Semester Project
year 2014-2015

Body Movements Recognition

Author:
Yoann Trellu
Supervisor:
Dr. Markus Kalisch

December 19, 2014

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Contents

1 Introduction **2**

2 About the Data **2**

3 Defining a movement **3**

4 Features extraction **5**
4.1 Bias Variance Trade off 5
4.2 Principal Component Analysis 5
4.3 Frequency Domain 9
4.4 The pocket problem 11
4.5 Further Features 13

5 Machine Learning Algorithms **13**
5.1 k-NN 13
5.2 K-Means Clustering 13
5.3 Tree Based Methods 14

6 Counting and classifying algorithm **17**

7 Conclusion **17**

1 Introduction

Analyzing physical training sessions, we try to recognize types of movements, count the number of repetitions and extract other relevant informations in an automated way. The data comes from the accelerometer within a smartphone which was put in the pocket during the entire training session.

For this task we take a machine learning approach allowing us to present an overview of basic algorithms in our framework. In particular, we will investigate principal component analysis, random forests, k-NN as well as k-means clustering.

We first present shortly the data sets, we then investigate features extraction, and we will finally apply and review the before mentioned algorithms. We also give the complete algorithm that can be implemented for automatic movements recognition.

2 About the Data

We will work on data coming from bodyweight training routines, where a smartphone was kept in the pocket during the entire workout. The accelerometer chip inside the device outputs raw three dimensional acceleration points at non constant intervals in time. The mean interval time between two measurements was around 25 milliseconds (ms) with a standard deviation of 6 ms . It is worth mentioning a few points :

1. As the gravity is a natural force acting on the smartphone, we expect to have an average of the normed acceleration lying around 10 m/s^2.

2. The smartphone was kept in the pocket meaning that it acted as a proxy for a sensor attached to the waist. It is a good position for an acceleration sensor as it is close to the center of gravity of the human body. Thus it is well suited for bodyweight exercises but is not a good location for exercises involving machines or movements where the waist is kept still.

3. Location of the sensor can be changed (for example around the arms using a strap). The same procedure applies.

4. Exercises may be executed differently by other people, therefore the analysis has to be done relative to one's body signature and not in absolute terms.

3 Defining a movement

How should one describe a movement (or repetition - "reps" in everyday language)? We should maybe ask this question at the end of this work and ask it in a different way: Having extracted all this information, how can I best describe and differentiate movements ?

From a mechanical point of view we know that a material point can be described by a function giving the three dimensional acceleration plus the roll, pitch and yaw. By considering the body as a rigid object, we can describe its movement by describing its center of gravity (or any other points on the body that is characteristic enough for the movement). We notice that to relax the assumption of a rigid body, we need more than one sensor placed sufficiently far apart.

For the time being let us focus on the acceleration data. We will come back to the other three values later on. Let us define $f : [0, T] \to \mathbb{R}^3$ to be the acceleration function where $[0, T]$ is the period of time during which a sequence of one type of movements has been monitored.

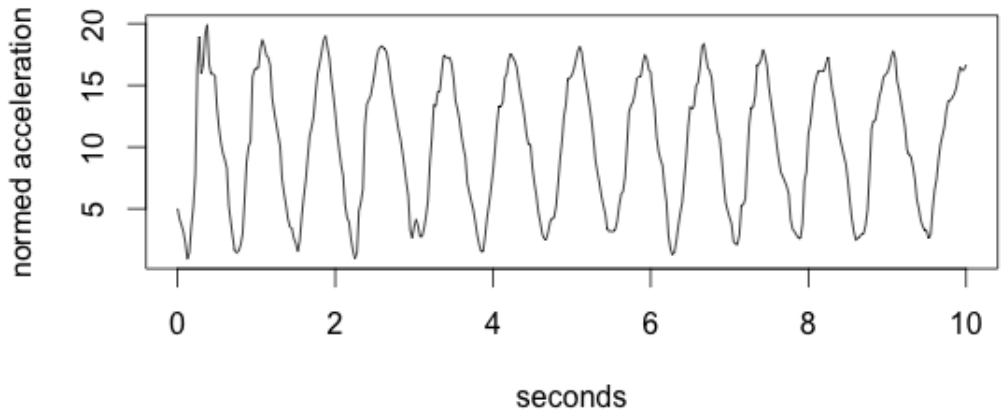

Figure 1: half of a pull up movement repeated during a 10 seconds frame.

Firstly, we observe that movements done during a training session are organized in a repetitive way with a given frequency or during a specific lap of time. Therefore we can infer a periodic character; let us give a first definition that uses the assumption of a repetitive scheme and of a perfect movement realization:

Definition (Perfect Mouvement)**.** *Having a repetitive change in the acceleration*

function f, a movement is defined to be the smallest element $f|_{[a,b]}$, that when repeated produces again the same repetitive change in acceleration.

Secondly, we think that the assumption of realizing a perfect movement sequence is too restrictive and needs to be relaxed. In particular we allow movements not to be equally spaced in time, we allow for noise, and we allow movements to *slightly* differ among themselves. However, we assume that the *signal to noise ratio* is high enough to allow for recognition. In other words, we assume the moving person has an execution that is good enough not to be shadowed by the ambient noise.

So let us first notice that every movement goes back and forth, up and down, and so on. From an acceleration point of view, the up movement needs to be triggered ($a > g$), and so does the down movement ($a < g$) - If the movement is perpendicular to gravity, only replace g by 0 in the above inequalities -.The function f should therefore contain a local maxima above gravity, and a local minima below. The first naive idea is then to define a movement by $f|_{[a,b]\cup[b,c]}$ where a and b are those stationary points and c is the next triggering point if movements are non-stop, some $f^{-1}(g)$ otherwise.

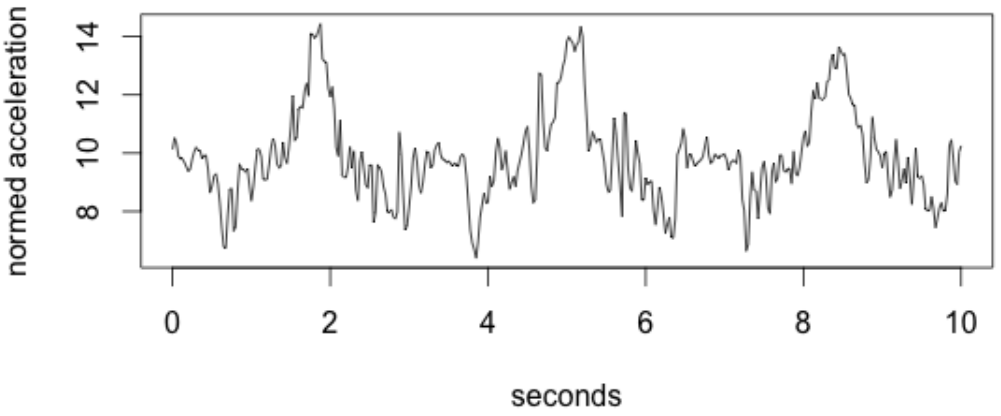

Figure 2: Three squat movements on one leg; because of instability, lots of noise is introduced.

Now we need to work on methods that will ignore stationary points created by noise. The first obvious one is to smooth the data using a moving average; the only remaining question being which smoothing parameter to use. We observe

this parameter could depend on the time required to realize the movement, as well as to which speed. We also observe that this method is not very robust to noise above some threshold.

A second method would be to list all stationary points, and determine a rule to omit them accordingly. We can assume the down movement will be realized almost at the same speed every time. Therefore we expect some determined amount of time between the local minima and the next relevant point (c).

Next, we will see how one can extract features to recognize the kind of movement. We will heavily rely on the definition given; that is we consider movements as part of a bigger set of the same movement. Thus, we first try to determine the type of exercise within a set, *and then* count the repetitions. This differ substantially from labeling every movement.

4 Features extraction

4.1 Bias Variance Trade off

In this section we want to summarize and describe the data with few variables containing the needed information to classify body movements. As we have a machine learning task, one may question the need for independence and interpretability of variables, since the more data we feed to the algorithms the better they will behave. However this is only true if we want to build an excellent classifier for data that is very specific.

If we want to design algorithms which are in some sense *robust* and general enough, we do need to consider the trade off between *bias* and *variance*. With lots of variables we reduce the bias and gain accuracy on specific data, but it will behave poorly if new incoming data differ slightly from the training set. In our framework we do need robustness since movements are - among others - sensitive to who is doing them, how tired they are and also the exercise difficulty.

As we first try to label sets, we cut our timeserie into a sequence of overlapping windows having the same time span. The span should be linked to the minimum time spent for the execution of a set, whereas the iteration parameter will be a trade off between the computational cost and the precision of the algorithm.

4.2 Principal Component Analysis

Using principal component analysis we want to describe the different forms we have seen in Figure 3. Let us first recall how this method works, its strengths and its weaknesses.

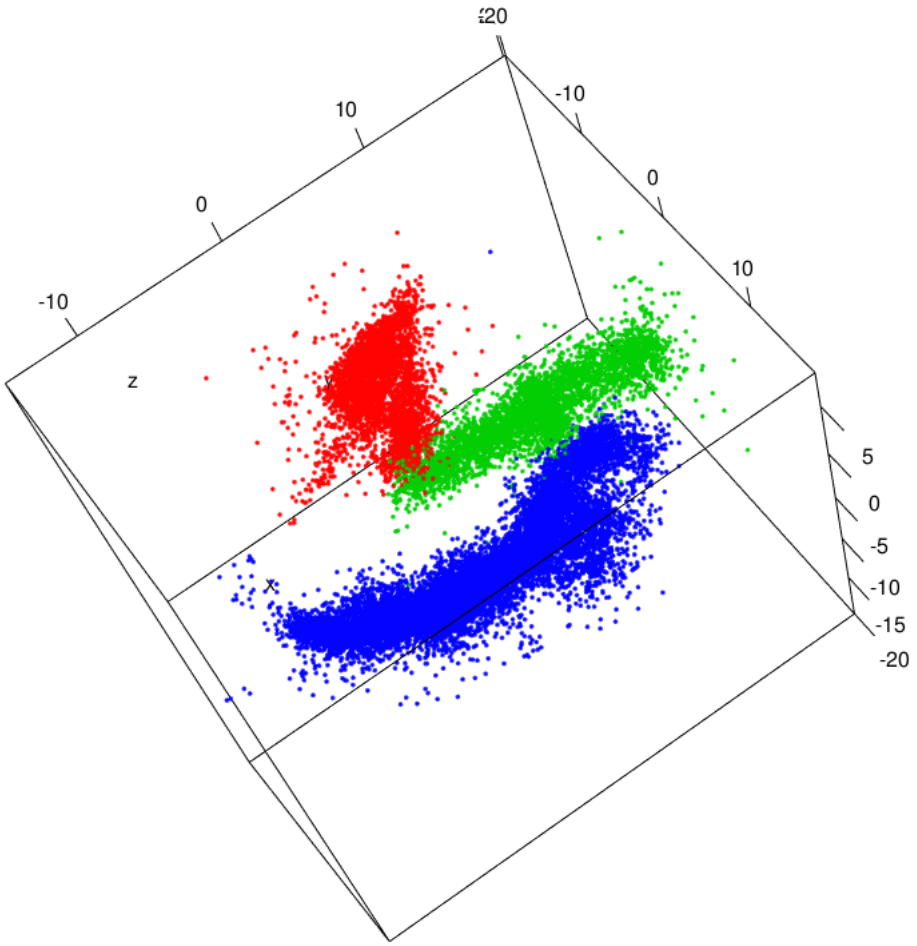

Figure 3: acceleration values during dips (in red), pull-ups (in green) and squats (in blue).

We state the problem as: how to represent data using low dimensional linear surfaces, to *best represent* the data. Let us assume that we have a data set in the form of an $n \times p$ matrix X having n observations described in p dimensions. Further, let us write the following maximization problem :

$$\begin{aligned} &\text{Maximize variance of } Z_1 \text{ where} \\ Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p \\ &\text{subject to } \sum \phi_{i1}^2 = 1. \end{aligned}$$

This essentially computes the normalized vector indicating the direction varying the most. That is, once we project all points along one vector (hence on a one dimensional space), this one will generate the biggest variance. Let us also notice that the vector is given by $\phi_1 = (\phi_{11}, \cdots, \phi_{p1})$.

Next, we solve for $j \in \{2, \cdots, p\}$ the similar problem :

$$\begin{aligned} &\text{Maximize variance of } Z_j \text{ where} \\ Z_j = \phi_{1j}X_1 + \phi_{2j}X_2 + \cdots + \phi_{pj}X_p \\ &\text{subject to } \sum \phi_{i1}^2 = 1 \\ &\text{and to } \phi_j \perp Span\{\phi_1, \cdots, \phi_{j-1}\}. \end{aligned}$$

We introduce the additional constraint that all principal directions are orthogonal between them. Typically, to solve this numerical problem we frame it as a linear programming one and then use singular value decomposition.

The strength of this approach is its understandability as well as its tractability. Linear representations are well understood. However, we are not allowing non linear representation of the data. As seen in Figure 3, it could have been useful to describe the squats more precisely (with a manifold?).

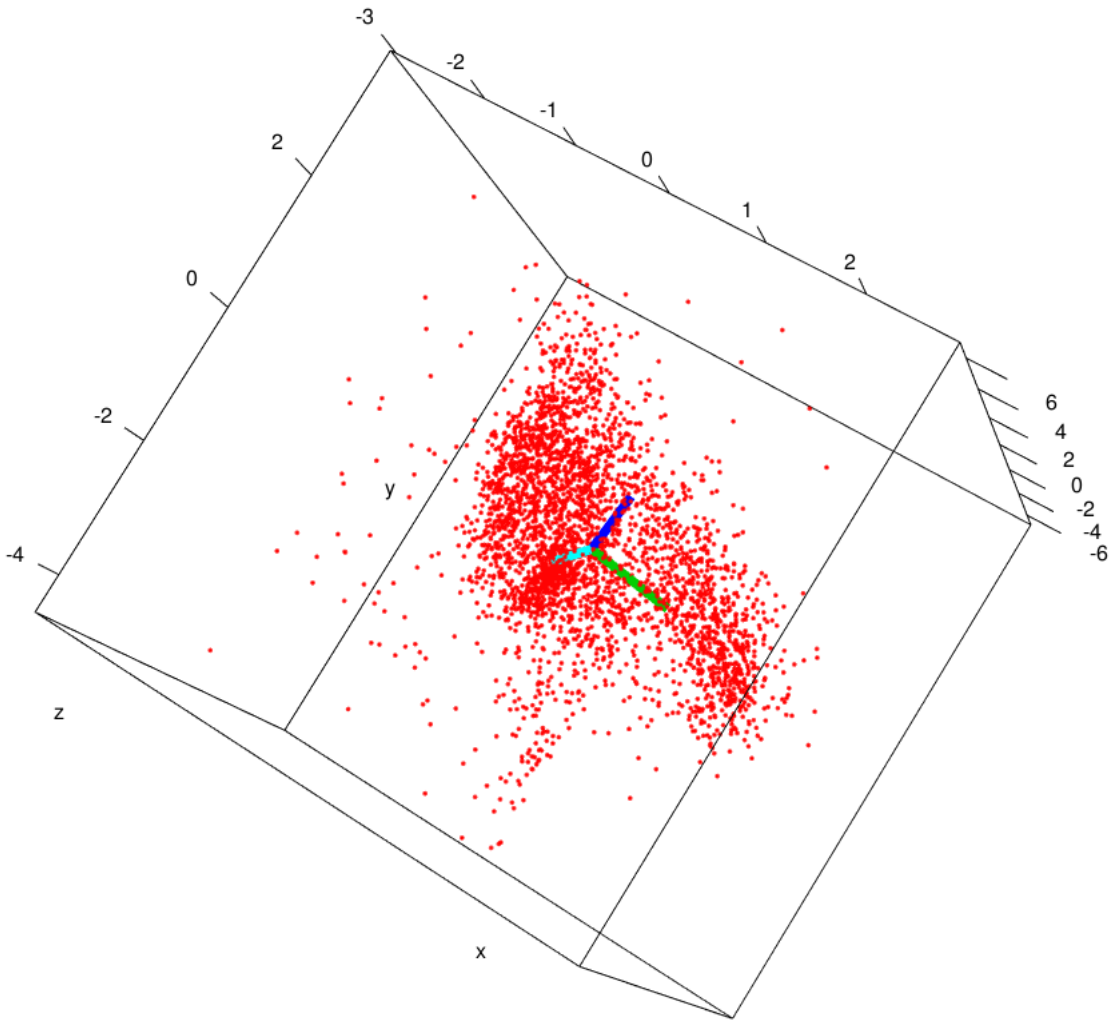

Figure 4: Red points are acceleration during dips movement, in green the first principal vector, in dark blue the second one and in light blue the third one.

4.3 Frequency Domain

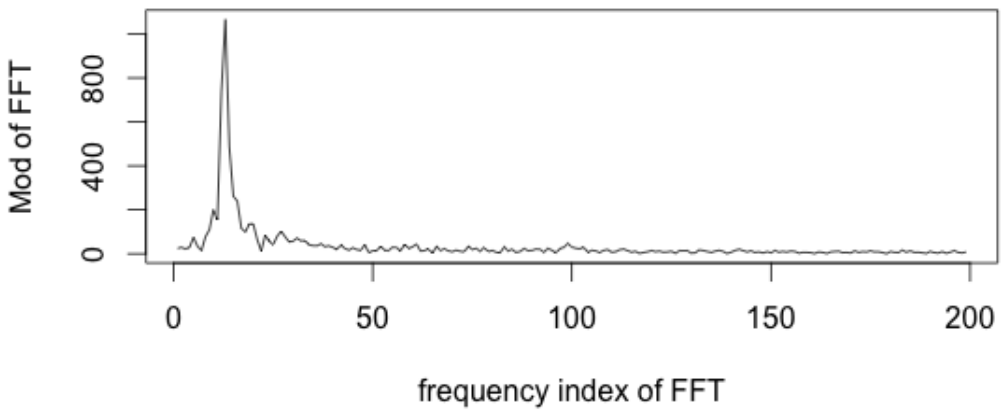

Figure 5: Plot of the FFT on the pull-up movement from Figure 1 (without mean frequency).

We now switch to the frequency domain using the Fast Fourier Transform (FFT). As we are in a discrete setting, frequencies can also be thought as "bins" in the following discussion. Again the aim is to find features to help us differentiate movement sets.

Looking at Figure 5, we notice a high peak indicating the frequency at which pull-ups have been made. Also from Figure 6 we notice two peaks instead of one, with added noise. Finally from Figure 7 we do not see any clear pattern.

Now to extract features from this plots, we use different values. The first one making most sense from looking at Figure 5, is the frequency containing the most energy. But looking at Figure 6, it does not make much sense anymore as we now have two frequencies of similar energy. Also to detect inactivity, it may be interesting to compute the *spread* of energy among frequencies. Let us define some quantities that will help us :

Definition (Spectral Centroid)**.** *The Spectral Centroid is the weighted mean of the frequencies, with weights being the energy magnitudes.*

The above quantity can be easily related to the mean of a random variable, while the spectral spread finds its analogy as the variance of the same random variable.

Definition (Spectral Spread)**.** *Squaring the distance between frequencies and the centroid, the Spectral Spread is the weighted mean of those distances, using energy magnitudes as the weights.*

Finally we also give a feature that is suitable to detect inactivity (or the lack of periodic movement) which is the proportion - say 95% - of frequencies (or bins) that contributes to the energy total.

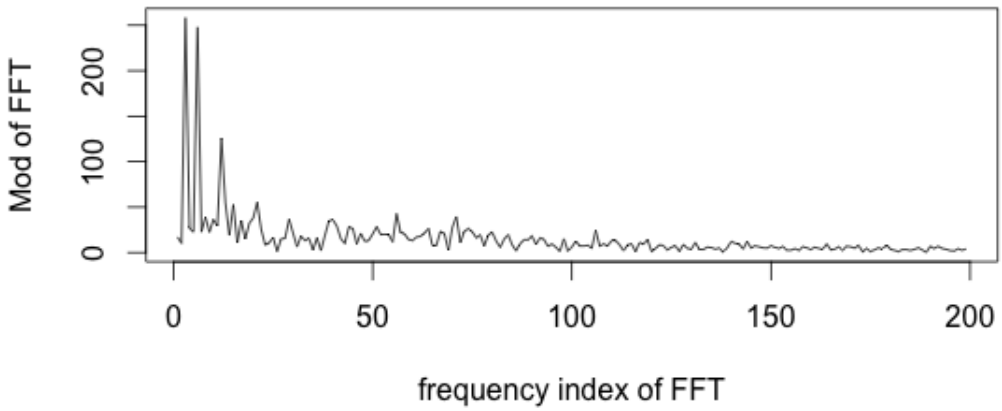

Figure 6: Plot of the FFT on the squat movement from Figure 2 (without mean frequency).

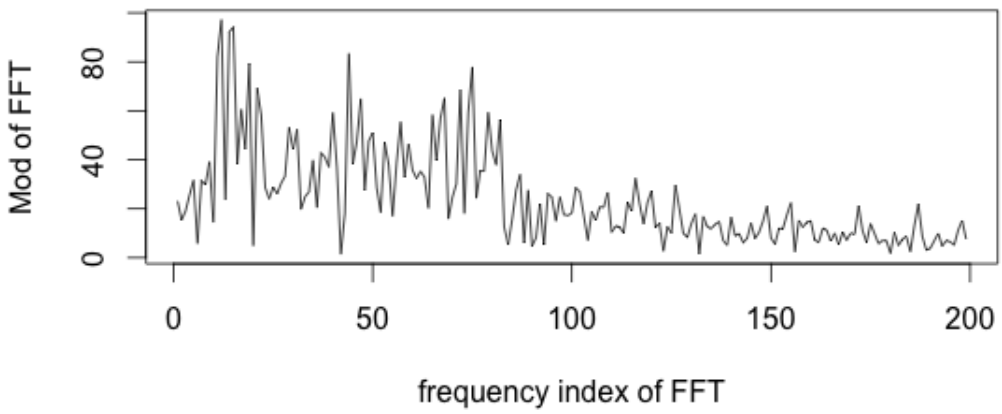

Figure 7: Plot of the FFT on a 10 seconds window taken while resting (without mean frequency).

4.4 The pocket problem

We now want to show that having the phone in the pocket instead of having it fixed around the waist has a big relaxation effect on the problem. We take the following coordinate system : putting the phone (as if we want to read something) on the palm of the right hand, we use coordinates created by the right hand rule. As an example, the z-axis (middle finger) will be perpendicular to the screen. In addition we also consider the coordinate system in the room say, where the zero is in a corner with axis as the edges. We denote this coordinate system by capital letters (X, Y, Z),

First let us assume the phone is fixed vertically on the side of the waist. Since it is fixed, the position of the (rigid) body and of the phone will always be the same. Thus using the first principal component we can infer in which direction the body moves. For example, jumping will be along the y-axis when lateral steps will be along the z-axis and push ups will be along the x-axis.

Yet from a movement direction point of view, push ups and a step forward are similar using this information. But with respect to (X, Y, Z) we can differentiate them. That is why we also need to introduce the body position by looking at how the gravity affects the sensor. For example, once in a push up position, the gravity vector will more or less be aligned with the y-axis. But a step forward implies we are standing up hence gravity acts on the x-axis.

Doing the analysis with the gravity field gives us 2 degrees of freedom. One along gravity and one along the perpendicular subspace. To see why it is not three, we can switch to a spherical coordinate system and see that rotations along one axis have no effect on gravity (the axis which is collinear with gravity).

For a fixed phone, we describe its position using two rotation angles, relative to a standing position. θ along Z and φ along x. Notice that we use both coordinate systems. We write them as:

$$\begin{aligned} x &= \cos\theta \\ y &= \sin\theta\sin\varphi \\ z &= \cos\varphi \end{aligned}$$

To understand why, we first notice that the first rotation θ preserves z since while standing up, Z and z are aligned. Then, since we rotate around x, this axis is preserved while others are rotated by φ.

It is important to notice that the two rotations are not commutative and therefore have to be understood in a given order. We assume that the first is θ (for example lying down) and then φ (and then holding a planck on the side, where $\varphi = \frac{\pi}{2}$). Computing these angles only requires trigonometric inverses and a particular care when defining the variables (such as introducing a modulo 2π, and only considering φ when θ is significantly different than 0).

Therefore, fixing the phone on the waist allows one to retrieve two position angles as information. How about leaving the phone in the pocket?

We first approximate the leg by a cylinder and assume that the phone can freely move around the cylinder, as long as the normal is aligned with the normal vector of the cylinder. In other words, the phone can slide, rotate, but its faces must point towards the leg. We also assume that the phone can be taken out of the pocket, put back in but in other directions. That is, the screen can point outward at first and then point inward once back in the pocket. From an extensive knowledge in pocket theory, it is reasonable to represent it as a quarter of a cylinder once pants are put on.

Using the above angular representation, we see why the pocket allows too much freedom to get the position of the body. Firstly, the rotation around the Z axis can be a rotation around z (phone on the right side of the pocket), around x (front of the pocket, lying on the side) or y (front of the pocket, standing).

Secondly, we also get the same problem for the second rotation as the phone can move around the leg, that is it can rotate along x if the phone was fixed as described before.

4.5 Further Features

Often the accelerometer chip will be combined with a gyroscope, a device that computes 3-D angular movements. Using engineering terms the dimensions are called roll, pitch and yaw. Even if this side has not been investigated, the above simple analysis can be reproduced in the same manner in order to detect the principal rotation axis and rotation force, as well as hopefully the same spectrum.

5 Machine Learning Algorithms

5.1 k-NN

k nearest neighbors may be the simplest supervised algorithm. Plotting points in a d dimensional space, with d being the number of features, we assign labels on points based on the majority of its k nearest neighbors labels. If k is even, ties are usually broken randomly. During our investigations, we experienced best results using a k around 10.

It is worth mentioning the problem of what distance to choose. In our case, we are dealing with real positive numbers having different units. One option is to scale every features using their standard deviations; another is to normalize them to one and then artificially inflate features that bring the most information. Using these scaling techniques we can use the euclidean distance in a tractable way.

Compared to tree based methods this approach lead to inferior results (compared by the number of mislabeling). To be noted, within a bayesian approach, we could imagine labeling the next training session relative to the previous centroids.

5.2 K-Means Clustering

Belonging to unsupervised methods, k-means clustering still needs informations on the labels, namely the number of them. Based on this number k, the algorithm will then define k disjoint sets where the union will contain all points. Now the rule to define the sets is the following :

$$\min_{C_1,\cdots,C_k} \left\{ \sum_{i=1}^{k} W(C_i) \right\} \tag{1}$$

where W is a dissimilarity measure. In other words we want to minimize the intra variation within each cluster. The trivial example for this measure is the sum of all pairwise squared euclidean distances. Using this methods with $k = 2$ and a suitable choice of features, it gave a rather good separation of activity windows and inactivity periods. We will see next a different method that aims at the same result.

5.3 Tree Based Methods

Let us begin with hierarchical clustering. We can divide this method as follow :

Agglomerative Clustering we define each cluster to be a point. At each step, the goal will be to merge two clusters, and this will be done using a distance measure. Running the steps and taking a look back we see that the leaves formed by the points are grouping into branches which are then again grouped into bigger branches, forming a tree.

Divisive Clustering Here, we first have on set containing all points. The steps will be to divide sets according to some rule. The structure will be the same as in the bottom up approach described above, but from a top down view.

We notice that the merging operation for agglomerative clustering appears *monotonic* has it is described. That is the similarity decreases monotonically from iteration to iteration. Indeed small clusters should be more intrinsically similar than larger ones. Yet, with some choices of distances (such as centroid based clustering), it is theoretically possible that the clustering will be non monotone at some steps.

Advantages of this method is its high tractability, nice representation of clusters and a good mean to pick the suitable number of clusters.

5.4 Results

To produce Figure 8, we ran the function hclust in R [1]. We specified the use of the centroid method as well as the euclidean distance measure. To be noted, to use this measure one has to normalize the data in a suitable way. Finally, we cut the tree at a height giving us 4 clusters. Further clusters do not lead to interesting clusters and have not been further investigated.

The labels displayed have only been added afterwards. They describe best what lies in their cluster. There are still some misclassifications among those clusters. We believe most of it can be removed for example using a median filter.

The cluster *moving legs* contains the squats movements as well as "leg raising, hanging to a bar". *Controlled up & down movements* contains all dips, push ups, pull ups and a few other movements were the motion is controlled and stable.

Next, we present results using the *randomForest* method that also uses trees to represent data. It has further improvement that makes it competitive to other classification algorithms.

Table 1, displays the confusion matrix from the output of the randomForest R function. Each number from 1 to 13 with 10 omitted, label a type of exercise. The

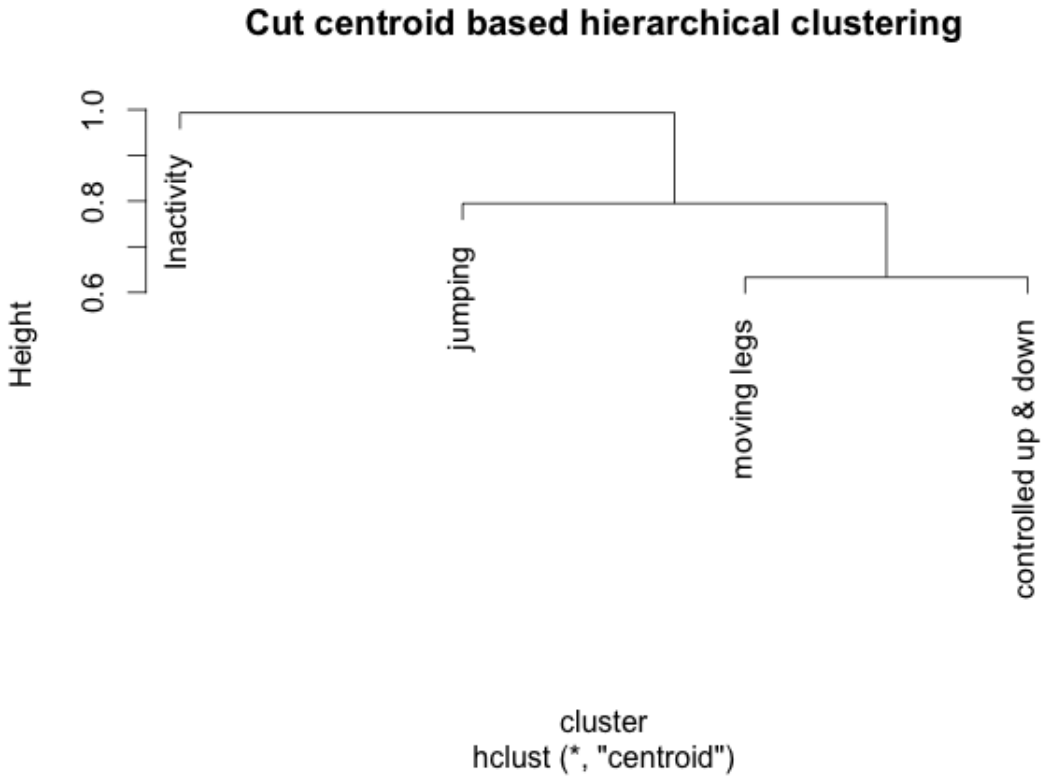

Figure 8: Restricted plot from a bottom up clustering using the centroid method.

rest label denotes inactivity. To name some exercises, we have the "dips" being 1, "pull-ups" being 4, or "one legged squats" being 5 & 6, left leg and then right leg.

We first observe a poor classification between 2 and 3. These where two kinds of "push-ups", where only the foot elevation has been changed. The next observation is the misclassification between all kinds of exercises with respect to "rest". We explain it with two things :

1. The "true" labeling has been done looking at the time series and labeling time periods according to our beliefs. It is almost sure that this process has included mistakes between rest and active phase. Indeed, we knew the order of the exercises but we did not knew the precise ending time of each set.

2. some frames will include both activity and inactivity, for example when a set ends.

	1	2	3	4	5	6	7	8	9	rest	11	12	13	cl.error
1	66	1	0	0	0	0	0	0	0	5	0	0	0	0.08
2	1	86	13	2	0	2	0	1	0	15	0	1	0	0.29
3	0	14	80	0	0	3	0	0	0	11	0	0	0	0.26
4	0	0	0	47	0	0	0	0	0	9	0	0	0	0.16
5	0	1	0	0	607	0	0	11	3	33	0	0	0	0.07
6	0	0	0	0	0	66	0	0	0	3	0	0	0	0.04
7	0	0	0	0	1	0	134	0	0	50	0	0	0	0.28
8	0	0	0	0	12	0	0	55	0	14	0	0	0	0.32
9	0	0	0	0	0	0	0	0	99	80	0	0	0	0.45
rest	1	5	7	6	26	1	25	3	6	2699	5	0	2	0.03
11	0	0	0	0	0	0	0	0	0	7	109	0	0	0.06
12	0	1	0	0	0	0	0	0	0	7	0	17	0	0.32
13	0	0	0	0	2	0	0	0	0	1	0	0	8	0.27

Table 1: Confusion table for randomForest function - R output.

6 Counting and classifying algorithm

We can finally write down all steps of the procedure :

Input Acceleration values from the entire training session.

Steps
1. Cut time series into overlapping x second frames.
2. **For each frame**, extract features using PCA and Spectral analysis.
3. Using 2-Means, differentiate between Activity and Inactivity frames.
4. Using a median filter, determine beginning and ending points of every exercise sets.
5. **For each set**, choose closest centroid in the feature space. Then use counting algorithm.

Output Enumeration of exercise types and repetition count for every type.

If mistakes Update (with Bayesian approach) the centroids coordinates.

7 Conclusion

We have first seen how to define a movement using information provided by a set, this allowed us to focusing on recognizing set types (more data) instead of movement types (fewer data). Indeed, recognizing a movement looks theoretically possible, as long as the sensor is fixed to the body.

Using 2-means and hierarchical clustering we have been able to differentiate with a good rate between activity and inactivity, important step to further characterize movements.

For exercises classification, errors are made especially between kinds of push ups whereas intrinsically different exercises were well differentiated.

Finally the counting algorithm offers a tractable way to count repetitions using down movements.

References

[1] R Development Core Team (2005). R: A language and environment for statistical computing, reference index version 2.x.x. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

[2] G. James, D. Witten, T. Hastie, R. Tibshirani (2013). Introduction to Statistical Learning, New York. ISBN 978-1-4614-7137-0