REAL-TIME CLASSIFICATION OF ROAD CONDITIONS

An Undergraduate Research Scholars Thesis

by

SCOTT WEAVER

Submitted to the Undergraduate Research Scholars program
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Research Advisor: Dr. Dylan Shell

May 2016

Major: Computer Engineering

TABLE OF CONTENTS

ABSTRACT

Real-Time Classification of Road Conditions

Scott Weaver
Department of Computer Science and Engineering
Texas A&M University

Research Advisor: Dr. Dylan Shell
Department of Computer Science and Engineering

Common navigation algorithms like A* or D* Lite rely on costs to determine an optimal path. Costs may incorporate distance, time, or energy consumption; however, they can include anything that affects travel along a path. Much research is done to improve planning algorithms based on a given cost, often without stating how to acquire that cost. Therefore, the focus of this research involves determining a method of accurately obtaining that cost in real-time by classifying environmental conditions. Specifically, this research employs K-Nearest Neighbor and Principal Component Analysis techniques to classify road conditions in order to determine the most informative parameters when measuring the cost of driving on those roads. This sensor-based classification approach may not only allow for improved automatic traction handling and path navigation, but also may be applied to any robotic system requiring real-time knowledge of environmental conditions.

ACKNOWLEDGMENTS

I would like to thank Dr. Dylan Shell for his time and support. His Artifical Intelligence teaching and research guidance has been paramount to the production of this thesis.

NOMENCLATURE

2D	Two Dimensional
3D	Three Dimensional
KNN	K-Nearest Neighbors
PCA	Principal Component Analysis
PFA	Principal Feature Analysis

CHAPTER I

INTRODUCTION

Motivating scenario

Imagine driving along a smooth path, which your GPS navigation system expects to be the fastest route. Recent construction has demolished the asphalt, or a winter storm has left a layer of black ice. In both cases, you must slow down your car in order to compensate for unexpected road conditions. If the car can correctly classify the condition of the road (asphalt, gravel, dirt, ice, etc.), then such conditions may be relayed back to the GPS system to serve as a warning for future drivers as to the cause of dangerous traffic conditions. Additionally, in the age of driverless cars, such road classification can improve traction handling. For instance, the robotic car may shift gears or slow down upon sensing unfavorable conditions like ice or loose gravel.

Objectives

This research explores a method of analyzing and classifying road conditions using only motion and audio sensor input. The purpose of this research is much broader than just roads; the same systematic approach of environment classification may be applied to any robotic system requiring real-time knowledge of environmental conditions.

The specific objectives are enumerated as follows:

1. Determine *if* audio and motion (vibration, rotation, etc.) sensor input can be used to distinguish between road conditions.
2. Determine which features (sensor data) are most useful when classifying road conditions.
3. Determine the accuracy of classification using different features.

Note: Objectives 2 and 3 are entirely dependent on the success of Objective 1.

Previous work

Much research attention has been paid to on-board video analysis. Video feeds can be used for real-time applications like obstacle detection, road sign classification, and classification of traffic conditions [5]. However, neither of these systems reveal anything about the physical condition of the roads.

It is also common to use inertial sensors on a vehicle to measure its vibration frequency response. These sensors monitor motion along the x-y-z axes to determine frequency and amplitude of suspension systems. This enables active monitoring of moving objects such as unmanned vehicles [7]; however, it does not draw conclusions as to the road conditions causing specific frequencies.

When compared with existing work, this research is novel for two reasons. First, it employs the use of *non-visual* sensors on moving vehicles for real-time classification. Second, the internal sensors are used to classify *external* conditions (roads), rather than simply measuring internal responses (suspension). This can be expanded to other moving object scenarios; for example, the same technique could be used to distniguish running from walking, assuming there is higher variation in acceleration while running.

Background

Classification algorithms are commonplace in the field of machine learning. From recognizing faces [11] and manufactured parts to determining road conditions, items may be classified based on unique attributes of those items. One common method of classification involves using the supervised learning technique K-Nearest Neighbors (KNN) to distinguish items based on their classification. Principal Component Analysis (PCA) and Principal Feature Analysis (PFA) can be used to reduce the dimensionality of the data by filtering out insignificant information.

This research utilizes the KNN-PCA/PFA approach because of the simplicity and accuracy of KNN, PCA's ability to filter out insignificant data, and PFA's ability to determine significant features.

Feature vectors

When classifying items, a classifier uses a set of identifying features. In the case of road classification, a feature may be the average acceleration along the z-direction, or the variance of the rotation about the x-axis. For n features, an n-dimensional feature vector is a point in n-dimensional space [6]. For example, in three dimensions the feature vector x is formed by features x_1, x_2, and x_3. This is represented mathematically by equation I.1, which is plotted in figure I.1.

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{I.1}$$

The accurately labeled feature vectors comprise the *training data* set. In the context of road classification, the training data contains feature vectors labeled with classes like "asphalt" or "gravel". Classifications may also be further broken down into speeds (30 mph vs. 60 mph) or any other distinguishing factor. The *test data* is the set of unlabeled feature vectors which are classified using KNN and the corresponding training data.

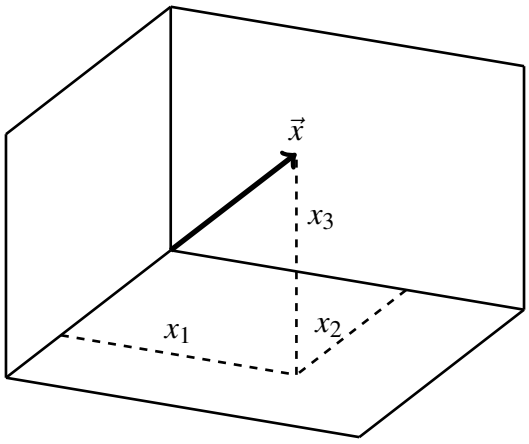


Fig. I.1.: 3-Dimensional Feature Vector

K-Nearest Neighbors

The K-Nearest Neighbors (KNN) approach is based on the concept that vectors (data points) of the same class should be closer in n-dimensional feature space, where n is the number of features [9].

As a result, for a test data point x of unknown class, the k nearest data points in the training data determine the class of x. In other words, if test data point x is plotted among the training data set, the most commonly occurring classification of the k nearest training data points determines the classification of x [13].

For example, figure I.2 shows that when using $k = 3$, the green dot is classified as a red triangle. However, when $k = 5$ the green dot is classified as a blue square. Thus, the value of k must be chosen carefully.

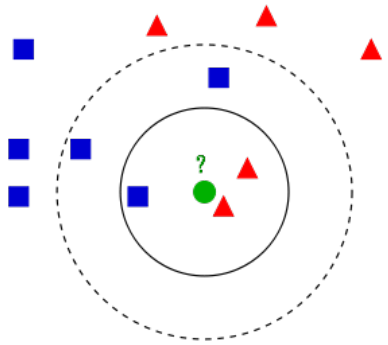


Fig. I.2.: KNN Classification for k=3 and k=5

Covariance matrix

Covariance is a measure of the strength of correlation between two features [12]. A positive correlation (X tends to increase as Y increases) corresponds to a positive covariance. The greater the correlation between two features means a larger (more non-zero) covariance. If X and Y are two features, then equation I.2 shows the covariance between X and Y, where N is the number of features (dimensions) and $\bar{x}$, $\bar{y}$ are the respective means.

$$cov(X,Y) = \sum_{i=1}^{N} \frac{(x_i - \bar{x})(y_i - \bar{y})}{N} \tag{I.2}$$

If $Y = X$, then the covariance is simply the variance of X. In other words, $cov(X,X) = var(X)$.

A covariance matrix is a symmetrical matrix which describes the covariance between all features. The element in the i, j position is the covariance between the i^{th} and j^{th} feature; element i, i describes the variance of feature i. The main diagonal of the covariance matrix is the set of elements such that $j = i$. For example, the main diagonal of the matrix in equation I.3 is shown in red. Element x_{11} is the variance of feature 11, element x_{22} is the variance of feature 22, etc.

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \tag{I.3}$$

Eigenalues and principal components

A set of eigenvectors and eigenvalues can be extracted from a covariance matrix. Let C be a covariance matrix such that $C = VDV^{-1}$, where D is a diagonal matrix and V forms the basis of a new orthagonal coordinate system [14]. Then the column vectors of matrix V comprise the set of eigenvectors, and the main diagonal of matrix D contains the set of corresponding eigenvalues.

Eigenvectors of a covariance matrix are important because they point in the directions that maximize feature variance. The eigenvalues describe the magnitude of each corresponding eigenvector. These eigenvector/eigenvalue pairs are the principal components of the covariance matrix.

Figure I.3 depicts the principal components of a two-dimensional data set. The first principal component (the eigenvector with the largest eigenvalue, shown in blue) points in the direction with the highest feature variance; the second principal component (in magenta) points in the direction orthogonal to the first principal component with the second greatest variance.

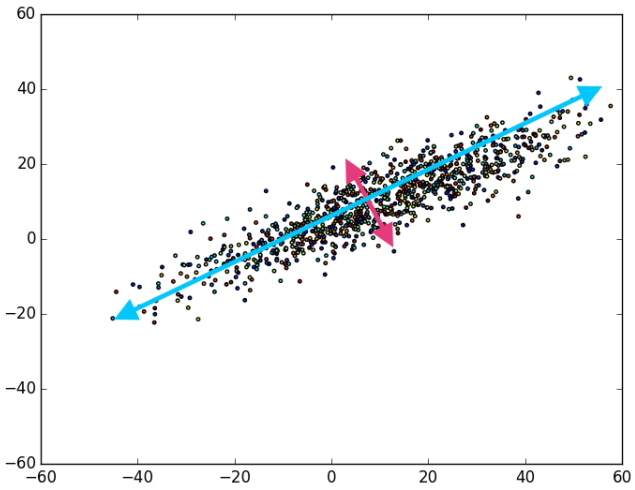


Fig. I.3.: 1st and 2nd Principle Components in 2D Space

Principal Component Analysis

It is common for dependent features to be very highly correlated. This means that only a subset of the original principal components are necessary to show a distinction between classifications. Principal Component Analysis (PCA) is a method which disgards the principal components with the smallest eigenvalues because they explain the least amount of variance.

The reduced coordinate system formed by the most significant principal components can then be used to classify the simpler data without losing much accuracy. Therefore, PCA is a technique commonly used to reduce the dimensionality of the classification space prior to performing KNN [10].

Figure I.4 depicts how a feature vector classified in three dimensional space can be projected onto a two dimensional plane of best fit. The best fit plane, directed along the first and second principal components, is chosen because it explains the maximum amount of variance in two dimensions. When converting from three dimensions to two, the data along the third principal component (which is orthogonal to the first two) is disgarded since it is least significant.

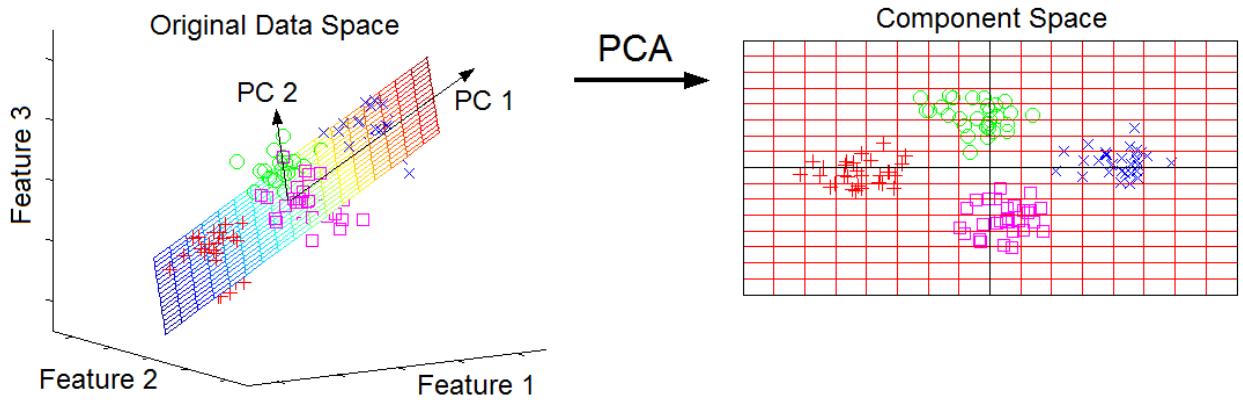


Fig. I.4.: PCA Dimension Reduction from 3D to 2D Space

Principal Feature Analysis

Although PCA can reduce the data by eliminating the least significant principal components, this doesn't help identify the importance of individual features. Therefore, Principal Feature Analysis (PFA) selects the features (rather than principal components) that explain the most amount of variance. This way, the features can be ranked by their individual contribution to the total variance, thus giving an indicator of feature significance.

CHAPTER II

METHODS

Approach overview

The following steps outline the approach used to develop a system that classifies road conditions:

1. Initial data collection
   (a) Raw sensor data from motion and audio sensors were collected via an Android device while driving on paved and unpaved roads. [1]
   (b) Upon collection, each data point was labeled with its known class. For example, a data point was labeled with "paved_30" if collected on a paved road while driving 30 mph.
2. Data analysis
   (a) Principal Component Analysis (PCA) was applied to the collected data to analyze principal component variance.
   (b) Principal Feature Analysis (PFA), a modified version of PCA, was applied to the data to analyze principal feature variance.
   (c) KNN was applied to the data before and after PCA and PFA to discover the most relevant features and principal components for accurate classification.
   (d) KNN was applied to each feature to determine classification accuracy of individual features and sensors.
3. Training data testing
   (a) Only the data features determined to be relevant by PFA and single-feature classification were collected from the sensors in a modified version of the Android application.
   (b) This application was employed to dynamically classify the test data using KNN.

[1] Data sets were collected in a 2001 Toyota Rav4, using a Samsung Galaxy S6.

Data collection

When collecting data, the Android application continuously monitors motion and audio sensors. Ideally, all sensors would be syncronized for simultaneous data measurements. However, due to practical limitations of the Android device, each sensor polls the environment with a specific frequency (44100 Hz for audio) [2].

Position sensors, like Android's magnetic field sensor, are avoided since this research aims at classifying environment data independent of location. Similarly, light and camera sensors are avoided to reduce dependency on sensor placement. The motion and audio sensors used are listed in table II.1.

Table II.1

Sensor	Units	Details
Microphone	volts (V) mapped to a 16 bit range [-32768, 32767]	Collects audio samples for processing using Android's AudioRecord API [1]
Gyroscope	radians per second (rad/s)	Measures rate of rotation around each axis (x, y, z)
Accelerometer	meters per second (m/s)	Measures acceleration along each axis (x, y, z), including acceleration due to gravity

The features are measured about a coordinate system relative to the Android device, shown in figure II.1 [3]. During data collection, the device was placed horizontally, face-up, with the top facing forward.

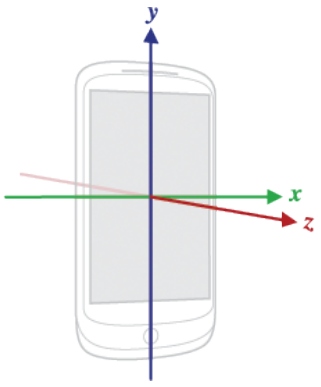


Fig. II.1.: Android sensor coordinate system

Feature representation

Each feature vector is not formed by raw sensor measurements, but instead by the mean or standard deviation of that data over a time interval. For instance, one feature could be the average acceleration along the x-axis over an interval of 1 second. For n features, each vector refers to a point in n-dimensional feature space. For multiple trials of the same road classification, these vectors should form a cluster that is distinct from clusters made by other classifications.

Central tendency and statistical dispersion

The mean is the most common measure of central tendency, and is calculated by finding the average of a set of data readings. The mean, $\bar{x}$, of N raw data points, x_i, is calculated using equation II.1.

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{II.1}$$

Statistical dispersion describes the spread of data. Standard deviation is a common measure of dispersion, which explains the average degree to which the data deviates from the mean [4]. Standard deviation is calculated using equation II.2.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{II.2}$$

Standard deviation, σ, is utilized rather than variance, σ^2, because it preserves the same units as the original data measurements.

Data normalization

Since each Android sensor collects data in different units, some features hold significantly more weight over others. For example, there is more spread over audio data, with values on the order of 10^4, than there is over the spread of any other sensor's values, on the order of 10^0 or 10^1.

By normalizing all features from 0 to 1, each feature has equal contribution to the overal covariance [8]. Then, significant sensors with inherently smaller units will not appear to be insignificant.

Equation II.3 shows how every measurement, x_i, in each feature is scaled between 0 and 1 by using the minimum and maximum values of that feature.

$$normalize(x_i) = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{II.3}$$

Data analysis

After the collection, calculation, and normalization of all features, the data set was split at random; one half of the data set became the training data, and the other half became the test data. Each data set (training and test) comprised a few dozen feature vectors, where each feature is either the mean or standard deviation of thousands of raw measurements.

Base classification

KNN was applied to classify each test data vector with respect to the training data set. The percentage of correctly predicted test data classifications established the base classification accuracy for the classifer. For example, if there were 100 test data points, and 95 of their classifications were predicted correctly, then the test data set had a 95% classification accuracy.

The k-value used in all KNN classifications depended on the data being classified. The k-value (number of neighbors) was chosen to be the lowest possible value that maximized accuracy. For instance, if the classification accuracy was highest at 95% using $k = 3$, 4, or 5 neighbors, then KNN was always performed using $k = 3$ for that data set.

Finding feature significance based upon both the variance and classification accuracy can paint a more complete picture as to which features are important when classifying road conditions.

Classification accuracy by variance

The base classification was performed using the complete data, with all n features and principal components ($n = 14$ in this research, for the mean and standard deviation of 7 different sensor

measurements). PCA was then conducted to determine how much variance is explained by each individual *principal component*. For instance, component 1 explains the most variance, and component 14 explains the least. Similarly, PFA was conducted to determine each *feature*'s contribution to the total variance.

Classification was then repeated using KNN, each time removing the principal component that explained the least variance, until only the most significant principal component remained. In other words, classification accuracy first was calculated using all 14 principal components, then using only the 13, 12, 11, ... , 2, 1 principal component(s) with the most variance.

The process of iterative KNN was also conducted using PFA, thus determining the correlation between classification accuracy and *feature* (rather than *principal component*) variance.

Classification accuracy breakdown by feature

Although ideally the classification accuracy contributed by each feature should correspond to the amount of variance it explains, this is not always the case. A feature could explain a lot of variance among the overall test data, but could appear to be insignifcant when it is the sole feature used for classification. Therefore, it is important to consider the classification accuracy using one feature at a time.

Similarly, it is important to consider the classification accuracy contributed by entire sensors. For example, let's assume the classifier only has access to the accelerometer; then how accurate can the training data predict the test data using only features collected by the accelerometer? What about only using the gyroscope, or microphone?

For the above reasons, KNN was repeated for each individual feature. Then, KNN was performed for all features of each sensor.

CHAPTER III

RESULTS

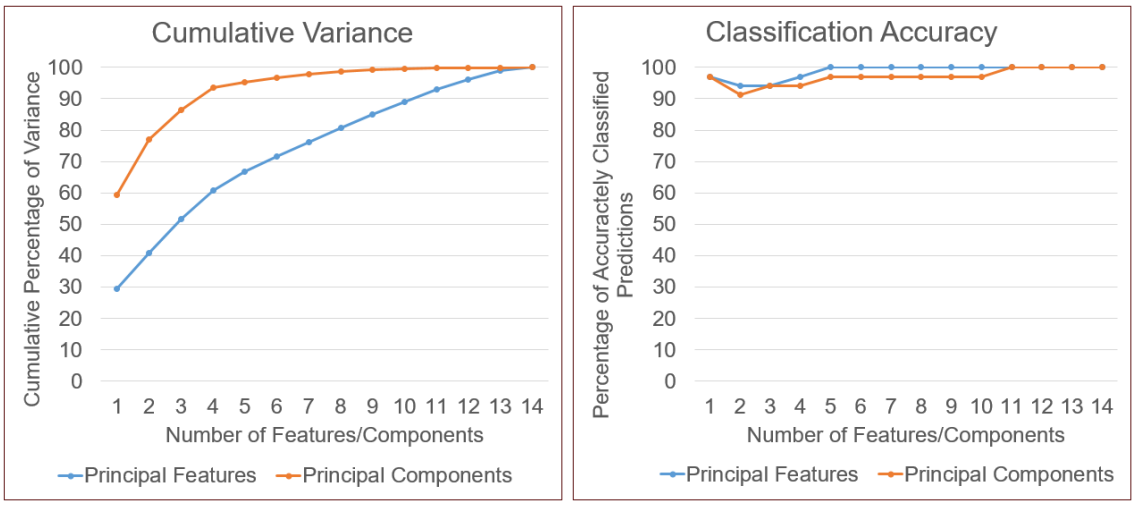
The first observation is that motion and audio sensor input can distinctly differentiate between paved (asphalt, concrete) and unpaved (dirt, gravel) roads. In fact, the complete set of data can classify road conditions with 100% accuracy. The same sensors can also accurately classify velocities (30 mph vs. 60 mph), but with different significant features.

Classification accuracy by variance

The graph in figure III.1a illustrates the distribution of variance among each of the 14 features and principal components. The first principal component is shown to explain 60% of the variance in the entire data set; only the first 4 principal components are required to explain almost 100% of the cumulative variance.

Since the principal components maximize variance, it makes sense that the principal components always explain more variance than do the corresponding number of features. Therefore, it is surprising how even though principal components from PCA always explain more variance, features from PFA always predict higher classification accuracy.

The graph in figure III.1b also depicts how only 3 to 4 features or principal components with the most variance are necessary to obtain near 100% classification. Classifying velocities produced similar results, not shown here.



(a) Cumulative Variance

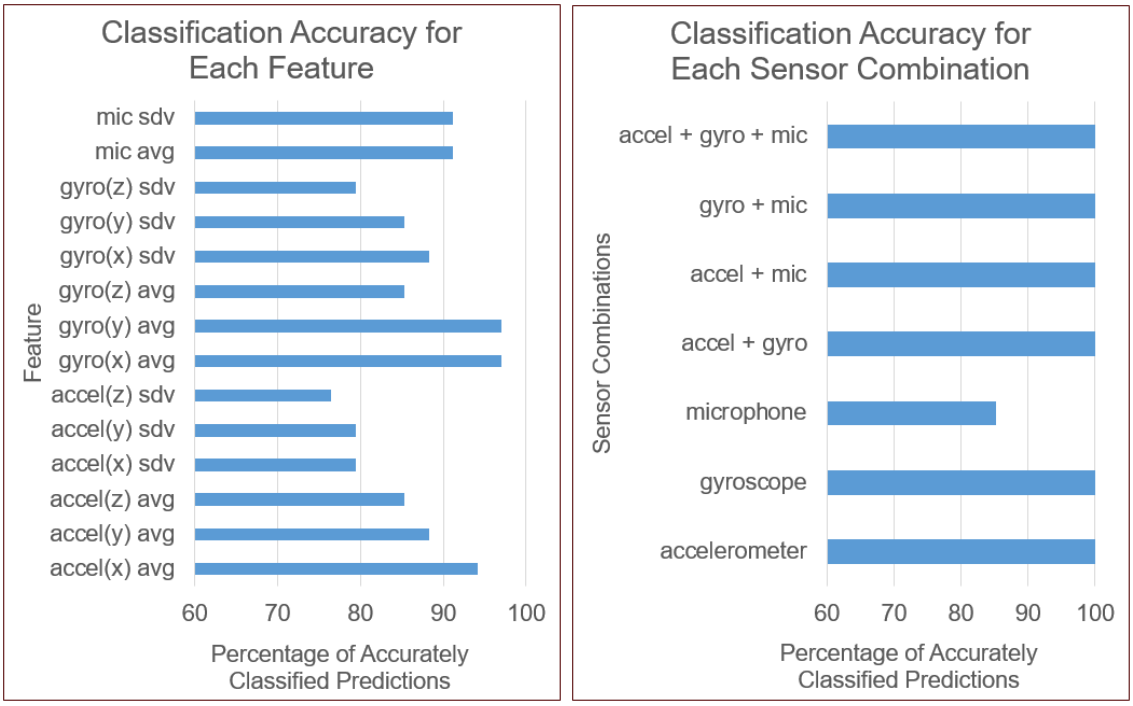
(b) Classification Accuracy

Fig. III.1.: Classifying Road Conditions: Paved vs Unpaved (at 30 mph)

Classification accuracy breakdown by feature

When classifying road conditions one feature at a time, shown in figure III.2a, multiple observations become apparent. First of all, higher feature variance usually (but not always) corresponds to higher single-feature accuracy. In other words, the more variance a feature explains, the more likely that feature is to accurately classify road conditions.

The individual classification of each feature also demonstrates how the average of each feature is more useful than the standard deviation, without exception. Additionally, measurements about the x- and y-axes are more useful than those about the z-axis.



(a) Classification Accuracy by Feature

(b) Classification Accuracy by Sensor

Fig. III.2.: Classifying Road Conditions: Paved vs Unpaved (at 30 mph)

CHAPTER IV

CONCLUSION

The results demonstrate conclusive findings for each of the three research objectives:

1. Road conditions (and velocities) can be classified using only motion and audio sensor input.
2. The average acceleration and rotation about the x- and y-axes are most significant when accurately classifiying road conditions.
3. Road conditions can be classified at near 100% accuracy. Only 3 to 4 features with high variance are necessary for near 100% classification accuracy.

Real-time classification and future work

Though the classification analysis determined that road conditions can be classified, it is important to consider whether or not this method can be applied to a real-time program. Therefore, the classification of road conditions were predicted in real-time by an Android application. Initial tests proved to be successful; however, additional work is necessary to analyze the accuracy in which real-time testing predicts classification.

REFERENCES

[1] Android. *AudioRecord*, (2015). [Online]. Available: http://developer.android.com/reference/android/media/AudioRecord.html.

[2] Android. *SensorEvent*, (2015). [Online]. Available: http://developer.android.com/reference/android/hardware/SensorEvent.html.

[3] Android. *Sensors Overview*, (2015). [Online]. Available: http://developer.android.com/guide/topics/sensors/sensors_overview.html.

[4] George Boeree. *Descriptive Statistics*, (2005). [Online]. Available: http://webspace.ship.edu/cgboer/descstats.html.

[5] C. Carincotte C. Parisot, J. Meessen and X. Desurmont. Real-time road traffic classification using on-board bus video camera. *Proc. IEEE Conference on Intelligent Transportation Systems*, (189-196), 2008.

[6] Richard Duda. *Feature Vectors*, (2008). [Online]. Available: http://www.cs.princeton.edu/courses/archive/fall08/cos436/Duda/PR_model/f_vecs.htm/.

[7] V. Bashev E. Iontchev, R. Miletiev and I. Simeonov. Study of the dynamic response and status of the vehicle suspension elements. *International Journal of New Computer Architectures and their Applications*, (45-51), 2013.

[8] Sergey Feldmen. *Should you apply PCA to your data?*, (2012). [Online]. Available: http://blog.explainmydata.com/2012/07/should-you-apply-pca-to-your-data.html.

[9] Roger Jang. *Data Clustering and Pattern Recognition*. CS Department, National Taiwan University, 2005.

[10] Matthias Scholz. *PCA - Principal Component Analysis*, (2006). [Online]. Available: http://www.nlpca.org/pca_principal_component_analysis.html.

[11] M. Turk and A. Pentland. Face recognition using eigenfaces. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (586-591), 1991.

[12] Eric Weisstein. *Covariance*, (2016). [Online]. Available: http://mathworld.wolfram.com/Covariance.html.

[13] Wikipedia. *k-nearest neighbors algorithm*, (2015). [Online]. Available: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.

[14] Wikipedia. *Eigendecomposition of a matrix*, (2016). [Online]. Available: https://en.wikipedia.org/wiki/Eigendecomposition_of_a_matrix.