A Framework for Unsupervised Online Human Reaching Motion Recognition and Early Prediction

Ruikun Luo[1] and Dmitry Berenson[1]

***Abstract*— This paper focuses on recognition and prediction of human reaching motion in industrial manipulation tasks. Several supervised learning methods have been proposed for this purpose, but we seek a method that can build models on-the-fly and adapt to new people and new motion styles as they emerge. Thus, unlike previous work, we propose an unsupervised online learning approach to the problem, which requires no offline training or manual categorization of trajectories. Our approach consists of a two-layer library of Gaussian Mixture Models that can be used both for recognition and prediction. We do not assume that the number of motion classes is known a priori, and thus the library grows if it cannot explain a new observed trajectory. Given an observed portion of a trajectory, the framework can predict the remainder of the trajectory by first determining what GMM it belongs to, and then using Gaussian Mixture Regression to predict the remainder of the trajectory. We tested our method on motion-capture data recorded during assembly tasks. Our results suggest that the proposed framework outperforms supervised methods in terms of both recognition and prediction. We also show the benefit of using our two-layer framework over simpler approaches.**

I. Introduction

Recognition of human motion is useful for human-robot interaction tasks, especially for human-robot collaboration in a shared workspace. In previous work, it has been shown that predicting human motion allows for more fluid executions of robot motions in a shared workspace [1]. An early prediction of where the human will move for a given task allows computing the predicted workspace occupancy, which the robot can avoid when generating its own motion.

In this paper, we focus on recognition and prediction of human reaching motions in industrial manipulation tasks, e.g. a human and a robot assembling components in a shared workspace. Human motion recognition and prediction is well studied in this context. However, most previous work adopts supervised learning methods which require an offline training process and manually-labeled training data. Such models implicitly depend on the accuracy of human labeling. However, the labels assigned in this context usually describe the purpose of the motion, and not its geometric features. Yet it is the geometric features that are the most important for predicting the human's future motion. Also, if the human changes the way they perform a given task or a new human, with a different method of doing the task, is observed, pre-trained models will not be able to predict the new style of motion. We seek a framework that can build models on-the-fly and apdat to new people and new motion styles as they emerge. Thus, unlike previous work, we propose an unsupervised online learning framework. To our knowledge, our proposed framework is the only unsupervised online learning framework for human reaching motion prediction and recognition.

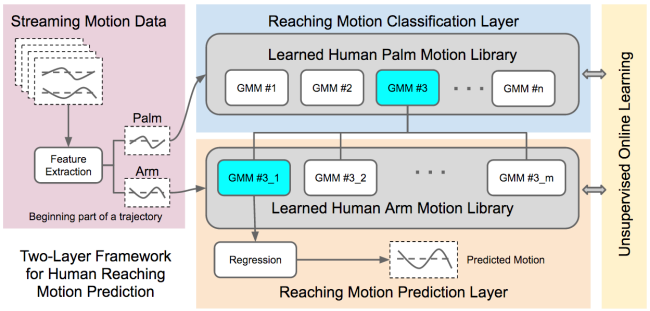


Fig. 1. A two-layer framework for human reaching motion prediction. The framework consists of a reaching motion classification layer and a reaching motion prediction layer. The first layer consists of a human *palm* motion library. The second layer consists of a set of human *arm* motion libraries, where each human *arm* motion library links to a class of palm motion in the human *palm* motion library. The two libraries are learned by the proposed unsupervised online learning algorithm.

To model human motion we use a two-layer library of Gaussian Mixture Models (GMMs) (see Fig. 1). GMMs are chosen as the library elements because they can conform to arbitrary trajectories and because they can be used as generative models. The first layer constructs GMMs for the human's palm position, and the second for the positions of the human's joint centers. Instead of manually grouping trajectories into pre-determined classes and learning a GMM for each class (as was done in [1]), our unsupervised learning approach iteratively clusters trajectories based on a similarity measure, and each cluster of trajectories corresponds to a GMM that models those trajectories.

Given a new observed trajectory, our unsupervised online learning algorithm either updates an existing GMM's parameters (using an incremental EM algorithm [2]) or initializes a new GMM if the current GMM library cannot "explain" the new trajectory. This allows us to generalize the library to new motion classes on-the-fly (e.g. if a new human is being observed or the task changes). Our framework also accounts for noise (i.e. atypical reaching motions performed in response to a disturbance) through the use of a membership-proportional prior for each GMM in the library.

Our framework can be used to recognize human trajectories, i.e. to determine which previous trajectories are

[1]Ruikun Luo and Dmitry Berenson are with the Robotics Engineering Program, Worcester Polytechnic Institute, Worcester, MA 01609, US rluo@wpi.edu, dberenson@cs.wpi.edu. This work is supported in part by the Office of Naval Research under Grant N00014-13-1-0735 and by the National Science Foundation under Grant IIS-1317462.

similar to the one currently observed. It can also predict human reaching trajectories. Given an observed portion of a trajectory, the framework can predict the remainder of the trajectory by first recognizing the trajectory, thus determining what GMM it belongs to, and then using Gaussian Mixture Regression (GMR) to predict the remainder of the trajectory.

Our contributions are as follows:

1) We propose an unsupervised online learning algorithm for human motion recognition.
2) We propose a two-layer framework for human motion prediction based on the proposed unsupervised online learning algorithm.

We tested our method on motion-capture data recorded during assembly tasks. Our results show that the proposed framework outperforms supervised methods that label trajectories according to the task being performed in terms of both recognition and prediction.

This paper is organized as follows: Section II will introduce related work. Section III will show the overview of the proposed framework. Section IV and V will present our proposed unsupervised learning algorithm and two-layer framework for human reaching motion prediction. Section VI shows the experimental results. Finally we conclude in Section VII.

II. Related Work

Our work contributes to the field of human motion prediction for human-robot collaborations. Most previous work in this area uses supervised learning for human motion recognition and prediction. In [3], [4], [5], the authors propose different types of feature representations of human motions for use inside a supervised learning framework. In [1], Mainprice et al. used GMM to learn human reaching motions. In [6], Mainprice et al. used Inverse Optimal Control (IOC) to learn a cost function under which demonstrated trajectories are optimal and use that cost function to do interactive re-planning to predict human reaching motion. Unlike their work, we focus on prediction of different types of human motions where classes of motion are not known a priori. In [7], Sun et al. used two-layered maximum-entropy Markov model (MEMM) for human activity detection from RGBD images. In [8], Koppula et al. used a Markov random field (MRF) to model both human activities and object affordance. They considered human activities as high level activities and sub-activities. Similar to that work, in [9], [10], Koppula et al. used Conditional Random Fields (CRFs) to model human activities and object affordance to predict human motions. This work has been recently extended in [11] to predict high-dimensional trajectories. In their work, the model will anticipate a human's sub-activity and then predict the motion trajectory for this sub-activity. Unlike that work, which considers the task-level prediction for known tasks, we consider the problem of early motion prediction without supervision, i.e. when the tasks have not been defined a priori. In our work, we recognize the observed part of a human's motion and then predict the remainder of this trajectory. The early prediction of human motion is useful for a robot to react quickly to human motion in a human-robot collaboration task. The above methods are all supervised learning algorithms, which require an offline training process and a batch of labeled training data. Unlike these previous works, we consider unsupervised online learning, which requires no manually-labeled data and offline training process.

Kulić et al. proposed an online incremental learning of full-body motion primitives in [12]. They segmented the human motion into several motion primitives and then use a Hidden Markov Model (HMM) to model both the structure of the primitives and each motion primitive. Unlike their method, we model sets of trajectories using a library of GMMs. We are interested in modeling human reaching motion, which is not clearly separable into primitives. Calinon et al. proposed incremental learning of gestures for humanoid robot imitation in [2]. The incremental training of a GMM is done by the human manually moving the robot. We use the same incremental EM method proposed in their work as part of our algorithm. However, unlike their work, our framework is given motions corresponding to different tasks and can cluster the motions into different classes. Unsupervised online learning GMMs has been studied in speech recognition [13], [14]. Unlike these works rely on a well-trained background GMM, our proposed unsupervised online learning algorithm requires no offline training.

III. Framework Overview

Our proposed framework for early prediction of human reaching motion is shown in Fig. 1. It consists of a reaching motion classification layer and a reaching motion prediction layer. The first layer is the current learned human *palm* motion library, which contains a set of GMMs where each GMM represents a class of human reaching motion. The second layer contains a set of human reaching *arm* motion libraries, where each human *arm* motion library links to a GMM in the human *palm* motion library in the first layer. The human motion libraries are learned by the proposed unsupervised online learning algorithm.

Given the observed part of the human's trajectory, the first layer classifies that motion into one of the GMMs in the current learned human *palm* motion library. The second layer uses the arm joint center position representation to classify that trajectory into a specific motion style (a GMM in the learned human *arm* motion library) and does regression to predict the remainder of the trajectory. The human motion libraries in both layers are learned using our proposed unsupervised online learning method (Fig. 2). After the framework observes the entire trajectory, the unsupervised online learning algorithm updates the human motion libraries.

IV. Unsupervised Online Learning Algorithm

In this section we introduce the core component of our framework: the unsupervised online learning algorithm for human motion recognition. The proposed algorithm is shown in Fig. 2. The algorithm builds and maintains a human

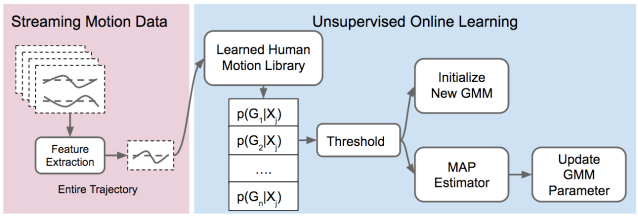


Fig. 2. Data flow for the unsupervised online learning algorithm.

motion library that consists of multiple GMMs where each GMM G_i represents a class of human motion. Given a motion trajectory X_j, the algorithm will first calculate the probabilities of this trajectory given each GMM - $p(X_j|G_i)$ for $i = 1, 2, ..$ and then calculate the posterior probability $p(G_i|X_j)$ (we will explain in Section IV-B). If all the posterior probabilities are smaller than a specified threshold, the framework will use this trajectory X_j to initialize a new GMM and store it in the human motion library. If some posterior probabilities are larger than that threshold, the algorithm will classify (maximum a posteriori estimation) this trajectory into a GMM class G_k with the highest probability $p(G_k|X_j)$. Then the algorithm will update the parameters of the GMM G_k. This approach is used at both levels of our framework.

A. Gaussian mixture model for human motion

Each GMM in the human motion library represents a class of human motion. G_i for $i = 1, 2, 3, ...$ represents each GMM in the library. X_j for $j = 1, 2, 3, ...$ represents a given human motion trajectory. X_j is an $L \times D$ matrix where L is the number of postures of a trajectory and D is the number of feature dimensions of the trajectory. In this paper, we consider three types of features to represent human postures: 1) palm position(PP), 2) arm joint center positions(AJCP), 3) arm configurations(AC). The feature comparison experiment in Section VI-A will show comparisons between these representations and the reason why we only use the first two feature representations in our framework.

Each GMM G_i is a combination of K multivariate gaussians gc_k for $k = 1, 2, 3, ..., K$. Let ξ_j^l be a vector that concatenates the time index l and the posture (e.g. a vector of AJCP). The probability of a posture ξ_j^l in GMM G_i represented by K multivariate gaussians is given by:

$$p(\xi_j^l|G_i) = \sum_{k=1}^{K} p(gc_k|G_i)p(\xi_j^l|gc_k, G_i)$$

where ξ_j^l is the lth row vector of X_j, representing the lth posture of trajectory X_j. $p(gc_k|G_i) = \pi_k$ (we will use π_k in the following section) is the prior probability of component gc_k in G_i. The the probability of ξ_j^l given gc_k and G_i is defined as follows:

$$\begin{aligned} p(\xi_j^l|gc_k, G_i) &= \mathcal{N}(\mu_k, \Sigma_k) \\ &= \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} e^{-\frac{1}{2}(\xi_j^l-\mu_k)^T\Sigma_k^{-1}(\xi_j^l-\mu_k)} \end{aligned}$$

where $\{\mu_k, \Sigma_k\}$ are the mean and covariance parameters of the Gaussian component gc_k. Thus the probability of

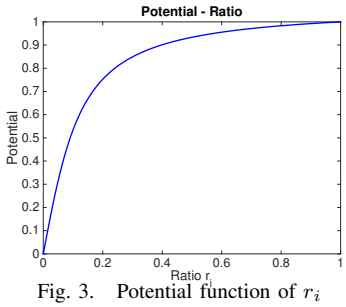


Fig. 3. Potential function of r_i

trajectory X_j in G_i is defined as follows:

$$p(X_j|G_i) = \prod_{l=1}^{L} p(\xi_j^l|G_i) \quad (1)$$

Using GMMs to represent human motion has an important advantage: we can use the parameters not only for classification, but also to do GMR, which can be used for prediction as shown in [15].

B. Threshold setup

Previous work ([1]) used Maximum likelihood estimation (MLE) to estimate the class label. This assumption makes sense because the number of motion classes is fixed and they do not need to compare different trajectories. However, in our proposed unsupervised online learning algorithm, we need to decide a threshold to determine whether a trajectory is used for initializing a new GMM or for updating an existing GMM's parameters. This threshold is a constant regardless of trajectory length, so we need to define a likelihood of each trajectory that is not as sensitive to trajectory length as (1).

Unlike (1), we assume the geometric mean of the postures' probability densities given G_i is the probability density of trajectory X_j in G_i, shown as follows:

$$p(X_j|G_i) = \sqrt[L]{\prod_{l=1}^{L} p(\xi_j^l|G_i)} \quad (2)$$

The geometric mean can balance the probability density with the length of the trajectory.

As an unsupervised online learning algorithm, the proposed algorithm can capture some noisy motions and build GMMs for these motions. The prior distribution of the GMMs thus should not be the uniform distribution. In this paper, we propose a prior distribution we call a "ratio prior", which can also be interpreted as a "regulariser":

$$p(G_i) = \frac{f(r_i)}{\sum_{i=1}^{M} f(r_i)} \quad (3)$$

where M is the current number of GMMs, r_i is the ratio between the number of trajectories classified in G_i and the total number of the trajectories observed so far. We define $f(r_i) = \arctan(10r_i)/\arctan 10$ as the potential function shown in Fig. 3. The potential will drop quickly as the ratio decreases to 0 and will increase smoothly as the ratio increases to 1. Thus GMMs with small numbers of trajectories will be assigned small values of the prior and GMMs with significant numbers of trajectories will be

Algorithm 1: Random Trajectory Generation

Input : Trajectory $X \in \mathbb{R}^{T \times D}$
Δ: Maximum distance to X
Precompute: $A =$ finite difference matrix (Eqn 5)
$R^{-1} = (A^T A)^{-1}$
$Q = R^{-1}$ with each column scaled such that the maximum elements is $1/L$
begin
  Generate dfference matrix δX where each column vector $\theta_i \sim \mathcal{N}(0, R^{-1})$ for $i = 1, 2, 3, .., D$
  while $\texttt{DTW}(\texttt{X}, \texttt{X} + \delta\texttt{X}) > \Delta$ **do**
    $\delta X = Q \delta X$
  end
end

assigned similar large values of prior. The ratio prior can be treated as a "denoising" function in order to reduce the influence of the noisy motions. Note that, at the beginning of the experiment, there will be a small number of GMMs and each GMM will have a small number of trajectories. The normalization of the prior ensures that each GMM receives a similar prior because there should be no prior information for each GMM at the beginning. In Section VI, we show that this ratio prior outperforms a uniform prior and MLE (no prior). The uniform prior is defined as $p(G_i) = 1/M$.

Combining (2) and (3), the posterior probability distribution of X_j is as follows:

$$\begin{aligned} p(G_i|X_j) &\propto p(G_i)p(X_j|G_i) \\ &= p(G_i)\sqrt[L]{\textstyle\prod_{l=1}^{L} p(\xi_j^l|G_i)} \end{aligned}$$

As the product of the probability is too small to represent accurately, we need to calculate the log of the probability. The log-likelihood of $p(G_i|X_j)$ can be computed as follows:

$$\log(p(G_i|X_j)) = \frac{1}{L}\sum_{l=1}^{L} \log p(\xi_j^l|G_i) + \log p(G_i) \qquad (4)$$

C. Initializing a GMM from single trajectory

In general, initialization of a new GMM requires a set of training data and uses K-means and expectation-maximization (EM) algorithms to compute the prior, mean and covariance matrix of each multivariate Gaussian component. The number of training trajectories and the variance of the trajectory data will influence the generality of the GMM. If the number of training trajectories is small or the training data is redundant, the GMM variance will be very low and all other trajectories will get near-zero probability given this GMM. This problem is especially acute when we try to generate a GMM from a single trajectory. For our framework, we need a way to generate a GMM from a single trajectory such that the variance is not too low. Thus we propose a Random Trajectory Generation (RTG) algorithm to generate random trajectories that are close to a given trajectory. The generated trajectories and the given trajectory can be used as training data to initialize a new GMM using the standard method in [2].

Similar to the STOMP algorithm [16], which sampled trajectories to estimate a gradient for optimization, our RTG algorithm also uses a finite differencing matrix A, which is an $(L+2) \times L$ matrix that, when multiplied by the position vector θ, produces accelerations $\ddot{\theta}$:

$$A = \begin{bmatrix} 1 & 0 & 0 & & 0 & 0 & 0 \\ -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & & 0 & 0 & 0 \\ & \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & & 1 & 0 & 0 \\ 0 & 0 & 0 & & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & & 0 & 1 & -2 \\ 0 & 0 & 0 & & 0 & 0 & 1 \end{bmatrix} \qquad (5)$$

where L is the length of the trajectory and position vector θ is the column vector of the postures of a given trajectory X. The idea of RTG is to generate a random difference trajectory δX and iteratively reduce the difference scale until the distance between the generated trajectory $X + \delta X$ and the given trajectory X is smaller than some given value. The RTG algorithm is show in Algorithm 1. The covariance matrix $R^{-1} = (A^T A)^{-1}$ and normalization matrix Q can ensure that the generated trajectory in each iteration keeps the same goal and start position. $Q = R^{-1}$, and each column vector is scaled such that the maximum element is $1/L$. Q is used to reduce the difference of the generated trajectory iteratively. Note that the maximum distance Δ can be used to control how close the generated trajectory is to the observed one. This value can help control the covariance of the initialized new GMM. The difference between trajectories is calculated using Dynamic Time Warping (DTW) [17], using Euclidean distance as the distance metric.

D. Update GMM parameters

When a given trajectory X has the log posterior probabilities $\log(p(G_i|X))$ (as in (4)) larger than the threshold, the algorithm uses MAP estimation to assign a GMM ID to this trajectory as follows:

$$\hat{i} = \underset{i}{\operatorname{argmax}} \log(p(G_i|X))$$

Then we use the directed update method for incremental EM from [2] to update the parameters of GMM $\hat{i}$. Recall that $X = \{\xi^l | l = 1, 2, 3, ..., L\}$, where ξ^l is a column vector. The incremental EM algorithm assumes ξ^l as training data, thus we have L data points for the current update. Let $\tilde{L}$ represent the number of all previous data points to train this GMM. We set the current GMM $\hat{i}$'s parameters $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$ as the initial parameters $\{\tilde{\pi}_k^{(0)}, \tilde{\mu}_k^{(0)}, \tilde{\Sigma}_k^{(0)}\}_{k=1}^K$. Let $\tilde{p}_{k,l} = p(k|\xi^l)$ represent the posterior probability, where k is the kth Gaussian component. Let $\{\tilde{E}_k^{(0)} = \sum_{l=1}^{L} \tilde{p}_{k,l}^{(0)}\}_{k=1}^K$. The incremental EM procedure is then:

E-step:

$$\begin{aligned} \tilde{p}_{k,l}^{(t+1)} &= \frac{\tilde{\pi}_k^{(t)} \mathcal{N}(\xi^l; \tilde{\mu}_k^{(t)}, \tilde{\Sigma}_k^{(t)})}{\sum_{i=1}^{K} \tilde{\pi}_i^{(t)} \mathcal{N}(\xi^l; \tilde{\mu}_i^{(t)}, \tilde{\Sigma}_i^{(t)})} \\ \tilde{E}_k^{(t+1)} &= \sum_{l=1}^{L} \tilde{p}_{k,l}^{(t+1)} \end{aligned}$$

M-step:

$$\begin{aligned}
\tilde{\pi}_k^{(t+1)} &= \frac{\tilde{E}_k^{(0)}+\tilde{E}_k^{(t+1)}}{L+\tilde{L}} \\
\tilde{\mu}_k^{(t+1)} &= \frac{\tilde{E}_k^{(0)}\tilde{\mu}_k^{(0)}+\sum_{l=1}^{L}\tilde{p}_{k,l}^{(t+1)}\xi^l}{\tilde{E}_k^{(0)}+\tilde{E}_k^{(t+1)}} \\
\tilde{\Sigma}_k^{(t+1)} &= \frac{\tilde{E}_k^{(0)}(\tilde{\Sigma}_k^{(0)}+(\tilde{\mu}_k^{(0)}-\tilde{\mu}_k^{(k+1)})(\tilde{\mu}_k^{(0)}-\tilde{\mu}_k^{(k+1)})^T)}{\tilde{E}_k^{(0)}+\tilde{E}_k^{(t+1)}} \\
&+ \frac{\sum_{l=1}^{L}\tilde{p}_{k,l}^{(t+1)}(\xi^l-\tilde{\mu}_k^{(t+1)})(\xi^l-\tilde{\mu}_k^{(t+1)})^T}{\tilde{E}_k^{(0)}+\tilde{E}_k^{(t+1)}}
\end{aligned}$$

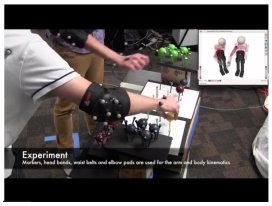
The two steps iterate until convergence.

V. Human Reaching Motion Early Prediction

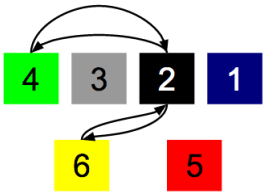
The purpose of early human motion prediction is to regress the remainder of a human's trajectory based on the observed part of the trajectory. We decompose the human motion early prediction problem into two steps: 1) human motion early recognition and 2) human motion trajectory regression. As we focus on the application of human motion prediction for human-robot collaboration tasks, we require regressing the whole arm trajectory (not only the palm trajectory) in order to compute the human's workspace occupancy. However, the results in Table I show that the proposed unsupervised online learning algorithm using PP features significantly outperforms the algorithm using AJCP features in the *recognition* task. As early recognition is vital for the early prediction problem, we propose a two layer framework for human reaching motion early prediction (Fig. 1). The first layer uses PP features and the second layer uses AJCP features. This two-layer framework can take the advantages of PP features (better recognition performance) and can still model the whole arm trajectory. Both layers use the proposed unsupervised online learning algorithm to build their motion libraries. The first layer builds a palm motion library and the second layer builds an arm motion library for each palm motion class as shown in Fig. 1. Note that as an online system, the framework will observe human motion trajectories one-by-one and human motion postures from each trajectory one-by-one. At the beginning of each trajectory, the framework will only do early prediction based on the current learned models. After observing this trajectory, the framework will then update the human motion libraries using the method in Fig. 2 for each layer.

Fig. 1 shows the data flow for the human reaching motion early prediction. The early prediction consists of the following steps:

1) *Feature extraction*: The framework observes the beginning part of the human motion and extracts two types of features: PP and AJCP.
2) *Human motion early recognition*: The first layer of the framework takes the PP features and uses MAP to estimate the palm motion class ID i (GMM ID in the library). The second layer takes the AJCP features and uses MAP to estimate the human arm motion class ID j in the arm motion library for the ith palm motion class.



(a)



(b)

Fig. 4. (a) Experiment setup. Two human subjects are performing the assembly task side by side. We only considered the motions for the "active" human on the right. (b) Reaching motions we considered in this paper were moving balls between location 2 and 4, and location 2 and 6.

3) *Human motion trajectory regression*: The second layer computes the regressed trajectory X' using [2] with the GMM parameters of the jth human arm motion class in the arm motion library for the ith palm motion class.
4) *Normalize regressed trajectory*: Move the regressed trajectory such that the beginning posture of the regressed trajectory overlaps with the end posture of the observed trajectory.

Though it may be possible to first predict the palm trajectory and then compute the Inverse Kinematics (IK) solutions on the predicted palm trajectory to generate the arm trajectory, we do not take this approach because the human arm has redundant DoFs. There are no unique IK solutions for a given palm pose, and it is difficult to predict which IK solution the human will choose.

VI. Experiments

To test our framework, we require an experiment where human subjects perform a variety of reaching motions in an industrial context. In this context, we can expect that a human will spend most of their time performing the given task in the same way, but will occasionally change their reaching motion to avoid another worker who has temporarily entered their space (e.g. to retrieve a part). Thus we also wish to see how our framework performs in the presence of noise (i.e. reaching motions that are atypical for the task at hand). In previous work [1] we observed that a human in isolation would produce the same stereotypical motion with low variance when asked to perform a repetitive reaching task. To produce a more realistic set of trajectories that includes noise we devised an experiment where two humans are performing an assembly task side-by-side (see Fig. 4(a)). We devised the assembly task so that the humans occasionally need to reach into their partner's workspace, therefore forcing their partner to change their reaching strategy to avoid them. The "active" human, the person on the right in our experiments, is the one whose motion we wish to predict. The other human is there to generate disturbances that force the active human to produce atypical (i.e. "noisy") trajectories.

The assembly task required the "active" human to move balls between location 2 and location 4 and between location 2 and location 6 as shown in Fig. 4(b). We used a VICON system to capture the human motions. Human subjects wore a suit consisting of nine markers and three rigid plates which were placed following the standards used in the field of

TABLE I

FEATURE REPRESENTATION COMPARISON

	PP	AJCP	AC
precision(%)	98.6 ± 3.4	98.1 ± 4.1	90.8 ± 7.3
recall(%)	98.6 ± 3.4	97.9 ± 5.2	87.0 ± 13.1

biomechanics [18]. Our suit consists of rigid marker plates attached to a belt, a headband and an elbow pad, a marker on the back of the hand, two on each side of the wrist, two on either side of the shoulder, and two markers straddling the sternum and xyphoid process. The VICON system runs at 100 fps. We used recordings from 3 pairs of human subjects doing the assembly task and each pair performed the assembly task 6 times. Thus we have 18 sets of experiment data. There were a total of 254 trajectories captured from the three "active" human subjects. The average number of frames in each trajectory is 107. The algorithm is implemented in MATLAB. The average runtime to process a trajectory (update the parameters or add a new GMM model) is 0.1s and the average runtime for one call of the prediction process is 0.0036s.

In the proposed two-layer framework, we setup the parameters as follows: To initialize a new GMM, we set the $\Delta = 10$ and generate 5 random trajectories for PP, however, we set $\Delta = 45$ and generate 10 random trajectories for AJCP. We set the threshold as -8 in the first layer and -108 in the second layer. The parameters were found by manual tuning.

A. Feature comparison experiment

In this section, we ran leave-one-out experiments to compare different human motion feature representations using supervised GMMs. In each round of the leave-one-out experiment, we used 1 of 18 sets of the experiment data as the testing data and other 17 sets as training data. We considered three types of feature representations: PP, AJCP and AC. The AJCP are positions for the right arm's palm, wrist, elbow and shoulder, which are recorded by our motion capture system. In this paper, we only considered the joint angles of the arm in the AC feature, which we obtain through IK on the set of markers. The dimensions of each feature representation are 3, 12, and 9, respectively. Table I shows the performance for each type of feature using supervised GMMs. Both the PP and AJCP outperform the AC and have no significant difference between them. Although the AC tries to reduce the influence of different body types, people with different body type will perform the same motion with different joint angles. Thus we only used PP features and AJCP features in the rest of the experiments.

B. Human reaching motion trajectory recognition

In this section, we compared our proposed unsupervised online learning algorithm (UOLA) with supervised GMMs (S-GMM) and semi-supervised online GMMs (SSO-GMM). For the S-GMM, we only used data from one pair of the human subjects' first run of the assembly task as training data and tested on the rest of the data as streaming input in order to simulate real world circumstance. For the SSO-GMM, we used the same training and testing data as supervised GMMs. The training data is used to initialize the 4 GMMs (4 classes of motions). Given a trajectory, the SSO-GMM classified this trajectory into one of the 4 GMMs (e.g. the ith GMM) and use this trajectory to update the ith GMM's parameters using the incremental EM algorithm in [2]. For our proposed algorithm (UOLA), there was no training data and we used the whole dataset as streaming test data. For each algorithm, we tested both the PP feature and AJCP feature. For our proposed algorithm, we also tested on different types of priors: ratio prior, uniform prior, and no prior (i.e. MLE). We ran the experiments 100 times for each model and each feature representation. Table II shows the performance of each model. It shows that all the models have better performance using the PP feature. The number of GMM shows that our algorithm is not over-clustering the dataset.

TABLE II

HUMAN REACHING MOTION TRAJECTORY RECOGNITION

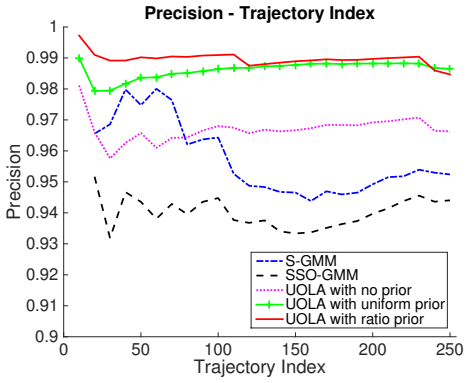
	Precision(%)	Recall(%)	# GMM
S-GMM (PP)	95.3 ± 1.5	94.2 ± 2.5	4
S-GMM (AJCP)	78.1 ± 5.7	68.0 ± 10.6	4
SSO-GMM (PP)	94.4 ± 1.9	92.3 ± 4.1	4
SSO-GMM (AJCP)	72.7 ± 6.4	30.0 ± 4.5	4
UOLA (PP, MLE)	$\mathbf{97.6 \pm 3.2}$	$\mathbf{96.9 \pm 5.6}$	9.4 ± 2.5
UOLA (AJCP, MLE)	84.8 ± 5.2	68.6 ± 13.8	14.6 ± 4.5
UOLA (PP, uniform)	$\mathbf{99.3 \pm 2.2}$	$\mathbf{98.8 \pm 4.5}$	17.0 ± 3.3
UOLA (AJCP, uniform)	86.8 ± 4.8	74.1 ± 14.2	17.5 ± 5.1
UOLA (PP, ratio)	$\mathbf{98.8 \pm 2.1}$	$\mathbf{98.6 \pm 4.0}$	16.3 ± 2.9
UOLA (AJCP, ratio)	85.9 ± 4.2	68.5 ± 13.4	17.4 ± 5.6

Fig. 5 shows the performance changes along with the trajectory indices streaming into the system. Here we only considered the PP feature as it always outperforms the AJCP feature. The figure shows that both proposed unsupervised online learning algorithm with ratio prior and uniform prior consistently outperform the baselines. The figure also shows that MAP estimation with ratio prior and uniform prior outperform the MLE estimation and have no significant difference between each other.

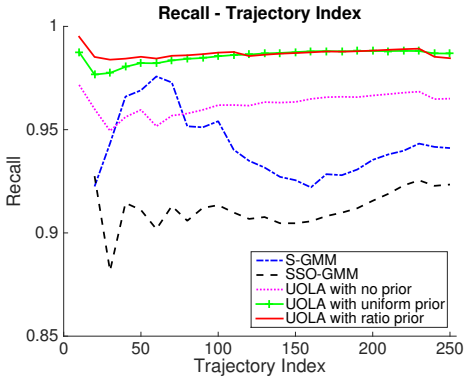
The precision and recall are computed as the average precision and recall over all 4 classes of motions. For the supervised or semi-supervised methods, the number of GMM is fixed and each GMM has the same label as the training data that trains this GMM. For the unsupervised methods, the number of GMM is not fixed and we are actually doing clustering of the trajectories. When we compute the precision and recall, the trajectories in the unsupervised GMMs will take the label of the ground truth label of the majority of the trajectories in that GMM. The experiment is set up this way to show that we cluster trajectories for the same task together, even though we do not know what the tasks are.

C. Human reaching motion trajectory early recognition

Early recognition of a human motion trajectory is the first step of human motion early prediction. The performance of early recognition is crucial for the human motion trajectory prediction, as using the correct generative model is necessary for trajectory prediction. In this section, we compared our

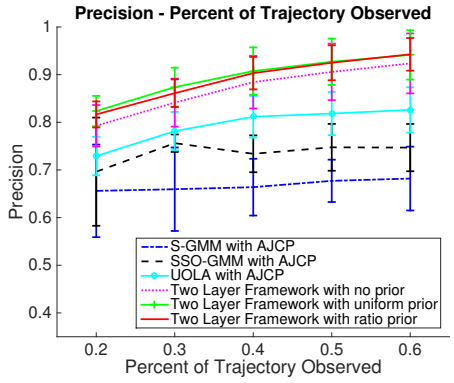


(a)

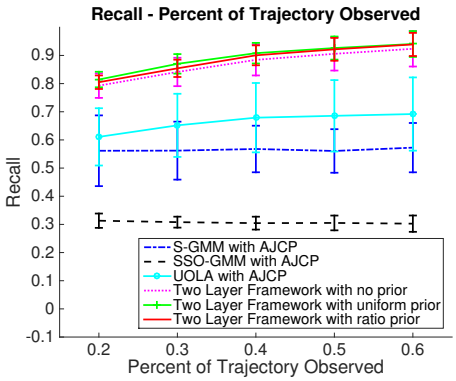


(b)

Fig. 5. (a) Precision changes vs. trajectory indices, (b) Recall changes vs. trajectory indices. The proposed unsupervised online learning algorithm with ratio prior and uniform prior consistently outperform the baselines.



(a)



(b)

Fig. 6. (a) Relationship between average precision and percent observed, (b) Relationship between average recall and percent observed. The proposed two-layer framework with uniform prior and ratio prior consistently outperform the baselines.

proposed two-layer framework using different types of prior with baseline methods on early recognition. Note that our final goal is to predict the remainder of a given arm motion trajectory, so we only considered the methods that have generative models for AJCP features. The baseline methods we considered in this section are S-GMM with AJCP features, SSO-GMM with AJCP features and UOLA with AJCP features (one-layer). We tested on having observed percentages $(20\%, 30\%, 40\%, 50\%, 60\%)$ of the trajectory for each method. We used the same testing data and training data for each method as mentioned in Section VI-B. We ran each method 100 times on the dataset. Fig. 6 shows the overall performance of each method. All of our proposed two-layer framework variants outperform the baselines. The proposed two-layer framework with ratio prior and uniform prior slightly outperform the framework with no prior.

D. Human reaching motion trajectory prediction

In this section, we used the same setup as the previous section, however, we focused on the performance of motion trajectory prediction for each method. The evaluation method is the DTW distance between the predicted trajectory and the remainder of the given testing trajectory. Fig. 7 shows the relationship between the average DTW distance and the percent observed of the trajectory. All of our proposed two-layer framework variants outperform the baselines. The framework using ratio prior outperforms the framework variants using uniform prior and using no prior. Note that our proposed framework with ratio prior significantly outperforms the baselines when given a small percentage of the observed part of the trajectory (e.g. 20%, 30%). This result indicates that using our proposed framework, the robot may react more quickly to human motion than using baseline methods.

Fig. 8 shows the qualitative results for the proposed two-layer framework with ratio prior compared with S-GMM. The results shows that our proposed framework consistently outperforms the baseline. Because the early recognition of the S-GMM is not correct, the prediction is not close to the real trajectory and even goes in the wrong direction (see Fig. 8(c)). Our proposed framework gives a better prediction even when just observing the first 20% of the trajectory (see Fig. 8(e)).

VII. Conclusion

We have presented a two-layer framework for unsupervised online human reaching motion recognition and early prediction. The framework consists of a two-layer library of GMMs. The library grows if it cannot "explain" a new observed trajectory by using the proposed unsupervised online learning algorithm. Given an observed portion of a trajectory, the framework can predict the remainder of the trajectory by first determining what GMM it belongs to, and then using GMR to predict the remainder of the trajectory. The proposed unsupervised online learning algorithm requires no offline

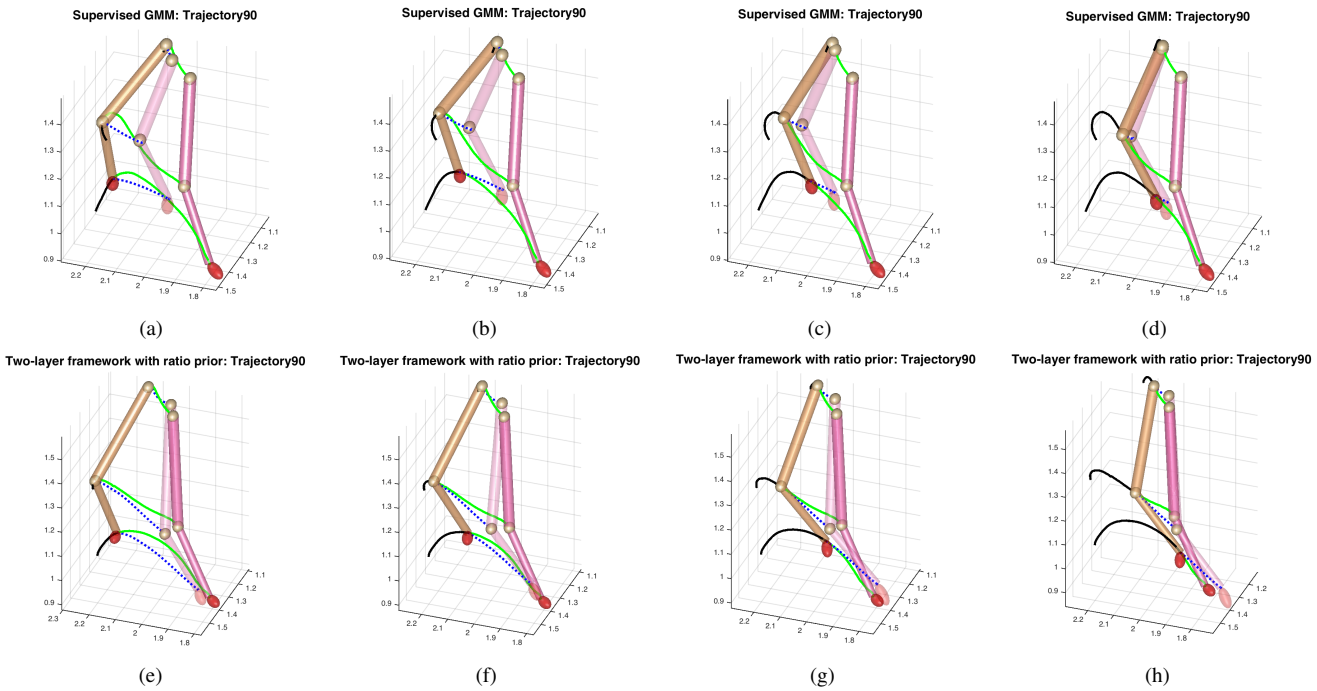


Fig. 8. A comparison of our framework to S-GMM on the 90th trajectory. The first row is using supervised GMMs and the second row is using our proposed two-layer framework with ratio prior. Each column is given different percents of observed trajectory (i.e. 20%, 30%, 40% and 50% respectively). The yellow arm is the end pose of the observed part of the trajectory, which is also the start pose of the remainder trajectory. The pink arm is the end pose of the remainder of the trajectory and the translucent pink arm is the end pose of the predicted trajectory. The black curves are the observed part of the trajectories for each joint. The green curves are the remainder trajectories for each joint, which are the ground truth. The blue doted curves are the predicted trajectories for each joint. The goal of prediction is to compute the blue dotted curves such that the distance between the blue curves and green curves are minimized.

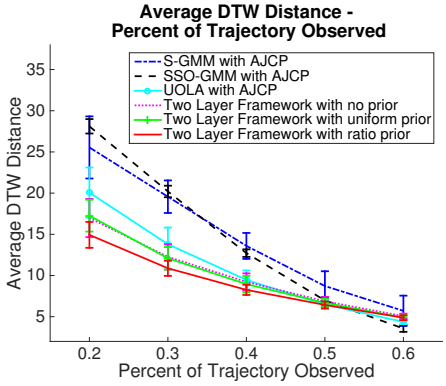


Fig. 7. Relationship between the average DTW distance and the percent observed of the trajectory for each model. Both of our proposed two-layer framework variants outperform the baselines.

training process or manual categorization of trajectories. The results show that our framework can generate models on-the-fly and adapt to new people and new motion styles as they emerge. Future work will explore how to use GMR to generate smoother predicted trajectories.

REFERENCES

[1] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *IROS*, 2013.

[2] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *HRI*, 2007.

[3] L. Xia, C.-C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *CVPRW*, 2012.

[4] Y. Zhao, Z. Liu, L. Yang, and H. Cheng, "Combing rgb and depth map features for human activity recognition," in *APSIPA ASC*, 2012.

[5] H. Zhang and L. E. Parker, "4-dimensional local spatio-temporal features for human activity recognition," in *IROS*, 2011.

[6] J. Mainprice, R. Hayne, and D. Berenson, "Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning," in *ICRA*, 2015.

[7] S. Jaeyong, P. Colin, S. Bart, and S. Ashutosh, "Unstructured human activity detection from rgbd images," in *ICRA*, 2012.

[8] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *IJRR*, 2013.

[9] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response." in *RSS*, 2013.

[10] H. Koppula and A. Saxena, "Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation," in *ICML*, 2013.

[11] Y. Jiang and A. Saxena, "Modeling high-dimensional humans for activity anticipation using gaussian process latent CRFs," in *RSS*, 2014.

[12] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *IJRR*, 2011.

[13] C. Barras, S. Meignier, and J.-L. Gauvain, "Unsupervised online adaptation for speaker verification over the telephone," in *The Speaker and Language Recognition Workshop*, 2004.

[14] Y. Zhang and M. S. Scordilis, "Effective online unsupervised adaptation of Gaussian mixture models and its application to speech classification," *Pattern Recognition Letters*, 2008.

[15] S. Calinon, "Robot programming by demonstration," in *Springer handbook of robotics*, 2008.

[16] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *ICRA*, 2011.

[17] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, 2007.

[18] G. Wu, F. C. Van der Helm, H. D. Veeger, M. Makhsous, P. Van Roy, C. Anglin, J. Nagels, A. R. Karduna, K. McQuade, X. Wang, F. Werner, and B. Buchholz, "ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion - Part II: shoulder, elbow, wrist and hand," *Journal of biomechanics*, 2005.