Sparse Hidden Markov Models for Surgical Gesture Classification and Skill Evaluation

Lingling Tao[†], Ehsan Elhamifar[†],
Sanjeev Khudanpur[†], Gregory D. Hager[‡], and René Vidal[*]

BME[*], CS[‡] and ECE[†] Dept., Johns Hopkins University, Baltimore MD, 21218, USA

Abstract. We consider the problem of classifying surgical gestures and skill level in robotic surgical tasks. Prior work in this area models gestures as states of a hidden Markov model (HMM) whose observations are discrete, Gaussian or factor analyzed. While successful, these approaches are limited in expressive power due to the use of discrete or Gaussian observations. In this paper, we propose a new model called sparse HMMs whose observations are sparse linear combinations of elements from a dictionary of basic surgical motions. Given motion data from many surgeons with different skill levels, we propose an algorithm for learning a dictionary for each gesture together with an HMM grammar describing the transitions among different gestures. We then use these dictionaries and the grammar to represent and classify new motion data. Experiments on a database of surgical motions acquired with the da Vinci system show that our method performs on par with or better than state-of-the-art methods.This suggests that learning a grammar based on sparse motion dictionaries is important in gesture and skill classification.

Keywords: surgical skill evaluation; surgical gesture classification; time series classification; sparse dictionary learning; hidden Markov models

1 Introduction

Direct instruction by an expert is arguably the most effective means of learning the art of surgery. However, due to reductions in the amount of one-on-one teaching [1], an expert may not always be available to oversee and guide residents and fellows. Robotic surgery systems, such as the da Vinci robot, provide a well-instrumented, controlled laboratory for recording surgical performance. Such recordings can be used to model surgeon expertise and help understand how to reflect this expertise back upon students in the form of teaching and training.

Prior Work. One approach to modeling surgical expertise is to use global measurements of the task, such as the time to completion [2, 3], the speed and number of hand movements [2], the distance travelled [3], force and torque signatures [3–5], etc. These methods are generally easy to implement, but lack a detailed description of the surgical procedure. Another approach is to use statistical models to decompose a surgical task into a series of pre-defined surgical gestures or *surgemes* [6–11]. For example, in a suturing task, the surgemes can be 'insert a needle', 'grab a needle', 'position a needle', etc. Notice that these surgemes often appear in some pattern, e.g., one surgeme often follows another one, or several surgemes form a motif. This is analogous to what we see in natural language, where the grammar constrains the generation of words. In the case of surgery, however, we know neither the words nor the grammar. Thus, we need to develop algorithms for discovering the grammar and for classifying gestures and skill. Hidden Markov models (HMMs) provide an excellent framework for doing this. The simplest approach is to model each surgeme as the state of an HMM and to vector-quantize the observations from each surgeme into discrete symbols [6, 7]. Alternatively, one can model the observations from each surgeme using a Gaussian [8]. However, parameter learning may not be robust when the data is high-dimensional because of the large number of parameters to be estimated. To address this issue, [9] combines Gaussian HMMs (G-HMMs) with Linear Discriminant Analysis (LDA) [12], while [10] proposes several variations of HMMs, such as Factor Analyzed HMMs (FA-HMMs), and Switched Linear Dynamical Systems (SLDSs), which model the observations as being generated from a lower-dimensional latent space. However, the observation model is still Gaussian, which may not be rich enough to capture the variability of complex gestures. While one could use Gaussian mixture models (GMMs) [11] or mixtures of factor analyzers (MFAs) to describe more complex motions, this would again results in a large number of parameters to be estimated.

Paper Contributions. To achieve a richer observation model, without dramatically increasing the number of parameters to be estimated, in this paper we propose to use a sparse model as the HMM observation model. More specifically, we propose to model each observation as a sparse linear combination of elements from a dictionary of atomic surgical motions associated with a specific surgeme. Therefore, the observations from each surgeme live in a union of K-dimensional subspaces, one subspace per choice of K out of N atoms. While other models such as MFAs also represent the data with a union of subspaces, the number of parameters in our model is much smaller because we assume that the coefficients are sparse, and so only a few dictionary elements are used to represent a given observation. As a consequence, our observation model is more expressive than a Gaussian or a FA, but the number of parameters does not grow as rapidly as in the case of GMMs or MFAs. In principle, the parameters of the proposed sparse HMM can be learned using an expectation maximization algorithm. However, the expectation step cannot be computed in closed form. We thus propose an approximate parameter learning algorithm based on a sparse dictionary learning technique called KSVD [13]. We then show that surgeme classification can be done using the Viterbi algorithm [14], as in the case of G-HMMs. Experiments show that combining HMMs with sparse dictionary learning improves gesture and skill classification and achieves stable performance for various sparsity levels.

2 Sparse HMMs for Surgical Gesture and Skill Classification

Given a surgery trial $\{\boldsymbol{y}_t \in \mathbb{R}^D\}_{t=1}^T$, the goal of gesture classification is to assign a surgeme label $s_t \in \{1, \dots, S\}$ to each frame, $\boldsymbol{y}_t$, while the goal of skill classifi-

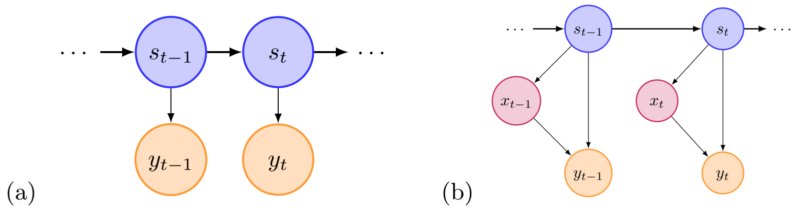


Fig. 1: Graphical models for standard HMMs (a) and HMMs with latent variables (b).

cation is to assign a skill level $z \in \{1, \ldots, L\}$ to the entire trial, $\{\boldsymbol{y}_t \in \mathbb{R}^D\}_{t=1}^T$. In this paper, we propose to model the trials using a sparse hidden Markov model (S-HMM). In this model, the surgeme label s_t is an unobserved hidden state, which is modeled as a Markov process characterized by the transition probability $q_{s's} = p(s_t = s | s_{t-1} = s')$. The observation at time t, $\boldsymbol{y}_t$, depends on the hidden state s_t via the emission probability density $p(\boldsymbol{y}_t | s_t)$.

In standard HMMs, which will be briefly reviewed in §2.1 and are illustrated in Fig. 1a, $p(\boldsymbol{y}_t | s_t)$ is assumed to be a Gaussian or a mixture of Gaussians. The parameters of this model can be learned using the Baum Welch algorithm [15], which is based on Expectation Maximization (EM). Given the model parameters, the hidden states can be inferred using the Viterbi algorithm.

In the proposed S-HHMs, which will be discussed in §2.2 and are illustrated in Fig. 1b, the observation $\boldsymbol{y}_t$ is as a sparse linear combination of elements from a dictionary of motion words. Therefore, $\boldsymbol{y}_t$ also depends on another hidden variable, namely the sparse coefficients $\boldsymbol{x}_t$. In §2.3, we show that parameter learning for this model is more difficult than for G-HMMs, because the E-step cannot be computed in closed form. We thus propose an approximate learning approach based on sparse dictionary learning [13]. In §2.4, we show that surgeme classification can be done by combining a Viterbi-like algorithm with sparse coding [16, 17]. Finally, in §2.5 we show how to use S-HMMs for skill classification.

2.1 Prior Work on Gesture and Skill Classification using HMMs

Much of the prior work on surgical gesture and skill classification uses HMMs [18, 6–11]. The main difference between different approaches is in how they model the emission probability density $p(\boldsymbol{y}_t | s_t)$. For example, [7] vector-quantizes the observations into discrete symbols, while [8] assumes a Gaussian distribution $p(\boldsymbol{y}_t | s_t = s) \equiv \mathcal{N}(\boldsymbol{u}_s, \boldsymbol{\Sigma}_s)$. These methods can leverage standard learning and inference algorithms. However, parameter learning is not robust with high-dimensional data due to the large number of parameters that need to be learned. Moreover, high-dimensional data often lie in low-dimensional subspaces, and this is not directly captured by a Gaussian distribution with an arbitrary covariance.

To address this problem, [9] uses a Gaussian model combined with LDA [12]. Alternatively, one can use Probabilistic PCA (PPCA) [19, 20] or Factor Analysis (FA) [21], as suggested in [10]. As illustrated in Fig. 1b, these models introduce a low-dimensional latent variable $\boldsymbol{x}_t \in \mathbb{R}^d$, where $d \ll D$, and model the observations as $\boldsymbol{y}_t = \boldsymbol{A}_{s_t}\boldsymbol{x}_t + \boldsymbol{u}_{s_t} + \boldsymbol{e}_t$, where $A_{s_t} \in \mathbb{R}^{D\times d}$, $\boldsymbol{u}_{s_t} \in \mathbb{R}^D$, and $\boldsymbol{x}_t$ and $\boldsymbol{e}_t$ are independent Gaussians distributed as $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\mathcal{N}(\boldsymbol{0}, \Sigma_{s_t})$, respectively. In PPCA, $\Sigma_{s_t} = \sigma_{s_t}^2\boldsymbol{I}$, while in FA, $\Sigma_{s_t} = \mathrm{diag}(\sigma_{1,s_t}^2, \dots, \sigma_{D,s_t}^2)$. [10] proposes efficient learning and inference methods for this model and shows that using a low-dimensional model improves gesture classification results. This is possible, in part, because one can marginalize over the latent variables and obtain the emission probabilities in closed form as $p(\boldsymbol{y}_t|s_t = s) \equiv \mathcal{N}(\boldsymbol{u}_s, \boldsymbol{A}_s\boldsymbol{A}_s^T + \Sigma_s)$. Therefore, PPCA-HMMs and FA-HMMs are particular cases of G-HMMs.

In practice, modeling the data with a single subspace (as done by PPCA and FA) might not capture the distribution of the data for complex surgemes. To address this issue, one can use a mixture of low-dimensional subspaces, as proposed in [11]. This can be done by using MFAs, whose density can be written as:

$$p(\boldsymbol{y}_t|s_t = s, \boldsymbol{x}_t = \boldsymbol{x}) \equiv \sum_{i=1}^{M} c_{si}\mathcal{N}(\boldsymbol{A}_{si}\boldsymbol{x} + \boldsymbol{u}_{si}, \Sigma_{si}), \tag{1}$$

where $c_{si} \in [0, 1]$ and $\sum_{i=1}^{M} c_{si} = 1$. In other words, c_{si} is the probability that $\boldsymbol{y}_t$ belongs to the i-th FA in the mixture. The drawbacks of using MFAs are that there are many parameters to be learned and that one needs to specify a priori the number of mixture components M and the dimension d of each FA.

2.2 Proposed Sparse Hidden Markov Model

In this section, we propose a new HMM that uses multiple subspaces to model the observations from each surgeme (thus being more general than single-subspace HMMs), but enforces sparsity constraints on the latent variables (thus rendering the parameter learning problem more robust). More specifically, we use recent advances in sparse dictionary learning and model the observation at time t as $\boldsymbol{y}_t = \boldsymbol{D}_{s_t}\boldsymbol{x}_t + \boldsymbol{e}_t$, where $\boldsymbol{D}_{s_t} \in \mathbb{R}^{D\times N}$ is an over-complete dictionary ($D < N$), $\boldsymbol{x}_t \in \mathbb{R}^N$ is a sparse latent variable, i.e., it has only a few nonzero entries, and $\boldsymbol{e}_t$ is independent Gaussian noise distributed as $\mathcal{N}(\boldsymbol{0}, \sigma_{s_t}^2\boldsymbol{I})$. As a result, the distribution of $\boldsymbol{y}_t$ given the latent variables is given by

$$p(\boldsymbol{y}_t|s_t = s, \boldsymbol{x}_t = \boldsymbol{x}) \equiv \mathcal{N}(\boldsymbol{D}_s\boldsymbol{x}, \sigma_s^2\boldsymbol{I}). \tag{2}$$

The key difference between our approach and MFAs in (1) is that, instead of fixing the number of mixture components M and their dimensions, we let the dictionary $\boldsymbol{D}_s$ be over-complete, but we choose a few columns of the dictionary using a sparse latent variable $\boldsymbol{x}_t$. This allows us to have an exponentially large number of subspaces to choose from and also to automatically pick the dimension of the low-dimensional subspace through the number of nonzero elements of $\boldsymbol{x}_t$.

To have a sparse latent variable, we use a Laplace prior on the distribution of $\boldsymbol{x}_t$ for each hidden state where

$$p(\boldsymbol{x}_t|s_t = s) \equiv \left(\frac{\lambda_s}{2}\right)^N \exp\left(-\lambda_s\|\boldsymbol{x}\|_1\right), \tag{3}$$

with a parameter $\lambda_s > 0$.

2.3 Parameter Learning in S-HMMs

Given N trials $\{\boldsymbol{y}^j_{1:T_j}\}_{j=1}^J$ from many surgeons with different skill levels and their surgeme labels $\{s^j_{1:T_j}\}_{j=1}^J$, our goal is to learn an S-HMM for these data. The parameters to be learned are the transition probabilities $Q = \{q_{s,s'}\}_{s,s'=1,\ldots,S}$ and the parameters for each surgeme model $\Theta_s = (\boldsymbol{D}_s, \sigma_s^2, \lambda_s)$, for $s = 1, \ldots, S$.

Since the surgeme labels are given, the transition probabilities can be directly computed from the frequency of surgeme transitions, and the remaining parameters can be learned separately from data corresponding to each surgeme s. Since $p(\boldsymbol{y}_t|s_t)$ depends on the hidden variable $\boldsymbol{x}_t$, we can use the EM algorithm to maximize the log-likelihood of the observations corresponding to surgeme s, $\mathcal{L}_{\Theta_s} = \sum_{j,t:s_t^j=s} \log p_{\Theta_s}(\boldsymbol{y}_t^j|s_t^j = s)$ w.r.t. the parameters Θ_s.

In the E-step we need to compute the expectation of the complete log-likelihood w.r.t. the posterior of $\boldsymbol{x}_t$, given the current parameters $\hat{\Theta}_s$, i.e.,

$$E_{\hat{\Theta}_s}(\mathcal{L}_{\Theta_s}) = \sum_{j,t:s_t^j=s} \int_{\boldsymbol{x}_t^j} \log p_{\Theta_s}(\boldsymbol{y}_t^j, \boldsymbol{x}_t^j|s_t^j = s) p_{\hat{\Theta}_s}(\boldsymbol{x}_t^j|\boldsymbol{y}_t^j, s_t^j = s) d\boldsymbol{x}_t^j. \tag{4}$$

However, this expression cannot be computed in closed form as in the case of G-HMMs. Following [22], we approximate the posterior as $p_{\hat{\Theta}_s}(\boldsymbol{x}_t^j|\boldsymbol{y}_t^j, s_t^j = s) = \delta(\hat{\boldsymbol{x}}_t^j)$, where $\hat{\boldsymbol{x}}_t^j = \arg\max_{\boldsymbol{x}} p_{\hat{\Theta}_s}(\boldsymbol{x}|\boldsymbol{y}_t^j, s_t^j = s) = \arg\max_{\boldsymbol{x}} p_{\hat{\Theta}_s}(\boldsymbol{y}_t^j|\boldsymbol{x}, s_t^j = s) p_{\hat{\Theta}_s}(\boldsymbol{x}|s_t^j = s)$. Therefore, the E-step reduces to the following ℓ_1-minimization problem

$$\hat{\boldsymbol{x}}_t^j = \arg\min_{\boldsymbol{x}} \; \hat{\lambda}_s \|\boldsymbol{x}\|_1 + \frac{1}{2\hat{\sigma}_s^2} \|\boldsymbol{y}_t^j - \hat{\boldsymbol{D}}_s \boldsymbol{x}\|^2, \tag{5}$$

which can be solved using a sparse coding algorithm such as Basis Pursuit [16]. With this approximation, we obtain the following approximate expectation

$$\begin{aligned} E_{\hat{\Theta}_s}(\mathcal{L}_{\Theta_s}) &\approx \sum_{j,t:s_t^j=s} \log\left(p_{\Theta_s}(\boldsymbol{y}_t^j, \hat{\boldsymbol{x}}_t^j|s_t^j = s)\right) = \sum_{j,t:s_t^j=s} \log\left(p_{\Theta_s}(\boldsymbol{y}_t^j|\hat{\boldsymbol{x}}_t^j, s_t^j = s) p_{\Theta_s}(\hat{\boldsymbol{x}}_t^j|s_t^j = s)\right) \\ &= \sum_{j,t:s_t^j=s} -\lambda_s \|\hat{\boldsymbol{x}}_t^j\|_1 - \frac{1}{2\sigma_s^2}\|\boldsymbol{y}_t^j - \boldsymbol{D}_s \hat{\boldsymbol{x}}_t^j\|_2^2 + N \log(\frac{\lambda_s}{2}) - \frac{D}{2}\log(2\pi\sigma_s^2). \end{aligned} \tag{6}$$

In the M-step we need to maximize the above quantity w.r.t. Θ_s, which gives:

$$\hat{\boldsymbol{D}}_s = \sum_{j,t:s_t^j=s} \boldsymbol{y}_t^j \hat{\boldsymbol{x}}_t^{jT} \Big(\sum_{j,t:s_t^j=s} \hat{\boldsymbol{x}}_t^j \hat{\boldsymbol{x}}_t^{jT}\Big)^{-1}, \; \hat{\lambda}_s = \frac{\sum_{j,t:s_t^j=s} N}{\sum_{j,t:s_t^j=s} \|\hat{\boldsymbol{x}}_t^j\|_1}, \; \hat{\sigma}_s^2 = \frac{\sum_{j,t:s_t^j=s} \|\boldsymbol{y}_t^j - \hat{\boldsymbol{D}}_s \hat{\boldsymbol{x}}_t^j\|_2^2}{\sum_{j,t:s_t^j=s} D}. \tag{7}$$

Interestingly, the above approximate EM algorithm involves an E-step where the MAP estimate of $\boldsymbol{x}_t^j$ is calculated given $\hat{\boldsymbol{D}}_s$ and an M-step where the dictionary $\boldsymbol{D}_s$ is updated based on $\hat{\boldsymbol{x}}_t^j$. This is analogous to the method of optimal directions (MOD) in sparse dictionary learning, which alternates between finding the sparse coefficients and updating the dictionary [23]. This opens the door to using faster and more accurate sparse dictionary learning methods that update $\boldsymbol{x}_t^j$ and $\boldsymbol{D}_s$ jointly. One such algorithm is KSVD [13], which uses the ℓ_0-seminorm instead of the ℓ_1-norm in the cost function. Since λ_s and σ_s^2 are not involved in KSVD, one can compute them afterwards by cross validation. We call this approximate learning method KSVD-HMM, and this is our method of choice.

2.4 Surgeme Classification using S-HMMs

Given a trial $\{\boldsymbol{y}_t\}_{t=1}^T$ and the S-HMM parameters $q_{s,s'}$ and Θ_s, $s, s' = 1, \dots, S$, our goal is to infer the sequence of surgeme labels $\{s_t\}_{t=1}^T$. In standard HMMs this can be done by the Viterbi algorithm [14], where one maximizes the joint probability of the hidden states and the observations

$$(\hat{s}_{1:T}) = \text{argmax } p(s_{1:T}|\boldsymbol{y}_{1:T}) = \text{argmax } p(s_{1:T}, \boldsymbol{y}_{1:T}). \tag{8}$$

However, unlike the Gaussian, PPCA and FA models discussed in §2.1, the marginal probability $p(\boldsymbol{y}_t|s_t)$ cannot be computed in closed form because $\boldsymbol{x}_t$ has a Laplace distribution. Nonetheless, in this section we show that the inference problem can still be solved using a dynamic programming approach. More specifically, we can write the following recursion

$$\begin{aligned}
\alpha_t(s,\boldsymbol{x}) &\triangleq \max_{s_{1:t-1},\boldsymbol{x}_{1:t-1}} p(s_{1:t-1}, \boldsymbol{x}_{1:t-1}, s_t = s, \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{y}_{1:t}) \\
&= \max_{s',\boldsymbol{x}'} \{ \max_{s_{1:t-2},\boldsymbol{x}_{1:t-2}} p(s_{1:t-2}, \boldsymbol{x}_{1:t-2}, s_{t-1} = s', \boldsymbol{x}_{t-1} = \boldsymbol{x}', s_t = s, \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{y}_{1:t}) \} \\
&= \max_{s',x'} \{ \max_{s_{1:t-2},\boldsymbol{x}_{1:t-2}} p(\boldsymbol{y}_t|\boldsymbol{x}_t = \boldsymbol{x}, s_t = s) \cdot p(\boldsymbol{x}_t = \boldsymbol{x}|s_t = s) \cdot q_{s',s} \\
&\qquad\qquad \cdot p(s_{1:t-2}, \boldsymbol{x}_{1:t-2}, s_{t-1} = s', \boldsymbol{x}_{t-1} = \boldsymbol{x}', \boldsymbol{y}_{1:t-1}) \} \\
&= p(\boldsymbol{y}_t|\boldsymbol{x}_t = \boldsymbol{x}, s_t = s) \cdot p(\boldsymbol{x}_t = \boldsymbol{x}|s_t = s) \cdot \max_{s',\boldsymbol{x}'} \{ q_{s',s} \cdot \alpha_{t-1}(s', \boldsymbol{x}') \}.
\end{aligned} \tag{9}$$

From the last equality, one can see that the value of $\boldsymbol{x}_t$ only affects the first two probabilities and has no influence on the last term. Now, since the number of states S is finite, for each s we can find the $\hat{\boldsymbol{x}}_s$ that maximizes $p(\boldsymbol{y}_t|\boldsymbol{x}, s)p(\boldsymbol{x}|s)$. That is, $\hat{\boldsymbol{x}}_s = \arg\min_{\boldsymbol{x}} \ \lambda_s \|\boldsymbol{x}\|_1 + \frac{1}{2\sigma_s^2} \|\boldsymbol{y}_t - \boldsymbol{D}_s \boldsymbol{x}\|^2$, which can be found using Basis Pursuit [16] or Orthogonal Matching Pursuit (OMP) [17]. Since the learning algorithm uses KSVD, which in turn uses OMP, we also use OMP here.

2.5 Skill Classification using S-HMMs

For skill classification, we model the data from different skill levels with different S-HMMs and classify a new trial by finding the model that gives the highest log-likelihood. More specifically, for each expertise level, we learn an S-HMM using KSVD-HMM, the approximate learning algorithm described in §2.3. This gives us three models, $\mathcal{M}_e$, $\mathcal{M}_i$ and $\mathcal{M}_n$ corresponding to expert, intermediate and novice. Given a test trial $\{\boldsymbol{y}_t \in \mathbb{R}^D\}_{t=1}^T$, the skill level z is given by:

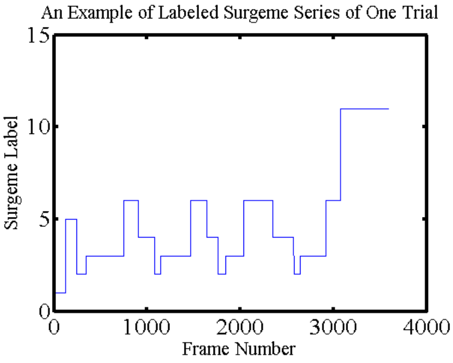
$$\hat{z} = \arg \max_{z \in \{e,i,n\}} p(\boldsymbol{y}_{1:T}, \boldsymbol{x}_{1:T}, s_{1:T}|\mathcal{M}_z). \tag{10}$$

3 Experiments

Dataset Description. To evaluate the proposed KSVD-HMM approach for surgeme classification and to compare it with other state-of-the-art methods, we use the California dataset described in [7, 24, **?**]. The dataset is acquired with the da Vinci surgical robot, which provides both kinematic data and high-resolution video data. For the experiments below, we use the kinematic data which consists

0. Idle motion
1. Reach for needle
2. Position needle
3. Insert needle / push needle through tissue
4. Move to middle with needle (left hand)
5. Move to middle with needle (right hand)
6. Pull suture with left hand
7. Pull suture with right hand
8. Orienting needle with two hands
9. Right hand assisting left while pulling suture
10. Loosen more suture
11. End of trial

(a)



(b)

Fig. 2: List of surgemes (a) and sample surgeme time series (b).

Table 1: Best surgeme classification percentages obtained by different methods

		MFA-HMM	KSVD-HMM	FA-HMM(1)	SLDS(1)	FA-HMM(3)	HLDA-HMM	SLDS(3)
SU	Setup 1	76.4	**81.1**	70.2	74.8	78.2	74.1	80.8
	Setup 2	59.8	**67.8**	N/A	N/A	57.2	N/A	67.1
NP	Setup 1	74.2	76.1	64.3	72.3	71.0	65.0	**77.6**
	Setup 2	46.6	59.3	N/A	N/A	42.7	N/A	**60.0**
KT	Setup 1	76.5	82.6	77.1	78.5	**82.8**	79.9	82.0
	Setup 2	65.1	65.7	N/A	N/A	**67.0**	N/A	66.0

of 78 variables describing the motion (velocity, rotation angle, position, etc.) of the master and slave robots. The dataset consists of 39, 26 and 36 trials, respectively, from three different tasks: suturing, needle passing and knot tying. Each task is performed by 8 surgeons of three expertise levels: expert, intermediate and novice. Typically each surgeon has around of $3 - 5$ trials for each task.

According to the definition of surgemes in [24], as listed in Fig. 2a, each of the trials is manually segmented into a sequence of surgemes, and the surgeme labels provide us the ground truth for surgeme classification. Each time series data consists, in general, of 11 different surgemes, as shown in Fig. 2b.

Experiment Setup. We create two different test setups. Setup 1 is the *leave-one-supertrial-out* setup, where we leave one trial from each one of the users out for testing, and use the remaining trials for training. Setup 2 is the *leave-one-user-out* setup, where we leave all the trials from one user out for testing and use the remaining trials for training.

Surgeme Classification. We evaluate the surgeme classification performance of KSVD-HMM and compare it to that of MFA-HMM on three datasets. For KSVD-HMM we vary the sparsity level K and for MFA-HMM we vary the number of subspaces M and the dimensions d. The parameters σ and λ in KSVD-HMM are obtained by cross validation. The best results for each method using each of the two setups are shown in Table 1. We also compare our results to those in [10] for FA-HMM, HLDA-HMM, and SLDS. Notice that in [10], each surgeme can be represented by 1 state HMM, or by a left-to-right HMM with 3 states.

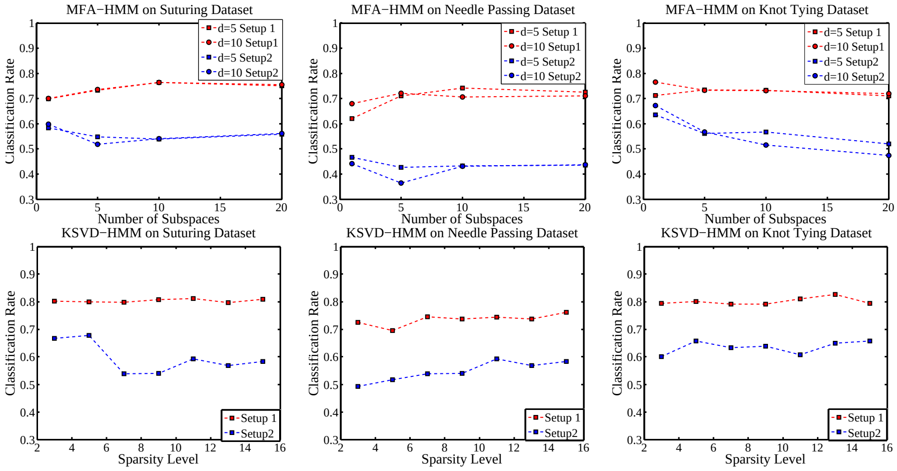


Fig. 3: Top: Surgeme classification rates of MFA-HMM as a function of the number of subspaces M and the subspace dimension d. Bottom: Surgeme classification results of KSVD-HMM as a function of the sparsity level K. Both methods are evaluated on three surgery tasks: suturing, needle passing and knot tying.

The first case is analogous to our model of one state per surgeme. The second case corresponds to a more sophisticated method in which surgemes are further decomposed into smaller components. The numbers 1 and 3 in parentheses after FA-HMM and SLDS indicate the number of HMM states used by [10] to represent each surgeme. We can see from Table 1 that, for suturing and setup 2 of knot tying task, KSVD-HMM outperforms even a more sophisticated 3-state HMM model, or 3-state SLDS model where both latent variables at time t depend on the latent variables at time $t-1$. For the other tasks, KSVD-HMM performs slightly worse than the 3-state SLDS, but is still better than any 1-state HMM model based on Gaussian models or SLDS. Overall, the proposed KSVD-HMM method performs on par with or better than state-of-the-art techniques.

Notice also that the performance of all methods decreases from setup 1 to setup 2. This is because in setup 2 all the trials from the same surgeon are excluded, which makes the classification problem more challenging because we only use the trials of the other surgeons.

Fig. 3 shows the effect of changing the parameters of each dictionary learning algorithm on the classification performance. From the plots in Fig. 3 we can see that the classification rates of MFA-HMM for different values of $M = 1, 5, 10, 15, 20$ and $d = 5, 10$ are in general lower than those of KSVD-HMM. Also note that for KSVD-HMM, the classification rates do not change much as we change the sparsity level $K = 3, 5, 7, 9, 11, 13, 15$. Thus, KSVD-HMM is less dependent on model selection than MFA-HMM, which makes it more favorable for classification using dictionary learning algorithms.

Skill Classification. We now evaluate the skill classification performance of KSVD-HMM and compare it to that of MFA-HMM. Table 2 shows the best classification results achieved by KSVD-HMM and MFA-HMM. For setup 1, where

Table 2: Best skill classification percentages obtained by MFA-HMM and KSVD-HMM.

	Suturing		Needle Passing		Knot Tying	
Setup	MFA-HMM	KSVD-HMM	MFA-HMM	KSVD-HMM	MFA-HMM	KSVD-HMM
Setup 1	92.3	97.4	76.9	96.2	86.1	94.4
Setup 2	38.5	59.0	46.2	26.9	44.4	58.3

we have different data from the same user in both training and testing, KSVD-HMM performs clearly better than MFA-HMM. Notice also that for setup 2, where we exclude the trials of the same surgeon, we obtain much lower classification rates than for setup 1. In addition to the fact that we have excluded the trials of the same surgeon, another reason for this drop is the relatively small number of overall training data in the dataset, which does not allow us to capture well a specific skill level. For example, in the suturing data, we only have two experts, two intermediates and four novices. We are currently collecting larger datasets to be able to better evaluate the sensitivity of different methods.

4 Conclusion

We have proposed a new model called sparse HMMs for the classification of gestures and skill in surgical tasks. In this model, the observations are expressed as linear combinations of elements from a dictionary with sparse coefficients. The experiments show that the proposed methods achieve stable performance for various sparsity levels and perform on par with or better than the state of the art. Future work involves evaluation of the proposed methods on larger datasets.

Acknowledgment

This project was supported by grants NSF 0931805 and NSF 0941362. The authors thank Carol Reiley for providing the annotated dataset, and Balakrishnan Varadarajan for numerous discussions on the subject.

References

1. Barden, C., Specht, M., McCarter, M., Daly, J., Fahey, T.: Effects of limited work hours on surgical training. Obstetrical & Gynecological Survey **58**(4) (2003) 244–245
2. Datta, V., Mackay, S., Mandalia, M., Darzi, A.: The use of electromagnetic motion tracking analysis to objectively measure open surgical skill in laboratory-based model. Journal of the American College of Surgery **193** (2001) 479–485
3. Judkins, T., Oleynikov, D., Stergiou, N.: Objective evaluation of expert and novice performance during robotic surgical training tasks. Surgical Endoscopy **1**(4) (2008)
4. Richards, C., Rosen, J., Hannaford, B., Pellegrini, C., Sinanan, M.: Skills evaluation in minimally invasive surgery using force/torque signatures. Surgical Endoscopy **14** (2000) 791–798
5. Yamauchi, Y., Yamashita, J., Morikawa, O., Hashimoto, R., Mochimaru, M., Fukui, Y., Uno, H., Yokoyama, K.: Surgical skill evaluation by force data for endoscopic sinus surgery training system. In: MICCAI. (2002) 44–51

6. Rosen, J., Hannaford, B., Richards, C., Sinanan, M.: Markov modeling of minimally invasive surgery based on tool/tissue interaction and force/torque signatures for evaluating surgical skills. IEEE Trans. Biomedical Eng. **48**(5) (2001) 579–591
7. Reiley, C.E., Hager, G.D.: Task versus subtask surgical skill evaluation of robotic minimally invasive surgery. In: MICCAI. (2009) 435–442
8. Rosen, J., Solazzo, M., Hannaford, B., Sinanan, M.: Task decomposition of laparoscopic surgery for objective evaluation of surgical residents' learning curve using hidden Markov model. Computer Aided Surgery **7**(1) (2002) 49–61
9. Varadarajan, B., Reiley, C., Lin, H., Khudanpur, S., Hager, G.: Data-derived models for segmentation with application to surgical assessment and training. In: MICCAI. (2009) 426–434
10. Varadarajan, B.: Learning and inference algorithms for dynamical system models of dextrous motion. PhD thesis, Johns Hopkins University (2011)
11. Leong, J., Nicolaou, M., Atallah, L., Mylonas, G., Darzi, A., Yang, G.: HMM assessment of quality of movement trajectory in laparoscopic surgery. In: MICCAI. (2006) 752–759
12. Lin, H.C., Shafran, I., Murphy, T.E., Bruckstein, A.M., Okamura, A.M., Yuh, D.D., Hager, G.D.: Automatic detection and segmentation of robot-assisted surgical motions. In: MICCAI. (2005) 802–810
13. Aharon, M., Elad, M., Bruckstein, A.M.: K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. on Signal Processing **54**(11) (2006) 4311–4322
14. Forney Jr, G.D.: The Viterbi algorithm. Proceedings of the IEEE **61**(3) (1973)
15. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann. Math. Statist **41**(1) (1970) 164–171
16. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20** (1998) 33–61
17. Tropp, J.A., Gilbert, A.C.: Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans on Information Theory **53**(12) (2007) 4655–4666
18. Dosis, A., Bello, F., Gillies, D., Undre, S., Aggarwal, R., Darzi, A.: Laparoscopic task recognition using hidden Markov models. Studies in Health Technology and Informatics **111** (2005) 115–122
19. Tipping, M., Bishop, C.: Probabilistic principal component analysis. Journal of the Royal Statistical Society **61**(3) (1999) 611–622
20. Tipping, M., Bishop, C.: Mixtures of probabilistic principal component analyzers. Neural Computation **11**(2) (1999) 443–482
21. G. J. McLachlan, D. Peel, R.W.B.: Modelling high-dimensional data by mixture of factor analyzers. Computational Statistics and Data Analysis **41** (2003) 379–388
22. Olshausen, B.A., Field, B.J.: Sparse coding with an overcomplete basis set: a strategy employed by V1? Vision Research (1997)
23. Engan, K., Aase, S.O., Husoy, J.H.: Method of optimal directions for frame design. IEEE International Conference on Acoustics, Speech, and Signal Processing (1999)
24. Reiley, C.E., Lin, H.C., Varadarajan, B., Vagolgyi, B., Khudanpur, S., Yuh, D.D., Hager, G.D.: Automatic recognition of surgical motions using statistical modeling for capturing variability. In: Medicine Meets Virtual Reality. (2008) 396–401