Analyzing Runners' Profiles Using Machine Learning Techniques Final Report

Toon Nolten, Joren Verspeurt
KULeuven
r0258654,r0258417

I. Introduction

In this report we present our results for the assignment on analysis of accelerometer data from runners using machine learning methods. The following methods were used:

- Support Vector Machines
- Decision Trees
- Naive Bayes

For these methods the following features were extracted from the given data:

- Total average and standard deviation of acceleration
- Average and standard deviation of acceleration peaks
- Index and value of highest peak in frequency domain
- Frequency and value of highest peak in power spectral density
- Ratio between average (and standard deviation) of acceleration in X, Y and Z direction and total acceleration
- Ratio between average (and standard deviation) of peak acceleration in X, Y and Z direction and total acceleration
- Average and standard deviation of time between steps (This is the only feature that is not calculated separately for every axis.)

II. Research Questions

We will try to answer the following questions:

- How do the chosen methods perform when used to classify runner acceleration time series into the following classes:
  - Trained or untrained
  - Kind of surface: asphalt, track or woodchip
- Which summarizing features of the data give the best results when classifying?

The performance of the classification methods is examined in different ways:

- What is the accuracy of the classification (measured in correct classifications per number of classified datasets)?
- How fast would the method classify running data when receiving live input?

III. Experiments

For each of the learners the implementation in the Python module *scikit-learn* was used.

For all of the experiments the given data was trimmed to keep only the useful parts. For each run the middle section was retained from the first point where the acceleration in the y-axis (the direction of travel) goes over 20% of the maximal value to the last point where the acceleration in the y-axis is still over 20% of the maximal value. Runs that contained too few data points or that contained anomalous data (for example a run that only contained noise with a couple of seemingly random high peaks) where removed from the dataset.

I. Finding minimal set of required features

The total amount of features considered is quite large (100 features) and some of them may provide more information than others. To make the classification more efficient combinations of features can be compared by their accuracy. Then only those combinations are kept that yield an accuracy sufficiently close to the accuracy achieved when using all of the features. This experiment is intractable to execute completely but the desired result can be approximated by using a search algorithm.

II. Influence of number of data points on accuracy

By training the classifiers on different numbers of data points the influence of the size of the data vector on the training can be examined. Sizes from 128 to 2048 in powers of 2 were tried.

III. Classification of live data

When using the classifier to classify a run while the data is being recorded the classifier should make a decision as fast as possible while still being accurate. Because the classifier has to be fast it makes decisions on small overlapping windows, and the classification at a certain time is the majority of some previous classifications, this to ensure a certain level of accuracy.

IV. Results

I. General classification accuracy

The accuracy of every learning method was determined for both classification categories using subject-fold cross-validation. In the following table the achieved accuracies are displayed for a single experiment (the values differ slightly per execution) using the different methods:

Method	'Trained' accuracy	'Surface' accuracy
Support Vector Machines	72.9% ± 17%	39.6% ± 12%
Decision Trees	47.5% ± 36%	46.9% ± 13%
Naive Bayes	56.6% ± 34%	44.7% ± 11%

II. Finding minimal set of required features

This experiment was not completed due to a lack of time.

III. Influence of number of data points on accuracy

Accuracy for a window size of 256 is displayed under 'General Classification Accuracy'. The accuracies below are for a run using a window size of 2048.

Method	'Trained' accuracy	'Surface' accuracy
Support Vector Machines	48.0% ± 37%	45.2% ± 17%
Decision Trees	41.2% ± 35%	42.8% ± 6%
Naive Bayes	55.3% ± 40%	42.3% ± 16%

The accuracies are generally lower than for the smaller window size. As computation also takes longer the efficiency is certainly lower.

IV. Classification of live data

We expected, a majority decision over history to be more accurate, but our experiment does not confirm this assumption.

V. Conclusions

For the 'trained' label SVMs produce the best results. For the 'surface' label Decision Trees generally produce the best results but the differences are small. Accuracy performance is weakly dependent on window size and accuracy doesn't increase with the window size. If the features could be normalized to make them runner-independent that would significantly increase the accuracy, as a classifier trained on a single runner produces very good results (experimental results not shown).

VI. Project Management

We underestimated the necessary time for this project and perhaps gave little bit too much priority to projects for other courses. As a result less experiments were performed than initially planned and our report is less detailed than we would have liked. The amount of time spent on this part of the project per team member:
Toon Nolten: 24 hours.
Joren Verspeurt: 16 hours.