Sensing and Classifying Roadway Obstacles: The Street Bump Anomaly Detection and Decision Support System

Theodora S. Brisimi, Setareh Ariafar, Yue Zhang, Christos G. Cassandras, and Ioannis Ch. Paschalidis

Abstract—We develop an anomaly detection and decision support system based on data collected through the Street Bump smartphone application. The system is capable of effectively classifying roadway obstacles into predefined categories using machine learning algorithms, as well as identifying actionable ones in need of immediate attention based on a proposed “anomaly index.” We introduce appropriate regularization to the classification algorithms we employ, which has the effect of utilizing a sparse set of relevant features to perform the classification. Further, our novel “anomaly index” allows us to prioritize among actionable obstacles. Results on an actual data set provided by the City of Boston illustrate the feasibility and effectiveness of our system in practice.

Index Terms—Classification, anomaly detection, machine learning, smart cities.

I. INTRODUCTION

According to the American Association of State Highway and Transportation Officials, as much as 50% of US roads and highways are in bad condition, thus, raising the likelihood of accidents. The Massachusetts State Transportation Department [1] received about 1,700 pothole complaints over the first quarter of 2014 and spent more than $800,000 filling them, while the City of Boston filled more than 10,000 potholes. In order to reduce the cost and automate the process of detecting road obstacles, the City has developed a smartphone application called Street Bump. This application makes use of the accelerometer and GPS capabilities of a typical smartphone to identify and locate “bumps” as a vehicle (where the application-carrying smartphone is present) drives through a roadway network. The term “bump” is generically used to describe a variety of obstacles, such as potholes, sunk castings, flat castings, utility patches, catch basins, train tracks, and speed bumps. Clearly, this enables all driving citizens equipped with Street Bump to contribute to a massive and continuous data collection process.

The purpose of this paper is to develop an anomaly detection and decision support system driven by the collected Street Bump data. The raw data do not differentiate between “actionable” and “non-actionable” bumps: the former include potholes and sunk castings which are caused by nature or accident and require prompt repair, whereas the latter are either expected or known obstacles (e.g., train tracks and speed bumps) which do not require immediate attention. We view the actionable bumps as anomalies which are to be differentiated from the rest. Further, we seek to develop a complete decision support system capable of accurately classifying all detected roadway obstacles into predefined categories (such as actionable and non-actionable) and of quantifying the severity of the actionable ones so as to assist the City in prioritizing them and dispatching repair crews in a timely and cost-effective manner.

Our approach is based on extracting features from the data collected, i.e., functions of the data organized into a number of signals (or time series) over time windows associated with a specific bump. We use a standard supervised classification approach assuming we have at our disposal a “labeled” training set. Classifiers trained in this way can subsequently be used to classify new “unlabeled” data.

Once the feature extraction process is completed, we follow two complementary approaches. The first approach is based on formulating a basic binary classification problem to differentiate actionable from non-actionable bumps. We employ a subset of supervised learning algorithms that has shown to be effective in a number of other applications (e.g., prediction of heart-related hospitalizations [2]). These algorithms include Support Vector Machines (SVM), AdaBoost and Logistic regression [3]. In some instances, we introduce specific sparsity-inducing regularizations that have the effect of improving the classifier’s performance by identifying a subset of most relevant features.

The second approach is inspired by anomaly detection problems that appear in a variety of applications (see e.g., [4], [5], [6] and references therein). Anomaly detection methods typically attempt to model “normal” behavior and detect deviations from it – the anomalies. In our setting, we will take advantage of structural properties in the problem. We exploit the fact that most actionable obstacles are due to natural phenomena that produce random obstacle configurations, whereas non-actionable ones are human-made and possess a significant degree of regularity. For example, vibration data from a flat casting fit much better the signature of a harmonic oscillation than those from a pothole. In particular, we quantify this measure of regularity through an “anomaly index” and develop two methods for obtaining such indices which can also be combined into one. The first method is based on a Mean Squared Error (MSE) metric to measure the deviation of a bump signal from that of a simple harmonic oscillation. The second method relies on measuring the entropy [7] of a bump signal so that bumps with higher
to these three time series (one for each coordinate) as the *signatures* of the bump. The *x*-coordinate signature of an anomalous bump is shown in Fig. 1, where the horizontal axis represents time in seconds and the vertical axis measures the acceleration in the *x*-axis (i.e., driving) direction.

**Fig. 1.** (Left): *x*-coordinate signature of an anomalous (actionable) bump. (Right): *x*-coordinate signature of a flat casting (non-actionable).

### B. Feature construction for the decision support system

From the collected data, we seek to form a set of features that are informative, non-redundant and facilitate the subsequent learning steps. We divide each of the three *n*-long coordinate time series into a number of *K* bins of length $d = \lfloor n/K \rfloor$. Because the total number of samples is not the same for all bumps, we appropriately truncate each time series to enforce the same number of samples for each coordinate and each bump. For each bump, let $\mathbf{x} = (x_1, \ldots, x_{dK})$ and similarly $y, z$ denote the vector of samples for the $x, y, z$ coordinates, respectively. Let $\mathbf{x}^{(k)} = (x_{d(k-1)+1}, \ldots, x_{dk})$, and similarly $y^{(k)}, z^{(k)}$, $k = 1, \ldots, K$, denote the vector of samples in the $k^{th}$ bin of the $x, y, z$ coordinate signatures, respectively. Last, we define the operators $M[\cdot]$, $R[\cdot]$ and $\sigma[\cdot]$ as the average, range and standard deviation of the elements of a vector. Then, the feature set is constructed as follows:

- **Basic bump features**: From the set of attributes defined above, we retain the latitude and longitude of the bump as well as the speed of the vehicle.

- **Bump distributional features**: From the *x*-coordinate signature, we calculate $M[x], \sigma[x], R[x]$ and $M[x^{(k)}], \sigma[x^{(k)}], R[x^{(k)}]$, where $k = 1, \ldots, K$. As an additional feature we take

$$D_\mathbf{x} = |\arg \max_i x_i - \arg \min_i x_i|.$$  

Next, we quantize the range of the elements of $\mathbf{x}$ into $B$ bins, and compute the empirical measure $h^{B}_\mathbf{x} = (h_\mathbf{x}(1), \ldots, h_\mathbf{x}(B))$ where

$$h_\mathbf{x}(b) = \frac{\sum_{i=1}^{dK} \mathbb{1}_{\{x_i \in B^b\}}}{dK}, \quad b = 1, \ldots, B,$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function. Last, we create a mapping: $x \rightarrow x_\mu = (M(x^{(1)}), \ldots, M(x^{(k)}))$, and we include $M[x_\mu], \sigma[x_\mu], R[x_\mu]$ into the feature set. The corresponding features are also calculated for the $y$- and $z$-coordinate signatures.

- **Temporal dependency features**: This set of features captures the dependencies between the signals in each bin for each coordinate. For the *x*-coordinate signature, we add to the feature set the covariance of the signals in consecutive bins i.e., $cov(x^{(k)}_t, x^{(k+1)}_t), \forall k$. We also account for covariances between bins further away by
including the following two types of features: (a) the maximum covariance
\[ C_{xx} = \max_{d+1 \leq j \leq K-1} \text{cov}(x^{(1)}, (x_j, \ldots, x_{j+d})) \]
and (b) the corresponding time lag
\[ L_{xx} = \arg \max_{d+1 \leq j \leq K-1} \text{cov}(x^{(1)}, (x_j, \ldots, x_{j+d})) - d + 1. \]
Correspondingly, we also calculate similar features for the other coordinates.

- Cross-coordinate dependency features: This set of features captures the dependencies between the three signatures. We include \( \text{cov}(x^{(k)}, y^{(k)}), \text{cov}(x^{(k)}, z^{(k)}), \text{cov}(y^{(k)}, z^{(k)}) \), \( \forall k \), and \( C_{xy}, L_{xy}, C_{yx}, L_{yx}, C_{yz} \) and \( L_{yz} \), where definitions are similar as above. For example,
\[ C_{xy} = \max_{d+1 \leq j \leq K-1} \text{cov}(x^{(1)}, (y_j, \ldots, y_{j+d})). \]

Let us denote by \( f^{(i)} \) the feature vector we have formed for each bump \( i \) as described above, where \( i = 1, \ldots, N \), \( N \) being the total number of bumps in the dataset. The dimensionality of the feature vector is denoted by \( D \). To avoid significant mismatch in the ranges of each element of \( f^{(i)} \), we normalize appropriately so that each element is in the \([0, 1]\) range.

C. Feature construction for the anomaly detection system

For the anomaly detection system, we follow a different path to pre-process the collected data. Same as before, we work with the vectors \( x, y \) and \( z \). By inspection of the signatures of the bumps (see Fig. 2), one cannot confidently identify the type of bump (i.e., actionable versus non-actionable). In an effort to enhance the differences between the two categories of bumps, we define a differential signal, to which we refer as the “\( \Delta \)-Signature Filter”, which evaluates the difference between the values of the signature in consecutive time steps and either accumulates them if there is no change in their sign, or it resets its value to the latest difference otherwise. Thus, for any signal \( \xi(k) \), \( k = 1, 2, \ldots \), we define:
\[
\delta(k) = \xi(k) - \xi(k-1),
\]
\[
\Delta(k) = \begin{cases} 
\Delta(k-1) + \delta(k) & \text{if } \delta(k)\delta(k-1) > 0 \\
\delta(k) & \text{if } \delta(k)\delta(k-1) \leq 0 \quad (1) \\
0 & \text{if } \delta(k) < c 
\end{cases}
\]

Intuitively, the sequence \( \delta(k) \) captures the vibration in the signature and \( \Delta(k) \) attempts to capture the trend in the signal. If the increments \( \delta(k) \) of the signal are consistently positive or negative over some period, this will result in a large value of \( \Delta(k) \), otherwise \( \Delta(k) \) resets itself. Noise in the signature appears as random oscillations. To suppress noise, we pass \( \Delta(k) \) through a high-pass filter that sets its value to zero if it is below a threshold \( c \) as seen in (1); in our case, we have used \( c = 0.4 \). Clearly, more obvious patterns are revealed through the \( \Delta \)-filtered signature; see Fig. 2. To verify this quantitatively, we conducted two more experiments: (a) We use the original signature and the \( \Delta \)-filtered signature of a bump as feature vectors given to a binary classification system using Support Vector Machines (SVMs); see next section for details. We found that the latter yields superior performance, roughly doubling the detection rate of actionable bumps. (b) We use the Fourier transforms of the signature and the \( \Delta \)-filtered signature of the bumps as feature vectors, which leads again to better results. We can therefore conclude that filtered signatures can enhance classification performance.

III. METHODOLOGY- DECISION SUPPORT AND ANOMALY DETECTION SYSTEM

In this section, we describe the methods that comprise the decision support and anomaly detection system. We aim at distinguishing between the actionable and the non-actionable (anomalous) bumps. We use two approaches: (a) a supervised binary classification approach, which classifies bumps as actionable or non-actionable, and (b) an anomaly detection approach which attempts to identify bumps that are significantly different from the rest. In the first approach, we employ some of the established machine learning algorithms, namely, Support Vector Machines (SVMs), logistic regression and AdaBoost with stumps as the weak learner. Because of the limited size of our dataset (labeled bump samples are expensive to gather) and to avoid over-fitting, we seek to limit the number of features that are used to make the classification decision, which leads us to introduce sparsity-inducing regularizers. This results into an \( \ell_1 \)-regularized logistic regression and a sparse SVM. By combining results from the methods above, we build a unified prioritized decision support system. In the second approach, we define the notion of a normal bump signal in two different ways – a normal signal has a sinusoidal pattern or a normal signal...
has an expected range of amplitude – and we measure how different a test bump is from the normal pattern.

A. Soft-Margin SVM and Sparse Soft-Margin SVM

SVMs seek to find a separating hyperplane $\mathbf{w}^T \mathbf{f}(i) + b$, where $\mathbf{f}(i)$ is the $D$-dimensional feature vector of the $i$th bump, between samples (bumps) of different classes [8]. The solution to the SVM problem is the hyperplane that maximizes the margin between the two classes, i.e., the distance from the decision surface to the closest data points. Oftentimes in real-world problems, the sample points are not linearly separable. For this reason, the original space is mapped through a kernel function into a higher dimensional space, where presumably linear separation can be achieved [8], [9]. What is more, we prefer a solution that better separates the majority of the data while ignoring a few misclassified samples (soft-margin SVM) [9]. This is achieved through a penalty term in the objective function of the SVM problem with a parameter $C$ that controls the relative weighting between the goal of making the margin large and ensuring that most examples have been correctly classified. The parameters of the kernel as well as the regularization parameter $C$ (cf. (2)) are selected through cross-validation.

In an effort to limit the number of features used by the classifier, we introduce a sparse soft margin SVM. Specifically, we impose two penalties in the SVM objective function: the aforementioned regularization penalty with the $C$ parameter, and an $\ell_1$-norm penalty $\sum_{i=1}^{D} |w_i|$ for the vector of coefficients $\mathbf{w}$ that define the SVM hyperplane. The optimization problem is formulated as follows, where $y^i \in \{-1, 1\}$ are labels (non-actionable or actionable, respectively) associated with the $i$th bump, $i = 1, \ldots, N$,

$$
\min_{\mathbf{w}, \xi, b, \mathbf{z}} \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N} \xi_i + P \sum_{i=1}^{D} z_i
\quad \text{s.t.} \quad y^i(\mathbf{w}^T \mathbf{f}(i) + b) \geq 1 - \xi_i, \quad \forall i,
\quad \xi_i \geq 0, \quad \forall i,
\quad z_i \geq w_i, \quad z_i \geq -w_i, \quad t = 1, \ldots, D.
$$

(2)

In this formulation, $\xi$’s are slack variables that allow for some bumps to be misclassified and $P$ is a parameter that controls sparsity, i.e., the number of features which will end up with non-zero coefficients $w_i$.

B. Logistic Regression and $\ell_1$ Regularization

Logistic Regression [10] is a linear, fairly simple classifier, widely used in many classification applications. For the $i$th bump labeled as $y^i \in \{0, 1\}$ (non-actionable or actionable, respectively), and given the feature vector $\mathbf{f}(i)$, we model the posterior probability of the actionable class as a logistic function:

$$P(y^i = 1|\theta, \beta; \mathbf{f}(i)) = h_{\theta, \beta}(\mathbf{f}(i)) = \frac{1}{1 + \exp(-\theta^T \mathbf{f}(i) - \beta)}$$

where $(\theta, \beta)$ are model parameters. It follows that $P(y^i|\theta, \beta; \mathbf{f}(i)) = (h_{\theta, \beta}(\mathbf{f}(i)))^{y^i} (1 - h_{\theta, \beta}(\mathbf{f}(i)))^{1-y^i}$, and the log-likelihood $L(\theta, \beta)$ of the data set takes the form (assuming independence of the training data):

$$L(\theta, \beta) = \sum_{i=1}^{N} y^i \log(h_{\theta, \beta}(\mathbf{f}(i))) + (1 - y^i) \log(1-h_{\theta, \beta}(\mathbf{f}(i))).$$

The parameters $(\theta, \beta)$ are selected by maximizing the log-likelihood using a gradient method. For the test samples, decisions are made by thresholding the log-likelihood ratio of the actionable class over the non-actionable class.

In the $\ell_1$ regularized Logistic Regression [11], when maximizing the log-likelihood we impose in the objective function an extra penalty term proportional to $|\theta|$, which has the effect of “selecting” a sparse set of features. This prevents overfitting and leads to a lower complexity model.

C. AdaBoost with Stumps

Boosting is an ensemble supervised learning method that constructs a strong classifier as a linear combination of “simple” “weak” classifiers [12]. A weak learner is a classifier that has accuracy just slightly better than random guessing. A decision stump, that we use as the weak learner, makes a prediction based on the value of just a single input feature. AdaBoost keeps a distribution of weights for the training sample points. During each round a weak classifier is trained, which tries to focus on the “difficult” data points (the ones that have been misclassified by the previous weak classifier) and the weights are updated based on the misclassification error. In the end, AdaBoost combines the decisions of these weak classifiers using an optimally weighted majority vote. The number of iterations is selected through cross-validation.

D. Decision Support System for Prioritizing Anomalies

All methods outlined so far, classify a test bump by comparing a classifier function of its features $g(\mathbf{f}(i))$ to a threshold $c$. In particular, a bump is classified as actionable if $g(\mathbf{f}(i)) \geq c$. We can therefore use the distance $g(\mathbf{f}(i)) - c$ as a way to prioritize among actionable bumps; the higher the distance the more confident we are about the bump being actionable. Logistic regression in particular, provides explicitly the likelihood of a bump being actionable, which can naturally be used to order actionable bumps. Furthermore, it is possible to seek consensus among the three classifiers, thus, further reducing false alarm rates.

E. Sinusoidal Fitting and a Mean Squared Error Metric

The idea behind this method is that the $\Delta$-filtered signature of non-actionable bumps exhibits a regular pattern very similar to a sinusoidal function. Therefore, if we compare a sine (or cosine) function to the $\Delta$-filtered signature of a bump (see Fig. 2), we can calculate a Mean Squared Error (MSE) as a measure of how good the fit is. The MSE captures the degree of bump irregularity, and is used in defining an anomaly index, as discussed in Section IV.

We begin by identifying time intervals such that $\Delta(k) = 0$ over more than one contiguous sample points. By eliminating such intervals, since they contain no valuable information, we concentrate on a typical interval $[t_0, t_0 + T_d]$ over which the continuous signal $\Delta(t)$ (see Fig. 2) is such that $\Delta(t) \neq 0$ except possibly at a finite number of sampling points $t_i \in [t_0, t_0 + T_d], i = 0, 1, \ldots, N$, with $t_N = t_0 + T_d$. We compare this signal to a sinusoid $f(t) = A \sin(\omega t - \theta_0) + b$ where the parameters $A, \omega, \theta_0$, and $b$ are determined as follows.
Let \( z_j \) be the \( j \)th zero crossing of \( \Delta(t) \) in \([t_0, t_0 + T_d] \) and define \( \Delta_{\text{max}} = \max_{t \in \{z_j, z_{j+1}\}} (\Delta(t)) \), \( j = 0, 1, \ldots, n_2 \). Then, \( A = \frac{1}{n_2} \sum_{i=1}^{n_2} |\Delta_i^{\text{max}}| \). Letting \( \theta_{b} \) be the observed number of periods of \( \Delta(t) \) in \([t_0, t_0 + T_d] \), we set \( \omega = \frac{\theta_{b}}{n_2} \) and \( \theta_0 = \omega t_0 \). Finally, \( b = \frac{1}{N} \sum_{k=1}^{N} \Delta(k) \). Thus, 

\[
MSE = \frac{1}{N} \sum_{k=1}^{N} [f(t_k) - \Delta(t_k)]^2
\]

is the MSE measuring the proximity of \( \Delta(t) \) to \( f(t) = A \sin(\omega t - \theta_0) + b \). We expect that non-actionable bumps have a better fit, hence lower MSE. Therefore, bumps with larger MSE, are identified as anomalies (see Fig. 2).

**F. A Bump Entropy Metric**

This method is inspired by the observation that the amplitude of \( \Delta \)-filtered signatures of actionable and non-actionable bumps have different ranges (typically for the actionable ones the range is \([-2, 2]\) (see Fig. 2), while the range for non-actionable ones is \([-1, 1]\)). We use the information-theoretic concept of entropy to characterize the degree of irregularity of a \( \Delta \)-filtered signature. For simplicity, let us concentrate on the positive amplitude range of \( \Delta(t) \), i.e., \([0, \Delta_{\text{max}}]\) and partition this interval into subintervals \([u_{i-1}, u_i] \), \( i = 1, \ldots, n \), so that \( u_0 = 0 \) and \( u_n = \Delta_{\text{max}} \). We then define \( H^+ = -\sum_{i=1}^{n} p_i \log(p_i) \) to be the bump entropy of the positive \( \Delta \)-filtered signature, where \( p_i \) is the fraction of time during which \( \Delta(t) \in [u_{i-1}, u_i] \) (similarly for \( H^- \) pertaining to the non-positive amplitude range of \( \Delta(t) \)). Formally, the total time during which \( \Delta(t) \in [u_{i-1}, u_i] \) involves the inverse images \( \Delta^{-1}(u_i) \) so that \( p_i = \frac{1}{T_d} \sum_{j} |\Delta^{-1}(u_{j+1}) - \Delta^{-1}(u_j)| \), where \( T_d \) is the length of the time interval considered (we omit details). Further, in order to amplify the effect of extreme points in the signature, i.e., unusually large positive (in the vicinity of \( \Delta_{\text{max}} \)) or negative amplitudes, we modify 

\[
H = H^+ + H^- = -\sum_{i} \log(p_i)
\]

In this way, a small number of extreme amplitudes can substantially increase the bump entropy by decreasing the logarithmic term values.

**IV. PERFORMANCE EVALUATION**

**A. Supervised classification**

We evaluate the performance of the learning algorithms measuring two error rates. The **miss detection rate** corresponds to the ratio of the actionable bumps that we wrongly predicted to be non-actionable. The **false alarm rate** corresponds to the ratio of the non-actionable bumps that we wrongly predicted to be actionable. By changing the decision threshold when classifying the test samples, we produce multiple pairs of the two error rates. We report the performance by plotting the miss detection rate versus the false alarm rate. This produces what we will call (with a slight abuse of terminology) the **Receiver Operating Characteristic (ROC)** curve. Typically, one chooses a point on the ROC curve to operate on, depending on the relative cost assigned to false alarms and miss-detections.

**B. Anomaly detection**

The MSE metric in (3) and bump entropy in (4) both measure the degree of irregularity of a bump in different ways. By combining them, we define an **Anomaly Index AI**: 

\[
AI = \lambda(MSE) + (1 - \lambda)H
\]

associated with each bump, where \( \lambda \in [0, 1] \) is a weight selected to place more emphasis on the MSE or the entropy. Based on AI, we can then generate a list in descending order so that the first several entries of the list identify the bumps most likely to be actionable and requiring immediate attention.

**V. EXPERIMENTAL RESULTS**

Our results are based on data collected by the City of Boston containing both controlled (an actual bump is driven over multiple times) and uncontrolled (bumps are detected from random vehicle trips in the city). The final dataset consists of 813 bumps (21 from controlled data, 792 from uncontrolled data). A signature contains anywhere between 48 to 65 samples, with the mode of the distribution being at 57. In our experiments, we have set \( K \) (the number of bins to partition the signatures) to 3, and \( B \) (the number of bins for the empirical measure \( \mathcal{H}_{2}(\cdot) \)) to 5. The feature vector that we construct for the binary classification formulation consists of 90 features. We have averaged the features over multiple records of a bump for controlled data (for each bump in this set, there are 2–4 readings.) The data set is roughly balanced since 50% of it is labelled as actionable and 41% as non-actionable. When processing the categories of the data to form the labels, we have omitted “screenable” categories, such as crosswalks, expansion joints, train tracks, speed bumps and road distortion/depression, since their location is known to the City, thus, they can be eliminated from our dataset. We have also omitted data for bumps described as “unidentifiable” or otherwise ill-conditioned.

In Fig. 3, a comparison between the performance of our five supervised classification methods is presented. The ROC for each method illustrates the trade-off between the miss detection rate and the false alarm error rates. RBF SVM refers to SVM using a radial basis function kernel [3].

![ROC curves for the 5 classification methods](image)
AdaBoost as a boosting learning algorithm achieves better performance compared to logistic regression and RBF-SVM. Sparse SVM achieves a better performance than RBF-SVM, and correspondingly $\ell_1$ regularized logistic regression is better than logistic regression. This indicates that combining feature selection techniques with the learning algorithms results in better prediction results. Comparing all five methods, Sparse SVM is the strongest classification method in all situations. If we fix a value of the false alarm rate, Sparse SVM provides the lowest miss detection rate, and if we fix a miss detection rate, the algorithm achieves the lowest false alarm rate error. Logistic regression seems to be the weakest classifier in all of our experiments. Notice that with 10% false alarm rate, we can correctly identify half the actionable bumps, which is a level of performance that can enable practical use of our algorithm according to City of Boston officials.

In Fig. 4, we plot the anomaly index values that correspond to each bump in our dataset using a weight $\lambda = 0.5$ in (5), which was empirically found to yield the highest accuracy (88%) for detecting actionable bumps. In Fig. 5 we show the top-28 of the bump list generated in descending order of anomaly index $AI$, thus, identifying bumps most likely to be actionable. Marked in yellow is the only non-actionable bump in the top-28 of this list.

**Fig. 4.** Normalized anomaly index values ($\lambda = 0.5$).

V. CONCLUSIONS AND FUTURE WORK

The goal of this work is to differentiate between actionable bumps which correspond to obstacles that require immediate attention, and non-actionable bumps (e.g., cobblestone streets, speed bumps) for which no immediate action is needed. This classification enables City officials to efficiently and effectively prioritize repairs. We developed two complementary methods to that end. The first method uses classification algorithms but introduces appropriate regularizations to select a sparse set of most useful features. The second method introduces an anomaly index which captures the degree of regularity of a bump, and uses this index to differentiate between more “normal” bumps (non-actionable) from the “anomalous” (actionable) bumps.

As a next step of this work, it is important to be able to differentiate between different types of obstacles; for example, to distinguish a pothole from a poorly repaired sunk casting. The vision is that the accelerometer and GPS data collected by the app can be used in additional applications. An example is detecting wet or icy road conditions or obstacles causing vehicles to experience abrupt motions in a horizontal/lateral, rather than vertical direction. All these results, combined with the ones by our decision support system, could potentially be integrated to create a global “road smoothness” or “road quality” metric, available to all citizens through appropriate web sites, or specialized apps, or even integrated into Google Maps, that can then be used to select the best route!

REFERENCES