A Quantum-Statistical Approach Towards Robot Learning by Demonstration

Sotirios P. Chatzis, Dimitrios Korkinof, and Yiannis Demiris

Abstract—Statistical machine learning approaches have been in the epicenter of the ongoing research work in the field of robot learning by demonstration in the last years. One of the most successful methodologies used for this purpose is Gaussian mixture regression (GMR). In this paper, we propose an extension of GMR-based learning by demonstration models, to incorporate concepts from the field of quantum mechanics. Indeed, conventional GMR models are formulated under the notion that all the observed data points can be assigned to a distinct number of model states (mixture components). In this work, we reformulate GMR models, introducing some quantum states constructed by superposing conventional GMR states by means of linear combinations. The so-obtained quantum statistics-inspired mixture regression algorithm is subsequently applied to obtain a novel robot learning by demonstration methodology, offering a significantly increased quality of regenerated trajectories for computational costs comparable to currently state-of-the-art trajectory-based robot learning by demonstration approaches. We experimentally demonstrate the efficacy of the proposed approach.

I. INTRODUCTION

Robot learning by demonstration has been an active research topic in the field of robotics during the last years, encompassing methods by which a robot can learn new skills by simple observation of a human teacher, in the same way humans learn new skills by imitation [1], [2], [3]. As a result, learning by demonstration alleviates the need of programming a robot how to perform a task, which can be rather tedious and expensive, can speed up reinforcement learning techniques, since it significantly reduces the search space of the learning algorithm, while, by making robots more user-friendly, it increases the appeal of applying robots to real-life environments.

To effect these goals, robot learning by demonstration combines methods from diverse research areas, such as machine learning, computer vision, and human-robot interaction. Current approaches towards learning by demonstration can be divided into two broad categories: low-level trajectory encoding, and high-level decomposition of a skill into action units, usually referred to as symbolic encoding. In this work, we focus on the former approach towards learning by demonstration. The first step towards the implementation of a successful trajectory-based learning by demonstration algorithm is choosing the right variables to encode the movements under consideration; usually, representations in the joint space, task space, or torque space are considered [4]. Subsequently, coming up with methods capable of successfully extracting and modeling the underlying patterns in the demonstrated motions comes to the fore, being the crucial factor that eventually determines the effectiveness of a developed learning by demonstration algorithm.

Several researchers have considered the application of statistical machine learning algorithms as the effective means to facilitate extraction of the trajectory patterns underlying a set of demonstrated skills. Indeed, one of the most popular trends of work in the field of trajectory-based robot learning by demonstration consists in the investigation of the utility of Gaussian mixture regression (GMR) [5]. GMR has been shown to be very successful in encoding demonstrations, extracting their underlying constraints, and reproducing smooth generalized motor trajectories through a Gaussian mixture model (GMM) trained by means of the expectation-maximization (EM) algorithm [6], while imposing considerably low computational costs [7], [8], [9], [10]. GMR-based approaches towards 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.

One interesting and worthwhile to mention variant of GMR is locally weighted projection regression (LWPR) [11]. This variant of GMR is especially tailored to the needs of robot learning by demonstration applications with high-dimensional input signals possibly including redundant and irrelevant input dimensions. LWPR, as well as reward-weighted LWPR variants [12], have been shown to yield similar or inferior performance compared to GMR under several experimental setups entailing low-dimensionality input signals (see, e.g., [13]), but seem to perform considerably better in applications where high-dimensional input signals are being modeled.

Recently, several researchers have considered the application of concepts from quantum mechanics in the field of machine learning [14]. The main notion behind these studies consists in the generalization of the probability distribution of the postulated models by introducing a density matrix, which is a self-adjoint positive-semidefinite matrix of trace one. Indeed, it has been shown (see, e.g., [15]) that the basic probability rule of quantum mechanics, widely known as the Born Rule, which gives rise to the concept of generalized probability by introduction of a density matrix, is closely related to spectral clustering and other machine learning algorithms based on spectral theory.

In [16], a combination of the margin maximization scheme with a probabilistic modeling approach is presented, facilitated
by incorporating the concepts of quantum detection and estimation theory [17]. In [18], a quantum Markov random field was proposed, based on the utilization of quantum statistics techniques, such as the concept of the density matrix; the method was successfully applied to image restoration. More recently, a quantum-statistical-mechanical extension of the Gaussian mixture model was presented in [19]; the proposed method was based on the representation of the model marginal likelihood as a function of a density matrix, and was shown to outperform conventional Gaussian mixture models in an image segmentation task. Finally, a novel regard towards variational Bayesian (VB) inference algorithms when applied to latent Dirichlet allocation (LDA) [22], one of the most popular probabilistic graphical models for topic-based document retrieval.

Motivated by these results, in this work we introduce a novel method for Gaussian mixture regression, inspired by concepts from the field of quantum statistics. Indeed, conventional GMR models are formulated under the notion that all the observed data points can be assigned to a distinct number of model states (mixture components). In this work, we reformulate GMR models, introducing some quantum states constructed by superposing conventional GMR states by means of linear combinations. To effect this goal, we reformulate the expression of the likelihood of conventional Gaussian mixture regression models into a special form (diagonal) density matrix, and we further show that this matrix can be generalized into a more generic, non-diagonal form. Based on this novel formulation, a quantum-mechanical-inspired expression of the conditional predictive distribution of the GMR model is eventually derived, and applied to yield a quantum-statistical approach towards robot learning by demonstration. We illustrate the efficacy of the proposed approach by considering a number of demanding robot learning by demonstration scenarios, and we compare its performance with state-of-the-art trajectory-based robot learning methodologies.

The remainder of this paper is organized as follows: In Section II, Gaussian mixture regression as applied to robot learning by demonstration is introduced in a concise manner. In Section III, we provide a brief review of concepts from the field of quantum information processing, with a special focus on density matrices, which provide a quantum-inspired extension of conventional probability, and the related calculus. In Section IV, we derive the proposed quantum Gaussian mixture regression (QGMR) model, and we elaborate on its application to robot learning by demonstration. In Section V, the experimental evaluation of the proposed algorithm is performed. The final section concludes this paper.

II. GAUSSIAN MIXTURE REGRESSION FOR ROBOT LEARNING BY DEMONSTRATION

In this section, we provide a brief overview of Gaussian mixture regression as applied in the context of robot learning by demonstration.

Let us define a trajectory as a set of position data (in joint space or task space) sequentially appearing over time. GMR can be used to retrieve smooth generalized trajectories with associated predictive variances expressing the variations of the trajectory variables. Indeed, following the standard setting of all regression algorithms, GMR consists in modeling the conditional expectation of a set of response variables $\beta \in \mathbb{R}^{p_z}$ given a set of predictor variables $\beta \in \mathbb{R}^{p_z}$, by exploiting the information available in a set of training observations $\{\beta_j, \beta_j\}^N_{i=1}$. In the case of trajectory-based robot learning by demonstration, the predictor variable $\beta$ might represent the current position of the moving end-effector, with the response variable $\beta$ being the velocity that must be adopted by the end-effector for the next time-step, in order to comply with the learnt trajectory.

To effect these goals, contrary to most of the traditional regression methodologies, GMR does not approximate the regression function in a direct fashion. In contrast, GMR postulates a GMM to model the joint probability distribution of the considered response and predictor variables ($\beta$ and $\beta$), i.e. it considers a model of the form

$$p(\beta, \beta; \pi, \{\mu_i, \Sigma_i\}_{i=1}^K) = \sum_{i=1}^K \pi_i N(\beta, \beta; \mu_i, \Sigma_i)$$

where $\pi = (\pi_i)_{i=1}^K$ are the prior weights of the mixture component densities, and $N(\cdot; \mu_i, \Sigma_i)$ is a Gaussian with mean $\mu_i$ and covariance matrix $\Sigma_i$. The postulated GMM (1) is trained by means of the EM algorithm [23], using a set of training data corresponding to a number of trajectories obtained by human demonstrators. Then, using the obtained GMM $p(\beta, \beta; \pi, \{\mu_i, \Sigma_i\}_{i=1}^K)$, Gaussian mixture regression retrieves a generalized trajectory by estimating at each time step the conditional expectation $E[\beta; \beta; \pi, \{\mu_i, \Sigma_i\}_{i=1}^K]$.

Let us express the means $\mu_i$ of the component densities of the postulated GMM (1) in the form

$$\mu_i = \begin{bmatrix} \mu^\beta_i \\ \mu^\beta_i \end{bmatrix}$$

with the $\mu^\beta_i$ being the mean of the variable $\beta$, and the $\mu^\beta_i$ being the mean of the variable $\beta$. Let us also introduce the notation

$$\Sigma_i = \begin{bmatrix} \Sigma_i^\beta & \Sigma_i^\beta \Sigma_i^\beta \\ \Sigma_i^\beta & \Sigma_i^\beta \Sigma_i^\beta \end{bmatrix}$$

for the covariance matrices of the model component densities. Then, it is easy to show that, based on (1) and the assumptions (2)-(3), the conditional probability $p(\beta; \beta; \pi, \{\mu_i, \Sigma_i\}_{i=1}^K)$ of the response variables $\beta$ given the predictor variables $\beta$ and the postulated GMM yields [24]

$$p(\beta; \beta; \pi, \{\mu_i, \Sigma_i\}_{i=1}^K) = N(\beta; \bar{\mu}, \Sigma)$$

where

$$\bar{\mu} = \sum_{i=1}^K \phi_i(\beta) \left[ \mu_i^\beta + \Sigma_i^\beta \Sigma_i^\beta \Sigma_i^\beta \right]$$

and

$$\Sigma = \sum_{i=1}^K \phi_i^2(\beta) \left[ \Sigma_i^\beta - \Sigma_i^\beta \Sigma_i^\beta \Sigma_i^\beta \right]$$
and
\[
\phi_1(\beta) = \frac{\pi_{1}N(\beta | \mu_{1}, \Sigma_{1})}{\sum_{k=1}^{K} \pi_{k}N(\beta | \mu_{k}, \Sigma_{k})}
\]
(7)

Based Eq. (4), predictions under the GMR approach can be obtained by taking the conditional expectations
\[
\hat{\beta} \equiv \mathbb{E} \left( \beta | \beta; \pi, \{ \mu_{i}, \Sigma_{i} \}_{i=1}^{K} \right), \text{ i.e.}
\]
(8)

As we observe, a significant merit of GMR consists in the fact that it provides a full predictive distribution, thus a predictive variance
\[
\mathbb{V} \left( \beta | \beta; \pi, \{ \mu_{i}, \Sigma_{i} \}_{i=1}^{K} \right) = \Sigma
\]
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.

One of the most significant advantages of GMR-based robot learning by demonstration can be traced to the prediction generation procedure: Contrary to most discriminative regression algorithms (e.g., SVMs [25], and Gaussian processes [26]), the computational time required for trajectory reproduction using Eq. (5) 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 by means of the EM algorithm. This way, prediction generation under GMR reduces to the estimation of a simple weighted sum of linear models; as a result, the regression phase is processed very quickly, which is advantageous because the reproduction of smooth trajectories is fast enough to be used at any appropriate time by the robot.

Finally, apart from GMM model estimation, another significant issue that always has to be addressed is data-driven selection of the appropriate number of GMM component densities. The number of component densities in the postulated classical system can be defined as
\[
\Phi = \sum_{i=1}^{q} \varpi_{i} u_{i} u_{i}^{T}
\]
(10)

In quantum statistics, the concept of probability is extended by introducing non-diagonal elements in the density matrix \( \Phi \) of the treated system. The states of a system in quantum statistics are defined by the unit vectors \( u \); the matrix \( uu^{T} \) corresponding to a state vector \( u \) is called dyad and has trace one, i.e. \( tr(uu^{T}) = u^{T}u = 1 \). On this basis, the density matrix \( \Phi \) generalizes the concept of probability distribution, and can be defined as a mixture of dyads in the sense
\[
\Phi = \sum_{i=1}^{q} \varpi_{i} u_{i} u_{i}^{T}
\]
(11)
where \( \varpi_{i} \) is the (prior) probability of the system state represented by the dyad of \( u_{i} \). Under this construction, the probability assigned to the unit vector \( u \) and its associated dyad \( uu^{T} \) yields
\[
p(u) = tr(\Phi uu^{T}) = u^{T}\Phi u
\]
(12)

Eq. (12) is widely known within the quantum mechanics community as the Born rule. Note that the unit vectors and their associated dyads in quantum systems have a straightforward natural interpretation. Consider for example a quantum system with 4 “pure” states; then, a state vector of the form \( u = (\frac{1}{2},0,\frac{\sqrt{3}}{2},0) \) represents the mixture of the first “pure” state and the third “pure” state of the system, with probabilities \( (\frac{1}{2})^2 = \frac{1}{4} \) and \( (\frac{\sqrt{3}}{2})^2 = \frac{3}{4} \), respectively.
IV. QUANTUM GAUSSIAN MIXTURE REGRESSION

A. Using Quantum Gaussian Mixture Models to Obtain a Regression Algorithm

Let us reconsider the case of GMR and the associated GMM postulated to represent the joint distribution of the modeled predictor and response variables. Let us consider a \( K \)-component postulated GMM, as in (1). We introduce the following matrices

\[
\Psi \triangleq - \begin{bmatrix}
\log \pi_1 & 0 & \ldots & 0 \\
0 & \log \pi_2 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & \log \pi_K
\end{bmatrix}
\]  

(13)

and

\[
\Omega(\hat{\beta}, \hat{\beta}) \triangleq \begin{bmatrix}
p(\beta, \hat{\beta}|\theta_1) & 0 & \ldots & 0 \\
0 & p(\beta, \hat{\beta}|\theta_2) & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & p(\beta, \hat{\beta}|\theta_K)
\end{bmatrix}
\]  

(14)

where \( \theta_i \triangleq \{\mu_i, \Sigma_i\} \), and \( p(\beta, \hat{\beta}|\theta_i) \) is the conditional joint probability of the predictor and response variables given that they are emitted from the \( i \)-th component density of the postulated GMM, i.e.

\[
p(\beta, \hat{\beta}|\theta_i) = N(\beta, \hat{\beta} | \mu_i, \Sigma_i)
\]  

(15)

Under this scheme, the likelihood \( p(\beta, \hat{\beta}|\pi, \{\mu_i, \Sigma_i\}_{i=1}^K) \) of the postulated model, given by (1), can be equivalently expressed in the form [19]

\[
p(\beta, \hat{\beta}|\pi, \{\mu_i, \Sigma_i\}_{i=1}^K) = \text{tr} \left[ \exp \left( -H(\beta, \hat{\beta}) \right) \right]
\]  

(16)

where

\[
H(\beta, \hat{\beta}) = \Psi - \log \Omega(\hat{\beta}, \hat{\beta})
\]  

(17)

the exponential \( \exp(A) \) of a matrix \( A \) is defined as

\[
\exp(A) \triangleq \sum_{\rho=0}^{\infty} \frac{1}{\rho!} A^\rho
\]  

(18)

and the logarithm \( \log(A) \) is given by

\[
\log(A) \triangleq -\sum_{\rho=1}^{\infty} \frac{1}{\rho} (I - A)^\rho
\]  

(19)

On the basis of the above construction, we have managed to express the likelihood (1) of the postulated GMM employed by the GMR model as a function of the density matrix of a quantum system with a special form (having a diagonal density matrix). Then, based on the relevant discussions of Section III, one may generalize the results of Eq. (16) to the case of a general (symmetric) density matrix, so as to obtain a quantum-statistical approach towards GMR-based trajectory learning by demonstration. Specifically, let us extend the diagonal matrix \( \Psi \) to a symmetric \( K \times K \) matrix as follows

\[
\Psi \triangleq - \begin{bmatrix}
\log \pi_1 & \gamma & \ldots & \gamma \\
\gamma & \log \pi_2 & \ldots & \gamma \\
\vdots & \vdots & \ddots & \vdots \\
\gamma & \gamma & \ldots & \log \pi_K
\end{bmatrix}
\]  

(20)

where \( \gamma \) is a hyperparameter related to the prior probability of a “mixed” model state (comprising whichever two “pure” model states). Then, based on (17), the symmetric \( K \times K \) matrix \( H(\beta, \hat{\beta}) \) of the model yields

\[
H(\beta, \hat{\beta}) = - \sum_{k=1}^{K} \sum_{k'=1}^{K} \xi_{kk'} \Delta_{kk'}
\]

where the coefficients \( \xi_{kk'} \) are defined as

\[
\xi_{kk'} \triangleq \begin{cases} 
\log \left\{ \pi_k p(\beta, \hat{\beta}|\theta_k) \right\}, & \text{if } k = k' \\
\gamma, & \text{if } k \neq k'
\end{cases}
\]

(21)

and the \( \Delta_{kk'} \) are \( K \times K \) matrices the \((l, l')\)th elements of which are defined as

\[
(\Delta_{kk'})_{ll'} \triangleq \delta(k - l)\delta(k' - l')
\]  

(22)

where \( \delta(\cdot) \) is the Kronecker’s delta function.

A GMM with likelihood expression of the form (16), where the density matrices \( H(\beta, \hat{\beta}) \) are given by the generalized expression (21), is usually referred to as a quantum Gaussian mixture model (QGMM) [19]. An issue of this formulation of QGMMs is that the integral in the denominator (regularization constant) of their likelihood (16) is difficult to compute analytically. To alleviate these issues, typically the following approximation is adopted [19]

\[
p(\beta, \hat{\beta}|\pi, \{\mu_i, \Sigma_i\}_{i=1}^K) = \frac{\text{tr} \left[ \exp \left( -H(\beta, \hat{\beta}) \right) \right]}{\text{tr} \left[ \exp \left( -\Psi \right) \right]}
\]  

(23)

Apparently, a GMR model the employed GMM of which is of QGMM type departs from the classical definition of the GMR model, providing a quantum-statistical approach towards GMR. The entailed density matrices of the so-derived model include some quantum effects, and are based on states constructed by superposing the model states corresponding to the mixture components of a classical GMR model.

B. Proposed Algorithm

Definition 1. We define as the quantum Gaussian mixture regression (QGMR) model a Gaussian mixture regression model employing a GMM with likelihood of the form (23), with the density matrices \( H(\beta, \hat{\beta}) \) given by the generalized expression (21).

To develop a trajectory-based robot learning by demonstration algorithm based on the QGMR framework, we have to: (i) provide an algorithm for model training using a set of human-generated demonstrations; and (ii) derive the expression of the conditional density of the response variables given the estimated joint distribution model.
1) Model Training: Let us consider a model fitting dataset comprising $N$ samples $\{y_j\}_{j=1}^N \triangleq \{[y_j; \hat{y}_j]\}_{j=1}^N$. Then, likelihood maximization for the postulated model (23) can be shown to yield the following estimators for the model parameters [19]

$$
\begin{align*}
\pi_i &\approx \frac{\sum_{j=1}^N \psi_{ij}}{N} \\
\mu_i &= \frac{\sum_{j=1}^N \psi_{ij} y_j}{\sum_{j=1}^N \psi_{ij}} \\
\Sigma_i &= \frac{\sum_{j=1}^N \psi_{ij} (y_j - \mu_i)(y_j - \mu_i)^T}{\sum_{j=1}^N \psi_{ij}}
\end{align*}
$$

and

$$
\psi_{ij} \triangleq \frac{\exp(-H(\beta_j, \beta_j))}{\exp(-H(\beta_j, \beta_j))}
$$

where

$$
\varphi_{ij} \triangleq \frac{\Delta_{ii} \exp(-H(\beta_j, \beta_j))}{\exp(-H(\beta_j, \beta_j))}
$$

and the matrices $\Delta_{ii}$ are given by (22).

2) Predictive Density: Having obtained the estimators of the QGMR model parameters, we can now proceed to the derivation of the model predictive density, that is the conditional density $p(\beta; \beta_j, \pi_j, \{\mu_i, \Sigma_i\}_{i=1}^K)$. For this purpose, we rely on the adoption of the classical expression

$$
p(\beta; \beta_j, \pi_j, \{\mu_i, \Sigma_i\}_{i=1}^K) = N(\hat{\mu}, \hat{\Sigma})
$$

for the predictive density of our model, where the predictive mean is given by

$$
\hat{\mu} = \sum_{i=1}^K \phi_i(\beta) \left[\mu_i + \Sigma_i \Sigma_i^{\beta \beta} (\Sigma_i^{\beta \beta})^{-1} (\beta - \mu_i^{\beta \beta}) \right]
$$

and the predictive covariance reads

$$
\hat{\Sigma} = \sum_{i=1}^K \phi_i^2(\beta) \left[\Sigma_i - \Sigma_i^{\beta \beta} (\Sigma_i^{\beta \beta})^{-1} \Sigma_i^{\beta \beta} \right]
$$

with the $\mu_i^{\beta \beta}$, $\mu_i$, $\Sigma_i$, $\Sigma_i^{\beta \beta}$, and $\Sigma_i^{\beta \beta}$ defined as in (2) and (3), and the estimates of $\mu_i$ and $\Sigma_i$ given by (25) and (26), respectively.

To determine the model state weight values $\phi_i(\beta)$ in Eqs. (29)-(30) for the QGMR model, we first consider the quantum GMM of the predictor variables

$$
p(\beta; \beta_j, \pi_j, \{\mu_i, \Sigma_i\}_{i=1}^K) = \frac{\exp(-H(\beta))}{\exp(-H(\beta))}
$$

where

$$
H(\beta) = 
\begin{bmatrix}
\log \left\{\pi_1 p(\beta; \mu_1^{\beta \beta}, \Sigma_1^{\beta \beta})\right\} & \cdots & \gamma \\
\gamma & \cdots & \gamma \\
\vdots & \vdots & \vdots \\
\gamma & \cdots & \log \left\{\pi_K p(\beta; \mu_K^{\beta \beta}, \Sigma_K^{\beta \beta})\right\}
\end{bmatrix}
$$

and $\Psi$ is defined in (20). Based on this model, selection of the values of $\phi_i(\beta)$ is conducted by setting them equal to the responsibilities of the quantum GMM (31) of the predictor variables, i.e.

$$
\phi_i(\beta) = \frac{\text{tr}[\Delta_{ii} \exp(-H(\beta))] \exp(-H(\beta))}{\text{tr}[\exp(-H(\beta))]}(33)
$$

V. EXPERIMENTAL EVALUATION

In this section, we provide a thorough experimental evaluation of the QGMR algorithm, in a series of applications dealing with robot learning by demonstration, and compare its performance to state-of-the-art methods in the field. Our source codes have been developed in Matlab R2010b, and were run on a Macintosh platform with an Intel Core i7 2.67 GHz CPU, and 4 GB RAM, running Mac OS X 10.6. The model-estimated values generated in MATLAB were sent in real-time to the NAO robot by means of a MATLAB-NAO communication protocol written in Python. Therefore, the results reported here are obtained from the actual robot, and not by means of simulation.

In our experiments, we employ a humanoid robotic platform, namely the NAO robot (academic edition), a humanoid robot with 27 degrees of freedom (DoF) [29]. The predictor variable $\beta$ used by the considered models is the position vector of the robot joints, whereas the response variable $\hat{\beta}$ is the velocity vector of the robot joints, that is the velocity that should be imposed on the robot joints so as to remain on the learnt trajectory.

Our approach is compared against two popular state-of-the-art methods for robot learning by demonstration, namely Gaussian mixture regression (GMR) [10], and the local Gaussian process regression (LGPR) method of [30]. The latter method clusters the input space into smaller subsets, and fits a dedicated Gaussian process regression model for each one of these subspaces. As such, it shares similarities with the GMR and QGMR methods, which also divide the input space into subspaces and postulate different regression models on each one of these subspaces. We utilize several performance metrics for our comparisons, selected on the basis of the individual characteristics of each experiment. Regarding model order selection for the GMR and QGMR methods, we repeat our experiments for various numbers of model states $Q$, and detect the values yielding the lowest generalization error rates. All our experiments are conducted with the hyperparameter $\gamma$ of the QGMR model density matrix set equal to $\gamma = 2$, a value heuristically determined to work well for our model.

In an attempt to account for the effect of bad local optima where the expectation-maximization algorithm might get trapped into in cases of poor model initialization, all our experiments have been executed multiple times, each time with different $k$-means initializations for the training algorithms of the evaluated models. Means and standard deviations of the performance of the compared algorithms over the executed multiple runs are provided, and the statistical significance of these results is assessed. Finally, we would like to underline that, in all our experiments, we have ensured that the total number of model component densities is at least one order of magnitude less than the number of available training data points (see also Table I and Figs. 4 and 6). This is a good method to ward off the possibility of model overfitting [23].
A. One-shot learning

In the following set of experiments, we evaluate the ability of the proposed model to learn and reproduce a skill from a single demonstration. We briefly describe the experiments below; more details concerning the used datasets are provided in Table I.

1) Lazy figure 8: 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 figure 8 (Fig. 1). The lazy figure 8 (L8) generation task is a classical benchmark for pattern generation methodologies [31], [32]. 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 dataset used 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 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. 2).

3) Lower body motion: Finally, we examine an experimental case involving movement of the lower robot body, simulating a lower abdominal muscle exercise (Fig. 3). This is one of the scenarios under investigation of the ALIZ-E EU FP7 project (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.

In this set of experiments, we wish to investigate the overall accuracy of reproduction of the learned trajectories. Therefore, we choose as our performance metric the mean square error (MSE) calculated over the whole length of the obtained trajectories. Especially in the case of the L8s experiment, we also compute the end-effector mean square error, so as to assess the accuracy of the eventual reproduction of the figures. The training trajectories are presented to the NAO robot by means of kinesthetics\(^2\); during this procedure, joint position sampling is conducted, with the sampling rate equal to 10 Hz. The number of joints actively participating in each experiment varies according to the specification of the performed motion types.

In our experiments, we use the training sequences obtained from human demonstrators (through kinesthetics) without further pre-processing. As such, our datasets also contain information pertaining to joints with minor contribution to the learned movements, thus further increasing the difficulty of the tasks. According to our experimental scenario, during the testing phase, the evaluated algorithms are initialized at points obtained by adding uniformly distributed noise \(U(0, 1)\) to the initial points of the training sequences, and the algorithms are executed so as to regenerate the (rest of the) learnt trajectories.

The error means and standard deviations resulting from 20 independent, but common for both methods, random initializations of the GMR and QGMR models are presented in Fig. 4. The best mean MSE for each method and the associated standard deviation, along with the LGPR error, are presented in Table II. In Table II, we have calculated the mean MSE along with its standard deviation for each value of the number of states, and presented the best result. We have also applied the Student-t statistical test on the obtained results to establish the statistical significance of our findings. The outcome of this statistical test is presented in Table IV; based on our findings, we can definitively deduce that, in every experiment, there is a statistically significant difference between the two main evaluated methods, namely the GMR and the proposed QGMR algorithm.

Additionally, from Fig. 4 we observe that GMR suffers from higher volatility and higher MSE errors than the QGMR approach, which achieves not only better but also more consistent results. Finally, we observe that an insufficient number of states is translated in slightly higher MSEs for both the evaluated methods, but also in higher model performance volatility. From Table II we can conclude that the mean error obtained by the GMR is from twice as high to approximately one order of magnitude higher than the QGMR. LGPR yielded competitive results, clearly inferior though to our approach in most cases.

As previously mentioned, especially in the case of the L8s experiment, we are interested to evaluate the end-effector error. This result is obtained by reproducing the original demonstration, as well as the predicted data generated by

---

\(^1\)Also referred to as “traveling”.

\(^2\)Manually moving the robot’s arms and recording the joint angles.
both the GMR and QGMR methods. Specifically, the NAO robot is given the joint angles of the demonstration and the model-predicted data. The end-effector positions are recorded and the resulting trajectories obtained by using the GMR and QGMR-generated predictions are then separately aligned with the end-effector data from the original demonstration. This way, we ensure fairness for both methods. We have chosen a case where both methods’ performance is neither the best nor the worst, for which the corresponding end-effector MSE results are presented in Table III.

### B. Multi-shot learning

In this experimental case, we use 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. More specifically, the tasks in question are the same ones described in the one-shot experimental scenario, namely the Lazy Figure 8, Ph. Education Exercise, and Blocking communicative gesture. The training trajectories are again presented to the NAO robot by means of kinesthetics; during this procedure, joint position sampling is conducted, with the sampling rate now equal to 20 Hz. For each of the three tasks, we have recorded 4 demonstrations and used 3 for training and 1 for testing purposes. Due to the temporal variations observed in the demonstrations, we have pre-processed the sequences using Dynamic Time Warping (DTW) [33], a method first used in speech recognition for signal alignment, combined with a low-pass filter to smooth the resulting trajectories. In Table I, we present some details concerning the number of points and the dimensionality of each dataset. It should be noted that the number of points for the multi-shot experiments is considerably higher compared to the one-shot scenario due to the higher sampling frequency as well as the oversampling that occurs as a result of the alignment of the trajectories.

The error metric used in this case is the mean square error, as we are again interested in the overall accuracy of movement reproduction. We would like to emphasize that, in our experiments, comparison of the reproduced trajectories to the demonstrated trajectories has been conducted using the time-aligned trajectories obtained by application of Dynamic Time Warping (DTW). As such, the calculated MSE statistics provide a genuine assessment of the performance difference between the evaluated methods, not affected by reproduction delays.

Additionally, especially for the needs of the Lazy Figure 8 experiment, we have also considered the end-effector error in a manner similar to the one-shot scenario (Table III). This way, we are able to definitively show that the demonstrations are not distorted by the preprocessing, that the proposed method yields better end-effector error, and that the proposed QGMR approach is able to successfully reproduce the demonstration in occasions where the GMR fails to do so. The visual result of the end-effector data is presented in Fig. 5, where we can clearly see that the QGMR result is closer to the original demonstration, and the GMR performs poorly in the reproduction of the L8 figures.

We have calculated the mean and standard deviation of the errors from 100 repetitions of the training and testing
Table II: One-shot and multi-shot learning experiments: Best mean MSE results for all evaluated methods.

<table>
<thead>
<tr>
<th>Task</th>
<th>One-shot learning MSE</th>
<th>Multi-shot learning MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GMR</td>
<td>LGPR</td>
</tr>
<tr>
<td>Blocking</td>
<td>3.8 \cdot 10^{-4} (±1.4 \cdot 10^{-4})</td>
<td>8.5 \cdot 10^{-5}</td>
</tr>
<tr>
<td>Ph. Education</td>
<td>0.0017 (±9.7 \cdot 10^{-4})</td>
<td>10.3 \cdot 10^{-5}</td>
</tr>
<tr>
<td>Lazy figure 8s</td>
<td>3.7 \cdot 10^{-4} (±6.1 \cdot 10^{-5})</td>
<td>2.4 \cdot 10^{-4}</td>
</tr>
</tbody>
</table>

Figure 4: One-shot learning experiments: Mean and standard deviation of the mean square error over the executed repetitions as a function of model size. **Blue:** GMR, **Green:** QGMR.

Table III: End-effector mean square errors for the lazy figure 8s experiment.

<table>
<thead>
<tr>
<th>Task</th>
<th>One-shot learning</th>
<th>Multi-shot learning</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GMR</td>
<td>LGPR</td>
<td>QGMR</td>
</tr>
<tr>
<td>Joint angles error</td>
<td>3.8 \cdot 10^{-4}</td>
<td>6 \cdot 10^{-5}</td>
<td>0.0059</td>
</tr>
<tr>
<td>End-effector error</td>
<td>0.0206</td>
<td>0.0032</td>
<td>0.2925</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Task</th>
<th>One-shot</th>
<th>Multi-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Null hypothesis</td>
<td>p-value</td>
</tr>
<tr>
<td>Blocking</td>
<td>rejected</td>
<td>0.0011</td>
</tr>
<tr>
<td>Ph. Education</td>
<td>rejected</td>
<td>7 \cdot 10^{-10}</td>
</tr>
<tr>
<td>Lazy Figure 8s</td>
<td>rejected</td>
<td>5.6 \cdot 10^{-25}</td>
</tr>
</tbody>
</table>

Figure 5: Multi-shot learning experiments: Visual representation of the end-effector data from both the original demonstration and the GMR and QGMR methods.
Figure 6: Multi-shot learning experiments: Mean and standard deviation of the mean square error over the executed repetitions as a function of model size. **Blue:** GMR, **Green:** QGMR.

Figure 7: Goodness of fit graph: **Black:** Training set, **Green:** Testing set, **Red:** GMR, **Blue:** QGMR.
procedures with common initialization for the GMR and the QGMR methods in each repetition. The obtained results are presented in Fig. 6. In Table II, we present the best MSE errors obtained by all the evaluated methods, and in Table IV we show the results of the Student-t test used to assess the statistical significance of our findings. As we observe, compared to our approach, the GMR method achieves an optimal error 2 times higher and one order of magnitude higher in the L8 and Ph. Education experiments, respectively. Fig. 6 also reveals lower errors and higher consistency of the results. As far as the “Blocking” experiment is concerned, the optimal error results are much closer for the two evaluated methods. However, the QGMR method again achieves lower and more consistent errors, especially for a higher number of states. Moreover, the Student-t test rejects, with very high certainty, the null hypothesis that the mean error values of the two methods belong to the same random distribution. Similar to the one-shot learning scenario, GMR is unable to successfully reproduce any of the learned trajectories. Regarding LGPR, we observed competitive performance, clearly inferior though to our approach.

Concluding, in Fig. 7 we present a graphical representation of the fit of the model to the data, where we depict the 3 training sequences (black), the testing sequence (green), the GMR-predicted data (red), and the means and standard deviations of the QGMR model. As all trajectories are of high dimensionality, this graph was obtained by effectively reducing the data dimensions to \( D = 2 \), by application of the Karhunen-Loève transform (KLT). In order to calculate the corresponding covariance matrices of the model in this low-dimensional space, we sampled from the distributions \( \{N(\mu_j, \Sigma_j)\}_{j=1}^Q \), where \( Q \) is the number of model states, and subsequently found the covariance matrices of the low-dimensional projections of the sampled data. We observe that the QGMR model fits the data very well, which is not always the case for the GMR method. We also observe that in those segments where the demonstration trajectories differ from each other, there is also a high uncertainty of the model fit.

VI. CONCLUSIONS

In this paper, we presented a quantum-statistical approach towards trajectory-based robot learning by demonstration. The proposed approach is based on an extension of conventional Gaussian mixture regression formulations, effected by introducing the concept of quantum states, which can be constructed by superposing conventional GMR model states in a linear fashion. To derive our model, we reformulated the expression of the likelihood of conventional Gaussian mixture regression models into a special form (diagonal) density matrix, and we further showed that this matrix can be generalized into a more generic, non-diagonal form.

The so-obtained quantum GMR model was applied to yield a quantum-statistical approach towards robot learning by demonstration, and its efficacy was illustrated by considering a number of demanding robot learning by demonstration scenarios, with its performance being compared to state-of-the-art robot learning by demonstration methodologies. As we showed, our method allows for a significant performance increase, while imposing computational requirements similar to its alternatives, since prediction under all these approaches eventually reduces to a sum of linear regression models. Based on our results, we can definitively conclude that the proposed approach is especially suitable for learning complex demonstration trajectories, under both a sparse one-shot and a multi-shot learning setting. The MATLAB implementation of the QGMR method shall be made available through the website of the authors.

ACKNOWLEDGMENT

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

REFERENCES


