Relative Hidden Markov Models for Video-Based Evaluation of Motion Skills in Surgical Training

Qiang Zhang, *Student Member, IEEE* and Baoxin Li, *Senior Member, IEEE*

Abstract—A proper temporal model is essential to analysis tasks involving sequential data. In computer-assisted surgical training, which is the focus of this study, obtaining accurate temporal models is a key step towards automated skill-rating. Conventional learning approaches can have only limited success in this domain due to insufficient amount of data with accurate labels. We propose a novel formulation termed Relative Hidden Markov Model and develop algorithms for obtaining a solution under this formulation. The method requires only relative ranking between input pairs, which are readily available from training sessions in the target application, hence alleviating the requirement on data labeling. The proposed algorithm learns a model from the training data so that the attribute under consideration is linked to the likelihood of the input, hence supporting comparing new sequences. For evaluation, synthetic data are first used to assess the performance of the approach, and then we experiment with real videos from a widely-adopted surgical training platform. Experimental results suggest that the proposed approach provides a promising solution to video-based motion skill evaluation. To further illustrate the potential of generalizing the method to other applications of temporal analysis, we also report experiments on using our model on speech-based emotion recognition.

Index Terms—Relative hidden markov model, relative learning, temporal model, emotion recognition, surgical skill

1 Introduction

Human capability in mastering body motion is the key in domains such as sports, rehabilitation, surgery and dance. Computer-based approaches have been developed over the years for facilitating acquiring (e.g., training in sports and surgery) or regaining (e.g., in rehabilitation) such motion-related skills by human subjects. One central task faced by systems using such approaches is the analysis of motion skills based on some temporal sensory data. With such analysis, skill metrics may be extracted and assigned to a given movement and feedback may accordingly be provided to the subjects for taking actions to improve the underlying skill. For example, [1] utilized control trajectories and motion capture data for human skill analysis, [2] reported motion skill analysis in sports using data from motion sensors, [3] studied computational skill rating in manipulating robots, and [4] considered hand movement analysis for skill evaluation in console operation.

Among others, surgery-related applications have attracted increasing interests, where motion expertise is the primary concern. To improve their motion expertise, surgeons often have to go through lengthy training processes. In recent years, simulation-based surgical training platforms have been developed and widely applied in surgical education. One prominent example is the Fundamentals of Laparoscopic Surgery (FLS) Trainer Box (www.flsprogram.org). With such platforms, it is possible to develop computational approaches to provide objective and quantifiable performance metrics, overcoming the shortcomings in traditional training that relies on costly practice of direct supervision by senior surgeons. Recognizing the sequential nature of motion data, many analysis approaches utilize state-transition models, such as the Hidden Markov Model (HMM). For example, [5] provided an HMM-based method to evaluate surgical residents' learning curve. The method first constructs different HMMs for each different levels of expertise, and then calculates a probability distance between the expert and a novice resident. The magnitude of the probability distance is used to rate the level of the novice resident. HMM was also adopted in [6] to measure motion skills in surgical tasks, where a recorded video is first segmented into basic gestures based on velocity and angle of movement, with segments of the gestures corresponding to the states of an HMM. In [7], Hierarchical Dircichlet process hidden Markov model (HDPHMM [8]) was utilized, which relaxed the requirement of predefining the number of the states for the model.

One practical difficulty in these approaches is that they require the skill labels for the training data since the HMMs are typically learned from sets of data streams with corresponding skill levels. Labeling the skill of a trainee is currently done by senior surgeons, which is not only a costly practice but also one that is subjective and less quantifiable. Thus it is difficult, if not impossible, to obtain a large amount of data with sufficiently reliable skill labels for HMM training. This problem has also been encountered in other fields such as image classification. For example, in [9], it was argued that using binary labels to describe images is not only too restrictive but also unnatural and thus relative visual attributes were used and classifiers were trained based on such features. Relative information has also been used in other applications, e.g., distance metric learning [10], face verification [11], and human-machine interaction [12].

• *The authors are with the Computer Science and Engineering, Arizona State University, Tempe AZ 85287.*
E-mail: {qzhang53, baoxin.li}@asu.edu.

Manuscript received 24 Feb. 2014; revised 21 Sept. 2014; accepted 25 Sept. 2014. Date of publication 1 Oct. 2014; date of current version 8 May 2015.
Recommended for acceptance by D. Xu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2014.2361121

In this paper, we propose a novel formulation termed *Relative Hidden Markov Model* and develop an algorithm for obtaining a solution under this model. The proposed method utilizes only relative ranking (based on certain attribute of interest, or motion skill in the surgical training application) between pairs of inputs, which is easier to obtain and often more consistent. This is especially useful for applications like video-based surgical training, where the trainees go through a series of training sessions with their skills improving over time, and thus the time of the sessions would already provide natural relative ranking of the skills at the corresponding time. The proposed algorithm effectively learns a model from the training data so that the attribute under consideration (i.e., the motion skill in our application) is linked to the likelihood of the inputs under the learned model. The learned model can then be used to compare new data pairs. For evaluation, we first design synthetic experiments to systematically evaluate the model and the algorithm, and then experiment with real data captured on a commonly-used surgical training platform. The experimental results suggest that the proposed approach provides a promising solution to the real-world problem of motion skill evaluation from video.

The key contribution of the work lies in the novel formulation of learning temporal models using only relative information and the proposed algorithm for obtaining solutions under the formulation. Discussion of its relationship to the latent support vector machine is also provided to assist the understanding of why the proposed formulation is suitable for the proposed scenarios. Additional contributions include the specific application of the proposed method to the problem of video-based motion skill evaluation in surgical training, which has seen increasing importance in recent years. An earlier exposition of the proposed method can be found in [13]. This current paper represents a full exploration of the method, including a new learning algorithm that is more efficient, new comparative analysis of the method, and new and updated experiments. In particular, to illustrate that the proposed model is general in nature but not confined to video-based skill analysis, we report its application to a different problem, emotion recognition using speech. To facilitate further exploration and validation by other researchers, source code accompanying this paper has been made publicly available.[1]

In the remainder of this paper, we first review some of the related work in Section 2 and describe basic notations of the HMM in Section 3. The proposed method is then presented in Section 4, including a new algorithm for obtaining solutions in Section 4.3 and discussion of its relationship to latent support vector machine in Section 4.4. The proposed method is evaluated on three types of data in Section 5, including synthetic data (Section 5.1) and videos from surgical simulation systems (Section 5.2), and speech data (Section 6). The paper is concluded in Section 7. In this paper, we use upper-case bold font (e.g., $\mathbf{X}$) for matrices, lower-case bold font (e.g., $\mathbf{x}$) for vectors. We use $\mathbf{X}^i$ to represent i_{th} sequence, $\mathbf{X}_t^i$ for the t_{th} frame of sequence $\mathbf{X}^i$.

1. The code is available at www.public.asu.edu/~bli24/Code SoftwareDatasets.html.

2 Related Work

In this section, we first review two categories of existing work, discriminative learning for hidden Markov models and learning based on relative information, which are most related to our approach. Distinction between our proposed method and the reviewed work will be briefly stated. We also briefly discuss a few more related efforts on skill evaluation in surgery.

Discriminative learning for HMM. Maximum-likelihood methods for learning HMM (e.g., the forward-backward algorithm) in general do not guarantee the discrimination ability of the learned models. To this end, several discriminative learning methods for HMM have been proposed. In [14], a discriminative training method for HMM was proposed based on perceptron algorithms. The methods iterates between the Viterbi algorithm and the additive update of the models. Hidden Markov support vector machine (HM-SVM) was proposed in [15], which combines SVM with HMM to improve the discrimination power of the learned model. These methods are "supervised" in nature, and thus the labeling of the state sequence is required for the training data, which limits their practical use. In [16], another discriminative learning method for HMM was proposed, which only requires the labels of the training sequences. The method initializes the HMMs with maximum-likelihood method and then updates the models with SVM. One drawback is that, the updated models do not always lead to valid HMMs, which could be problematic for a physics-driven problem where the model states have real meanings (like the gesture elements in [6]). Our proposed method requires neither the labeling of the states nor the class label for the training sequences, which are difficult to obtain or even not accessible in many applications. Instead, only a relative ranking of the training data is used, and the resultant model is a valid HMM.

Learning with relative information. Several methods for learning with relative information have been proposed recently. In [10], a distance metric is learned from relative comparisons. Considering the limited training examples for object recognition, [17] proposes an approach based on comparative objective similarities, where the learned model scores high for objects of similar categories and low for objects of dissimilar categories. In [11], comparative facial attributes were learned for face verification. The method of [9] learns relative attributes for image classification and the problem is formulated as a variation of SVM. Similar idea was also been used in [12] for the purpose of human-machine interaction. In [18], relative attribute feedback, e.g., "Shoe images like these, but sportier", is used to improve the performance of image search. Relative information between scene categories has also been used to enhance the performances of scene categorization in [19]. These approaches are mostly for image-based attributes, whereas our current task is on modeling sequential data, for which it is natural to assume that the most relevant attributes (e.g., motion skills) are embedded in a temporal structure. This is what our proposed method attempts to address. Efforts has been observed for estimating the true continuous label of the data from a set of pairwise ranking of training data [20], [21]. However,

those methods do not directly learn a model for ranking/labeling new data.

Skill evaluation for surgical simulations. Objective evaluation of surgical skills has been a topic of research for many years. The authors of [22], [23] used the time of each data, total path traveled and the number of hand movements to rate the surgical skills. It is evident that some of the criteria recommended in these studies (e.g., time of completion) may be relatively easily measured with proper sensory data, while some others cannot be (e.g., respect for tissues). A technique proposed in [24] called task deconstruction was implemented in a recent system by [25]. They used Markov Models to model a sequence of force patterns or positions of the tools. They showed that their Markov Models were suitable for decomposing a task (such as suturing) into basic gestures, and then the proficiency of the complex gesture could be analyzed. While this study offered an intriguing approach to expertise analysis, it required an expert surgeon to provide specifications for building the topology of the model; hence it cannot be easily generalized to new procedures. A similar idea was also utilized in [26]. Jun et al. [27] proposed to segment the training data into modular sub-procedures or therbligs and performance is measured over each sub-procedure.

3 Basic Notations of HMM

In this section, we briefly describe HMM and introduce some basic notations that will be used later. An HMM can be defined by a set of parameters: the initial transition probabilities $\pi \in \mathbb{R}^{K\times 1}$, the state transition probabilities $A \in \mathbb{R}^{K\times K}$ and the observation model $\{\phi_k\}_{k=1}^{K}$, where K is the number of states. There are two central problems in HMM: 1) learning a model from the given training data; and 2) evaluating the probability of a sequence under a given model, i.e., the decoding problem.

In the *learning problem*, one learns the model (θ) by maximizing the likelihood of the training data ($\mathbb{X}$):

$$\theta^* : \max_{\theta} \prod_{\mathbf{X}^i \in \mathbb{X}} p(\mathbf{X}^i|\theta) \sim \max_{\theta} \sum_{\mathbf{X}^i \in \mathbb{X}} \log p(\mathbf{X}^i|\theta), \qquad (1)$$

where $\mathbb{X}$ is the set of i.i.d. training sequences.

One efficient solution to the above problem is the well-known Baum-Welch algorithm [28]. Another scheme, namely the segmental K-means algorithm [29], may also be used to seek a solution, and it has been shown that the likelihoods under models estimated by either of the two algorithms are very close [29]. When the training data include sequences of multiple categories, multiple models would be learned and each model will be learned from data of each category independently.

In the *decoding problem*, given a hidden Markov model, one needs to determine the probability of a given sequence $\mathbf{X}$ being generated by the model. Generally we are more interested in the probability associated with the optimal state sequence ($\mathbf{z}^*$), i.e., $p(\mathbf{X}, \mathbf{z}^*|\theta) = \max_{\mathbf{z}} p(\mathbf{X}, \mathbf{z}|\theta)$. The optimal state path can be found via the Viterbi algorithm. To use HMM in classification, we first compute the probability of the given sequence drawn from each model, then we choose the model yielding the maximal probability.

4 Proposed Method

Based on the previous discussion, we are concerned with a new problem of learning temporal models using only relative information. This is a problem arising naturally in many applications involving motion or video data. In the case of video-based surgical training, the focus is on learning to rate/compare the performance of the trainees from recorded videos capturing their motion. To this end, in recognition of some fruitful trials of HMMs in this application domain, we propose to formulate the task as one of learning a *Relative Hidden Markov Model*, which not only maximizes the likelihood of the training data, but also maintains the given relative rankings of the input pairs. In its most basic form, the proposed model can be formally expressed as (following the notations defined in Eqn. (1))

$$\begin{aligned} \theta \quad &: \quad \max_{\theta} \prod_{\mathbf{X}^i \in \mathbb{X}} p(\mathbf{X}^i|\theta) \\ \text{s.t.} \quad &\quad F(\mathbf{X}^i, \theta) > F(\mathbf{X}^j, \theta), \forall (i,j) \in \mathbb{E}, \end{aligned} \qquad (2)$$

where $F(\mathbf{X}, \theta)$ is a score function for data $\mathbf{X}$ given by model θ, which is introduced to maintain the relative ranking of the pair $\mathbf{X}^i$ and $\mathbf{X}^j$ and $\mathbb{E}$ is the set of given pairs with prior ranking constraint. Different score functions may be defined, e.g., data likelihood and data likelihood ratio, as described in the following sections in Section 4.1 and Section 4.2.

From this formulation, the difference between the proposed method and any of the existing HMM-based methods is obvious. In an existing HMM-based method, a set of models is trained using the training data of each category independently. That is, explicit class labels are required for each training sequence. The proposed model has the following unique features:

- The model does not require explicit class labels. What needed is only a relative ranking.
- The model explicitly considers the ranking constraint between given data pairs, whereas independently-trained HMMs in existing methods cannot guarantee it.
- Only one model is learned for the entire set of data. There are two benefits: more data for training and less computation during testing.

Our method is also different from the existing work on learning with relative attributes in that it models sequential data and the relative ranking information is capsulated in a temporal dynamic model of HMM (albeit new algorithms are thus called for), which has demonstrated performance in modeling physical phenomena like human movements.

In the following sections, we present two instantiations of the general model expressed in Eqn. (2), and develop the corresponding algorithms in each case. It will become clear that the first model (Section 4.1), while being intuitive, has some practical difficulties, which motivated us to develop the improved model of Section 4.2. Both models/algorithms are presented (and evaluated later in Section 5) for the progressive nature of the methods and for facilitating the understanding of the improved model and algorithm of Section 4.2, which is the recommended solution.

4.1 The Baseline Model

While one may use different score functions for F in Eqn. (2) for comparing the input pairs, upon successful training the likelihoods of the sequences should reflect the original ranking. Hence we may set $F(\mathbf{X}^i, \theta) = p(\mathbf{X}^i|\theta)$. With this, the formulation in Eqn. (2) can be rewritten as

$$\begin{aligned} \theta \quad &: \quad \max_{\theta} \prod_{\mathbf{X}^i \in \mathbb{X}} p(\mathbf{X}^i|\theta) \\ \text{s.t.} \quad &\quad p(\mathbf{X}^i|\theta) > p(\mathbf{X}^j|\theta), \forall (i,j) \in \mathbb{E}. \end{aligned} \tag{3}$$

It has been proven in [30] that, the marginal likelihood is dominated by the likelihood with the optimal path and their difference decreases exponentially with the length (number of frames) of a sequence. This idea was used in segmental K-means algorithm and similarly we can approximate the marginal data likelihood $p(\mathbf{X}|\theta)$ by the likelihood with optimal path $p(\mathbf{X}, \mathbf{z}^*|\theta)$ (when there is no ambiguity, we will use $\mathbf{z}$ for $\mathbf{z}^*$), which can be written as:

$$\begin{aligned} \log p(\mathbf{X}, \mathbf{z}|\theta) = &\log p(\mathbf{X}_1|\boldsymbol{\phi}_{\mathbf{z}_1}) + \log \pi(\mathbf{z}_1) \\ &+ \sum_{t=2}^{T} \left[\log p(\mathbf{X}_t|\boldsymbol{\phi}_{\mathbf{z}_t}) + \log \mathbf{A}(\mathbf{z}_t|\mathbf{z}_{t-1})\right] \end{aligned} \tag{4}$$

For some observation models, e.g., multinomial (more details in Appendix A), we can write $\log p(\mathbf{X}^i, \mathbf{z}^i|\theta) = \theta^T h(\mathbf{X}^i, \mathbf{z}^i)$. Accordingly, Eqn. 3 can be finally written as

$$\begin{aligned} \theta \quad &: \quad \max_{\theta \in \Omega} \theta^T \sum_{i:\mathbf{X}^i \in \mathbb{X}} h(\mathbf{X}^i, \mathbf{z}^i) \\ \text{s.t.} \quad &\quad \theta^T h(\mathbf{X}^i, \mathbf{z}^i) \geq \theta^T h(\mathbf{X}^j, \mathbf{z}^j) + \rho, \forall (i,j) \in \mathbb{E}, \end{aligned} \tag{5}$$

where $\rho \geq 0$ defines the required margin between the logarithms of likelihood for a pair of data and Ω defines the set of valid parameters for the hidden Markov model, i.e.:

$$\begin{aligned} \theta(i) \leq 0 \,; \quad &\sum_{i:\theta(i) \in \log(\boldsymbol{\pi})} e^{\theta(i)} = 1, \\ \sum_{i:\theta(i) \in \log(\mathbf{A}_j)} e^{\theta(i)} = 1 \,; \quad &\sum_{i:\theta(i) \in \log(\boldsymbol{\phi}_j)} e^{\theta(i)} = 1, \end{aligned} \tag{6}$$

where $i : \theta(i) \in \log(A_j)$ is the set of the indexes which corresponds to the j_{th} row of matrix A.

For the model in Eqn. (3), we assumed that every pairwise ranking constraint provided in the data is correct (or valid). However, in real data, there may be outliers in such training pairs. To handle this, we further introduce some slack variables ϵ and η, and accordingly Eqn. (5) can be written as following:

$$\begin{aligned} \theta \quad &: \quad \max_{\theta \in \Omega} \theta^T \sum_{\mathbf{X}^i \in \mathbb{X}} h(\mathbf{X}^i, \mathbf{z}^i) - \gamma \sum_{(i,j) \in \mathbb{E}} \epsilon_{ij} \\ \text{s.t.} \quad &\quad \theta^T [h(\mathbf{X}^i, \mathbf{z}^i) - h(\mathbf{X}^j, \mathbf{z}^j)] + \epsilon_{ij} \geq \rho, \forall (i,j) \in \mathbb{E} \\ &\quad \epsilon_{ij} \geq 0, \end{aligned} \tag{7}$$

where γ is the weight for the penalty term $\sum_{(i,j) \in \mathbb{E}} \epsilon_{ij}$. For initialization, we can set $\epsilon_{ij} = 0$. We will defer the optimization algorithm for Eqn. (7) to Section 4.3. After the model is learned, it can be used to a testing pair: For each sequence we evaluate the data likelihood via the Viterbi algorithm and use the logarithm of the data likelihood as the score of the data. By definition, the obtained scores can be used to compare the pair.

4.2 The Improved Model

In the model described in Eqn. (7), we compare the logarithm of the data likelihood, which is, according to Eqn. (4), roughly proportional to the length of the data. Thus a shorter sequence is likely to have a larger score. This means that the learned model would be biased towards shorter sequences. If the observation describes a long, periodic event, e.g., repeating an action multiple times within a sequence, we may consider normalizing the logarithm of the data likelihood by the number of frames of the observation. However, this cannot be applied directly for non-periodic observations like sequences from surgical simulation, where the length of a sequence (corresponding to the time taken for completing a task) is one of the skill metrics.

To overcome the above practical problem, we consider an improved version. Recall that in HMM, we classify a sequence based on the model with which the sequence gets the maximal likelihood, i.e., it is the ratio of data likelihood with different models that decides the label of the data. For example, if $\log \frac{p(\mathbf{X}, \hat{\mathbf{z}}|\theta_1)}{p(\mathbf{X}, \tilde{\mathbf{z}}|\theta_2)} > 0$, then we assign $\mathbf{X}$ to Model θ_1. Thus we propose to use the ratio of the data likelihoods of two HMMs as the score function, i.e., $F(\mathbf{X}, \theta) = \log \frac{p(\mathbf{X}, \hat{\mathbf{z}}|\theta_1)}{p(\mathbf{X}, \tilde{\mathbf{z}}|\theta_2)}$, where we "partition" the original model into two models (or, effectively, we train a pair of HMMs simultaneously). This results in the following improved model:

$$\begin{aligned} \theta_1, \theta_2 \quad &: \quad \max_{\theta_1, \theta_2} \sum_{i \in \Xi_1} \log p(\mathbf{X}^i, \hat{\mathbf{z}}^i|\theta_1) + \sum_{j \in \Xi_2} \log p(\mathbf{X}^j, \tilde{\mathbf{z}}^j|\theta_2) \\ &\quad - \gamma \sum_{(i,j) \in \mathbb{E}} \epsilon_{ij} \\ \text{s.t.} \quad &\quad \log \frac{p(\mathbf{X}^i, \hat{\mathbf{z}}^i|\theta_1)}{p(\mathbf{X}^j, \tilde{\mathbf{z}}^j|\theta_2)} - \log \frac{p(\mathbf{X}^j, \hat{\mathbf{z}}^j|\theta_1)}{p(\mathbf{X}^j, \tilde{\mathbf{z}}^j|\theta_2)} + \epsilon_{ij} \geq \rho \\ &\quad \epsilon_{ij} \geq 0, \end{aligned} \tag{8}$$

where Ξ_1 is the set of data associated with Model θ_1 (Ξ_2 for Model θ_2), $\hat{\mathbf{z}}^i$ is the optimal path for sequence x^i with Model θ_1 and $\tilde{\mathbf{z}}^i$ for optimal path with Model θ_2.

With $\log \frac{p(\mathbf{X}^i, \hat{\mathbf{z}}^i|\theta_1)}{p(\mathbf{X}^j, \tilde{\mathbf{z}}^j|\theta_2)} = \theta_1^T h(\mathbf{X}^i, \hat{\mathbf{z}}^i) - \theta_2^T h(\mathbf{X}^i, \tilde{\mathbf{z}}^i)$, we can rewrite the model in Eqn. (9) (similar to Eqn. (7)):

$$\begin{aligned} \theta \quad &: \quad \max_{\theta \in \Omega} \theta^T \begin{bmatrix} \sum_{i \in \Xi_1} h(\mathbf{X}^i, \hat{\mathbf{z}}^i) \\ \sum_{j \in \Xi_2} h(\mathbf{X}^j, \tilde{\mathbf{z}}^j) \end{bmatrix} - \gamma \sum_{(i,j) \in \mathbb{E}} \epsilon_{ij} \\ \text{s.t.} \quad &\quad \theta^T \begin{bmatrix} h(\mathbf{X}^i, \hat{\mathbf{z}}^i) - h(\mathbf{X}^j, \hat{\mathbf{z}}^j) \\ h(\mathbf{X}^j, \tilde{\mathbf{z}}^j) - h(\mathbf{X}^i, \tilde{\mathbf{z}}^i) \end{bmatrix} + \epsilon_{ij} \geq \rho \\ &\quad \epsilon_{ij} \geq 0, \end{aligned} \tag{9}$$

where $\theta = [\theta_1^T, \theta_2^T]^T$. The optimization algorithm for Eqn. (9) will be presented in Section 4.3. After we learn the model with the improved algorithm, we can apply it to a given pair by first computing their likelihoods with respect to the

"sub-models" given by θ_1 and θ_2 (with the Viterbi algorithm), and then we use the logarithm of the ratio of the data likelihoods as the score to rank/compare the pair.

The learned models θ_1 and θ_2 serve as a unified model to rank the data. We may view them as the centers of two clusters, where the distances of the data to those two centers can be related to the ranking score.

It needs to be emphasized that the improved model is not equivalent to a supervised HMM with two classes. In a 2-class HMM setting, two models will be independently trained with their respective training sets. Here, the proposed model trains two "sub-models" jointly with only relative ranking constraints. Specifically, if there is no further information for Ξ, we could assume that $\Xi_1 = \{i|(i,j) \in \mathbb{E}, \forall j\}$ and $\Xi_2 = \{j|(i,j) \in \mathbb{E}, \forall i\}$, and thus there could be overlaps between Ξ_1 and Ξ_2 (which will become clear in the experiment with synthetic data in Section 5). This situation is not even allowed by a supervised HMM setting. We do not require any extra properties for Ξ_1 and Ξ_2.

4.3 Algorithms for Updating the Model

One important step of both the baseline algorithm and the improved algorithm is updating the models, as formulated in Eqn. (7) and Eqn. (9) accordingly. It is a nonlinear programming problem (due to the nonlinear equality constraint). In our previous paper, we solved it by the primal-dual interior point method, which is of dimension $K(1+K+D)+|\mathbb{E}|$ (or $2K(1+K+D)+|\mathbb{E}|$) with $2|\mathbb{E}|+K(1+K+D)$ (or $2|\mathbb{E}|+2K(1+K+D)$) linear inequality constraints and $1+K+D$ (or $2(1+K+D)$) nonlinear equality constraints for the baseline model (or the improved model). Although the Hessian matrix is diagonal, the computational cost could be still very high when there are a large number of training pairs. In this section, we propose a new algorithm by utilizing the special structure of the problems in Eqn. (7) and Eqn. (9).

Eqn. (7) (similarly for Eqn. (9)) can be written in the following form:

$$\begin{aligned} \theta, \epsilon \quad &: \quad \min_{\theta,\epsilon} \mathbf{f}^T\theta + \gamma 1^T\epsilon \\ \text{s.t.} \quad &: \quad \mathbf{A}\theta + \epsilon \leq \rho \\ &\quad\ \ \mathbf{C}e^{\theta} = 1 \\ &\quad\ \ \theta \leq 0\,; \epsilon \geq 0. \end{aligned} \tag{10}$$

For example, for Eqn. (7), we have $\mathbf{f} = -\sum_{\mathbf{X}^i \in \mathbb{X}} h(\mathbf{X}^i, \mathbf{z}^i)$, $\mathbf{A}$ and $\mathbf{C}$ are constructed according to Eqns. (7) and (6).

Eqn. (10) is a nonlinear programming problem (due to the nonlinear equality constraint). To solve this problem, we first introduce a slack variables $\boldsymbol{\phi}$, where $\log \boldsymbol{\phi} = \theta$. Then Eqn. (10) can be rewritten into the following problem:

$$\begin{aligned} \theta, \epsilon, \boldsymbol{\phi} \quad &: \quad \min_{\theta,\epsilon,\boldsymbol{\phi}} \mathbf{f}^T\theta + \gamma 1^T\epsilon \\ \text{s.t.} \quad &: \quad \mathbf{A}\theta + \epsilon \leq \rho \\ &\quad\ \ \mathbf{C}\boldsymbol{\phi} = 1 \\ &\quad\ \ \log \boldsymbol{\phi} = \theta \\ &\quad\ \ \theta \leq 0; \epsilon \geq 0; 0 \leq \boldsymbol{\phi} \leq 1. \end{aligned} \tag{11}$$

According to Eqn. (11), $\boldsymbol{\phi}$ will be a valid hidden Markov model (or hidden Markov model pairs $[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2]$ for the improved model). We then apply the Augmented Lagrange multiplier method to the equality constraint $\log \boldsymbol{\phi} = \mathbf{u}$ of the problem in Eqn. (11):

$$\begin{aligned} \theta, \epsilon, \boldsymbol{\phi} \quad &: \quad \min_{\theta,\epsilon,\boldsymbol{\phi}} \mathbf{f}^T\theta + \gamma 1^T\epsilon + \\ &\qquad < \lambda, \theta - \log \boldsymbol{\phi} > + \frac{\mu}{2}\|\theta - \log \boldsymbol{\phi}\|_2^2 \\ \text{s.t.} \quad &: \quad \mathbf{A}\theta + \epsilon \leq \rho \\ &\quad\ \ \mathbf{C}\boldsymbol{\phi} = 1 \\ &\quad\ \ \theta \leq 0; \epsilon \geq 0; 0 \leq \boldsymbol{\phi} \leq 1, \end{aligned} \tag{12}$$

where λ is the Lagrange multiplier and μ is some nonnegative constant. In Eqn. (12), the nonlinear equality constraint is removed.

Eqn. (12) can be solved via block coordinate descent by iterating between the following two sub-problems:

Sub-problem 1: fix $\boldsymbol{\phi}$ to solve θ and ϵ, which is

$$\begin{aligned} \theta, \epsilon \quad &: \quad \min_{\theta,\epsilon,\boldsymbol{\phi}} \mathbf{f}^T\theta + \gamma 1^T\epsilon + \\ &\qquad < \lambda, \theta - \log \boldsymbol{\phi} > + \frac{\mu}{2}\|\theta - \log \boldsymbol{\phi}\|_2^2, \\ \text{s.t.} \quad &: \quad \mathbf{A}\theta + \epsilon \leq \rho \\ &\quad\ \ \theta \leq 0; \epsilon \geq 0. \end{aligned} \tag{13}$$

It is a quadratic programming problem with linear inequality constraints.

Sub-problem 2: fix θ and ϵ to solve $\boldsymbol{\phi}$, which is

$$\begin{aligned} \boldsymbol{\phi} \quad &: \quad \min_{\boldsymbol{\phi}} < \lambda,\ \theta - \log \boldsymbol{\phi} > + \frac{\mu}{2}\|\theta - \log \boldsymbol{\phi}\|_2^2 \\ &\quad\ \ \mathbf{C}\boldsymbol{\phi} = 1 \\ &\quad\ \ 0 \leq \boldsymbol{\phi} \leq 1. \end{aligned} \tag{14}$$

It is a nonlinear problem with linear constraints.

Given the special structures of $\mathbf{C}$, where each column has one and only one element being nonzero (recall Eqn. (6)), Sub-problem 2 can be separated into a set of smaller problems:

$$\begin{aligned} \boldsymbol{\phi}^k \quad &: \quad \min_{\boldsymbol{\phi}^k} < \lambda^k, \theta^k - \log \boldsymbol{\phi}^k > + \frac{\mu}{2}\|\theta^k - \log \boldsymbol{\phi}^k\|_2^2 \\ &\quad\ \ \mathbf{1}^T\boldsymbol{\phi}^k = 1 \\ &\quad\ \ 0 \leq \boldsymbol{\phi}^k \leq 1, \end{aligned} \tag{15}$$

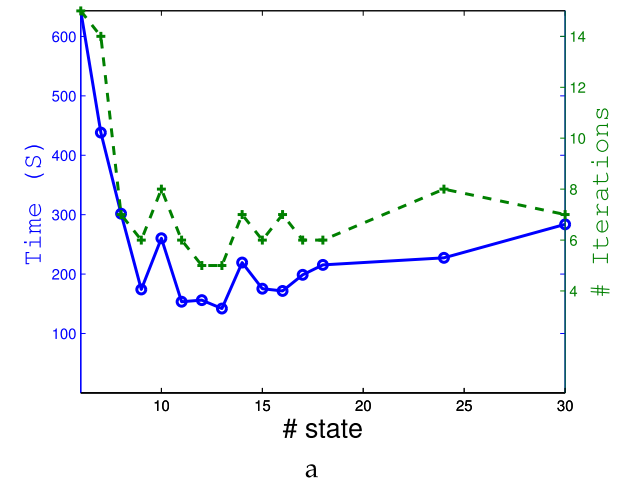
where k is the set of indexes of columns, whose values are nonzero at the k_{th} row of $\mathbf{C}$. Those smaller problems are again a nonlinear problem with linear constraint, whose dimensions are only K (number of states) or D (number of feature dimensions).

To solve this problem we can use the primal-dual interior point method, whose gradient and hessian are computed as

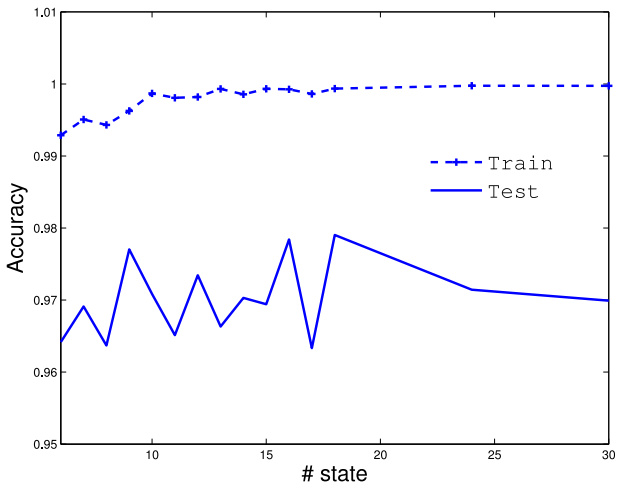
$$J = \frac{-\lambda^k + \mu^k \log \boldsymbol{\phi}^k - \mu^k\theta^k}{\boldsymbol{\phi}^k},$$

$$H = \Lambda\left(\frac{\lambda^k - \mu\,\log \boldsymbol{\phi}^k + \mu\theta^k + \mu}{\boldsymbol{\phi}^k \cdot \boldsymbol{\phi}^k}\right),$$

where $\Lambda(\cdots)$ converts a vector to a diagonal matrix. In addition, we can compute the starting point of the problem in Eqn. (15) as: by taking the gradient of the objective function

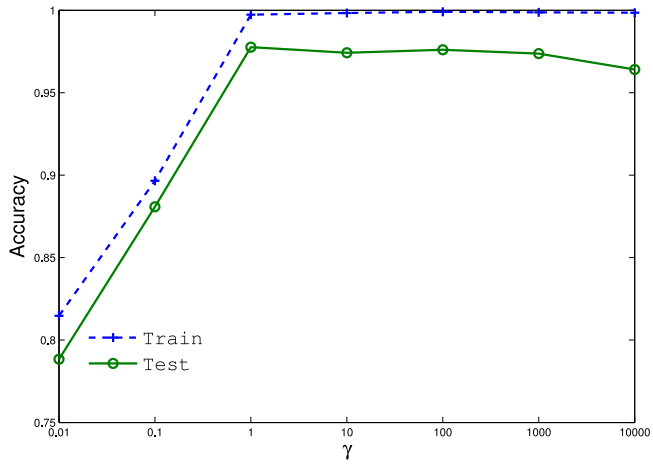


a

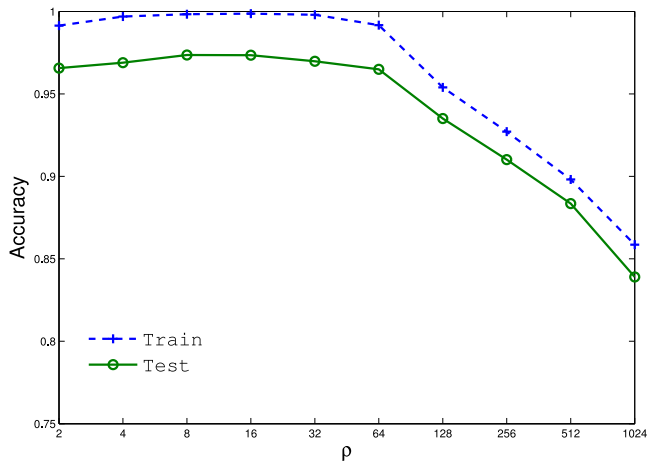


b

Fig. 1. The experiment result with different numbers of states: (a) the computational time (blue solid curve) and number of iterations needed for convergence (green dashed curve); (b) the accuracy of the improved method. The *X*-axis is the number of states.



a



b

Fig. 2. The accuracy of the improved method: (a) with different γ (ρ is fixed to 10), which controls the weight of the penalty term with slack variables; (b) with different ρ (γ is fixed to $1,000$), which controls the margin of the learned models.

with regard to $\log \phi^k$, we have $-\lambda^k + \mu(\log \phi^k - \theta^k) = 0$, i.e., $\phi^k = e^{(\theta^k + \frac{\lambda^k}{\mu})}$. The linear constraint can be solved simply by projection, i.e., $\phi^k = \frac{1}{N} e^{(\theta^k + \frac{\lambda^k}{\mu})}$, where $N = \sum e^{(\theta^k + \frac{\lambda^k}{\mu})}$.

Algorithm 1. Algorithm for the Baseline (Improved) Model

Input: $\mathbb{X}$, $\mathbb{E}$, ρ, γ, σ (, Ξ_1 and Ξ_2)
Output: ϕ
Initialization: Initialize ϕ (or ϕ_1 and ϕ_2) via ordinary HMM learning algorithm, $\lambda = \frac{\log\theta}{|\theta|_2}$ and $\mu = \frac{1.25}{|\theta|_2}$;
while not converged **do**
  Compute the optimal path $\mathbf{z}$ (or $\hat{\mathbf{z}}$ and $\tilde{\mathbf{z}}$) for each sequence with ϕ (or ϕ_1 and ϕ_2);
  solve Sub-problem 1;
  solve Sub-problem 2;
  update $\lambda = \lambda + \mu(\theta - \log\phi)$ and $\mu = \mu \times \sigma$;
  check convergence;
end while

Finally, we briefly summarize the algorithms for the baseline model (Eqn. (7)) and the improved model (Eqn. (9)) below (noting the similarity in form of the algorithms and thus putting them compactly together):

According to [31], the proposed method will converge to the local minimum of the problem in Eqn. (10). And for convergence, we check $\frac{\|\theta - \log\phi\|_2}{\|\theta\|_2}$. If it is smaller than some value, e.g., 10^{-6}, the algorithm will be terminated. In initialization, $|\theta|_2$ is the vector L_2 norm of of θ.

Remarks on the Parameters. The parameter γ controls the weight of the penalty term with the slack variables, which is similar to the functionality of C in support vector machines [32]. The parameter ρ controls the desired gap of the score of two data points, i.e., $\frac{p(\mathbf{X}^i,\mathbf{z}^i|\theta)}{p(\mathbf{X}^j,\mathbf{z}^j|\theta)} \geq e^{\rho}\ \forall(i,j) \in \mathbb{E}$ in the baseline model and $\frac{p(\mathbf{X}^i,\hat{\mathbf{z}}^i|\theta_1)}{p(\mathbf{X}^i,\tilde{\mathbf{z}}^i|\theta_2)} \frac{p(\mathbf{X}^i,\tilde{\mathbf{z}}^j|\theta_2)}{p(\mathbf{X}^i,\hat{\mathbf{z}}^j|\theta)} \geq e^{\rho}\ \ \forall(i,j) \in \mathbb{E}$ in the improved model. In Section 5.1, we will evaluate different parameter settings (Fig. 2), which leads us to set $\gamma = 1,000$ and $\rho = 10$ in our final experiments. The parameter σ controls the convergence speed of the algorithm, which should be a positive number larger than 1. σ is typically within $1.1 - 1.5$, and 1.25 is used in this paper.

The proposed algorithm, compared with the one used in [13], has lower computational cost, due to the removal of

TABLE 1
Comparing the Method in [13] and the Proposed Method for Updating the Baseline Model, with Regarding to the Problem Size, Number of Linear Constraints and Nonlinear Constraints

	Method in [13]	Proposed Method	
		Sub-problem 1	Sub-problem 2
Problem Size	$K(1+K+D)+\lvert\mathbb{E}\rvert$	$K(1+K+D)+\lvert\mathbb{E}\rvert$	K(or D)
# Linear Const.	$2\lvert\mathbb{E}\rvert+K(1+K+D)$	$2\lvert\mathbb{E}\rvert+K(1+K+D)$	1+2K(or 1+2D)
# Nonlinear Const.	$1+K+D$	0	0

For Sub-problem 2 of the proposed method, it can be divided into several smaller problems.

the nonlinear equality constraint with augmented Lagrange multiplier. For Sub-problem 1, the quadratic term is a diagonal matrix and many solvers (e.g., CPLEX) can solve it quite efficiently. Sub-problem 2 is a nonlinear minimization problem with linear equality constraints; however, it can be decomposed into several smaller problems.

A comparison between the method in [13] and the proposed method for updating the baseline model is shown in Table 1. In Section 5.1, we will also compare the computational time of those two methods under varying $\mathbb{E}$ on synthetica data (Fig. 6).

4.4 Relationship to Existing Methods

The proposed method is related to latent support vector machine [33]. Given a training set of input-output pairs $\{(x_i, y_i)\}_{i=1}^n$, where $y_i \in \{-1, 1\}$, Latent SVM tries to learn a predictor of the form:

$$f_w(x) = \max_z w^T \Psi(x, z), \tag{16}$$

where w is the parameter of the predictor, $\Psi(x, z)$ is the feature mapping function and z is the latent variable. The training stage of Latent SVM can be formulated as the following problem:

$$\min_w \frac{1}{2}\|w\|_2^2 + C\sum_i \max(0, 1 - y_i f_w(x_i)). \tag{17}$$

Latent SVM is a non-convex problem, as the latent variable is unknown, and the coordinate descent approach is used for solving this problem.

Given a training set $\{(x_i, y_i)\}_{i=1}^n$, where $x_i = (\mathbf{x}_i^L, \mathbf{x}_i^R)$ is a pair of sequences and $y_i \in \{-1, 1\}$ is the ranking of the pair, by defining the feature mapping function as $\Psi(x_i, z_i) = [h(\mathbf{x}_i^L, \mathbf{z}_i^L) - h(\mathbf{x}_i^R, \mathbf{z}_i^R)]$, with the latent variable $z_i = (\mathbf{z}_i^L, \mathbf{z}_i^R)$ being a pair of state sequences for the pair $x_i = (\mathbf{x}_i^L, \mathbf{x}_i^R)$, we have

$$\begin{aligned} &\min_w \frac{1}{2}\|w\|_2^2 + C\sum_i \epsilon_i \\ \text{s.t.} \quad &y_i \max_{\mathbf{z}_i^L, \mathbf{z}_i^R} \left\{w^T\left[h\left(\mathbf{x}_i^L, \mathbf{z}_i^L\right) - h\left(\mathbf{x}_i^R, \mathbf{z}_i^R\right)\right]\right\} + \epsilon_i \geq 1 \\ &\epsilon_i \geq 0. \end{aligned} \tag{18}$$

We can find that Eqn. (18) is similar to our baseline model (Eqn. (7)), except for the following differences.

1) In Eqn. (18), the L_2 norm is applied to the parameter of the predictor w (which is related to the margin). In the proposed methods we require w to be a valid hidden Markov model while defining a fixed-margin, i.e., ρ. Thus the proposed method can always guarantee the learned model is a valid hidden Markov model.
2) In Eqn. (18), the two state sequences z (i.e., the latent variables) are optimized jointly, where no known efficient solution is available. In the proposed method, the two state sequences are optimized separately with regarding to the likelihood, which can be solved efficiently via dynamic programming (i.e., the Viterbi algorithm);
3) Given the model learned by the latent SVM, we can only rank a pair of sequences. However, the model learned by the proposed method is capable of not only ranking a pair of sequences but also assigning a score for each sequence.

Those differences make the proposed method (both the baseline model and the improved model) more suitable for modeling the sequential data, e.g., video, speech.

5 Experiments

In this section, we evaluate the proposed methods, including the baseline method and the improved method, using both synthetic data (Section 5.1) and realistic data collected from the surgical training platform FLS box (Section 5.2). The performance of the proposed methods is compared with a supervised 2-class HMM. (Lacking a comparative approach in the literature that is both unsupervised and works with only relative rankings, this is believed to be a reasonable way of generating a reference point to assess the proposed methods.)

Since we do not have the label information for the training data, we train the HMM as follows. For the HMM algorithm, we initialize the two sets as $\Xi_1 = \{i|(i, j) \in \mathbb{E}, \forall j\}$ and $\Xi_2 = \{j|(i, j) \in \mathbb{E}, \forall i\}$. Each of the sets is then used to train a HMM. Note, the data generated from data-generating Models $\theta_2 \sim \theta_5$ could be included in both Ξ_1 and Ξ_2. Thus existing discriminative learning methods for HMM could not be applied here.

5.1 Evaluation with Synthetic Data

To evaluate the proposed method, we generate synthetic data as follows. We first generate six different HMMs (θ_1 to θ_6, referred as data-generating models), from each of which we draw 200 sequences, with the length being uniformly distributed between 80 to 120. Each data-generating model has five states. For the sequences from each data-generating model, we randomly assign 50 of them to the training set

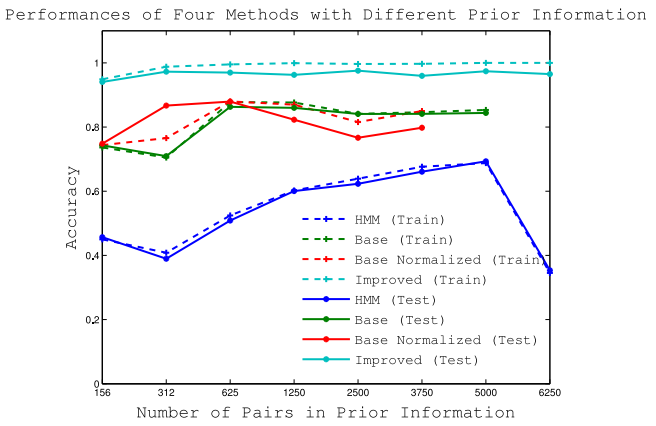


Fig. 3. The results of four methods on training set (dashed curve) and testing set (solid curve) with different numbers of training pairs.

and the remaining to the testing set. We assume there exists a score function such that $F(\mathbf{X}^i) > F(\mathbf{X}^j)$ if and only if $\mathbf{X}^i \sim \theta_k$, $\mathbf{X}^j \sim \theta_l$ and $k < l$. That is, the sequences from a data-generating model with a lower index are viewed to have a higher score (or ranking) than those from a data-generating model with a higher index. A set of pairs $\{(i,j)|\mathbf{X}^i \sim \ \theta_k, \mathbf{X}^j \sim \theta_{k+1}, k = 1, \ldots, 5\}$ are then formed accordingly, some of which are then randomly selected as the training pairs $\mathbb{E}$.

We use the proposed methods and also HMM to learn models from the training pairs. The learned models are then used to evaluate the testing set, i.e., how many testing pairs that they rank the same as the ground truth. The result of the methods with different numbers of training pairs is summarized in Fig. 3, where due to the computational time it takes, we do not have the results for the baseline method when there are more than $3{,}750$ training pairs. From Fig. 3, we can find that the improved method achieves the best results on both the training set and the testing set; and the HMM method gives the worse result. In addition, the performance of both of the proposed methods stabilized after certain number of training pairs. However the performance of the HMM method drops dramatically when the number of training pairs reaches about $6{,}250$. It can be explained by that the two HMMs share a lot of common data (for those generated by $\theta_2 \sim \theta_5$) and the models are trained independently without considering their discrimination ability. Normalizing the logarithm of the data likelihood does not improve the performance of baseline method, which could be explained by that, all the sequences have roughly the same length, i.e., $80 \sim 120$.

Fig. 4 shows the logarithm of the data likelihood ratio with the models learned by the improved method, when about $1,250$ training pairs are provided. This clearly demonstrates that, although we formed the training pairs only with data from data-generating models of adjacent indexes (i.e., i and $i+1$), the learned model is able to recover the strict ranking of the original data. We can also try to classify the data into six models, by thresholding the logarithm of data likelihood ratio, where, for the model learned with the improved method, the classification accuracy is 86.44 and 98.60 percent for testing and training respectively.

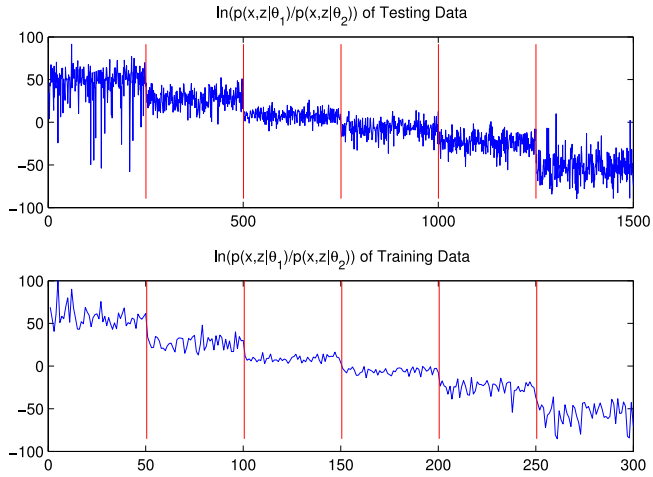


Fig. 4. The logarithm of the data likelihood ratio with the models learned by the improved method. Top: the result for the testing set. Bottom: the result for the training set. The data are grouped (as the section partitioned by the red lines) according to the data generation model from which they are synthesized.

Convergence and Speed. For empirically understanding the convergence behavior of the improved method, we plot in Fig. 5 the objective value in the model as a function of the number of iterations. We can find that the improved method converges fairly quickly (within about 14 iterations) and the value of the objective function monotonically increases.

We also compare the computational time of the optimization method in [13] (shown as the red/upper curve) and the proposed optimization method (in Section 4.3 and shown as the green/lower curve) in solving the improved model under varying number of training pairs in Fig. 6. In [13], a primal-dual interior point method is utilized to update the model; while in this paper, we design an augmented Lagrange multiplier method which utilizes the special structure of the objective function of the problem. From the plot, we can find that the proposed optimization method has a much lower computational cost than the one proposed in [13].

Parameter Selection. To understand the effect of parameters to the performances of the improved method, including accuracy and computation cost, we evaluate it with

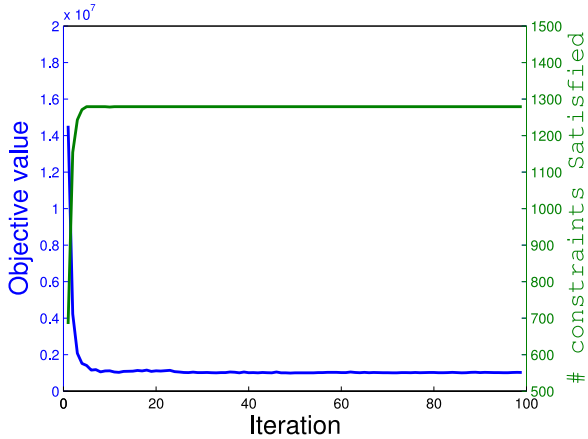


Fig. 5. The convergence behavior of the improved method, where around $1,250$ training pairs were used. The blue curve/axis shows the value of the objective function, and the green curve/axis shows the number of constraints satisfied.

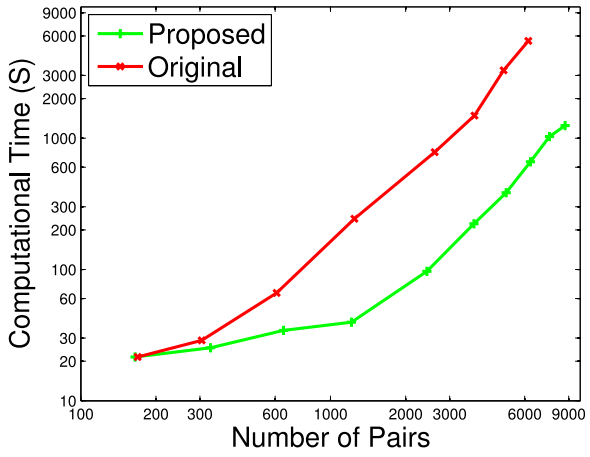


Fig. 6. The computation time for solving the improved model with the method proposed in [13] (red/upper curve) and the method proposed in Section 4.3 (green/lower) under varying number of training pairs. For illustration purpose, we use log-log plot, where X-axis is the number of training pairs (from around 125 to around $9,000$) and *Y*-axis is the computation time in unit second (from about 20 to around $6,000$). The time is measured in Matlab on a dual-core PC platform.

varying combination of parameters. First we learn the model with varying numbers of states (K), from 6 to 30. The result is shown Fig. 1. From Fig. 1b, we can find that, though the accuracy for the training data increases with the number of states, the accuracy for testing doesn't following this trend, which indicates a potential risk of overfitting. The computational time and number of iteration until convergence get minimum when the number of states is 11-13. We also do experiment with different combinations of γ (controlling the weight of the penalty term with slack variables) and ρ (controlling the margin of the model), where the experiment result is shown in Fig. 2. From this experiment we can find that, $\gamma \in [1, 1{,}000]$ and $\rho \in [4, 32]$ are good choices.

It is obvious from this experiment that the sequences are different from (or similar to) each other only because they are from different (or the same) data-generating models, whereas their relative ranking can be arbitrarily defined. In the end, the proposed methods will learn a temporal model to reflect the defined rankings. This suggests that, as long as we can assume there are some data-generating models for the given sequential data, we can use the proposed methods to learn a relative HMM. This is the basis for applying the approach to the surgical training data in the following section, where it is reasonable to assume that movement patterns of subjects with different skill levels may be modeled by different underlying HMMs while the ranking can be based on the time of training, which reflects the skill level of the subject at the time.

5.2 Skill Evaluation Using Surgical Training Video

We now evaluate the proposed method using real videos captured from the FLS trainer box, which has been widely used in surgical training. The data set contains 546 videos captured from 18 subjects performing the "peg transfer" operation, which is one of the standard training tasks a resident surgeon needs to perform and pass. The number of frames in each video varies from $1,000$ to $6,000$ (depending on the trainees' speed in completing a training session). The data set covers a training period of four weeks, with every trainee performing three sessions each week.

In the training, the subject needs to lift six objects (one by one) with a grasper by the non-dominant hand, transfer the object midair to the dominant hand, and then place the object on a peg on the other side of the board. Once all six objects are transferred, the process is reversed, and the objects are to be transferred back to the original side of the board. The videos capture the entire process inside the trainer box, showing how the tools and objects are moved by the subject. The motion skill is related to how well the subjects perform in such operation. In the existing practice, senior surgeons rate the performance of the trainees based on such videos. Our goal is to perform the rating automatically with the proposed model.

Based on the reasonable assumption that the trainees improve their skills over time (which is the whole point of having the resident surgeons going through the training before taking the exam), the time of recording is used to rank the recorded videos *within each subjects' corpus* (i.e., a later video is associated with a better skill). Other than this relative ranking, there are no other labels assumed for the video, e.g., there is no rank information between videos of different subjects (which would be hard to obtain anyway, since there is no clearly-defined skill levels for a group of trainees with diverse background). Based on this, we randomly pick 300 pairs for training, similar to the experiment using synthetic data.

Feature Extraction. We use the "bag of words" approach for feature extraction from the videos as follows. The spatio-temporal interest point detector [34] is applied to obtain the histogram-of-gradient (HoG) features, which was found to be useful in target application in the literature [35]. K-means ($k = 100$) is then used to build a code book for the descriptors of the interest points. Finally, the code book is used to obtain a histogram of interest points for each frame, and thus each video is represented as a sequence of histograms. This representation, compared with the existing way of using bag of words in action recognition, i.e., transforming each video into a single histogram, can better capture the temporal information of the data. For all three methods, we set the number of states to ten.

After learning the models from the training data, we compute the score of the test data as the logarithm of data likelihood (for the baseline method) or the logarithm of the data likelihood ratio (for the improved method and the HMM). We compare these scores for each pair of the testing data (within each subject) and compute the percentage of correctly labeled pairs (recall that, we use their time of recording as ground truth). To demonstrate the advantage of the proposed method, we also compare with the "relative attribute" method [9] (referred as "SVM" in the following discussions), which relies on ranking SVM. For "relative attribute", we represent each video as a histogram by accumulating the sequence of histograms of the video along the temporal direction.

The result is summarized in Table 2, where the improved method obtained a significantly better result than the other approaches, including "relative attribute". Surprisingly, the baseline method even performed slightly worse than the HMM method. This is largely due to the wide range of

TABLE 2
The Result for Experiment on Evaluating Surgical Skills

Method	SVM	HMM	Baseline	Improved
# Pairs	6335	6363	6215	**6993**
Accuracy	78.91%	79.39%	77.54%	**87.25%**

There are 8,015 *pairs in total (only* 300 *for training), excluding the comparisons among data of different subjects.*

variations of the length of the input sequences. Fig. 7 shows the computed scores with the learned models, where for better illustration purpose we group them by their subject ID and within each subject's corpus we sort the videos by their recording time. From the figure, we can find that the improved method (bottom) reveals a more clear trend for the data than both the HMM method (top) and the baseline method (middle), i.e., the scores of the data increase over times (X-axis) for each subject (segments within the red lines). It is worth emphasizing that only one joint model is learned from ranked pairs of subjects with potentially varying skill levels. Still the learned model is able to recover the improving trend, independent of the underlying skill levels.

As shown in Fig. 7, the model learned with the proposed method can be used for comparing not only the videos of the same subjects but also the videos from different subjects, where the logarithm of data likelihood ratio can be used as a measurement of the skills. However, it is not possible to quantitatively measure the accuracy in comparing videos from different subjects, due to the lack of ground truth information for videos from different subjects.

It is also interesting to look at what the jointly-learned models look like. Fig. 8 depicts the two models learned by the improved method in this real-data based experiment. From the figure, we can see that the two models have different transition patterns. For example, the transition from State 8 to States 2 and 5 are only observed in Model 1. This may be linked to different motion patterns for data of different surgical skills, with the hidden states corresponding to some underlying action elements (and thus the transition patterns vary with the skill).

6 Additional Validation Using Speech Data

Although the proposed approach was evaluated above in the context of motion skill analysis in surgical training, using visual data as the input, the approach itself is general and applicable for other applications involving temporal data. To show that the proposed method can be used to solve temporal inference problems other than video-based motion skill assessment, we now consider an exemplar problem, speech-based emotion recognition, where the attribute of interest (the underlying emotion of a speaker) needs to be inferred from sequential data. Emotion recognition has received attention from researchers due to its broad applications. For example, in human-machine interaction, better responses can be made if the emotional state of the human can be recognized. Existing work on this in the literature mainly focuses on developing models for assigning the labels like "pleasing", "angry" and "neural" to the data, e.g., [36], [37], [38], [39]. Most of the those efforts are

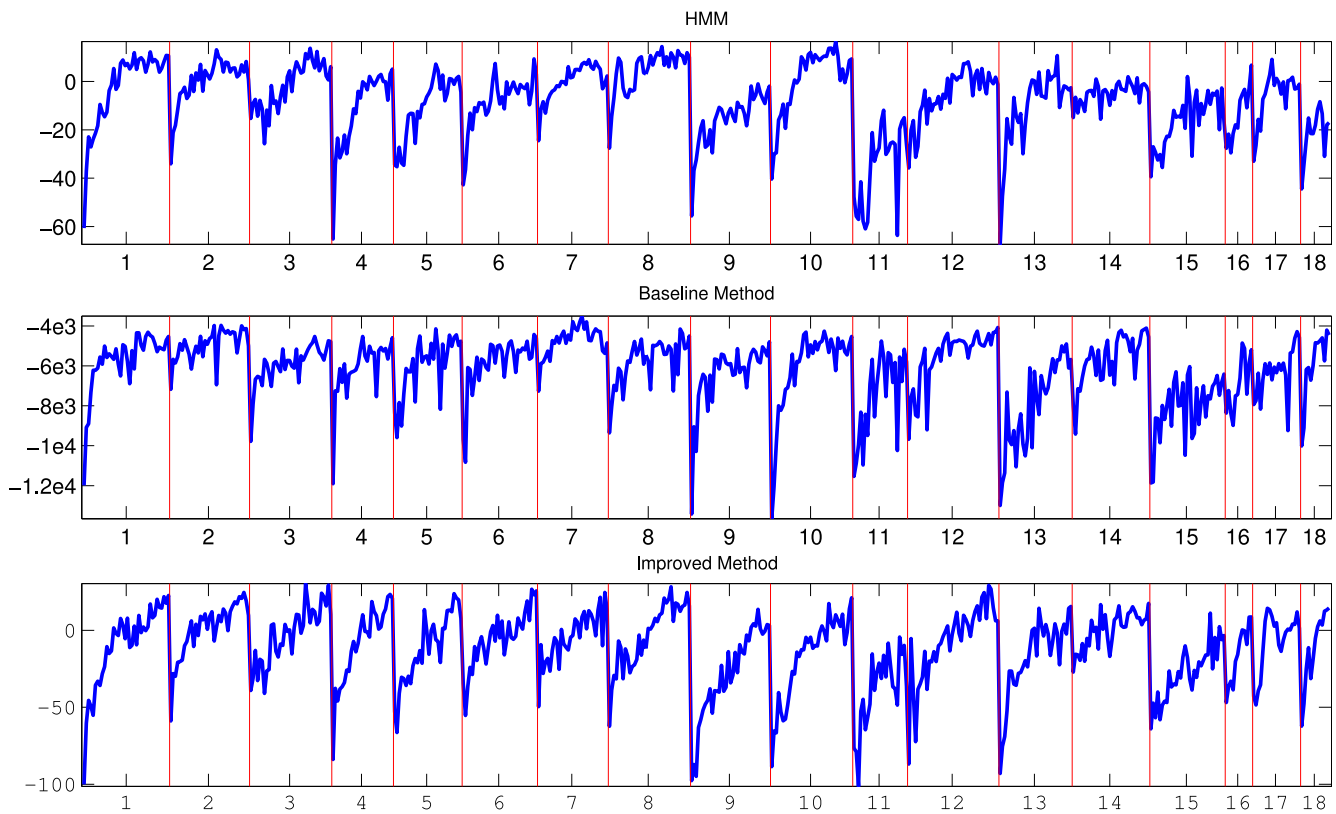


Fig. 7. Top: the logarithm of the data likelihood ratio from two models learned by HMM. Middle: the logarithm of data likelihood with the model learned by the baseline method. Bottom: the logarithm of the data likelihood ratio with the models learned by the improved method. The red vertical lines separate the data of different subjects, where X-axis is the corresponding subject ID. Within each subjects' corpus, the videos are sorted according to their time of recording.

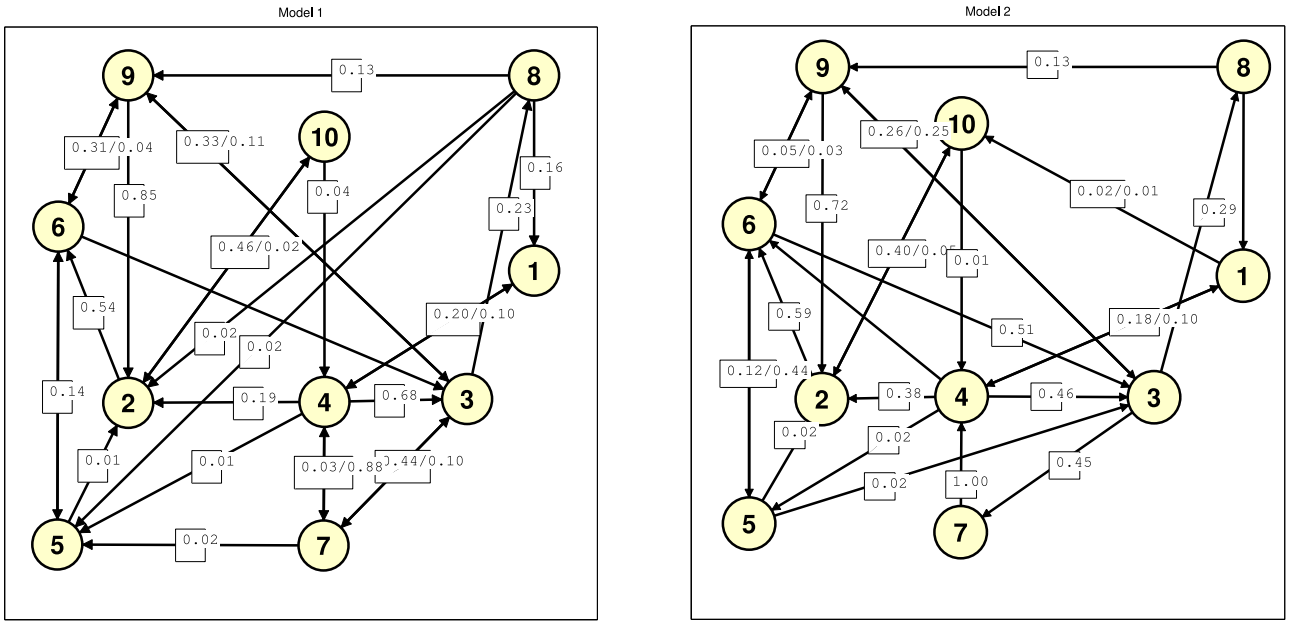


Fig. 8. The two component models (Model 1 for Ξ_1 and Model 2 for Ξ_2) learned by the improved method, where we only draw the edges with a transition probability larger than 0.01 and ignore self transitions. The number attached to each edge indicates the transition probability.

supervised in natural, i.e., the ground truth labeling for the training data is required. For example, [40] used support vector machines, [36] used hidden Markov models, both utilizing fully-labelled data. The ground truth data typically require manual labeling by human, which is an error-prone process especially if absolute labels must be assigned to ambiguous data. With the proposed model, we can support learning with only relative labels like "Audio a is more pleasing than Audio b", which is easier to obtain and also less error-prone.

In this experiment, we use Utsunomiya University Spoken Dialogue Database For Paralinguistic Information Studies (UUDB)[41](http://uudb.speech-lab.org), which contains 4840 assets labeled across six dimensions (pleasantness, arousal, dominance, credibility, interest and positivity) on a scale of 1 to 7. The ground truth is based on the average of scores of three annotators. For our experiment, we pick the assets which are longer than 1 second to ensure the effectiveness of emotional recognition, which results in 991 assets, where half of the data are used for training and the remaining for testing. For generating the ground truth pairs, we randomly picks 1000 pairs from the training assets. Note that, we say two assets are similar, if the difference of the labeled scores of two assets is within the range of $(-1, 1)$.

For feature extraction, we use Hidden Markov Model Toolkit (HTK)[42], where the MFCC coefficients are extracted with the following configurations: sampling rate is 100 HZ, windows size is 25 millisecond, number of filter bank channels is 26, cepstral liftering coefficient is 22 with 12 cepstral parameters and the feature vector is normalized. K-means is applied to the MFCC coefficients of all the training data to generate a code book of 64 elements. Finally, each data is converted to a sequences of histograms. We use the same set of parameters as the previous experiment.

The experimental results are reported in Table 3, where we also provide a comparison to the relative attribute [9] as referred by "SVM". From the table, we can find that the improved method consistently outperforms than both plain HMM and also the baseline method in all six dimensions. We also find that the baseline method gets low accuracy on this experiment, which can be explained by that the length of the audio (in number of temporal frames) varies dramatically and the baseline method obviously cannot handle this variation very well.

7 Discussions and Conclusions

In this paper, we presented a new formulation for the problem of learning temporal models using only relative information. Algorithms were developed under the formulation, and experiments using both synthetic and real data were performed to verify the performance of the proposed method. In essence, the proposed method attempts to learn an HMM with relative constraints. Such a setting is useful for many practical applications where relative attributes are easier to obtain while explicit labeling is difficult to get. The application of video-based surgical training was the focus of this study, and the evaluation results using realistic data suggests that the proposed method provides a promising solution to the problem of motion skill evaluation from

TABLE 3
The Result for Experiment on UUDB Datasets

Dimension	SVM	Improved	Baseline	HMM
Pleasantness	75.25%	**77.30%**	57.96%	75.05%
Arousal	82.11%	**86.95%**	55.74%	69.55%
Dominance	74.13%	**87.95%**	63.04%	77.32%
Credibility	69.15%	**76.68%**	55.11%	71.74%
Interest	76.91%	**81.90%**	62.56%	78.07%
Positivity	68.08%	**74.99%**	67.84%	70.36%
Average	74.27%	**81.28%**	53.14%	73.72%

We evaulate the accuracy of ranking pairs with the learned models compared with the ground truth ones.

videos. For future work, we plan to extend the proposed method to cover different observation models so that more types of applications may be handled. That also includes investigating alternative feature spaces which may be more effective for the target problem.

APPENDIX A

For multinomial observation model, i.e., $p(\mathbf{X}_t|\phi_{\mathbf{z}_t}) = \prod_{d=1}^{D} \phi_{\mathbf{z}_t}(l)^{\mathbf{X}_t(l)}$, where D is the dimension of each frame, $\mathbf{X}_t(l)$ is the l_{th} dimension of $\mathbf{X}_t$ and $\phi_{\mathbf{z}_t}$ are the parameters of observation model with State $\mathbf{z}_t$, we can further define the following variables for each sequence $\mathbf{X}^i$:

$$\begin{aligned}
\mathbf{n}^i \in \mathbb{R}^{K\times 1} \quad &: \quad \mathbf{n}^i(k) = \delta\big(\mathbf{z}_1^i = k\big),\\
\mathbf{O}^i \in \mathbb{R}^{K\times D} \quad &: \quad \mathbf{O}^i(k,d) = \sum_{t:\mathbf{z}_t=k} \mathbf{X}_t^i(d),\\
\mathbf{M}^i \in \mathbb{R}^{K\times K} \quad &: \quad \mathbf{M}^i(k,l) = \sum_{t=2}^{T} \delta\big(\mathbf{z}_{t-1}^i = k\big)\delta\big(\mathbf{z}_t^i = l\big),
\end{aligned}$$

where $\delta(\cdot)$ is Dirac Delta function. Then the log likelihood with the optimal path can be written as:

$$\begin{aligned}
&\log p(\mathbf{X}^i, \mathbf{z}^i|\theta)\\
&\quad = \sum_l \mathbf{n}^i(l)\log\,\pi(l) + \sum_{k,l} \mathbf{M}^i(k,l)\log\,\mathbf{A}(k,l)\\
&\qquad + \sum_{k,d} \mathbf{O}^i(k,d)\log\,\phi_k(d)\\
&\quad = \theta^T h(\mathbf{X}^i, \mathbf{z}^i), \qquad (19)
\end{aligned}$$

where $\theta = [\log\,\pi; \mathrm{vec}(\log\,\mathbf{A}); \mathrm{vec}(\log\,\phi)]$, $h(\mathbf{X}^i, \mathbf{z}^i) = [\mathbf{n}^i; \mathrm{vec}(\mathbf{M}^i); \mathrm{vec}(\mathbf{O}^i)]$ and vec converts matrix to vector.

ACKNOWLEDGMENTS

The work was supported in part by a grant (#0904778) from the National Science Foundation (NSF). Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] F. Duan, Y. Zhang, N. Pongthanya, K. Watanabe, H. Yokoi, and T. Arai, "Analyzing human skill through control trajectories and motion capture data," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Aug. 2008, pp. 454–459.

[2] K. Watanabe and M. Hokari, "Kinematical analysis and measurement of sports form," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 549–557, May 2006.

[3] S. Suzuki, N. Tomomatsu, F. Harashima, and K. Furuta, "Skill evaluation based on state-transition model for human adaptive mechatronics (HAM)," in *Proc. 30th Annu. Conf. IEEE Ind. Electron. Soc.*, 2004, vol. 1, pp. 641–646.

[4] S. Satoshi and H. Fumio, "Skill evaluation from observation of discrete hand movements during console operation," *J. Robot.*, vol. 2010, 2010.

[5] J. Rosen, M. Solazzo, B. Hannaford, and M. Sinanan, "Task decomposition of laparoscopic surgery for objective evaluation of surgical residents' learning curve using hidden Markov model," *Comput. Aided Surgery*, vol. 7, pp. 49–61, 2002.

[6] K. Kahol, N. C. Krishnan, V. N. Balasubramanian, S. Panchanathan, M. Smith, and J. Ferrara, "Measuring movement expertise in surgical tasks," in *Proc. 14th Annu. ACM Int. Conf. Multimedia*, 2006, pp. 719–722.

[7] Q. Zhang and B. Li, "Video-based motion expertise analysis in simulation-based surgical training using hierarchical Dirichlet process hidden markov model," in *Proc. Int. ACM Workshop Med. Multimedia Anal. Retrieval*, 2011, pp. 19–24.

[8] E. Fox, "Bayesian nonparametric learning of complex dynamical phenomena," Ph.D. thesis, MIT, Cambridge, MA, USA, 2009.

[9] D. Parikh, and K. Grauman, "Relative attributes," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 503–510.

[10] M. Schultz, and T. Joachims, "Learning a distance metric from relative comparisons," in *Proc. Advances Neural Inf. Process. Syst.*, 2004, p. 41.

[11] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar, "Attribute and simile classifiers for face verification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 29, 2009–Oct. 2, 2009, pp. 365–372.

[12] D. Parikh, A. Kovashka, A. Parkash, and K. Grauman, "Relative attributes for enhanced human-machine communication," in *Proc. AAAI Conf. Artif. Intell.*, 2012.

[13] Q. Zhang and B. Li, "Relative hidden Markov models for evaluating motion skill," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 548–555.

[14] M. Collins, "Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms," in *Proc. ACL-02 Conf. Empirical Methods Natural Language Process.*, 2002, pp. 1–8.

[15] Y. Altun, I. Tsochantaridis, and T. Hofmann, "Hidden Markov support vector machines," in *Proc. 20th Int. Conf. Mach. Learning*, 2003, vol. 20, no. 1, p. 3.

[16] A. Sloin and D. Burshtein, "Support vector machine training for improved hidden Markov modeling," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 172–188, Jan. 2008.

[17] G. Wang, D. Forsyth, and D. Hoiem, "Comparative object similarity for improved recognition with few or no examples," in *Proc. 23rd IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3525–3532.

[18] A. Kovashka, D. Parikh, and K. Grauman, "Whittlesearch: Image search with relative attribute feedback," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 2973–2980.

[19] I. Kadar and O. Ben-Shahar, "Small sample scene categorization from perceptual relations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 2711–2718.

[20] S. Guo, S. Sanner, T. Graepel, and W. Buntine, "Score-based Bayesian skill learning," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2012, pp. 106–121.

[21] P. Dangauthier, R. Herbrich, T. Minka, and T. Graepel, "Trueskill through time: Revisiting the history of chess," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 337–344.

[22] N. R. Howells, M. D. Brinsden, R. S. Gill, A. J. Carr, and J. L. Rees, "Motion analysis: A validated method for showing skill levels in arthroscopy," *Arthroscopy: J. Arthroscopic Related Surgery*, vol. 24, no. 3, pp. 335–342, 2008.

[23] H. Lin, I. Shafran, D. Yuh, and G. Hager, "Towards automatic skill evaluation: Detection and segmentation of robot-assisted surgical motions," *Comput. Aided Surgery*, vol. 11, no. 5, pp. 220–230, 2006.

[24] A. G. Gallagher, E. M. Ritter, H. Champion, G. Higgins, M. P. Fried, G. Moses, C. D. Smith, and R. M. Satava, "Virtual reality simulation for the operating room: proficiency-based training as a paradigm shift in surgical skills training," *Ann. Surgery*, vol. 241, no. 2, p. 364, 2005.

[25] J. Rosen, J. Brown, L. Chang, M. Sinanan, and B. Hannaford, "Generalized approach for modeling minimally invasive surgery as a stochastic process using a discrete Markov model," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 3, pp. 399–413, Mar. 2006.

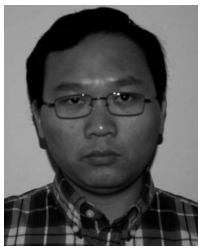
[26] K. Kahol, N. C. Krishnan, V. N. Balasubramanian, S. Panchanathan, M. Smith, and J. Ferrara, "Measuring movement expertise in surgical tasks," in *Proc. 14th Annu. ACM Int. Conf. Multimedia*, 2006, pp. 719–722.

[27] S.-k. Jun, P. Singhal, M. Sathianarayanan, S. Garimella, A. Eddib, and V. Krovi, "Evaluation of robotic minimally invasive surgical skills using motion studies," in *Proc. Workshop Performance Metrics Intell. Syst.*, 2012, pp. 198–205.

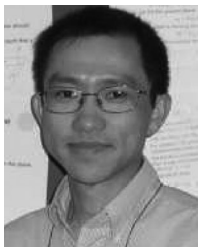
[28] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.* [Online]. *41(1)*, pp. 164–171. Available: http://www.jstor.org/stable/2239727

[29] B. Juang and L. Rabiner, "The segmental *k*-means algorithm for estimating parameters of hidden Markov models," *IEEE Trans. Acoustics, Speech Signal Process.*, vol. 38, no. 9, pp. 1639–1641, Sep. 1990.

[30] N. Merhav and Y. Ephraim, "Maximum likelihood hidden Markov modeling using a dominant sequence of states," *IEEE Trans. Signal Process.*, vol. 39, no. 9, pp. 2111–2115, Sep. 1991.
[31] D. P. Bertsekas, "Constrained optimization and Lagrange multiplier methods," in *Computer Science and Applied Mathematics*, Boston, MA, USA: Academic, vol. 1, 1982.
[32] C.-C. Chang and C.-J. Lin. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* [Online]. 2, pp. 27:1–27:27, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
[33] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[34] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, no. 2, pp. 107–123, 2005.
[35] Q. Zhang, L. Chen, Q. Tian, and B. Li, "Video-based analysis of motion skills in simulation-based surgical training," in *Proc. Int. IS&T/SPIE Electron. Imaging*, 2013, p. 86670A.
[36] T. L. Nwe, S. W. Foo, and L. C. De Silva, "Speech emotion recognition using hidden Markov models," *Speech Commun.*, vol. 41, no. 4, pp. 603–623, 2003.
[37] B. Schuller, G. Rigoll, and M. Lang, "Hidden Markov model-based speech emotion recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2003, vol. 2, pp. II–1.
[38] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognit.*, vol. 44, no. 3, pp. 572–587, 2011.
[39] A. Tarasov and S. J. Delany, "Benchmarking classification models for emotion recognition in natural speech: A multi-corporal study," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. Workshops*, 2011, pp. 841–846.
[40] K. H. Kim, S. Bang, and S. Kim, "Emotion recognition system using short-term monitoring of physiological signals," *Med. Biol. Eng. Comput.*, vol. 42, no. 3, pp. 419–427, 2004.
[41] H. Mori, T. Satake, M. Nakamura, and H. Kasuya. (2008). UU database: A spoken dialogue corpus for studies on paralinguistic information in expressive conversation. *Proc. Int. Conf. Text, Speech Dialogue*, pp. 427–434 [Online]. Available: http://uudb.speech-lab.org/
[42] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young. (1994). Large vocabulary continuous speech recognition using htk. *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, vol. 2, pp. II/125–II/128 [Online]. Available: http://htk.eng.cam.ac.uk/



Qiang Zhang received the BS degree in electronic information and technology from Beijing Normal University, Beijing, China, in 2009 and the PhD degree in computer science from Arizona State University, Tempe, Arizona in 2014. Since 2014, he has been with Samsung, Pasadena, CA, as a staff research scientist in computer vision. His research interests include image/video processing, computer vision and machine vision, specialized in sparse learning, face recognition, and motion analysis. He is a student member of the IEEE.



Baoxin Li (S'97-M'00-SM'04) received the PhD degree in electrical engineering from the University of Maryland, College Park, in 2000. He is currently an associate professor of computer science and engineering with Arizona State University, Tempe. From 2000 to 2004, he was a senior researcher with SHARP Laboratories of America, Camas, WA, where he was the technical Lead in developing SHARP's HiIMPACT Sports technologies. From 2003 to 2004, he was also an adjunct professor with the Portland State University, Portland, OR. He holds nine issued US patents. His current research interests include computer vision and pattern recognition, image/video processing, multimedia, medical image processing, and statistical methods in visual computing. He won the SHARP Laboratories' President Award twice, in 2001 and 2004. He also received the SHARP Laboratories' Inventor of the Year Award in 2002. He received the National Science Foundation's CAREER Award from 2008 to 2009. He is a senior member of the IEEE.