Personal Trainer in Your Pocket

Mohammad Amanzadeh, Constantine Roros, Sayed Hadi Hashemi

Department of Computer Science, College of Engineering

{amanzad2, roros2, hashemi3}@illinois.edu

I L L I N O I S
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Abstract

When starting a new sport or practicing a new exercise program, it is extremely important to learn basic movements the right way, otherwise a long learning period or serious injuries are not unexpected. This highlights the importance of having a professional personal trainer. The problem is continuous access to such a person is expensive and not always possible.
This problem pushes the market to provide custom-designed sport gadgets. These gadgets are attached to the human body or accessories, then collect various data from built-in sensors. After that companion software analyze this data to evaluate the performance of the user.
Despite the promising outcome, there are some limitations that slow down the vast usage of these devices. First, these gadgets are still not affordable for everyone, and second their application is usually limited to one particular sport or even one particular movement, and can not be extended to others.
One way to overcome these limitations is to use more available devices for the training purposes.
In this work we propose a method to develop a personal trainer using common sensors of smartphone or health tracker. A Hidden Markov Model (HMM) has been trained for each action and classify the input motions by comparing the likelihood of HMMs trained on each action. We would also use a function of likelihood to assess the similarity of an action to the baseline.

Goals

- ✓ Detect start and end of an activity in a recording
- ✓ Find the sport by classification of the recording
- ✓ Find the exact movement by classification of the activity
- ✓ Score the performance of the user by evaluating the similarity of the recording and the activity baseline

Figure 1: Targeted Setting
Right: Smartphone with Armband, Left: Health Tracker Wristband

Data Accusation

We have recorded 8 different activates (Table 1). For each activity 2 different setting has been used: **Armband**, and **Wristband** (Figure 1). For each setting, 5 recordings have been done to generate a total of **80** recordings.
Recordings collect a time series of various features from an "iPhone 6" sensors (Table 2).

Table 1: List of collected activities

Exercise		Basketball			Tennis		
Squat	Press	Jump Shot	One-Hand Pass	Two-Hand Pass	Forehand	Backhand	Serve

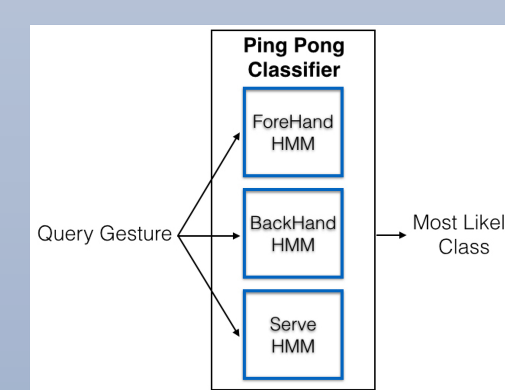
Table2: List of collected features in each recording

	Feature	Description
General	**Time**	Time at the beginning of the recording.
	Time Stamp	Time as a long integer.
	Record Time	Time since recording.
GPS	**Coordination {lat, long}**	The geographical coordinate information.
	Altitude	The altitude measured in meters.
	Speed	The instantaneous speed of the device in meters per second.
	Course	The direction in which the device is traveling.
	Vertical Accuracy	The radius of uncertainty for the location, measured in meters.
	Horizontal Accuracy	The radius of uncertainty for the location, measured in meters.
	Local Time Stamp	The time at which this location was determined.
Accelerometer	**Acceleration {X,Y,Z}**	The acceleration value for the {x,y,z} axis of the device.
Magnetometer	**Heading {X,Y,Z}**	The geomagnetic data (measured in microteslas) for the {x,y,z}-axis.
	True Heading	The heading (measured in degrees) relative to true north.
	Magnetic Heading	The heading (measured in degrees) relative to magnetic north.
	Heading Accuracy	The maximum deviation (measured in degrees) between the reported heading and the true geomagnetic heading.
Gyroscope	**Motion Rotation Rate {X,Y,Z}**	The rotation rate as measured by the device's gyroscope.
Pedometer	**Activity**	Getting the Type of Activity: stationary, walking, running, automotive, cycling, or unknown.
	Activity Confidence	The confidence that the motion data is accurate.
	Activity Start	The time at which the change in motion occurred.
	Steps	The number of steps taken by the user.
	Last Step Date	The time at which steps are counted.
Processed Data	**Rotation {X,Y,Z}**	The rotation rate for each of the three axes in radians per second
	Yaw	The yaw of the device, in radians.
	Roll	The roll of the device, in radians.
	Pitch	The pitch of the device, in radians.
	User Acceleration {X,Y,Z}	The acceleration that the user is giving to the device.
	Device Orientation	The way user holds the device

Method

We trained a different classifier for movements in each sport. Each classifier consists a group of Hidden Markov Models trained on each movement (Figure 2-Right). The features used to train the classifiers are derivative of acceleration (jerk) and rotation both in 3 axes. We don't use the accelerometer data directly because the the accelerometer output has a constant value for the gravity combined with the acceleration in 3 axes. This causes a low frequency component with high magnitude in the signal. Taking the derivative of the signal will remove this low frequency.

We cannot wait until we receive all the data for a specific movement and we should perform the recognition in real-time. In addition, we don't have any information about the beginning and end of the gestures. Therefore we use a sliding window and segment the input sequence into equal parts of size N, with N-1 overlap. For training the HMMs the windows do not have overlap to avoid redundancy in data. We use 4 hidden states for the HMMs with Left-Right state architecture. The states are only allowed to have self transition and transition to the next state. Each state is modeled with a Gaussian. Figure 3 shows the F1 score of each action in the tennis actions classifier.



	ForeHand	BackHand	Serve
F1 - Entire Length	1	1	1
F1 - Window Size 16	0.86	0.89	0.84

Figure 2: Proposed Method
Right: The classifier's architecture for the gestures in each sport
Left: F1 score for recognizing each action using the whole gesture and sliding window (16 frames)
Use 10 recordings for training and 5 for testing

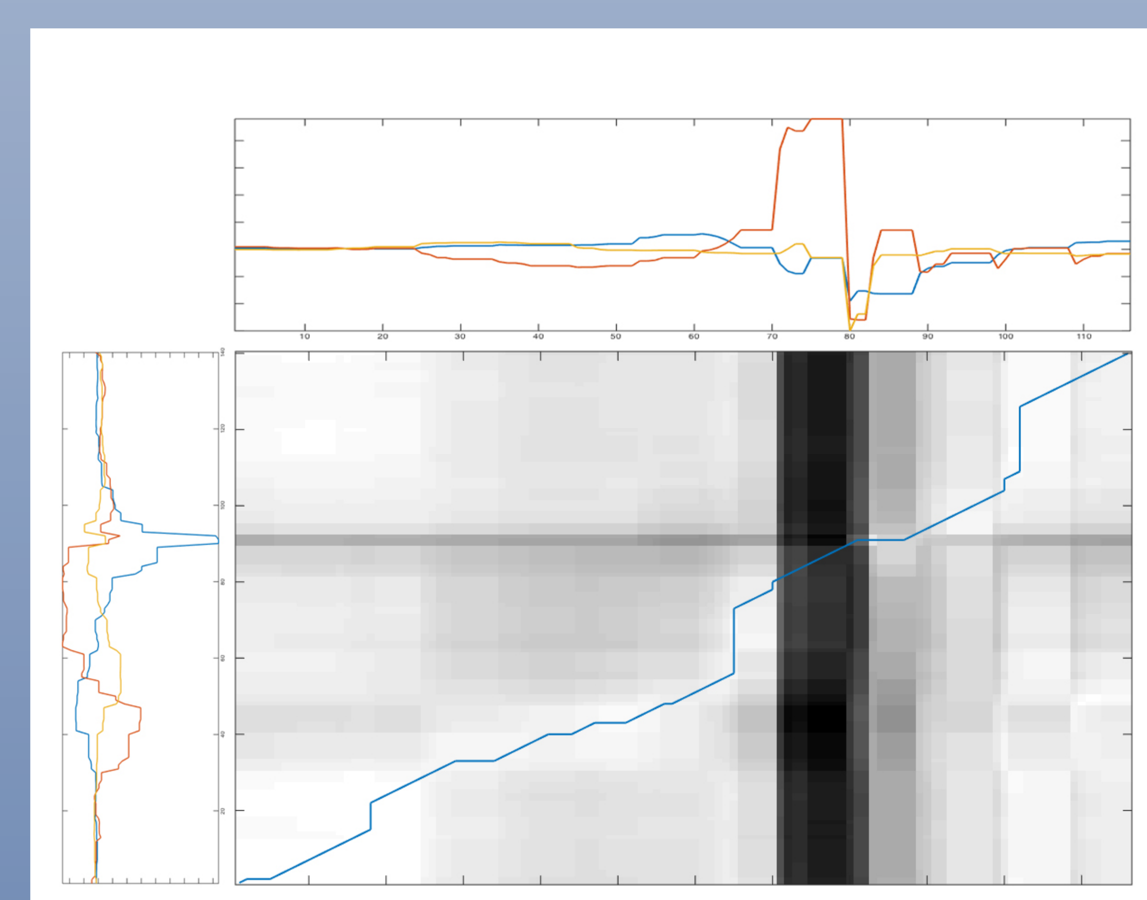


Figure 4: Distance between the X,Y,Z rotation of a good and bad serve
X-Axis: Bad Data, Y-Axis: Good Data, X: Blue, Y: Red, Z: Yellow

Results

In order to find how accurate a movement is performed we used "Dynamic Time Warping" to match the rotation of an input gesture to a good sample of that gesture. By comparing the difference of rotation signals at each time we can tell how accurate is each part of the input gesture. As we can see in figures 4, 5, and 6, the bad action has a high distance in rotation when initiation the serve. The reason is because in that sample the racket was gripped loosely and this cause a high shake at the beginning of performance.

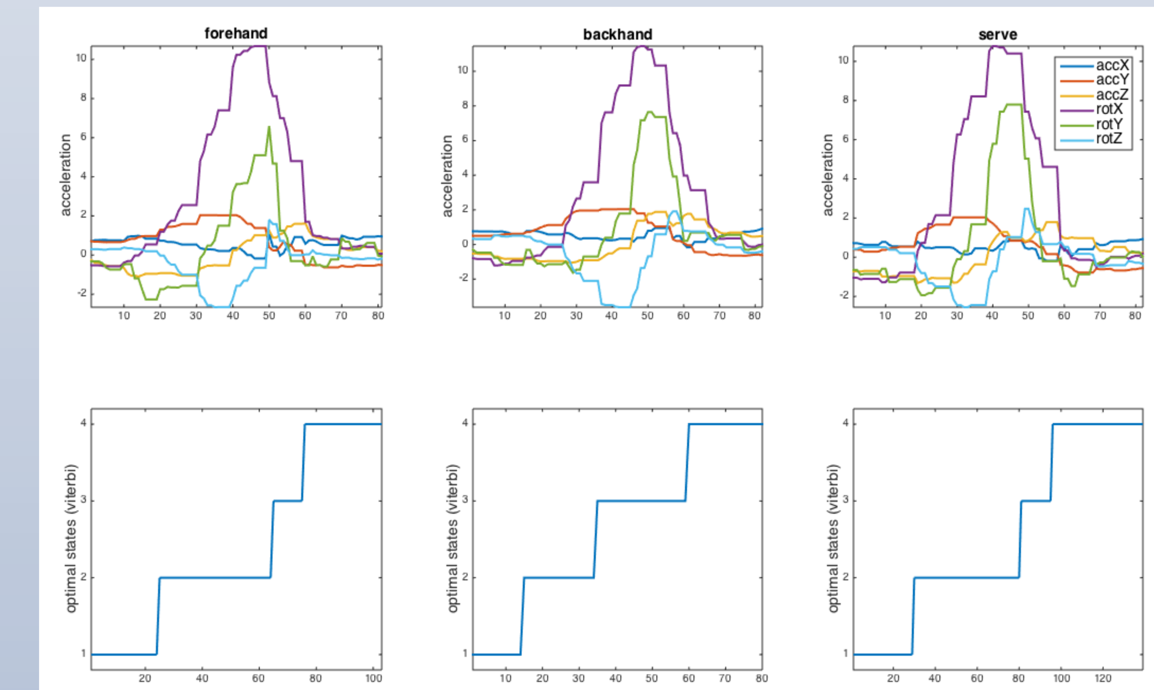


Figure 3: Most likely state sequence of a serve test action using Viterbi

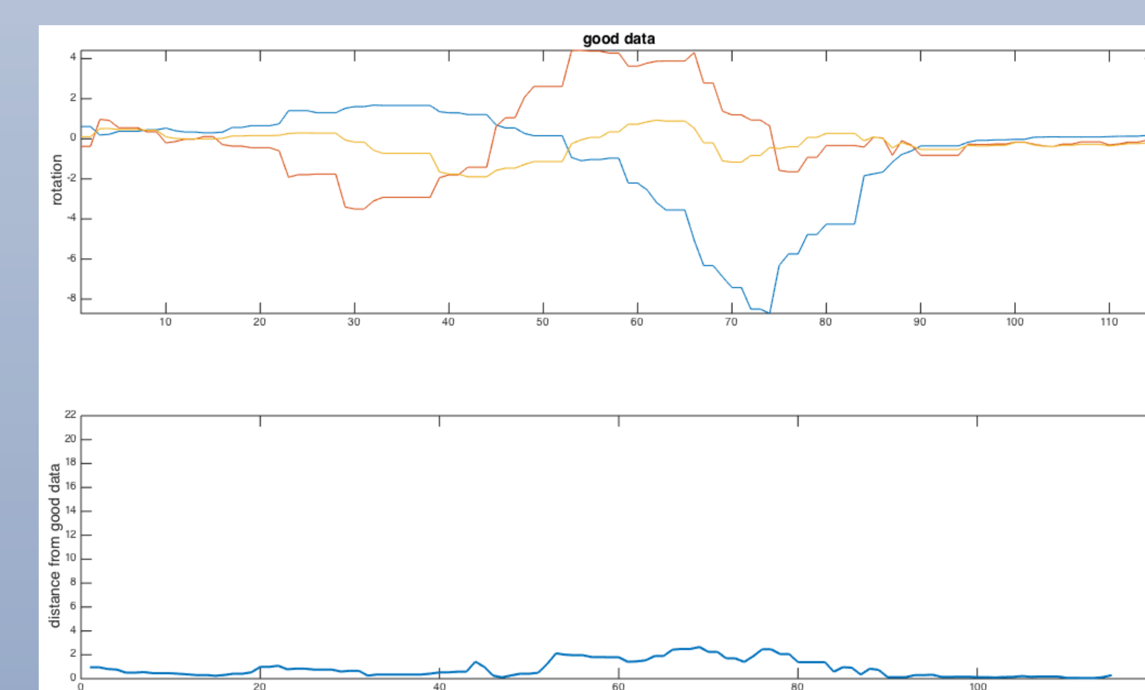


Figure 5: Distance over time for a good serve
X-Axis: Timestamp, X: Blue, Y: Red, Z: Yellow

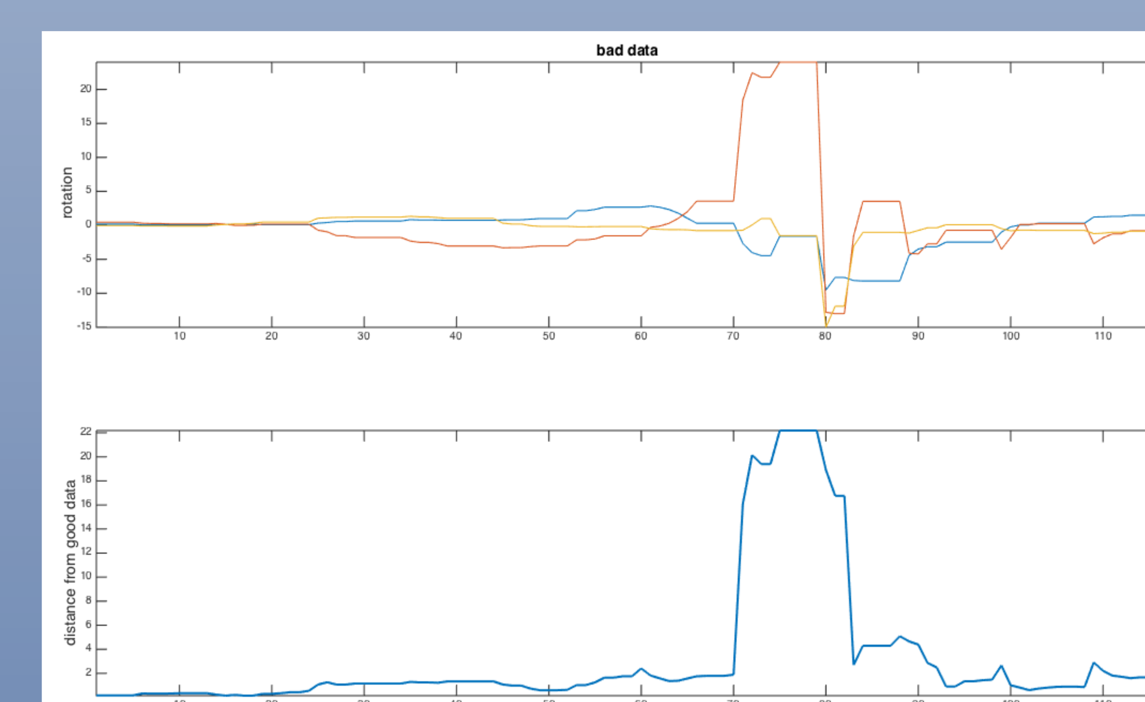


Figure 6: Distance over time for a bad serve
X-Axis: Timestamp, X: Blue, Y: Red, Z: Yellow