GESTURE IMITATION USING MACHINE LEARNING TECHNIQUES

Itauma Isong Itauma, Hasan Kivrak, and Hatice Kose

Faculty of Computer Engineering, Istanbul Technical University, Turkey  
{itauma, kivrakh, hatice.kose}@itu.edu.tr

ABSTRACT
This study is a part of an ongoing project which aims to assist in teaching Sign Language (SL) to hearing-impaired children by means of non-verbal communication and imitation-based interaction games between a humanoid robot and a child. In this paper, the problem is geared towards a robot learning to imitate basic upper torso gestures (SL signs) using different machine learning techniques. RGBD sensor (Microsoft Kinect) is employed to track the skeletal model of humans and create a training set. A novel method called Decision Based Rule is proposed. Additionally, linear regression models are compared to find which learning technique has a higher accuracy on gesture prediction. The learning technique with the highest accuracy is then used to simulate an imitation system where the Nao robot imitates these learned gestures as observed by the users.

1. Introduction
This study is based on the idea of developing a robust interface for rotation invariant Gesture Recognition. It involves the view-based detection and recognition of static arm gestures by using a single camera. With the availability of RGBD cameras like Kinect, the job on image processing in determining a good feature set for gesture classification became easier.

Also, the study provides a system for computing the joint angles based on the kinematics of the skeletal view relative to the Kinect camera and these values are passed to a Nao robot simulated environment. Based on the Nao’s degree of freedom and kinematic constraints, estimated joint angles (recognized gestures) are simulated to reflect a sense of imitation. This study contributes to the development of different learning techniques that recognize different human gestures.

The previous work is presented in Section 2. The implementation and the methods for data preparation are detailed in Section 3.1. Next, we performed offline supervised learning (Section 3.2) to compute parameters of different models in the literature. A novel method called Decision Based Rule is also proposed. Finally, the computed parameters of the different models are tested to determine which learning technique has a higher accuracy and less error in prediction as shown in Section 4. Conclusion and future work are presented in Section 5.

2. Background
The purpose of this study is to design a system from which a humanoid robot can imitate upper body gestures with the aim of using it to teach sign languages to people (especially children) through interactive games. Non-verbal interaction games based on turn-taking tested with primary school students successfully in [1]. Based on this game structure, in our new game with SL, the robot is able to express a word in Sign Language (SL) among a set of chosen words using hand movements, body and face gestures. The aim of this game is to assist sign language tutoring especially for preschool children [2] [3].

Further more, this study will emphasize how the robot is taught to imitate human gestures (SL signs). The problem is simplified by teaching the robot several basic gestures, including some SL signs (from American SL, and Turkish SL) as shown in Figure 1 using different machine learning techniques. The robot learns to imitate the human using the learned model. The RGBD sensor is used to compute joint angles from the skeletal model of the human using an approach based on Pythagoras theorem [4].

Our work is inspired by previous studies which use three-dimensional (3D) Cartesian coordinates (XYZ position). In [5], a system that recognizes gestures using 3D trajectories consisting of a reduced set of key-points was proposed. In [6], the authors described trajectory learning from multiple demonstrations with a 3D dimensional model of the human hand for pick and place operations. In [7], the authors proposed the Maximum Margin algorithm that solves imitation problems by learning linear mappings from features to cost functions in a planning domain. Also, [7] demonstrated that imitation learning of long horizon and goal-directed behavior can be naturally formulated as a structured prediction problem over a space of policies. In [8], the authors discuss that imitation learning is reduced to a regression problem. In addition, [8] demonstrated the validity of their approach by learning to map motion capture data from human actors to a humanoid robot, and the composition of several regression models yields qualitatively better imitation results than using a single, more complex regression model.

3. Implementation
Several works in the literature make use of supervised learning such as [8] and [9]. In supervised learning, a feature vector and a target label \( x_1, x_2, x_3, \ldots, x_n \rightarrow Y \) are assumed to be given. For example, the feature vectors are different computed hand joint angles and the target label is the desired gesture. Machine learning is carried out on past experience to create a hypothesis that fits the features to the labels. The goal is to choose a function among a family of functions \( f(x) = Y \) that allows us to predict gestures \( Y \) based on new
feature data $x$. Ten different discriminative hand joint angles ('RShoulderRoll', 'RShoulderPitch', 'RElbowRoll', 'RElbowYaw', 'RHand', 'LShoulderRoll', 'LShoulderPitch', 'LElbowRoll', 'LElbowYaw', 'LHand') are used for the feature set. In supervised learning, given a dataset:

$$\text{DataSet}_{m,n} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,m} \end{pmatrix} \rightarrow Y_1$$

Given any $x_m$ or future vector $x$, it should predict the target label $Y$. \( f(x_m) = Y \).

### 3.1. Data Preparation

A platform known as OpenNI (Open Natural Interaction) provided by Prime Sense [10] is used to interface with the Kinect sensor unit. It offers an efficient solution to track a person and his joints in 3D space. NiUserTracker sample code [10] is used as a base for our implementation. The depth-sensor of Kinect is provided by Prime Sense [10] is used to interface with the Kinect as a base for our implementation. The depth-sensor of Kinect is used to gather depth information that enables OpenNI to gather the xyz coordinate system of the scene. Using an RGB camera instead would be computationally expensive and will not be efficient in real-time applications. The depth information is used for segmentation and 3D scene recognition for tracking the calibrated human body.

Basic mathematical geometry of 3D vectors is used in the computation of joint angles. A one-to-one mathematical model for human motion imitation using the calibrated skeletal view derived from OpenNI connected to the RGBD sensor was developed [11]. Using OpenNI, the human body is calibrated and the skeletal view is segmented and tracked.

#### 3.1.1. Joint Angle Computation

- Get Skeleton Joint Positions rightShoulderJoint, rightElbowJoint, rightHandJoint in terms of (x,y,z) plane.
- Build the directional vectors between the joints such rightShoulderElbow, rightHandElbow etc.
- Compute roll, pitch, yaw rotation matrix by projecting the vectors to the xy/yz/xz-plane or axis.
- Compute the vector in dependencies to the previous joints (elbow depends on shoulder and shoulder to torso).

The dot product of the directional vectors (righthandElbow and shoulderElbow) between the shoulder and elbow and the elbow to hand is computed in order to calculate the angles for the RightElbowRoll and LeftElbowRoll.

#### 3.1.2. Angle between 3D Vectors [12]

The dot product is used for computing the angle: \( \cos \theta \) is equal to dot product of two vectors. The formula for the angle \( \theta \) between two vectors is:

$$\cos \theta = \frac{f \cdot g}{\|f\| \cdot \|g\|} \quad (1)$$

We show that it is possible to use the RGBD camera (Kinect) to implement an imitation system. The computation was based on points in a 3D Cartesian coordinate system.

A survey was carried out on several users as shown in Table 1. This is the result of our previous work which from observation, the simulated Nao robot imitated actions considered unsafe to apply on a real Nao robot. The problem was due to noise in the environment and the misalignment of the segmented skeletal image from which the human pose was computed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Demo</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>User3</td>
<td>male</td>
<td>[11]</td>
<td>There are some gestures that can not be performed on the right hand due to wrong computation.</td>
</tr>
</tbody>
</table>

#### 3.1.3. Good Hand Joint Feature

Based on the one-to-one robot control shown in the previous section and the observations from different users, we noticed that there are manipulations that can not be implemented on the real Nao robot due to its degree of freedom and singularities. So we decided to make use of different machine learning techniques to create a system in which the Nao robot learns the observed human gesture and performs the right imitation. We implemented this learning system using Decision Based Rule [13] and linear regression learning techniques [8], [9] for offline learning of joint feature parameters for robot control which will be explained in the next section. Figure 2 shows the system design for Joint Angle computation.

#### 3.2. Learning Techniques For Model Representation

The task is to predict the correct behavior based on user joint angles of the different features. This problem is tackled as both a regression whereby we predict real-valued output and a classification whereby we predict discrete-valued output. Table 2
shows the experimental statistics carried out in this study in generating both training set and test set.

- **Notations used:**
  - m= number of training examples
  - \(x', y'\)= input features and output behaviors
  - \(X\)= input feature set matrix
  - \((x, y)\)= one training example
  - \((x^{(i)}, y^{(i)})\)= \(i^{th}\) training example

<table>
<thead>
<tr>
<th>Tablo 2: Data Set.</th>
<th>Training values</th>
<th>Test values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Users</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>n= Number of features</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>m= Number of examples</td>
<td>50</td>
<td>45</td>
</tr>
<tr>
<td>Number of behaviors</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

In designing a learning algorithm, we need to decide how we represent the hypothesis. Figure 3 shows a general representation of deciding on a hypothesis that can predict a behavior from the set of joint angles, \(n, h\) the hypothesis maps \(X's\) to \(y's\).

### 3.2.1. Linear Regression Analysis (LRA)

In linear regression, the idea is to choose \(\theta'\)s so that \(h_\theta(X)\) is close to \(y\) for our training examples \((x, y)\). This can be solved as an optimization problem:

\[
\min_{\theta_0, \theta_1} \sum_{i=1}^{m} (h_\theta(X^{(i)}) - y^{(i)})^2
\]

(2)

Find the value of \(\theta_0, \theta_1\) that makes the equation minimized. This is a squared error function used for regression problems. In this study, we have a set of 10 features. The hypothesis is:

\[
h_\theta(X) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_{10} X_{10}
\]

(3)

For convenience of notation, define \(X_0 = 1\)

\[
X = \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{10} \end{bmatrix} \in \mathbb{R}^{10+1}, \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{10} \end{bmatrix} \in \mathbb{R}^{10+1}
\]

\[
h_\theta(X) = \theta^T X
\]

This is known as **Multivariate linear regression**. We have multiple features in which we try to predict the value \(y\). In this study, both Gradient Descent and Normal Equation learning models (Figure 4) are used and compared.

#### 3.2.2. LRA with Gradient Descent

For \(n \geq 1\) the gradient descent algorithm is:

\[
\text{repeat}\{ \\
\quad \theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta) \\
\} \quad \text{(simultaneously update } \theta_i \text{ for every } i = 0, \ldots, n) \\
\text{where } \frac{\partial}{\partial \theta_i} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_i^{(i)}
\]

The gradient descent is used for minimizing the cost function \(J(\theta)\), \(\alpha\) is the learning rate, using the training value as represented in Table 2.

The gradient descent was run until convergence to find the final values of \(\theta\). Next, the value of \(\theta\) was used to predict the behavior of users from the computed joint angle features. Out of 50 behaviors from 10 users, this system had a 70% accuracy.

#### 3.2.3. LRA with Normal Equation

This gives a better way to solve for the parameter \(\theta\) analytically rather than solving it iteratively using Gradient Descent. In our experiment we have \(m = 50\) training examples. In order to implement this Normal Equation Method, an extra column \(X_0 = 1\) is added. Then we created a matrix of all features and called it \(X\). We did the same for label \(y\) which we want to predict. The normal equation is:

\[
\theta = (X^T X)^{-1} X^T y
\]

(4)

which gives the value of \(\theta\) that minimizes the cost function. Using Normal Equation, feature scaling is not needed. Out of 50 behaviors from 10 users, this system had a 72% accuracy.

### 3.3. Online Imitation System

Online learning from data sequences is a challenging aspect of machine learning. In this study, we developed a novel idea called **Decision Based Rule (DBR)** based Online Learning, which learns from streams of generated pose (position and orientation) of a human hand joint as shown in Figure 5. Online learning allows learning from a continuous stream of data (Figure 6). Decision is based on the current instance of data. It can adapt to changing user poses (i.e, if \(p(y|x, \theta)\) changes over time).

#### 3.3.1. DBR Human-Robot Imitation Model

- Given gestures: Hands Up, Sideways gestures, . . . .
- Goal: We derived a cost function that takes into consideration the space of manipulation.
The learning model was trained to distinguish between side, forward, pi, up and down gestures. Table 3 shows a confusion matrix that summarizes the results of linear regression normal equation algorithm. 50 behaviors were tested from 10 users - 10 side, 10 forward, 10 pi, 10 up and 10 down gestures. In this confusion matrix, the model predicted two forward gesture, of the ten actual side gestures, and of the ten forward gestures, it predicted that one was a up gesture and four were pi gestures. All correct predictions are located in the diagonal of the table, so it is easier to visually inspect the table for errors, as these are represented by any non-zero values outside the diagonal.

5. Conclusion

In this paper, we present the results of the different learning techniques used for imitation. Out of 50 behaviors from 10 users, the best result for linear regression was 72% accuracy and that of Decision Based Learning was 96%. Improving the feature set will lead to a better and faster imitation model. Decision Based Learning, a new imitation method for the humanoid robot was proposed.

The different methods in the literature are expensive in terms of computational cost, memory consumption and gesture recognition ability. [9] for example, is computationally expensive, unlike our approach using DBR. Our approach refers to the position of the X-, Y-, and Z-axis of the arm in 3D space in making a decision of the gesture.

6. Acknowledgment

This work was supported by Istanbul Technical University Scientific Research Projects foundation under the contract BAP 34255.

7. REFERENCES