Personalized Instruction of Physical Skills with a Social Robot

Alexandru Litoiu
Department of Computer Science
Yale University
New Haven, CT 06520 USA

Brian Scassellati
Department of Computer Science
Yale University
New Haven, CT 06520 USA

Abstract

While robots have been used extensively for the purpose of teaching symbolic knowledge, using robots to teach or refine motor skills of humans, such as swinging a bat, or shooting a basketball, is underserved. Robots are uniquely well situated to observe physical movements, identify problems, prioritize which problems to address first, and to patiently communicate personalized advice to the student. We propose an architecture to coach physical skills, and focus on the second and third of these challenges - identifying problems with the movements, and prioritizing which to address first - as applied to the domain of shooting a basketball. We present a supervised learning approach to prioritize which problems to work on, and propose the design of several user studies that will determine the effectiveness of the algorithm.

Introduction

Socially assistive robots, robots whose primary purpose is to aid humans through social interaction (Feil-Seifer and Mataric 2005), are becoming increasingly capable in understanding humans, and helping them improve at emotional (Leite et al. 2012), skill-based, and symbolic tasks (Leyzberg, Spaulding, and Scassellati 2014).

It is advantageous to teach humans symbolic tasks using robots rather than virtual agents due to the increased social presence, or embodiment, that a physical robot engenders. Bainbridge et al. (2008) showed that this physical presence affords increased compliance on behalf of the human. Moreover, Leyzberg et al. (2010) have shown that the embodiment effect also leads to more rapid learning gains compared to a virtual agent.

As suited as robots are for tutoring informational tasks, they may be better suited for tutoring physical tasks that, like themselves, are situated in the real world. The advantages conferred upon robots over virtual agents for teaching physical tasks are: 1) external sensing 2) mobility to follow the human, and acquire different vantage points 3) the ability to perform live demonstrations 4) the ability to act upon the world to aid the human. Grer, Salah, and Akn (2013) have taken advantage of these characteristics by using Nao robots to help elderly patients exercise to remain fit by demonstrating gestures, and Ros, Baroni, and Demiris have shown that children were attentive to the robot coach that demonstrated their physical dancing task. Robots have been used extensively in stroke rehabilitation to provide social support (Wade et al. 2011), as well as sensorimotor support (Volpe et al. 2000). In terms of observing the human's movement, detecting problems, and giving the human specific advice on how to improve the movement, past research, such as Bobbert et al., is mostly focused on non-robot platforms.

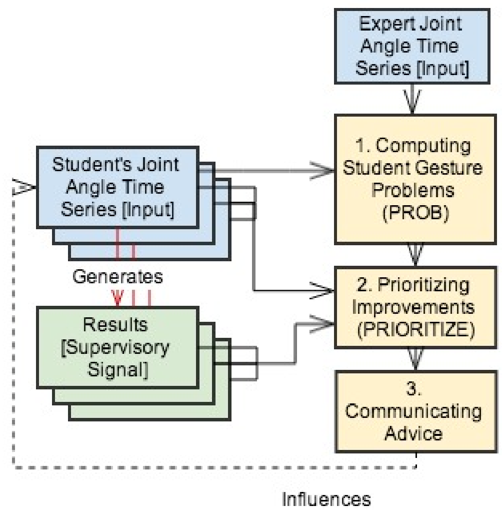


Figure 1: Robot architecture to observe and coach physical skills in humans.

We propose a high-level autonomous, personalized robot architecture to observe and coach physical activities towards a reference motion ("ground truth"), which is accepted to be correct. The architecture uses a supervisory binary signal (shown in Figure 1) which allows our architecture to determine which problems are most detrimental to the student's performance, and to coach the student to improve those problems first. This prioritization of fixing the most problematic issues, first, affords rapid improvement in short time-frames. We propose a study to evaluate the magnitude

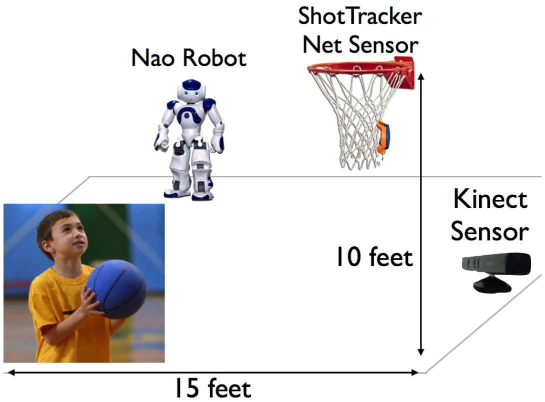


Figure 2: Physical setup for the basketball shooting application domain.

of the improvement generated by the architecture.

Application Domain

The application domain discussed in this paper is the act of shooting basketballs into a hoop from a set distance, depicted in Figure 2. Here, the input signal is the set of time series of all of the joints, collected by a Kinect 3D sensor. Another input, the ground truth, is a similar set of time series, collected by observing a professional. For each iteration, the supervisory signal is a binary variable representing whether the ball fell into the hoop or not, perceived using a commercially available ShotTracker sensor on the net. The architecture is best suited for situations in which the supervisory signal is perceivable by a robot, or is automatically present, rather than requiring manual coding by a human, but it can be used in both situations.

Proposed Architecture

The architecture receives a "Student Joint Angle Time Series" for each of the joints. This skeletal data time series is collected using a pre-built and commercially available sensor, such as a Kinect. The architecture also receives an "Expert Joint Angle Time Series" for each joint of a ground truth expert demonstration.

In Module 1 (PROB), we identify student movement problems, relative to the ground truth. This module detects differences between the user's movement, and the motion that we accept to be correct. This module first preprocesses the incoming signals by aligning their tempo with the ground truth, then bucketing the continuous signal into discrete time slices, and subtracting off the corresponding ground truth time slices. At the output of module 1, we are left with a measure of how different each joint angle is from the ground truth, for each time slice.

Module 2, the Improvement Prioritization Module (PRIORITIZE), takes in three data streams as input: i) the problems with the student's observed motion (the output of PROB), ii) the past history of all joint angle time series data of all past sessions across students, and iii) the corresponding supervisory signals for all past sessions of joint angle time series. The module's role is then to use the past history of all joint angle time series data, and corresponding supervisory signals to determine which time slices on which joints are most predictive of a successful action. In other words, it determines which problems cause bad results, and which have no effect on the results. The module then outputs the problem with the student's observed motion that is most likely to improve results, if changed. By focusing on the most significant problem, the student will be able to improve his results most rapidly.

Module 3 is the Human-Robot Interaction component of the architecture. Given the most significant problem that needs to be solved, as input, the advice module determines how best to communicate advice on how to fix this problem. Here, the robot will decide whether to give a physical demonstration, verbal advice, and also, which advice to give. This module will itself adapt to give more personalized advice over time. As the robot learns whether a particular student prefers physical versus verbal advice, the student's advice frequency, etc. this module will personalize to the needs of the student.

Computing Student Gesture Problems

The PROB module finds "problems" in the observed motion relative to the ground truth. It receives as input a set of time-series data, each for a different joint. Each time series is a collection of (timestamp, joint-angle) tuples. It finds the "problems" by first i) aligning the tempo of the signal with the ground truth signal, so that the signals can be subtracted from one another. If this step weren't performed, then if a motion were identical to the ground truth, but with an additional 2 second pause at the start, the entire signal would be offset by 2 seconds from the ground truth. Subtracting the ground truth would result in finding problems at every timestep, even though there are no problems, since the signals are identical. ii) PROB then buckets each time-series into more discrete time-intervals. Whereas the raw time-series for a joint might have thousands of data-points, we average all of the data points from t=0.0s to t=0.1s, and assign that value those to bucket 1, etc. iii) Finally, PROB subtracts the ground truth bucket 0 from input bucket 0, and ground truth bucket 1 from input bucket 1, etc. The output of PROB is the differences between the input signal and the ground truth, for each bucket in each joint. These differences then become the features for the following module, PRIORITIZE. The reason we perform bucketing was to decrease the total number of features, and to make the subtraction step easier.

Prioritizing Improvements

The prioritization module works by selecting the most detrimental problem towards performance from the student's personalized problems, inputted from PROB. PRIORITIZE therefore selects the problem that, if improved, would confer the highest amount of benefit to the student. In order to determine which problems are significant and which are not, the PRIORITIZE module performs a regression on all of the past data, from all sessions and all students.

Table 1 contains example data that we use to demonstrate an example. For each graph, the data to the left of the ver-

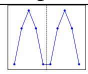
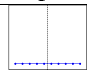
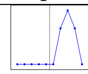
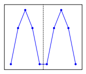
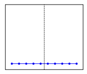
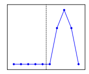
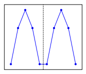
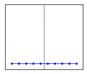
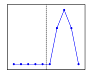
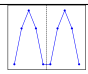
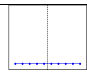
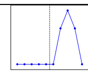
Shot #	**Student 1** [input,result]	**Student 2** [input,result]	**Student 3** [input,result]
1	 , 0	 , 1	 , 1
2	 , 0	 , 1	 , 1
3	 , 1	 , 1	 , 1
Avg	 , 0.33	 , 1	 , 1

Table 1: Example historical data for regression by prioritization module. All graphs are time series for the same joint (y-axis is angle, x-axis is time). Bucket 1 is to the left of the dotted line, and bucket 2 is to the right.

tical dotted line is the time-series content that is averaged into bucket 1. The data to the right of the dotted line corresponds to bucket 2. In this particular case, buckets containing a spike have an average of 1, and buckets not containing spikes average to 0. The features for the joint in shot #1 by student 1 are (1, 1), since the average value of both buckets is 1. Similarly, shot #1 by student 2 gives (0, 0), and shot #3 by student 3 gives (0, 1). Using this historical input data, and the historical supervisory signals (also included in Table 1, as result), we can perform a regression to determine the weights of each feature, and thus how much each problem affects the end result.

To illustrate this, if one performs a simple linear regression on the Table 1 data, one gets the weights $\theta_0 = 0, \theta_1 = -0.67$, and $\theta_2 = 0$, indicating that feature 1 is important, whereas feature 2 is not important.

Upon finding the historical weights, PRIORITIZE looks at each problem feature from the input signal sent in from PROB, and considers whether the signal is significant in affecting performance. To do this, it sees whether that feature has a high historical weighting. For example, using the historical data in Table 1, and the input features (1,1) from PROB, we know that both bucket 1 and bucket 2 for the joint in question are on average 1 unit larger from the ground truth. Which problem is affecting performance more? Historically, $\theta_1 = -0.67$ whereas θ_2 is 0. So, having a problem in bucket 1 significantly affects performance, whereas having a problem in bucket 2 does not. PRIORITIZE therefore outputs that time slice 1 for the joint in question is the most significant problem that needs to be addressed.

Proposed Evaluation

We must first study whether the PRIORITIZE module accurately identifies which problems are hindering the student's performance by collecting a corpus of data, splitting it into training, validation, and testing sets, and verifying that the regression can accurately predict whether the test set balls will fall into the net, based on their input skeletal data.

After we have determined that the PRIORITIZE module is able to understand what problem features are associated with performance hits (and we have built up a corpus of historical data), we need to validate to what degree its prioritization is in accordance with the opinion of a coach. We ask a group of 3 professional basketball coaches to code the top 3 problems with the motion of each video in the corpus. We compare the output of PRIORITIZE against the lists compiled by the coaches in order to generate statistics of how frequently coaches' number 1, number 2, and number 3 recommendations are in accordance with the output of PRIORITIZE.

When the entire system is complete, it will be possible to determine the effectiveness of the system's advice. We ask each of a group of students to shoot a number of balls, receive advice from the system, and then shoot another session of balls. We measure the improvement in their shooting accuracy, across sessions. We then compare this improvement to control conditions. In the first control condition, the robot gives no advice, but remains present. We test this condition to ensure that it is the advice, and not the robot's presence that engenders improved performance. In the second control condition the robot gives advice, but it is selected by a behind-the-scenes coach. This control condition serves as a comparison to identify how effective our system's advice is relative to the advice of a coach.

References

Bainbridge, W. A.; Hart, J.; Kim, E. S.; and Scassellati, B. 2008. The effect of presence on human-robot interaction. In *The 17th IEEE International Symposium on Robot and Human Interactive Communication*, RO-MAN 2008, 701–706. Munich, Germany: IEEE.

Bobbert, M. F.; Hoogendoorn, M.; van Soest, A. J.; Stebletsova, V.; and Treur, J. 2013. Ambient support by a personal coach for exercising and rehabilitation. In *Human Aspects in Ambient Intelligence*. Springer. 89–106.

Feil-Seifer, D., and Mataric, M. J. 2005. Defining socially assistive robotics. In *International Conference on Rehabilitation Robotics*, 465–468. IEEE.

Grer, B.; Salah, A.; and Akn, H. 2013. A robotic fitness coach for the elderly. In Augusto, J.; Wichert, R.; Collier, R.; Keyson, D.; Salah, A.; and Tan, A.-H., eds., *Ambient Intelligence*, volume 8309 of *Lecture Notes in Computer Science*. Springer International Publishing. 124–139.

Leite, I.; Castellano, G.; Pereira, A.; Martinho, C.; and Paiva, A. 2012. Long-term interactions with empathic robots: Evaluating perceived support in children. In Ge, S.; Khatib, O.; Cabibihan, J.-J.; Simmons, R.; and Williams, M.-A., eds., *Social Robotics*, volume 7621 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 298–307.

Leyzberg, D.; Spaulding, S.; Toneva, M.; and Scassellati, B. 2010. The physical presence of a robot tutor increases cognitive learning gains. In *34th Annual Conference of the Cognitive Science Society*, Cog. Sci. '12.

Leyzberg, D.; Spaulding, S.; and Scassellati, B. 2014. Personalizing robot tutors to individuals' learning differences. In *Proceedings of the 2014 ACM/IEEE International Con-*

ference on Human-robot Interaction, HRI '14, 423–430. Bielefeld, Germany: ACM.

Ros, R.; Baroni, I.; and Demiris, Y. 2014. Adaptive humanrobot interaction in sensorimotor task instruction: From human to robot dance tutors. *Robotics and Autonomous Systems*.

Volpe, B.; Krebs, H.; Hogan, N.; Edelstein, L.; Diels, C.; and Aisen, M. 2000. A novel approach to stroke rehabilitation robot-aided sensorimotor stimulation. *Neurology* 54(10):1938–1944.

Wade, E.; Parnandi, A.; Mead, R.; and Matari, M. 2011. Socially assistive robotics for guiding motor task practice. *Paladyn* 2(4):218–227.