Probability-weighted Temporal Registration for Improving Robot Motion Planning and Control Learned from Demonstrations

Chris Bowen and Ron Alterovitz

University of North Carolina at Chapel Hill, NC 27599, USA,
{cbbowen,ron}@cs.unc.edu

Abstract. Many existing methods that learn robot motion planning task models or control policies from demonstrations require that the demonstrations be temporally aligned. Temporal registration involves an assignment of individual observations from a demonstration to the ordered steps in some reference model, which facilitates learning features of the motion over time. We introduce probability-weighted temporal registration (PTR), a general form of temporal registration that includes two useful features for motion planning and control policy learning. First, PTR explicitly captures uncertainty in the temporal registration. Second PTR avoids degenerate registrations in which too few observations are aligned to a time step. Our approach is based on the forward-backward algorithm. We show how to apply PTR to two task model learning methods from prior work, one which learns a control policy and another which learns costs for a sampling-based motion planner. We show that incorporating PTR yields higher-quality learned task models that enable faster task executions and higher task success rates.

1 Introduction

Registering a time sequence of observations to a reference model is a common subproblem in many robotics algorithms, including algorithms for learning motion planning task models and control policies from demonstrations as in Fig. 1 (e.g., [8, 14, 7]). This subproblem may be referred to as time alignment, time warping, or *temporal registration* (the term we use in this paper). Formally, temporal registration is an assignment of individual observations (from a sequence of observations) to the ordered steps in some underlying reference model. In the domain of robot learning from demonstrations, the observations correspond to the time-ordered data collected during each demonstration, where each demonstration must be temporally aligned to an underlying task model (e.g., a reference demonstration or task representation) to facilitate learning features of the motion over time from a set of demonstrations. Temporal registration abstracts away differences in execution speed both between and within demonstrated trajectories and is often a critical step to learning effective task models.

Many prior methods for learning synthesizable models rely on temporal registration using dynamic time-warping (DTW) [19] or its variants (discussed in Sec.

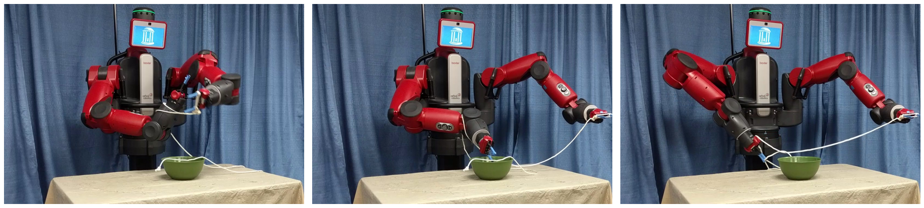
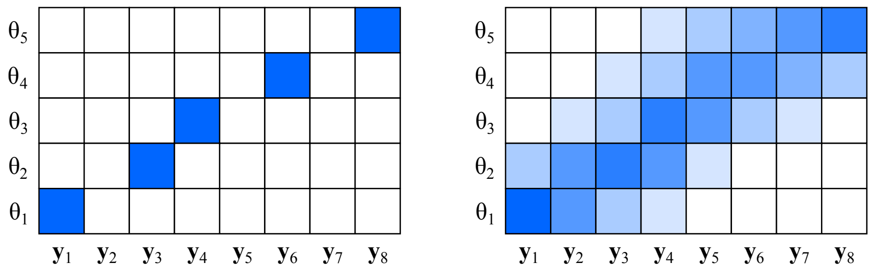


Fig. 1. The Baxter robot performing a knot-tying task learned from demonstrations. The temporal registration of demonstrations can have a significant impact on the quality of the learned task model, and better registration can improve task performance or reduce the number of required demonstrations.



(a) Traditional approach (DTW) (b) Our approach (PTR)

Fig. 2. Comparison between traditional temporal registration (e.g., using dynamic time warping (DTW)) and probability-weighted temporal registration (using our approach, PTR) to register a sequence of observations $\{\mathbf{y}_1, \ldots, \mathbf{y}_8\}$ to a reference model $\{\theta_1, \ldots, \theta_5\}$. In our approach, we register all samples to the task model, avoid degenerate registrations, and assign probabilities to the alignment of each observation, which can be utilized by a learning algorithm to improve the quality of a learned task model.

2), either as a preprocessing step (e.g., [8, 25, 23]) or interleaved with model estimation (e.g., [4, 7]). DTW can be viewed as a maximum-likelihood approach to temporal registration. However, maximum-likelihood approaches are inherently prone to becoming caught in local minima, yielding suboptimal registrations when a different initialization would produce better results. Furthermore, DTW and many of its variants may drop some or even most observations from the temporal registration, which can produce degenerate temporal registrations in which a time step of the reference model has too few observations aligned to it. Most importantly, DTW does not consider uncertainty in the temporal registration.

We introduce *probability-weighted temporal registration (PTR)*, a more general form of temporal registration that explicitly captures uncertainty in the temporal registration. Instead of assuming each observation is registered to (at most) one time step of the reference model, we instead compute probability-weighted assignments (as shown in Fig. 2). These weighted assignments can be leveraged in robot learning algorithms to yield more robust task models. Specifically, we propose to use the classical forward-backward algorithm [2] to compute probability-weighted temporal registration. This approach is not new, but we present a framework based on a tensor product graph in which DTW and the forward-backward algorithm are in fact remarkably similar in terms of implementation. Using this graphical approach, we describe a novel modification for avoiding degenerate registrations that improves registrations across methods in practical applications.

We apply PTR to two existing task learning methods from prior work, one which learns a task model for a control policy [4] and another which learns a task model that informs edge costs for a sampling-based motion planner [7]. We show that incorporating PTR yields higher-quality learned task models that enable faster task executions and higher task success rates on challenging tasks both in simulation and on the Baxter robot.

2 Related Work

Temporal registration is necessary for solving many estimation, classificiation, and clustering problems in robotics, signal processing, and other fields. The most commonly used method for temporal registration is dynamic time warping (DTW), which has been successfully applied to problems such as gesture recognition [6] and robot task learning using Guassian mixture models [8, 25] as a preprocessing step. Similar approaches have been applied to the problem of learning to manipulate deformable objects [14]. Other methods instead integrate DTW or similar methods into an iterative estimation process [4, 7]. DTW has also been used to learn and subsequently execute tasks in the presence of external perturbations [5]. Collaborative tasks add an additional cause for variable demonstration speed, which has been addressed using related modifications to DTW [1] or gradient descent to overcome the non-smooth nature of DTW [17]. That weakness of DTW, combined with the desire for a global solution rather than one which relies on local optimization, is a motivator for our work. Along these same lines, a probabilistic interpretation of DTW has been shown to improve modeling accuracy and data efficiency [18].

When viewed more specifically as the problem of registering demonstrations to an underlying hidden Markov model (HMM) [23] or semi-Markov model [21], DTW is quite similar to the well-known Viterbi algorithm [24] for inference of the hidden model variables. Work has been done to produce variants of the Viterbi algorithm for finding registrations with specific properties [22], lower risk [15], or approximations for switching linear dynamic systems (SLDSs). For SLDSs in particular, alternative approaches to estimation have been explored [20], and SLDSs have been combined with reinforcement learning to good effect [13].

Estimation of an HMM using the Viterbi algorithm in a simple expectation-maximization framework yields the Viterbi path-counting algorithm [10]. However, the classical forward-backward algorithm has nicer properties. Prior work has explored efficient forward-background algorithm computation for HMMs [27]. We explore how use of the forward-backward algorithm versus DTW or the Viterbi algorithm impacts learning and subsequent execution for HMM-based task models.

In this paper, we describe and compare existing and new variants of temporal registration on two robot learning methods from prior work [4, 7], and we find measurable improvements by using a temporal registration approach that captures uncertainty and enforces non-degeneracy constraints.

3 Problem Definition

Consider a user-provided demonstration given by a sequence $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ where each $\mathbf{y}_i$ is an observation which might be in the state space of the robot or some arbitrary feature space. It is often necessary for learning or recognition algorithms to register such a demonstration to a different sequence $\Theta = \{\theta_1, \ldots, \theta_T\}$ of *time steps* (e.g. in a learned task model) while abstracting away differences in demonstration speed. We are generally interested in a registration which minimizes some measure of loss $L(\mathbf{y}_i, \theta_t)$ between matched elements of each sequence and where $T \leq n$.

The most common approach to this problem in the domain of robotics is DTW, which produces for each θ_t an assignment $i = \iota_t$ (where ι is the Greek letter iota) corresponding to $\mathbf{y}_i$ (see Fig. 2(a)). In particular, this assignment minimizes $\sum_t^T L(\mathbf{y}_{\iota_t}, \theta_t)$ subject to a strict monotonicity constraint $\iota_t < \iota_{t+1}$. Stronger variants of this constraint may also be enforced as shown in Sec. 4.4.

However, this approach discards much of the demonstration and fails to encode uncertainty in the temporal registration. To address these weaknesses, we consider probability-weighted temporal registrations which assign to each $\mathbf{y}_i$ a weight $\omega_{i,t}$ for each θ_t (see Fig. 2(b)). These weights sum to one and so form a distribution. Here, we minimize $\sum_{i,t}^{n,T} \omega_{i,t} L(\mathbf{y}_{\iota_t}, \theta_t)$ subject to a monotonicity constraint similar to that above generalized to distributions (see Sec. 4.1).

To make these temporal registrations truly probability-weighted, we impose a restriction on the loss L, namely that it be defined as the negative log likelihood for some statistical model $\mathcal{L}(\theta_t \mid \mathbf{y}_i)$: $L(\mathbf{y}_i, \theta_t) \equiv -\log \mathcal{L}(\theta_t \mid \mathbf{y}_i)$. This model is relevant to a variety of learning approaches including those we consider in Sec. 5.

The problem we consider of temporal registration of one user-provided demonstration to a sequence of time steps readily extends to the common problem of registering multiple demonstrations. Each demonstration can be temporally registered to the model independently, the model can then be re-estimated given the resulting registrations, and the process repeats in an expectation-maximization loop until convergence, as done in other approaches (e.g., [9, 4, 7]).

4 Method

In this section, we describe a graphical approach to probability-weighted temporal registration using the forward-backward algorithm along with a practical improvement to any temporal registration method that fits into this graphical framework. In Sec. 4.1 and Sec. 4.2, we discuss two maximum-likelihood temporal registration methods: DTW and the Viterbi algorithm. We recast both approaches as shortest paths on a tensor product graph. In Sec. 4.3, we next discuss a true expectation-maximization algorithm that produces a temporal registration with probability-weighted assignments for the observations. Again, we reformulate this approach using the tensor product graph and show that it is in fact nearly the same algorithm as the Viterbi algorithm. Finally, in Sec. 4.4 we show that any of these methods can be improved by modifying the tensor product graph to enforce a non-degeneracy constraint, which is particularly valuable when integrated with robot learning and using very few demonstrations as input, as shown in Sec. 6. To our knowledge, the reformulation of these algorithms in terms of a tensor product graph is novel, as is the modification to enforce the non-degeneracy constraint. The combination of using the forward-backward algorithm and enforcing the non-degeneracy constraint gives us our complete method for probability-weighted temporal registration (PTR).

4.1 Dynamic Time Warping as a Graph Algorithm

The structure of the DTW algorithm follows the familiar dynamic programming approach of iteratively solving each subproblem in a table. In particular, let $C[t,i]$ denote the cost of the best registration of the first t time steps $\{\theta_1,\ldots,\theta_t\}$ to the first i observations $\{\mathbf{y}_1,\ldots,\mathbf{y}_i\}$. We then have for all $t \leq T$ and $i \leq n$:

$$C[0,0] = 0 \qquad C[t,0] = C[0,i] = \infty \tag{1}$$
$$C[t,i] = \min(C[t,i-1] + c_{\mathrm{ins}}(t), C[t-1,i] + c_{\mathrm{del}}(i), C[t-1,i-1] + c(t,i))$$

where $c_{\mathrm{ins}}(t)$ denotes the cost of not matching θ_t to any observation, $c_{\mathrm{del}}(i)$ the cost of not matching observation $\mathbf{y}_i$ to any time step, and $c(t,i)$ the cost of matching θ_t with $\mathbf{y}_i$. The actual registration ι may then be constructed by traversing this table.

The best temporal registration is the one which maximizes $\mathcal{L}(\iota \mid Y, \Theta)$, that is the likelihood of the entire registration given both sequences. Because the loss function we assumed depends only $\mathbf{y}_i$ and θ_t where $\iota_t = i$, it satisfies the Markov property, and thus the joint likelihood is simply $\prod_{t=1}^{T} \mathcal{L}(\theta_t \mid \mathbf{y}_{\iota_t})$, and maximizing this is equivalent to minimizing $\sum -\log \mathcal{L}(\theta_t \mid \mathbf{y}_{\iota_t})$. Substituting

$$c_{\mathrm{insert}}(t) = \infty \qquad c_{\mathrm{match}}(t,i) = -\log \mathcal{L}(\theta_t \mid \mathbf{y}_i) \tag{2}$$

in the recurrence above yields $C[t,i] = -\log \mathcal{L}(\Theta \mid Y, \iota_{1\ldots t})$ where $\iota_{t'} \leq i$. By monotonicity of log, minimizing $C[T,n]$ maximizes $\mathcal{L}(\Theta \mid Y, \iota)$.

Letting $c_{\mathrm{insert}}(t) = \infty$ ensures that every time step is registered to an observation, but much of the prior work in robotics further assumes $c_{\mathrm{delete}}(i) = 0$ [19]. That is, not every observation needs to be considered during registration, and only the best T observations contribute to model estimation. Not only does this not match the underlying model, it discards information which could otherwise be useful for robust parameter estimation. In the next section, we will consider an alternative, the Viterbi algorithm. But first, it will be convenient to recast the DTW algorithm as a graph algorithm. This view will become an important step in unifying and extending all the temporal registration algorithms we consider.

To view DTW as a graph algorithm, we first construct graphs representing both the time steps and the demonstration. The demonstration simply becomes a sequential graph of its observations (the *demonstration graph*). Time steps encode a more complex relation, but one which can be described by a hidden

(a) Bayesian network (b) Restricted transitions (time graph)

Fig. 3. Discrete-time hidden Markov model in which time steps comprise the discrete states and observations are those from the demonstrations.

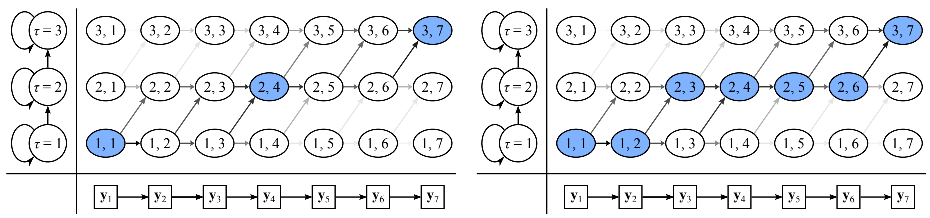


Fig. 4. (left) Tensor product of the time graph (vertical axis) and demonstration graph (horizontal axis). Darker edges are more probable and blue vertices show the DTW temporal registration. **(right)** Blue vertices show the maximum likelihood temporal registration, which corresponds to the shortest path in the graph.

Markov model (HMM). To see this, consider the time steps to be the discrete states, denoted τ_i, of a time-homogeneous hidden Markov model (HMM) with observation distributions parameterized by $\{\theta_1, \ldots, \theta_m\}$ and transition probabilities given by $p_{t,t'}$. To enforce a monotonicity constraint, we restrict the state transitions as shown in Fig. 3(b), and this forms the *time graph*. Finally, we then consider the tensor product [26] of these two graphs (see Fig. 4).

Note that unlike the time graph, we omit self-edges in the demonstration graph, which similar to setting $c_{\text{insert}}(t) = \infty$ above, ensures that every time step is matched with an observation. We will generalize this constraint in Sec. 4.4.

We next assign to each edge from (t, i) to (t', i') a cost as follows:

$$c_{\text{DTW}}((t, i) \to (t', i')) = \begin{cases} -\log \mathcal{L}\left(\theta_{t'} \mid \mathbf{y}_{i'}\right) & t \neq t' \\ 0 & t = t' \end{cases}. \tag{3}$$

Under these edge costs, the DTW registration can be recovered from the shortest path from $(1, 1)$ to (T, n), where a match occurs whenever the time step changes along the path. We note that the cost itself will differ from the dynamic programming approach by the cost of $(1, 1)$, but this constant factor is unimportant for the purposes of minimization. Going forward, we will ignore similar discrepancies by constant factors without note.

4.2 Temporal Registration Using the Viterbi Algorithm

As we alluded to previously, this variant of DTW does not accurately reflect the underlying HMM because the result may not include some of the observations. However, there is a similar algorithm which does find the true maximum-likelihood registration of a sequence of observations to an HMM, namely the

Viterbi algorithm [24], which has previously been used in task learning and recognition [9, 16]. This algorithm can be viewed as a specific instance of DTW with an appropriate choice of costs, but it is more illuminating to describe using the graph we introduced in the prior section.

To do so, we first need to reparameterize the registration, so that instead of mapping time steps to observations via ι_t, we map observations to time steps via τ_i. Then we need only change the edge costs as follows:

$$\begin{aligned} c((t,i) \to (t',i')) &= -\log \mathcal{L}\left(\tau_{i'} = t' \mid \tau_i = t, Y, \Theta\right) \\ &= -\log p_{t,t'} - \log \mathcal{L}\left(\theta_{t'} \mid \mathbf{y}_{i'}\right) \ . \end{aligned} \tag{4}$$

Recall that $p_{t,t'}$ is the probability of transitioning from time step t to t', so this is not only arguably simpler than the DTW edge costs (3), but correctly considers transition probabilities (estimated in Sec. 5). Under these edge costs, the Viterbi registration is again the shortest path from $(1,1)$ to (T,n), but we consider θ_t to be matched with $\mathbf{y}_i$ whenever (t,i) occurs in this path. In particular, a time step may be matched with multiple observations, while every observation will be matched with exactly one time step (see Fig. 4). This coincides with the underlying HMM model.

To see that the shortest path indeed corresponds to the maximum-likelihood registration, we again rely on monotonicity and the Markov property as follows:

$$\begin{aligned} \arg\min_{\tau} \sum_{i=1}^{n-1} -\log \mathcal{L}\left(\tau_{i+1} \mid \tau_i, Y, \Theta\right) &= \arg\max_{\tau} \prod_{i=1}^{n-1} \mathcal{L}\left(\tau_{i+1} \mid \tau_i, Y, \Theta\right) \\ &= \arg\max_{\tau} \mathcal{L}\left(\tau \mid Y, \Theta\right) \ . \end{aligned} \tag{5}$$

However, this approach still suffers from the issues associated with maximum likelihood approaches, namely local minima and sensitivity to noise.

One question that arises is what shortest path algorithm to use. Because the edge costs may be negative, Bellman-Ford [3] is a reasonable choice. However, we can relate this better to DTW and the forward-backward algorithm by noting that because the demonstration graph is a directed acyclic graph (DAG), so must be the tensor product graph. Because the graph is acyclic, we need not perform the multiple passes usually required by the Bellmann-Ford algorithm thanks to the availability of a topological ordering. With this simplification, the Bellmann-Ford algorithm may be described by a recurrence:

$$C(v) = \bigoplus_{e}^{u \to v} C(u) \otimes c(e) \tag{6}$$

where $a \oplus_{\mathrm{ML}} b = \min(a,b)$ and $a \otimes_{\mathrm{ML}} b = a + b$ with initial condition on the source vertex $s \in \mathrm{V}$ set to the identity of $\otimes$, that is $\mathrm{R}(s) = 1_{\otimes}$. Expanding this out again yields the general DTW algorithm. The ML subscripts on these operators indicate that this choice produces maximum-likelihood estimates, and in the next section, we will show that simply changing our choice of semiring [11] (associative $\oplus$ and $\otimes$ with the distributive property) effectively yields the forward-backward algorithm.

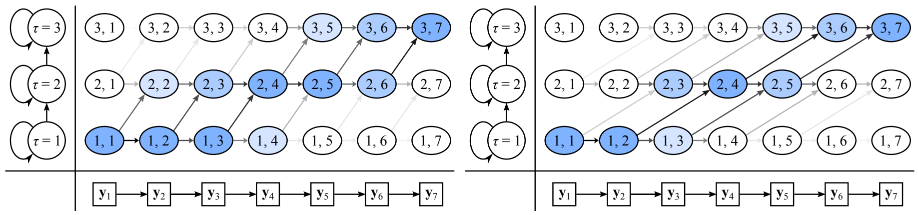


Fig. 5. (left) Tensor product of the time graph (vertical axis) and demonstration graph (horizontal axis). Darker edges are more probable and shaded vertices show the distribution of likely registrations in PTR. **(right)** Modified to exclude degenerate registrations ($K_\Delta = 2$).

4.3 PTR Using the Forward-Backward Algorithm

In contrast to the Viterbi algorithm, the forward-backward algorithm [2] computes not only the most likely temporal registrations, but the posterior probabilities of all possible temporal registrations. Rather than presenting this approach in its original terms, however, we present an equivalent formulation using a graph algorithm that is more amenable to further modifications. The graph is the same as the one used in the previous section and shown in Fig. 4. However, to implement the forward-backward algorithm, we instead use:

$$a \oplus_{\mathrm{FB}} b = -\log(e^{-a} + e^{-b}) \qquad a \otimes_{\mathrm{FB}} b = a + b \tag{7}$$

which are simply the sum and product of the probabilities in negative log-space, producing (unnormalized) distributions over registrations given prior observations. Specifically,

$$\mathcal{L}\left(\tau_i = t \mid \Theta, \mathbf{y}_1, \ldots, \mathbf{y}_{i-1}\right) \propto e^{-\mathrm{C}(t,i)} \ . \tag{8}$$

To extend this to distributions over registrations given the entire sequence of observations, one need only compute the value $\mathrm{C}'(v)$ of each vertex in the reverse graph. The likelihood of an observation $\mathbf{y}_i$ registering to time step t is then:

$$\mathcal{L}\left(\tau_i = t \mid \Theta, Y\right) \propto e^{-(\mathrm{C}(t,i)+\mathrm{C}'(t,i))} \ . \tag{9}$$

This view of these algorithms as recurrences over a tensor product graph enables us to further improve registrations by modifying the graph in Sec. 4.4.

4.4 Non-degenerate Temporal Registration

For many practical use cases, it is desirable to ensure that a sufficient number K_Δ of observations are registered to each time step. We say that such registrations are non-degenerate because if they are subsequently used to estimate covariance matrices as in prior learning methods [4, 7], this guarantee is necessary for the problem to be well-posed. Even with other choices of parameters or estimators, it may be desirable to enforce such a constraint because human demonstrators naturally perform precise parts of a task more slowly. This can be thought of as a stronger version of the Itakura parallelogram constraint used for DTW [12].

This constraint can be enforced by modifying the structure of the graph used in Fig. 5 to ensure that all paths satisfy the K_Δ constraint. An example of this modified graph structure is shown in Fig. 5. The edge costs must then be modified similarly for the Viterbi and forward-backward algorithms:

$$c((t,i) \to (t',i')) = -(i'-i-1) \cdot \log p_{t,t} - \log p_{t,t'} - \sum_{k=i}^{i'-1} \log \mathcal{L}\left(\theta_t \mid \mathbf{y}_k\right). \quad (10)$$

Note that this cost is identical to that given in (4) when $i' = i+1$, that is, when the time step does not change.

The observation weights may then be computed from the vertex values using a similar approach to that used previously for the forward-backward algorithm:

$$\mathcal{L}\left(\tau_i = t \mid \theta, Y\right) = e^{-\mathrm{C}(t,i)-\mathrm{C}'(t,i)} + \sum_{i'=i+1-K_\Delta}^{i-1} e^{-\mathrm{C}(t,i')-c((t,i')\to(t+1,i'+K_\Delta))-\mathrm{C}'(t+1,i'+K_\Delta)} \quad (11)$$

The combination of using the forward-backward algorithm and enforcing the non-degeneracy constraint gives us our complete method for PTR.

5 Application to Robot Learning from Demonstrations

We apply PTR to the estimation of two different task models from prior work on robot learning from demonstrations [4, 7]. These models were selected because they both explicitly require temporal registration. The method of van den Berg et al. [4] learns a task model encoding parameters of a control policy. The method of Bowen et al. [7] learns a task model that is used by a sampling-based motion planner to compute costs for edges in a roadmap such that the shortest path in the roadmap will accomplish the learned task.

Consider a set of user-provided demonstrations $\{Y^{(1)}, \ldots, Y^{(m)}\}$ where each $Y^{(j)}$ is a sequence $\{\mathbf{y}_1^{(j)}, \ldots, \mathbf{y}_{n_j}^{(j)}\}$, where we use a parenthesized superscript to indicate per-demonstration parameters. We reproduce the core learning algorithm of van den Berg et al. [4] in Algorithm 1 with small notational changes for parity and to highlight the underlying HMM model. Next consider the moderately abridged implementation of the method of Bowen et al. [7] shown in Algorithm 2. We focus here on learning the task models; details on implementing the controllers and motion planners using these models are in [4, 7].

Although each of these algorithms operates in a different space, we note that both fit the general mold of expectation-maximization (EM) methods, interleaving estimation of model parameters θ and latent parameters (τ or ι and $R^{(j)}$). More accurately, however, these are maximization-maximization methods, because their respective registration steps compute the best (in some sense) single registration given θ rather than a distribution over all possible registrations.

To apply PTR to the methods of van den Berg et al. and Bowen et al., we need only replace DTW and extend the model parameter expectation steps

Algorithm 1 EstimateVanDenBerg2010(T, $Y^{(1)}, \ldots, Y^{(m)}$)

Initialize $R^{(j)} = \mathcal{I}$ and $\iota_t^{(j)} = \left\lceil \frac{\left|Y^{(j)}\right| \cdot t}{T} \right\rceil$.
while not converged **do**
  for $t = 1$ **to** T **do**
    $Z_t = \left\{ (\mathbf{y}_i^{(j)}, R^{(j)}) \mid i = \iota_t^{(j)} \right\}$
  $\hat{\Theta} \leftarrow$ KalmanSmoother($Z_1, \ldots, Z_n$)
  for $j = 1$ **to** m **do**
    $R^{(j)} \leftarrow \arg\max_R \mathcal{L}\left(R \mid Y^{(j)}, \hat{\Theta}\right)$
    $\hat{\iota}^{(j)} \leftarrow \arg\max_\iota \mathcal{L}\left(\iota \mid Y^{(j)}, \hat{\Theta}\right)$ ▷ *Dynamic time-warping*

Algorithm 2 Estimate_Bowen_2015(T, $Y^{(1)}, \ldots, Y^{(m)}$)

Initialize $\hat{\tau}_i^{(j)} = \left\lceil \frac{T \cdot i}{\left|Y^{(j)}\right|} \right\rceil$.
while not converged **do**
  for $t = 1$ **to** T **do**
    $Z_t = \left\{ \mathbf{y}_i^{(j)} \mid \hat{\tau}_i^{(j)} = t \right\}$
    $\hat{\theta}_t \leftarrow$ MaximumLikelihoodEstimator(Z_t)
  for $j = 1$ **to** m **do**
    $\hat{\tau}^{(j)} \leftarrow \arg\max_\tau \mathcal{L}\left(\tau \mid Y^{(j)}, \hat{\Theta}\right)$ ▷ *Viterbi algorithm*

of these methods to handle weighted observations to accommodate probability-weighted temporal registrations. More specifically, when estimating θ_t, we use all $\mathbf{y}_i^{(j)}$, but weight each by its posterior likelihood of being registered to time step t, $\omega_{t,i}^{(j)} = \mathcal{L}\left(\tau_i^{(j)} = t \mid Y^{(j)}\right)$, which is given by (8). This produces a true expectation-maximization method, and in the method of Bowen et al., yields the Baum-Welch algorithm, which has the general effect of smoothing the estimation compared to the maximum likelihood approach, reducing (but not eliminating) local minima and improving robustness (see results in Sec. 6).

The transition probabilities in the HMM (Fig. 3(b)) may be estimated by applying Bayes' rule, yielding:

$$p_{t,t'} = \mathcal{L}\left(\tau_{i+1} = t' \mid \tau_i = t, Y\right) = \frac{\sum_{i,j} \omega_{t,i}^{(j)} \omega_{t',i+1}^{(j)}}{\sum_{i,j} \omega_{t,i}^{(j)}}. \tag{12}$$

In the method of van den Berg et al., at each time step t the Kalman update step can be performed once for each observation $\mathbf{y}_i^{(j)}$, with weight $\omega_{t,i}^{(j)}$ applied by scaling the covariance matrix $R^{(j)}$ of the observation by $1/\omega_{t,i}^{(j)}$. Equivalently, and more efficiently, a single Kalman update step can be done at each time step t by combining the weighted observations as follows:

$$\hat{\Sigma}_t = \left(\sum_{i,j} \omega_{t,i}^{(j)} {R^{(j)}}^{-1} \right)^{-1} \qquad \hat{\mu}_t = \hat{\Sigma}_t \left(\sum_{i,j} \omega_{t,i}^{(j)} {R^{(j)}}^{-1} \mathbf{y}_i^{(j)} \right)$$

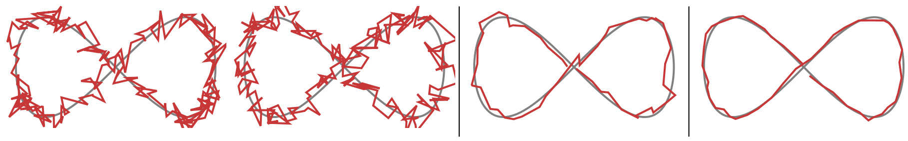


Fig. 6. Learning a drawing task with reference trajectory shown in gray. **(left)** Example demonstrations of the simulated drawing task with medium Gaussian observation noise. Trajectories learned from these two demonstrations using the DTW-1 **(center)** and PTR-3 **(right)** temporal registration algorithms.

6 Results

We evaluate the impact of PTR by applying it to two previously-developed robot learning methods that require temporal registration of demonstrations. In particular, we apply our temporal registration to the robot learning methods of van den Berg et al. [4] and Bowen et al. [7]. We use *Name*-K_Δ to indicate the various temporal registration methods (e.g., DTW-1 or PTR-12). Note that when $K_\Delta = 1$, as in DTW-1 or PTR-1, these are equivalent to their unconstrained variants, DTW and the forward-backward algorithm respectively. Computation was performed on an Intel Xeon E5-1680 CPU with 8 cores running at 3.40 GHz.

6.1 PTR Applied to the van den Berg et al. Method

We first apply PTR to the method of van den Berg et al. [4], which learns a task model encoding parameters of a control policy and enables speedups on task performance. We consider a drawing task and a knot tying task. In both tasks, the number of time steps T was the length of the shortest demonstration divided by K_Δ, which is the maximum number for which a temporal registration exists and matches the original paper for $K_\Delta = 1$.

Simulated Drawing Task For the simulated drawing task, instead of using human demonstrations, we perturbed a canonical figure eight using one of two noise models. The goal then, was to recover this canonical motion, allowing us to empirically evaluate different temporal registration approaches both in terms of learning time and error. The first noise model is that assumed by the underlying model, where observations within a demonstration are corrupted by *i.i.d.* Gaussian noise as seen in Fig. 6. Results for low, medium, and high noise amplitudes with various numbers of demonstrations are shown in Fig. 7.

We performed the same experiments with Brownian motion noise, effectively adding Gaussian velocity noise, as shown in Fig. 8. While this does not match the assumptions of the underlying model, it produces demonstrations much more similar to what a human might. Results are shown in Fig. 9.

Under both noise models, our method of PTR exhibited lower error relative to DTW, particularly with high noise and when only a small number of demonstrations are available. The improvements were most dramatic when the noise model matched the underlying method assumptions because noise which violates the independence assumptions introduces bias that temporal registration alone cannot correct. Both approaches required comparable learning time.

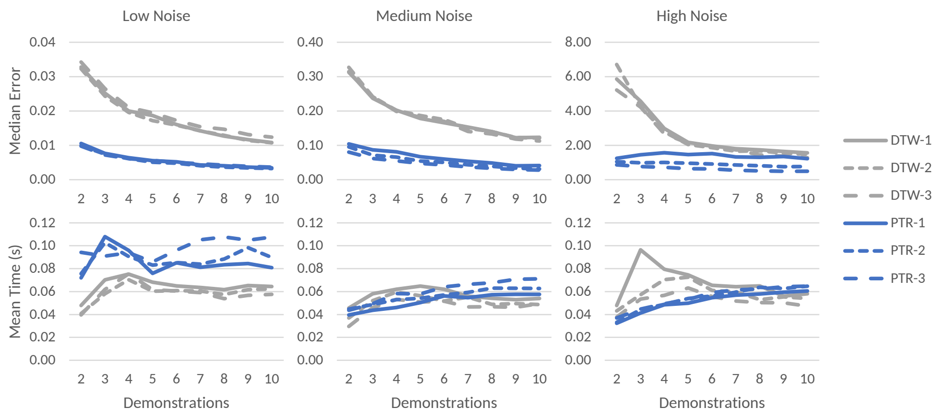


Fig. 7. Results for the simulated drawing task with Gaussian observation noise. PTR exhibited lower error relative to DTW, particularly when only a small number of demonstrations are available.

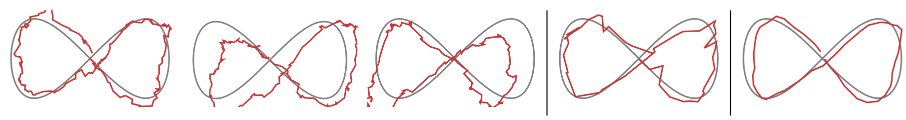


Fig. 8. (left) Example demonstrations of the simulated drawing task with high Brownian motion noise. Trajectories learned from these three demonstrations using the DTW-1 **(center)** and PTR-3 **(right)** temporal registration algorithms.

Physical Knot Tying Task Our physical knot tying task was similar to that described by van den Berg et al. However, it was demonstrated and executed on the Baxter robot, which has more restrictive dynamics limitations than the Berkeley Surgical Robot on which the original experiments were performed. As in the original paper, we divided the task into three phases: an initial loop, a grasp, and an extraction (see Fig. 1). Unlike in the original paper, we learned models for all three phases rather than only the first and third. We performed five demonstrations at 20 Hz of the first two phases and three of the third. These demonstrations ranged from 16 to 30 seconds in length. For an execution to be considered successful, we required the robot to tie a slip knot without exceeding its kinematic or dynamics limitation, including avoiding self-collisions. To evaluate methods in the context of van den Berg et al.'s superhuman performance, we executed the algorithms at progressively greater speeds until the method failed and recorded the maximum multiple of the average demonstration speed at which the task was still performed successfully. Results separated by task phase are shown in Table 1.

A minimum of three samples was used for the non-degenerate variant, which corresponds to a 0.15 second window. In the first phase of the task, DTW-1 failed first due to incorrect placement of the rope, while PTR-1 and DTW-3 encountered self-collisions. Only PTR-3 reached the velocity limits of the robot. In the second and easiest phase, the cause of failure for every temporal registration

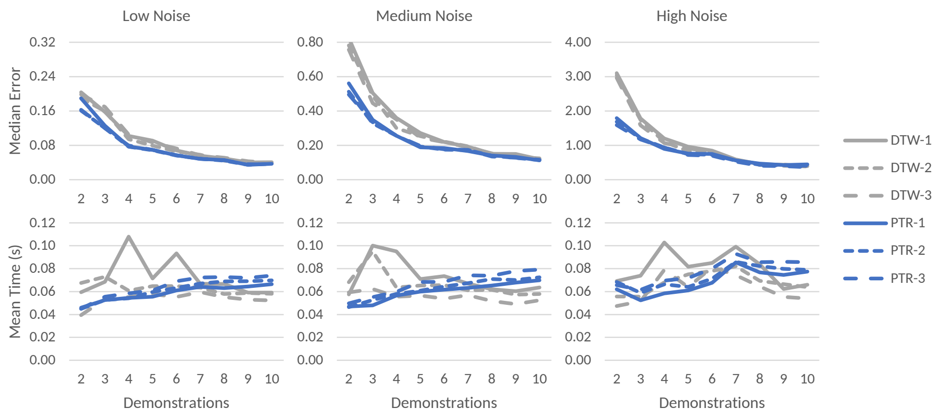


Fig. 9. Results for the simulated drawing task with Brownian motion noise. PTR exhibited lower error relative to DTW, particularly when only a small number of demonstrations are available.

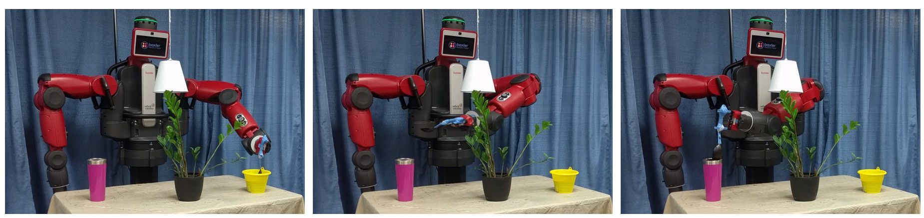


Fig. 10. Using the method of Bowen et al. [7] we learned a task wherein the Baxter robot scooped powder from the yellow container and transferred it into the magenta one without spilling while avoiding obstacles in the environment.

method was reaching the velocity limits of the robot. However, some methods resulted in smoother registrations and subsequent motions, permitting overall faster execution. In the third and most difficult phase, DTW-1 exceeded the velocity limits of the robot while the other three methods failed to extract the arm through the newly-formed loop in the rope. In the case of PTR-1, even at 1x demonstration speed, this was the cause of failure because the registration failed to isolate a crucial part of the task. PTR-3 achieved the highest speedup for all phases of the knot-tying task, showing the benefits of probability-weighted temporal registration with the non-degenerate registration feature.

6.2 PTR Applied to the Bowen et al. Method

We next apply PTR to the method of Bowen et al. [7], which learns a task model that is used by a sampling-based motion planner to compute costs for edges in a roadmap such that the shortest path in the roadmap will accomplish the learned task. For task model learning, [7] used the Viterbi algorithm for temporal registration, so it is this approach we compare against. Similarly, we set $T = 24$ to match the original paper.

Phase	Registration	Max Motion Speedup	Learning Time (s)
1	DTW-1	2.5	0.40
	PTR-1	2.3	0.64
	DTW-3	2.4	0.33
	PTR-3	**3.3**	0.52
2	DTW-1	2.2	0.05
	PTR-1	3.0	0.35
	DTW-3	2.7	0.04
	PTR-3	**3.1**	0.19
3	DTW-1	1.1	0.85
	PTR-1	-	0.57
	DTW-3	2.0	0.35
	PTR-3	**3.2**	1.20

Table 1. Results for the knot-tying task using the method of van den Berg et al. with different temporal registration approaches. Motion speedup indicates the maximum multiple of the average demonstration speed at which the task was still performed successfully.

Demonstrations	Registration	Success	Learning Time (s)
10	Viterbi-1	80%	2.86
	PTR-1	90%	78.64
	PTR-12	**100%**	87.31
5	Viterbi-1	60%	3.16
	PTR-1	80%	36.37
	PTR-12	**90%**	36.12

Table 2. Results for the powder transfer task using the method of Bowen et al. [7] with different demonstration counts and temporal registration approaches. PTR-12 had the highest success rate, regardless of the number of demonstrations.

We performed a powder transfer task on the Baxter robot shown in Fig. 10, as specified in [7]. In this task, the robot is to scoop powder onto a spoon from a source container (the yellow bucket) and transfer it to a destination container (the magenta thermos) while avoiding obstacles (e.g., the plant on the table and the white hanging lamp shade). The robot learned the task using the method and features of Bowen et al. [7] from 10 kinesthetic demonstrations (in which no obstacles were present). Task models were learned using three different temporal registration approaches. To evaluate each model, we introduced the obstacles and randomly sampled scenarios with container positions such that the transfer would always cross the centerline of the table to ensure each scenario was challenging. An execution was considered successful if it transferred powder from one container to the other without spilling. Results are shown in Table 2.

As with the van den Berg method, the use of PTR yielded a measurably better task model in terms of success rate relative to the Viterbi algorithm for temporal registration. Learning times for PTR were substantially longer, but still very reasonable for an off-line process. The primary cause of the failures that occurred during task execution was slightly missing the cup during the dumping motion, resulting in spilled powder. We observe that the number of demonstrations had a marginally greater impact when using the Viterbi algorithm for registration than when using PTR. As with the method of van den Berg et al., PTR-12 performed best.

7 Conclusion

Many existing methods for robot learning from demonstrations require registering a time sequence of observations to a reference model, either for aligning demonstrations during preprocessing or as an integral part of task model estimation. We introduced probability-weighted temporal registration (PTR), a more general form of temporal registration that explicitly captures uncertainty in the registration. Instead of assuming each observation is registered to (at most) one time step of the reference model like DTW, we use the forward-backward algorithm to compute probability-weighted assignments and avoid degenerate registrations. We applied PTR to two learning methods from prior work on both simulated and physical tasks and showed that incorporating PTR into robot learning algorithms can yield higher-quality task models that enable faster task executions and higher task success rates.

In future work, we would like to apply PTR to other robotics algorithms that require temporal registration and to automatically determine the best non-degeneracy parameter.

Acknowledgments. We thank Armaan Sethi for his assistance evaluating methods. This research was supported in part by the U.S. National Science Foundation (NSF) under Awards IIS-1149965 and CCF-1533844.

References

1. Amor, H.B., Neumann, G., Kamthe, S., Kroemer, O., Peters, J.: Interaction primitives for human-robot cooperation tasks. In: IEEE Int Conf. Robotics and Automation (ICRA), pp. 2831–2837 (2014)
2. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Annals of Mathematical Statistics **41**(1), 164–171 (1970)
3. Bellman, R.: On a routing problem. Tech. rep., DTIC Document (1956)
4. van den Berg, J., Miller, S., Duckworth, D., Hu, H., Wan, A., Goldberg, K., Abbeel, P.: Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA), pp. 2074–2081 (2010)
5. Berger, E., Sastuba, M., Vogt, D., Jung, B., Ben Amor, H.: Estimation of perturbations in robotic behavior using dynamic mode decomposition. Advanced Robotics **29**(5), 331–343 (2015)
6. Bodiroža, S., Doisy, G., Hafner, V.V.: Position-invariant, real-time gesture recognition based on dynamic time warping. In: ACM/IEEE Int. Conf. Human-Robot Interaction (HRI), pp. 87–88 (2013)
7. Bowen, C., Ye, G., Alterovitz, R.: Asymptotically-optimal motion planning for learned tasks using time-dependent cost maps. IEEE Trans. Automation Science and Engineering **12**(1), 171–182 (2015)
8. Calinon, S.: A tutorial on task-parameterized movement learning and retrieval. Intelligent Service Robotics **9**(1), 1–29 (2016)

9. Calinon, S., Guenter, F., Billard, A.: On learning, representing, and generalizing a task in a humanoid robot. IEEE Trans. Systems, Man and Cybernetics–Part B **37**(2), 286–298 (2007)
10. Davis, R.I., Lovell, B.C.: Comparing and evaluating HMM ensemble training algorithms using train and test and condition number criteria. Formal Pattern Analysis & Applications **6**(4), 327–335 (2004)
11. Herstein, I.N.: Topics in algebra. John Wiley & Sons (2006)
12. Itakura, F.: Minimum prediction residual principle applied to speech recognition. IEEE Trans. Acoustics, Speech, and Signal Processing **23**(1), 67–72 (1975)
13. Krishnan, S., Garg, A., Liaw, R., Thananjeyan, B., Miller, L., Pokorny, F.T., Goldberg, K.: SWIRL: A sequential windowed inverse reinforcement learning algorithm for robot tasks with delayed rewards. In: WAFR (2016)
14. Lee, A.X., Lu, H., Gupta, A., Levine, S., Abbeel, P.: Learning force-based manipulation of deformable objects from multiple demonstrations. In: IEEE Int. Conf. Robotics and Automation (ICRA), pp. 177–184 (2015)
15. Lember, J., Koloydenko, A.A.: Bridging Viterbi and posterior decoding: a generalized risk approach to hidden path inference based on hidden Markov models. Journal of Machine Learning Research **15**(1), 1–58 (2014)
16. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and Viterbi path searching. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
17. Maeda, G., Ewerton, M., Lioutikov, R., Amor, H.B., Peters, J., Neumann, G.: Learning interaction for collaborative tasks with probabilistic movement primitives. In: IEEE-RAS Int. Conf. Humanoid Robots, pp. 527–534 (2014)
18. Marteau, P.: Times series averaging and denoising from a probabilistic perspective on time-elastic kernels (2016). URL http://arxiv.org/abs/1611.09194
19. Müller, M.: Dynamic time warping. Information Retrieval for Music and Motion pp. 69–84 (2007)
20. Pavlovic, V., Rehg, J.M., MacCormick, J.: Learning switching linear models of human motion. In: Advances in neural information processing systems, pp. 981–987 (2001)
21. Tanwani, A.K., Calinon, S.: Learning robot manipulation tasks with task-parameterized semitied hidden semi-Markov model. IEEE Robotics and Automation Letters **1**(1), 235–242 (2016)
22. Titsias, M.K., Holmes, C.C., Yau, C.: Statistical inference in hidden Markov models using k-segment constraints. Journal of the American Statistical Association **111**(513), 200–215 (2016)
23. Vakanski, A., Mantegh, I., Irish, A., Janabi-Sharifi, F.: Trajectory learning for robot programming by demonstration using hidden Markov model and dynamic time warping. IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics **42**(4), 1039–1052 (2012)
24. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Information Theory **13**(2), 260–269 (1967)
25. Vuković, N., Mitić, M., Miljković, Z.: Trajectory learning and reproduction for differential drive mobile robots based on GMM/HMM and dynamic time warping using learning from demonstration framework. Engineering Applications of Artificial Intelligence **45**, 388–404 (2015)
26. Weichsel, P.M.: The Kronecker product of graphs. Proc. American Mathematical Society **13**(1), 47–52 (1962)
27. Yu, S.Z., Kobayashi, H.: An efficient forward-backward algorithm for an explicit-duration hidden Markov model. IEEE signal processing letters **10**(1), 11–14 (2003)