So you want to build a ML system

Jan Macháček

March 1, 2018

Abstract

Amongst other things, a machine learning system should be built! It can solve anything, there are so many wonderful frameworks, so many conference talks that show just how easy it is to throw together just a few line of Python, and hey presto!–a machine that can recognise digits in images, recognise hot dog and *not* hot dog... it even runs on a mobile phone. This essay is about the difficulties that are lurking in the execution and running of a robust & mature machine learning system.

1 What can ML do?

The exercise analysis system sets out to be a computerized personal for resistance or strength exercise. Exercise systems collect heart rate data as the baseline indication of physical exertion. Depending on the chosen sport, the systems typically collect GPS and accelerometer data. Geolocation, acceleration, and heart rate are fairly comprehensive source of data for outdoors sports: think running or cycling. For cycling, geolocation sampled at, say, 50 Hz reliably establishes the route ridden, but is also the source of data for speed and acceleration; with underlying elevation data & the weight of the rider and the bicycle, it is possible to estimate power output. With the heart rate, it is possible to get a fairly accurate view of the activity in Figure 1.

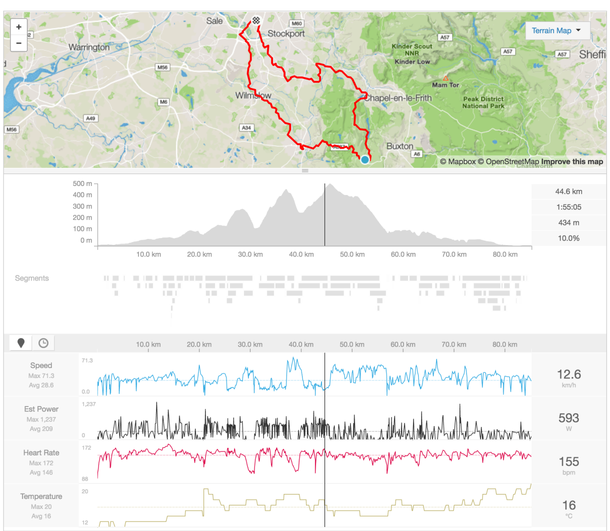

Figure 1: An afternoon ride

In addition to the post-ride analysis, the sensors provide immediate feedback during the ride. A modern cycling computer records and displays speed, elevation, distance, heart rate, pedalling cadence, and power. An athlete can use these values to inform his or her training; for example, one might train for a time trial aiming for a particular power output over a specific distance. (See Figure 2.)

Figure 2: Cycling power training

Everything becomes much more difficult indoors; and then even more difficult when not in a swimming pool, on a rowing machine, or on a spinning bike; when the user turns to resistance exercise. Some exercise machines now come with Bluetooth "beacons" that advertise the details of the activity. Any sensors are less likely to be embedded in free weights (and even if there were sensors, the sensors cannot reveal the exercise the user is performing; though identification such as 20 kg dumbbell is useful, and acceleration & rotation is even more useful). There are no machines involved at all in body-weight exercises.

To build a system for resistance exercise analysis that "matches" the features of a cycling system *without requiring super high-tech gym*, it will be necessary for the users to *bring their own sensors*. A smartwatch can provide acceleration, rotation, and heart rate[1]; these three sensor readings can be fused with sensor readings from a smartphone that the user might wear in his or her pocket. This gives additional acceleration and rotation. Unfortunately, heart rate lags behind any exertion, so it'll only increase by the time the user is done with a set of a particular exercise (assuming the exercise is done with appropriate intensity); the last easily accessible sensor, geolocation (if at all available due to the location of the exercise floor) will show that the user is not moving much. This leaves only the acceleration and rotation values as useful indications of the movement being performed.

A proof-of-concept application that allows the users to wear a sensor on their wrists and use their mobile phones to *mark the start and end of each exercise*, together with *the name of the exercise* provides the data to build & verify the first models.

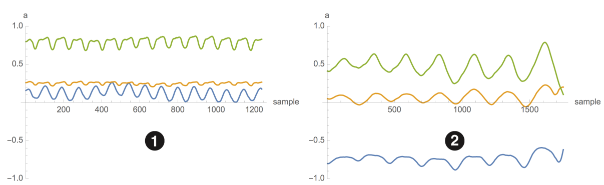

Figure 3: Two exercises

Figure 3 shows the acceleration from a watch worn on the left wrist. The acceleration is along the x, y, and z axes (with the Earth contributing $9.8\,\mathrm{ms}^{-2}$) sampled at 50 Hz; exercise 1 is the straight bar biceps curl, and 2 is the chest dumbbell flyes. You–the readers, humans–can work out the number of repetitions and you can clearly spot that these are two different exercises, and if you saw enough of these charts, you would be able to learn to identify the two exercises. It should be possible to build a machine function to do the same.

1.1 Sensors are not enough

Suppose there is indeed a function that takes the sensor readings and returns the details of the exercise being performed: *classify (sensor data)* $\rightarrow$ *exercise* . Even if this function performs perfectly (i.e. it identifies the movement that corresponds to the exercise for every *sensor data* that matches the exercise, even if the exact *sensor data* was not in the training set), it only allows to build a system that lags behind the users' movement. Even if the system could read the data from the sensors with zero delay, the system would only be able to tell the user what he or she is doing after "seeing" some minimum amount of data, introducing lag. High lag will cause the user experience to be rather frustrating. In our testing, a 100 ms lag is noticeable; 250 ms lag is still tolerable; anything over 750 ms is downright confusing. A good example is the number of repetitions: with an exercise whose single repetition takes around 1 s, a 750 ms lag will mean that the system shows 5$^{\text{th}}$ repetition when in fact the user is starting on his or her 7$^{\text{th}}$. Compare this to the sensor readings that the cycling computer reports: speed, altitude and similar can be reported as they are measured (typically a few times per second); cadence and power readings usually lag behind the instantaneous measurements because the computers display average values over a short time window. Just like the repetitions in resistance exercise, the lag is confusing; though if the athlete holds fairly steady power output, the reading usually matches the percieved effort.

Following a user that maintains a log of exercises shows

[1] Though most smartwatches report "beats per minute", the watch measures changes of the blood flow velocity, which is directly correlated to heart rate. Nevertheless, the watch does not measure heart rate itself.

References

[1] Alfie Kohn. *Why Incentive Plans Cannot Work*. `https://hbr.org/1993/09/why-incentive-plans-cannot-work`.