

# Towards a Learning Traffic Incident Detection System

Tomas Singliar and Milos Hauskrecht

Computer Science Department, University of Pittsburgh, Pittsburgh, PA 15260

{tomas, milos}@cs.pitt.edu

June 14, 2006

## Abstract

The state of the art in traffic incident detection is dominated by approaches that require significant manual tuning. Our hypothesis is that these time-consuming solutions can be successfully eliminated with the help of machine learning methods and past traffic data collected nowadays on major highways. We show that combining the output of a set of simple, imperfectly tuned, “off-the-shelf” detectors via classification methods is a promising way to obtain a detector with an acceptably low false-positive rate and high and fast recall. We evaluate the performance of a number of simple and combined detectors on real-traffic data and incidents recorded for a section of highway in the Pittsburgh metropolitan area. We show that a relatively simple support vector machine classifier solution outperforms the widely used baseline, the California 2 algorithm. Finally, we discuss the possibilities of improving detector performance by accounting for certain untimeliness of accident recording.

## 1. Introduction

The cost of highway accidents is significantly reduced by their prompt detection. While public reporting and 911 phone calls remain the major source of traffic accident reporting, an automated detection of accidents is becoming an increasingly viable option, thanks to, primarily, the recent increase in the deployment of sensor networks on US roadways.

---

Appearing in *Proceedings of the Workshop on Machine Learning Algorithms for Surveillance and Event Detection at the 23<sup>rd</sup> International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s). This research was supported by the National Science Foundation grants ANI-0325353 and CMS-0416754.

Incident detection systems (IDS) are complex arrangements of technological, organizational and human resources and rely on a variety of inputs, including, but not limited to, sensing and camera equipment, radio reports from police patrols and cell phone calls from the driving public. Our data show that there currently exists a significant delay between the occurrence of an accident and the incept of the traffic management response. The traffic management center, through which the sensor measurements flow, nevertheless often learns about an accident from police reporting.

When an accident occurs, it may reduce the capacity of the affected roadway by blocking one or several lanes. If the highway is operating near its capacity, a congestion forms. It is this *unexpected* congestion that we may hope to observe in the stream of sensor readings.

There are several traditionally recognized families of incident detection algorithms [8]. The most widely deployed one is that of “pattern recognition” algorithms. These algorithms, represented in our study by the so-called California #2 algorithm, typically employ combinations of simple thresholding detectors. However, the tuning of these thresholds requires extensive involvement of traffic experts, as the settings typically do not transfer to a new site and need to be set manually for each traffic sensor location.

While traffic engineers are an expensive resource, large volumes of traffic data have been recorded and are readily available. Our hypothesis is that we can do without the time-consuming human calibration of the detector and instead extract the necessary knowledge from the data using machine learning techniques.

The purpose of this paper is to investigate whether incident detection can profitably be approached as a supervised classification problem and explore the classification power of a set of simple “out-of-the-box” traffic features and their combinations. To test the hypothesis we examine traffic data for Pittsburgh metropolitan area and related incident reports.

We demonstrate that for the available data, a machine learning approach can outperform manually constructed detectors and does not require per-sensor manual tuning.

## 2. Data

The data are collected by a network of sensors that use a host of physical principles to detect passing vehicles (inductive loop detector, microwave and laser detectors, among others). Three traffic quantities are normally observed and aggregated over a time period: the average speed, the volume (number of passing vehicles) and occupancy (the percentage of road length taken up by cars – “traffic density”). The typical aggregation period ranges from 30 seconds to 5 minutes; we have 5 minute aggregates available. We refer to the collection of aggregated measurements from one time interval as a *datapoint*. A set of datapoints is referred to as a *dataset*.

The evaluation takes place on the most accident-prone segment of highway in Pittsburgh. The training dataset was obtained by considering traffic and accident data for the westbound lanes of I376 in a (approximately) 1-mile segment including the Squirrel Hill Tunnels. The segment is defined by the positions of two sensors, to which we will henceforth refer as the *upstream* and *downstream* sensors. There are 37 incidents in this segment verified by hand to leave a signature in the sensor measurements.

Incidents that the Traffic Management Center (TMC) was aware of are noted in the data: their approximate location, time of accident and time of clearing by emergency responders. These incidents are recorded with a delay, so often the accident’s effect is apparent in the data well before the “official” starting time, often as much as one hour ( Figure 1 ). In a supervised framework, we need to label the data as to the occurrence of the accident. Given the unreliability in incident time detection, any datapoint up to 15 minutes prior to the accident is labeled as “accident”. As the effects of an accident persist for some time, we also label all datapoints up to 5 minutes after accident clearing as “accident”.

## 3. Traffic features and detectors

In this paper, a *detector* is any algorithm that takes as input the sensor readings, current and past, and produces a continuous stream of binary outputs signifying the presence of an incident. Conceptually, we associate a detector with a physical sensor location when we consider sensors in isolation. More realisti-

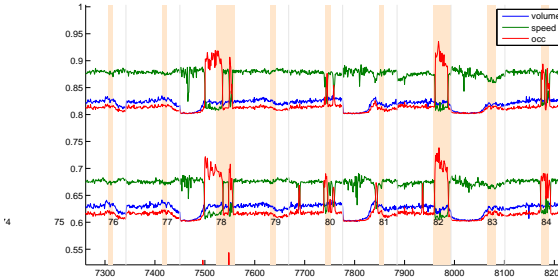


Figure 1. A section of the raw data (best viewed in color at magnification). The red, green and blue lines represent average occupancy, average speed and total volume observed in and aggregated over 5 minute intervals. Number of interval is on the horizontal axis, the vertical axis is scaled to accommodate all curves for two sensors. The vertical pale orange stripes are accidents as recorded by PennDOT. Some accidents square with congestions perfectly (incident beginning around interval 7950), but some leave no observable trace in the data (cca 7420) or their dynamics is unclear (cca 7530).

cally, we associate detectors with roadway segments between sensors when we work with data from more than one sensor.

In most incident detection applications, simple thresholding detectors and their expert-designed combinations are the state of the art. There are two types of features widely regarded as useful in incident detection. The first type captures deviations from normal conditions and the threshold values are typically characterized in terms of distance of the current reading from its mean, measured in standard deviations. The features of the second type characterize the dynamics of the system and are based on temporal differences of sensor readings.

Activity monitor operating characteristic (AMOC) curves are traditionally used for evaluation of rare event detection performance. AMOC curves relate false alarm rate (FAR) to time-to-detection and can be drawn under the assumption that all events are eventually detected. This may be the case in, for instance, disease outbreak detection [4], where every outbreak is eventually detected, but we cannot hope to detect all accidents. We have to introduce an artificial time-to-detection limit for accidents that remain undetected. For the sake of readability, we will not use AMOC curves in this evaluation, but rather provide them in an online supplement <sup>1</sup>.

A false alarm occurs when the system raises an alarm,

<sup>1</sup><http://www.cs.pitt.edu/~tomas/papers/icml06w>

but no accident is present. The FAR is the number of false alarms divided by the number of detector invocations. The detection rate (DR), is the number of accidents actually detected, divided by the number of accidents that occurred; thus higher DR is more desirable. These measures are conflicting in the sense that we can typically increase one at the expense of the other. A *performance envelope* curve relates FAR and DR and gives a more appropriate description of a detector’s performance. However, the curve can be deceiving as the cardinality of the dataset is much larger than the number of incidents. As a consequence, the curve is sensitive to prevalence of incidents and the intuition that the random guess will achieve a curve close to the  $(0, 0) - (1, 1)$  diagonal, which holds true for ROC curves [7], does not transfer to performance envelopes. With a performance envelope or an ROC curve, the area under the curve (AUC) is a scalar summary statistic, suitable for comparisons.

Accidents are observed indirectly, via their effect on the traffic flow. Their interference with traffic is strong when the highway is near its operating capacity and when the accident is major and blocks at least a single lane. However, it is fundamentally difficult to detect non-blocking accidents and those that occur under light load, as the deviation from normal traffic patterns may be negligible. The aggregation period of our data is 5 minutes. This also limits the achievable performance, as in minor accidents, the roadway is often blocked only for a few minutes before it is cleared. Fortunately, missing such minor incidents carries smaller cost.

The target performance at which a system is considered at least marginally useful depends on where the system is to be deployed. A study [9] surveying traffic managers found that would seriously consider using an algorithm that achieves a DR over 88% and FAR under 2%. Perhaps an even better metric for measuring how many false alarms the users will accept is the positive predictive value (PPV), also known as precision. It is plausible that users will find it preferable to say what proportion of unsubstantiated alarms (to the total number of alarms) they are willing to tolerate than to say the same about FAR. Moreover, tolerability of FAR will depend on the time interval between successive ID algorithm invocations.<sup>2</sup>

Finally, not all false alarms are bad. While some alarms will not be caused by accidents, they often will

<sup>2</sup> The users in the above study were very liberal in their tolerance of false alarms. A medium-sized city will have hundreds of detector sites. A 2% FAR would require the managers to tend to several alarms per minute, most of them false. The users were victims to the base rate fallacy.

indicate unusual traffic conditions that, by definition, should be of concern for the traffic managers.

### 3.1. Train/test splitting

Our dataset is one long sequence. It matters how we split it up into shorter sequences that will be used as learning instances. The straightforward random split cannot be used as it relies on the iid assumption. It is better to divide the train/test split by incidents, making sure an entire incident sequence makes it into one and only one of the sets.

To create the training set, we first select  $I_{train}$  “incident” sequences of preset length  $L$  so that the reported time of the incident falls in the middle of the incident sequence.  $C$  “control” sequences without an incident are selected so that no incident is recorded within additional  $L/2$  datapoints before and after the control sequence. This safeguards against the imprecise accident recording. By choosing  $I_{train}$  and  $C$ , the class prior in the training set can be biased towards incident occurrences. The testing set consists of the  $I_{test} = I_{all} - I_{train}$  incident sequences that were not selected for the training set. Additional sequences without accidents are added so that the testing set has class prior equal to that in the entire dataset.

To obtain the experimental statistics, we use 10 different train/test splits using the above method. All statistics reported are averages and standard deviations across this splits. Error bars in the graphs represent one standard deviation.

### 3.2. Univariate detector methods

Virtually every “pattern recognition” detection algorithm is built on basic threshold detectors. Let us examine what detection power they have. We first look at detectors that are local, in that they use information from only one sensor, and oblivious, in that they forget all about the past.

We note that the results in this paper were obtained on data that was in no way “preprocessed”. The most obvious such step, suppressing diurnal trends by subtracting the daily mean [3], perhaps surprisingly did not result in significant changes in the reported performance.

In Figures 2 and 3, we see the performance of the most basic detectors. Detector **Occ**( $s_{up}, t_0$ ) detects an accident whenever the occupancy sensor reading at the upstream sensor  $s_{up}$  exceeds a threshold. The second argument in parentheses,  $t_0$ , is to denote that the algorithm uses measurement most recent at the time of the detector invocation. Detector **Spd**( $s_{up}, t_0$ ) outputs 1

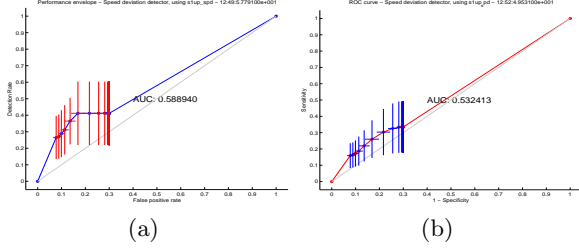


Figure 2. Performance envelope and ROC curve for a simple detector:  $\mathbf{Spd}(s_{up}, t_0)$ , operating on the upstream sensor. Threshold is varied from the minimal to the maximal value of the reading found in data.

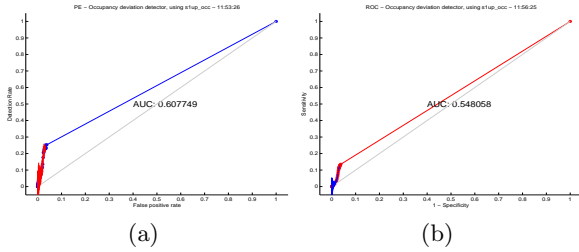


Figure 3. Performance envelope and ROC curve for  $\mathbf{Occ}(s_{up}, t_0)$ . The threshold is varied from 0 to 5 standard deviations above the mean occupancy.

if the speed falls below a detection threshold.

To improve on the false positive rate, we can combine these detectors via an AND-conjunction. Similarly, to improve on the detection rate, it is natural to combine them with an OR-gate. More generally, we might require that  $k$  of the ensemble of  $m$  predictors output 1 for the combined detector to output 1.

### 3.3. Temporal variation

Now we consider, in isolation, features that capture temporal variation in flow. Intuitively, sharp changes in flow characteristic may be indicative of accident, while congestion from capacity saturation should have a more gradual onset. The temporal derivative features are designed to enable this distinction.

Both the detectors utilizing spatial differences and those using temporal differences outperform the local detectors, as seen in Figure 4. The detector of occupancy spike in particular shows a DR of about 30% around FAR 1%.

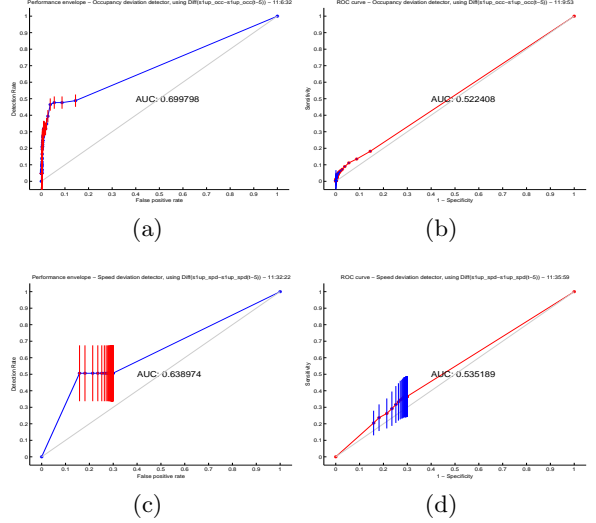


Figure 4. Performance envelope and ROC curves of temporal derivative detectors: top – occupancy spike detector  $\mathbf{Occ}(s_{up}, t_0 - t_1)$ , bottom – speed dip detector  $\mathbf{Spd}(s_{up}, t_0 - t_1)$ . Threshold varied from  $\mu$  to  $\mu + 5\sigma$ .

### 3.4. Spatial relations

The intuition behind including, as an input to a detector, the reading of the neighboring sensor is that accidents and benign congestions can be distinguished by the flow characteristics at the downstream sensor. When an accident constricts the roadway capacity, we observe a congestion upstream of the accident. Unlike a benign congestion, an accident should cause a *drop* in the downstream sensor volume measurement. The power of the difference detectors alone is shown in Figure 5.

There is hope in these curves: the detection rate exceeds 50% and the area under ROC curve indicates significantly nonrandom detection behavior, but the 0.5% target FAR still only yields about 30% detection rate and the coverage is sparse, indicating high sensitivity to threshold setting.

## 4. The California algorithm

The algorithm known as “California” TSC-2 is a popular baseline model against which new detection algorithms are most often compared. Improvements over TSC-2 have been proposed [10], but it remains a widely used algorithm. TSC-2 proceeds as follows:

- Let  $Occ(s_i)$  denote occupancy at the upstream sensor  $s_i$  and  $Occ(s_{i+1})$  the same at the downstream sensor. If  $Occ(s_i) - Occ(s_{i+1}) > T_1$ , proceed to the next step.

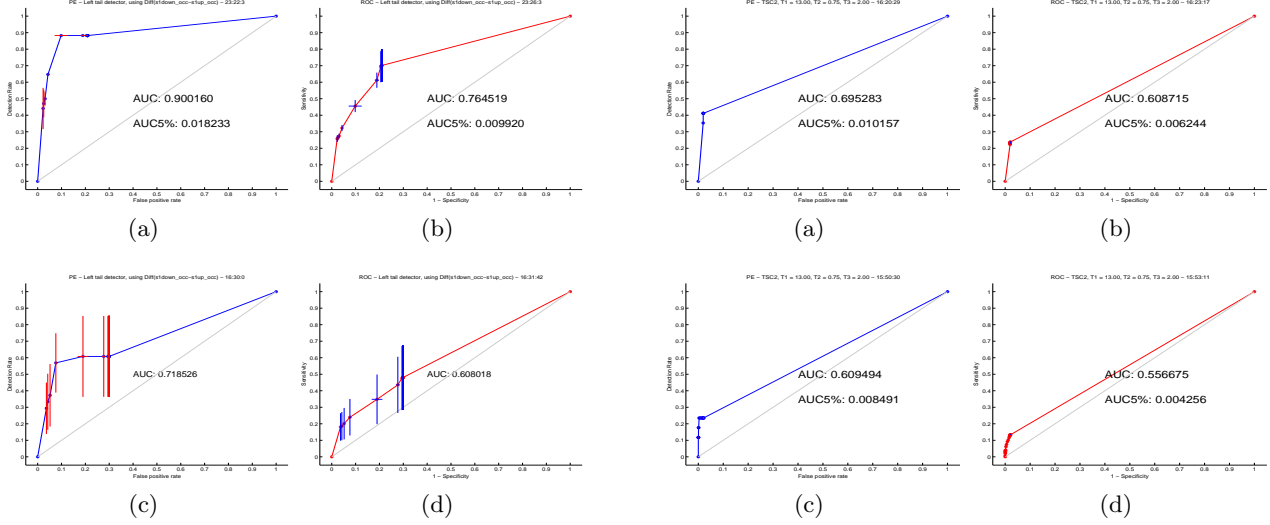


Figure 5. ROC curves for two difference detectors:  $\text{Occ}(s_{up} - s_{down}, t_0)$  for one road segment in the top row;  $\text{Spd}(s_{up} - s_{down}, t_0)$  for another segment in the bottom row. Threshold is varied in the range  $(\mu, \mu + 5\sigma)$ .

- If  $(\text{Occ}(s_i) - \text{Occ}(s_{i+1}))/\text{Occ}(s_i) > T_2$ , proceed to the next step. The rationale behind this step is while a capacity-reducing accident will always produce large absolute differences in occupancy, these may also be produced under almost stalled traffic conditions.
- If  $(\text{Occ}(s_i) - \text{Occ}(s_{i+1}))/\text{Occ}(s_{i+1}) > T_3$ , wait until the next reading. If  $T_3$  is still exceeded, flag an alarm. The wait is introduced to cut down on false alarms.

Thresholds  $T_1, T_2, T_3$  need to be calibrated manually for each road segment. We calibrated the TSC-2 algorithm for one segment by inspecting the ROC curves drawn for each threshold parameter  $T_1$  through  $T_3$ , holding the remaining parameters fixed. Since the algorithm uses a conjunction of conditions, we used the setting giving the highest DR before drawing the ROC curves for the next parameter. The best performance was 0.288 DR at 0.001 FAR at the final calibration  $T_1 = 13.0$ ,  $T_2 = 0.77$ ,  $T_3 = 5.0$  and verified by an exhaustive procedure trying all possible settings of the three parameters on a discrete grid covering a wide range of parameter values. The performance characteristics of the California 2 detector are in Figure 6.

The steep slope of the initial section of the ROC curve is very desirable as the most difficult challenge here is the low FAR. However, the model only detects a third of the incidents at best. This is hardly acceptable for practical purposes unless more fine-grained data are

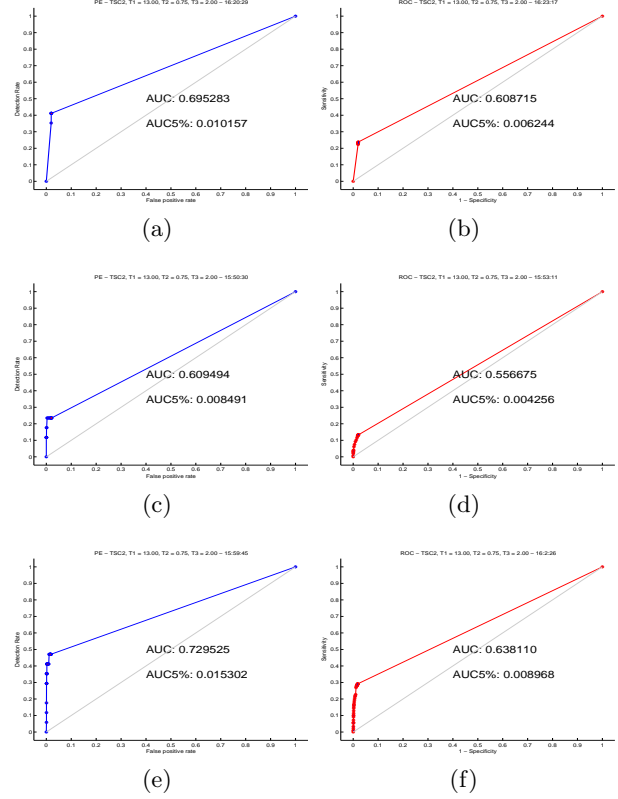


Figure 6. Performance of the California 2 algorithm with performance envelopes and ROC curves. Top, varying the  $T_1$  threshold in the (1, 10) range; middle,  $T_2$  threshold, range (0.2, 1); bottom,  $T_3$  threshold, range (1, 10).

made available that could boost its performance.

## 5. Support vector machines

Naturally, the next step is to combine all of the features. Unfortunately, there is no clear method of doing this. There are too many combinations - do we look at derivatives of spatial differences? Spatial differences of derivatives? Some complex Boolean formula of all of these? Noting that all the threshold detectors are special cases of linear combinations, a support vector machine (SVM) is their proper generalization.

Since we use SVM code designed for equal misclassification cost, we supersample the minority (“accident”) class training datapoints to simulate unequal cost.

In the first SVM experiment, the learner gets as features all the readings at sensors  $s_{up}$  and  $s_{down}$  at the current time. In subsequent experiments, this basic set is extended with other classes of features. The results can be seen in Figure 7.

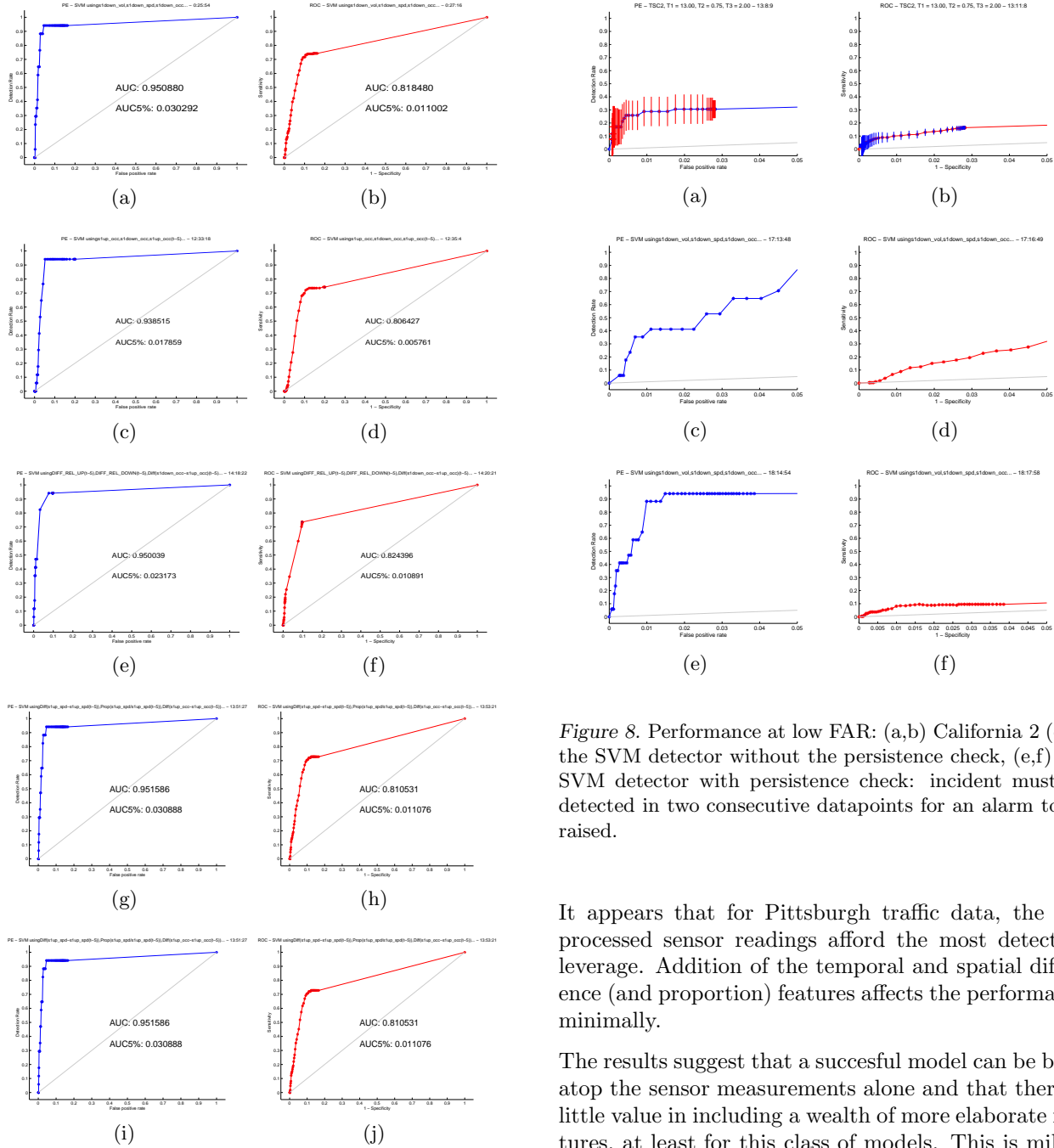


Figure 7. Performance of the SVM model for different feature sets. The features are: (a,b) All readings for the two sensors defining the monitored road segment. (c,d) Only readings that are used to calculate California 2 features. (e,f) California 2 features (the occupancy ratios). (g,h) All of current and previous step measurements. (i,j) All current measurements together with differences and proportions of the corresponding readings at the upstream and downstream sensors. For drawing the curves, the intercept of the SVM hyperplane is varied in the (-1,1) range, giving a lower estimate on the “true” ROC curve [1].

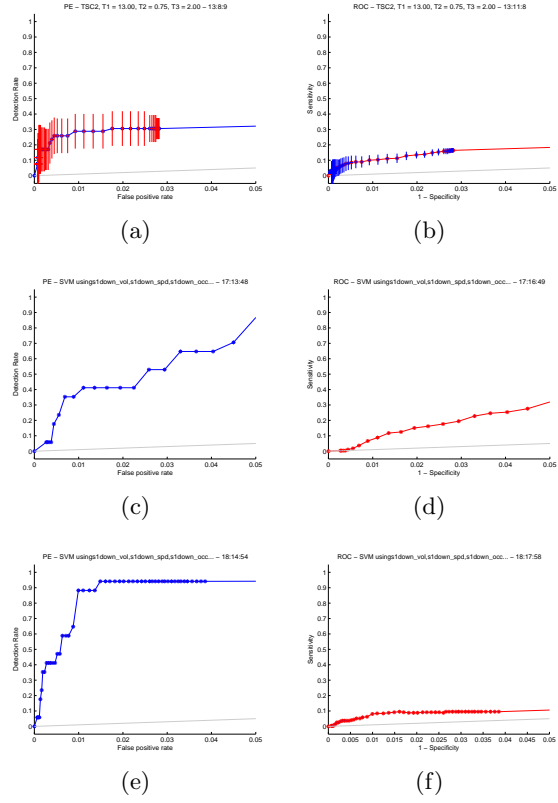


Figure 8. Performance at low FAR: (a,b) California 2 (c,d) the SVM detector without the persistence check, (e,f) the SVM detector with persistence check: incident must be detected in two consecutive datapoints for an alarm to be raised.

It appears that for Pittsburgh traffic data, the unprocessed sensor readings afford the most detection leverage. Addition of the temporal and spatial difference (and proportion) features affects the performance minimally.

The results suggest that a successful model can be built atop the sensor measurements alone and that there is little value in including a wealth of more elaborate features, at least for this class of models. This is mildly surprising in light of Figures 5 that demonstrates that spatial difference is among the best single features. The whole appears to be more than the sum of the parts.

The California 2 algorithm, which uses differences and proportions between occupancies at the upstream and downstream sensors, without incorporating the actual values, does poorly in the higher FAR portion of the curve. However, in the very low FAR rates, California 2 beats the SVM detectors. This occurs because it only signals an incident after it has been verified by the last

step, a persistence check. The other detectors tend to quickly spit out the alarm around a suspect datapoint. However, a persistence check can be applied to any detector and dramatically improve its low FAR rate, as demonstrated in Figure 8 for the SVM-based detector.

## 6. Summary and future work

The design of incident detection algorithms is extremely sensitive to particularities of the deployment location. For this reason, it requires much expert effort to put an IDS in place.

In this paper, we examined the performance of simple detectors and their combinations. Most simple predictors are weak and need to be combined to yield usable incident detection algorithms.

A simple support vector machine learning scheme was able to outperform the model underlying much current practice. Moreover, this performance is achieved in presence of significant noise in the class labeling.

It appears that simple sensor measurements are quite sufficient for the achieved level of detection performance and the value of constructing more elaborate features is questionable.

Many questions remain open whose answers promise to improve incident detection. For instance, Coupled Hidden Markov Models have somewhat disappointed researchers [2, 5] in the past. Yet, CHMMs seem to be just the right generative probabilistic model. Why exactly does this natural model underperform? Might the question be profitably sidestepped if we learn a discriminative model such as a CRF instead?

Finally, the challenging problem of noisy data labeling remains open and we believe it is the immediate obstacle to further improvements. The most promising avenue of attack is to treat the true state of the road as a hidden variable. The concluding subsection is devoted to an illustration of the concept.

### 6.1. Accounting for label shifts with Dynamic Naive Bayes

An incident is typically recorded some time after it happens, when the TMC learns about it. Therefore, the onset of the incident will typically be labeled as incident-free. The class label is therefore noisy, in a skewed way. This implies that we should not regard the incident records as the ground truth, but rather as an observation of the hidden state of the accident and track the progression of the accident.

We define an dynamic Bayesian network model, with a

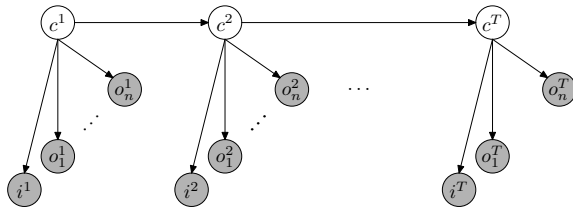


Figure 9. The Dynamic Naive Bayes graphical model.

	$p(i = v   s =$	$ns)$	$ae)$	$as)$	$rp)$
$v = 0$		0.01	0.30	0.90	0.10
$v = 1$		0.99	0.70	0.10	0.90

Table 1. The “anchoring” conditional probability table

single discrete hidden state variable  $s$  and a number of conditionally independent univariate Gaussian observation nodes  $o_1, \dots, o_n$ . There is also a distinguished binary observation  $i$ , which is the incident state as observed by the TMC (Figure 9).

We attribute the following semantics to the values of the hidden state variable: Normal steady state  $ns$  is expected to undergo slow changes. The accident effect buildup  $ae$  captures the first minutes after an accident, characterized by a rapid spike in occupancy at upstream sensor, volume drop at the downstream sensor and a drop in speed at the upstream sensor. In the accident steady state  $as$ , capacity remains impaired, the upstream occupancy remains at the saturation limit, speed and throughput are lowered. The recovery phase  $rp$  is characterized by increasing speed at the upstream sensor and volume hovering near capacity at the downstream sensor.

The conditional distribution  $p(i|s)$  is set by hand (Table 1) and intended to anchor the rows and columns of the inter-slice transition matrix  $p(s^n | s^{n-1})$  to their intended interpretation. The inter-slice transition matrix as well as the conditional probabilities  $p(o_n | s^n)$  are learned from data.

An alarm threshold  $\theta_a$  is set and alarm is triggered at time  $n$  if  $p(s^n = ae | \mathbf{o}^n) + p(s^n = as | \mathbf{o}^n) \geq \theta_a$ .

Unfortunately, the performance of the simple Dynamic NB models falls short of the SVM detector performance. This is most likely so because the Naive Bayes structural assumptions are a poor fit for the data. While it may also be the case that the coarse grain of the data does not warrant such fine distinction, a similarly subpar performance was seen in a model with only 2 states.

On the other hand, it is unclear how to assign a prob-

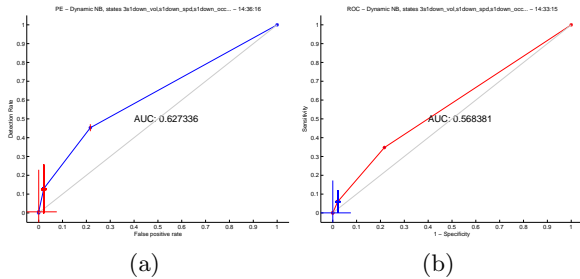


Figure 10. Performance envelopes and ROC curves for a Dynamic NB using both sensors’ measurements and their differences and proportions. The curve is drawn by varying  $\theta_a$  in the range (0.8, 1).

abilistic interpretation to the SVM so that it may be integrated into a framework that considers time. Here lies a challenge for future work: a well performing classifier that can be “dynamized”, such as a Bayesian network.

Promise can be also found in bootstrapping approaches, where the output of a classifier learned on noisy data, could be used to improve the estimate of when the accident begins and thus, thanks to better quality of labeling, lead to more a more powerful detector. One advantage of this technique is its generality, any classifiers, even of different type, can be used.

## 7. Acknowledgements

We wish to thank the reviewers for their comments. Since we could not accomodate their very helpful suggestions due to space considerations, we decided to provide an online supplement of AMOC curves at <http://www.cs.pitt.edu/~tomas/papers/icml06w>.

## References

- [1] Francis Bach, David Heckerman, and Eric Horvitz. On the path to an ideal ROC curve: Considering cost asymmetry in learning classifiers. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, Jan 6-8, 2005, Savannah Hotel, Barbados*, pages 9–16. Society for Artificial Intelligence and Statistics, 2005. (Available electronically at <http://www.gatsby.ucl.ac.uk/aistats/>).
- [2] Peter Bickel, Chao Chen, Jaimyoung Kwon, John Rice, Pravin Varaiya, and Erik van Zwet. Traffic flow on a freeway network. In *Proceedings of a Workshop on Nonlinear Estimation and Classification*. Mathematical Sciences Research Institute, 2001.
- [3] George Box and F.M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, Oakland, CA, second edition, 1976.
- [4] Tom Fawcett and Foster Provost. Activity monitoring: Noticing interesting changes in behavior. In Chaudhuri and Madigan, editors, *Proceedings on the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 53–62, San Diego, CA, 1999.
- [5] Jaimyoung Kwon and Kevin Murphy. Modeling freeway traffic with coupled hmms, May 2000.
- [6] O. L. Mangasarian and David R. Musicant. Lagrangian support vector machine classification. Technical Report 00-06, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, June 2000. <ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-06.ps>.
- [7] Foster J. Provost and Tom Fawcett. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Knowledge Discovery and Data Mining*, pages 43–48, 1997.
- [8] Martin P.T., Perrin H.J., and Hansen B.G. Incident detection algorithm evaluation. Technical Report UTL-0700-31, Utah Traffic Laboratory, July 2000.
- [9] Stephen G. Ritchie and Baher Abdulhai. Development, testing and evaluation of advanced techniques for freeway incident detection. Technical Report UCB-ITS-PWP-97-22, California Partners for Advanced Transit and Highways (PATH), 1997.
- [10] Y.J. Stephanedes and J. Hourdakakis. Transferability of freeway incident detection algorithms. Technical Report Transportation Research Record 1554, Transportation Research Board, National Research Council, 1996.