Mining Complex Time-Series Data by Learning Markovian Models

Yi Wang, Lizhu Zhou, Jianhua Feng and Jianyong Wang
Department of Computer Science,
Tsinghua University, Beijing, China, 100084
yi.wang.2005@gmail.com

Zhi-Qiang Liu
School of Creative Media,
City University of Hong Kong,
Tachee Avenue, Kowloon, Hong Kong

Abstract

In this paper, we propose a novel and general approach for time-series data mining. As an alternative to traditional ways of designing specific algorithm to mine certain kind of pattern directly from the data, our approach extracts the temporal structure of the time-series data by learning Markovian models, and then uses well established methods to efficiently mine a wide variety of patterns from the topology graph of the learned models. We consolidate the approach by explaining the use of some well-known Markovian models on mining several kinds of patterns.

We then present a novel high-order hidden Markov model, the variable-length hidden Markov model (VLHMM), which combines the advantages of well-known Markovian models and has the superiority in both efficiency and accuracy. Therefore, it can mine a much wider variety of patterns than each of prior Markovian models. We demonstrate the power of VLHMM by mining four kinds of interesting patterns from 3D motion capture data, which is typical for the high-dimensionality and complex dynamics.

1. Introduction

Time-series data have been widely considered rich in valuable patterns [7]. Many initiative and creative works have been proposed to mine these patterns. Usually, one or more algorithms are designed to mine a certain kind of pattern. As an example, a series of works contribute to mine the periodic patterns: Han and et al. [6] concentrate on mining the *partial* periodic patterns. Sheng and et al. [13] improve this work by a more efficient algorithm. Ma and Hellerstein [8] and Elfeky and et al. [4] addresse on efficiently mining of periodic patterns with unknown periods. In particular, Ma and Hellerstein's algorithm explicitly takes the uncertainty caused by noise under consideration. Aref and et al. [1] propose mining periodic patterns from online and incremental data.

In addition to periodic pattern, most other patterns are also about the temporal structure of the time-series data. And various mining methods usually have to cope with some commonly existing difficulties, including high-dimensionality of multivarite data, uncertainty caused by noise or imprecise specification of mining objectives, and highly varied dynamics of the time-series. Therefore, we would like to propose a unified framework of mining various kinds of patterns from time-series data while coping with the common difficulties. In our framework, the first stage is to extract the temporal structure of the time-series data by learning Markovian models; then well established methods can be used to efficiently mine a wide variety of patterns from the learned temproal structure. In this framework, the above listed difficulties are mostly handled by the statistical learning stage, which, in particular, encodes the temporal structure by the topology graphs of Markovian models, and makes it convenient to use graph theory algorithms to mine from the topology graph.

In Section 2, we consolidate the framework by incorporating well-known Markovian models into it for mining several kinds of interesting patterns. Table 1 lists the comparison of applicabilities of these models. In Section 3, we present the *variable-length hidden Markov model* (VLHMM), which combines the advantages of the discussed Markovian models and achieves superiorities in both efficiency and accuracy. Thereby, VLHMM can mine a wider variety of patterns than other Markovian models. In Section 4, we demonstrate the power of VLHMM by mining periodic patterns, atomic patterns, patterns similar with given examples, and local dynamical complexity from 3D motion capture data, which is famous for high-dimensionality and complex dynamics.

2. Learning for Mining

In this section, we explain learning HMM, n-HMM, and VLMM for mining kinds of patterns. The discussion, to some extent insight into the mechanisms of these models, is the basis to explain VLHMM and its applications to mining

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

Table 1. Comparisons between the Markovian models on mining time-series data.

	Model multivariate data	Capture complex dynamics	Save training data	Mine periodic patterns	Search by example	Mine atomic patterns	Mine local dynamical complexity
HMM	✓	×	✓	✓	✓	×	×
n-HMM	✓	✓	×	✓	✓	×	×
VLMM	×	✓	✓	✓	×	✓	✓
VLHMM	✓	✓	✓	✓	✓	✓	✓

various kinds of patterns.

2.1. The Hidden Markov Model

The HMM [11] considers a discrete internal system that switches over a finite set of states $\mathcal{S} = [1, N]$. On each time clock t, the internal system changes its state from, q_{t-1}, to a new state, q_t, according to a transition probability mass function (pmf), $P(q_t \mid q_{t-1})$. At the same time, it emits an *observable*, o_t, according to an output probability density function (pdf), $p(o_t \mid q_t)$. Given a series of observables, $\boldsymbol{O} = \langle o_t \rangle_{t=1}^{T}$, learning HMM is to estimate the "invisible" transition pmfs and output pdfs, denoted by the model parameter set, $\boldsymbol{\Theta} = \{p(o \mid j), P(i \mid j)\}$, where $i, j \in \mathcal{S}$.

As a hidden model, HMM has the output pdfs that can emit continuous- or discrete, and vector- or scalar observables. This gives HMM the flexibility to model a large variety of time-series data. Because the value of q_t depends only on q_{t-1}, which is called *first-order Markovian assumption*, a learned HMM can be represented by a topology graph whose vertices are the states and each edge connecting state i and j is weighted by $P(i \mid j)$. Figure 1(a) shows the topology graph of an example HMM with 3 hidden states. The output pdf $p(o \mid i)$ defined on each state i summarizes o's that have the same distribution, so each state represents a typical group of o's and the edges represent probabilistic transitions between the typical groups.

Given the topology graph learned from a given time-series, mining (partially) periodic patterns turns simply into finding circles in the graph. Detecting circles in a graph can be achieved in polynomial time [5], but enumerating all circles is NP-hard [3]. However, various efficient approximation algorithms can be employed. For example, the Dijkstra's algorithm [3] is able to find the shortest circle passing through a given vertex. We can also use the polynomial-time algorithms [10] and [14] to find the graph for *strongly connected component*s (SCC), which contains at least a circle. The divide-and-conquer strategy taht detects circles in each SCC often reduces the consumption remarkably.

Given an example sequence of observables with arbitrary length, we can use the Viterbi algorithm [3] [11] to find a path of states that are the the most likely one that had generated the sequence. Using the Viterbi algorithm, we can mine all segments of the training sequence, $\boldsymbol{O}$, that are similar with a given example sequence $\hat{\boldsymbol{O}}$ by: (1) learning an HMM $\boldsymbol{\Theta}$ from $\boldsymbol{O}$; (2) using the Viterbi algorithm to map $\hat{\boldsymbol{O}}$ to paths, say one of them, $\hat{\boldsymbol{Q}}$, in the topology graph of $\boldsymbol{\Theta}$; (3) mapping $\boldsymbol{O}$ to a path $\boldsymbol{Q}$ in the topology graph; (4) locating all sub-strings, say $\hat{\boldsymbol{Q}}$, in $\boldsymbol{Q}$. As $\boldsymbol{Q}$ and $\boldsymbol{O}$ has the same length, the locations of $\hat{\boldsymbol{Q}}$ had appeared in $\boldsymbol{Q}$ correspond to segments of $\boldsymbol{O}$ that are similar with $\hat{\boldsymbol{O}}$.

2.2. The Fixed-Length n-th Order HMM

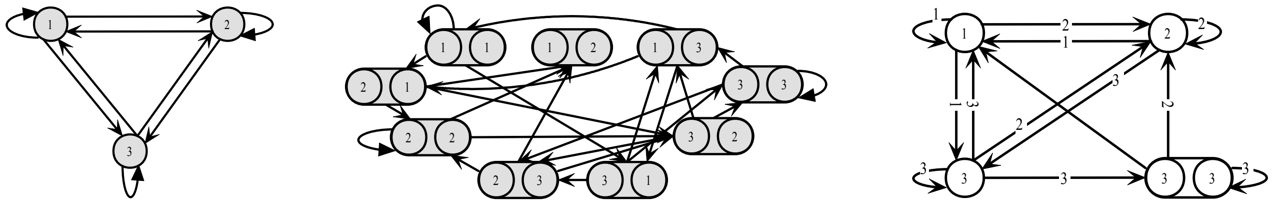
A known problem of HMM is that the first-order Markovian assumption restricts it from accurately modeling time-series data with highly varied dynamics, as it is often when the value of q_t depend not only on q_{t-1} but also $q_{t-2}, q_{t-3}, \ldots$. To increase the accuracy of modeling as well as the accuracy of mining, we can alternatively use the fixed-length n-th order HMM (n-HMM) [11] [9], which captures complex dynamics more accurately by modeling n-th order transition pmfs, $P(q_t \mid q_{t-1}, \ldots, q_{t-n})$, and output pdfs, $p(o_{t-1} \mid q_{t-1}, \ldots, q_{t-n})$. The sequence $\langle q_{t-i} \rangle_{i=1}^{n}$ is called a *context*. For an n-HMM with N hidden states, because $q_t \in \mathcal{S} = [1, N]$, there would be N^n contexts. HMM is a specialization of n-HMM with $n = 1$.

The vertices of the topology graph of a learned n-HMM correspond to the contexts, and the weight edges correspond to the probabilistic transitions between contexts. An example of a 2-HMM is shown in Figure 1(b). The edge from the node $(2, 1)$ to $(1, 2)$ denotes the transition from the context $\langle q_{t-1} = 2, q_{t-2} = 1 \rangle$ to the new state $q_t = 1$ and forms the new context $\langle q_t = 1, q_{t-1} = 2 \rangle$.

Similar with HMM, n-HMM can also be used to mine periodic patterns and example-matched patterns but with more accuracy. However, unbiased learning, or sufficient statistics, of the N^n output pdfs often requires a large number of training data and training time. For large n, the cost may be intractable.

2.3. The Variable-length Markov Model

The VLMM [12] is not a "hidden" model, it does not have output pdfs and can only model sequences of discrete



(a) Topology graph of an HMM (b) Topology graph of a 2-HMM (c) Topology graph of a VLMM/VLHMM

Figure 1. The topology graphs of HMM, n-HMM and VLMM/VLHMM

values. But it attracts us since it learns a set of shortest contexts with variable-length. This is because VLMM learns longer contexts to model parts of the training sequence with more complex dynamics, and vice versa. The minimum length of each context ensures the minimum number of total learned contexts [2].

The topology graph of VLMM also have vertices corresponding to contexts and edges weighted by transition probabilities. Figure 1(d) shows an example. Suppose a transition from context $\langle q_{t-1} = 3, q_{t-2} = 3\rangle$ to a new state $q_t = 2$. If the new context $\langle q_t = 2, q_{t-1} = 3, q_{t-2} = 3\rangle$ is not in the set of learned context, but $\langle q_t = 2\rangle$ is in, as shown in Figure 1(d), then it is to say that $\langle q_t = 2\rangle$ has been long enough for accurate determining of the next state. Therefore the transition is represented by the edge connecting $(3, 3)$ with (2).

As each context has the minimum length to accurately determine the next one, it statistically represents some *atomic* dynamics within the training sequence. It useful to discover such atomic patterns. For example, the atomic patterns of a Web log may convey important basic browsing habits of the users.

We also make use of the fact that the length of VLMM context is proportional to the complexity of local dynamics. The change of local dynamical complexity of the training sequence, $\boldsymbol{O} = \{o_t\}_{t=1}^{T}$, over time can be represented by the 2D curve, $L = \{l_t\}_{t=1}^{T}$, where l_t is the length of the context at time t.

3. The Variable-Length Hidden Markov Model

The VLHMM is a novel hidden Markovian model which borrows the idea from VLMM to learn a set of minimum number of contexts with minimum- and variable-length. This brings VLHMM much less output pdfs and transition pmfs than n-HMM, and thus can be learned much more efficiently. Compared with HMM, VLHMM has appropriately longer contexts that account for more accurate modeling. The VLHMM should not be considered a trivial extension to the VLMM, because the former is a hidden model and is learned by iteratively optimizing multiple objectives, Minimum-Entropy and Maximum-Likelihood, whereas the latter one has no output pdfs and is learned by minimizing the entropy alone. VLHMM is also intrinsicly different with HMM and n-HMM as its parameter set contains not only the output pdfs and transition pmfs but also the context set that are learned from the data.

The parameter set of VLHMM, $\boldsymbol{\Lambda} = \{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$, includes the contexts set, $\mathcal{C} = \{\boldsymbol{c}_l\}_{l=1}^{C}$, the transition pmfs, $\mathcal{A} = \{a(q; \boldsymbol{c}_l)\}_{l=1}^{C}$, where $a(q; \boldsymbol{c}_l) = P(q \mid \boldsymbol{c}_l)$ and $q \in \mathcal{S} = [1, N]$, and the output pdfs, $\mathcal{B} = \{b(o; \boldsymbol{c}_l)\}_{l=1}^{C}$, where $b(o; \boldsymbol{c}_l) = p(o \mid \boldsymbol{c}_l)$. In this paper, we model $b(o; \boldsymbol{c}_l)$ by the Gaussian distribution, which is parameterized by a mean vector, $\boldsymbol{\mu}_l$, and a covariance matrix $\boldsymbol{\Sigma}_l$.

The primary objective of learning is to estimate an optimal $\boldsymbol{\Lambda}^*$ that maximizes the likelihood with the given data, $\boldsymbol{O} = \langle o_t\rangle_{t=1}^{T}$:

$$\boldsymbol{\Lambda}^* = \underset{\boldsymbol{\Lambda}}{\operatorname{argmax}}\, P(\boldsymbol{O} \mid \boldsymbol{\Lambda}) \ . \tag{1}$$

For each o_t, we use the variable $\boldsymbol{m}_t \in \mathcal{C}$ to denote the context that generates o_t, so the likelihood function can be written in an iterative form:

$$Q(\boldsymbol{\Lambda} \mid \hat{\boldsymbol{\Lambda}}) = \sum_{t} \sum_{\substack{i\in\mathcal{S} \\ \boldsymbol{m}_t = c(\langle i, \boldsymbol{m}_{t-1}\rangle)}} P(\boldsymbol{m}_t \mid \boldsymbol{O}, \hat{\boldsymbol{\Lambda}}) \cdot \Bigg[\log b(\boldsymbol{o}_t; \boldsymbol{m}_t) + \sum_{\substack{i\in\mathcal{S} \\ \boldsymbol{m}_t = c(\langle i, \boldsymbol{m}_{t-1}\rangle)}} P(\boldsymbol{m}_{t-1} \mid \boldsymbol{O}, \hat{\boldsymbol{\Lambda}}) \cdot \log a(\boldsymbol{m}_t; \boldsymbol{m}_{t-1}) \Bigg], \tag{2}$$

where $\hat{\boldsymbol{\Lambda}}$ is the previous estimate of the model parameters, and the function call $\boldsymbol{m}_{t+1} = c(\langle i, \boldsymbol{m}_t\rangle)$ denotes the operation that adds the new hidden state $i \in \mathcal{S}$ to the current context $\boldsymbol{m}_t$ to form the new context $\langle i, \boldsymbol{m}_t\rangle$, and finds a context $\boldsymbol{m}_{t+1} \in \mathcal{C}$ with length $|\boldsymbol{m}_{t+1}|$ that matches the $|\boldsymbol{m}_{t+1}|$ most recent states of $\langle i, \boldsymbol{m}_t\rangle$.

By considering $\boldsymbol{M} = \langle m_t\rangle_{t=1}^{T}$ as the hidden variables, we maximize $Q(\boldsymbol{\Lambda} \mid \hat{\boldsymbol{\Lambda}})$ by deriving an EM algorithm. This algorithm alternatively executes an E-step and a M-step until converge to $\boldsymbol{\Lambda}^*$.

E-step. Making use of $\boldsymbol{m}_{t+1} = c(\langle i, \boldsymbol{m}_t\rangle)$ to represent the high order transition between states into the first or-

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

IEEE COMPUTER SOCIETY

der transition between contexts, we designed a Forward-Backward algorithm to efficiently computes $\gamma_t(\boldsymbol{c}) = P(\boldsymbol{m}_t = \boldsymbol{c} \mid \boldsymbol{O}, \hat{\boldsymbol{\Lambda}})$. For context $\boldsymbol{c}$ we define and calculate the *forward-probability*,

$$\alpha_t(\boldsymbol{c}) = P\left(\langle o_u \rangle_{u=1}^{t},\ \langle q_u \rangle_{u=t-|\boldsymbol{c}|+1}^{t} = \boldsymbol{c} \,\middle|\, \hat{\boldsymbol{\Lambda}}\right) \ , \tag{3}$$

inductively by:

$$\begin{aligned} \alpha_n(\boldsymbol{c}) &= a(\boldsymbol{c}; \mathbf{start}) \cdot b(o_n; \boldsymbol{c}) \ , \\ \alpha_{t+1}(\boldsymbol{c}) &= \sum_{\substack{q \in \mathcal{S}, \boldsymbol{c}' \in \mathcal{C} \\ c(\langle q, \boldsymbol{c}' \rangle) = \boldsymbol{c}}} \alpha_t(\boldsymbol{c}') \cdot a(\boldsymbol{c}; \boldsymbol{c}') \cdot b(o_{t+1}; \boldsymbol{c}) \ . \end{aligned} \tag{4}$$

Similarly, we define the *backward probability*,

$$\beta_t(\boldsymbol{c}) = P\left(\langle o_u \rangle_{u=t+1}^{T} \,\middle|\, \langle q_u \rangle_{u=t-|x|+1}^{t} = \boldsymbol{c},\ \hat{\boldsymbol{\Lambda}}\right) \ ,$$

and compute it by:

$$\begin{aligned} \beta_T(\boldsymbol{c}) &= 1 \ , \\ \beta_t(\boldsymbol{c}) &= \sum_{\substack{q \in \mathcal{S}, \boldsymbol{c} \in \mathcal{C} \\ \boldsymbol{c}' = c(\boldsymbol{c}, q)}} a(\boldsymbol{c}'; \boldsymbol{c}) \cdot b(o_{t+1}; \boldsymbol{c}') \cdot \beta_{t+1}(\boldsymbol{c}') \ . \end{aligned} \tag{5}$$

Given $\alpha_t(\boldsymbol{c})$ and $\beta_t(\boldsymbol{c})$, $\gamma_t(\boldsymbol{c})$ can be computed as,

$$\gamma_t(\boldsymbol{c}) = \frac{\alpha_t(\boldsymbol{c}) \cdot \beta_t(\boldsymbol{c})}{P(O \mid \hat{\boldsymbol{\Lambda}})} = \frac{\alpha_t(\boldsymbol{c}) \cdot \beta_t(\boldsymbol{c})}{\sum_{\boldsymbol{c} \in \mathcal{C}} \alpha_T(\boldsymbol{c})}. \tag{6}$$

M-step. Solving the partial derivative equation $\partial Q / \partial \mathcal{B} = 0$, we get the updating rule for the Gaussian output pdfs

$$\mu_l = \frac{\sum_t \gamma_t(\boldsymbol{c}^{(l)}) \cdot o_t}{\sum_t \gamma_t(\boldsymbol{c}^{(l)})} \ , \tag{7}$$

$$\Sigma_l = \frac{\sum_t \gamma_t(\boldsymbol{c}^{(l)}) \cdot (o_t - \mu_l)(o_t - \mu_l)^T}{\sum_t \gamma_t(\boldsymbol{c}^{(l)})} \ . \tag{8}$$

Solving $\partial Q / \partial \mathcal{A} = 0$ and using the temporary variable,

$$\begin{aligned} \xi_t(\boldsymbol{c}, j) &= P\left(\langle q_u \rangle_{u=t-|\boldsymbol{c}|+1}^{t} = \boldsymbol{c},\ q_{t+1} = j \,\middle|\, \boldsymbol{O}, \boldsymbol{\Lambda}\right) \\ &= \frac{\alpha_t(\boldsymbol{c}) a(j; \boldsymbol{c}) b(o_{t+n}; \boldsymbol{c}') \beta_{t+1}(\boldsymbol{c}')}{P(O \mid \hat{\lambda})} \ , \end{aligned}$$

we get the updating rule for the transition pmfs,

$$a(j; \boldsymbol{c}) = \frac{\sum_t \xi_t(\boldsymbol{c}, j)}{\sum_t \gamma_t(\boldsymbol{c})} \ . \tag{9}$$

The context set $\mathcal{C}$ is updated by optimizing a Minimum-Entropy criterion: if a context, $\boldsymbol{c} = \langle j_k^{(l)} \rangle_{k=1}^{n}$, and its shortened version, $\boldsymbol{c}^- = \langle j_k^{(l)} \rangle_{k=1}^{n-1}$, have similar accuracy at determining the next hidden state and the emitted observable, we prefer the shorter one. A reasonable qualitative comparison of $\boldsymbol{c}$ and $\boldsymbol{c}^-$ is the Kullback-Leibler (KL) divergence between the transition pmf $P(q_t \mid \langle q_{t-i} \rangle_{i=1}^{n} = \boldsymbol{c})$ and $P(q_t \mid \langle q_{t-i} \rangle_{i=1}^{n-1} = \boldsymbol{c}^-)$,

$$\Delta \mathrm{H_a}(\boldsymbol{c}; \boldsymbol{c}^-) = P(\boldsymbol{c}) \sum_{q \in \mathcal{S}} \left(P(q \mid \boldsymbol{c}) \cdot \log \frac{P(q \mid \boldsymbol{c})}{P(q \mid \boldsymbol{c}^-)} \right) \ .$$

Similarly, the accuracy of using $\boldsymbol{c}$ and $\boldsymbol{c}^-$ at determining the current output can be compared by,

$$\Delta \mathrm{H_b}(\boldsymbol{c}; \boldsymbol{c}^-) = P(\boldsymbol{c}) \cdot \int P(o \mid \boldsymbol{c}) \cdot \log \frac{P(o \mid \boldsymbol{c})}{P(o \mid \boldsymbol{c}^-)} \, \mathrm{d}o \ .$$

The right-hand side terms of $\Delta \mathrm{H_a}$ and $\Delta \mathrm{H_a}$ can be calculated by,

$$\begin{aligned} P(\boldsymbol{c}) &= \sum_t \gamma_t(\boldsymbol{c}) \ , \\ P(q \mid \boldsymbol{c}) &= a(q \mid \boldsymbol{c}) \ , \quad P(q \mid \boldsymbol{c}^-) = a(q \mid \boldsymbol{c}^-) \ , \\ P(o \mid \boldsymbol{c}) &= b(o \mid \boldsymbol{c}) \ , \quad P(o \mid \boldsymbol{c}^-) = b(o \mid \boldsymbol{c}^-) \ , \end{aligned}$$

If $\Delta \mathrm{H_a} < \epsilon_a$ and $\Delta \mathrm{H_b} < \epsilon_b$, where ϵ_a and ϵ_b are user specified thresholds, $\boldsymbol{c}^-$ and $\boldsymbol{c}$ should be similarly accurate at capturing the local dynamics, and we use $\boldsymbol{c}^-$ for more efficient modeling. Otherwise we use $\boldsymbol{c}$. We desgined a tree pruning-growing algorithm to apply the Minimum-Entropy criterion for updating $\mathcal{C}$. Details about the algorithm can be found in our technical report [15].

4. Mining Motion Capture Data with VLHMM

Given a learned VLHMM, the mining algorithms and methods that we explained in Section 2.3 with HMM, n-HMM and VLMM can be used to mine all the kinds of patterns listed in Table 1. Because of the limit number of pages, readers are recommended to refer to the 3D visualization of the empirical results in our technical report [15].

In our experiments, we use a 10-minute modern dance motion captured from a professional dancer under 33.3 frame-per-second. Each frame records the 3D rotations of the 19 major joints of human body, parametrized by the exponential map. To capture detailed dynamics, we also compute the difference between adjacent frames, so the dimensionality is doubled to 120-dimensional. We then apply Principle Component Analysis (PCA) to reduce the dimensionality to 56. The learning algorithm of VLHMM is implemented in C++. It took about 2.4 hours to learn 1716 contexts from the motion. Figure 7 of [15] shows the distribution of the lengths of these contexts. The topology graph of the learned VLHMM can be accessed online at `http://dbgroup.cs.tsinghua.edu.cn/wangyi/motion-engine/mm.png`.

In mining periodic patterns, learning temproal structre by VLHMM would results in more accurate result than using HMM, because the first-order Markovian assumption of

IEEE
COMPUTER
SOCIETY

HMM judges whether a motion segment is loopble by if the ending frame is similar with the starting frame, whereas the VLHMM considers motion segments are loopable only if the ending moving dynamics, represented by a context, can transit to the starting movement smoothly, i.e, with large transition probability to the context representing the starting movement. Figure 3 of [15] shows three periodic patterns that we mined using VLHMM from the captured modern dance.

In mining the example-match patterns, the accuracy achived by VLHMM inspired us for two interesting applications. Figure 5 of [15] shows one of them — beautifying coarsely captured motion. In this experiment, the example motion has low frame-rate and unprofessional performance, but the new motion synthesized from the mined pattern has high frame-rate and professional performance as the training motion. Figure 6 of [15] shows another application of synthesizing fine-detailed motion (top) from an example (bottom) of only a few keyframes scratched manually by an artist. The frames in the synthesized motion have similar but different poses than corresponding keyframes in the example.

Mining atomic patterns with VLHMM is to some extent different with the method used with VLMM. Although the contexts of VLHMM also has the statistical property of atomic behaviors, they are not observable/frame sequences as the contexts of VLMM, so they cannot be visualized and presented directly. In order to infer the motion frames correspond to a context $\boldsymbol{c}$, we designed the following algorithm. Known that the previous context of $\boldsymbol{c}$ with the form of $\langle j_2, j_3, \ldots \rangle$, we can enumerate the context set to find a context $\boldsymbol{c}^-$ in the form of $\langle j_2, j_3, \ldots \rangle$ that maximizes the transition probability $a(\boldsymbol{c}; \boldsymbol{c}^-)$. Then we append the prototypical pose represented by $\boldsymbol{c}^-$, i.e., the mean vector $\boldsymbol{\mu}_{\boldsymbol{c}^-}$ of $b(o; \boldsymbol{c}^-)$, to $\boldsymbol{\mu}_{\boldsymbol{c}}$. Repeating this process for $|\boldsymbol{c}|$ times, we can back-trace the motion frames as the atomic pattern represented by context $\boldsymbol{c}$.

Using $\gamma_t(\boldsymbol{c})$, we can compute the expected length of the contexts that generates the motion frame o_t as $l(t) = \sum_{t=1}^{T} \gamma_t(\boldsymbol{c}) \cdot |\boldsymbol{c}|$, where $|\boldsymbol{c}|$ is the length of the context $\boldsymbol{c}$. Visualizing the 2D curve $l(t)$ along t provides an intuitively and concise overview of the training motion, which, itself is a high-dimensional multivariate curve and is hard to be intuitively visualized. Figure 7 of [15] shows the curve $l(t)$ mined from the captured modern dance. Parts of the curve varied significantly correspond to motion segments with complex dynamics and vice versa.

5. Conclusion and Acknowledgment

In this paper, we propose to mine various kinds of patterns from the topology graphs of Markovian models as an alternative to directly mining the data. We show that three kinds of well known Markovian models can be used to mine five kinds of patterns with interesting temporal structures. We also introduce a novel model, the VLHMM, which is versatile on mining a wider variety of patterns than other models. We thank Microsoft Research Asia for providing the training motion data. This project is supported in part by the National Science Foundation of China No. 60573061 and Hong Kong RGC CERG CityU1247/03E.

References

[1] W. G. Aref, M. G. Elfeky, and A. K. Elmagarmid. Incremental, online, and merge mining of partial periodic patterns in time-series databases. *IEEE Trans. Knowledge and Data Engineering*, 16(3):332–342, 2004.

[2] P. Bühlmann and A. J. Wyner. Variable length Markov chains. *Annals of Statistics*, 27:480–513, 1999.

[3] T. Corman, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 1990.

[4] M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid. Periodicity detection in time series databases. *IEEE Trans. Knowledge and Data Engineering*, 17(7):875–887, 2005.

[5] C. P. Gabor, K. J. Supowit, and W.-L. Hsu. Recognizing circle graphs in polynomial time. *Journal of the ACM*, 36(3):435–473, 1989.

[6] J.-W. Han, G.-Z. Dong, and Y.-W. Yin. Efficient mining of partial periodic patterns in time series database. In *IEEE ICDE*, pages 106–115, March 1999.

[7] J.-W. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[8] S. Ma and J. L. Hellerstein. Mining partially periodic even patterns with unknown periods. In *IEEE ICDE*, pages 205–214, April 2001.

[9] J.-F. Mari, D. Fohr, and J.-C. Junqira. A second-order HMM for high performance word and phoneme-based continuous speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996.

[10] E. Nuutila and E. Soisalon-Soininen. On finding the strongly connected components in a directed graph. *Information Processing Letters*, 49:9–14, 1993.

[11] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[12] D. Ron, Y. Singer, and N. Tishby. The power of Amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, 25(2–3):117–149, 1996.

[13] C. Sheng, W. Hsu, and M. L. Lee. Mining dense periodic patterns in time series data. In *IEEE ICDE*, pages 115–115, April 2005.

[14] R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[15] Y. Wang. Mining complex time-series data by learning markovian models. Technical report, Department of Computer Science, Tsinghua University. http://dbgroup.cs.tsinghua.edu.cn/wangyi/motion-engine/icdm.pdf, 2006.

IEEE COMPUTER SOCIETY