Learning to Detect Adverse Traffic Events from Noisily Labeled Data

Tomáš Šingliar and Miloš Hauskrecht

Computer Science Dept, University of Pittsburgh, Pittsburgh, PA 15260
{tomas,milos}@cs.pitt.edu

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 network measurements. First, we show that a rather straightforward supervised learner based on the SVM model outperforms a fixed detection model used by state-of-the-art traffic incident detectors. Second, we seek further improvements of learning performance by correcting misaligned incident times in the training data. The misalignment is due to an imperfect incident logging procedure. 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 consistently leads to improved detection performance in the low false positive region.

1 Introduction

The cost of highway accidents is significantly reduced by prompt emergency response. With real-time traffic flow data, automated incident detection systems promise to detect accidents earlier than human operators. Earlier response and accident impact mitigation lead to significant savings of money and life.

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 [1]. 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. Transferability of detection algorithms is a central concern in automatic incident detection [2]. We investigate how machine learning can help design transferable detection algorithms.

Machine learning approaches to automatic incident detection are made possible by the wealth of data collected by networks of traffic sensors installed nowadays on many highways. Models that can be automatically tuned from data could reduce or eliminate costly recalibrations and improve performance [3,4,5,6]. We experiment with SVM-based detection and show it easily outperforms the optimally calibrated standard model (California 2).

J.N. Kok et al. (Eds.): PKDD 2007, LNAI 4702, pp. 236–247, 2007.

However, the learning framework can be further improved. In particular, the labels for incident data 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 badly aligned data yields a suboptimal detector.

We approach the alignment problem using machine learning methods as well. We propose a new dynamic Bayesian network [7] that models the misalignment problem probabilistically with respect to traffic flow quantities and the label position. We train the model on the manually realigned data from a *single* highway segment. Once learned, the model can be transferred to other highway segments to correct the incident labeling. 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.

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 (Figure 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 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.

[1] The 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.

Fig. 1. 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 12^{th}, 6:40am), 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.

2.2 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 invocations. The detection rate (DR) is the number of correctly detected incidents divided by the number of incidents that actually occurred. Receiver operating characteristic (ROC) curves [8] 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 [9]. 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" 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 target performance at which a system is considered useful depends chiefly on its users. A study [5] 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 present the detection models that operate on the original data supplied by the TMC.

3.1 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 [2]. California #2 (abbreviated CA2) is a fixed-structure model that proceeds as follows:

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

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 Section 5.

3.2 Model Learning and Features

The CA2 algorithm uses a surprisingly limited set of features. Could a better detection performance be achieved if the detector took advantage of multiple features? And which features? Clearly, the readings at the upstream sensor s_{up} and the downstream sensor s_{down} at the time of detection should be included. Sharp changes in traffic flow may also indicate an accident. Therefore, we include features computed as differences and proportions of the traffic variates to their previous value. Finally, unlike a benign congestion, an accident should cause radically different flow characteristics at the upstream and downstream sensors. This motivates the inclusion of features that correlate the measurements spatially, as differences and proportions of the respective measurements at upstream and downstream sensors.

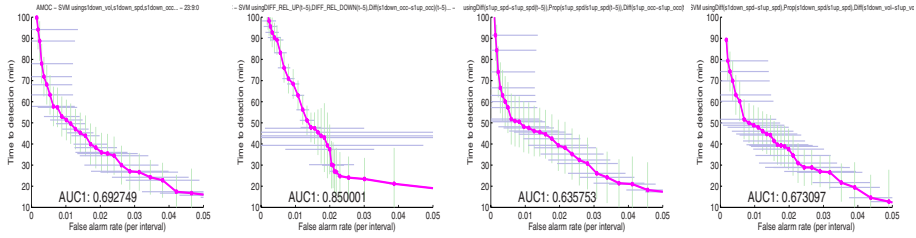

Fig. 2. Performance of the SVM model, for different feature sets. The features are: (a) All readings for the sensor. (b) California 2 features (the occupancy ratios). (c) All of current and previous step measurements. (d) All current measurements together with differences and proportions of the corresponding 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 [10]. For each value of the detection threshold, we compute the average FAR and TTD over 10 train/test splits and draw the graph point as well as both of the corresponding error bars. The area under the portion of the curve up to 1% FAR is reported as AUC1.

3.3 SVM Detector

Having defined the potentially informative features, we pick a learner from the palette of learning tools. We had two reasons for choosing the SVM model [11]. First, in preliminary experiments it outperformed logistic regression and several variations of dynamic Bayesian network detectors [12]. 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 data were assigned class 1, the remaining datapoints class -1. Misclassification cost was selected 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.

The performance of the SVM detector using different feature sets can be seen in the curves and the associated $AUC_{1\%}$ values in Figure 2. 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 an important feature [1]. This could be in part explained by the fact that our data are 5 minute averages and the sharp temporal effects 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).

3.4 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). Best performance is attained at $k = 1$ and all SVM experiments are therefore conducted using that persistence value.

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 locations. 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 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 is 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.

By definition, misalignment can only occur in positive instances, i.e. those sequences that contain an incident. We need a method to correct the alignment of incident labels in the training set so that the learned model accuracy may improve.

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 Figure 3. In the model, A represents the true accident time and takes on a value from $\{1, \ldots, L\}$, 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

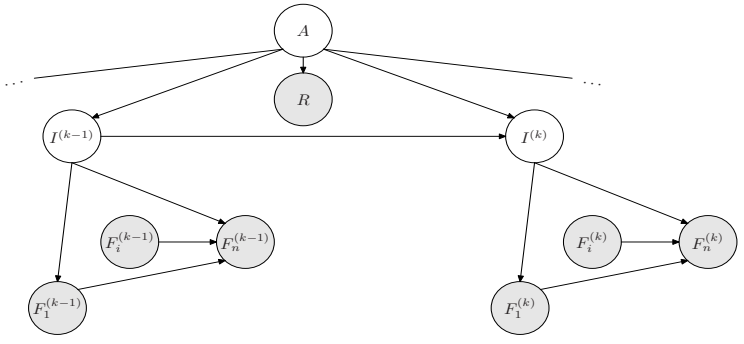

Fig. 3. Two slices of the temporal probabilistic model for realignment. As usual, shaded nodes represent observed random variables; unshaded nodes correspond to latent variables. There are a total of L slices; the superscript (k) denotes the variable's instantiation in the k-the time slice.

traffic measurement features $F_1, \ldots, F_n$. 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" $F(t) = Occ(t) - Occ(t-1)$ 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 $\{1, \ldots, L\}$. Second, the conditional distribution is also fixed, expressing that once an incident starts impacting traffic, it continues to do so:

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

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 $p(I_1, \ldots, I_L) > 0$ to sum over.

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

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

where $\mathbf{F}^{(t)} = (F_1^{(t)}, \ldots, F_n^{(t)})$ 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, $j = 1, \ldots, S$ be the test sites. While one could learn the model in a fully unsupervised manner with the EM algorithm [13], 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:

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

Then, inference in the model parameterized with Θ^A_{ML} 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:

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

In the EM algorithm that estimates $\Theta^{B_j}_{MAP}$, 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 Equation 2) does not change in two consecutive iterations. For our dataset, the EM always converges in less than 5 iterations.

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.

Table 1. Sites included in the evaluation, with number of incidents

Site	S_{Train}	S_{Test1}	S_{Test2}	S_{Test3}
# incidents	145	100	97	92

5.1 Evaluation Framework

We evaluated our model on four 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

To represent incident impact on traffic, we 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.

Table 2. Summary of the $AUC_{1\%}$ performance statistics for the three detection algorithms and four evaluation sites. Some sites are more amenable to automatic detection, but consistent improvement is noted from CA2 to SVM on original data to SVM on realigned data.

Detector	Site			
	S_{Train}	S_{Test1}	S_{Test2}	S_{Test3}
CA2	0.838	0.451	1.177	1.180
SVM/orig	0.682	0.179	0.807	0.474
SVM/realign	0.547	0.149	0.763	0.389

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

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.

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 three methods (CA2, SVM, SVM after relabeling) are summarized in Table 2. 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 Figure 4. 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.

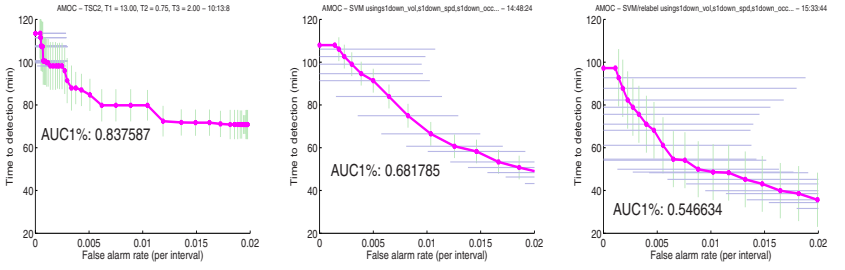

Fig. 4. Train site A with human-labeled data. Detection performance of (a) California 2 (b) SVM learned on original labeling, (c) SVM learned on the relabeled data.

6 Conclusions

Learning is a viable approach to construction of incident detection algorithms. It 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, the 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.

While the realignment algorithm was motivated by and presented in context of incident detection, it is generally applicable to situations where events are marked noisily in data streams. For instance, similar uncertainty in labeling alignment accompanies detection of intonation events in speech recognition [14].

References

1. Martin, P., Perrin, H., Hansen, B.: Incident detection algorithm evaluation. Technical Report UTL-0700-31, Utah Traffic Laboratory (July (2000)
2. Stephanedes, Y., Hourdakis, J.: Transferability of freeway incident detection algorithms. Technical Report Transportation Research Record 1554, Transportation Research Board, National Research Council (1996)
3. Dia, H., Rose, G., Snell, A.: Comparative performance of freeway automated incident detection algorithms (1996)

4. Rose, G., Dia, H.: 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 (1995)
5. Ritchie, S.G., Abdulhai, B.: 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)
6. Parkanyi, E., Xie, C.: A complete review of incident detection algorithms and their deployment: What works and what doesn't. Technical Report NETCR37, New England Transportation Consortium (2005)
7. Dean, T., Kanazawa, K.: A model for reasoning about persistence and causation. Computational Intelligence 5, 142–150 (1989)
8. Provost, F.J., Fawcett, T.: Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Knowledge Discovery and Data Mining, pp. 43–48 (1997)
9. Cooper, G., Dash, D., Levander, J., Wong, W.K., Hogan, W., Wagner, M.: Bayesian biosurveillance of disease outbreaks. In: Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI-04), pp. 94–103. AUAI Press, Arlington, Virginia (2004)
10. Bach, F., Heckerman, D., Horvitz, E.: On the path to an ideal ROC curve: Considering cost asymmetry in learning classifiers. In: Cowell, R.G., Ghahramani, Z. (eds.) Proceedings of AISTATS05, pp. 9–16 (2005)
11. Mangasarian, O.L., Musicant, D.R.: 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
12. Singliar, T., Hauskrecht, M.: Towards a learning incident detection system. In: ICML 2006 Workshop on Machine Learning Algorithms for Surveillance and Event Detection (2006)
13. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood for incomplete data via the EM algorithm. Journal of Royal Statistical Society 39, 1–38 (1977)
14. Taylor, P.A.: Analysis and synthesis of intonation using the Tilt model. Journal of the Acoustical Society of America 107(3), 1697–1714 (2000)