

CS229 Final Report: Activity Recognition Using Cell Phones

Travis Geis, Sanjay Srinivas, Rohit Talreja
11 December 2015

Introduction

Motivation

With the increasing prevalence of mobile phones with Internet connectivity and position and acceleration sensors, tracking phone users' activities has become easier and more common. The popularity of the "quantified self" trend indicates that users like being able to record their daily activities using objective metrics like calories burned, steps taken, and total distance travelled. Users might for example track their fitness activity to report their health to insurers or share and compete with friends. They might also track their driving or bicycling habits to estimate their environmental impact.

However, in most cases, tracking fitness-related activity or other daily habits depends on the users' explicit opening of the relevant app to begin data recording and to indicate what activity the user will record. If there were a way for the mobile phone to decide automatically what the user is doing at a given time, then the phone could track the user's activity throughout the day even if the user does not remember to start the tracking.

Automatic identification of the user's activity also makes possible many more helpful features. For example, the phone could determine that a user is driving and temporarily silence notifications to reduce distraction. If the user is sitting for long periods of time, the phone could remind the user to take periodic breaks. The automatic recognition of the user's activity would give the phone more information about the current context of use, which could allow application developers to make more useful applications with fewer distractions.

Our project seeks to determine the current activity of a smartphone user, such as walking or sitting, using sensor data gathered from a typical smartphone. As such, we trained our learning algorithms on a dataset of sensor data gathered from a waist-mounted smartphone. The input is a vector of numerical data derived from raw accelerometer and gyroscope recordings. The output of each algorithm is a label identifying the activity the user is performing.

Related Work

Bao and Intille were the first to use acceleration data (obtained from five biaxial accelerometers worn simultaneously on different parts of the body) to perform activity recognition. However, special body-mounted sensors are inconvenient compared to smartphones. Kwapisz et al. demonstrated the effectiveness of a single Android cellphone as a sensor platform for activity recognition, achieving prediction accuracies of over 90% for multiple activities.

Power and processing limitations on mobile devices constrain the choice of classification algorithms. Anguita et al.¹ demonstrated that multi-class SVMs can be an effective classifier for human activities, and that the multi-class SVM can be optimized to use fixed-point arithmetic with little loss of accuracy, which makes the SVM an ideal candidate for recognition tasks on resource-constrained mobile phones.

Ravi et al. performed an exhaustive study of which classification algorithms (e.g. decision trees, k-nearest neighbors, SVM, naive Bayes) achieved the greatest accuracy when used as the base-level classifiers for a multi-layer model (including boosting, bagging, and plurality voting). They ran each combination on a variety of different dataset segregation models: 1) data collected for a single subject on one day used as training data and another day used as test data, 2) data collected for a single subject on one day used as training data and data from another subject used as test data, 3) data collected from multiple subjects that was mixed and cross-validated, and 4) data collected for a single subject over multiple days that was mixed together and cross-validated. They concluded that the way data is segregated into training and test set makes a significant (as high as 30%) difference in accuracy when all other parameters are the same.

¹ Anguita et al., "Human Activity Recognition on Smartphones."

Dataset and Features

There are many possible ways to collect data on user motion, such as through wrist-mounted accelerometers, multiple body-mounted sensor packs, and through inertial and positional sensors housed inside most modern cell phones. Since our goal was to predict smartphone user activity based on the sensor readings of the phone, we selected a dataset² that most closely matches the data that a typical phone application could gather: readings directly from the phone's sensors.

The data comprise labeled examples of 30 people performing six different activities: walking, climbing stairs, descending stairs, sitting, standing, and lying down. We consider walking, climbing stairs, and descending stairs to be “motion activities,” in contrast to sitting, standing, and lying down, which we deem “sedentary activities.” The activities within each category have similar accelerometer and gyroscope readings.

The data were gathered through a waist-mounted smartphone (specifically, a Samsung Galaxy SII released in 2011 and considered top-of-the-line for that time). The data include both accelerometer and gyroscope readings, sampled at a constant rate of 50 Hz, which were then segmented into 2.56-second windows with a window overlap of 50%. Along with raw readings, the dataset includes filtered and derived features such as gravity acceleration separated from body acceleration via a low-pass filter, jerk, standard deviation, and cross-axis correlation.

Of the volunteer subjects, 70% were selected randomly to contribute to the training dataset, and 30% for the test set. In total, the training and test matrices include 561 features per data point, with data split into 7,352 training examples and 2,947 test examples.

Methods

We employed the Support Vector Machine (SVM) algorithm for our supervised learning tasks. The SVM algorithm is an optimal-margin classifier that seeks to label the training examples such that the smallest geometric margin between the training examples and the decision boundary is maximized.³ In other words, the SVM seeks to minimize

$$\begin{aligned} \min_{\gamma, w, b} \frac{1}{2} \|w\|^2 \\ \text{s.t. } y^{(i)}(w^T x^{(i)} + b) \geq 1, i = 1, \dots, m, \end{aligned}$$

where the geometric margin is proportional to $1/w$, so minimizing w will maximize geometric margin. We used the SVM implementation provided by libSVM.⁴

We employed the k-means unsupervised clustering algorithm for our unsupervised learning experiments. K-means is an iterative algorithm that alternates between assigning points to the nearest cluster, and updating the cluster centroid to reflect the mean position of the points assigned to that cluster.⁵ The cluster centroids can be initialized randomly or deterministically. Training examples are indexed from $i=1$ through n , while centroids are indexed from $j=1$ through m .

In other words, repeating until convergence, for every i , the algorithm sets

$$c^{(i)} := \arg \min_j \|x^{(i)} - \mu_j\|^2$$

which corresponds to assigning each point to the nearest cluster, and for each j , the algorithm sets

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}$$

where m is the number of centroids and μ_j is the position of the j th centroid.

² Anguita et al., “A Public Domain Dataset for Human Activity Recognition Using Smartphones.”

³ Ng, Andrew. “CS229 Lecture Notes 3.”

⁴ Chih-Chung Chang et al.

⁵ Ng, Andrew. “CS229 Lecture Notes 7a.”

Throughout our experiments, we trained our learning algorithms on the 70% of the subjects comprising the training set, and then measured the test-set accuracy on the remaining 30% of subjects who comprise the test set. An algorithm’s accuracy is defined as the percentage of test points that it classifies correctly.

Experiments

Binary SVM Classification

We began our analysis on the dataset by applying a binary SVM classifier to each pair of activities. The goal was to determine which activities were easily pairwise distinguishable from each other (i.e. did not involve similar motions) and which were more difficult to distinguish. This also gave us a baseline accuracy metric for assessing the multi-class SVM algorithm, since if accuracy on binary classification was close to chance (~50%), it would be more difficult to make an accurate multi-class prediction.

As predicted, there was high accuracy when distinguishing between a motion activity and a sedentary activity (eg. walking vs. standing) but a much lower accuracy when distinguishing between two motion activities (walking vs. walking upstairs) or two sedentary activities (sitting vs. standing). In the last example, the accuracy was only marginally above chance. Based on this data, we were optimistic that the multiclass algorithm would be able to differentiate movement-based activities from sedentary ones, but concluded that it might err when choosing the correct activity from within those categories. The prediction accuracies are summarized in the table below.

	Walking Upstairs	Walking Downstairs	Sitting	Standing	Lying Down
Walking	57.9%	77.1%	99.6%	99.6%	99.3%
Walking Upstairs		76.7%	98.4%	98.6%	97.8%
Walking Downstairs			98.8%	98.9%	98.6%
Sitting				53.9%	67.3%
Standing					67%

Table 1: Binary classification accuracy as a percentage of correctly classified test data. There were ~1000 test data points for each pair of activities compared.

Multi-Class SVM Classification

Even though some of the binary classification results were quite promising, we were unsure how well they would generalize with six output labels. Generalization accuracy is important because the ability to label the activity as one of six, rather than explicitly comparing the likelihood of every pair of activities, would be more helpful to applications seeking the general context of phone use.

Previous work by Ravi et al. demonstrated effective classification using twelve features gathered from the smartphone sensors. They used twelve input features derived from a single triaxial chest-mounted accelerometer. Employing a multi-class SVM, they demonstrated a 63.00% prediction accuracy when training the SVM on one set of users and testing on a different set of users performing the same activities, as our dataset is designed to facilitate.

To keep the computational cost of prediction low, we decided select up to 30 input features to use in a multi-class SVM. However, we did not know which features would be most useful, so we used a feature selection algorithm to determine which 30 features to use.

Feature Selection

Our feature selector uses a linear forward search. To choose a feature, it iterates over all possible features, and for each feature, trains the SVM on that feature along with the other previously-chosen ones. The feature resulting in the lowest test error is then added to the chosen feature list, and the process repeats until the algorithm has chosen the desired number of input features. The 5 best-performing features chosen by the algorithm are shown in Table 2.

Feature	Name	Axis	Standalone Prediction Accuracy (%)
4	Body Acceleration Standard Deviation	X	40.96
5	Body Acceleration Standard Deviation	Y	41.19
54	Gravity Acceleration, Min	Y	24.16
56	Gravity Acceleration, SMA	N/A	33.80
64	Gravity Acceleration, Entropy	Y	31.93

Table 2: Features chosen by the feature selection algorithm, and their prediction accuracies when used alone. “SMA” denotes “Signal Magnitude Area.”

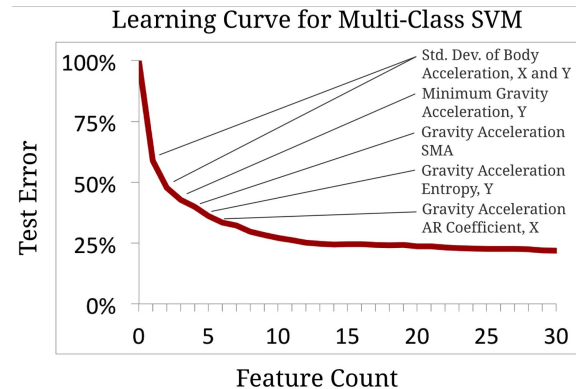


Figure 1: Learning curve showing the progression of forward search feature selection.

Classification Results

Running the feature selection algorithm using the selected 30 features we were able to get a multi-class SVM prediction accuracy of 78.1%, a 15.1% increase over the baseline method described by Ravi et al. Although there are 561 total features in the dataset, it is not computationally efficient nor helpful for accuracy to use more than these 30 since, as we see in Figure 1, the learning curve converges after approximately 15 features.

K-means

Since there were six distinct activities, we were curious if an unsupervised learning algorithm like k-means could accurately partition the data and converge to one cluster per activity. With six randomly-initialized centroids, we achieved convergence within just 23 iterations using standard Euclidean distance as the similarity metric and all 561 input features. We also experimented with initializing one centroid to each activity label (non-random initialization) but found that in 95% of 50 trials this non-random initialization also converged to 6 activity clusters. In the remaining 5% of trials, the deterministically-initialized algorithm converged to fewer than six clusters.

Figure 2 shows the cluster composition of a sample k-means trial. We can see that for each cluster, there is a unique activity label that dominates that cluster. Specifically, the mode of Cluster 1 is the label “walking”, the mode of Cluster 2 is the label “climbing stairs”, and so on. However, the clusters didn’t clearly partition the data into six distinct activities, specifically, the mode activity label only accounted for an average of 47% of the data points in each cluster. Clusters with a centroid of a sedentary activity, such as sitting, contained a significant number of data points from the other two sedentary activities, standing and lying down. Similarly, clusters with a centroid of a motion activity contained a significant portion of data points from the other motion activities. While this led to low accuracy in choosing which one of six labels to apply to a test point, we achieved over 98% accuracy in binary classification in choosing to which category the activity belongs (sedentary vs. motion).



Figure 2: K-means activity cluster composition

Further SVM Experiments

Prior work by Ravi et al. has shown prediction accuracy improvements from using multi-layer SVMs. In the multi-layer SVM, multiple “base-level” SVMs attempt to classify the input example, and their output labels are fed as inputs, along with other input features, into an upper-level SVM. For our base-level classifiers, we used six “1 vs. all” SVMs. The i th base-level 1 vs. all SVM classifies an input as label “ i ” or “not i ”, where i is 1 to 6. These SVMs use all input features. The upper-level multi-class SVM then uses the base-level SVM outputs as additional input features, augmenting the 30 automatically-selected features.

Inaccuracies in the predictions of the base-level SVMs degraded the output of the upper-level SVM when compared to the standard multi-class SVM. The multi-layer SVM achieved a test set accuracy of 76.8%.

We also experimented with plurality voting in our attempts to improve accuracy. In plurality voting, the 6 base-level SVMs similarly assign labels to the training examples. A vote counter tallies the votes for a given example to decide which label to apply, breaking ties randomly. We’re able to use plurality voting without normalizing scores since there are roughly equal proportions of each label in the training set. Using all input features, voting achieved a test-set accuracy of 64.4%.

Conclusions

Of the algorithms we implemented, the multi-class SVM with 30 automatically-selected features produced the highest test accuracy of 78.1%. This was better than our other algorithms (multi-layer SVM, plurality voting, and k-means) and also the baseline algorithm from Ravi et al. As mentioned earlier, we think that similarity of the sensor readings for certain activities significantly hampered the accuracy.

We believe that the accuracy of the multi-class SVM can be increased by training on a larger data set, specifically one that has data from a particular subject in both the training and test sets rather than splitting up subjects to either the training or test set. We could also train and test the SVM a single subject, which would allow better prediction accuracy for that subject at the cost of generalizability. However, this might actually be preferable since the overall goal is to recognize the current activity for only the primary user of the phone.

Algorithm	Accuracy
Multi-Class SVM	78.1%
Multi-Layer SVM	76.8%
Voting	64.4%
K-Means	47.4%

Figure 3: Test set accuracy of each algorithm

Works Cited

1. Anguita, Davide, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.
2. ---. "Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine." IWAAL 2012. Vitoria-Gasteiz, Spain December 2012.
3. Chang, Chih-Chung and Chih-Jen Lin, LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1--27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin>
4. Kwapisz, Jennifer R., Gary M. Weiss, and Samuel A. Moore. "Activity Recognition Using Cell Phone Accelerometers." *ACM SIGKDD Explorations Newsletter* 12 (Dec. 2010): 74-82. Print.
5. Ng, Andrew. "CS229 Lecture Notes 3." Stanford University.
6. ---. "CS229 Lecture Notes 7a." Stanford University.
7. Ravi, N., Dandekar, N., Mysore, P., & Littman, M. (2005). Activity Recognition from Accelerometer Data. American Association for Artificial Intelligence.