Recognizing Human Activities Through Predictive Modeling With Feature Selection

Introduction:

The advent of smartphones has ushered in a new era in which people can keep track of and monitor their daily activities with ease. That said, smartphones, in particular those manufactured by Samsung, have embedded accelerometers and gyroscopes that can record three-axial velocity and three-axial acceleration of their users at a constant rate of 50Hz. The said convenience then can generate a myriad of features with accelerometer and gyroscope signals.

Smartphone producers would like to have their products predict human activities, be it sitting, walking, or simply laying, using only a handful of the generated features without compromising accuracy since such a feat would shave off unnecessary memory usage and processing power.

With that said, our aim was to encapsulate human activities with five prominent features among 561 in total. Using exploratory analysis, feature selection methods and Bayes Network algorithm, we were able to extrapolate human activities with 88% accuracy from available data.

Data Collection and Preparation:

Data Collection

For our analysis, we used "Human Activity Recognition Using Smartphones Data Set,"[1] which is made available to the public by the courtesy of the University of California Irvine. As a side note, the data is an accumulation of extensive experiments carried out with 30 volunteers who performed six daily activities - walking, walking upstairs, walking downstairs, sitting, standing and laying - while wearing a smartphone on their waists. The data is available in two formats - txt and rda. We downloaded the data in rda format on January 26th using R programming language.

Data Preparation

Our team prominently used Weka and R for this particular project since Weka churns out much information without going through the hassle of writing and checking long lines of code. Given how much time we had, it seemed reasonable to resort to a program that is intuitive and straightforward. Nevertheless, sometimes, the project demanded more than what Weka is capable of. For such occasions, our group decided to use R.

With that in mind, it was imperative that data quality is up to a reasonable standard. Fortunately, the data set was overall complete, consistent and believable to begin with. Likewise, we found that the set contained no missing values by using "is.na" function in R on each column. Therefore, we saw no need to undertake imputation step of data preprocessing. What with the thorough explanation of each feature in the 'Readme.txt' file of the data set, our group encountered only a few difficulties in understanding the data. In terms of redundancy, the data included no duplicate columns or tuples in terms of values. However, we did note that it contained duplicate column names; researchers apparently recorded three attributes - "fBodyAcc-bandsEnergy()", "fBodyAccJerk-bandsEnergy()", and "fBodyGyro-bandsEnergy" - for 14 pairs of test subjects twice. For instance, there were two columns that were named "fBodyAcc-bandsEnergy()-1,8." Nonetheless, each column had distinct values. The same can be said of other pairs of columns that had the same column names. Therefore, our team

concluded that the researchers must have had made a mistake naming columns when recording the test subjects. Columns with the same names projected no threat to validity and accuracy of the project since values in each column themselves were not redundant.

For our analysis, we have initially split the data into two sets - training, and test - in the ratio of 80:20 in accordance with the industry standard through random sampling function in R. Since Weka can carry out cross-validation with the training set, Weka users can often forgo creating a separate validation set.

Exploratory Analysis:

Sizing up data through visualization is a key step in exploratory analysis. Thankfully, Weka conveniently provides distribution of each feature as soon as users import their data. That being the case, we had to convert our training and test sets into csv format since Weka can only handle rdff and csv files. R language has “write.csv” function that can effortlessly turn a r file into a csv file. Our team encountered “not unique attribute names” error when we tried to have Weka read then-converted csv file because the there were columns with the same names as mentioned above. Therefore, our group had to make individual column names unique with “make.unique” function in R.

Every attribute in the set is numerical except one feature that informed the readers which activity a test subject was undergoing through nominal values such as “walking,” “laying,” and etc. The majority of the features had normal distributions. There were also a fair number of either right skewed or left skewed distributions. For the most part, the range for each attribute was [-1, 1] which made normalization unnecessary.

One pattern that stood out from visualization was that three-axial attributes, divided into three columns for each axis, had unambiguously similar distribution to one another. For example, a feature named “tBodyAcc-energy()” is split into three columns for each axis - “tBodyAcc-energy()-X”, “tBodyAcc-energy()-Y,” and “tBodyAcc-energy()-Z” - and distribution in one axis seems to resemble that in another. It seemed obvious at first but one should note the pattern in light of the fact that the data set also has a handful of such features that are strikingly different from one another with respect to their distributions. For instance, “tGravityAcc-mean()-X,” “tGravityAcc-mean()-Y,” and “tGravityAcc-mean()-Z” are vastly dissimilar to each other. Therefore, the investigators reasoned that a good starting point of the research would be to tinker with three-axial attributes that had axises with distinct distributions since they contribute the least to the redundancy of the data.

Methods:

Feature Selection

Feature selection was probably the most vital part of the whole project due to a constraint that we had to form a subset of at most five attributes that can aptly encapsulate the data set. A gratuitous selection of features could potentially impair accuracy of our model. Therefore, our team had to come up with a sensible method of sorting out “good” features from a wide pool of “bad” ones. Before testing the waters, we judged that reading past studies pertaining to the same topic would give us a good guidance as to how we should go about filtering attributes. Luckily, our group came upon a good number of relevant studies.[2]

One such research that caught our eyes was “Physical Activities Monitoring Using Wearable Acceleration Sensors Attached to the Body,”[3] in which researchers used correlation-based feature selection in conjunction with ReliefF. That said, it seemed sound to use

correlation-based feature selection for our project as well because we have came across a sizeable number of attributes that correlate with one another during exploratory analysis. After all, correlation-based feature selection reduces dimensions by keeping attributes that are correlated with class but uncorrelated with each other.[4] Accordingly, our group had Weka run correlation-based feature selection as attribute evaluator on the training set along with BestFirst search as search method. We also ran the same feature selection algorithm again in conjunction with GreedyStepWise as our search method so that we can compare the performances of the two different search methods. The results were comparable in that they were both able to filter out a large chunk of "unworthy" features to produce a list of 47 attributes that gave the readers the most amount of information about the data as a whole. In effect, correlation-based feature selection method shrunk the number of the dimensionality of the set by 92%.

Nonetheless, we wanted to reduce the list further so that, in the worst case, our group can construct a model using a subset of at most five features with desirable predictability by brute force. Certainly, we could have used RelifF on the list as Maurer and his team[5] have done in their research. However, ReliefF is inept at discriminating redundant attributes. As our group suspected that the list might still contain inter-correlated features, we searched for another feature selection method, which led us to employ Principal Component Analysis on our result.

Principal Component Analysis (PCA) seemed promising since the result from the previous feature selection method sieved all the attributes whose value ranges surpassed the normal bounds of [-1, 1] out of the data. That said, having normalized values before running PCA was critical since values that are high in magnitude can easily influence the outcome. We ran PCA and Ranker search method conjointly since Weka only allows its users to use Ranker search method, which ranks attributes by their individual evaluations, when they want to apply the analysis on their data.[6] Weka, then, enumerated 22 principal components, which equates to 55% reduction in dimensionality. Afterward, our team discontinued pruning components as a list of 21 features appeared to be sufficiently narrow. Furthermore, arrays of attributes that we generated through further feature selection on the data failed to match our expectation. In summation, our group cut a haphazard collection of 563 features into a list of 22 valuable components.

Model Construction, Selection and Evaluation

From our final list, we decided to build a model with the first five principal components. These were: "tBodyAcc-max()x," "tBodyAcc-arCoeff()-Z,4," "tBodyAcc-correlation()-X,Y," "tGravityAcc-mean()-X," and "tGravityAcc-mean()-Y (Table 1)." Figure 1 is a time series graph that show change in magnitude of each attribute with respect to the change in time. Interestingly, they are attributes that came either from the accelerometer data or transformations of the accelerometer data; gyroscope was not necessary. Then, with the said five features selected in addition to our classifier - "activity" - we opened classifier tab and built three models using three different classifiers, namely Random Forest, Bayesian Network, and Naive Bayes.

Since random forest is perhaps the most frequently used classifier for many research papers and alike, our team built a model using random forest machine learning algorithm as a benchmark to compare the results we would receive from other classifiers.

We reasoned that using Bayesian classifiers would further pin down the correlations between the features. Therefore, we built a model using Naive Bayes which assumes that the components are conditionally independent and another model with Bayesian Network which

learns the dependence at the construction time.[7] All in all, they have all returned acceptable results. Our analysis and evaluation of each model is shown below (Table 2).

Feature	Definition
tBodyAcc-max()-X	The largest body acceleration value along x axis over time
tBodyAcc-arCoeff()-Z, 4	Autoregression coefficients with Burg order equal to 4 for body acceleration over time
tBodyAcc-correlation()-X, Y	Correlation coefficient between x and y axis in light of body acceleration
tGravityAcc-mean()-X	Mean value of gravity acceleration along x axis
tGravityAcc-mean()-Y	Mean value of gravity acceleration along y axis

Table 1. Definition of first five principal features

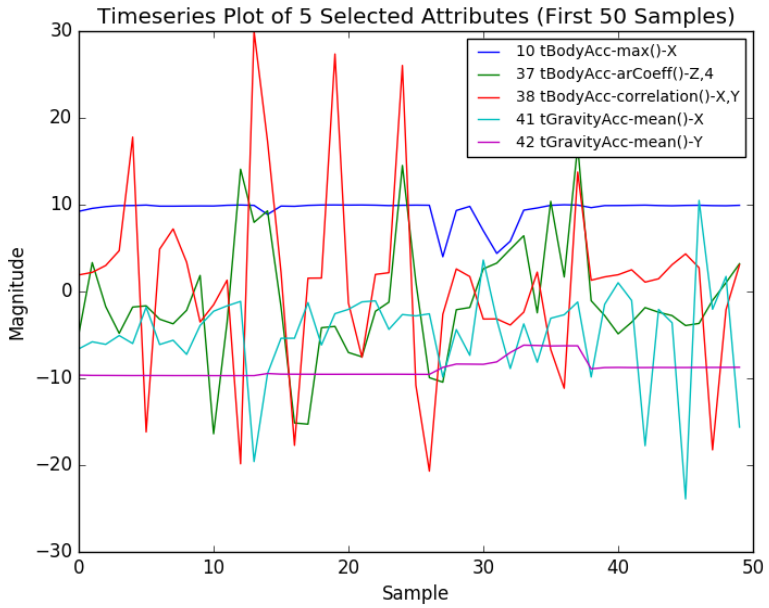

Figure 1. Time series plot that displays change in magnitude for each of five selected attributes.

Classifier	Correctly Classified	Incorrectly Classified	Accuracy
Random Forest (Bench Mark)	5827	0	100%
Naive Bayes	4907	920	84.2114%
Bayes Network	5257	570	90.218%

Table 2. Comparison of each model

Results and Analysis:

Since Bayes Network performed better with respect to accuracy compared with Naive Bayes, our group chose the model constructed through Bayes Network as our main model.

Then, using Weka, we re-evaluated the model with the test set we have created in R. The readers should note that our team test the model only once to ensure that the research stays unbiased (Table 3 & 4).

Correctly Classified Instances	1343 (88.0656%)
Incorrectly Classified Instances	182
Kappa Statistics	0.8563
Mean Absolute Error	0.0522
Root Mean Squared Error	0.1736
Total Number of Instances	1525

Table 3. Bayesian Network Model Test Result

Standing	Sitting	Laying	Walk	Walkdown	Walkup
992	98	0	0	0	0
162	842	0	0	0	0
0	1	1132	0	0	0
0	0	0	849	30	83
0	0	0	30	737	16
0	0	0	97	53	704

Table 4. Confusion Matrix

Considering the fact that random forest model can classify the data with 93.377% accuracy (+5% discrepancy in comparison with Bayes Network), our model produces a fairly satisfactory result.

Conclusion:

For this research, we used correlation-based feature selection and principal component analysis to reduce the dimensionality of the data set. We initially started the study with 561 attributes but were able cut that number by 96% in the end to come up with an array of 22 noteworthy features that could best classify 6 human activities. Then, we chose the five most notable components out of the said list of 22: “tBodyAcc-max()x,” “tBodyAcc-arCoeff()-Z,4,” “tBodyAcc-correlation()-X,Y,” “tGravityAcc-mean()-X,” and “tGravityAcc-mean()-Y.” With this set of features, we attained the accuracy of 93% with random forest which we used as a benchmark and 88% with Bayes Network.

A model that can recognize human activities with accuracy is of a great use for smartphone manufacturers, app developers and such. However, that accuracy is contingent on the amounts of data. Therefore, it is crucial to create a predictive model whose accuracy stays consistent no matter how large the dimensionality is. For example, even though our data is relatively small, we noticed that varying size of test set can slight decrease the accuracy of all

three models. Thus, for our future work, we would like to investigate if we can mitigate the problem.

References:

1. UCI Human Activity Recognition Dataset
   Link:
   https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

2. A table of past studies on Human Activity Recognition
   Link: https://www.hindawi.com/journals/cmmm/2016/4073584/tab3/

   * Additional Paper:
   Feature Selection for Human Activity Recognition with iPhone Inertial Sensors
   Link: https://www.researchgate.net/profile/Nuno_Silva18/publication
256679456_Features_Selection_for_Human_Activity_Recognition_with_iPhone_Inertial_Sensors/links/0deec52399bf54f71b000000.pdf?origin=publication_list

3 & 5. Physical Activities Monitoring Using Wearable Acceleration Sensors Attached to the Body
   Link: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130851

4. Explanation of Correlation-based Feature Selection
   Link: https://en.wikipedia.org/wiki/Feature_selection#Correlation_feature_selection

6. Explanation of Ranker Search Method and Weka's implementation of the algorithm.
   Link: http://weka.sourceforge.net/doc.dev/weka/attributeSelection/Ranker.html

7. Data Mining: Concepts and Techniques 3rd Edition.