Classification of Human Activities from Time Series Accelerometer Data

Lucas Lofaro

November 19, 2017

GitHub Project Repository

https://github.com/lucasmlofaro/udacity_ml_capstone

I Definition

Project Overview

Today's smart phones come equipped with accelerometers and gyroscopes that capture our every move. What if we could leverage that data to make our devices a little more cognizant of how we spend our days? Just imagine the insights we could gain from real-time feedback about health and well-being. But it all starts with recognizing patterns in the data that allow us to determine which activity someone is performing at any given time. Once our devices understand the activities that make up our daily routine, they can begin to make recommendations that could improve our health or optimize our fitness regiment. Such awareness would also make our devices more secure since they would be able to detect deviations from our walking pattern, which could indicate that someone has stolen our device. The dataset for this project can be found in the UCI Machine Learning Repository.

Problem Statement

Our goal in this project will be to use the data collected from a chest-mounted accelerometer to determine which activity a participant was performing. Ultimately, our solution will be able to classify human activity with an accuracy better than chance, which in this case is 1 in 7, or about 14%. Our algorithm will only be able to classify activities from amongst the seven basic activities identified in our dataset [1].

To solve this problem, we will follow a three-phase procedure based on the scheme in Figure 6.

Phase 1: Feature Extraction

The data is provided as a stream of data samples containing the components of the acceleration vector at each moment. We cannot feed this kind of data directly into our classification algorithms, so we must first extract the pertinent data into a usable form. In this project we will build a nine-dimensional feature vector consisting of various statistical measures and characteristic qualities that can derived from the data.

Phase 2: Classification

Once we have extracted the relevant feature vectors, we can apply different classification algorithms to group the vectors and identify the corresponding activities. This is a multi-class classification problem because we are attempting to determine which category, amongst a number of possibilities, each vector belongs to. We will test two different classification algorithms, the details of which can be found in the **Algorithms and Techniques** portion of section II

Phase 3: Evaluation

After training the algorithms on the data, we will compare their results to see which method achieved higher accuracy. While our intuition might tell us one approach should work better than another, we have no way of knowing which approach will work better in the end. This is why we have chosen to implement multiple algorithms that rely on distinct assumptions. To quantify performance, we will rely on two classification metrics, the details of which can be found in the **Metrics** portion of Section I.

Metrics

Classification accuracy will serve as the primary evaluation metric for this project. This is a standard metric for classification problems and will allow us compare our results against the benchmark model discussed in section II. In addition, since our gaol is to distinguish different forms of activity, correct classification is of chief concern. Classification accuracy, a_c, is defined as the ratio of correctly classified samples, n, to the sample size, n_0.

$$a_c = \frac{n}{n_0}$$

We will also consider the Fowlkes-Mallows Index [2], which is calculated as the geometric mean of precision and recall.

$$FM = \sqrt{\text{prec} \cdot \text{rec}} = \sqrt{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}}$$

This is a slightly more appropriate metric for this problem than the more traditional `f1_score` because of the properties of the geometric mean. In general, the geometric mean responds to percentage changes in the inputs without regard to the range of the inputs. In our case, this will allow us to compare the performance of the two models even if they operate at different scales.

We interpret a larger value of the Fowlkes-Mallows Index to mean that the clustering is more similar to actual labels. Whichever algorithm produces the higher Fowlkes-Mallows Index will be considered a better representation of the true classification.

II Analysis

Data Exploration

The data used in this project comes from the UCI Machine Learning Repository [1]. The data set consists of fifteen participants, each of whom performed seven different activities:

1. Working at Computer
2. Standing Up, Walking and Going up/down stairs

3. Standing
4. Walking
5. Going Up/Down Stairs
6. Walking and Talking with Someone
7. Talking while Standing

The data is formatted as a time series where each sample provides the x, y, and z components of acceleration, as well as the corresponding activity label (see Figure 1).

	x							y		...	z		\
	count	mean	std	min	25%	50%	75%	max	count	...	75%	max	count	
activity										...				
0	1.0	1880.000000	NaN	1880.0	1880.0	1880.0	1880.0	1880.0	1.0	...	2366.0	2366.0	1.0	
1	52875.0	1874.238771	31.407085	321.0	1859.0	1876.0	1891.0	3154.0	52875.0	...	2379.0	3607.0	52875.0	
2	505.0	1876.835644	9.184329	1846.0	1872.0	1877.0	1881.0	1935.0	505.0	...	2385.0	2405.0	505.0	
3	12785.0	1864.957450	28.524095	1670.0	1851.0	1866.0	1875.0	2071.0	12785.0	...	2378.0	2745.0	12785.0	
4	29315.0	1856.009858	51.820782	1617.0	1827.0	1854.0	1883.0	2103.0	29315.0	...	2421.0	2666.0	29315.0	
5	3620.0	1892.182320	46.531890	1734.0	1862.0	1887.0	1924.0	2043.0	3620.0	...	2409.0	2689.0	3620.0	
6	1500.0	1874.610667	35.538295	1691.0	1857.0	1883.0	1893.0	1995.0	1500.0	...	2384.0	2563.0	1500.0	
7	15500.0	1879.182774	14.089323	1711.0	1874.0	1880.0	1885.0	2128.0	15500.0	...	2374.0	2556.0	15500.0	

	mean	std	min	25%	50%	75%	max
activity							
0	1942.000000	NaN	1942.0	1942.00	1942.0	1942.0	1942.0
1	2072.268615	61.702273	920.0	2034.00	2051.0	2112.0	2841.0
2	2022.974257	11.674884	1990.0	2017.00	2022.0	2027.0	2095.0
3	1980.616504	30.009402	1806.0	1964.00	1974.0	1990.0	2249.0
4	1984.631554	50.962880	1732.0	1954.00	1983.0	2015.0	2313.0
5	2039.090884	60.475561	1879.0	1987.00	2036.5	2080.0	2279.0
6	1972.384000	35.787498	1833.0	1955.75	1969.0	1984.0	2132.0
7	1958.098645	18.106520	1826.0	1951.00	1959.0	1964.0	2202.0

Figure 2: Summaries for the key statistics of the x, y, and z components of acceleration for one participant.

In this case, a simple accelerometer was mounted to the participant's chest and recorded a continuous stream of data while the participant performed the activity. A summary of the data for a single participant can be seen in Figure 2. Each participant performed all seven activities for a total of 105 (participant, activity) pairs. Of those pairs, 84 will be used for training while the remaining 21 will be used for testing.

	x	y	z	activity
47435.0	1899	2347	2092	1
104360.0	1786	2329	1988	7
42553.0	1928	2323	2123	1
56264.0	1847	2345	1999	3
79630.0	1813	2335	2074	4

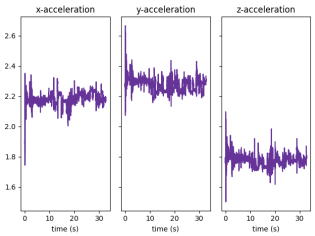
Figure 1: Random sample of entries in the dataset

Note that **activity 0** in Figure 2 is not really a separate activity. Further inspection shows that there is only every one point labelled **0**, explaining why the standard deviation is `NaN`. These are most likely test points, and we will simply ignore them in our analysis.

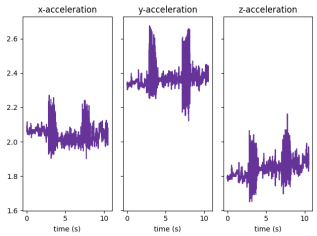
Exploratory Visualization

Much like a fingerprint, each activity has distinctive qualities that uniquely identify it. Certain activities will be rhythmic in nature (e.g. walking or climbing up and down stairs) while others will be characteristically arrhythmic (e.g. talking). In some cases we might also expect the magnitude of one component to be larger than another (at least on average). The presence—or absence—of any of these traits will give clues to the identity of an unlabelled activity.

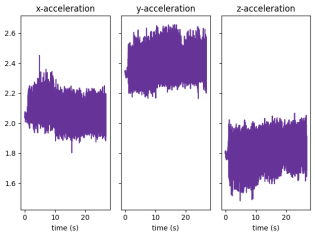
In order to gain some physical intuition, we can visualize the data in a couple of ways. First we'll look at the acceleration graphs produced by a single participant performing multiple activities (Figure 3). We can see that each graph has its own characteristics. For example, in Figure 3a the means of the x and y components are very close and the mean of the z component is smaller, while the means of each component in Figure 3d all differ. Also, the graph in Figure 3c shows much more accelerated (higher-frequency) motion than the graph in Figure 3b. To the human eye these graphs appear distinct enough that a machine learning algorithm should be capable of classifying the data.



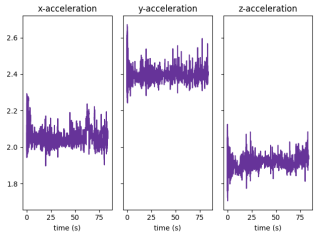
(a) Participant 7 working at a computer.



(b) Participant 7 standing.



(c) Participant 7 walking.



(d) Participant 7 talking while standing.

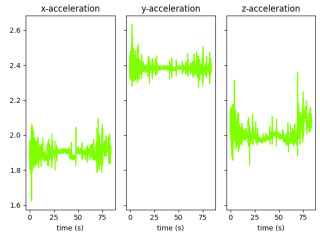
Figure 3: Same participant performing multiple activities.

Now, we could also compare the graphs of the same activity performed by different participants (Figure 4). Here we can see that the graphs share many qualities, and it is fairly clear that they are all doing the same thing. In each the graph the relative magnitudes, dispersion of means, and natural frequency all appear consistent. This is also good news for our clustering algorithms since it confirms feature vectors for the same activity look similar across all participants.

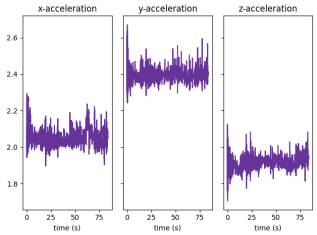
Algorithms and Techniques

In this project we will consider two possible classification algorithms (see Figure 5):

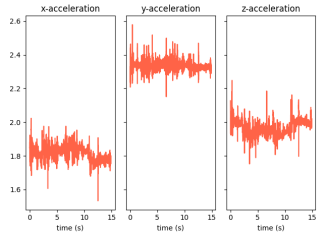
k-Means Clustering If we assume that data from different people performing the same activity will look similar, then we can apply the canonical classification algorithm to group similar vectors together. The algorithm has two key steps:



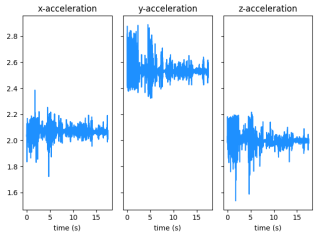
(a) Participant 1 talking while standing.



(b) Participant 7 talking while standing.



(c) Participant 12 talking while standing.



(d) Participant 15 talking while standing.

Figure 4: Different participants performing the same activity.

1. Centroid Placement
2. Cluster Assignment

We must consider the initialization of the centroids to get the algorithm started, but that is a detail. Once it gets going, the algorithm assigns each data point to its closest centroid, then reassigns the centroid position based on the average of all of the points assigned to it. This process continues until the positions of the centroids converge to a stationary position (at least to within a threshold).

Though we can't visualize it, this algorithm assumes that these data points form "clouds" in nine-dimensional space, meaning that the points tend to cluster into groups. One potential problem with this assumption is that some of these activities consist of combinations of other activities. For example, an activity like walking and talking with a friend is composed of the more fundamental activities of (1) walking and (2) talking. This could cause some of those clusters to overlap or blend together, making it difficult to distinguish where one group ends and the other begins.

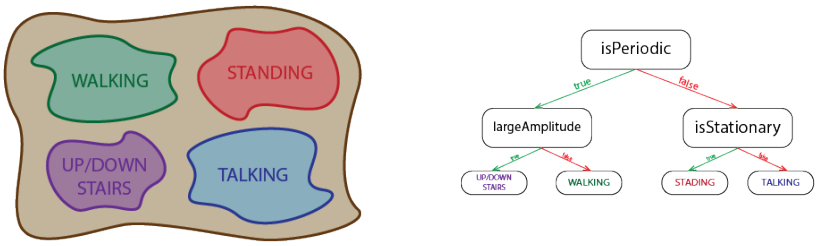
The advantage of this algorithm is that is relatively simple to implement and will give us a good understanding for the dispersion of points in the dataset. That is helpful in this problem because it will tell us if two data points corresponding to the same activity really are similar, as defined by the distance in feature space. That would mean that the same activity looks similar in all people, which would validate our assumption above.

Decision Tree Classifier Alternatively, we might expect that a few heuristics, or rules of thumb, could easily classify the data. For instance, a human might look at the data and recognize that it is periodic and therefore rule out activities such as standing still or talking. If we can determine the right exclusion criteria then we should be able to separate the data with only a few simple rules. This approach assumes the activities have unique traits that can be used to split the data by logical deduction.

This algorithm works by finding the best line that separates the data, then continues to divide the each subsection by the same method. In this way the data is split based on a series of choices that ultimately identify which activity the point corresponds to. It is very possible, and even likely, that multiple leaf nodes will contain the same activity.

That means there are some obvious indicators of the activity as well as others that are less conspicuous.

Decision trees are advantageous here because we can visualize the decision making process and get a better sense for which features best separate the data. We can use this information to potentially improve our feature selection in future models if we find that one features is more or less significant that we might have initially expected.



(a) k-means clustering (b) Decision tree classifier

Figure 5: Simplified graphical representations of possible classification algorithms.

Fourier Analysis

Since many of these activities are periodic in nature, we can extract useful information by transforming into the frequency domain. While the time domain is helpful for capturing changes in quantities like position, velocity, and acceleration, the frequency domain gives us a holistic view of the how that motion evolves throughout the experiment. Fourier analysis attempts to break our signal down into its component frequencies by modelling the function as sum of periodic functions.

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \cos(nx)$$

The coefficients a_k and b_k are the amplitudes corresponding to the frequencies of each sinusoid. The relative magnitudes ($\|a_k\|$ and $\|b_k\|$) tell us how much each fundamental frequency contributes to the final signal. Larger amplitudes have the most impact on the overall shape of the function, so we will only concern ourselves with the most dominant frequencies.

Benchmark

A study on the automatic classification of on-body accelerometers [3] will serve as our benchmark for this project. While the implementation of this model differs significantly from the solution outlined here, the conceptual methodology is nearly identical (Figure 6). The benchmark model also identifies seven fundamental activities, though many differ from those in our datasets. This model trains classifiers for each individual in an attempt to capture unique mannerisms. Since our goal is to detect the generic motion involved in certain tasks, that level of modelling is not required here. Ultimately, this study achieved a classification accuracy of 84% (over 99% after rejecting spurious data).

Figure 6: Conceptual scheme of a generic classification system with supervised learning [3]

III Methodology

Data Preprocessing

Because we are dealing with time series data, we will need to extract the most relevant features to form a vector that can fed into our classification algorithms. For the purposes of this project, the vector will contain the mean, standard deviation, and amplitude of the dominant frequency for each component of acceleration.

$$\vec{v} = \left(\vec{\mu}, \vec{\sigma}, \vec{f_0}\right) = \left(\mu_x, \mu_y, \mu_z, \sigma_x, \sigma_y, \sigma_z, f_{0x}, f_{0y}, f_{0z}\right)$$

mean As we saw in Figures 3 and 4, the relative positions of the mean values for each component give us a clue as to which activity is being performed.

std. dev. In this context, the standard deviation gives us an idea of how volatile the movement is. Figures 3a and 3c demonstrate contrasting volatility.

dom. freq. Each of the graphs Figures 3 and 4 appears periodic in nature. Periodicity implies a fundamental frequency, and a higher frequency implies more rapid motion. This will be useful when trying to distinguish between activities like standing and walking.

Building the feature vector is relatively straightforward, except for the frequency analysis. Both the mean and standard deviation are taken care of by `numpy`'s built-in `mean()` and `std()` functions respectively. We need to use Fourier analysis to extract the dominant amplitudes by breaking the signal into its component frequencies. The "amount" of each frequency contained in the original signal is the amplitude we seek. This can be achieved through the use of `numpy`'s fast Fourier transform function, `fft()`. This function returns complex amplitudes, which contain both a magnitude and phase. In the context of this problem the phase is of no real concern to us, so we simply discard it. The maximum of the returned amplitudes corresponds to the dominant frequency.

Implementation

Our first step is to separate the data into training and testing sets. We use the `shuffle` and `train_test_split` functions from `sklearn` to break the data set into 84 training points (80%) and 21 testing points (20%). Next, we train two `sklearn` classifiers, `KMeans` for clustering and `DecisionTreeClassifier` for the decision tree. Both classifiers are fit with the same test data to ensure accurate comparison. Unless otherwise specified, the algorithms were initially run with their default parameters. Normally we would perform normalization before training the classifier, but in this case it is unnecessary because all of the data is on the same order of magnitude.

Since we have seven distinct activities, we will run `KMeans` with `n_clusters=7` to make sure we obtain seven groups. In theory there should be a bijective mapping between groups and activities, meaning that all data points in each group correspond to the same activity and no two data points from different groups correspond to the same activity.

One benefit of using a decision tree is that we can actually visualize the process that our algorithm uses to deduce the correct activity. We can produce a graphical representation of of the tree using `graphviz` (Figure 7a. Each leaf node corresponds to a classification, and each parent node acts like a conditional statement that is used to separate the data. To classify a sample we follow a branch of the tree starting at the root and working our way down depending on the results of tests made at each successive node.

Because the dataset only consists of 105 data points, the performance of the clustering algorithm seemed to depend heavily on how the data was split into training and testing sets. For that reason we chose to measure performance over and average of trials. This procedure proved to be valuable as we began adjusting the algorithm since we could definitively say that performance was or was not improving.

Refinement

When we extracted the features from the time series data we decided to use only the most dominant amplitude in our analysis, however it is possible that other fundamental frequencies could be relevant. In order to compare the importance of each frequency, we can simply convert the corresponding amplitude to a relative percentage of the dominant amplitude.

$$\|f_k\|_{\text{rel}} = 100 \cdot \frac{\|f_k\|}{\|f_0\|}$$

If any other amplitudes are of comparable size—say at least 5% of the dominant magnitude—we may consider adding that amplitude as another feature. After analysis we found that none of the subsequent frequencies exceeded 1% of the amplitude of the dominant frequency. Even though this additional effort did not yield any significant result, it is still important to consider the potential effect of underlying frequencies.

While there aren't may parameters we can tune in the k-means classification algorithm, the centroid initialization can be rather important. To ensure that randomly initializing the centroids will not affect our results, we can simply run the algorithm multiple times with different initializations. Each of those models is assessed using standard 5-fold cross validation, and we select the model with the best score.

Similarly, we can use cross validation to improve our results for the decision tree model. In this case we use a grid search to optimize the depth of the tree, the minimum number of leaf nodes, and the minimum number of samples on which the data can be split. In theory, the tree's `max_depth` shouldn't need to exceed

$$\log_2 n = \log_2 105 \approx 7,$$

where n is the number of data samples. The ranges for each parameter and the optimal values are as follows:

- `max_depth` $\in [1, \log_2 n]$, optimal value: `5`
- `min_sample_split` $\in [2, n/2]$, optimal value: `3`
- `min_sample_leaf` $\in [1, 10]$, optimal value: `1`

We can see the results of optimizing the tree in Figure 7b. After tuning the parameters the tree looks more balanced and seems to have more effectively separated the data.

IV Results

Model Evaluation and Validation

To find out which algorithm performed better on the testing set, we will compare each model's classification accuracy and Fowlkes-Mallows index. A summary of the initial and final scores are reported in Table 1. Interestingly, the decision tree algorithm outperformed the clustering algorithm by a landslide. This result gives us a better intuition for distribution of the data. Because the clustering algorithm had trouble classifying points, our assumption about the similarity of activities amongst individuals seems to be incorrect. That means, for example, that each person walks a little differently, so ones person's walk might look like someone else's jog. Geometrically, we can also say that points corresponding to the same activity are not necessarily located near each other.

	`KMeans`	`DecisionTreeClassifier`
Accuracy	5.93% / 11.85%	77.78% / 87.41%
Fowlkes-Mallows	0.37 / 0.4	0.59 / 0.76

Table 1: Scores for both the inital and optimized models are shown as an average results from ten iterations of the each algorithm. Scores should be read `initial_score / final_score`

Moreover, the decision tree was successful because the activities themselves have distinguishing features. Even if no two people walk the same, walking still looks very different from standing or talking regardless of who you are. We can even see how gracefully the algorithm sorted the data in Figure 7.

After tuning the parameters on the decision tree, we found that the `max_depth` of the tree decreased and the tree became more balanced. This is intuitively pleasing since we would expect that each activity could be distinguished with relative ease. That is, there should be no need to dig down seven or eight layers to figure out what something is.

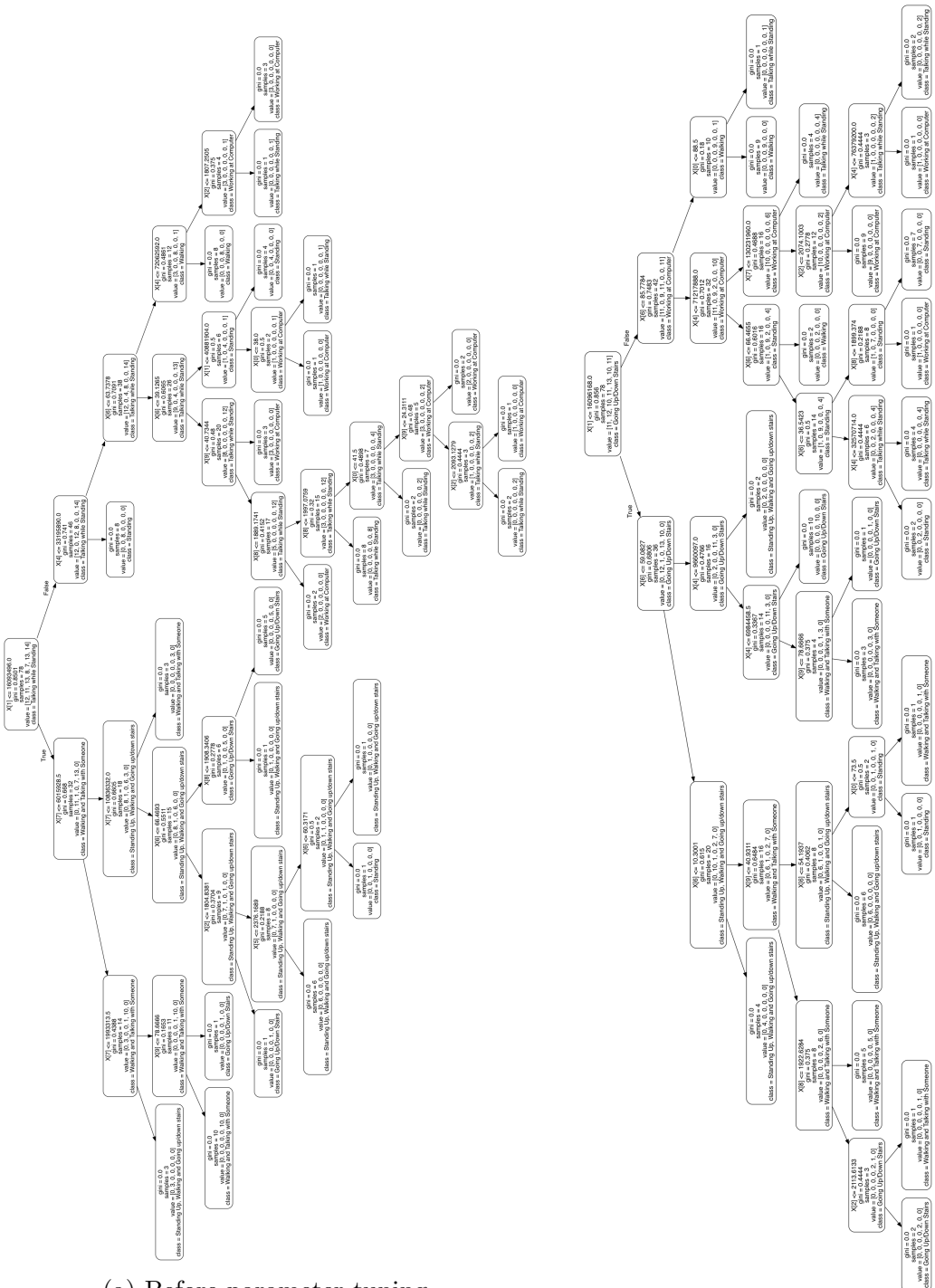
Fowlkes-Mallows index improved much more significantly for the decision tree classifier than the clusering algorithm. This tells us that the improvements to the decision tree algorithm were much more effective. Despite an increase in classification accuracy, the clustering algorithm did not really benefit from additional tuning. This likely has to do with size of the dataset and the fact that classification accuracy is somewhat dependent on the particular split of the data.

Now, since we have seven distinct activities we are trying to classify, the probability of guessing the correct answer randomly is 1 in 7, or about 14%. This tells us that the optimized clustering algorithm is essentially no better than the roll of a die. However, the final decision tree performs more than 6 times better than chance, which is certainly a significant result.

Justification

Analysis performed with the benchmark model II yielded classification accuracy of 84%. While we extracted only nine features, the benchmark model used more than 300 features to characterize each point. In addition, the benchmark model created activity classifiers each individual rather than one classifier for all participants. Finally, the benchmark model also considered hierarchical clustering, whereas we assumed all activities were independent. The highest accuracy achieved with our optimized decision tree model was about 89%, and was on average about 80%. Our best results outperform the benchmark, however our average result prove that our algorithm is essentially on par with the benchmark.

Given the simplifying assumptions of our model, the results are significant enough to prove that classification of human activities from accelerometer data is plausible and decently accurate. We must also keep in mind that our dataset is fairly small, and we cannot expect our algorithm to generalize beyond the simple activities we have classified here. However, for the purposes of this project, the solution is more than sufficient to classify a handful of human activities.



(a) Before parameter tuning

(b) After parameter tuning

Figure 7: `graphviz` representations of decision trees

V Conclusion

Free-Form Visualization

Because our final model is a decision tree, we have the opportunity to visualize the decision making process. As we can see in Figure 7, our algorithm runs through a sequence of conditional nodes until it reaches a leaf node, where it is classified as one of our fundamental activities. In the case of the optimized tree in Figure 7b, we have to ask at most six questions before we figure which activity we are dealing with. Each branch of the tree can be though of a logical chain of thought that leads to the deduction of an activity.

Each node separates the data samples into two camps: those that satisfy the node's condition (the `true` branch) and those that don't (the `false` branch). This procedure is continued until all of the samples in each camp correspond to a single activity. Therefore, the class label on each leaf node tells us which activity to ascribe to each data point.

Reflection

Over the course of this project we follow a typical pipeline for solving of the problem of classifying human activities from accelerometer data. We began by extracting useful features from the time series data such as mean, standard deviation, and dominant frequency. Each of these characteristics is meant to be a distinguishing feature that represents the underlying patterns inherent in the activity. We used Fourier analysis to help us isolate the underlying frequencies in the data and identify the most import amplitudes. Next, we collected these features into vectors and fed them into a couple of classification algorithms. We tried to cluster the data because we assumed activities would clump together, and we tried decision trees because we assumed a set of heuristics might be sufficient to model the problem. We evaluated both algorithms according to their classification accuracy and found that clustering didn't work as well as we would hope. On the other hand, decision trees worked quite well, classifying the correct activity about 60% of the time.

Ultimately the most challenging parts of this project were the frequency analysis and cross-validation. At first it wasn't obvious how important different fundamental frequencies would be, so we had to compare each amplitude to the dominant frequency to see if the magnitude was large enough to be considered significant. Cross-validation proved difficult because the grid search needed a small enough space of search parameters to be relatively fast, but still produce a stronger model. Though we did not ultimately produce a better model, we found optimizations for the search space to decrease the number of possible parameter pairings in the gird search.

While the results of this project are promising, the model itself relies on simplifications of the typical best practices for solutions to similar problems. In general, the methodology and principles of this project should be followed by subsequent research, but the precise techniques can vary depending upon the level of detail that needs to be modelled. Overall, our results are as good as could be expected for a simplified model and a rather small dataset.

Improvement

As a first step to improving this model, we could add a few more features that better capture the essence of each activity. For example, we could use numerical integration techniques to derive the velocity and position graphs for each activity and subsequently extract addition features form that data as well. Since we are working with a dataset that contains only 105 data points, the best we can hope for is a more precise description of each point. Using position

data might be more effective for classifying stationary activities, like talking or standing, while velocity data might help for classifying walking or going up and down stairs.

A further generalization might consider data from accelerometers that are worn on the wrist or held in the pocket like a cell phone. Since most of us don't walk around with accelerometers mounted to our chest, it would be beneficial to see if data collected from the devices we already use could still provide the same degree of accuracy.

As it stands, this solution serves more as proof of concept than anything else. There is a lot of room to build upon what we have here to develop models that are much more robust and capable of classifying many more activities. This project only scratches the surface of what can be done with accelerometer data and doesn't even begin to explore all of the potential application once we have reliable activity recognition.

References

[1] O. Casale, P. Pujol and P. Radeva. Personalization and user verification in wearable systems using biometric walking patterns. *Personal and Ubiquitous Computing*, 2012.

[2] E. B. Fowlkes and C. L. Mallows. Machine learning methods for automatic classification of human physical activity from on-body accelerometers. *Journal of the American Statistical Association*, 1983.

[3] A. Mannini and A. M. Sabatini. Machine learning methods for automatic classification of human physical activity from on-body accelerometers. *sensors*, 2010.