Towards smart robots: rock-paper-scissors gaming versus human players

Gabriele Pozzato, Stefano Michieletto, and Emanuele Menegatti

Intelligent Autonomous Systems Lab (IAS-Lab)
Department of Information Engineering (DEI)
University of Padova
{pozzatog,stefano.michieletto,emg}@dei.unipd.it
http://robotics.dei.unipd.it

Abstract. In this project a human robot interaction system was developed in order to let people naturally play rock-paper-scissors games against a smart robotic opponent. The robot does not perform random choices, the system is able to analyze the previous rounds trying to forecast the next move. A Machine Learning algorithm based on Gaussian Mixture Model (GMM) allows us to increase the percentage of robot victories. This is a very important aspect in the natural interaction between human and robot, in fact, people do not like playing against "stupid" machines, while they are stimulated in confronting with a skilled opponent.

1 Introduction

This research aims to develop an application to let users playing with a smart robot in a series of rock-paper-scissors games. Users can interact naturally with the robot without any remote controls or keyboards: this application wants to be as user-friendly as possible. Humans can communicate with the robot via an RGB-D sensor using a default set of positions and gestures. The used robot is a NXT of the LEGO Mindstorms series (Fig. 1).

Vision system provided by the sensor and control of the NXT are coordinated using ROS (Robot Operative System). Analyzing depth images given by the sensor we are able to understand positions and gestures of the human being. The application development consists of three steps. At first, the system recognizes the user pose, at the pose we associated the number of games to play, a game is composed by three rounds. Once a round is started, a hand gesture recognition algorithm developed by LTTM (Multimedia Technology and Telecommunications Lab) looks at the move played by the human. At the same time, the system chooses the robot gesture, using an Artificial Intelligence (AI) algorithm we developed. The algorithm is able to infer human moves starting from its previous game experience.

The remainder of the paper is organized as follows: in Section 2 the robotics interface is presented. In Section 3 the learning system used to infer human strategies is described. Section 4 presents the tests performed on the AI algorithm and the related results. Finally, Section 5 contains conclusions.
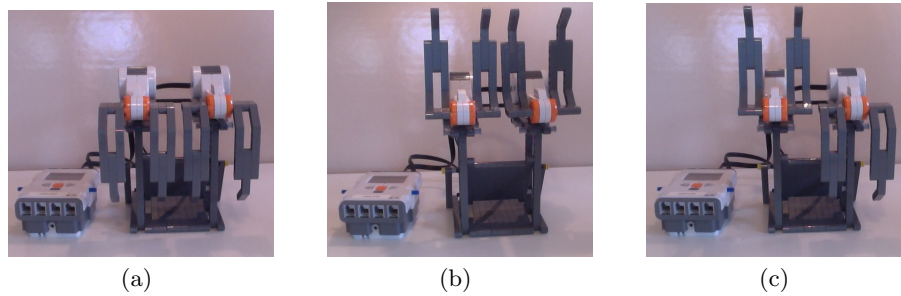
Fig. 1. Robot gestures: rock (a), paper (b), scissors (c)

2 The robotics interface

The robot used in this work is a NXT of the LEGO Mindstorms series. NXT is a low cost programmable robotics kit developed by LEGO. The core of this kit is an "intelligent" brick (Fig. 2a) where are located the CPU, a Flash memory and several I/O ports to control motors and sensors. The aim of this kit is providing a platform that can be simply used also by inexperienced users.

In our research, the central unit of the LEGO platform (NXT brick) is used to control two different motors simulating the three human gestures. The three configurations are showed in Figure 1.
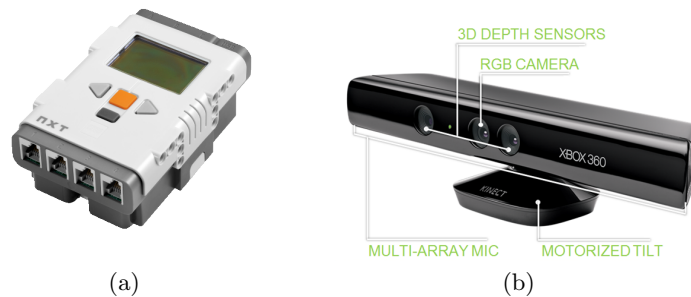
Fig. 2. The hardware used in this work: the NXT brick (a) and the Microsoft Kinect sensor (b)

For the understanding of gestures played by human users a RGB-D sensor is used. In specific the vision system is provided by the low cost Microsoft Kinect sensor (Fig. 2b). Communication and synchronization between sensor and robot are obtained using ROS (Robot Operative System). This open-source framework allows us controlling the used hardware via C++ and Python programming languages.

3 The learning system

As humans do not play random moves it is also fundamental the robot use different strategies in order to increase its probability to win a match. The AI algorithm performing the robot choice is based on Gaussian Mixture Model. Different works have already been developed [1] [2] in our lab by using GMM to solve complex problems. In this case we tested the GMM flexibility by solving a problem that sounds simply if we look at the possible options, but equally hard to solve by identifying a clear strategy even for humans being.

3.1 Gaussian Mixture Model

A GMM [3] θ is a parametric probability density function represented as a weighted sum of K Gaussian components given by the equation:

$$p(\boldsymbol{x} \mid \theta) = \sum_{k=1}^{K} \omega_k g(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k), \tag{1}$$

where $\boldsymbol{x}$ is a vector $\boldsymbol{x} = \{x_1, \ldots, x_d\}$ with size d and ω_k are the weights of the various components $g(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k)_{k=1}^{K}$. Each component is a d-variate Gaussian function:

$$g(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2} \mid \Sigma_k \mid^{1/2}} exp\left\{ -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)' \Sigma_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k) \right\} \tag{2}$$

with mean vector $\boldsymbol{\mu}_k$ and covariance matrix Σ_k. The weights ω_k satisfy the normalization property $\sum_{k=1}^{K} \omega_k = 1$.

The GMM is parametrized by mean vectors, covariance matrices and mixture weights from all the K components. These parameters are represented by the notation:

$$\boldsymbol{\theta} = \{\theta_k\}_{k=1}^{K} = \{\omega_k, \boldsymbol{\mu}_k, \Sigma_k\}_{k=1}^{K} \tag{3}$$

Extremely important is the training stage and the maximum likelihood (ML) parameters estimation. Using the Expectation-Maximization algorithm we obtained the value of the parameters which maximize the likelihood. Furthermore the estimation of the optimum K is reached using the Bayesian Information Criterion (BIC) [4]. BIC selects the better model in a class of models with different number of parameters. In this way we determined the model with the minor number of parameters that guarantees a good data fitting. For further details see [5] and [6].

3.2 Rock-Paper-Scissors

A dataset $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ with N the number of instances is used to train the GMM. GMM features $\boldsymbol{x}_j|_{j=1}^{N}$ have dimension $d = 8$ and they are defined as

$$\boldsymbol{x}_j = \{(r_{j-3}, h_{j-3}), (r_{j-2}, h_{j-2}), (r_{j-1}, h_{j-1}), (r_j, h_j)\} \tag{4}$$
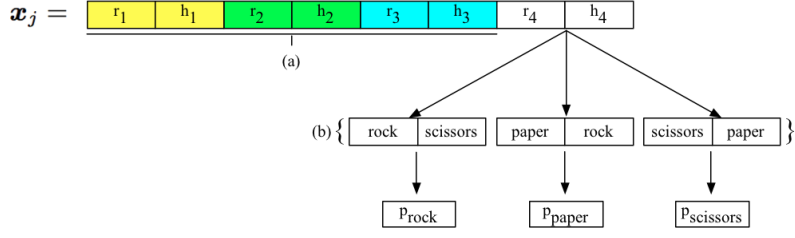
Fig. 3. (a) The three previous rounds used to determine r_4. (b) The three pairs used to complete the sequence of three previous rounds. From this sequences we obtain the probabilities of rock, paper and scissors.

where (r_j, h_j) is the jth round played, with r_j as robot move and h_j human move. Both $r_j, h_j \in \{rock, paper, scissors\}$. Each instance in the dataset consists of four rounds, the jth one and three previously played.

The idea is using a history of the three previous rounds to determine the gesture r_j the robot will play in the current game to contrast the h_j gesture played by the human. In the testing phase we complete the sequence of three previous games in Fig. 3(a) with each of the three "favorable" pairs in Fig. 3(b), in order to obtain three prior probabilities related to these configurations. The pairs are called "favorable" because they correspond to a robot victory.

Thus, after obtaining the three completed sequences called $\boldsymbol{x}_{rock}$, $\boldsymbol{x}_{paper}$ and $\boldsymbol{x}_{scissors}$ in relation to the first symbol of completion we obtain three priors from the GMM for rock (p_{rock}), paper (p_{paper}) and scissors ($p_{scissors}$):

$$p_i = P(\boldsymbol{x}_i) = \sum_{m=1}^{K} P(\theta_i = m \mid \boldsymbol{x}_i), \tag{5}$$

where θ_k denote the kth weighted Gaussian component and $i = rock, paper, scissors$.

Using these updated probabilities we choose the robot gesture generating a random number n in the interval [0,1] and then partitioning the interval [0,1] in three disjoint subintervals. The final gesture will be:

- Rock, if $n \in [0, p_{rock})$,
- Paper, if $n \in [p_{rock}, p_{rock} + p_{paper})$,
- Scissors, if $n \in [p_{rock} + p_{paper}, 1]$.

After the round is played and we have obtained both robot and human gestures we are able to complete a new sequence $\boldsymbol{x}_j$ and we could use it in order to update the training set. GMM is updated every iteration t reloading the entire training set.

BIC is executed every 2^t iterations, limiting the value of K to 50.
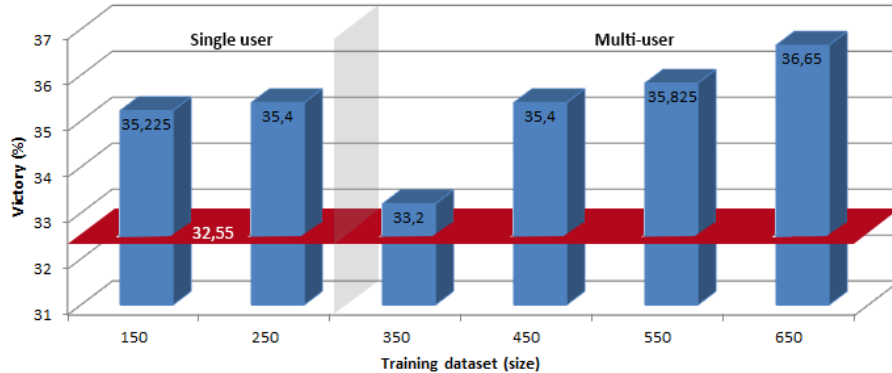
Fig. 4. The histogram shows the percentages of robot victories for training sets of different size. The red plane shows the percentage of victories obtained with the equiprobability policy in the first test.

4   Experimental results

We tested our approach comparing it with the standard random choice strategy. Tests are performed on 200 rounds and we averaged the results on 20 iterations.

Random strategy Rock, paper and scissors are equiprobable and robot gestures are chosen randomly. In this case we obtain a percentage of robot victories of 32.55% (red plane in Fig. 4).

Our GMM based strategy The strategy adopted is the one previously described in this paper. Probabilities were updated over time and the test was performed on different size training dataset: five different initial dataset were considered from 150 to 650 elements (Fig. 4). Datasets composed by 150 and 250 elements are referred to a single user, while in the ones composed by 350, 450, 550, and 650 elements we asked to several users to try out our system. Every user played in more than one session. A session was composed from ten to twenty consecutive games. In the single user cases it is easy to see a quick increment of the percentage of victories of even with relative small size datasets. With a dataset of 250 instances, the robot will win with a probability of the 35,4%. In the multi-user version more data were needed for GMM settling. When the model starts generalizing and managing an higher number of users the percentage grows to a maximum of 36.65% for the dataset of 650 instances.

Our choice to adopt a specific strategy using a GMM approach in order to predict next human move with respect using a random strategy is also supported by humans' percentages of use of the three gestures rock, paper and scissors (Fig. 5). We can notice that users do not play randomly but they adopt a precise strategy.
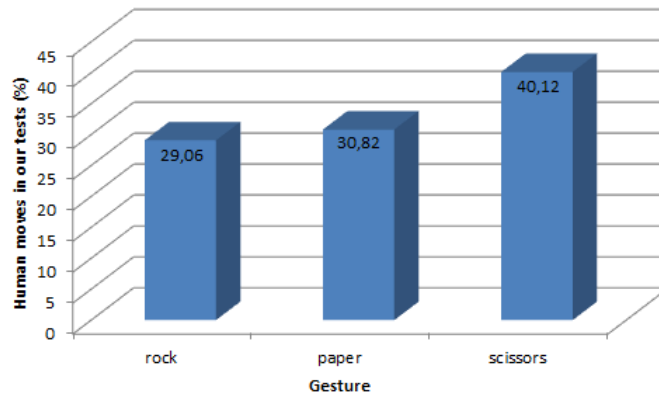
Fig. 5. The histogram shows humans' percentages of use of the three gestures.

5 Conclusions

In conclusion we have implemented an AI algorithm to allow human users playing naturally with an intelligent robot. Considering the maximum training dataset size (650 instances) the chosen approach has led to a robot winning percentage of 36.65% with respect the percentage of 32.55% obtained with a random choice strategy.

In this work we focused our attention to the victory percentages of the robot without taking care of how much human users are interested and involved in the game. In a similar way it is possible to fit with evidence and practice in game design, where the aim is to match the ability of the opponent to keep her/him involved in the game. In fact, we record both human and robot moves so we can easily choose the policy by using the history of previously played matches.

In the future we want to test the algorithm on larger datasets for a better estimation of winning percentage and implement an algorithm able to generate a specific model for every individual user. Finally an interesting improvement could be a comparison between the GMM trained strategy and a strategy choosing the gesture to play according to the human percentages of use reported in Fig. 5.

Acknowledgment

References

1. Michieletto, S., Chessa, N., Menegatti, E.: Learning how to approach industrial robot tasks from natural demonstrations. In: Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts ARSO2013, IEEE (2013)
2. Michieletto, S., Rizzi, A., Menegatti, E.: Robot learning by observing humans activities and modeling failures. In: IROS workshops: Cognitive Robotics Systems (CRS2013), IEEE (Nov 2013)
3. Reynolds, D.: Gaussian mixture models. Encyclopedia of Biometric Recognition (2008)
4. Konishi, S., Kitagawa, G.: Bayesian information criteria. In: Information Criteria and Statistical Modeling. Springer Series in Statistics. Springer New York (2008) 211–237
5. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA (2001)
6. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. 2 edn. Pearson Education (2003)