Robotics and Autonomous Systems 60 (2012) 789–802

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

A nonparametric Bayesian approach toward robot learning by demonstration

Sotirios P. Chatzis *, Dimitrios Korkinof, Yiannis Demiris

Department of Electrical and Electronic Engineering, Imperial College London, United Kingdom

ARTICLE INFO

Article history:
Received 10 May 2011
Received in revised form
6 February 2012
Accepted 13 February 2012
Available online 21 February 2012

Keywords:
Gaussian mixture regression
Robot learning by demonstration
Dirichlet process
Variational Bayes
Nonparametric statistics

ABSTRACT

In the past years, many authors have considered application of machine learning methodologies to effect robot learning by demonstration. Gaussian mixture regression (GMR) is one of the most successful methodologies used for this purpose. A major limitation of GMR models concerns automatic selection of the proper number of model states, i.e., the number of model component densities. Existing methods, including likelihood- or entropy-based criteria, usually tend to yield noisy model size estimates while imposing heavy computational requirements. Recently, Dirichlet process (infinite) mixture models have emerged in the cornerstone of nonparametric Bayesian statistics as promising candidates for clustering applications where the number of clusters is unknown a priori. Under this motivation, to resolve the aforementioned issues of GMR-based methods for robot learning by demonstration, in this paper we introduce a nonparametric Bayesian formulation for the GMR model, the Dirichlet process GMR model. We derive an efficient variational Bayesian inference algorithm for the proposed model, and we experimentally investigate its efficacy as a robot learning by demonstration methodology, considering a number of demanding robot learning by demonstration scenarios.

1. Introduction

In the last years, robot learning by demonstration has turned out to be one of the most active research topics in the field of robotics. Robot learning by demonstration encompasses methods by which a robot can learn new skills by simple observation of a human teacher, similar to the way humans learn new skills by imitation [1–8]. Coming up with successful robot learning by demonstration methodologies can be of great benefit to the robotics community, since it will greatly obviate the need of programming a robot how to perform a task, which can be rather tedious and expensive, while, by making robots more user-friendly, it increases the appeal of applying robots to real-life environments.

Toward this end, robotics researchers have utilized a multitude of methodologies from as diverse research areas as machine learning, computer vision [9], and human–robot interaction [10]. Learning by demonstration algorithms may comprise learning an approximation to the state-action mapping (mapping function), or learning a model of the world dynamics and deriving a policy from this information (system model). Mapping function learning comprises classification-based and regression-based approaches. Classification approaches categorize their input into discrete classes, thus the input to the classifier is the robot state, and the discrete output classes are robot actions. Gaussian Mixture Models (GMMs), decision trees, Bayesian networks, and hidden Markov models are typical methods used to effect the classification task. Regression approaches map demonstration states to continuous action spaces resulting from combining multiple demonstration set actions. As such, typically regression approaches apply to low-level trajectory-based learning by demonstration, and not to high-level behaviors. Finally, the system model approach uses a state transition model of the world, and from this derives a policy, typically by means of reinforcement learning (RL). As such, it usually has the drawback of high computational demands, due to the considerably large dimensionality of the entailed search space of the RL algorithm.

Recently, several researchers have also considered developing libraries of dynamic movement primitives (DMPs) as a way to facilitate generalization of the learned models to new unseen situations [11,12]. DMPs are sets of differential equations that represent the task's dynamics. Generalization using DMPs is effected by parameterizing them with new appropriate start and goal positions to generalize to novel situations, with the advantage of good robustness to perturbation. This is typically performed by application of regression methods based on local weighting of training data at execution time.

In this work, we focus on trajectory-based learning by demonstration techniques. The most popular trends of work in this field consist in the investigation of the utility of probabilistic generative models, such as Gaussian mixture regression (GMR) and derivatives, [13] hidden Markov models [14], and Gaussian process regression [2]. GMR, in particular, has been shown to be very

* Corresponding author.
E-mail address: s.chatzis@imperial.ac.uk (S.P. Chatzis).

0921-8890/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2012.02.005

successful in encoding demonstrations, extracting their underlying constraints, and reproducing smooth generalized motor trajectories, while imposing considerably low computational costs [15,1]. GMR-based approaches toward learning by demonstration rely on the postulation of a Gaussian mixture model to encode the covariance relations between different variables (either in the task space, or in the robot joints space). If the correlations vary significantly between regions, then each local region of the state space visited during the demonstrations will need a few Gaussians to encode these local dynamics. Given the required number of Gaussians and a set of training data (human-generated demonstrations), the expectation-maximization (EM) algorithm is eventually employed to estimate the parameters of the model.

The most common data-driven methodologies for GMR model selection, that is determination of the appropriate number of GMR model component densities, are typically based on the popular Bayesian information criterion (BIC) for finite mixture models [16], or other related likelihood-based or entropy-based model size selection criteria [17]. However, such model selection methods suffer from significant drawbacks: To begin with, they entail training of multiple models (to select from), a tedious procedure which can be applied only up to a limited extent, due to its computational demands. Moreover, effectiveness of the BIC criterion is contingent on a number of conditions, which are not necessarily fulfilled in real-life application scenarios [17]; thus, BIC-based approximations are rather prone to yielding noisy model size estimates. Most significantly, likelihood- and entropy-based model selection criteria are notorious for their heavy overfitting proneness, hence often leading to over-estimation of the required model size [18].

Dirichlet process mixture (DPM) models are flexible Bayesian nonparametric models which have become very popular in statistics over the last few years, for performing nonparametric density estimation [19–21]. Briefly, a realization of a DPM can be seen as an infinite mixture of distributions with given parametric shape (e.g., Gaussian). This theory is based on the observation that an infinite number of component distributions in an ordinary finite mixture model tends on the limit to a Dirichlet process prior [20,22]. Indeed, although theoretically a DPM model has an infinite number of parameters, it turns out that inference for the model is possible, since only the parameters of a finite number of mixture components need to be represented explicitly; this can be done by means of an elegant and computationally efficient truncated variational Bayesian approximation [23]. Eventually, as a part of the model fitting procedure, the nonparametric Bayesian inference scheme induced by a DPM model yields a posterior distribution on the proper number of model component densities [24], rather than selecting a fixed number of mixture components. Hence, the obtained nonparametric Bayesian formulation eliminates the need of doing inference (or making arbitrary choices) on the number of mixture components necessary to represent the modeled data.

Under this motivation, in this work we introduce a nonparametric Bayesian approach toward Gaussian mixture regression, with application to robot learning by demonstration. Our approach is based on the consideration of a GMR model with a countably infinite number of constituent states, and is effected by utilization of a Dirichlet process (DP) prior distribution; we shall be referring to this new model as the Dirichlet process Gaussian mixture regression (DPGMR) model. Inference for the DPGMR model is conducted using an elegant variational Bayesian algorithm, and is facilitated by means of a stick-breaking construction of the DP prior, which allows for the derivation of a computationally tractable expression of the model variational posteriors. Our novel mixture regression methodology is subsequently applied to yield a nonparametric Bayesian approach toward robot learning by demonstration, the efficacy of which is illustrated by considering a number of demanding robot learning by demonstration scenarios.

The remainder of this paper is organized as follows: In Section 2, Gaussian mixture regression as applied to robot learning by demonstration is introduced in a concise manner. In Section 3, we provide a brief review of concepts from the field of Dirichlet process mixture models, emerging in the cornerstone of nonparametric Bayesian statistics. In Section 4, we derive the proposed nonparametric Bayesian approach toward robot learning by demonstration. In Section 5, the experimental evaluation of the proposed algorithm is performed. The final section concludes this paper.

2. Gaussian mixture regression for robot learning by demonstration

Let us consider the current position of the moving end-effector of a robot as the predictor variable $\boldsymbol{\beta}$ of our machine learning algorithm, and the velocity that must be adopted by the robot's end-effector at the next time-step, in order to comply with the learnt trajectory, as the algorithm's response variable $\dot{\boldsymbol{\beta}}$. GMR postulates a model of the conditional expectation of the set of response variables $\dot{\boldsymbol{\beta}}$ given the set of predictor variables $\boldsymbol{\beta}$, by exploiting the information available in a set of training observations $\{\boldsymbol{\beta}_j, \dot{\boldsymbol{\beta}}_j\}_{j=1}^N$.

A significant advantage of GMR-based methodologies is that, contrary to most traditional regression methodologies, GMR does not directly approximate the regression function but postulates a GMM to model the *joint* probability distribution of the considered response and predictor variables ($\dot{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}$), i.e. it considers a model of the form

$$p(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}|\boldsymbol{\pi}, \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K) = \sum_{i=1}^K \pi_i \mathcal{N}(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{1}$$

where $\boldsymbol{\pi} = (\pi_i)_{i=1}^K$ are the prior weights of the mixture component densities, and $\mathcal{N}(\cdot|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is a Gaussian with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. As a result, contrary to most discriminative regression algorithms (e.g., SVMs [25], and Gaussian processes [26]), the computational time required for trajectory reproduction does not increase with the number of demonstrations provided to the robot, which is a particularly important property for lifelong learning robots. Indeed, the available model training data provided by the employed human demonstrators is processed in only an off-line fashion, to obtain the estimates of the model parameters. This way, prediction generation under GMR reduces to a simple weighted sum of linear models, which is advantageous because trajectory reproduction becomes fast enough to be used at any appropriate time by the robot.

The GMM (1) postulated under the GMR approach is trained by means of the EM algorithm [17], using a set of training data corresponding to a number of trajectories obtained by human demonstrators. Then, using the obtained GMM $p(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}|\boldsymbol{\pi}, \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K)$, Gaussian mixture regression retrieves a generalized trajectory by estimating at each time step the conditional expectation $\mathbb{E}[\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}; \boldsymbol{\pi}, \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K]$. Expressing the means $\boldsymbol{\mu}_i$ of the component densities of the postulated GMM (1) in the form

$$\boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_i^{\beta} \\ \boldsymbol{\mu}_i^{\dot{\beta}} \end{bmatrix} \tag{2}$$

and introducing the notation

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_i^{\beta} & \boldsymbol{\Sigma}_i^{\beta\dot{\beta}} \\ \boldsymbol{\Sigma}_i^{\dot{\beta}\beta} & \boldsymbol{\Sigma}_i^{\dot{\beta}} \end{bmatrix} \tag{3}$$

for the covariance matrices of the model component densities, we can show that, based on (1) and the assumptions (2)–(3),

the conditional probability $p(\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}; \boldsymbol{\pi}, \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K)$ of the response variables $\dot{\boldsymbol{\beta}}$ given the predictor variables $\boldsymbol{\beta}$ and the postulated GMM yields [27]

$$p(\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}; \boldsymbol{\pi}, \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K) = \mathcal{N}(\dot{\boldsymbol{\beta}}|\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) \tag{4}$$

where

$$\hat{\boldsymbol{\mu}} = \sum_{i=1}^{K} \phi_i(\boldsymbol{\beta})[\boldsymbol{\mu}_i^{\dot{\beta}} + \boldsymbol{\Sigma}_i^{\dot{\beta}\beta}(\boldsymbol{\Sigma}_i^{\beta})^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_i^{\beta})] \tag{5}$$

$$\hat{\boldsymbol{\Sigma}} = \sum_{i=1}^{K} \phi_i^2(\boldsymbol{\beta})[\boldsymbol{\Sigma}_i^{\dot{\beta}} - \boldsymbol{\Sigma}_i^{\dot{\beta}\beta}(\boldsymbol{\Sigma}_i^{\beta})^{-1}\boldsymbol{\Sigma}_i^{\beta\dot{\beta}}] \tag{6}$$

and

$$\phi_i(\boldsymbol{\beta}) = \frac{\pi_i \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}_i^{\beta}, \boldsymbol{\Sigma}_i^{\beta})}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}_k^{\beta}, \boldsymbol{\Sigma}_k^{\beta})}. \tag{7}$$

Based on the result (4), predictions under the GMR approach can be obtained by taking the conditional expectations $\mathbb{E}(\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}; \boldsymbol{\pi}, \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K)$, i.e.

$$\widehat{\dot{\boldsymbol{\beta}}} = \mathbb{E}(\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}; \boldsymbol{\pi}, \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K) = \hat{\boldsymbol{\mu}}. \tag{8}$$

As we observe, a significant merit of GMR consists in the fact that it provides a full predictive distribution, thus a predictive variance

$$\mathbb{V}(\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}; \boldsymbol{\pi}, \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K) = \hat{\boldsymbol{\Sigma}}$$

is available at any position of the end-effector. Therefore, GMR offers a model-estimated measure of predictive uncertainty not only at specific positions but continuously along the generated trajectories.

Data-driven selection of the appropriate number of GMR states (model component densities) is a crucial procedure for successfully applying GMR-based robot learning by demonstration: The number of postulated GMR states determines the compromise between accuracy and smoothness of the obtained response (bias-variance tradeoff). Optimal model size *(order)* selection for finite mixture models is an important but very difficult problem which has not been completely resolved. Usually, penalized likelihood-based or entropy-based criteria are used for this purpose [17], such as the Bayesian information criterion (BIC) of Schwarz [16], and variants [18].

The BIC model selection criterion as applied to a GMR-fitted GMM used for trajectory-based robot learning by demonstration consists in the determination of the number of model component densities which minimizes the metric

$$\mathcal{L} \triangleq -2\sum_{n=1}^{N} \log p(\{\boldsymbol{\beta}_n, \dot{\boldsymbol{\beta}}_n\}_{n=1}^N|\boldsymbol{\pi}, \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K) + d \log N \tag{9}$$

where d is the total number of model parameters, hence a function of the number of mixture component densities K, and N is the number of available model training data points. BIC has been shown to provide consistent model order estimators under certain conditions [28]. However, these conditions are not necessarily fulfilled in real-world application scenarios [17]. Additionally, BIC has been found to fit too few components when the model for the component densities (here, the Gaussian assumption) is valid and the sample size is not very large [29]. Finally, if the model for the component densities is not valid, then it has been found to fit too many components [17]. This is the most common issue that plagues GMR when it comes to its robot learning by demonstration applications, since it is a problem practitioners are quite often confronted with, and it may severely undermine the performance of the GMR-based learning by demonstration algorithm, by giving rise to overfitting issues.

The main aim of this work is to resolve these very issues of GMR-based robot learning by demonstration, by coming up with a method that allows for automatic, data-driven determination of the proper number of model component densities K, without being vulnerable to overfitting.

3. Dirichlet process mixture models

Dirichlet process models were first introduced by Ferguson [30]. A DP is characterized by a base distribution G_0 and a positive scalar α, usually referred to as the innovation parameter, and is denoted as $\mathrm{DP}(G_0, \alpha)$. Essentially, a DP is a distribution placed over a distribution. Let us suppose we randomly draw a sample distribution G from a DP, and, subsequently, we independently draw N random variables $\{\Theta_n^*\}_{n=1}^N$ from G:

$$G|\{G_0, \alpha\} \sim \mathrm{DP}(G_0, \alpha) \tag{10}$$

$$\Theta_n^*|G \sim G, \quad n = 1, \ldots N. \tag{11}$$

Integrating out G, the joint distribution of the variables $\{\Theta_n^*\}_{n=1}^N$ can be shown to exhibit a clustering effect. Specifically, given the first $N-1$ samples of G, $\{\Theta_n^*\}_{n=1}^{N-1}$, it can be shown that a new sample Θ_N^* is either (a) drawn from the base distribution G_0 with probability $\frac{\alpha}{\alpha+N-1}$, or (b) is selected from the existing draws, according to a multinomial allocation, with probabilities proportional to the number of the previous draws with the same allocation [31]. Let $\{\Theta_c\}_{c=1}^K$ be the set of distinct values taken by the variables $\{\Theta_n^*\}_{n=1}^{N-1}$. Denoting as f_c^{N-1} the number of values in $\{\Theta_n^*\}_{n=1}^{N-1}$ that equal to Θ_c, the distribution of Θ_N^* given $\{\Theta_n^*\}_{n=1}^{N-1}$ can be shown to be of the form [31]

$$p(\Theta_N^*|\{\Theta_n^*\}_{n=1}^{N-1}, G_0, \alpha) = \frac{\alpha}{\alpha+N-1}G_0 + \sum_{c=1}^{K} \frac{f_c^{N-1}}{\alpha+N-1}\delta_{\Theta_c} \tag{12}$$

where δ_{Θ_c} denotes the distribution concentrated at a single point Θ_c. These results illustrate two key properties of the DP scheme. First, the innovation parameter α plays a key-role in determining the number of distinct parameter values. A larger α induces a higher tendency of drawing new parameters from the base distribution G_0; indeed, as $\alpha \to \infty$ we get $G \to G_0$. On the contrary, as $\alpha \to 0$ all $\{\Theta_n\}_{n=1}^N$ tend to cluster to a single random variable. Second, the more often a parameter is shared, the more likely it will be shared in the future.

A characterization of the (unconditional) distribution of the random variable G drawn from a Dirichlet process $\mathrm{DP}(G_0, \alpha)$ is provided by the stick-breaking construction of Sethuraman [23]. Consider two infinite collections of independent random variables $\boldsymbol{v} = (v_c)_{c=1}^{\infty}$, $\{\Theta_c\}_{c=1}^{\infty}$, where the v_c are drawn from the Beta distribution $\mathrm{Beta}(1, \alpha)$, and the Θ_c are independently drawn from the base distribution G_0. The stick-breaking representation of G is then given by Sethuraman [23]

$$G = \sum_{c=1}^{\infty} \pi_c(\boldsymbol{v})\delta_{\Theta_c} \tag{13}$$

where

$$\pi_c(\boldsymbol{v}) = v_c \prod_{j=1}^{c-1}(1 - v_j) \quad \in [0, 1] \tag{14}$$

and

$$\sum_{c=1}^{\infty} \pi_c(\boldsymbol{v}) = 1. \tag{15}$$

The stick-breaking representation of the DP makes clear that the random variable G drawn from a DP is discrete. It shows explicitly that the support of G consists of a countably infinite sum of atoms located at Θ_c, drawn independently from G_0. It is also apparent that the innovation parameter α controls the mean value of the stick variables, v_c, as a hyperparameter of their prior distribution; hence, it regulates the effective number of the distinct values of the drawn atoms [23].

Under the stick-breaking representation (13) of the Dirichlet process, the atoms Θ_c, drawn independently from the base distribution G_0, can be seen as the parameters of the component distributions of a mixture model comprising an unbounded number of component densities, with mixing proportions $\pi_c(\boldsymbol{v})$. This way, DP mixture (DPM) models are formulated [22].

Let $\boldsymbol{y} = \{\boldsymbol{y}_n\}_{n=1}^N$ be a set of observations modeled by a DPM model. Then, each one of the observations $\boldsymbol{y}_n$ is assumed to be drawn from its own probability density function $p(\boldsymbol{y}_n|\Theta_n^*)$ parametrized by the parameter set Θ_n^*. All Θ_n^* follow a common DP prior, and given the discreteness of G, may share the same value Θ_c with probability $\pi_c(\boldsymbol{v})$. Introducing the indicator variables $\boldsymbol{x} = (x_n)_{n=1}^N$, with $x_n = c$ denoting that Θ_n^* takes on the value of Θ_c, the modeled dataset $\boldsymbol{y}$ can be described as arising from the process

$$\boldsymbol{y}_n|x_n = c;\ \Theta_c \sim p(\boldsymbol{y}_n|\Theta_c) \tag{16}$$

$$x_n|\boldsymbol{\pi}(\boldsymbol{v}) \sim \text{Mult}(\boldsymbol{\pi}(\boldsymbol{v})) \tag{17}$$

$$v_c|\alpha \sim \text{Beta}(1, \alpha) \tag{18}$$

$$\Theta_c|G_0 \sim G_0 \quad (c = 1, \ldots, \infty) \tag{19}$$

where $\boldsymbol{\pi}(\boldsymbol{v}) = (\pi_c(\boldsymbol{v}))_{c=1}^\infty$ is given by (14), and $\text{Mult}(\boldsymbol{\pi}(\boldsymbol{v}))$ is a Multinomial distribution over $\boldsymbol{\pi}(\boldsymbol{v})$.

4. Proposed approach

Let $\boldsymbol{y} = \{\boldsymbol{y}_n\}_{n=1}^N$, with $\boldsymbol{y}_n = \{\boldsymbol{\beta}_n, \dot{\boldsymbol{\beta}}_n\}$ being the set of predictor variables and response variables the joint distribution of which is represented by means of a postulated GMR model. We want to model this data by means of a nonparametric Bayesian formulation of the GMR model. For this purpose, we postulate a GMR model with a countably infinite number of states. To formulate such a model, we begin by postulating a Gaussian DPM model for the joint distribution of the $\boldsymbol{\beta}$ and $\dot{\boldsymbol{\beta}}$, and we further derive the expressions for the conditional predictive distribution of the response variables $\dot{\boldsymbol{\beta}}$ given the predictor variables $\boldsymbol{\beta}$.

Denoting as $\boldsymbol{x} = (x_n)_{n=1}^N$ the labels of the GMR states emitting the fitting data $\boldsymbol{y}$, we have

$$\boldsymbol{y}_n|x_n = c;\ \Theta_c \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{R}_c) \tag{20}$$

for the state-conditional likelihoods of the model, where $\Theta_c = \{\boldsymbol{\mu}_c, \boldsymbol{R}_c\}$, and $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{R}_c)$ is a Gaussian distribution with mean $\boldsymbol{\mu}_c$ and precision (inverse covariance) $\boldsymbol{R}_c$, while it holds

$$p(\boldsymbol{x}) = \prod_{n=1}^N p(x_n|\boldsymbol{\pi}(\boldsymbol{v})) \tag{21}$$

with the $p(x_n = c|\boldsymbol{\pi}(\boldsymbol{v}))$ being the prior probabilities of the model states, stemming from the imposed Dirichlet process, given by (14) and (17).

Definition. We denote as the Dirichlet process Gaussian mixture regression (DPGMR) model a Gaussian mixture regression model with a countably infinite number of states, based on the introduction of a Dirichlet process as the prior of its state emission probabilities.

4.1. Inference for the DPGMR model

Inference for DPM-type models can be conducted under a Bayesian setting, typically by means of variational Bayes (e.g., [32]), or Monte Carlo techniques (e.g., [33]). Here, we prefer a variational Bayesian approach, due to its considerably better scalability in terms of computational costs. Bayesian inference involves introduction of a set of appropriate priors over the model parameters, and derivation of the corresponding (approximate) posterior densities. We choose conjugate-exponential priors, as this selection greatly simplifies inference and interpretability [27]. Hence, we impose a joint Normal–Wishart distribution over the means and precisions of the Gaussian likelihoods of the model states

$$p(\Theta_c) = \mathcal{NW}(\boldsymbol{\mu}_c, \boldsymbol{R}_c|\lambda_c, \boldsymbol{m}_c, \omega, \boldsymbol{\Psi}_c). \tag{22}$$

We mention that in (22) we have assumed a common value for the hyperparameters ω of the model component densities, i.e., $\omega_c = \omega_{c'} = \omega,\ \forall c \neq c'$. We make this hyperparameter tying assumption so as to simplify the resulting expression of the model predictive density, derived in Section 4.2. Additionally, taking under consideration the effect of the innovation hyperparameter α on the number of effective component densities (states) of a DPM-type model, we choose to also impose a (hyper-)prior over the innovation hyperparameter α of the DPGMR model. We use a Gamma prior with

$$p(\alpha) = \mathcal{G}(\alpha|\gamma_1, \gamma_2). \tag{23}$$

Our variational Bayesian inference formalism for the DPGMR model consists in derivation of a family of variational posterior distributions $q(.)$ which approximate the true posterior distribution over the infinite sets $\boldsymbol{v} = (v_c)_{c=1}^\infty$ and $\{\boldsymbol{\mu}_c, \boldsymbol{R}_c\}_{c=1}^\infty$, and the innovation parameter α. Apparently, under this infinite dimensional setting, Bayesian inference is not tractable. For this reason, we employ a common strategy in DPM literature, formulated on the basis of a truncated stick-breaking representation of the DP [32]. That is, we fix a value K and we let the variational posterior over the v_i have the property $q(v_K = 1) = 1$. In other words, we set $\pi_c(\boldsymbol{v})$ equal to zero for $c > K$. Note that, under this setting, the treated DPGMR model involves a full DP prior; truncation is not imposed on the model itself, but only on the variational distribution to allow for a tractable inference procedure. Hence, the truncation level K is a variational parameter which can be freely set, and not part of the prior model specification.

Let $W = \{\boldsymbol{v}, \alpha, \boldsymbol{x}, \boldsymbol{\mu}_c, \boldsymbol{R}_c\}_{c=1}^K$ be the set of hidden variables and unknown parameters of the DPGMR model over which a prior distribution has been imposed, and Ξ be the set of the hyperparameters of the imposed priors, $\Xi = \{\lambda_c, \boldsymbol{m}_c, \omega, \boldsymbol{\Psi}_c, \gamma_1, \gamma_2\}_{c=1}^K$. Variational Bayesian inference consists in the introduction of an arbitrary distribution $q(W)$ to approximate the actual posterior $p(W|\Xi, \boldsymbol{y})$, which is computationally intractable [27]. Under this assumption, the log marginal likelihood (log evidence), $\log p(\boldsymbol{y})$, of the model yields [34]

$$\log p(\boldsymbol{y}) = \mathcal{L}(q) + \text{KL}(q\|p) \tag{24}$$

where

$$\mathcal{L}(q) = \int \text{d}W q(W) \log \frac{p(\boldsymbol{y}, W|\Xi)}{q(W)} \tag{25}$$

and $\text{KL}(q\|p)$ stands for the Kullback–Leibler (KL) divergence between the (approximate) variational posterior, $q(W)$, and the actual posterior, $p(W|\Xi, \boldsymbol{y})$. Since KL divergence is nonnegative, $\mathcal{L}(q)$ forms a strict lower bound of the log evidence, and would become exact if $q(W) = p(W|\Xi, \boldsymbol{y})$. Hence, by maximizing this lower bound $\mathcal{L}(q)$ (variational free energy) so that it becomes as tight as possible, not only do we minimize the KL-divergence between the true and the variational posterior, but we also implicitly integrate out the unknowns W.

Due to the considered conjugate prior configuration of the DPGMR model, the variational posterior $q(W)$ is expected to take

the same functional form as the prior, $p(W)$ [18]; thus, it is expected to factorize as

$$q(W) = q(\boldsymbol{x})q(\alpha)\left(\prod_{c=1}^{K-1} q(v_c)\right)\prod_{c=1}^{K} q(\boldsymbol{\mu}_c, \boldsymbol{R}_c) \tag{26}$$

with

$$q(\boldsymbol{x}) = \prod_{n=1}^{N} q(x_n). \tag{27}$$

Then, the variational free energy of the model reads (ignoring constant terms)

$$\begin{aligned}\mathcal{L}(q) = &\sum_{c=1}^{K}\int \mathrm{d}\boldsymbol{R}_c \int \mathrm{d}\boldsymbol{\mu}_c \left[q(\boldsymbol{\mu}_c, \boldsymbol{R}_c)\right.\\ &\left.\times \log\frac{p(\boldsymbol{\mu}_c, \boldsymbol{R}_c|\lambda_c, \boldsymbol{m}_c, \omega, \boldsymbol{\Psi}_c)}{q(\boldsymbol{\mu}_c, \boldsymbol{R}_c)}\right]\\ &+ \int \mathrm{d}\alpha q(\alpha)\left\{\log\frac{p(\alpha|\gamma_1, \gamma_2)}{q(\alpha)}\right.\\ &\left.+ \sum_{c=1}^{K-1}\int \mathrm{d}v_c q(v_c)\log\frac{p(v_c|\alpha)}{q(v_c)}\right\}\\ &+ \sum_{c=1}^{K}\sum_{n=1}^{N} q(x_n = c)\left\{\int \mathrm{d}\boldsymbol{v} q(\boldsymbol{v})\log\frac{p(x_n = c|\boldsymbol{\pi}(\boldsymbol{v}))}{q(x_n = c)}\right.\\ &\left.+ \int \mathrm{d}\boldsymbol{R}_c \int \mathrm{d}\boldsymbol{\mu}_c q(\boldsymbol{\mu}_c, \boldsymbol{R}_c)\log p(\boldsymbol{y}_n|\Theta_c)\right\}.\end{aligned} \tag{28}$$

The analytical expression of the variational free energy $\mathcal{L}(q)$ can be found in the Appendix.

Derivation of the variational posterior distribution $q(W)$ involves maximization of the variational free energy $\mathcal{L}(q)$ over each one of the factors of $q(W)$ in turn, holding the others fixed, in an iterative manner [35]. By construction, this iterative, consecutive updating of the variational posterior distribution is guaranteed to monotonically and maximally increase the free energy $\mathcal{L}(q)$, which functions as the convergence criterion of the derived inference algorithm for the DPGMR model [18].

Let us denote as $\langle . \rangle$ the posterior expectation of a quantity. We begin with the posterior distributions over the DP parameters. From (28), we have

$$q(v_c) = \mathrm{Beta}(\eta_{c,1}, \eta_{c,2}) \tag{29}$$

where

$$\eta_{c,1} = 1 + \sum_{n=1}^{N} q(x_n = c) \tag{30}$$

$$\eta_{c,2} = \langle\alpha\rangle + \sum_{c'=c+1}^{K}\sum_{n=1}^{N} q(x_n = c') \tag{31}$$

and

$$q(\alpha) = \mathcal{G}(\alpha|\hat{\gamma}_1, \hat{\gamma}_2) \tag{32}$$

where

$$\hat{\gamma}_1 = \gamma_1 + K - 1 \tag{33}$$

$$\hat{\gamma}_2 = \gamma_2 - \sum_{c=1}^{K-1}[\psi(\eta_{c,2}) - \psi(\eta_{c,1} + \eta_{c,2})] \tag{34}$$

and $\psi(.)$ denotes the Digamma function.

Similar, regarding the posteriors over the likelihood parameters, we have

$$q(\Theta_c) = q(\boldsymbol{\mu}_c, \boldsymbol{R}_c) = \mathcal{NW}(\boldsymbol{\mu}_c, \boldsymbol{R}_c|\tilde{\lambda}_c, \tilde{\boldsymbol{m}}_c, \tilde{\omega}, \tilde{\boldsymbol{\Psi}}_c) \tag{35}$$

where we introduce the notation

$$\tilde{\gamma}_c \triangleq \sum_{n=1}^{N} q(x_n = c) \tag{36}$$

$$\bar{\boldsymbol{y}}_c \triangleq \frac{\sum_{n=1}^{N} q(x_n = c)\boldsymbol{y}_n}{\tilde{\gamma}_c} \tag{37}$$

$$\boldsymbol{\Delta}_c \triangleq \sum_{n=1}^{N} q(x_n = c)\,(\boldsymbol{y}_n - \bar{\boldsymbol{y}}_c)\,(\boldsymbol{y}_n - \bar{\boldsymbol{y}}_c)^T \tag{38}$$

and, we have

$$\tilde{\omega} = \omega + 1 + \frac{1}{K}\sum_{c=1}^{K}\tilde{\gamma}_c \tag{39}$$

$$\tilde{\boldsymbol{\Psi}}_c = \boldsymbol{\Psi}_c + \boldsymbol{\Delta}_c + \frac{\lambda_c\tilde{\gamma}_c}{\lambda_c + \tilde{\gamma}_c}(\boldsymbol{m}_c - \bar{\boldsymbol{y}}_c)(\boldsymbol{m}_c - \bar{\boldsymbol{y}}_c)^T \tag{40}$$

$$\tilde{\lambda}_c = \lambda_c + \tilde{\gamma}_c \tag{41}$$

$$\tilde{\boldsymbol{m}}_c = \frac{\lambda_c\boldsymbol{m}_c + \tilde{\gamma}_c\bar{\boldsymbol{y}}_c}{\tilde{\lambda}_c}. \tag{42}$$

Finally, the posteriors over the model states generating the data yield

$$q(x_n = c) \propto \tilde{\pi}_c(\boldsymbol{v})\tilde{p}(\boldsymbol{y}_n|\Theta_c) \tag{43}$$

where

$$\begin{aligned}\tilde{\pi}_c(\boldsymbol{v}) &\triangleq \exp(\langle\log\pi_c(\boldsymbol{v})\rangle)\\ &= \exp\left[\sum_{c'=1}^{c-1}\langle\log(1 - v_{c'})\rangle + \langle\log v_c\rangle\right]\end{aligned} \tag{44}$$

and

$$\begin{aligned}\tilde{p}(\boldsymbol{y}_n|\Theta_c) &\triangleq \exp(\langle\log p(\boldsymbol{y}_n|\Theta_c)\rangle)\\ &= \exp\left[-\frac{d}{2}\log 2\pi + \frac{1}{2}\langle\log|\boldsymbol{R}_c|\rangle\right.\\ &\quad\left.- 2\langle(\boldsymbol{y}_n - \boldsymbol{\mu}_c)^T\boldsymbol{R}_c(\boldsymbol{y}_n - \boldsymbol{\mu}_c)\rangle\right].\end{aligned} \tag{45}$$

The expressions of the posterior expected values included in the update Eqs. (29)–(45) can be found in the Appendix.

4.2. Predictive density

Having obtained the (variational) Bayesian estimators of the DPGMR model parameters, we can now proceed to the derivation of the model predictive density, that is the conditional density $p(\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}; \boldsymbol{y})$, where $\boldsymbol{y}$ is the training set used for model estimation.

Let us first consider the expression of the joint predictive density $p(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}|\boldsymbol{y})$ of our model. Based on the formulation of the Gaussian DPM employed by our model, we have

$$\begin{aligned}p(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}|\boldsymbol{y}) = &\int \mathrm{d}\boldsymbol{v} q(\boldsymbol{v})\sum_{c=1}^{K} p(x = c|\boldsymbol{\pi}(\boldsymbol{v}))\int \mathrm{d}\boldsymbol{\mu}_c\\ &\times \int \mathrm{d}\boldsymbol{R}_c q(\boldsymbol{\mu}_c, \boldsymbol{R}_c)p(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}|x = c; \boldsymbol{\mu}_c, \boldsymbol{R}_c)\end{aligned} \tag{46}$$

which yields a Student's-t predictive density of the form [27]

$$p(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}|\boldsymbol{y}) = \sum_{c=1}^{K} \langle \pi_c(\boldsymbol{v}) \rangle \mathrm{St}\left(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}|\tilde{\boldsymbol{m}}_c, \boldsymbol{S}_c, \tilde{\omega} + 1 - d\right) \tag{47}$$

where d is the total dimensionality of the modeled input space $\{\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}\}$, $\tilde{\boldsymbol{m}}_c$ is given by (42), and the covariance matrix of the predictive density $\boldsymbol{S}_c$ yields

$$\boldsymbol{S}_c = \frac{1 + \tilde{\lambda}_c}{(\tilde{\omega} + 1 - d)\tilde{\lambda}_c} \tilde{\boldsymbol{\Psi}}_c \tag{48}$$

while the expression of the posterior expectations $\langle \pi_c(\boldsymbol{v}) \rangle$ can be found in the Appendix.

Having obtained the expression of the predictive density $p(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}|\boldsymbol{y})$, shown to be of a Student's-t form, we can now proceed to the derivation of the conditional predictive density $p(\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}; \boldsymbol{y})$ of the response variables $\dot{\boldsymbol{\beta}}$ given the predictor variables $\boldsymbol{\beta}$. Let us consider the state-conditional expression of the predictive distribution (47). Setting

$$\tilde{\boldsymbol{m}}_c = \begin{bmatrix} \tilde{\boldsymbol{m}}_c^{\boldsymbol{\beta}} \\ \tilde{\boldsymbol{m}}_c^{\dot{\boldsymbol{\beta}}} \end{bmatrix} \tag{49}$$

and

$$\boldsymbol{S}_c = \begin{bmatrix} \boldsymbol{S}_c^{\boldsymbol{\beta}} & \boldsymbol{S}_c^{\boldsymbol{\beta}\dot{\boldsymbol{\beta}}} \\ \boldsymbol{S}_c^{\dot{\boldsymbol{\beta}}\boldsymbol{\beta}} & \boldsymbol{S}_c^{\dot{\boldsymbol{\beta}}} \end{bmatrix} \tag{50}$$

we can write

$$\begin{bmatrix} \boldsymbol{\beta} \\ \dot{\boldsymbol{\beta}} \end{bmatrix} \Bigg| x = c; \boldsymbol{y} \sim \mathrm{St}\left(\begin{bmatrix} \tilde{\boldsymbol{m}}_c^{\boldsymbol{\beta}} \\ \tilde{\boldsymbol{m}}_c^{\dot{\boldsymbol{\beta}}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{S}_c^{\boldsymbol{\beta}} & \boldsymbol{S}_c^{\boldsymbol{\beta}\dot{\boldsymbol{\beta}}} \\ \boldsymbol{S}_c^{\dot{\boldsymbol{\beta}}\boldsymbol{\beta}} & \boldsymbol{S}_c^{\dot{\boldsymbol{\beta}}} \end{bmatrix}, \tilde{\omega} + 1 - d \right) \tag{51}$$

where

$$p(x = c) = \langle \pi_c(\boldsymbol{v}) \rangle . \tag{52}$$

As discussed, e.g., in [18], the Student's-t distribution can be equivalently written as an infinite sum of Gaussians with the same means and scaled covariances, where the covariance scalars are Gamma-distributed latent variables:

$$\mathrm{St}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \int_0^{\infty} \mathcal{N}\left(\boldsymbol{x} \middle| \boldsymbol{\mu}, \frac{1}{u}\boldsymbol{\Sigma}\right) \mathcal{G}\left(u \middle| \frac{\nu}{2}, \frac{\nu}{2}\right) \mathrm{d}u. \tag{53}$$

Based on this result, (51) can be equivalently expressed as

$$\begin{bmatrix} \boldsymbol{\beta} \\ \dot{\boldsymbol{\beta}} \end{bmatrix} \Bigg| x = c, u; \boldsymbol{y} \sim \mathcal{N}\left(\begin{bmatrix} \tilde{\boldsymbol{m}}_c^{\boldsymbol{\beta}} \\ \tilde{\boldsymbol{m}}_c^{\dot{\boldsymbol{\beta}}} \end{bmatrix}, \frac{1}{u} \begin{bmatrix} \boldsymbol{S}_c^{\boldsymbol{\beta}} & \boldsymbol{S}_c^{\boldsymbol{\beta}\dot{\boldsymbol{\beta}}} \\ \boldsymbol{S}_c^{\dot{\boldsymbol{\beta}}\boldsymbol{\beta}} & \boldsymbol{S}_c^{\dot{\boldsymbol{\beta}}} \end{bmatrix} \right) \tag{54}$$

where

$$u \sim \mathcal{G}\left(\frac{\tilde{\omega} + 1 - d}{2}, \frac{\tilde{\omega} + 1 - d}{2} \right). \tag{55}$$

Using (54), the conditional probability of the response variable $\dot{\boldsymbol{\beta}}$ given the value of the predictor variable $\boldsymbol{\beta}$ reads

$$\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}, u, x = c; \boldsymbol{y} \sim \mathcal{N}\left(\hat{\boldsymbol{\mu}}_c, \frac{1}{u}\hat{\boldsymbol{\Sigma}}_c\right) \tag{56}$$

whence

$$\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}, u; \boldsymbol{y} \sim \mathcal{N}\left(\sum_{c=1}^{K} \langle \pi_c(\boldsymbol{v}) \rangle \hat{\boldsymbol{\mu}}_c, \frac{1}{u} \sum_{c=1}^{K} \langle \pi_c(\boldsymbol{v}) \rangle^2 \hat{\boldsymbol{\Sigma}}_c \right) \tag{57}$$

where

$$\hat{\boldsymbol{\mu}}_c = \tilde{\boldsymbol{m}}_c^{\dot{\boldsymbol{\beta}}} + \boldsymbol{S}_c^{\dot{\boldsymbol{\beta}}\boldsymbol{\beta}} (\boldsymbol{S}_c^{\boldsymbol{\beta}})^{-1} (\boldsymbol{\beta} - \tilde{\boldsymbol{m}}_c^{\boldsymbol{\beta}}) \tag{58}$$

$$\hat{\boldsymbol{\Sigma}}_c = \boldsymbol{S}_c^{\dot{\boldsymbol{\beta}}} - \boldsymbol{S}_c^{\dot{\boldsymbol{\beta}}\boldsymbol{\beta}} (\boldsymbol{S}_c^{\boldsymbol{\beta}})^{-1} \boldsymbol{S}_c^{\boldsymbol{\beta}\dot{\boldsymbol{\beta}}}. \tag{59}$$

Eventually, based on the result (57) and the property (53) of the Student's-t distribution, the conditional predictive distribution of our model turns out to yield

$$\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}; \boldsymbol{y} \sim \mathrm{St}\left(\sum_{c=1}^{K} \langle \pi_c(\boldsymbol{v}) \rangle \hat{\boldsymbol{\mu}}_c, \sum_{c=1}^{K} \langle \pi_c(\boldsymbol{v}) \rangle^2 \hat{\boldsymbol{\Sigma}}_c, \tilde{\omega} + 1 - d \right). \tag{60}$$

Predictions using the DPGMR model can be conducted using the conditional predictive mean of our model as the estimate of the response variables $\dot{\boldsymbol{\beta}}$ at any prediction time point, i.e.

$$\hat{\dot{\boldsymbol{\beta}}} = \mathbb{E}[\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}; \boldsymbol{y}] = \sum_{c=1}^{K} \langle \pi_c(\boldsymbol{v}) \rangle \hat{\boldsymbol{\mu}}_c. \tag{61}$$

The associated conditional predictive variance of the model offers a measure of uncertainty regarding the generated predictions. It yields

$$\mathbb{V}[\dot{\boldsymbol{\beta}}|\boldsymbol{\beta}; \boldsymbol{y}] = \frac{\tilde{\omega} + 1 - d}{\tilde{\omega} - 1 - d} \sum_{c=1}^{K} \langle \pi_c(\boldsymbol{v}) \rangle^2 \hat{\boldsymbol{\Sigma}}_c. \tag{62}$$

5. Experimental evaluation

In this section, we present our experimental evaluation of the DPGMR algorithm in a series of applications dealing with robot learning by demonstration. More specifically, we compare algorithm performance against well established, state-of-the-art methods in the field of robotics, namely Gaussian mixture regression (GMR) [1], and Gaussian process regression (GPR) [36,37]. We have considered three application scenarios with potential practical applicability under an one- and a multi-shot learning setting. In all our experiments, we have utilized joint angle data, which present a great challenge for learning algorithms (compared, e.g., to end-effector data). Our source codes have been developed in Matlab R2011b, and were run on a PC with an Intel Core i7 3.4 GHz CPU, and 16 GB RAM, running Ubuntu Linux 11.04.

For the purposes of our experimental evaluation, we have employed the NAO robot (academic edition), a humanoid robotic platform with 27 degrees of freedom [38]. The training trajectories were presented to the robot by means of kinesthetics, that is manually moving the robot's arms and recording the joint angles. During this procedure, joint position sampling was conducted, with the sampling rate set to 20 Hz. The robot joints actively participating in each experiment varied according to the specification of the performed motion types (for details cf. Table 1 and Fig. 1). The aforementioned joint angle data were collected using a fully threaded NAO-Matlab communication protocol developed by the authors in Python.

5.1. Experimental setup

In our experiments, the predictor variable $\boldsymbol{\beta}$ used by the considered models was the position vector of the robot joints, whereas the response variable $\dot{\boldsymbol{\beta}}$ was the velocity vector that should be imposed on the robot joints so as to remain on the learnt trajectory.

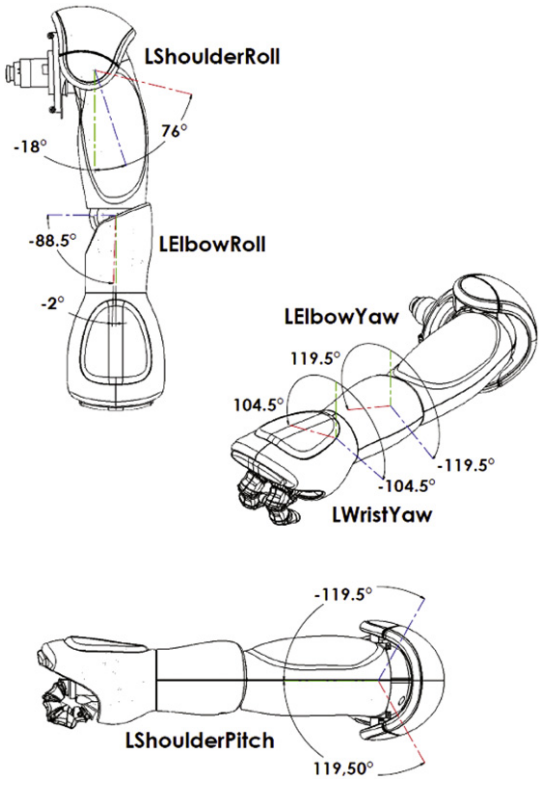
Regarding the multi-shot learning experiment, we used multiple demonstrations of each task, so as to capture the variability of the human action, and evaluate our model's ability to *generalize learned trajectories*. Training was conducted using three out of a total of four available sequences, and the generalization capabilities of the compared methods were evaluated using the fourth data sequence. Due to the temporal variations observed in the demonstrations, we pre-processed the sequences using Dynamic Time Warping (DTW) [39], a method first used in speech recognition

Table 1
NAO robot joints participating in each experiment and corresponding ranges of movement.

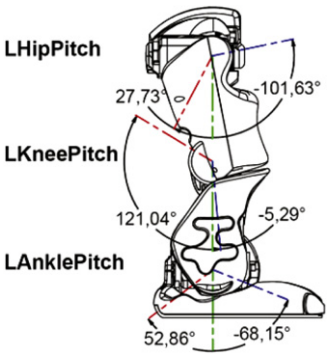
Joints/tasks	*Lazy eight figures*	Ph. education	Blocking	Range (rads)
LShoulderPitch	√			[−2.0857, 2.0857]
LShoulderRoll	√		√	[−0.3142, 1.3265]
LElbowYaw	√		√	[−2.0857, 2.0857]
LElbowRoll	√		√	[1.5446, 0.0349]
RShoulderPitch			√	[−2.0857, 2.0857]
RShoulderRoll			√	[−0.3142, 1.3265]
RElbowYaw			√	[−2.0857, 2.0857]
RElbowRoll			√	[1.5446, 0.0349]
LHipPitch		√		[−1.773912, 0.484090]
LAnklePitch		√		[−1.189516, 0.922747]
RHipPitch		√		[−1.773912, 0.484090]
RAnklePitch		√		[−1.189516, 0.922747]

Table 2
Number of data points and dimensionalities of the used datasets.

Task	One-shot learning dataset		Multi-shot learning dataset		Units
	#Data points	#Dimensions	#Data points	#Dimensions	
Blocking	218	8	1086	8	rad
Ph. education	213	5	592	5	rad
Lazy eight figures	239	5	717	5	rad



(a) Left arm joint specifications.



(b) Left leg joint specifications.

Fig. 1. NAO robot joint specifications and corresponding ranges of movement, presented in degrees.
Source: Aldebaran robotics [38].

for signal alignment. Subsequently, we used a low-pass filter to smooth out anomalies resulting from the alignment. In Table 2, we present some details concerning the number of points and the dimensionality of each dataset.

In the case of the one-shot learning scenario, the aim was to evaluate our approach under a sparser setting. For this purpose, we used only the testing sequences of each task, after subjecting them to undersampling, so as to obtain a total sequence length of approximately 200 samples in each case. Testing was conducted by adding uniformly distributed noise $\mathcal{U}(0, 1)$ to the initial points of the used sequences (also used for model training), and running the algorithms so as to regenerate the (rest of the) learnt trajectories. Taking into consideration the maximum joint ranges presented in Table 1, it becomes obvious that the induced noise levels give rise to a substantial deviation from the initial trajectory starting points.

To measure the performance of the evaluated algorithms, we utilize the mean square error (MSE) along the entire sequence length as our error metric. We have excluded the time component from MSE calculation, due to its trivial form; in fact, the time

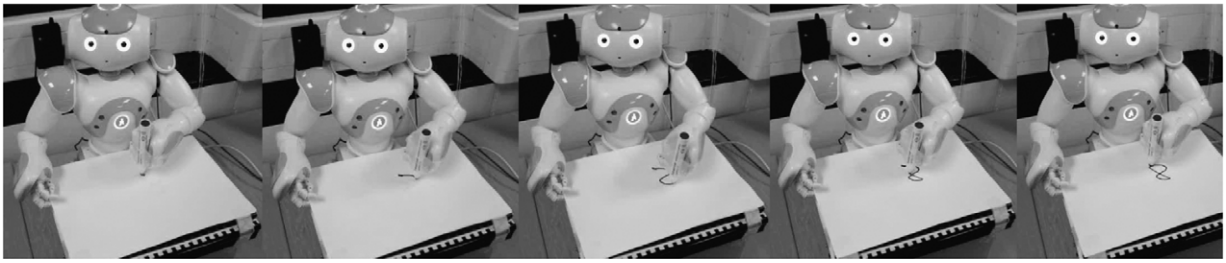


Fig. 2. NAO robot during the lazy eight figures experiment.

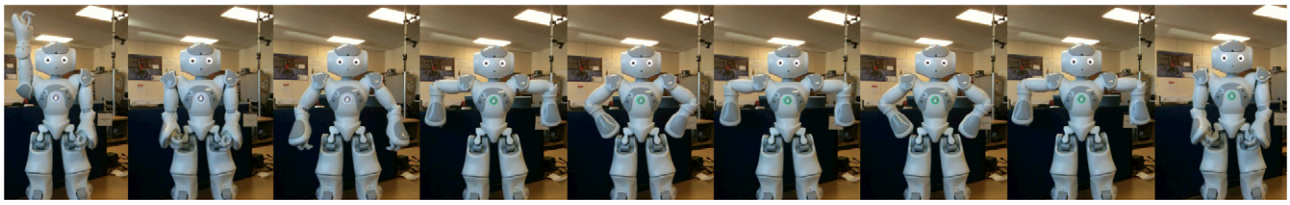


Fig. 3. Communicative gesture for the violation "Blocking".

variable is merely a line dichotomizing the 1*st* quadrant (ϵ : $y = x$). We also provide graphical illustrations of the generated trajectories, by projecting them onto a 2-dimensional space, so as to allow for a qualitative assessment of the obtained prediction results.

Regarding model size selection, we have repeated our experiments for various numbers of model states K to examine how model performance is affected by this selection. We experimented with values of K greater than 25, and low enough to ensure that the number of estimated model parameters (i.e., the components of the $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, and the π_i) does not exceed the number of training data points. This way, we ward off the possibility of overfitting for the GMR model, as suggested in [17]. Note that application of such precautionary measures to avoid overfitting is not necessary for the DPGMR model. As already discussed, the DPGMR model, being a nonparametric Bayesian model, essentially imposes a prior on the number of underlying model states. Hence, the number of states K in the case of the DPGMR model corresponds to a variational truncation level that expresses the maximum allowed number of model states. From these states, only a small subset will eventually manifest itself with any significance level ϵ, after model training. This subset of the DPGMR model states shall henceforth be referred to as the model *active components*.

In an attempt to account for the effect of poor model initialization, which may lead model training under both the EM algorithm and the variational Bayesian approach to yield bad local optima as model estimators, our experiments using the GMR and DPGMR models were executed multiple times for each considered number of (maximum) model states K, with different random initializations each time. Means and standard deviations of the employed performance measures over the executed multiple runs are computed for each K value, and the statistical significance of these results is assessed by means of the Student-t statistical hypothesis test.

We briefly describe the conducted experiments below.

1. ***Lazy eight figures***: In this experiment, we evaluate the considered methods in terms of their applicability in teaching a robot by demonstration how to draw a complex figure. The considered figure comprises a *lazy eight figures* (Fig. 2). The *lazy eight figures* (L8) generation task is a classical benchmark for pattern generation methodologies [40,41]. From the first impression, the task appears to be trivial, since an 8 figure can be interpreted as the superposition of a sine on the horizontal direction, and a cosine of half the sine's frequency on the vertical direction. A closer inspection though will reveal that in reality this seemingly innocent task entails surprisingly challenging stability problems, which come to the fore especially when using very limited model training datasets. The used dataset consists of joint angle data from drawing 3 consecutive *L8s*.
2. **Upper body motion:** In the case of upper body motion, our experiments involve a higher number of joints, thus further increasing the dimensionality and, consequently, the complexity of the addressed problem. We examine learning and reproduction of a communicative gesture used by Basketball officials, with potential applicability in the case of a robotic referee. We have chosen a gesture that poses a challenge on the learning by demonstration algorithm in terms of the implied motion complexity, namely the sign concerning the violation "blocking"[1] (Fig. 3).
3. **Lower body motion:** Finally, we examine an experimental case involving movement of the lower robot body, simulating a lower abdominal muscle exercise (Fig. 4). This is one of the scenarios under investigation of the ALIZ-E EU FP7 project (http://www.aliz-e.org/), where robots are used as companions to diabetic and obese children in pediatric ward settings over extended time periods, and learn along with the children various sensorimotor activities (e.g. dance, games, and physical exercises) so that they can practice and improve together.

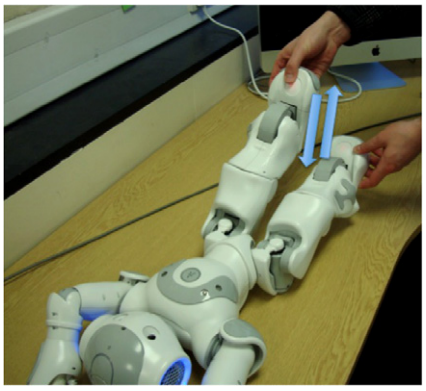
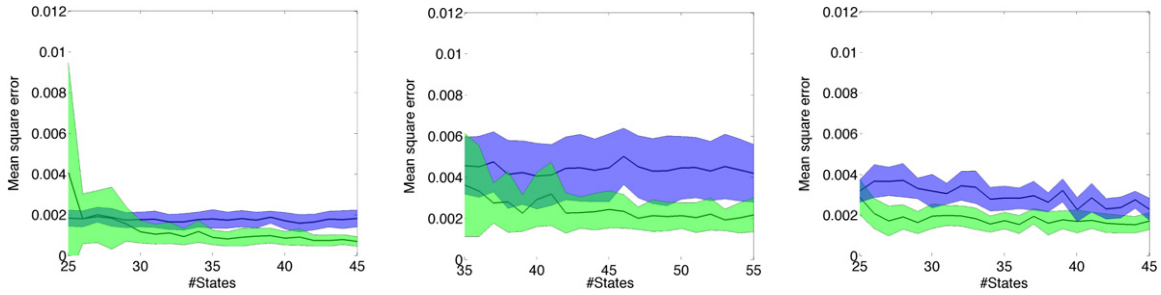


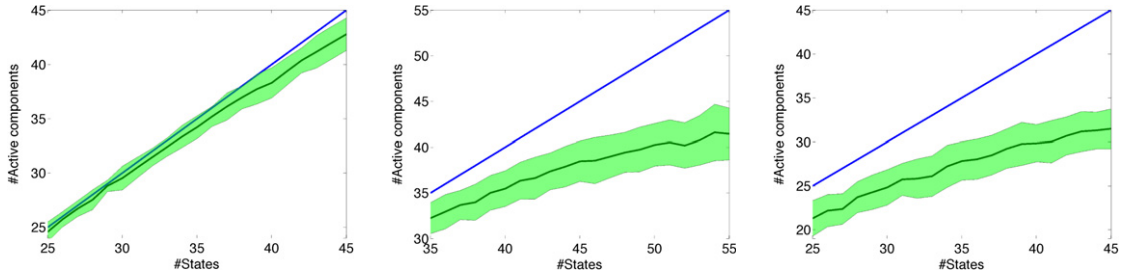
Fig. 4. Physical education exercise for the lower abdominal muscles.

[1] Also referred to as "traveling".



(a) *Lazy eight figures* experiment. (b) Ph. E. exercise experiment. (c) Blocking gesture experiment.

Fig. 5. One-shot LbD scenario, MSE plots. GREEN: DPGMR, BLUE: GMR. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) *Lazy eight figures* experiment. (b) Ph. E. exercise experiment. (c) Blocking gesture experiment.

Fig. 6. One-shot LbD scenario: number of DPGMR model active components plots. GREEN: #active components, BLUE: #initial states. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
One-shot learning experiments: MSE results obtained by GPR, and best mean MSE results for the GMR and DPGMR methods.

Task	One-shot learning MSE (time excluded)		
	GMR	GPR	DPGMR
Lazy eight figures	$16 \cdot 10^{-4}$ ($\pm 4.24 \cdot 10^{-4}$)	0.062818	$6.9 \cdot 10^{-4}$($\pm 2.4 \cdot 10^{-4}$)
Ph. education	$41 \cdot 10^{-4}$ ($\pm 16 \cdot 10^{-4}$)	0.436098	$19 \cdot 10^{-4}$($\pm 5.21 \cdot 10^{-4}$)
Blocking	$23 \cdot 10^{-4}$ ($\pm 5.94 \cdot 10^{-4}$)	0.395844	$15 \cdot 10^{-4}$ ($\pm 4.08 \cdot 10^{-4}$)

Table 4
Statistical significance results from the Student-t test. Obtained p-values below 10^{-2} indicate high statistical significance.

Task	One-shot		Multi-shot	
	Null hypothesis	p-value	Null hypothesis	p-value
Lazy eight figures	rejected	0.0028	rejected	$2.03 \cdot 10^{-9}$
Ph. education	rejected	$2.78 \cdot 10^{-13}$	rejected	$7.10 \cdot 10^{-8}$
Blocking	rejected	$5.21 \cdot 10^{-11}$	rejected	$1.85 \cdot 10^{-10}$

5.2. One-shot learning

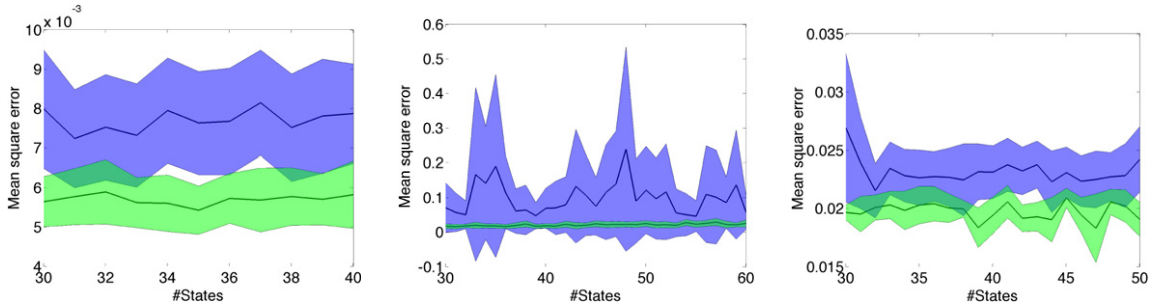
As far as the one-shot learning experiment is concerned, training and testing of the GMR and DPGMR models were repeated 100 times for each K value, using different random initializations at each iteration. The means and standard deviations of the so-obtained MSE values of the GMR and DPGMR methods are presented in Fig. 5 for all experiments. In Table 3, we present the best mean MSEs obtained by the GMR and DPGMR methods, and the results for GPR. Regarding the performance of GPR, we notice that the method proves to be unsuitable and fails to learn the presented trajectories. As a result, the errors induced are much higher compared to the other two evaluated methods.

Regarding comparison between GMR and DPGMR, we observe that the proposed method contributes to a significantly improved performance in all the conducted experiments. The error results also are more consistent, as a much lower standard deviation of the MSE values is achieved in almost all cases. Elaborating on that, we can see that the best DPGMR mean error is less than half of that obtained by GMR in the case of the L8s experiment ($\simeq$43%), and the Ph. E. exercise experiment ($\simeq$46%). In the blocking communicative gesture experiment, the improvement is approximately 34%. The standard deviation of the observed error values is also consistently lower in most cases. The L8s experiment was the only exception to that rule; apparently, this phenomenon occurred as a consequence of the fact that at lower numbers of initial states (K) the truncation level of the DPGMR was obviously much less than needed to adequately model the observed trajectories, thus leading to poor model fits. Additionally, utilizing the Student-t test we are able to evaluate the statistical significance of our findings, regarding the mean MSEs of GMR and DPGMR for different numbers of states. As can be seen in Table 4, the null hypothesis that both sequences of mean MSEs belong to distributions with the same mean is rejected with considerable certainty.

Finally, in Fig. 6 we depict the obtained mean number of DPGMR active components, as well as their corresponding standard deviation. A very important conclusion that can be drawn from these plots is that introduction of a Dirichlet process prior resulted in significantly smaller models, thus avoiding both the unnecessary complexity and the excessive computational burden of GMR, without the need to resort to unreliable likelihood- or entropy-based model selection criteria. Indeed, the DPGMR

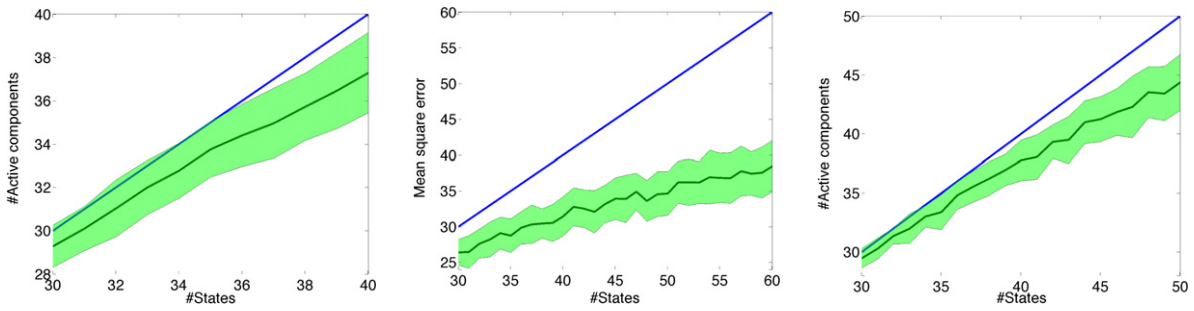
Table 5
Multi-shot learning experiments: MSE results obtained by GPR, and best mean MSE results for the GMR and DPGMR methods.

Task	Multi-shot learning MSE (time excluded)		
	GMR	GPR	DPGMR
Lazy eight figures	0.0072 ($\pm 12 \cdot 10^{-4}$)	0.059959	0.0054 ($\pm 6 \cdot 10^{-4}$)
Ph. education	0.0493 ($\pm$0.0475)	0.305892	0.0160 ($\pm$0.0072)
Blocking	0.0215 ($\pm$0.0023)	0.381941	0.0183 ($\pm$0.0029)



(a) *Lazy eight figures* experiment. (b) Ph. E. exercise experiment. (c) Blocking gesture experiment.

Fig. 7. Multi-shot LbD scenario, MSE plots. GREEN: DPGMR, BLUE: GMR. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) *Lazy eight figures*. (b) Ph. E. exercise. (c) Blocking gesture.

Fig. 8. Multi-shot LbD scenario: number of DPGMR model active components plots. GREEN: #active components, BLUE: #initial states. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

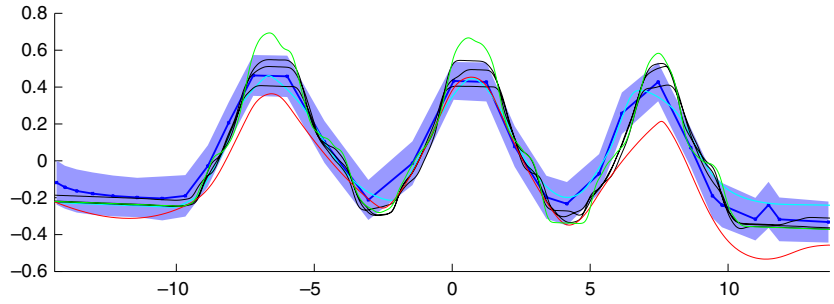
model is able to achieve, on average, up to 30% model complexity reduction compared to GMR. Note that the inferred number of active components for the DPGMR model varies depending on the random initialization of the training algorithm, hence the obtained variance of the number of active components provided in these graphs. We must underline though, that this variability is not an undesirable property: In the case of the DPGMR model, the number of active components is regarded as yet another model parameter, and its value is determined in conjunction with the values of the rest of the model parameters, so as to optimize the variational lower bound $\mathcal{L}(q)$ in (25). Thus, different starting points for the model training algorithm will necessarily yield different local optimal for $\mathcal{L}(q)$, with the number of model active components being one of the optimized parameters.

5.3. Multi-shot learning

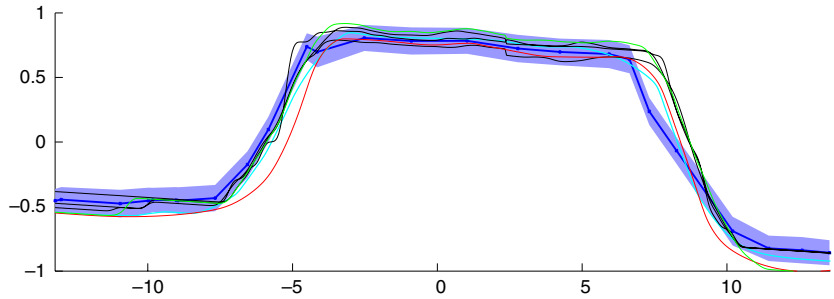
For the multi-shot learning experiment, we have calculated the mean MSE and its standard deviation resulting from 50 repetitions of the training and testing procedures for the GMR and DPGMR methods, as well as the performance of GPR. The obtained MSE results are presented in Fig. 7 for the GMR and DPGMR methods, and the best error results for the GMR and DPGMR methods along with the performance of GPR are provided in Table 5. The results of the Student-t test regarding the comparison between GMR and DPGMR can be seen in Table 4, and, finally, the number of active components of the DPGMR method are depicted in Fig. 8.

Commenting on the results, DPGMR achieves an error reduction of approximately 27.7% in the L8s experiment, 66.3% in the Ph. E. experiment, and 14.9% in the Blocking communicative gesture experiment, compared to GMR. The standard deviation of the DPGMR results is also lower, indicating that the postulated model is more consistent. Even in the blocking experiment, where Table 5 shows a higher STD for DPGMR, it can be seen in Fig. 7(c) that this behavior consists an isolated case, and the yielded STD is in general lower than the one obtained by GMR. The Student-t tests show an even higher statistical significance of the difference between GMR and DPGMR, compared to the previous experiment. Similarly, GPR does not perform adequately well and fails to predict the presented trajectories, regardless of the considerably higher number of training data points available in this experiment.

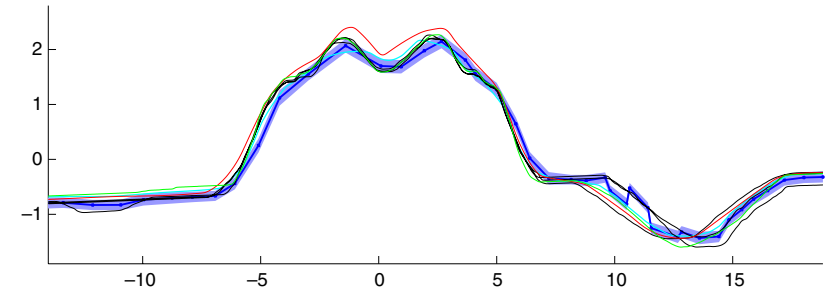
As far as the model size is concerned, the highest reduction is achieved in the Ph. E. exercise experiment, by as much as 35.9%. In the case of L8s and blocking communicative gesture, DPGMR yields moderately reduced models, by as much as 6.8% and 11.4%, respectively. It should be noted that, in this experiment, the number of training data points vastly exceeds the number of maximum model parameters, resulting from the selection of the maximum K values. Hence, we would not expect the DPGMR model to yield any significant model reduction. However, our results have indicated that DPGMR obtains much smaller models compared to GMR even under such an experimental setting. Thus, we manage to empirically prove the remarkable advantages of DPGMR in terms of the resulting computational efficiency, which



(a) *Lazy eight figures.*



(b) Ph. E. exercise.



(c) Blocking gesture.

Fig. 9. Multi-shot LbD scenarios, goodness of fit plots. **BLACK**: training data, GREEN: testing sequence, RED: GMR prediction, CYAN: DPGMR prediction, BLUE: model means and STDs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

is of crucial importance to the practical applicability of learning by demonstration algorithms in modern robotic platforms.

Concluding, in Fig. 9 we provide a graphical representation of the goodness of fit to the data of the GMR and DPGMR models. We depict the 3 training sequences (in black), the testing sequence (in green), the GMR-predicted data (in red), the DPGMR-predicted data (in cyan) and the means and standard deviations of the DPGMR model. As all trajectories are of higher dimensionality than can be depicted, this graph was obtained by effectively reducing the data dimensions to $D = 2$, by application of the Karhunen–Loeve transform (KLT). In order to calculate the corresponding covariance matrices of the DPGMR model in this low-dimensional space, we obtained $1k$ samples from the posterior distributions $\{N(\cdot|\mu_m, \Sigma_m)\}_{m=1}^{M}$, where M is the number of active components, and subsequently found the covariance matrices of the low-dimensional projections of those sampled points. We observe that the DPGMR predictions fit the data much better than the GMR-obtained ones.

5.4. Computational costs

Let us now investigate the computational costs of DPGMR, as compared to its considered competitors, that is GMR and GPR. As one may expect, based on Eqs. (29)–(45), DPGMR training for a given value of the maximum number of model states K requires exactly the same computational costs as GMR for the same value of K. Additionally, as already discussed, GMR also demands training multiple models (for different K values) to select from, which is not the case for DPGMR, which conducts inference over the proper number of model states. As such, DPGMR training eventually turns out to be much more efficient than GMR training, as the need of training multiple models to select from gets obviated in the case of DPGMR.

Regarding real-time testing, we have observed that DPGMR offers a considerable improvement in the required computational time costs over GMR, which, not surprisingly, is almost equal to the model size reduction it offers compared to GMR. This was expectable enough, given the expression of the DPGMR-generated predictions (61), which shows that a GMR and a DPGMR model with the same number of states impose exactly the same computational costs to generate predictions, which increase in a linear fashion with the effective size of the models (see, e.g., Fig. 10).

Finally, we would like to mention that both GMR and DPGMR are considerably more efficient compared to GPR. Indeed, contrary to GMR and DPGMR, GPR computational costs for prediction generation increase with the number of model training data points. As such, it came to no surprise to us that GPR required 3 orders of magnitude longer time to generate a prediction, in the case of the multi-shot scenario, and at least double the time in the case

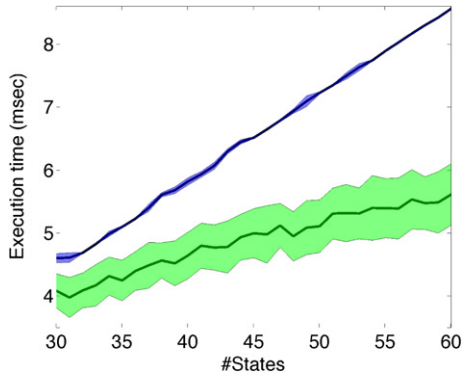


Fig. 10. Time required to generate a prediction with the GMR (BLUE) and DPGMR (GREEN) models, in the multi-shot Ph. E. exercise experiment. We observe that DPGMR is much faster than GMR for the same value of the parameter K, as a result of the induced model size reduction DPGMR offers. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of the one-shot scenario. We would also like to mention that GMR and DPGMR predictions required time of the magnitude of 10^{-3} seconds, typically lying in the interval [3.4,9] ms. On the contrary, GPR execution times started from as low as 10 ms in the one-shot experiments and were as high as 19.5 s per prediction for the multi-shot*blocking communicative gesture*experiment.

6. Conclusions

In this paper, we presented a nonparametric Bayesian approach toward trajectory-based robot learning by demonstration. The proposed approach is based on the postulation of a Gaussian mixture regression model comprising a countably infinite number of states, and is facilitated by the imposition of a Dirichlet process prior over the model states. The proposed approach allows for the automatic determination of the proper number of GMR model states, without the need of resorting to model order selection criteria, application of which is rather tedious and notorious for yielding noisy model order estimates with heavy overfitting proneness.

Our novel approach was evaluated considering a number of experimental scenarios, and its performance was compared to state-of-the-art robot learning by demonstration methodologies based on Gaussian mixture regression and Gaussian process regression. As we showed, our method, exploiting the robustness of Bayesian estimation, and the effectiveness of nonparametric Bayesian models in automatic model size determination, allows for a significant performance increase, while imposing computational requirements for trajectory regeneration similar to the GPR/GMR methods, since prediction under all these approaches eventually reduces to a sum of linear regression models. The MATLAB implementation of the DPGMR method shall be made available through the websites of the authors.

Acknowledgment

This work has been partially funded by the EU FP7 ALIZ-E project (grant 248116).

Appendix

From (28), and the expressions of the model posteriors (29)–(45), we have

$$\begin{aligned}\mathcal{L}(q) = &\sum_{c=1}^{K}\langle \log p(\boldsymbol{\mu}_c, \boldsymbol{R}_c|\lambda_c, \boldsymbol{m}_c, \omega, \boldsymbol{\Psi}_c)\\ &- \log q(\boldsymbol{\mu}_c, \boldsymbol{R}_c)\rangle_{q(\boldsymbol{\mu}_c,\boldsymbol{R}_c)}\\ &+ \langle \log p(\alpha|\gamma_1,\gamma_2) - \log q(\alpha)\rangle_{q(\alpha)}\\ &+ \sum_{c=1}^{K-1}\langle \log p(v_c|\alpha) - \log q(v_c)\rangle_{q(v_c),q(\alpha)}\\ &+ \sum_{c=1}^{K}\sum_{n=1}^{N} q(x_n = c)\left\{\langle \log p(x_n = c|\boldsymbol{\pi}(\boldsymbol{v}))\rangle_{q(\boldsymbol{v})}\right.\\ &\left.- \log q(x_n = c) + \langle \log p(\boldsymbol{y}_n|\Theta_c)\rangle_{q(\boldsymbol{\mu}_c,\boldsymbol{R}_c)}\right\} \end{aligned} \tag{63}$$

where

$$\begin{aligned}&\langle \log p(\boldsymbol{\mu}_c, \boldsymbol{R}_c|\lambda_c, \boldsymbol{m}_c, \omega, \boldsymbol{\Psi}_c)\rangle_{q(\boldsymbol{\mu}_c,\boldsymbol{R}_c)}\\ &= -\log Z(\omega, \boldsymbol{\Psi}_c) - \frac{d}{2}\log 2\pi + \frac{d}{2}\log\lambda_c\\ &\quad - \frac{\tilde{\omega}\lambda_c}{2}(\tilde{\boldsymbol{m}}_c - \boldsymbol{m}_c)^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}_c^{-1}(\tilde{\boldsymbol{m}}_c - \boldsymbol{m}_c)\\ &\quad - \frac{\lambda_c d}{2\tilde{\lambda}_c} - \frac{\tilde{\omega}}{2}\mathrm{tr}[\boldsymbol{\Psi}_c(\tilde{\boldsymbol{\Psi}}_c)^{-1}]\\ &\quad + \frac{\omega - d}{2}\left[-\log\left|\frac{\tilde{\boldsymbol{\Psi}}_c}{2}\right| + \sum_{k=1}^{d}\psi\left(\frac{\tilde{\omega}+1-k}{2}\right)\right] \end{aligned} \tag{64}$$

$$\begin{aligned}&\langle \log q(\boldsymbol{\mu}_c, \boldsymbol{R}_c)\rangle_{q(\boldsymbol{\mu}_c,\boldsymbol{R}_c)}\\ &= -\log Z(\tilde{\omega}, \tilde{\boldsymbol{\Psi}}_c) - \frac{d}{2}\log 2\pi\\ &\quad + \frac{\tilde{\omega} - d}{2}\left[-\log\left|\frac{\tilde{\boldsymbol{\Psi}}_c}{2}\right| + \sum_{k=1}^{d}\psi\left(\frac{\tilde{\omega}+1-k}{2}\right)\right]\\ &\quad - \frac{\tilde{\omega}d}{2} - \frac{d}{2} + \frac{d}{2}\log\tilde{\lambda}_c \end{aligned} \tag{65}$$

$$Z(\omega, \boldsymbol{\Psi}_c) = \pi^{d(d-1)/4}\left|\frac{\boldsymbol{\Psi}_c}{2}\right|^{-\omega/2}\prod_{k=1}^{d}\Gamma\left(\frac{\omega+1-k}{2}\right) \tag{66}$$

$$\begin{aligned}&\langle \log p(\boldsymbol{y}_n|\Theta_c)\rangle_{q(\boldsymbol{\mu}_c,\boldsymbol{R}_c)}\\ &= -\frac{d}{2}\log 2\pi + \frac{1}{2}\langle \log|\boldsymbol{R}_c|\rangle_{q(\boldsymbol{\mu}_c,\boldsymbol{R}_c)}\\ &\quad - \frac{1}{2}[\langle (\boldsymbol{y}_n - \boldsymbol{\mu}_c)^{\mathrm{T}}\boldsymbol{R}_c(\boldsymbol{y}_n - \boldsymbol{\mu}_c)\rangle_{q(\boldsymbol{\mu}_c,\boldsymbol{R}_c)}] \end{aligned} \tag{67}$$

$$\begin{aligned}&\langle (\boldsymbol{y}_n - \boldsymbol{\mu}_c)^{\mathrm{T}}\boldsymbol{R}_c(\boldsymbol{y}_n - \boldsymbol{\mu}_c)\rangle_{q(\boldsymbol{\mu}_c,\boldsymbol{R}_c)}\\ &= \frac{d}{\tilde{\lambda}_c} + \tilde{\omega}(\boldsymbol{y}_n - \tilde{\boldsymbol{m}}_c)^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}_c^{-1}(\boldsymbol{y}_n - \tilde{\boldsymbol{m}}_c) \end{aligned} \tag{68}$$

$$\langle \log|\boldsymbol{R}_c|\rangle_{q(\boldsymbol{\mu}_c,\boldsymbol{R}_c)} = -\log\left|\frac{\tilde{\boldsymbol{\Psi}}_c}{2}\right| + \sum_{k=1}^{d}\psi\left(\frac{\tilde{\omega}+1-k}{2}\right) \tag{69}$$

$$\begin{aligned}&\langle \log p(v_c|\alpha) - \log q(v_c)\rangle_{q(v_c),q(\alpha)}\\ &= \langle \log\Gamma(1+\alpha)\rangle_{q(\alpha)} - \langle \log\Gamma(\alpha)\rangle_{q(\alpha)} - \log\Gamma(1)\\ &\quad + (\langle\alpha\rangle_{q(\alpha)} - \eta_{c,2})\langle \log(1 - v_c)\rangle_{q(v_c)} - \log\Gamma(\eta_{c,1} + \eta_{c,2})\\ &\quad + \log\Gamma(\eta_{c,2}) + \log\Gamma(\eta_{c,1}) - (\eta_{c,1} - 1)\langle \log(v_c)\rangle_{q(v_c)} \end{aligned} \tag{70}$$

$$\langle \log v_c\rangle = \psi(\eta_{c,1}) - \psi(\eta_{c,1} + \eta_{c,2}) \tag{71}$$

$$\langle \log(1 - v_c)\rangle = \psi(\eta_{c,2}) - \psi(\eta_{c,1} + \eta_{c,2}) \tag{72}$$

$$\begin{aligned}&\langle \log p(\alpha|\gamma_1,\gamma_2) - \log q(\alpha)\rangle_{q(\alpha)}\\ &= -\log\Gamma(\gamma_1) + \log\Gamma(\hat{\gamma}_1) + \gamma_1\log\gamma_2 - \hat{\gamma}_1\log\hat{\gamma}_2\\ &\quad + (\gamma_1 - \hat{\gamma}_1)\langle \log\alpha\rangle_{q(\alpha)} - (\gamma_2 - \hat{\gamma}_2)\,\langle\alpha\rangle_{q(\alpha)} \end{aligned} \tag{73}$$

$$\langle \alpha \rangle = \frac{\hat{\gamma}_1}{\hat{\gamma}_2} \tag{74}$$

$$\langle \log \alpha \rangle_{q(\alpha)} = \psi(\hat{\gamma}_1) - \log(\hat{\gamma}_2) \tag{75}$$

$$\begin{aligned} &\langle \log p(x_n = c|\boldsymbol{\pi}(\boldsymbol{v})) \rangle_{q(\boldsymbol{v})} \\ &\quad = \langle \log \pi_c(\boldsymbol{v}) \rangle \\ &\quad = \sum_{c'=1}^{c-1} \langle \log(1 - v_{c'}) \rangle + \langle \log v_c \rangle \end{aligned} \tag{76}$$

where d is the dimensionality of the input $\boldsymbol{y}_n$, while $\Gamma(1) = 1$. Finally, regarding the expressions of the $\langle \pi_c(\boldsymbol{v}) \rangle$ in (47), we have

$$\langle \pi_c(\boldsymbol{v}) \rangle = \langle v_c \rangle \prod_{j=1}^{c-1} (1 - \langle v_j \rangle) \tag{77}$$

where

$$\langle v_c \rangle = \frac{\eta_{c,1}}{\eta_{c,1} + \eta_{c,2}}. \tag{78}$$

In the above, $\psi(.)$ denotes the Digamma function, and $\Gamma(\cdot)$ the Gamma function.

References

[1] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Robot programming by demonstration, in: B. Siciliano, O. Khatib (Eds.), Handbook of Robotics, Springer-Verlag, Secaucus, NJ, USA, 2008, pp. 1371–1394.
[2] B. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Robotics and Autonomous Systems 57 (2009) 469–483.
[3] A. Skoglund, B. Iliev, R. Palm, Programming-by-demonstration of reaching motions–a next-state-planner approach, Robotics and Autonomous Systems 58 (2010) 607–621.
[4] A. Billard, Y. Epars, S. Calinon, S. Schaal, G. Cheng, Discovering optimal imitation strategies, Robotics and Autonomous Systems 47 (2004) 69–77.
[5] A. Billard, M.J. Mataric, Learning human arm movements by imitation: evaluation of a biologically inspired connectionist architecture, Robotics and Autonomous Systems 37 (2001) 145–160.
[6] M. Lopes, J. Santos-Victor, A developmental roadmap for learning by imitation in robots, IEEE Transactions in Systems Man and Cybernetic - Part B: Cybernetics 37 (2007) 308–321.
[7] P. Pastor, H. Hoffmann, T. Asfour, S. Schaal, Learning and generalization of motor skills by learning from demonstration, in: Robotics and Automation, 2009. ICRA '09. IEEE International Conference on Robotics and Automation, May 2009, pp. 763 –768.
[8] B.D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Robotics and Autonomous Systems 57 (2009) 469–483.
[9] M. Lopes, J. Santos-Victor, Visual learning by imitation with motor representations, IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics 35 (2005) 438–449.
[10] Y. Demiris, B. Khadhouri, Hierarchical attentive multiple models for execution and recognition (HAMMER), Robotics and Autonomous Systems 54 (2006) 361–369.
[11] A. Ude, A. Gams, T. Asfour, J. Morimoto, Task-specific generalization of discrete and periodic dynamic movement primitives, IEEE Transactions on Robotics 26 (5) (2010) 800–815.
[12] P. Pastor, H. Hoffmann, T. Asfour, S. Schaal, Learning and generalization of motor skills by learning from demonstration, in: International Conference on Robotics and Automation, ICRA, 2009.
[13] Z. Ghahramani, M.I. Jordan, Supervised learning from incomplete data via an EM approach, Advances in Neural Information Processing Systems 6 (1994) 120–127.
[14] A.G. Billard, S. Calinon, F. Guenter, Discriminative and adaptive imitation in uni-manual and bi-manual tasks, Robotics and Autonomous Systems 54 (2006) 370–384.
[15] D. Lee, Y. Nakamura, Mimesis scheme using a monocular vision system on a humanoid robot, in: Proc. IEEE International Conference on Robotics and Automation, ICRA, 2007, pp. 2162–2168.
[16] G. Schwarz, Estimating the dimension of a model, The Annals of Statistics 4 (1978) 461–464.
[17] G. McLachlan, D. Peel, Finite Mixture Models, Wiley Series in Probability and Statistics, 2000.
[18] S. Chatzis, D. Kosmopoulos, T. Varvarigou, Signal modeling and classification using a robust latent space model based on t distributions, IEEE Transactions on Signal Processing 56 (2008) 949–963.
[19] S. Walker, P. Damien, P. Laud, A. Smith, Bayesian nonparametric inference for random distributions and related functions, Journal of the Royal Statistical Society B 61 (1999) 485–527.
[20] R. Neal, Markov chain sampling methods for Dirichlet process mixture models, Journal of Computational and Graphical Statistics 9 (2000) 249–265.
[21] P. Muller, F. Quintana, Nonparametric Bayesian data analysis, Statistical Science 19 (2004) 95–110.
[22] C. Antoniak, Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems, The Annals of Statistics 2 (1974) 1152–1174.
[23] J. Sethuraman, A constructive definition of the Dirichlet prior, Statistica Sinica 2 (1994) 639–650.
[24] D. Blei, M. Jordan, Variational methods for the Dirichlet process, in: 21st Int. Conf. Machine Learning, New York, NY, USA, July 2004, pp. 12–19.
[25] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[26] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.
[27] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.
[28] B.G. Leroux, Consistent estimation of a mixing distribution, Annals of Statistics 20 (1992) 1350–1360.
[29] G. Celeux, G. Soromenho, An entropy criterion for assessing the number of clusters in a mixture model, Classification Journal 13 (1996) 195–212.
[30] T. Ferguson, A Bayesian analysis of some nonparametric problems, The Annals of Statistics 1 (1973) 209–230.
[31] D. Blackwell, J. MacQueen, Ferguson distributions via Pólya urn schemes, The Annals of Statistics 1 (1973) 353–355.
[32] D.M. Blei, M.I. Jordan, Variational inference for Dirichlet process mixtures, Bayesian Analysis 1 (2006) 121–144.
[33] Y. Qi, J.W. Paisley, L. Carin, Music analysis using hidden Markov mixture models, IEEE Transactions on Signal Processing 55 (2007) 5209–5224.
[34] M. Jordan, Z. Ghahramani, T. Jaakkola, L. Saul, An introduction to variational methods for graphical models, in: M. Jordan (Ed.), Learning in Graphical Models, Kluwer, Dordrecht, 1998, pp. 105–162.
[35] D. Chandler, Introduction to Modern Statistical Mechanics, Oxford University Press, New York, 1987.
[36] D. Grimes, R. Chalodhorn, R. Rao, Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. in: Proc. Robotics: Science and Systems, RSS, 2006, pp. 1–8.
[37] D. Nguyen-Tuong, J. Peters, Local gaussian process regression for real-time model-based robot control, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS, 2008, pp. 380–385.
[38] Aldebaran robotics, NAO robot academic edition, http://www.aldebaran-robotics.com/.
[39] C.S. Myersand, L.R. Rabiner, A comparative study of several dynamic time-warping algorithms for connected word recognition, The Bell System Technical Journal 60 (7) (1981) 1389–1409.
[40] B. Pearlmutter, Gradient calculation for dynamic recurrent neural networks: a survey, IEEE Transactions on Neural Networks 6 (1995) 1212–1228.
[41] P. Zegers, M.K. Sundareshan, Trajectory generation and modulation using dynamic neural networks, IEEE Transactions on Neural Networks 14 (2003) 520–533.

Sotirios P. Chatzis received the M. Eng. degree in Electrical and Computer Engineering with distinction from the National Technical University of Athens, in 2005, and the Ph.D. degree in Machine Learning, in 2008, from the same institution. From January 2009 till June 2010 he was a Postdoctoral Fellow with the University of Miami, USA. Currently, he is a post-doctoral researcher with the Department of Electrical and Electronic Engineering, Imperial College London. His major research interests comprise machine learning theory and methodologies with a special focus on hierarchical Bayesian models, reservoir computing, robot learning by demonstration, copulas, quantum statistics, and Bayesian nonparametrics. His Ph.D. research was supported by the Bodossaki Foundation, Greece, and the Greek Ministry for Economic Development, whereas he was awarded the Dean's scholarship for Ph.D. studies, being the best performing Ph.D. student of his class. In his first five years as a researcher he has first-authored 23 papers in the most prestigious journals of his research field.

Dimitrios Korkinof received the Diploma in Electrical & Computer engineering (M.Sc. equivalent) from the Aristotle University of Thessaloniki, Greece. He graduated in 2010 with excellent academic achievement and 3rd in his class.

He is currently pursuing a Ph.D. at the Department of Electrical & Electronic Engineering of Imperial College London, where he is researching aspects of statistical machine learning theory with applications to robotics.

His current academic interests include Bayesian statistics, variational inference, stochastic processes and nonparametric methods for computer vision, action recognition and other robotics-related applications.

Yiannis Demiris is a senior lecturer of Imperial College London. He has significant expertise in cognitive systems, assistive robotics, multi-robot systems, robot human interaction and learning by demonstration, in particular in action perception and learning. Dr Demiris' research is funded by the UK's Engineering and Physical Sciences Research Council (EPSRC), the Royal Society, BAE Systems, and the EU FP7 program through projects ALIZ-E and EFAA, both addressing novel machine learning approaches to human–robot interaction. Additionally the group collaborates with the BBC's Research and Development Department on the "Learning Human Action Models" project. Dr Yiannis Demiris has guest edited special issues of the IEEE Transactions on SMC-B specifically on Learning by Observation, Demonstration, and Imitation, and of the Adaptive Behavior Journal on Developmental Robotics. He has organized six international workshops on Robot Learning, BioInspired Machine Learning, Epigenetic Robotics, and Imitation in Animals and Artifacts (AISB), was the chair of the IEEE International Conference on Development and Learning (ICDL) for 2007, as well as the program chair of the ACM/IEEE International Conference on Human–Robot Interaction (HRI) 2008. He is a Senior Member of IEEE, and a member of the Institute of Engineering & Technology of Britain (IET).