Identifying Expertise Using Kinect Data

Cs 229 - Autumn 2015

Camila Pereira
Stanford University
camilap@stanford.edu

1. Introduction

Project based models of teaching and the use of maker spaces is becoming an important trend in modern education. But how to assess how well a student is doing on these activities? One hypothesis is that the set of gestures and time a student applies in solving tasks is a predictive of how much expertise he acquired. These kinetic and kinematic variables could suggest how long a student takes to solve a problem, how long he spends in analyses before testing a tactic or if he is trying a number of random strategies. When watching a student performing problem-solving tasks, an experienced teacher could infer the signs that his apprentice is evolving. This skill could be scaled if we were able to evaluate the performance of a student automatically, helping other teachers to identify the effectiveness of educational activities in several maker spaces. In this project, I apply machine learning to evaluate participants in a workshop organized in the Stanford's fablab, by comparing them with graduate engineering students (experts) using data from a Kinect sensor. The data consists of the coordinates from 30 body points, taken every 0.02 seconds. I used supervised learning algorithms (Logistic Regression, SVM) to evaluate the possibility of recognizing expertize using this data.

2. Related Work

Previous work used data from kinect sensors to separate patterns of gait (Begg, R. & Kamruzzaman, J., 2005, Parajuli, M et al, 2012). On these works, the authors aimed to distinguish young from old participants by their gait pattern. They reported good results using SVM, what motivated me to test these algorithms.
In their study, Begg & Kamruzzaman extracted features like range of movement (kinect features) and angles for ankles and knee (kinematic features). The former had more positive influence on the results, indicating that kinematic analysis might be a good choice of features for these kind of data. Zafrulla et al. (2011), who used data from a kinect to recognize sign language, used PCA to reduce the dimension of the dataset, though no impact on the learning efficacy was reported.
Parajuli et al.(2012) reported using KPCA, an algorithm that performs PCA using a kernel, getting better results than using the usual PCA algorithm to select features for gait recognition. Those two last studies motivated me to test PCA on my dataset.

3. Dataset and Features

The data used was extracted from a kinect sensor during the execution of an experiment at Stanford's fablab. In this experiment, 20 high schools seniors ("novices") participating in a workshop performed six assembling puzzles. Three of them were took place before and three after the workshop. Three participants dropped the workshop and their data was excluded. As a control group, 18 mechanical engineering graduate students ("experts") performed the same tasks.
The puzzles were executed in the same 3' by 6' table, and recorded for 5 minutes.

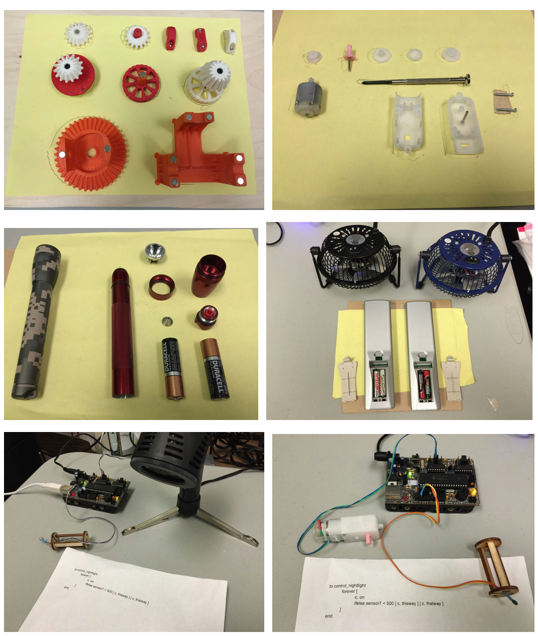

Fig.1 Materials used in the tasks

The execution of the experiments were recorded by a kinect sensor, generating the horizontal, vertical and depth coordinates for 25 joint points each 0.02 seconds.
The total number of examples was 183.
It means that there were around 10,000 x 75 features cells of data for each task

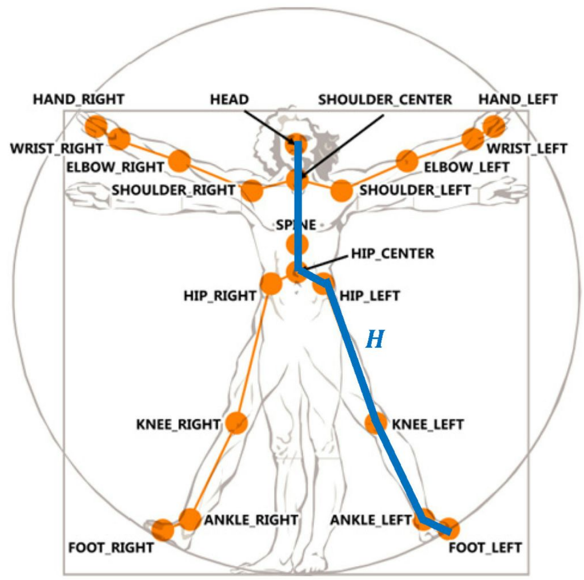

Fig.2 Illustration of the joint points recorded by the kinect sensor

(which is a training example). To get more meaningful gesture features, I excluded the points below the hips (because they were not appearing in the video) and extracted the following information from this data:

- Time to execute the task
- Task code (1-6)
- Average speed for each joint
- Minimum speed for each joint
- Maximum speed for each joint
- Range of motion for each point
- Average joint angles for hands, tips, thumbs, wrists, elbows, shoulders and spine

Besides optimizing the results, I executed Feature Selection to identify which features were more relevant to determine the level of expertise. The best results were achieved with the following nine features:

- Time to execute the task
- Average speed of the left hand
- Maximum speed of the Spine_Mid
- Maximum speed of the head
- Maximum speed of the left Wrist
- Maximum speed of the right elbow
- Range of movement of head
- Range of movement of the left thumb
- Left elbow-shoulder-spine angle

I also performed PCA, finding K = 25. The results were poor, though.

4. Methods

4.1 Leave-one-out Cross Validation

For each algorithm, I tested the accuracy on the validation set. I first splitted the data in 80% for training and 20% for testing, but due to the small size of the sample the results were very low, under 50% of accuracy.

I then chose a Leave-one-out cross validation method, training on the whole data except for one example at a time and testing on this remaining example. The accuracy would then be calculated as the average result of each test.

4.2. Model Selection

My goal was to separate experts from novices, thus I used a {0,1} label to the data. To select the best model for this classification, I applied the following classification methods: Logistic Regression, SVM with Linear, polynomial and gaussian Kernels.

Logistic Regression is a classification algorithm that only allows values between 0 and 1, using the sigmoid function:

$$g(z) = \frac{1}{1+e^{-z}} \qquad (1)$$

This classification chooses 0 for $g(z) < 0.5$ and 1 for $g(z) \geq 0.5$.

SVM was used aiming to find an optimal separating hyperplane in a higher dimensional space, transforming the data by means of a kernel function. In this project, I tested the learning with the following kernels:

Linear:

$$K(x,y) = x^T y \qquad (2)$$

Polynomial (degree 2):

$$K(x,y) = (x^T y + 1)^2 \qquad (3)$$

Gaussian kernel:

$$K(x,y) = exp(-\gamma\|x-y\|^2) \qquad (4)$$

4.2.1. Data Scaling

I my first trial with SVM, I hadn’t scaled the data, and the results were indeed very poor. I then executed data scaling before training and testing, which gave better results. As this approach did not seem very accurate for me, I then executed data scaling on the training and test sets separately. It improved the results even more.

4.2.2.Choice of C and Gamma

To calibrate the SVM algorithm, I used grid-search for determining C and Gamma for the gaussian kernel, testing values for C = 0.1, 1, 10, 100, 1000 and gamma = 0.001, 0.01, 0.1, 1, 10; then, I refined the intervals.

4.3. Forward Search for Feature Selection

Because of the small amount of data and large number of features, the dataset was prone to overfitting. Indeed, I was getting 100% train accuracy, and around 70% of validation accuracy. Also, it was interesting to determine which of the features are more indicative of expertise.

4.4. PCA

Principal Component Analysis is a statistical procedure that reduces the number of variables influencing an observation by selecting the ones with the largest variance. Although it is not indicated for feature selection and optimization, some studies used versions of PCA to extract features from kinect data. Thus, I decided it would be worth to test it. To select the number of components (k), I used the following criterium:

$$\frac{\sum_{i=i}^{k} \lambda_i}{\sum_{j=1}^{N} \lambda_j} > threshold$$

Where λ_i's are the elements of the diagonal matrix in the singular value decomposition

5. Results

The table 1 summarizes the results I obtained using the initial features I extracted. Thus, I was getting the best results with SVM using a gaussian kernel, which was giving 29.12% mistaken predictions. As I mentioned in the methods section, I used a grid-search algorithm to select C and gamma, finding the optimum values for C = 3 and gamma = 0.02.

To optimize the results, I performed a Forward Search algorithm for Feature Selection. It selected nine features, described in the “Dataset and Features” section. With the new features, I runned the model selection again, reaching the

I also tested the results for applying PCA results displayed in the table 2.

Table1. Model comparison results, without Feature Selection

Algorithm	Training Accuracy	Test Accuracy (Looc val.)
Logistic Regression	76.5%	65.03%
SVM Linear Kernel	61.05%	54.10%
SVM Polynomial Kernel	49.45%	65.38%
SVM - Gaussian Kernel	100%	70.88%

Table2. Model comparison results, after Feature Selection

Algorithm	Training Accuracy	Test Accuracy (Looc val.)
Logistic Regression	76.5%	65.03%
SVM Linear Kernel	70.68%	68.13%
SVM Polynomial Kernel	75.82%	65.38%
SVM - Gaussian Kernel	100%	80.22%
PCA + SVM	100%	55.50%

The best performance, 80.22% on test accuracy, was achieved using a gaussian kernel. The optimum values for C and gamma were C = 4 and gamma = 0.125.

using as threshold 0.90, 0.95 and 0.99, achieving the best result of 55.5% test accuracy with $k = 25$, using a threshold of 0.99.

6. Future Work

The classification results were somehow satisfactory, although the small size of the dataset, and consequent difficulty in performing test plus cross-validation instead of only the former as I did, might have influenced the results. More and different data in future work can improve the conclusions.

An important factor to advance the use of machine learning in educational applications and assessment is to identify which gestures, postures and execution factors (like speed) are related to good problem-solving. An interesting follow up to this project would be to perform the same analysis substituting the average of speed and angles, like I used, for the standard deviation. Maybe this would identify better the divergences that are indicative of solving a puzzle like a novice or an expert.

Another possible improvement is to use unsupervised learning algorithms to extract features.

7. References

Begg, R., & Kamruzzaman, J. (2005). A machine learning approach for automated recognition of movement patterns using basic, kinetic and kinematic gait data. *Journal of biomechanics*, *38*(3), 401-408.

Parajuli, M., Tran, D., Ma, W., & Sharma, D. (2012, August). Senior health monitoring using Kinect. In *Communications and Electronics (ICCE), 2012 Fourth International Conference on* (pp. 309-312). IEEE.

Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., & Presti, P. (2011, November). American sign language recognition with the kinect. In*Proceedings of the 13th international conference on multimodal interfaces*(pp. 279-286). ACM.