

Learning to detect incidents from noisily labeled data

Tomáš Šingliar · Miloš Hauskrecht

Received: 20 December 2007 / Revised: 14 May 2009 / Accepted: 27 July 2009 /
Published online: 30 September 2009
Springer Science+Business Media, LLC 2009

Abstract Many deployed traffic incident detection systems use algorithms that require significant manual tuning. We seek machine learning incident detection solutions that reduce the need for manual adjustments by taking advantage of massive databases of traffic sensor measurements. We first examine which traffic flow features are most useful for the incident detection task. Then we show that a supervised learner based on the SVM model outperforms a fixed detection model used by state-of-the-art traffic incident detectors. However, the performance of a supervised learner suffers from temporal noise in the data labels due to imperfections of the incident logging procedure. Correcting these misaligned incident times in the training data achieves further improvements in detection performance. We propose a label realignment model based on a dynamic Bayesian network to re-estimate the correct position (time) of the incident in the data. Training on the automatically realigned data then consistently leads to improved detection performance in the low false positive region.

Keywords Incident detection · Road network · SVM · Dynamic Bayesian networks

1 Introduction

Operation of real-world infrastructure systems is often interrupted with events that interfere with their expected function. Prompt detection of these events is critical for recovery of the system and for mitigation of further costs associated with the incidents. With advances in sensor instrumentation, many of these systems can now be better monitored and occurrence of the incidents automatically detected. Our work focuses on methods for early and accurate detection of road traffic incidents.

Editors: Dragos Margineantu, Denver Dash, and Weng-Keen Wong.

T. Šingliar (✉) · M. Hauskrecht
Computer Science Dept., University of Pittsburgh, Pittsburgh, PA 15260, USA
e-mail: tomas.singliar@boeing.com

M. Hauskrecht
e-mail: milos@cs.pitt.edu

The most widely deployed traffic incident detection models are fixed-structure models that combine and threshold a set of signals such as volumes, speed and speed derivatives (Martin et al. 2000). Tuning of these thresholds requires extensive involvement of traffic experts. What is worse, as the settings extracted for one site typically do not transfer to a new site, the tuning costs are multiplied by the number of sites in the network. Models that can be automatically tuned from data can reduce or eliminate costly recalibrations and improve detection performance (Dia et al. 1996; Rose and Dia 1995; Ritchie and Abdulhai 1997; Parkanyi and Xie 2005). Inter-site transferability of detection algorithms is a central concern in automatic incident detection (Stephanedes and Hourdakakis 1996). Machine learning offers a recipe for transferable detection algorithms: re-learn the detector on data from the new site, possibly borrowing power from the old site in form of prior knowledge. We are presenting just such a solution.

If incident labels are available, the detection models can be trained in supervised mode and the detection task is one of classification. Alternatively, incident detection can operate by detecting anomalies in the behavior of the traffic system. In such a case, incident labels are not needed and any anomalies are assumed to be incidents. However, not every anomaly is in fact an incident. This mismatch leads to generation of false positives and a reduction in detection accuracy. Thus it is reasonable to expect supervised approaches to outperform unsupervised anomaly detection. Since we do have access to incident logs, we adopt the supervised learning view.

First, we gauge the informativeness of particular traffic flow features by constructing simple detectors using a single feature at a time. Then, we attempt to combine features in a probabilistic detector. A dynamic naive Bayes detector seems to capture the known phases of congestion onset and clearance, but the resulting detection performance is disappointing. We are much more successful with a SVM-based detection and show it easily outperforms the optimally calibrated standard model (the “California 2” algorithm).

Real-world data are often subject to bias and noise coming from many different sources. Acknowledging and explicitly modeling the biases may translate to further detection improvements. In our case, we have noticed that incident logs are imperfect; the initial time of incidents is logged with a variable delay. Consequently, the incident label may be misaligned with the onset of the observed changes in traffic flow caused by the incident. Training a learner with such temporally misaligned class labels leads to suboptimal detector performance.

To work around the incident logging problem, we propose a new dynamic Bayesian network (Dean and Kanazawa 1989) that models the misalignment. It relates traffic flow quantities and the observed label position to the actual time of incident manifestation in data. We train the model on manually realigned data from a *single* highway segment. Once learned, the realignment model can be transferred to other highway segments to correct the incident labeling—a preprocessing stage. The transfer procedure is a variant of EM, starting with priors obtained from the hand-aligned data and combining them with data from the target site to adapt to the new location. The realignment model generates new incident labels in temporal data that are then used to train a supervised classifier such as a SVM to obtain the detection algorithm. This approach allows us to learn, with a *limited* amount of human effort, a more reliable detection model. We demonstrate the improvement in detector quality on traffic flow and incident data collected in the Pittsburgh highway network.

This paper elaborates on and synthesizes results from conference papers (Šingliar and Hauskrecht 2006, 2007) and gives results on more simple detectors. In addition, the Tree Augmented Naive Bayes alignment method is presented with experimental results. We extend evaluation to additional test sites to strengthen confidence in the performance of the detector transfer method.

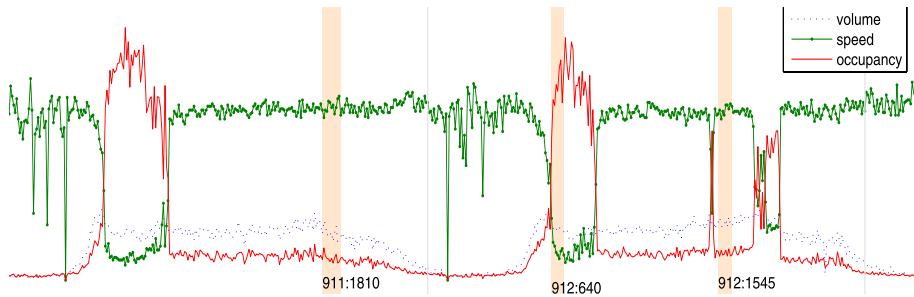


Fig. 1 (Color online) A section of the raw data. The *red (solid)*, *green (solid with markers)* and *blue (dotted)* lines represent average occupancy, average speed and total volume, respectively. Time is on the *horizontal axis*. The *vertical (orange) stripes* represent the reported accidents durations. A *thin grey vertical line* is drawn at midnight of each day. The numbers at the bottom encode accident time as recorded by TMC. Some accidents square with congestions perfectly (912:640—September 12th, 6:40 am), some are slightly shifted (912:1545) and some even have no observable effect on traffic (911:1810). The incident at 912:640 is mostly obscured by morning peak traffic—compare to the morning traffic on the previous day

2 The data and detection task

In this section, we look at the available data and define the incident detection task together with the relevant performance metrics.

2.1 Traffic and incident data

The data are collected by a network of sensors that use a number of physical principles to detect passing vehicles. Three traffic quantities are normally observed and aggregated over a time period: the average *speed*, *volume* (number of passing vehicles) and *occupancy* (the percentage of road taken up by cars). Incidents that the metropolitan 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 (Fig. 1). Short free-text accident descriptions are also available.

The detection task is to continuously observe the data stream and raise an alarm when the readings indicate an incident.¹ An incident restricts the capacity of the roadway by blocking one or more lanes, forcing drivers to slow down to navigate around it. This will result at a temporary drop in the number and density of vehicles passing the downstream sensor. Upstream of the accident, a jam forms. When the tail end of the jam approaches the nearest sensor, it will cause a drop in measured speed and increase in vehicle density. The time when the sensors detect the anomaly depends on the utilization of the highway, distance to the sensors and severity of the incident. A large amount of noise obscures this simple pattern, as traffic behaves stochastically and other unobserved conditions such as weather vary widely.

2.2 Performance metrics

A false alarm occurs when the system raises an alarm, but no accident is present. The false alarm rate (FAR) is the number of false alarms divided by the number of detector invoca-

¹The term *incident* includes vehicular accidents as well as unscheduled emergency roadwork, debris on the road and many other hazards. Most incidents are accidents and we will use the terms interchangeably.

tions. Receiver operating characteristic (ROC) curves (Provost and Fawcett 1997) are the standard metric designed to quantify detection of one-off binary events. Because accidents affect the traffic for a longer span of time and the detections are not equally valuable around the beginning and the end of the span, we instead prefer the activity monitor operating characteristic (AMOC) curve as the primary performance metric. AMOC curves are used for evaluation of rare event detection performance, such as detection of disease outbreaks (Fawcett and Provost 1999; Cooper et al. 2004). AMOC curves relate false alarm rate (FAR) to time-to-detection (TTD). TTD is defined here as the difference between the time of the start of the first data interval that was labeled as “accident” by the detector and the reported incident time. Note that this number can be negative because of the delayed incident recording. As we cannot guarantee to detect all accidents, we introduce a two-hour time-to-detection limit for accidents that remain undetected. When a scalar metric is desired, we compare detectors on $AUC_{1\%}$, the area under the curve restricted to the (0, 0.01) sub-interval of FAR. This is a better indicator of detector performance in the usable low-false-positive region than the area under the entire curve. The detection rate (DR) is the number of correctly detected incidents divided by the number of incidents that actually occurred. A performance envelope (PE) is a curve relating the false alarm rate and the detection rate. So as not to flood the paper with numerous figures, we only report AMOC curves. The interested reader can find the ROC and PE curves at <http://www.cs.pitt.edu/~tomas/papers/MLJ07/> or in a precursor work (Šingliar and Hauskrecht 2006).

The target performance at which a system is considered useful depends chiefly on its users. A study (Ritchie and Abdulhai 1997) surveying traffic managers found that they would seriously consider using an algorithm that achieves a DR over 88% and FAR under 2%. For any rare event detection system, a low FAR is absolutely essential. A system with a high FAR subjects its users to “alarm fatigue”, causing them to ignore it.

3 The detection models

In this section, we describe a number of detection algorithms, beginning with some related techniques and proceeding to describe simple “unsupervised” techniques that do not take advantage of the available incident information. Then, we introduce the commonly deployed California #2 detection algorithm as a baseline. Finally, we show that “supervised” detectors—those that do leverage the incident information and combine multiple features can attain better performance.

3.1 Related work

Automatic incident detection has received intense attention of civil engineers. Several families of incident detection algorithms have been developed (Martin et al. 2000). In general, the detectors can either be crafted by hand or learned from data.

The California class of algorithms are essentially simple decision trees, where the tests are typically of the form $traffic_quantity \leq T_i$. T_i is a threshold, parameter of the procedure that needs to be calibrated. They represent here the class of expert-designed detectors.

The alternative to using large amounts of expert labor is to fit the parameters of the model from recorded data. There are two basic approaches, unsupervised and supervised.

Traditionally, incident detection took the unsupervised learning approach of modeling the normal conditions and reporting any deviations as anomalies. Naturally so—it used to be and still is much easier to obtain data on normal conditions than incident situations. Practically any general anomaly detection method can be applied to incident detection. A simple

t -test for increase in occupancy over 10 most recent time intervals as opposed to 10 preceding intervals was used in Antoniadou and Stephanedes (1996) to detect with sparse detector stations. Some detection algorithms can be model-less: In double exponential smoothing (Busch and Fellendorf 1990), the first exponential smoothing generates an intermediate sequence which is the input of a second smoothing equation. A jump in the doubly smoothed sequence is indicative of an incident. No explicit characterization or model of “normal conditions” is present.

Time-series algorithms such as ARIMA have been used to signal an incident when an observed value lies in the tail of the predictive distribution given previous k values. They have mostly fallen out of favor because they tend to underestimate the variance of traffic measurements which leads to unacceptable false alarm rates (Martin et al. 2000).

Finally, we have the learning approaches that are supervised in the sense that they work with a set of incidents which allow the data to be clearly labeled as incident/non-incident conditions. This allows the problem to be recast as a binary classification problem: given a set of features derived from recent data, classify the current conditions as normal or incident. Previous work concerning supervised learning of incident detection from loop detector data used almost exclusively neural networks (Dia et al. 1996; Rose and Dia 1995). Recently, Zhang and Taylor (2006) presented a straightforward Bayesian network approach with good results, but strong likelihood of overfitting caused by the way they handled their very small set of incidents. Here, we present an SVM-based incident detection technique.

Another widely traveled computational avenue of incident detection is processing of video from traffic cameras (Michalopoulos et al. 1993) in order to detect traffic stall. As we do not have access to video data, we will not relate to that body of work.

Frequently, a detector working well in the lab simulation settings disappoints in the actual deployment (Parkanyi and Xie 2005). The detectors were often tested on data sets that might not be properly calibrated. That is, the proportion of incident instances in test set was much higher than could be expected in real deployment. This paper pays close attention to the design of the evaluation procedure to ensure that the calibration is preserved and that we can be confident in the evaluation results.

3.2 Simple univariate models

The information value of data-derived features depends strongly on the temporal aggregation of the available data. In this section, we examine the detection power of simple distribution-tail detectors on our data, which has five-minute aggregation.

In Fig. 2 we see the performance of the most basic detectors. Detector **Occ**(s_{up}, t_0) detects an accident whenever the occupancy 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 if the speed falls below a detection threshold. While occupancy is clearly a more powerful feature, speed measurements can, as we show later, contribute important information in combination with other features.

A fundamental expectation for incident detection is that an incident changes the traffic flow abruptly. The abruptness should be best captured by features that are temporal derivatives of the sensor measurements. The performance of detectors based solely on thresholding temporal derivatives of speed and occupancy is graphed out in Fig. 3. Again, occupancy derivative seems to be the more informative feature.

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 discontinuity

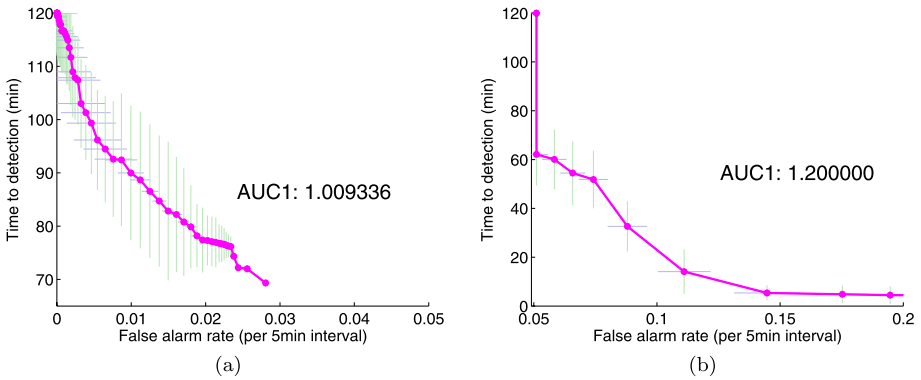


Fig. 2 AMOC curves for two simple detectors: (a) $\text{Occ}(s_{up}, t_0)$ and (b) $\text{Spd}(s_{up}, t_0)$. The speed detector has no practical value here, but the occupancy detector begins to detect incidents even with a very low sensitivity setting, translating to low false positive rates. The curve is plotted by varying the threshold from the minimal to the maximal value of the reading found in data. For each threshold, TTD and FAR are noted. This evaluation is repeated with 10 random sub-samplings of the incident database. The curve points and the bars plotted are the average and standard deviation respectively. AUC1 plotted in the graph is the area under the first 1% of the curve. Note that different portions of the FAR range are shown in the two figures

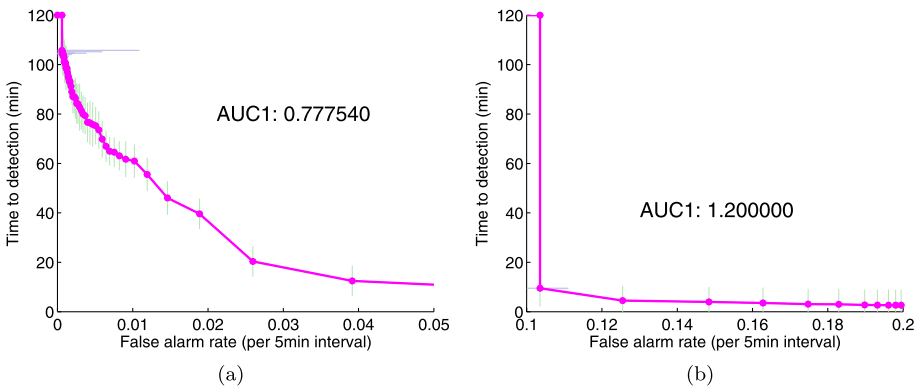


Fig. 3 AMOC curves of temporal derivative detectors (a) occupancy spike detector $\text{Occ}(s_{up}, t_0 - t_1)$, (b) speed dip detector $\text{Spd}(s_{up}, t_0 - t_1)$. Threshold varied from μ to $\mu + 5\sigma$. Again, the speed detector is not very useful. The labels above the figures record experiment setup parameters and should be disregarded

of flow characteristics between the upstream and downstream sensors. 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 and occupancy measurements. The power of the difference detectors alone is shown in Fig. 4.

3.3 Adaptive detectors

The diagrams above inform us about the value of a feature for incident detection. However, the simple thresholding fails to take into account that measurements may vary in time with normal traffic and no incident involved. We may assume that such variation is smooth under normal condition. Thus a natural learning approach would be a form of t -test. Suppose we

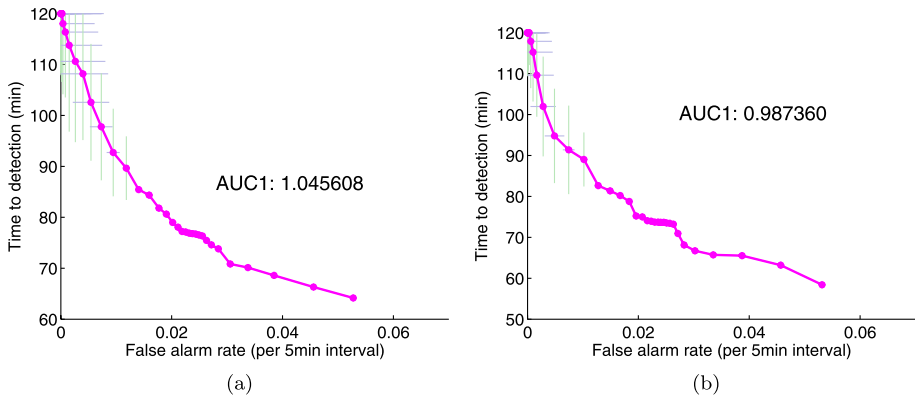


Fig. 4 AMOC curves for two spatial difference detectors: **(a)** $\text{Occ}(s_{\text{sup}} - s_{\text{down}}, t_0)$ and **(b)** $\text{Spd}(s_{\text{sup}} - s_{\text{down}}, t_0)$. Threshold is varied in the range $(\mu, \mu + 5\sigma)$. In contrast to the two previous cases, the spatial difference in speed is even more indicative of an incident than a difference in occupancies

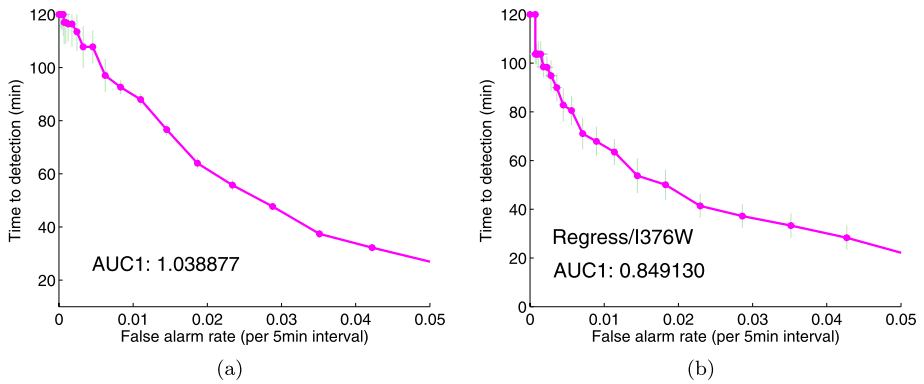


Fig. 5 AMOC curves for **(a)** a t -test detector (see text), working with upstream volume, **(b)** a regression detector registering drop in density downstream of the incident. These are the respectively best-performing features for these models and $N = 10$. The location is the same as for other detectors evaluated in this section, a westbound portion of Parkway East in Pittsburgh, denoted “I376W”

need to decide whether to raise an alarm on measurement x^t given previous N measurements x^{t-1}, \dots, x^{t-N} . Under the assumption of normal distribution, we can check whether x^t lies in the (right or left, depending on the feature) tail of the maximum-likelihood normal distribution given the observations x^{t-1}, \dots, x^{t-N} . We call this the t -test detector. Even better, we could attempt to follow the trend with linear regression. The linear regression detector learns from the N most recent datapoints and predicts the value x^t . Again, the value is anomalous if it lies in a tail of the predictive distribution.

See Fig. 5 for the corresponding graphs. As expected, the t -test detector using upstream volume is weaker than the predictive regression model. With the regression model, the downstream occupancy was the best discriminating feature. In both cases it is easy to see how the features are directly affected by an incident. However, the regression model is still inferior to the upstream occupancy spike detector, the best of the simple anomaly detection approaches. The lack of performance boost with a learning method can be explained by the fact that the

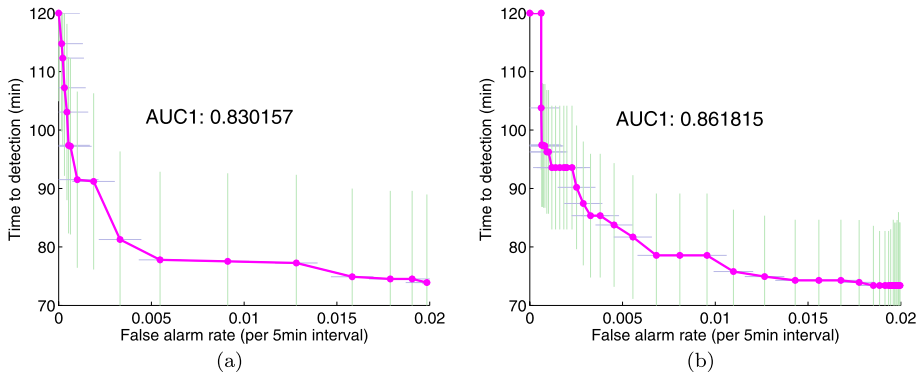


Fig. 6 Performance of the California 2 algorithm. (a) The curve obtained by varying the T_2 threshold, (b) curve from T_3 sweep

difference (“spike”) features are already designed to capture any abrupt jumps and thus the regression detector tends to pick up the same spikes or drops in feature values.

3.4 The California 2 model

“California #2” is a popular model against which new detection algorithms are often compared and runs in most deployed incident detection systems (Stephanedes and Hourdakis 1996). California #2 (abbreviated CA2) is a fixed-structure model that proceeds as follows:

- Let $\mathbf{Occ}(s_{up})$ denote occupancy at the upstream sensor s_{up} and $\mathbf{Occ}(s_{down})$ the same at the downstream sensor. If $\mathbf{Occ}(s_{up}) - \mathbf{Occ}(s_{down}) > T_1$, proceed to the next step.
- If $(\mathbf{Occ}(s_{up}) - \mathbf{Occ}(s_{down}))/\mathbf{Occ}(s_{up}) > T_2$, 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 $(\mathbf{Occ}(s_{up}) - \mathbf{Occ}(s_{down}))/\mathbf{Occ}(s_{down}) > T_3$, wait until the next reading. If T_3 is still exceeded, flag an alarm. The wait is introduced to cut down on false alarms and is referred to as a persistence check.

Thresholds T_1, T_2, T_3 need to be calibrated manually for each sensor site. Without access to an expert, but with plenty of data, we resorted to an exhaustive parameter grid-search as described in Sect. 5. Figure 6 shows the performance of the optimally calibrated CA2 algorithm when varying two parameters.

3.5 Dynamic Naive Bayes

We define a dynamic Bayesian network model, with a 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 (Fig. 7a).

Inspired by traffic flow theory (Kerner and Lieu 2004; Schadschneider et al. 2005), which describes analogous flow states, 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

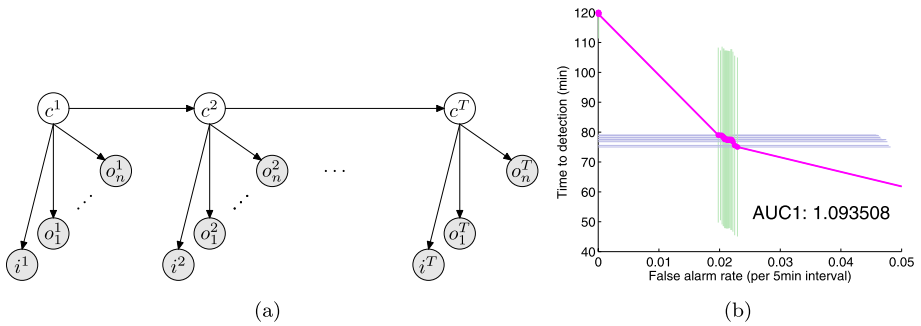


Fig. 7 The Dynamic Naive Bayes graphical model for detection and its performance characteristic. Note how the performance points cluster around FAR 0.2. This is a sign that there are few posteriors with moderate values and the posteriors form a strongly bimodal distribution. Sensitivity of such a detector will be difficult to adjust

Table 1 The “anchoring” conditional probability table

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

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 serves to anchor the rows and columns of the inter-slice transition matrix $p(s^n|s^{n-1})$ to their intended interpretation.² The inter-slice transition matrix as well as the conditional probabilities $p(o_n|s^n)$ are learned from data.

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

Unfortunately, the performance of the simple Dynamic NB models falls short of extracting the gains we expect from a detector combining multiple features. 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 between traffic flow states, a similarly weak performance was seen in a model with only 2 states. Also, the distribution of posterior is unfavorable, resulting in a detector whose sensitivity is difficult to adjust.

²The probabilities are of course somewhat arbitrary, but what is important is their relative magnitude. The imprecise specification may be impairing the detection performance of the dynamic Naive Bayes detector. While we pursue a different way of getting a better detector, the issue could be corrected by also learning the parameters of the conditional $p(i|s)$ and adopting, for instance the approach of Niculescu et al. (2006). In that work, a method is presented to learn maximum likelihood parameters subject to inequality constraints on individual probability values. On the other hand, it is only a single conditional distribution out of many in the model. Thus the “clustering” aspect of the Naive Bayes should prevail and the fixed conditional probability table really only serves to start up the EM algorithm in a favorable region of the parameter space and to assign an interpretation.

On the other hand, it is unclear how to assign a probabilistic interpretation to the SVM detector described below or how it may be integrated into a framework that considers time. Herein lies a challenge for future work: a well performing classifier that can be “dynamized”, such as a Bayesian network. We will combine the advantages of both SVMs and dynamic Bayesian networks in our integrated framework presented in Sect. 4.

3.6 SVM detector

Let us now try the support vector machine (SVM) as the detection model. We had two reasons for choosing the SVM (Mangasarian and Musicant 2000). First, in preliminary experiments it outperformed logistic regression and several variations of dynamic Bayesian network detectors (Šingliar and Hauskrecht 2006). Second, the SVM is fairly robust to irrelevant features, allowing us to include features that are weakly informative individually, but perhaps become strong predictors in aggregate. The SVM was learned in the straightforward way. Datapoints falling into the intervals labeled as “incident” in the TMC records were assigned class 1, the remaining datapoints class -1 . Misclassification cost was selected so as to balance for unequal class sizes: if there are N instances of class 1 and M instance of class -1 , then the misclassification of “ -1 ” as “1” costs N/M and 1 vice versa. This prevents the SVM from learning the “majority class” classification rule, as it often does with very skewed datasets.

The performance of the SVM detector using different feature sets can be seen in the curves and the associated $AUC_{1\%}$ values in Fig. 8. It appears that for our data, the direct sensor readings (speed, volume, occupancy) provide most of the detection leverage. Addition of the spatial difference (and proportion) features affects the performance minimally. The temporal difference features do bring a small improvement, albeit one that fails to confirm the perception of temporal difference as a crucially important feature (Martin et al. 2000). This could be in part explained by the fact that our data are 5 minute averages and the sharp temporal effects often thought critically important for detection are somewhat averaged out. A detector using the features of the CA2 algorithm is included for comparison. The results confirm our intuition: the SVM detectors using multiple features outperform that using only CA2 features (the comparison to CA2 itself follows later).

3.7 Persistence checks

False alarm rate can be traded for detection time with the alarm signal post-processing technique known as persistence check. k -persistence check requires that the alarm condition persist for k additional time periods before the alarm is raised. Note that CA2 already has a built-in 1-persistence check in its last step. We experimented with the optimal (in the sense of minimizing $AUC_{1\%}$) choice of k for the SVM detector with the basic measurement features (on the training site data).

The idea of persistence check can be applied to any detector. The expected effect is that the persistence check will eliminate many fluke occurrences that resemble an accident and thus lower the FAR of the new detector. Figure 9 shows the performance statistics for the SVM-based detector without and with persistency filtering. The results clearly illustrate the benefit of persistency filtering. SVM attains its best performance at $k = 1$ and all subsequent SVM experiments are therefore conducted using that persistence value. However, it is possible to “overdo” a persistent check and the performance to suffer on account of the delayed or missed detections.³

³Note how the graph for $k = 1$ “plateaus” at a slightly higher value of time-to-detection than the other two graphs. We believe this is an artifact of the TTD penalty imposed for missing a detection completely, which

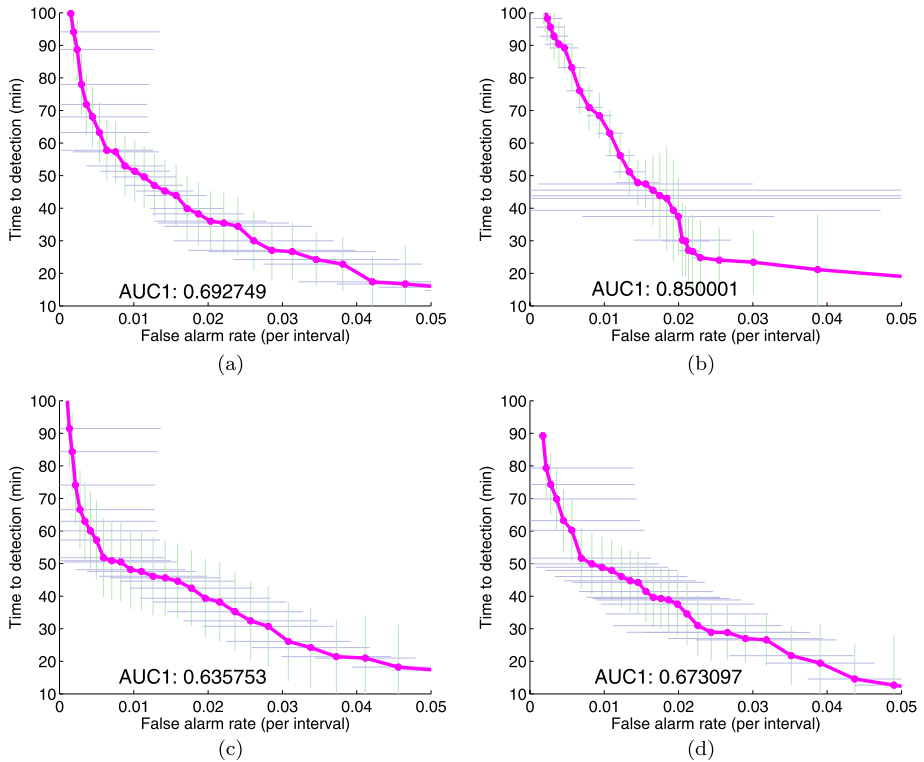


Fig. 8 Performance of the SVM model, for different feature sets. The features are: **(a)** All readings for the sensor. **(b)** California 2 features (the occupancy ratios). **(c)** All of current and previous time interval measurements. **(d)** All current measurements together with differences and 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 bound on the true performance (Bach et al. 2005). Thus, the AMOC curve is slightly pessimistic of SVM performance. The area under the portion of the curve up to 1% FAR is reported as AUC1

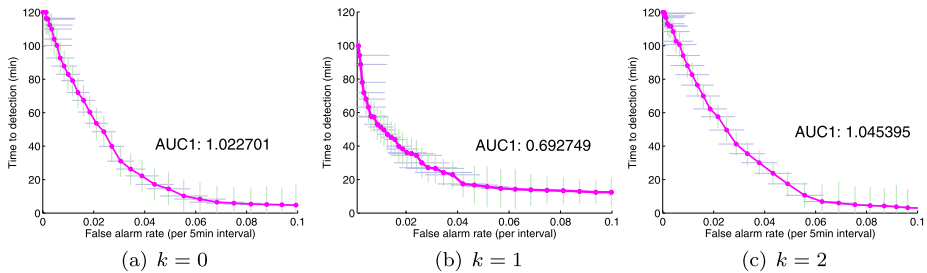


Fig. 9 Performance with persistence checks: **(a)** the SVM detector without the persistence check, **(b)** the SVM detector with 1-persistence check—incident must be detected in two consecutive datapoints for an alarm to be raised, **(c)** the SVM detector with 2-persistence check

increases as we go from $k = 0$ to $k = 1$. The curve continues to descend beyond the extent of the FAR axis.

Table 2 Predictor performance summarized (cross-validated average area under AMOC curve). Smaller AUC is better

Features	Detector	AUC
basic	Occ (s_{up}, t_0)	1.200
	Spd (s_{up}, t_0)	1.009
temporal	Occ ($s_{up}, t_0 - t_1$)	0.778
	Spd ($s_{up}, t_0 - t_1$)	1.200
spatial	Occ ($s_{up} - s_{down}, t_0$)	1.046
	Spd ($s_{up} - s_{down}, t_0$)	0.987
learning	t -test (Vol (s_{up}), $N = 10$)	1.039
	regression (Occ (s_{down}), $N = 10$)	0.849
	Dynamic NB (all measures)	1.094
	SVM (all measures)	0.693

In summary, the SVM detector turned out to be the best performer on the original data—see the summary in Table 2. But, we know the data is not perfect! How does it affect the learner’s performance? The following section answers that question.

4 Label realignment model

Our objective is to detect the accident as soon as its impact *manifests* in sensor readings. This time will always lag the time the accident actually *happened*. The lag amount depends, among other factors, on the capacity utilization of the roadway and the relative position of the sensor and accident location. The time that the incident is *reported* to the TMC and logged in the database may precede or follow the time of manifestation. Differences between these three times lead to label misalignment.

There are two things that the detector can latch onto: the short period when the accident’s impact builds up (e.g. speed falls) around the sensor, and the longer steady state condition with lowered speeds or jammed traffic. To optimize detection time, we should focus the detector at the transient period. The transient period is very short and any misalignment will cause the accident start label to fall outside of it. It is therefore crucial for supervised learning that the label be precisely aligned with the observed impact of the accident. The end-of-incident labels are less important: by the time the incident is cleared, the emergency response has already taken place. We do not attempt to align incident clearing times.

In the supervised approach, we use both positive and negative learning instances. Because of the alignment problem, we take for negative instances such points in time that are far enough from any reported incidents. Conversely, we need to take a wide enough window around a reported incident to make sure the incident indeed manifests in the selected subsequence. We need a method to correct the alignment of incident labels in the positive training instances so that the learned model accuracy may improve. The method also needs to scale, in terms of human labor required, to networks with large numbers of detection sites.

As we go to $k = 2$, the curve changes shape—“squeezes to the left”, bringing the events that give rise to that descending portion into the range of the graph.

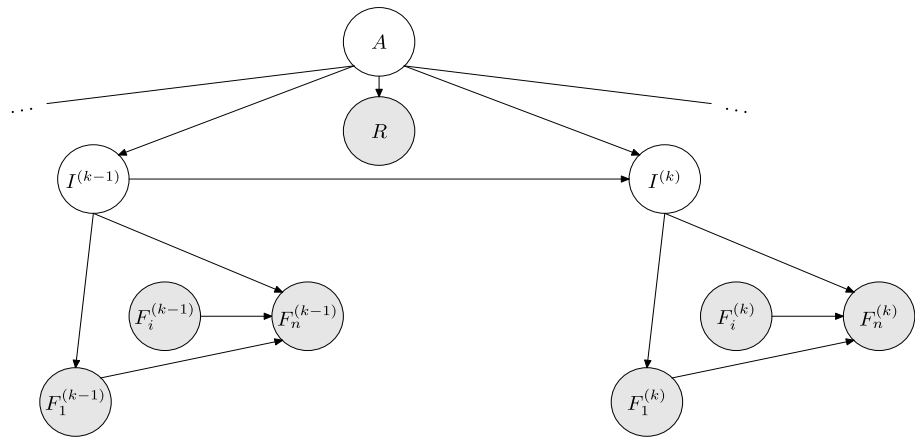


Fig. 10 Two slices of the temporal probabilistic model for realignment. As usual, shaded nodes represent observed random variables; unshaded nodes correspond to latent variables. There are a total of L slices; the superscript (k) denotes the variable’s instantiation in the k -th time slice

4.1 A model of incident sequences

Consider a single positive sequence S of traffic feature vectors. An incident start label r denotes the point in sequence S where the single incident is reported to occur. The label realignment task is to output the label ℓ pointing where the incident truly began to manifest in S . For label realignment, we model the sequence of feature vectors with a special dynamic Bayesian network model, shown in Fig. 10. In the model, A represents the true accident time and takes on a value from $\{1, \dots, L\}$, where L is the sequence length. Each *impact* variable $I^{(k)}$ is a binary indicator of incident impacting the traffic flow at time k . Each I is a part of the intra-slice Bayesian network that models the interaction between the traffic measurement features F_1, \dots, F_n . We place no restrictions on the within-slice network in general. In order to keep the model presentation simple, we do not draw arrows between the features in subsequent slices. However, features may depend on values at other nearby timepoints; for instance the “occupancy derivative” $F(t) = \mathbf{Occ}(t) - \mathbf{Occ}(t - 1)$ depends on the previous measurement.

The variables A and $I^{(k)}$ have a special relationship, expressed in their probability distributions. First, we express the lack of prior knowledge about A by defining the prior $P(A)$ to be the uniform distribution on the set $\{1, \dots, L\}$. Second, the conditional distribution is also fixed, expressing that once an incident starts impacting traffic, it continues to do so:

$$\begin{aligned}
 p(I^{(k)} = 1 | A = k', I^{(k-1)} = 1) &= 1, \\
 p(I^{(k)} = 1 | A = k', I^{(k-1)} = 0) &= 1 \quad \text{if } k = k', \\
 &= 0 \quad \text{otherwise.}
 \end{aligned}
 \tag{1}$$

We can afford this simplification because we only want to model the accident onset and disregard the accident clearing event.

The report time R depends on the true accident time and is assumed to obey a conditional Gaussian distribution: $P(R|A = k) \sim N(k + \mu, \sigma^2)$, with μ, σ identical for all k . Equivalently, the amount of misalignment has a stationary Gaussian distribution: $R - A \sim N(\mu, \sigma^2)$.

4.2 Inference for realignment

We perform inference in this model in its unrolled form. Basic variable elimination is the best suited inference method for the realignment model. It deals well with the unusual distributions and is also efficient for this model, because the special form of the inter-slice probability distribution simplifies the inference task—there are only L indicator sequences with $p(I_1, \dots, I_L) > 0$ to sum over. Dynamic networks with such sink-state variables are in general amenable to more efficient inference exploiting the “persistence” property (Šingliar and Dash 2008).

Using the probability distribution p defined by the above model, the label alignment task can be formulated as a posterior mode problem. Given that the data places the incident start label at r , we reassign the label to ℓ , so that

$$\ell = \operatorname{argmax}_k p(A = k | R = r, \mathbf{F}^{(1)}, \dots, \mathbf{F}^{(L)}), \tag{2}$$

where $\mathbf{F}^{(t)} = (F_1^{(t)}, \dots, F_n^{(t)})$ is the t -th observation vector.

4.3 Learning and transfer to new locations

Now, we must parameterize a separate model for each sensor pair defining a highway segment (site). Let A denote the single calibration (training) site and let $B_j, j = 1, \dots, S$ be the test sites. While one could learn the model in a fully unsupervised manner with the EM algorithm (Dempster et al. 1977), there is little guarantee that the learning would converge to the intended interpretation. Instead, we first learn p^A , the sequence model for A , from manually aligned data. Manual alignment gives us a fully observed dataset $\mathbf{X}^A = (\mathbf{F}^A, R^A, I^A, A^A)$ and maximum likelihood learning becomes trivial:

$$\Theta_{ML}^A = \operatorname{argmax}_{\Theta} p(\mathbf{X}^A | \Theta). \tag{3}$$

Then, inference in the model parameterized with Θ_{ML}^A can be applied to realign the labels for the B -sites where the manual annotation is unavailable. Of course, accident impact at each site B_j differs from that of the site A . The resulting labeling will be imperfect, but it still provides a good initial estimate. The EM-algorithm for estimation of Θ^{B_j} can proceed from there with a smaller risk of converging to an undesirable local optimum. Additionally, the sufficient statistics obtained in the estimation of Θ^A are stored and used to define the conjugate prior over Θ^{B_j} . Thus the resulting parameterization of a testing site model is a maximum a posteriori (MAP) estimate:

$$\Theta_{MAP}^{B_j} = \operatorname{argmax}_{\Theta} p(\mathbf{X}^{B_j} | \Theta) p(\Theta | \Theta_{ML}^A). \tag{4}$$

In the EM algorithm that estimates $\Theta_{MAP}^{B_j}$, the expectation step corresponds to inference of the unobserved labels A^{B_j} and I^{B_j} . The maximization step re-estimates the parameters of the conditional distributions $p(R|A)$ and $p(F_i|I)$. We consider the EM converged if the labeling (the posterior modes, see (2)) does not change in two consecutive iterations. For our dataset, the EM always converges in 5 or fewer iterations.

Table 3 Sites included in the evaluation, with number of incidents and indication of the roadway. Some roadways have multiple incident sites

Site	S_{Train}	S_{Test1}	S_{Test2}	S_{Test3}	S_{Test4}	S_{Test5}	S_{Test6}
Road	I376W	RT-28S	I279N	I279S	I376E 2	I376E 1	RT-28N
inc	145	100	97	92	107	84	100

5 Experimental evaluation

In this section we describe the experimental setup and report the results. All statistics reported are averages and standard deviations across 10 cross-validation splits, even where error bars were dropped for sake of readability. Error bars in all graphs represent one standard deviation.

5.1 Evaluation framework

We evaluated our model on six sites with the highest numbers of accident reports in the area. The incident reports at S_{Train} were manually aligned to the incident manifestations in the data. The manual realignment was also aided by the free-text incident descriptions from the TMC database.

We evaluate the models under the cross-validation framework. The dataset consists of three long sequences per sensor, one for each of the three traffic variates. We divide the data into train/test splits 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. In the rare case that more than one incident should occur in or in the vicinity of a sequence, we exclude such sequence from the data. 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, with $I_{train} : I_{test} \approx 70 : 30$, sequence length $L = 100$ and $C = 50$ for training. For testing, instead of choosing a fixed C , we make sure the proportion of positive (incident) instances approximates the proportion observed in the entire dataset.

In each cross-validation fold, the positive training sequences are realigned and the quality of the detection is evaluated on the testing set, using the *original incident labeling*. While testing on the original labeling will result in a measurement that is somewhat off, the skew will be consistent across detectors and relative comparisons remain valid. If we evaluated on the realigned data, we would run the risk of having both the realignment algorithm and the detector make the same mistake in lockstep, losing touch with the data.

5.2 Detection and alignment model specifics

We evaluate the alignment model with two choices of the interaction model for traffic flow features (the intra-slice Bayes network).

Naive Bayes slices To represent incident impact on traffic, we first use a Naive Bayes intra-slice model with binary indicator I and two features, F_1 : the difference in occupancy at the upstream sensor in the previous and following interval and F_2 : the same difference in speed. Both features are assumed to follow a conditional Gaussian distribution. This model makes do with very little information and strong independence assumptions.

TAN slices Secondly, relaxing the independence assumptions made by the NB model, we model the traffic impact of an incident with a Tree-Augmented Naive Bayes slice structure (Friedman et al. 1997). The Tree-Augmented Naive Bayes (TAN) model has been shown to improve, often dramatically, the modeling precision over the plain NB model; yet it retains favorable inferential complexity. In this arrangement, the class variable I again corresponds to the incident indicator. As features, we use all traffic quantities and their spatial and temporal differences and proportions. Since all these are continuous, we will again use a Gaussian random variable for each quantity.

The learning of a TAN model has two distinct phases: structure and parameter learning. The structure is learned so as to obtain the tree structure that has the maximum potential to capture dependencies present in the data. Chow and Liu (1968) prove that this is achieved by a maximum-weight spanning tree, where edges are weighted by the mutual information. For a Gaussian, the mutual information is

$$I(x_u, x_v) = -\frac{1}{2} \log(|\hat{\Sigma}_k|/(\sigma_u^2 \sigma_v^2)),$$

where (x_u, x_v) is an arbitrary pair of random variables, $\hat{\Sigma}_k$ is the maximum likelihood estimate of the 2×2 -element covariance matrix of (x_u, x_v) ; and σ_u^2 and σ_v^2 are its diagonal elements.

Once the structure is fixed, standard expected loglikelihood ascent (EM) can be used.⁴ We use the Bayes Network Toolbox (Murphy 2001) to actually represent the network and implement the learning.

To fit TAN in our framework, we first learn the structure and parameters on the hand-aligned data from the training site. The entire dataset is used, which ensures greater stability of the learned tree structure M^A . The structure is fixed and parameters Θ^A are learned from the training data only. In the transfer phase for the new sites, the network structure is kept ($M^B = M^A$) and only parameters Θ^B are learned. In principle, the structure M^B can also be learned, starting from the training site model M^A . However, we found experimentally that the learned structures M^B exhibit too wide a variance if structure learning is allowed to take place in the transfer step, leading us to conclude there is simply not enough data for structure learning. Moreover, only positive—by definition anomalous—instances are included in the transfer step learning! For better clarity, the learning setup pseudocode is shown in Algorithm 1.

The TAN model contains all of the features mentioned above (the measurements, their spatial and temporal differences and proportions). It also makes as few independence assumptions as possible for it to remain tractable. Thus it lies on the “opposite end of the tractable spectrum” as Naive Bayes. If we find similar levels of performance in the NB and TAN models, we may conclude with a degree of confidence that realignment works because of the temporal dynamic Bayes net structure, as we expected.

⁴The prior $p(I)$ must be discarded as the learned network is plugged into the larger dynamic network, where $p(I^k|A)$ is fixed. It is anyway incorrect for the realignment task, where the proportion of positive instances is much higher.

Algorithm 1 Evaluation setup for model transfer to new site

```

 $M^A \leftarrow \text{LearnStructure}(\text{AllData}(\text{OrigSite}))$ 
 $\Theta^A \leftarrow \text{LearnMLParams}(\text{AllData}(\text{OrigSite}))$ 
for  $\text{fold} = 1$  to 10 do
   $[\text{TrainData}_{\text{fold}}, \text{TestData}_{\text{fold}}] \leftarrow \text{Split}(\text{AllData}(\text{NewSite}))$ 
   $\Theta^B \leftarrow \text{LearnMAPParams}(\text{TrainData}_{\text{fold}}, M^A, \Theta^A)$ 
   $\text{NewLabels} \leftarrow \text{argmax}_A p(A^B | \text{TrainData}_{\text{fold}}; \Theta^B, M^A)$ 
   $\text{Detector}_{\text{fold}} \leftarrow \text{LearnSVM}(\text{TrainData}_{\text{fold}}, \text{NewLabels})$ 
end for
 $\text{AMOC} \leftarrow \text{Evaluate}(\text{Detector}_{\cdot}, \text{TestData}_{\cdot})$ 

```

Inference in TAN model can still be performed in time linear in L and n , the number of features. Recall we are after the posterior $p(A|\mathbf{F}, R)$. By Bayesian network probability decomposition,

$$p(A|\mathbf{F}, R) \propto \sum_{\mathbf{I}} p(A)p(R|A)p(\mathbf{I}|A) \prod_{k=1}^L p(\mathbf{F}^k|I^k).$$

Moreover, the term $p(\mathbf{I}|A)$ is deterministic and the prior on A is uniform:

$$p(A|\mathbf{F}, R) \propto p(R|A) \prod_{k=1}^L p(\mathbf{F}^k|I^k(A)).$$

(See also Šingliar and Dash 2008 for a more general view of such simplifications of inference.) The probabilities $p(\mathbf{F}^k|I^k)$ for $I^k = 0$ and $I^k = 1$ are easily obtained by definition of Bayesian network probability when observed, and using standard inference when missing. The TAN structure ensures that $p(\mathbf{F}^k|I^k)$ is tractable.

California 2 The CA2 algorithm is normally tuned by experts who choose the three thresholds. Since we did not have services of an expert, we found the parameterization by an exhaustive procedure trying all possible settings of the three parameters on a discrete grid covering a wide range of parameter values. The “best performance” for the purpose of parameterization was defined as the best DR at a fixed FAR of 1%. This was an extremely time-consuming procedure that is impractical for a metropolitan network with hundreds of sensors, not to mention uninteresting from the learning perspective.

5.3 Experimental results

Naive Bayes slices The root of the mean squared difference between the hand-labeled incident manifestations and the recorded events is approximately 8.5 intervals. After automatically re-aligning the recorded events with the incidents, the RMS difference decreases to approximately 2.2 intervals. The decrease in training error affirms that the model indeed picks up the accident effect.

The average amount of misalignment at the training site is only 2.2 minutes (incidents are on average logged 2.2 minutes after they become observable in data), but with a standard deviation of more than 43 minutes. This is a serious amount of misalignment, it implies that the label position is on average off by 8 or 9 time steps.

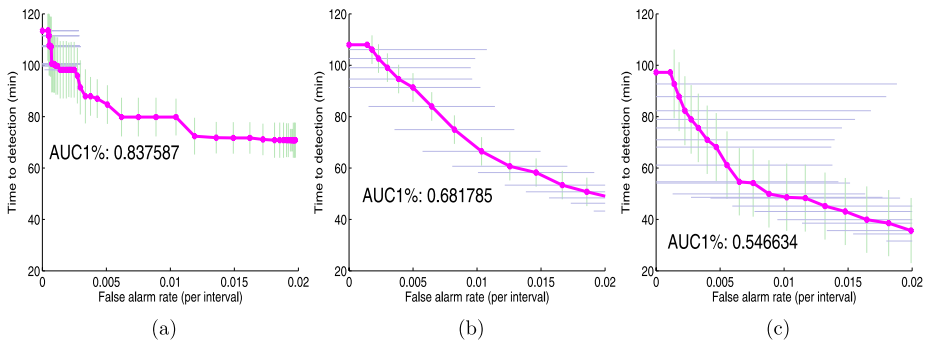


Fig. 11 Train site A with human-labeled data. Detection performance of (a) California 2, (b) SVM learned on original labeling, (c) SVM learned on the data relabeled by the “Naive Bayes” dynamic model

Table 4 Summary of the $AUC_{1\%}$ performance statistics for the three detection algorithms and six evaluation sites. Some sites are more amenable to automatic detection, but a consistent improvement is noted from CA2 to SVM on original data to SVM on realigned data

Area under first 1% of FAR range		Site						
		Train	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Road		376W	28S	279N	279S	376E 2	376E 1	28N
Method	CA2	0.838	0.451	1.177	1.180	1.108	0.726	1.081
	SVM/orig	0.682	0.179	0.807	0.474	0.613	0.740	0.808
	SVM/NB	0.547	0.149	0.763	0.389	0.561	0.712	0.771
	SVM/TAN	0.553	0.152	0.783	0.294	0.567	0.698	0.720

The quality of the resulting labels is most relevantly measured by the improvement in the $AUC_{1\%}$ performance metric of a classifier learned on the realigned data. The $AUC_{1\%}$ values for the four methods (CA2, SVM, SVM after NB relabeling, SVM after TAN relabeling) are summarized in Table 4. The standard deviation of TTD and FAR obtained together with the 10-fold cross-validated averages are represented by the vertical and horizontal bars, respectively, around each operating point on the curves in Fig. 11. The table shows that the SVM detector learned on the original data consistently improves over the CA2 method for every testing site. Similarly, the SVM detector learned on the label-realigned data realizes an improvement over the original SVM detector. The absolute performance varies significantly between testing sites as it depends on a number of site specifics: the distance between the accident site and the upstream sensor, volume of traffic, the presence of a shoulder lane where the vehicles may be removed from the flow of traffic, etc.

TAN slices Realignment with the TAN network also resulted in an improvement over learning with original misaligned data in all but one case. In comparison with the Naive Bayes model, the experiments are inconclusive. However, one can discern more volatility in TAN performance. We believe that the more complex TAN model begins to suffer from the limited amount of data where the simpler NB consistently succeeded.

6 Conclusions

Our experiments showed that no single traffic feature may provide the most discriminative performance. Detectors that combine multiple features top out at a better performance.

All detectors (at least those discussed here) produce a “signal”, whether it is the distance from mean measured in standard deviations, posterior probability of incident from a graphical model, or the distance of the point in measurement space from the separating hyperplane. A good detector will have a signal that is almost uniformly distributed in its respective range. A well spread-out distribution of signal allows for effective adjustment of the detector’s sensitivity and specificity. Detectors that do not have this property, as we have seen in case of the dynamic Bayes net detector, will be unable to achieve some operating points, especially the desirable ones with low false alarm rates.

Supervised learning is a viable approach to construction of incident detection algorithms. We have shown it surpasses the performance of unsupervised anomaly detection techniques and easily leads to detectors that outperform traditional hand-crafted detectors. With sufficient data now available, it can do away with the problem of manual tuning and re-tuning of the detectors to adapt to new deployment locations and changing traffic patterns.

However, data obtained from such complex systems is inherently noisy. We proposed an algorithm that deals successfully with noise in event label timing and demonstrated that it improves the data quality to allow more successful learning of incident detectors.

Of course, a number of specific questions about our approach remain open. One could devise finer incident models and offset distributions; relax the assumption of independence of time-to-recording and incident impact severity—a more severe accident is perhaps more easily noticed. Explicitly modeling time-of-day and the expected traffic pattern looks especially promising as it permits the definition of an “unexpected” congestion, presumably more indicative of an accident. On the other hand, we advocate a cautious approach to more complex models. While data on traffic flow is plentiful, incidents and related data are (fortunately) relatively sparse, which makes more complex models vulnerable to overfitting, as we began to notice with the TAN realignment model, especially with regard to structure learning.

While the realignment algorithm was motivated by and presented in context of incident detection, it is generally applicable to situations where time of events is marked noisily in data streams. For instance, similar uncertainty in labeling alignment accompanies detection of intonation events in speech recognition (Taylor 2000).

Acknowledgements The authors gratefully acknowledge the domain expertise of Dr. J.S. Lin, Associate Professor of Civil Engineering, University of Pittsburgh; and the support of Dr. Louise K. Comfort, Professor of Public and International Affairs, also of University of Pittsburgh. We are also grateful for the time and effort on the part of the anonymous reviewers, whose suggestions helped very much to improve the paper. This research was supported by NSF grants ANI-0325353 and CMS-0416754. Thank you!

Appendix: Supplemental material

Not all performance curves were included in the paper, for sake of readability. <http://www.cs.pitt.edu/~tomas/papers/MLJ07/> is the online location for all such supplemental references.

References

Antoniades, C. N., & Stephanedes, Y. J. (1996). Single-station incident detection algorithm (SSID) for sparsely instrumented freeway sites. In *Transportation engineering*.

- Bach, F., Heckerman, D., & Horvitz, E. (2005). On the path to an ideal ROC curve: considering cost asymmetry in learning classifiers. In R. G. Cowell & Z. Ghahramani (Eds.), *Proceedings of AISTATS05* (pp. 9–16).
- Busch, F., & Fellendorf, M. (1990). Automatic incident detection on motorways. *Traffic Engineering and Control*, 31(4), 221–227.
- Chow, C. J. K., & Liu, C. N. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3), 462–467.
- Cooper, G., Dash, D., Levander, J., Wong, W.-K., Hogan, W., & Wagner, M. (2004). Bayesian biosurveillance of disease outbreaks. In Proceedings of the 20th annual conference on uncertainty in artificial intelligence (UAI-04) (pp. 94–103).
- Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, 5, 142–150.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood for incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39, 1–38.
- Dia, H., Rose, G., & Snell, A. (1996). Comparative performance of freeway automated incident detection algorithms.
- Fawcett, T., & Provost, C. (1999). Activity monitoring: noticing interesting changes in behavior. In *Proceedings of the fifth international conference on knowledge discovery and data mining (KDD-99)*.
- Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29, 131–163.
- Kerner, B. S., & Liu, H. (2004). *The physics of traffic: empirical freeway pattern features*. Berlin: Springer.
- Mangasarian, O. L., & Musicant, D. R. (2000). *Lagrangian support vector machine classification* (Technical Report 00-06). Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, June 2000. <http://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-06.ps>.
- Martin, P. T., Perrin, H. J., & Hansen, B. G. (2000). *Incident detection algorithm evaluation* (Technical Report UTL-0700-31). Utah Traffic Laboratory, July 2000.
- Michalopoulos, P. G., Jacobson, R. D., Anderson, C. A., & DeBruycker, T. B. (1993). Automatic incident detection through video image processing. In *Traffic engineering and control*, February 1993.
- Murphy, K. (2001). The Bayes net toolbox for Matlab. *Computing Science and Statistics*, 33, 1024–1034.
- Niculescu, R. S., Mitchell, T. M., & Bharat Rao, R. (2006). Bayesian network learning with parameter constraints. *Journal of Machine Learning Research*, 7, 1357–1383.
- Parkanyi, E., & Xie, C. (2005). *A complete review of incident detection algorithms and their deployment: what works and what doesn't* (Technical Report NETCR37). New England Transportation Consortium.
- Provost, F. J., & Fawcett, T. (1997). Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In *Knowledge discovery and data mining* (pp. 43–48).
- Ritchie, S. G., & Abdulhai, B. (1997). *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).
- Rose, G., & Dia, H. (1995). Freeway automatic incident detection using artificial neural networks. In *Proceedings of the international conference on application of new technology to transport systems* (Vol. 1, pp. 123–140).
- Schadschneider, A., Pöschel, T., Kühne, R., Schreckenberg, M., & Wolf, D. E. (Eds.) (2005). *Traffic and granular flow*. Berlin: Springer.
- Stephanedes, Y. J., & Hourdakis, J. (1996). *Transferability of freeway incident detection algorithms* (Technical Report Transportation Research Record 1554). Transportation Research Board, National Research Council.
- Šingliar, T., & Hauskrecht, M. (2006). Towards a learning incident detection system. In *ICML 2006 workshop on machine learning algorithms for surveillance and event detection*.
- Šingliar, T., & Hauskrecht, M. (2007). Learning to detect adverse traffic events from noisily labeled data. In *LNCS: Vol. 4702. Proceedings of principles and practice of knowledge discovery in databases PKDD 2007* (pp. 236–247). Berlin: Springer.
- Šingliar, T., & Dash, D. H. (2008). Efficient inference in persistent dynamic Bayesian networks. In *Proceedings of the 24th conference on uncertainty in artificial intelligence, UAI-2008* (pp. 494–502).
- Taylor, P. A. (2000). Analysis and synthesis of intonation using the Tilt model. *Journal of the Acoustical Society of America*, 107(3), 1697–1714.
- Zhang, K., & Taylor, M. A. P. (2006). Towards universal freeway incident detection algorithms. *Transportation Research. Part C, Emerging Technologies*, 14(2), 68–80.