final prediction functions can be written as:

$$\text{MD-MPP:} \quad \mathcal{F}(Y') = \arg\max_{c\in\mathbf{C}}\{\Pr^c(Y'|\mathbf{\Lambda})\};$$
$$\text{MD-MPP+TD:} \quad \mathcal{F}(Y') = \arg\max_{c\in\mathbf{C}}\{\Pr^c(Y'|\mathbf{\Phi})\}.$$

The bases of our method are following two insights: 1) m.t.s. can be interpreted as an instantiation of a Multi-MPP. Each dimension of Multi-MPP characterizes the temporal dynamics of a particular property of the object, where both timing and strength information are kept; 2) The identification of sequential patterns among multiple variables of m.t.s. allows us to utilize these sequential cues for early classification.

Specifically, our approach consists of two stages. The first stage encodes the m.t.s. as a multi-level discretized marked point process, which not only characterizes the temporal dynamics of m.t.s., but also provides discretized intensity map that governs the generation of discrete events streams. The second stage analyzes these discrete events streams to discover the temporal correlations. We will go to the details of each component of our approach in following subsections.

A. Temporal Dynamics

In this paper, we focus on multivariate time series data. In our opinion, these observations can be (1) adequately represented as a collection of perceptual events that are tied to a common clock or constant frame rate, and (2) decoded according to the temporal statistics of such events. The need therefore arises to formulate and evaluate recognition strategies that can operate on representations based on the firing patterns of various events. In the following, we will introduce our multilevel-discretized marked point-process (MD-MPP) model $\tilde{\tilde{\mathbf{N}}}_X$ to achieve this.

Given a d-dimensional m.t.s. $X = \{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, ..., \mathbf{x}_{t_{|X|}}\}$, where $\mathbf{x}_t = \langle x_t^1, x_t^2, ..., x_t^d\rangle$, and $t = t_1, t_2, ..., t_{|X|}$. We consider each dimension x^i as a noisy detector of certain perceptual event. Those event detectors generate continuous values which indicate the strength or the confidence about the detection. We call these continuous value based observations as *marks*. Then the corresponding marked point process representation of X is:

$$\tilde{\tilde{\mathbf{N}}}_X = \left\langle \{N_X, M_X^1\}, \{N_X, M_X^2\}, ..., \{N_X, M_X^d\}\right\rangle, \qquad (2)$$

where $\{N_X, M_X^i\} = \{(t_k, x_{t_k}^i)\}$, and $k = 1, ..., |X|$. We can see that different variables shared a common clock N_X.

To allow for more clarity in understanding the approach, we will develop the model step by step by relaxing assumptions from ideal case to the real case. In computation dealing with time series, the number of time instants are often in the hundreds or thousands. Dealing with so many variable density function is cumbersome at best and often impractical. We need to think about special cases which may simplify things. The first drastic simplification is to have an ideal case with following assumptions:

Assumption 1: For each event x^i, the corresponding point process is an independent stochastic sequence (to be relaxed in Section IV-A);

Assumption 2: We have a perfect detector for each event x^i, namely, $x_t^i \in \{0, 1\}$, where a spiking ($x_t^i = 1$) indicates occurrence of x^i, or there is no detection of x^i ($x_t^i = 0$) (to be relaxed in Section IV-A);

Assumption 3: For m.t.s. X, events are independent to each other (to be relaxed in Section IV-B).

Based on above assumptions, we have first representation model for m.t.s.: *Stationary point process*:

$$\begin{aligned}\Pr(N^i) &= \prod_{k=1}^{|X|} \frac{(\lambda^i\Delta t)^{\mathbf{1}_{N^i}(t_k)}}{\mathbf{1}_{N^i}(t_k)!} e^{-\lambda^i\Delta t} \qquad (3)\\ &= (\lambda^i\Delta t)^{m^i} e^{-\lambda^i T}\end{aligned}$$

where X is a m.t.s., $N^i = \{t_k | x_{t_k}^i = 1, t_k \le t_{|X|}\}$ is the point process for event x^i of X. $N^i(t_{|X|}) = |\{t_k | x_{t_k}^i = 1, t_k \le t_{|X|}\}| = m^i$ is the numbers of detection of event x^i in X, $\Delta t = t_{k+1} - t_k$ is the time interval between two consecutive observations. Assuming the whole process is contained in interval $(0, T]$, then $T = (|X| - 1)\Delta t$. The indicator function $\mathbf{1}_{N^i}(t_k)$ is 1 if $t_k \in N^i$ and 0 otherwise.

Given training dataset $\mathbf{D} = \{\langle X_j, C_j\rangle : C_j \in \mathbf{C}, j = 1, ..., n\}$, and point process representation $\tilde{\mathbf{N}} = \left\langle N^1, N^2, ..., N^d\right\rangle$ and duration time T, the data likelihood can be computed as

$$\Pr(\tilde{\mathbf{N}}|\mathbf{D}) = \prod_{i=1}^{d} \Pr(N^i|\mathbf{D}) = \prod_{i=1}^{d} (\lambda^i(\mathbf{D})\Delta t)^{m^i} e^{-\lambda^i(\mathbf{D})T} \qquad (4)$$

where $\lambda^i(\mathbf{D})$ depends both on the event and the training data.

Then training this model amounts to estimating $\lambda^i(\mathbf{D})$ for each $\langle i, \mathbf{D}\rangle$ pair. If we are given n training sequences containing in $\mathbf{D}$, and there are m_j^i of landmarks (spiking) of event x^i in j-th training sample, then, we can estimate $\lambda^i(\mathbf{D})$ by using the maximum log-likelihood estimation, which is:

$$\begin{aligned}\lambda^{i^*}(\mathbf{D}) &= \arg\max_{\lambda^i} \log(\lambda^i(\mathbf{D})\Delta t)^{m^i} e^{-\lambda^i(\mathbf{D})T}) \qquad (5)\\ &= \frac{\Sigma_{j=1}^n m_j^i}{\Sigma_{j=1}^n \Delta t|X_j|}.\end{aligned}$$

Relax Assumption 1.

Next, we will relax assumption 1 by adding *dynamic* property in the point process representation. This follows the *piece-wise stationary global-wise dynamic point process*:

$$\Pr(N^i) = \prod_{s=1}^{S} \frac{(\lambda^i(s)\Delta t\Delta\tau)^{m_s^i}}{m_s^i!} e^{-\lambda^i(s)\Delta t\Delta\tau} \qquad (6)$$

where assuming we evenly divide the time line into S pieces of equal length segments. Inside each segment, the point process is assumed stationary. $\Delta\tau = \lfloor |X|/S \rfloor$ is the division length in term of number of observations, so the progress level at the end of s-th time division is $p = (s\Delta\tau\Delta t)/(|X|\Delta t) = s\Delta\tau/|X|$. m_s^i is the number of detection of event x^i in time division s.

Given training m.t.s. dataset $\mathbf{D}$, and point process representation $\widetilde{\mathbf{N}}$, the data likelihood can be computed as

$$\begin{aligned}\Pr(\widetilde{\mathbf{N}}|\mathbf{D}) &= \prod_{i=1}^{d}\Pr(N^i|\mathbf{D}) \\ &= \prod_{i=1}^{d}\prod_{s=1}^{S}\frac{(\lambda^i(s,\mathbf{D})\Delta t\Delta\tau)^{m_s^i}}{m_s^i!}e^{-\lambda^i(s,\mathbf{D})\Delta t\Delta\tau}\end{aligned} \tag{7}$$

where the intensity function $\lambda^i(s,\mathbf{D})$ depends on the event, the time division (progress level) and the training data.

Then training this model amounts to estimating $\lambda^i(\mathbf{D})$ for each tuple $\langle i,s,\mathbf{D}\rangle$. If we are given n training sequences containing in $\mathbf{D}$, and there are $m_{j,s}^i$ of landmarks (spiking) of event x^i in j-th training sample's s-th division, then, we can estimate $\lambda^i(s,\mathbf{D})$ by using the maximum log-likelihood estimation, which is:

$$\lambda^{i*}(s,\mathbf{D}) = \frac{\Sigma_{j=1}^{n} m_{j,s}^i}{\Sigma_{j=1}^{n}\Delta t\Delta\tau_j}. \tag{8}$$

Relax Assumption 2.

In practise, we always get a noisy detector for each event, such as m.t.s. data. In the following, we will keep piece-wise stationary property and relax assumption 2 by allowing event detectors generating continuous values which may indicate the strength or the confidence about the detection. We call these continuous value based observations as *marks*, then the whole m.t.s. can be extended to a marked point process representation. To deal with this complexity, we introduce the *multilevel-discretized marked point-process* (MD-MPP).

Algorithm 1 Guess the progress level p^*

1) **Find the possible range from training set**: Let $\Delta\tau_{\min} = \min\{|X_j|/S : j \in \{1,...,n\}\}$, $\Delta\tau_{\max} = \max\{|X_j|/S : j \in \{1,...,n\}$, Then, $\tau_{\mathbf{D}} = [\Delta\tau_{\min}, \Delta\tau_{\max}]$.
2) **Determine the minimum number of segments**: $S' = \min\{\lceil |Y'|/\Delta\tau\rceil : \Delta\tau \in \tau_{\mathbf{D}}\}$, which ensures different guesses of $\Delta\tau$ will be evaluated with the same number of segments, so that the likelihoods computed in step 3 will be comparable.
3) **Evaluate the likelihood**:

$$\Delta\tau^* = \arg\max_{\Delta\tau\in\tau_{\mathbf{D}}}\prod_{i=1}^{d}\prod_{l=1}^{L}\prod_{s=1}^{S'}\frac{(\lambda_{i,s,l}\Delta\tau)^{m_{s,l}^i(\Delta\tau)}}{e^{\lambda_{i,s,l}\Delta\tau}}$$

In this case, intensity parameter λ will depend on both time and mark. In this paper, we assume all feature dimensions have been normalized to $[0,1]$ respectively, which results in the mark space within $[0,1]$. We build a multi-level discretization of mark space by splitting it into L levels. Then the point process factors into L levels of independent processes operating in each level of the mark space for a particular event.

$$\begin{aligned}&\Pr(\{N_X, M_X^i\}) \\ &= \prod_{l=1}^{L}\prod_{s=1}^{S}\frac{(\lambda^i(s,l)\Delta t\Delta\tau)^{m_{s,l}^i}}{m_{s,l}^i!}e^{-\lambda^i(s,l)\Delta t\Delta\tau}\end{aligned} \tag{9}$$

Given training m.t.s. dataset $\mathbf{D}$, and multivariate marked point process representation $\ddot{\widetilde{\mathbf{N}}}$, the data likelihood can be computed as

$$\begin{aligned}\Pr(\ddot{\widetilde{\mathbf{N}}}|\mathbf{D}) &= \prod_{i=1}^{d}\Pr(\{N^i, M^i\}|\mathbf{D}) \\ &= \prod_{i=1}^{d}\prod_{l=1}^{L}\prod_{s=1}^{S}\frac{(\lambda^i(s,l,\mathbf{D})\Delta t\Delta\tau)^{m_{s,l}^i}}{m_{s,l}^i!}e^{-\lambda^i(s,l,\mathbf{D})\Delta t\Delta\tau}\end{aligned} \tag{10}$$

where the intensity function $\lambda^i(s,l,\mathbf{D})$ depends on the event, the time division (progress level), the mark space level and the training data. Now, we can formalize our two key steps in early classification.

1. Learning MD-MPP. Given n training samples, the maximum log-likelihood estimation of model parameters is:

$$\lambda^{i*}(s,l,\mathbf{D}) = \frac{\Sigma_{j=1}^{n} m_{j,s,l}^i}{\Sigma_{j=1}^{n}\Delta t\Delta\tau_j}. \tag{11}$$

where $m_{j,s,l}^i$ is the number of landmarks of event x^i in j-th training sample's s-th time division, l-th level of mark space.

2. Early Classification. Given an ongoing testing m.t.s Y', and a trained model $\mathbf{\Lambda} = \{\lambda_{i,s,l}|L,S,\mathbf{D}\}$ (for simplicity, we use $\lambda_{i,s,l}$ to represent $\lambda^{i*}(s,l,\mathbf{D})$). First, we construct a structure of Y' by factoring it over time line and mark space in the same way as trained model, so that dynamics can be matched. Then, the likelihood of Y' is:

$$\Pr(Y'|\mathbf{\Lambda}) \propto \prod_{i=1}^{d}\prod_{l=1}^{L}\prod_{s=1}^{\lceil p^*S\rceil}(\lambda_{i,s,l}\Delta\tau^*)^{m_{s,l}^i}e^{-\lambda_{i,s,l}\Delta\tau^*} \tag{12}$$

where $p^* = |Y'|/(\Delta\tau^* S)$ is our best guessed progress level of Y'. Since the length of m.t.s. can be different, given an ongoing testing m.t.s., we may not know when it will be finished. Therefore, we need to 'guess' the 'right' progress level of it first. Then we can apply our model appropriately. This is an important merit of our approach. Algorithm 1 shows the detail of how we compute p^*.

*B. Sequential Cue (**Relax Assumption 3**).*

Although MD-MPP provides a good modeling of temporal dynamics for m.t.s., the unrealistic independency assumption between events (Assumption 3) is not relaxed yet. In real applications, m.t.s. always has strong correlations among variables. For instance, in the execution of a particular human action, a few joints will change their angles immediately after other few joints rotated to some degree according to the underlying cognitive "recipe" of that action. The identification of temporal correlations among events allows us to utilize these sequential patterns for early classification, which improves the reasoning capability of our model. As illustrated in Fig. 2, if we only

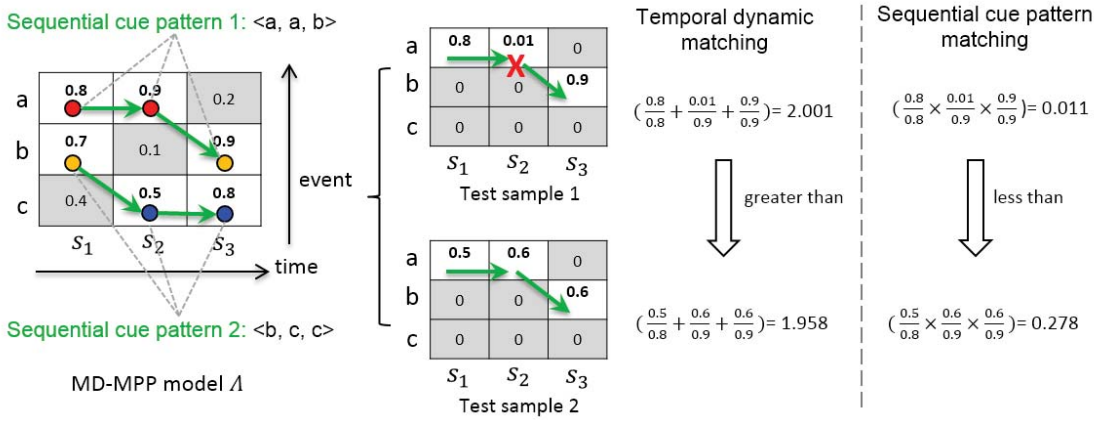


Fig. 2. Illustrations of Temporal Dynamics and Sequential Cue, based on our MD-MPP representation. Numbers in the table constitutes the model parameter, where each value indicates the firing rate of a certain event within a particular time division. Based on a trained model Λ, we can sample event streams according to these firing rates. For example, event stream $\langle a, a, b\rangle$ and $\langle b, c, c\rangle$ will have more chance to be sampled. With sufficient sampling, these type of Sequential Cue patterns can be mined by using Algorithm 2.

consider temporal dynamics, test sample 1 will have a better match with the point process model. But in terms of sequential cue patterns, test sample 2 results in a significantly better match. As a complement of MD-MPP model, we introduce the *sequential cue* component of our approach.

Algorithm 2 Construction of Sequential Cue model $\mathbf{\Phi}$ (O-order bounded PST)

1) **Sampling event streams set**: Let $\overline{\mathbf{D}}_{\mathbf{\Lambda}} = \{\bar{a}_1, ..., \bar{a}_v\}$ be the training set for learning model $\mathbf{\Phi}$.
2) **Creating candidate set U**: Assume h is a subsequence of $\bar{a}_r (r = 1, ..., v)$. If $|h| < O$ and $\Pr(h) > \eta$, then put h in $\mathbf{U}$. η is a user specified minimal probability requirement for an eligible candidate. $\Pr(h)$ is computed from frequency count.
3) **Testing every candidate h** $\in \mathbf{U}$: For any $h \in \mathbf{U}$, test following two conditions:
   - (1) $\Pr(e|h) \geq \alpha$, which means the context subsequence h is meaningful for some event e. Here, α is defined by user to threshold a conditional appearance.
   - (2) $\frac{\Pr(e|h)}{\Pr(e|\text{suf}(h))} \geq \beta$, or $\leq 1/\beta$, which means the context h provides extra information in predicting e relative to its longest suffix $\text{suf}(h)$. β is a user specified threshold to measure the difference.
   - **Then**, if h passes above two tests, add h and its suffixes into $\mathbf{U}$.
4) **Smoothing the probability to obtain** $\phi(e|h)$:
   For each h in $\mathbf{U}$, if $\Pr(e|h) = 0$, we assign a minimum probability γ. In general, the *next event probability function* can be written as:
   $\phi(e|h) = (1 - |\mathbf{E}|\gamma)\Pr(e|h) + \gamma$. Here, γ is the smoothing factor defined empirically.

Our basic idea is to formalize the notion of *sequential cue* by discovering sequential structure that comes from the order in which events (m.t.s. variables) occur over time. So we need to generate representative discrete event streams by quantizing continuous m.t.s. observations. Since MD-MPP characterizes the rate of occurrence (intensity function) of events at different time division (segments), then we can easily sample event from each segment according to this rate, which results in a discrete event stream. If we generate sufficient number of sample sequences, then the sequential cue patterns will be preserved in the sampling set. Fig. 2 gives an example showing how we sampled these representative sequences of events from our trained MD-MPP model. Then the task of finding temporal correlations among features (events) becomes a problem of mining sequential patterns.

Specifically, let $\mathbf{E} = \{e^i : i = 1, ..., d \times L\}$ be the set of events[2]. And $\overline{\mathbf{D}}_{\mathbf{\Lambda}} = \{\bar{a}_1, ..., \bar{a}_v\}$ consists of v times sampling according to $\mathbf{\Lambda}$. For instance, $\bar{a}_r = \{e_s^r\}_{s=1}^S, r \in \{1, ..., v\}$ is a sampled *event stream*, which means at the j-th segment, we sampled one event $e_s^r \in \mathbf{E}$. We can notice that $\bar{a}_i \in \mathbf{E}^*$, $|\bar{a}_i| = S$. Specific sampling probability of each event at a particular time (segment) can be computed according to:

$$\Pr_{\text{sample}}(\text{event} = e|\text{segment} = s) = \frac{\lambda_{e,s}}{\sum_{e' \in \mathbf{E}} \lambda_{e',s}} \tag{13}$$

Now the goal is to learn a model $\mathbf{\Phi} = \{\phi(e|h) : h \in \mathbf{E}^*, e \in \mathbf{E}\}$, which associates a history h with next possible event e. We call function $\phi(e|h)$ the *next event probability function*. If we define the *history* at j-th time segment of event stream $\bar{a}^i$ as the subsequence $h_j(\bar{a}^i) = \{e_j^i | j \leq S\}$, then the log-likelihood of event stream $\bar{a}^i$, given a Sequential Cue model $\mathbf{\Phi}$, can be written as:

$$\Pr(\bar{a}^i|\mathbf{\Phi}) = \sum_{j=1}^{S} \log \phi(e_j^i | h_{j-1}(\bar{a}^i)) \tag{14}$$

Given an ongoing testing m.t.s. Y', and a trained model $\mathbf{\Phi} = \{\phi(e|h)|\overline{\mathbf{D}}_{\mathbf{\Lambda}}\}$. First, we construct a structure of Y' by factoring it over time line and mark space. We use $\Delta\tau^*$ as the segment

[2]With multilevel-discretized representation, the total number of events becomes $d \times L$. The MD-MPP model can be rewritten as $\mathbf{\Lambda} = \{\lambda_{e,s} | e \in \mathbf{E}, s \in \{1, ..., S\}\}$ for convenience.

length of Y', then we have $S^* = \lceil |Y'|/\Delta\tau \rceil$ segments. Similar to the training process of $\mathbf{\Lambda}$, we can build a MD-MPP representation for Y' by itself, $\mathbf{\Lambda}_{\mathbf{Y}'} = \{\lambda'_{i,s,l}|L, S^*, Y'\}$, from which a set of w representative event streams of Y', $\overline{\mathbf{D}}_{\mathbf{\Lambda}_{Y'}} = \{\bar{b}_1, ..., \bar{b}_w\}$, can be sampled in the same way. Then, the likelihood of Y' is:

$$\Pr(Y'|\mathbf{\Phi}) \propto \sum_{i=1}^{w} \Pr(\bar{b}_i|\mathbf{\Phi})) \tag{15}$$

In terms of specific implementation of $\mathbf{\Phi}$, we adopt the Variable order Markov Model (VMM) [28], which is a category of algorithms for prediction of discrete sequences. It can capture both large and small order Markov dependencies extracted from $\overline{\mathbf{D}}_{\mathbf{\Lambda}}$. Therefore, it can encode richer and more flexible Sequential Cue. This can be done efficiently by constructing a probability suffix tree (PST) [29], [30], a fast implementation algorithm of VMM. Algorithm 2 shows the detail of this process.

C. Final Early Classification Function

Giving an ongoing m.t.s. Y', we can now construct our final early classification function $\mathcal{F}(Y')$ by using the knowledge learned from Section IV-A and IV-B, namely *time dynamics* model $\Pr(Y'|\mathbf{\Lambda})$ and *sequential cue* model $\Pr(Y'|\mathbf{\Phi})$. We use MD-MPP to denote the first model, and MD-MPP+TD to denote the second model. The early classification performances are evaluated on both of the two models. For a domain application with $|\mathbf{C}|$ classes, our final prediction functions of two models can be written as:

$$\text{MD-MPP:} \quad \mathcal{F}(Y') = \arg\max_{c \in \mathbf{C}} \{\Pr^c(Y'|\mathbf{\Lambda})\};$$
$$\text{MD-MPP+TD:} \quad \mathcal{F}(Y') = \arg\max_{c \in \mathbf{C}} \{\Pr^c(Y'|\mathbf{\Phi})\}.$$

V. Experimental Studies

In this section, we present a comprehensive evaluation of our methods (MD-MPP and MD-MPP+TD) on modeling accuracy and time efficiency using three real-world data sets. We have compared with the state-of-the-art methods including 1NN+DTW [17], ECTS [1], MSD [3] and HMM[3]. Table II summarizes the baselines.

A. Data Sets

We utilized three real world datasets: CMU human motion capture dataset [31], UCI Australian Sign Language (Auslan) dataset [32], and freeway occupancy dataset (PEMS-SF) [32]. The following details the collection and preprocessing of the three datasets.

Human Action Data: The dataset was composed of dozens of synchronized motion capture actions performed by more than one hundred subjects. In our experiment, we select the MoCap data of 9 common action classes performed by different subjects, which consists of 10 samples per class on average (total 91 samples) with average duration of 839 frames. The 9 action classes include *run*, *pick-up*, *walk*, *jump*, *sitting on a motorcycle*, *boxing*, *cartwheel*, *chicken dance* and *golf swing*. The human body model consists of 34 bones with hierarchical structures. Each action is specified by m.t.s. observations on motion angles of body bones, which describe both moving direction and magnitude of joints, as well as the dynamic relationships between bones. Fig. 3 shows the human body model with above mentioned hierarchical structure. The original full body Degree of Freedoms (DOFs) are 62. We discard some unimportant joint angles, such as fingers, toes, thumb in the experiments. Finally, we select 19 body joints which cover the DoFs of radius, humerus, tibia, femur and the upper back.

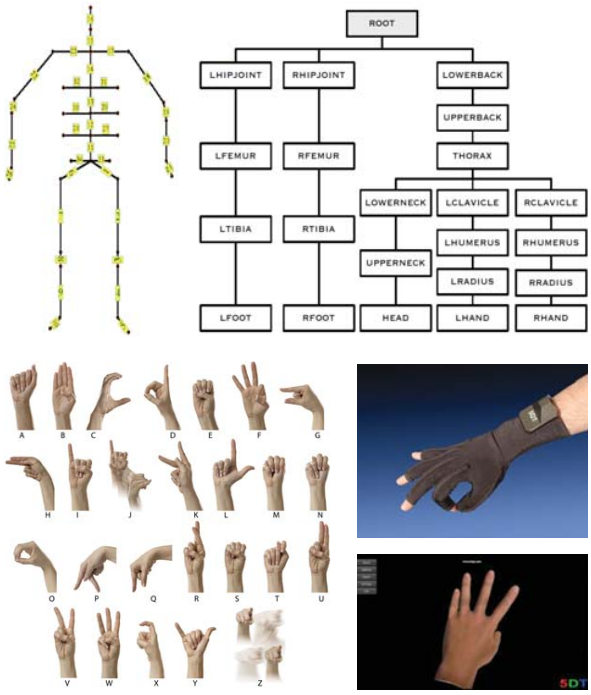
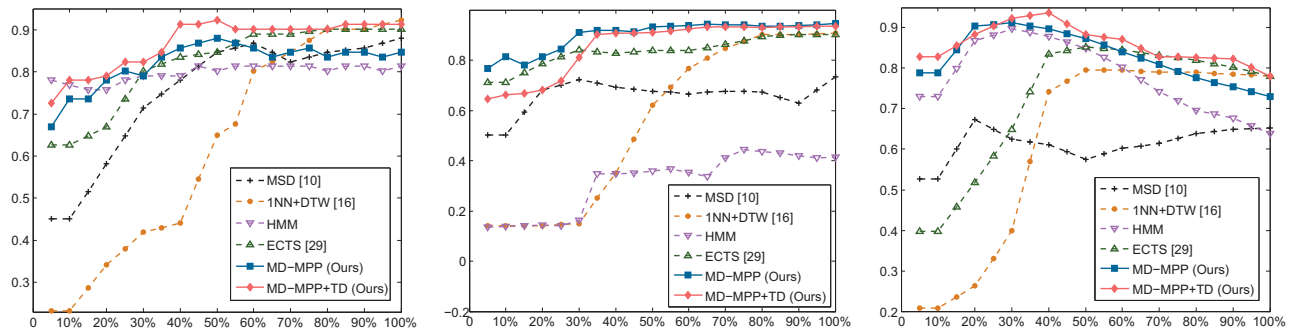


Fig. 3. Evaluation datasets. Top: CMU pose. Bottom: UCI sign language.

Sign Language Data: This dataset was composed of sample of Auslan (Australian Sign Language) signs [33], [34], in which 27 samples of each of 95 Auslan signs (in total 2565 signs) were captured from a native signer using high-quality position trackers (two Fifth Dimension Technologies gloves). Each hand has 11 degrees of freedom (i.e. roll, pitch and yaw as well as x, y and z), which results in a total of 22 dimensional m.t.s. observations of signs, The average length of each sign is approximately 57 frames, where the refresh rate is close to 100 frames per second.

Freeway Occupancy Data: This data set was collected from the California Department of Transportation PEMS website, which describes the occupancy rate of hundreds of car lanes of San Francisco bay area freeways. The occupancy rates are normalized to $[0, 1]$. The measurements cover the period from Jan. 1st 2008 to Mar. 30th 2009. We consider each day in this database as a sample, which is a m.t.s. observation with dimension 963 (the number of sensor points on different locations in

[3]We used following public available toolbox as our HMM implementation: http://www.cs.ubc.ca/ murphyk/Software/HMM/hmm.html



(a) CMU human action dataset (b) PEMS freeway occupancy dataset (c) UCI sign language dataset

Fig. 4. Performance comparisons on three datasets (see text for detailed discussions). In each figure, the vertical axis is the classification accuracy averaged over all classes, and the horizontal axis is the observation ratio, which can be viewed as the normalized time line ($(0, T] \rightarrow (0, 1]$).

TABLE II
SUMMARY OF THE FOUR BASELINES USED FOR QUANTITATIVE COMPARISON WITH OUR ALGORITHM

Methods	Rationale	Description
One-nearest-neighbor DTW (1NN+DTW) [17]	The state-of-the-art time series classification algorithm.	The dynamic time warping (DTW) based distance measurements between test and training time series are computed for use in 1NN classifier. For m.t.s., the distance are measured as the average of component u.t.s. distances.
Early Classification on Time Series (ECTS) [1]	An extension of 1NN classifier to achieve early classification.	The MPL (Minimum Prediction Length) for a cluster of similar time series are computed first. At the testing phase, the learned MPLs are used to select the nearest neighbour from only "qualified" candidates in terms of MPL. For m.t.s., the distance are measured as the average of component u.t.s. distances.
Multivariate Shapelets Detection (MSD) [3]	An extension of time series shapelets to achieve early classification.	Multivariate shapelets are extracted using a sliding-window based strategy.These shapelets are then pruned according to the weighted information gain.
Hidden Markov Model (HMM)	An effective statistical model for temporal pattern recognition	The HMM is selected as a representative of generative model based methods. A model is trained for each class. Decisions are based on likelihoods ranking.

the highways). To reduce the computational burden, we select 20 dimensions m.t.s. as our observations. All the sensors have equal sampling rate (one observation every 10 minutes), so that all time series have equal length of $6 \times 24 = 144$. The task on this dataset is to identify each observed day as the correct day of the week, from Monday to Sunday, e.g. label it with an integer in $\{1, 2, 3, 4, 5, 6, 7\}$. In total, we have 440 samples from 7 classes (63 samples per class on average).

B. Performance Comparison

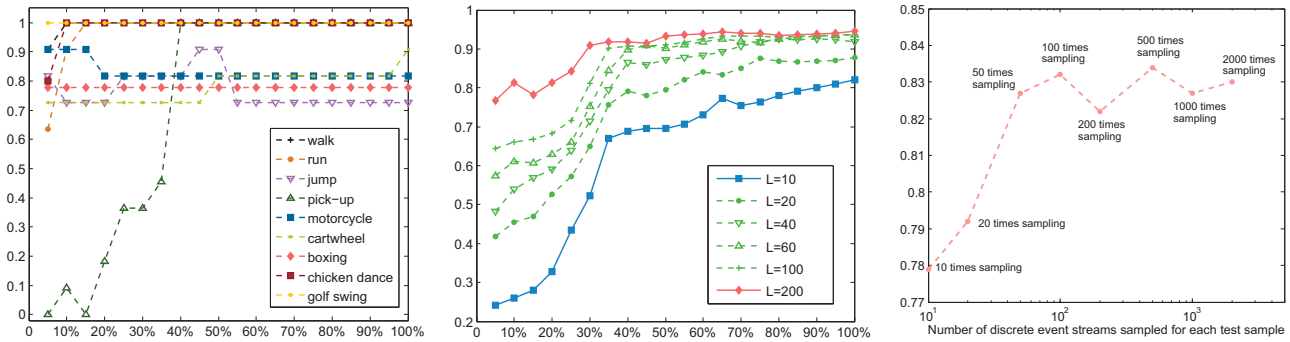
We compare our algorithms of m.t.s. early classification in Section IV (MD-MPP and MD-MPP+TD) with existing alternatives that we discussed in Section II and summarized in Table II. We evaluate the classification accuracy by using the standard "leave-one-out" method in all three datasets. Different from traditional classification task, for early classification, we focus on the predictive power of each method. An early classifier should use an observation ratio as small as possible to make an accurate prediction. To evaluate this, we do classification along the progress of time series, and predict the class label at different progress levels (observation ratio) of time series. Specifically, we use 5% of full m.t.s. duration as an interval of generating a new prediction result.

Model Construction. For the Human Action Data, we construct a MD-MPP model by splitting mark space into 10 levels ($L = 10$) and dividing the time line into 20 pieces of equal length segments ($S = 20$). To construct a MD-MPP+TD model, we train an order 3-bounded PST ($O = 3$) first, then do 100 times sampling ($w = 100$) of event streams for each m.t.s. at testing phase. For the Sign Language Data, we set $L = 20$, $S = 10$, $O = 3$, and $w = 100$. For the Freeway Occupancy Data, we set $L = 200$, $S = 40$, $O = 2$, and $w = 100$. Parameters in four baselines are tuned to be optimal.

Results. Fig. 4 summarizes the quantitative comparison between our methods and four baselines. These graphs help us make the following observations:

(1) Our algorithms significantly outperform all the compared methods in most cases, and achieve high prediction accuracy over different levels of observation ratio. In terms of full-length classification (at observation ratio 100%), 1NN-DTW is the most comparable one to ours, which demonstrates its robustness as the state-of-the-art method for time series classification. At early stages of observation ($< 30\%$), MSD and ECTS can outperform 1NN-DTW to accomplish better early classification due to their designs on utilizing early cues. As a latent state model, HMM is relatively less dependent on full length observation. Table V-B shows detailed comparisons of six methods on three datasets.

(2) Each domain has different predictability, which means the discriminative segments of m.t.s. may appear at different stages of time series. As illustrated in Fig. 4, we achieved near-



(a) Performance on different actions (CMU) (b) Model parameter: sampling event streams (c) Model parameter: mark-space levels

Fig. 5. Detailed results on Human Action dataset (sub-figure a) and model parameter analysis (sub-figure b and c).

TABLE III
PERFORMANCE COMPARISONS ON THREE DATASETS (PERCENTAGE AS OBSERVATION RATIOS).

Methods	CMU pose dataset					PEMS dataset					UCI signLanguage dataset				
	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
1NN+DTW [17]	0.342	0.441	0.802	0.901	**0.923**	0.141	0.350	0.766	0.902	0.907	0.263	0.742	0.796	0.790	0.781
ECTS [1]	0.670	0.835	0.890	0.901	0.901	0.786	0.827	0.839	0.893	0.902	0.518	0.835	0.846	0.820	0.781
MSD [3]	0.582	0.780	0.868	0.846	0.879	0.682	0.693	0.664	0.673	0.734	0.674	0.612	0.603	0.639	0.653
HMM	0.758	0.791	0.813	0.802	0.813	0.143	0.348	0.352	0.436	0.414	0.868	0.879	0.802	0.696	0.638
MD-MPP (Ours)	0.780	0.857	0.857	0.835	0.846	**0.814**	**0.918**	**0.939**	**0.934**	**0.945**	**0.904**	0.897	0.841	0.777	0.731
MD-MPP+TD (Ours)	**0.791**	**0.912**	**0.901**	**0.901**	0.912	0.682	0.907	0.925	0.930	0.936	0.884	**0.937**	**0.871**	**0.827**	**0.781**

optimal classification accuracy at the observation ratio of 40% in the Human Action Data, 30% in the Freeway Occupancy Data, and 20% in the Sign Language Data respectively. Fig 6 shows the corresponding detailed results in confusion matrices respectively. Interestingly, as indicated in Fig. 4 (c), with the increasing of information observed, the prediction accuracy is not necessarily go up, which means more noise is introduced at the late stages of m.t.s.. It is probably because that different signs end in the same way, such as open palms or fists.

(3) A few interesting observations that are reasonably in accordance with our domain knowledge. First, in Fig. 5 (a), we present detailed performance of our approach over 9 different action classes in the Human Action dataset. The action "pick-up" is difficult to be recognized at early stages, because it is executed by first walking to the object, then picking up it. The component sub-action "walking to object" makes it confusing with the class "walk". Another component sub-action "crouching to pick up object" makes it confusing with the class "jump". Secondly, in Fig. 4 (b), we can see an abrupt growth at observation ratio around 30% to 40%, which exactly corresponds to the traffic hours in the morning from 7:00 am to 9:30 am, where the discriminative information lies in.

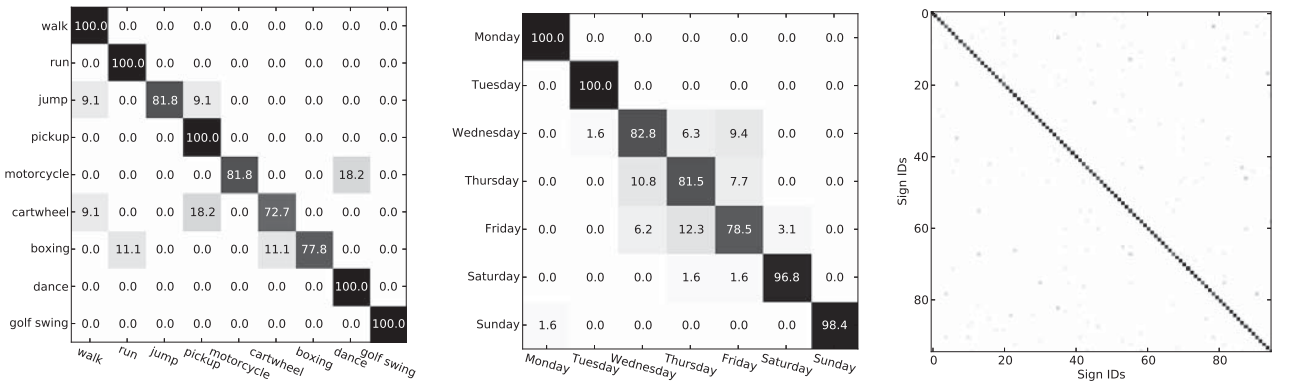
(4) Sequential cue patterns are important for some domains, as shown in Action and Sign Language datasets (Fig. 4 (a), Fig. 4 (c)), but less useful for some other domains, such as Freeway Occupancy data (Fig. 4 (b)). This is because variables/events have strong correlations (for example, bones connected to each other) in Action and Sign Language datasets. But, variables, such as sensor points at different locations of transportation system, may have very little chance to be dependent on each other, when the system is huge.

C. Model Parameters

To show the impact of model parameters to the results, we present Fig. 5 (b) and Fig. 5 (c) as illustrations of two key parameters in our approach: one is the number of mark space levels L; another one is the sampled event stream number w. Fig. 5 (b) shows the trend of performance improvement with increasing number of mark space levels in PEMS dataset, which suggests that this dataset prefers a more detailed discretization of mark space. Fig. 5 (c) proved our claim that "sufficient" number of sampling of discrete event streams will preserve most of the sequential pattern information. Also, the "sufficient" times is not necessarily a very big number. As shown in the figure, a relatively small number (100) of sampling can already achieve near-optimal performance (results from Human Action Data).

VI. CONCLUSION

In this paper, we propose a novel approach to early classify multivariate time series (m.t.s.) data by modeling two types of time pattern: *temporal dynamics* and *Sequential Cue*. The major contributions include a Multilevel-Discretized Marked Point Process (MD-MPP) model for representing m.t.s.; and a sequential cue model (MD-MPP+TD) to characterize the sequential patterns among multiple time series variables. We have empirically shown that our approach is superior in the early classification task of m.t.s., in terms of both accuracy and time efficiency. Our approach does not assume that all the data have the same length of duration, but it relies on the



(a) CMU dataset (at **40%** observation ratio) (b) PEMS dataset (at **30%** observation ratio) (c) UCI dataset (at **20%** observation ratio)

Fig. 6. Confusion matrices on three datasets.

segments of different progress levels to be roughly matched among samples in the same class. Future work will extend this model to domain dependent density functions.

ACKNOWLEDGMENT

This research is supported in part by the NSF CNS award 1314484 and 1449266, ONR award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, U.S. Army Research Office Young Investigator Award W911NF-14-1-0218, and Air Force Office of Scientific Research award FA9550-12-1-0201.

REFERENCES

[1] Z. Xing, J. Pei, and P. S. Yu, "Early prediction on time series: a nearest neighbor approach," in *IJCAI*, 2009, pp. 1297–1302.

[2] Z. Xing, J. Pei, P. Yu, and K. Wang, "Extracting interpretable features for early classification on time series," *SDM*, 2011.

[3] M. Ghalwash and Z. Obradovic, "Early classification of multivariate temporal observations by extraction of interpretable shapelets," *BMC Bioinformatics*, vol. 13, no. 1, p. 195, 2012.

[4] T. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.

[5] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: a survey and empirical demonstration," in *ACM SIGKDD*, 2002, pp. 102–111.

[6] Z. Zhang, J. Cheng, J. Li, W. Bian, and D. Tao, "Segment-based features for time series classification," *The Computer Journal*, vol. 55, no. 9, pp. 1088–1102, 2012.

[7] L. Ye and E. Keogh, "Time series shapelets: a new primitive for data mining," in *ACM SIGKDD*, 2009, pp. 947–956.

[8] L. Wei and E. Keogh, "Semi-supervised time series classification," in *ACM SIGKDD*, 2006, pp. 748–753.

[9] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *ICML*, 2006, pp. 1033–1040.

[10] V. Eruhimov, V. Martyanov, and E. Tuv, "Constructing high dimensional feature space for time series classification," in *Knowledge Discovery in Databases: PKDD*, 2007, pp. 414–421.

[11] H. Katagiri, I. Nishizaki, T. Hayashida, and T. Kadoma, "Multiobjective evolutionary optimization of training and topology of recurrent neural networks for time-series prediction," *The Computer Journal*, vol. 55, no. 3, pp. 325–336, 2012.

[12] P. Geurts, "Pattern extraction for time series classification," in *Principles of Data Mining and Knowledge Discovery*, 2001, pp. 115–127.

[13] B. Hu, Y. Chen, J. Zakaria, L. Ulanova, and E. J. Keogh, "Classification of multi-dimensional streaming time series by weighting each classifier's track record," in *ICDM*, 2013, pp. 281–290.

[14] D. F. Silva, V. M. A. de Souza, and G. E. A. P. A. Batista, "Time series classification using compression distance of recurrence plots," in *ICDM*, 2013, pp. 687–696.

[15] S. Ando and E. Suzuki, "Time-sensitive classification of behavioral data," in *SDM*, 2013, pp. 458–466.

[16] C. Myers and L. Rabiner, "A comparative study of several dynamic time-warping algorithms for connected word recognition," *The Bell System Technical Journal*, vol. 60, no. 7, pp. 1389–1409, 1981.

[17] E. Keogh, "Exact indexing of dynamic time warping," in *Proceedings of the 28th International Conference on Very Large Data Bases*. VLDB Endowment, 2002, pp. 406–417.

[18] I. Batal, L. Sacchi, R. Bellazzi, and M. Hauskrecht, "Multivariate time series classification with temporal abstractions," in *Florida Artificial Intelligence Research Society Conference*, 2009.

[19] M. W. Kadous and C. Sammut, "Classification of multivariate time series and structured data using constructive induction," *Machine Learning*, vol. 58, no. 2, pp. 179–216, 2005.

[20] G. He, Y. Duan, T. Qian, and X. Chen, "Early prediction on imbalanced multivariate time series," in *CIKM*, 2013, pp. 1889–1892.

[21] A. Bregón, M. A. Simón, J. J. Rodríguez, C. Alonso, B. Pulido, and I. Moro, "Early fault classification in dynamic systems using case-based reasoning," in *Current Topics in Artificial Intelligence*, 2006, pp. 211–220.

[22] J. J. Rodríguez, C. J. Alonso, and H. Boström, "Boosting interval based literals," *Intelligent Data Analysis*, vol. 5, no. 3, pp. 245–262, 2001.

[23] A. Jansen and P. Niyogi, "Point process models for event-based speech recognition," *Speech Communication*, vol. 51, no. 12, pp. 1155–1168, 2009.

[24] A. Gunawardana, C. Meek, and P. Xu, "A model for temporal dependencies in event streams," in *NIPS*, 2011, pp. 1962–1970.

[25] G. Kim, L. Fei-Fei, and E. P. Xing, "Web image prediction using multivariate point processes," in *ACM SIGKDD*, 2012, pp. 1068–1076.

[26] K. Prabhakar, S. Oh, and P. Wang, "Temporal causality for the analysis of visual events," in *CVPR*, 2010, pp. 1967–1974.

[27] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes. Vol. I.* Springer-Verlag, 2003, vol. 1.

[28] R. Begleiter, R. El-Yaniv, and G. Yona, "On prediction using variable order markov models," *J. Artificial Intelligence Research*, vol. 22, pp. 385–421, 2004.

[29] D. Ron, Y. Singer, and N. Tishby, "The power of amnesia: Learning probabilistic automata with variable memory length," *Machine Learning*, vol. 25, no. 2, pp. 117–149, 1996.

[30] K. Li, J. Hu, and Y. Fu, "Modeling complex temporal composition of actionlets for activity prediction," in *ECCV*, 2012, pp. 286–299.

[31] CMU, "http://mocap.cs.cmu.edu."

[32] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[33] A. Ibarguren, I. Maurtua, and B. Sierra, "Layered architecture for real-time sign recognition," *The Computer Journal*, vol. 53, no. 8, pp. 1169–1183, 2010.

[34] J. F. Lichtenauer, E. A. Hendriks, and M. Reinders, "Sign language recognition by combining statistical dtw and independent classification," *IEEE TPAMI*, vol. 30, no. 11, pp. 2040–2046, 2008.