Detecting Semantic Anomalies in Truck Weigh-In-Motion Traffic Data Using Data Mining

Orna Raz[1], Rebecca Buchheit[2], Mary Shaw[3], Philip Koopman[4], and Christos Faloutsos[5]

ABSTRACT

Monitoring data from event-based monitoring systems are becoming more and more prevalent in civil engineering. An example is truck weigh-in-motion (WIM) data. These data are used in the transportation domain for various analyses, such as analyzing the effects of commercial truck traffic on pavement materials and designs.

It is important that such analyses use good quality data or at least account appropriately for any deficiencies in the quality of data they are using. Low quality data may exist due to problems in the sensing hardware, in its calibration, or in the software processing the raw sensor data. The vast quantities of data collected make it infeasible for a human to examine all the data.

We propose a data mining approach for automatically detecting semantic anomalies—unexpected behavior—in monitoring data. Our method provides automated assistance to domain experts in setting up constraints for data behavior.

We show the effectiveness of our method by reporting its successful application to data from an actual WIM system: experimental data the Minnesota department of transportation collected by its Minnesota road research project (Mn/ROAD) facilities. The constraints the expert set up by applying our method were useful for automatic anomaly detection over the Mn/ROAD data: they detected anomalies the expert cared about—unlikely vehicles and erroneously classified vehicles—and the misclassification rate was reasonable for a human to handle (usually less than 3%). Moreover, the expert gained insights about the system behavior, such as realizing that a system-wide change had occurred. The constraints detected, for example, periods in which the WIM system reported roughly 20% of the vehicles classified as three axle single unit trucks to have one axle!

Keywords: Data analysis, Civil engineering, Traffic, Data mining, Machine learning

INTRODUCTION

[1]School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213 USA
[2]Civil Eng. Dept. Carnegie Mellon University, Pittsburgh PA 15213 USA
[3]School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213 USA
[4]Electric Eng. Dept., Carnegie Mellon University, Pittsburgh PA 15213 USA
[5]School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213 USA

Truck weigh-in-motion (WIM) data are an example of event-based monitoring data. Monitoring data are collected to measure the state of infrastructure elements. Such data are often used in infrastructure management systems to assist local, state and federal agencies in developing strategies to maintain their infrastructure. Monitoring data provide support for predicting deterioration, scheduling maintenance, and calculating life cycle costs. Within an infrastructure management system, monitoring data are usually used as historic precedent to predict future deterioration (Hudson et al. 1997).

Monitoring data are also used to analyze design decisions and support research activities. For example, the Minnesota road research project (Mn/ROAD) data we use in this paper are included in the Long Term Pavement Performance (LTPP) Project (LAW PCS 1999). The LTPP project uses the Mn/ROAD data and similar data from other states to support pavement performance analysis and to design better pavements.

WIM data, such as the Mn/ROAD data, are collected from an event-based monitoring system; every vehicle that passes over the WIM scale is an event that yields a recorded observation. This particular WIM scale is embedded in the infrastructure itself and is meant to monitor the infrastructure. Multiple software algorithms process the sensor data to estimate various attributes, such as the vehicle class, and filter out unreasonable values.

The data processing path between vehicles crossing the scale and the resulting recorded observations is fraught with opportunities for data quality degradation. For example, a sensor may be defective or wrongly calibrated, or software may assume an incorrect system state. These may result in unreasonable values, such as values that are physically impossible, or values that are improper for the vehicle class.

Because such data are used for analysis, it is important that the data be of good quality. Unfortunately, independent information about correctness rarely exists, making it challenging to verify the data quality. A first step toward achieving good quality data is the ability to detect anomalies.

Other work (Buchheit et al. 2003) concentrated on finding the best automated techniques for detecting and cleaning aggregated data, using domain knowledge. Buchheit also detected individual anomalies using the Minnesota Truck regulations as a model of data behavior. However, in that work, aggregate properties, such as the daily sum of vehicle weight, were used to find patterns of systematic error in the data. Our work is complementary because it concentrates solely on individual observations. In addition, our general purpose inference framework expands on Buchheit's work by helping the users to elicit their own, possibly more complete, model of the data's behavior.

We concentrate on *semantic anomalies*—unreasonable values. Once anomalies are detected the data may be cleaned or proper allowances may be made in the analysis to account for the data quality.

Detecting semantic anomalies requires a precise model of proper behavior: a semantic anomaly is behavior that is outside this model. However, though users (e.g., analysts) often have accurate expectations for the behavior of the data, they usually are unable to state these formally and precisely. For example, a user may expect trucks reported by a WIM scale to be physically plausible but may not be able to specify all the properties and values that define such plausibility.

We propose an automated method for assisting a user in creating a precise model of proper data behavior. Our predicate inference framework contains a template mechanism that interacts with a user to make the user's expectations for the data behavior precise. The framework then uses the precise expectations to detect semantic anomalies in the data. It uses data mining—applying statistical and machine learning techniques to help discover meaningful information in the data. Though these techniques characterize various aspects of the data, characterizing *relevant* behavior requires eliciting the user's expectations as well.

We apply our general purpose predicate inference framework to the Mn/ROAD data. An expert interacts with the template mechanism to make the expert's expectations precise. We then use the resulting model to detect semantic anomalies in the Mn/ROAD data.

We show that the template mechanism is effective; we measure effectiveness both by the insights the expert gains (the usefulness of the process) and the detection and misclassification rates (the usefulness of the resulting model). We were able to detect anomalies that surprised our expert, as they suggested system (hardware and software) behavior the expert was unaware of. Moreover, because our approach is automated, it detects anomalies quickly. In comparison, it had taken the data providers several months to notice the same problems independently. This is probably because the amount of data is very large, and it is hard to know what to look for.

WEIGH-IN-MOTION DATA

The data we use in our case-study are experimental data the Minnesota Department of Transportation collected in its Mn/ROAD research facilities between January 1998 and December 2000. The data have over three million observations for ten commercial vehicle types out of fourteen total vehicle types.

The Mn/ROAD research division operates a two-lane mainline test road equipped with a weigh-in-motion (WIM) scale. The test road is an active highway segment that runs parallel to a 3.5 mile section of I-94 westbound, near Otsego, Minnesota (Minnesota Department of Transportation

2002). The WIM scale uses four single load cell detectors in a sealed frame and four loop detectors embedded in the pavement to observe passing vehicles. The WIM scale measures the individual axle spacings and weights for each vehicle that passes over the scale. The gross weight of each vehicle is calculated based on this information. Length and speed are derived from the time that passes between noticing axles. In addition, the time when the vehicle crosses the scale, the lane in which the vehicle was traveling, and any error codes generated by the WIM scale are recorded. These measurements are sent to a nearby computer. Software algorithms then calculate a classification number and the equivalent standard axle loads (ESALs) (American Association of State Highway and Transportation Officials 1986) for each vehicle. The classification numbers are based on the Federal Highway Administration (FHWA) vehicle classification system, modified to include a class for invalid vehicles and a class for vehicles that do not fit into the FHWA schema (Federal Highway Administration 2001). ESALs are a dimensionless quantity that describes the usage of a pavement surface; an ESAL value of 1.0 is a standard truck. The computer saves the vehicle information into a text file; a software algorithm then imports this text file into a database. It assigns a unique identification number to each vehicle, purportedly filters out unreasonable values, and ignores personal vehicles (the WIM scale records all vehicles that pass on it).

Roughly one million vehicles are added to the data set per year. The number of observations the system collects varies by vehicle type. For example, it collects two thousand observations during roughly three weeks for each of vehicle types 4 and 6 and during roughly two days for type 9.

We treat the Mn/ROAD data as a time-stamped sequence of observations. Each observation has attribute values for a single commercial vehicle: date and time (accurate to the millisecond), vehicle type (one of ten classes), lane (one of two classes), speed (mph—miles per hour), error code (one of twenty five classes), length (feet), ESAL, number of axles, and weight (kips—kilo-pounds).

Several states in the USA are collecting truck WIM data and analyzing them to better understand transportation issues. Though there are different WIM scales, the basic data are very similar.

WIM data have been used extensively for analysis of transportation design issues. This includes Mn/ROAD research projects, such as pavement performance, preventive maintenance, and low volume road design (Minnesota Road Research Section 2003), as well as research projects and analyses at other states (Beshears et al. 1998; Najafi and Blackadar 1998; Lee and Souny-Slitine 1998; Clayton et al. 2002). There is also research examining additional applications of WIM data, for example, real time enforcement of truck weight restrictions (Andrle et al. 2002).

However, little work addresses data quality issues. Assessing data quality is complementary to analyzing the data. The Vacuum system (Buchheit 2002; Buchheit et al. 2002; Buchheit et al. 2003) was applied to the Mn/ROAD data for assessing the data quality and cleaning the data. However, the analysis concentrated on aggregated data: the daily sum of attribute values, such as daily sum of ESAL. Our analysis is complementary. It concentrates on individual observations and our framework can support various existing techniques. The kind of anomalies we detect are vehicles that do not seem to belong to their assigned class and vehicles that have attribute values that are improbable (e.g., too low or too high). Such anomalies may explain anomalies in the aggregated data.

POSSIBLE DATA PROBLEMS

The Mn/ROAD data are produced by an event-based monitoring system. The sensor data are further processed by multiple software algorithms, written by different contractors. Many things can go wrong in this process. For example, there may be problems in the physical calibration of the WIM scale, inaccurate sensing, improper processing done by software, or undesirable interactions among multiple software algorithms processing the data.

Such problems may cause an observation to have attribute values such that a real vehicle is not in its correct class (it has realistic attribute values but these are very different from values of the same attributes in other vehicles of its assigned class) or a vehicle is physically highly improbable. Such unreasonable values are semantic anomalies that we want to detect.

Anomaly detection is a first step toward improving the sensing system—both hardware and software. It enables further analysis such as indicating whether an anomaly is a system failure, locating the failure source, and taking remedial actions.

Buchheit (Buchheit 2002) distinguishes between two categories of error types in event-based monitoring data: errors in aggregated data and errors in individual observations. We concentrate on problems in individual observations. Buchheit classifies individual observation error types as: missing observation, duplication of same observation, garbling errors—occur when a real-world value is incorrectly recorded or is missing from a data set, and combination errors—occur when two events are recorded as a single event or when a single event is decomposed into two events. Detecting unreasonable values due to garbling or combination requires knowledge about the semantics of the data and is, therefore, harder to automate than detecting missing or duplicated values. We, therefore, concentrate on detecting unreasonable values.

PROPOSED SOLUTION

We are interested in detecting semantic anomalies such as the unreasonable values in WIM data that we have defined above. Detecting such values requires knowledge about the semantics of the data. In addition, different analyses may rely on different aspects of the data behavior. Users, even experts, use the data for a particular purpose and have expectations about the behavior of the data that are relevant to the specific usage/analysis.

If an expert could look at all the observations and could concentrate while doing so, the expert would most likely detect the unreasonable values. However, expert time is expensive, humans find it hard to concentrate on repetitive tasks, and the quantities of data to inspect are often large.

Alternatively, the expert could define a model of proper data behavior for the specific usage/analysis the expert is involved with. An anomaly would then be a value that is outside this model. This model would be based on the expert's expectations for the data behavior. However, users' expectations are informal and imprecise, though they are reasonably accurate.

We propose a method that provides automated assistance to users in making their expectations for the data behavior precise. This method relies only on: (1) data: the observable behavior of the system over some period of time, which we term a data feed and (2) minimal user feedback in the form of classifying the output of the data mining techniques into three categories. We use the resulting model to detect semantic anomalies in the data. We do this through our predicate inference framework.

Data mining and machine learning have been used for civil engineering applications (Melhem and Cheng 2003; Reich 1997; Arciszewski and Rossman 1992). However, these applications concentrate on analysis and prediction. Our approach is complementary: we use data mining for detecting anomalies in the data prior to using the data for such analysis and prediction.

Soibelman et al. (Soibelman and Kim 2002) propose a process for data preparation for knowledge discovery in data bases, in the construction domain. Our method provides automation for such a process, using the techniques in our inference tool kit. Our method could be used in any domain that would benefit from using the resulting constraints on data behavior.

Approaches of modeling expert knowledge are complementary to our approach of suggesting predicates using unsupervised learning. Caldas et al. (Caldas and Soibelman 2003; Caldas et al. 2002) propose mechanisms for project collaboration, coordination,and information exchange. They deal mostly with text documents. Maher et al. explore case-based approaches to structural design of buildings in a large body of work, an example of which is (Maher and Balachandran 1994). Simoff et al. explore virtual environments for learning about design, for example in (Simoff and Maher 1997).

We use and adapt existing unsupervised learning techniques from the areas of statistics and machine learning. Co-training (Blum and Mitchell 1998) investigates ways to reduce the human effort that labeling data for supervised learning requires. Active learning (Cohn et al. 1996) investigates statistical ways to select the most promising training data for a technique. We ask the user to classify the output of a technique, rather than its input.

Our approach of inferring the characteristics of a data feed from its behavior is similar to work in the areas of program analysis and testing. Daikon (Ernst et al. 2000) dynamically discovers likely program invariants from program executions. We incorporate Daikon in our predicate inference tool kit. "Bugs as deviant behavior" (Engler et al. 2001) infers beliefs from source code so it is inappropriate for data. "Specifications mining" (Ammons et al. 2002) uses a machine learning approach for discovering specifications of interface protocols. However, it uses techniques specific to code. "Observation-based testing" (Dickinson et al. 2001) uses clustering and visualization techniques to identify unusual program executions. We have similar techniques in our tool kit.

PREDICATE INFERENCE FRAMEWORK

We present our framework concentrating on the domain of monitoring systems and on a particular data set: the Mn/ROAD data. Our framework is domain independent; a detailed discussion of our framework and its general applicability appears in (Raz et al. 2003).

Figure 1 gives a synopsis of our predicate inference framework. This framework has three major stages: (1) setting up a model of proper behavior by eliciting precise user expectations; this stage relies on a novel template mechanism and is the focus of this paper, (2) using the precise expectations as a proxy for missing specifications to detect semantic anomalies in the data; previous work (Raz et al. 2002) discussed this stage, and (3) updating the precise expectations to account for evolving system behavior or user expectations; we defer this stage to future work.

The mechanisms that support the above stages are: (1) the *technique tool kit*—a collection of existing statistical and machine learning techniques that we support and adapt, (2) the *template mechanism*—a mechanism that guides the human attention required in making expectations precise using templates that document the predicates a particular technique can output, and (3) the *anomaly detector*—a mechanism that uses the precise expectations as a model of proper behavior and reports as anomalies observations that falsify the expectations. Details about these mechanisms follow.

THE TEMPLATE MECHANISM

We characterize a predicate inference technique by the types of predicates it can produce. Templates capture the form of these predicates. For example, an inference technique may find a probable range for the values of a given attribute, e.g., the length attribute. The corresponding template would be $\# \leq \text{length} \leq \#$, where # is a numeric value. Our method concentrates on numeric valued attributes. However, the template variable # can be a category value (e.g., lane = 1). Figure 2 gives a synopsys of how the template mechanism works.

An inferred predicate is a "complete instantiation" of a template. The template mechanism uses this complete instantiation for templates of "accept" predicates. Classifying a predicate as either "reject" or "update" may make the template instantiation partial by rendering the instantiation of all the numeric values in one or more dimensions void. See the description of Rectmix in the next Section for an example.

The template mechanism treats the predicate inference techniques as black boxes and uses the instantiated templates to filter the predicates a technique infers. It constructs and updates the model of proper behavior from instantiated templates of "accept" and "update" predicates. It will never present the user or the anomaly detector with predicates that match templates of previously rejected predicates. The template mechanism eliminates techniques that are not relevant for this user and data: it will not employ an inference technique if the user rejects all the predicates that are associated with this technique.

Premises of our template mechanism include (1) it is easier for a user to understand expectations about data behavior when presented with examples. It is especially useful to examine examples of anomalous behavior, with the predicates that flagged them as anomalous, and (2) it is easier for a user to choose from a list of inferred predicates than to create this list, so having a machine synthesize the list is helpful.

THE TECHNIQUE TOOL KIT

The tool kit consists of multiple predicate inference techniques. These are existing machine learning and statistical techniques that we support and adapt. Users may add techniques to the tool kit. The truck WIM case study that follows uses two of the tool-kit techniques: Rectmix and Percentile. We selected these techniques because they work best on the Mn/ROAD data: their predicates describe data behavior that the expert cares about. For this data feed, the other techniques either describe irrelevant behavior or produce predicates that are less precise or redundant with respect to the Rectmix and Percentile predicates. A description of the techniques we used and their selection follows.

Each technique is likely to be useful only for data with certain characteristics. This provides an initial technique filtering criterion. We use measurement scales (Fenton and Pfleeger 1997) for this purpose. Measurement scales enable matching data with techniques that perform manipulations appropriate to the data scale. For example, the lane attribute of the Mn/ROAD data feed has a nominal scale—there is no notion of ordering or magnitude associated with the numbers used to specify lanes. Therefore, it is meaningless to apply mathematical predicates to this attribute. If a technique performs transformations that are inappropriate to all the attributes of a data feed the technique should not be applied to that data feed.

In addition, different users may find different techniques useful or the output of one technique may be largely redundant with another for the data of interest. Different techniques make different assumptions and often use different vocabularies. Therefore, it may be useful to apply multiple techniques to the same data. However, a large number of techniques is likely to burden the human. The template mechanism supports filtering techniques partially on discriminating ability (effectiveness in anomaly detection) and partially on output comprehensibility. Filtering by these criteria enables the user to select the techniques that promise the best use of human attention.

Our technique tool kit currently consists of five techniques. We selected these techniques because they expose different aspects of the data and because their output is easy for a human to understand. We breifly present these techniques, and summarize their output for the Mn/ROAD data.

The Rectmix technique Rectmix (Pelleg and Moore 2001) is a clustering algorithm that supports soft membership (a point can probabilistically belong to multiple clusters). The clusters it finds are hyper-rectangles in N-space. Rectmix provides a measure of uncertainty called sigma (an estimate of the standard deviation) for each dimension. Anomalies are points that are not within a rectangle. Though clusters rarely have a hyper-rectangle shape in reality, Rectmix has the significant advantage of producing output that is easy to understand: a hyper-rectangle is simply a conjunction of ranges, one for each attribute (see Table 1). Rectmix has two parameters: the number of rectangles and the number of sigmas of uncertainty to allow.

Rectmix always outputs hyper-rectangles, so it has a single template: $\# \leq A_1 \leq \# \wedge ... \wedge \# \leq A_n \leq \#$, where n is the number of attributes (dimensions). Table 1 gives an example of user classification for predicates that Rectmix outputs for a subset of the Mn/ROAD data. The corresponding templates have numeric values in one dimension—the axle attribute—because the user chose to void the other attribute values. For example, the template for the first predicate is $\#\leq$length$\leq\# \wedge \#\leq$ESAL$\leq\# \wedge 3\leq$axles$\leq 3 \wedge \#\leq$weight$\leq\#$.

The Percentile technique The x percentile of a distribution is a value in the distribution such that $x\%$ of the values in the distribution are equal or below it. Percentile calculates the range between the x and 100-x percentiles and allows $y\%$ uncertainty. Percentile only assumes values are somewhat centered and tolerates extreme values.

Percentile predicates are a probable range for the values of each attribute. Percentile has a single template: #≤A≤#. Table 2 gives an example of user classification and resulting instantiated templates for predicates that Percentile infers over a subset of the Mn/ROAD data. Percentile (x=25, y=25%) works well for speed, length, axles, and weight, but not for ESAL (ESAL seems to be exponentially distributed).

The K-means technique (Duda et al. 2000) is a clustering algorithm with hard membership: it partitions points into distinct clusters. Anomalies are points that are furthest away from the center of their cluster, according to the Euclidean distance metric K-means uses.

K-means templates are a set of k cluster centers: $C_1, ..., C_k$, where $C_i = (A_1 = \# \wedge ... \wedge A_n = \#)$ and n is the number of attributes. For the Mn/ROAD data, when requesting k=2–4 clusters, the centers make sense. However, the furthest observations in each cluster are not necessarily anomalies. This means that the measure of multi-dimensional Euclidean distance is not meaningful for this data. In addition, soft membership is more appropriate for the Mn/ROAD data than hard membership. Therefore, K-means is not a good choice for the Mn/ROAD data.

The Association Rules technique (Agrawal et al. 1993) finds probabilistic rules in an 'if then' form. The rules reflect correlations among attributes but cannot know about cause and effect. They can only give examples with specific values. The advantage is that association rules may detect correlations that may be due to complicated relations. The disadvantage is that they cannot suggest a general relation to explain the correlation. Association rules work on categorical data so numeric data is first divided into bins.

Association rules templates are of the form 'if $E_1 \wedge ... \wedge E_m$ then E_x', where $E_i \in \{\# \leq A_i \leq \# \mid A_i \geq \# \mid A_i \leq \#\}$ and $m < n$, the number of attributes. For the Mn/ROAD data, association rules work rather well. An example of the rules they produce is 'if length<34.8∧ESAL<0.11∧2≤axles≤3 then weight<19.2'. However, not all the rules contain all of these four attributes, and the cause and effect relation is often absent (as is the case above: ESAL is calculated based on the other attribute values). In general, Rectmix performs better over the Mn/ROAD data.

The Daikon technique (Ernst et al. 2000) was developed for the program analysis domain. It dynamically discovers likely program invariants over program execution traces by checking whether pre-defined relations hold. We map Daikon's program points and variables to our observations and attributes, respectively. Daikon assumes the data is clean, but our data contains anomalies. Therefore, we use voting: we run Daikon on multiple subsets of the data and use the invariants that appear in multiple subsets. Daikon is very effective when strong correlations that can be described by its pre-defined relations exist.

We have a template for each of Daikon's pre-defined relations. Because the Mn/ROAD data has mostly statistical correlations the only useful predicates Daikon outputs are 'axles $\in \{\#\}$'. However, Rectmix and Percentile produce similar predicates, so we prefer these techniques for Mn/ROAD data.

Pre-processing Data pre-processing may be necessary before the template mechanism interacts with the user. Pre-processing often helps to overcome technique weaknesses. This includes: (1) setting parameters of inference techniques, (2) performing data transformations, (3) selecting attributes, and (4) clustering.

To determine technique parameters, the template mechanism runs each technique with several values for each parameter and lets the user select the combination that best reflects the user's expectations. Alternatively, the user may choose to use the default parameter values.

Data transformations are usually straight forward and automated. For example, normalizing each numeric valued attribute to have mean one and standard deviation zero is a common transformation that is necessary for techniques that assume similarly scaled attributes (e.g., Rectmix) and data with differently scaled attributes (e.g., the Mn/ROAD data).

Attribute selection is useful for techniques that produce multi-dimensional templates both because these techniques tend to work better with less dimensions (attributes) and because as the number of dimensions increases it becomes harder for a human to understand and visualize the results. If different classes of data (clusters) exist, they are likely to behave differently. Therefore, the template mechanism runs the techniques on data in each class to enable the user to create a separate model for each class.

Any attribute selection technique could be used with our method. We found Principal Component Analysis (PCA) useful for attribute selection and clustering of the Mn/ROAD data because these data have linear correlations.

PCA (Jolliffe 1986) is a way to reduce the dimensionality of the data thus enabling visualization. PCA generates a new set of variables, called principal components, where each principal component is a linear combination of the original variables. If linear correlations exist, PCA can serve for attribute selection because it indicates which of the attributes are most strongly linearly correlated.

Looking at the data helps in finding different classes of the data. To visualize the data, we plot the observations along the first two principal components. To check for clusters, we color the observations according to each of the attributes (a color for each value of a categorical valued attribute, a color for each bin of a numeric valued attribute).

We observe by looking at the PCA plots for the Mn/ROAD data that either vehicle type or axles can be used to cluster the data and the resulting clusters overlap. We choose vehicle type as the data class. The first principal component indicates a linear correlation among length, ESAL, axles, and weight. Therefore, we select these attributes as input for techniques that produce multi-dimensional templates (e.g., Rectmix). The other components do not indicate interesting correlations. The second axis indicates mostly the speed of a vehicle. Therefore, we add the speed attribute for analysis by techniques that produce one-dimensional templates (e.g., Percentile).

CASE STUDY HYPOTHESIS

We test our template mechanism by having an expert interact with it to set up a model of proper behavior for the Mn/ROAD data. Our case study tests the following hypothesis: the template mechanism helps users make their expectations precise. Further, the template mechanism, along with the technique tool kit and the anomaly detector, effectively direct the human attention necessary in setting up a model of proper behavior and in analyzing the resulting anomalies.

If our hypothesis is correct then

1. The resulting model of proper behavior will be useful in detecting semantic anomalies in the Mn/ROAD data.
2. The user, an expert in this case, will gain insights about the WIM system through interaction with the template mechanism and through analysis of anomalies.

CASE STUDY METHODOLOGY

As described in the pre-processing Section, we begin by looking for clusters and selecting attributes. As a result, the template mechanism interacts with the user for each class (vehicle type) separately and inputs the selected attributes to techniques in the tool kit.

For the purpose of validating our template mechanism, we select three out of the ten vehicle types that the data contain. We select the most common vehicle type (type 9, about two million observations) and two additional types (type 4 and type 6, about one hundred thousand observations each). The existing documentation defines these types as follows: type 9 vehicles are five-axle single trailer trucks, type 6 vehicles are three-axle single unit trucks, and type 4 vehicles are buses. On the basis of preliminary analysis, the vehicle types seem similar enough that we can use the same techniques over them. We first let the user create a model for two of the types (4 and 6) then we let the user create a model for the third type (9) using the techniques and parameters the user chose for the first two types. This works well, supporting our preliminary analysis regarding the similarity of the vehicle types with respect to the tool-kit techniques. The same techniques and parameters should work well for the other vehicle types as well, but doing so is beyond the scope of our work.

A domain expert sets up the model of proper behavior. We give this model to the anomaly detector. The anomaly detector runs over subsets of the data. We sort the data by time and divide it into subsets of two thousand consecutive observations each, to simulate the on-line data nature.

To analyze the model, we determine the resulting detection rate and the misclassification rate. The detection rate calculates how many attributes the model flags as anomalies out of the total number of attributes. It is an objective measure because the results of using the model for anomaly detection are binary: normal or anomalous. However, it is important to also analyze the usefulness of the model. The misclassification rate quantifies the usefulness of the model. Because we do not have independent information on correctness this is necessarily subjective. We concentrate on whether the model is effective in detecting anomalies the user cares about, not on whether it detects all the anomalies.

CASE STUDY DETECTION RATE

We detect anomalies over the Mn/ROAD data using the model the expert has set up. Tables 3, 4, and 5 list the Rectmix model the expert has set up (predicates outputed by Rectmix) for vehicle types 4, 6, and 9, respectively. Table 6 lists the Percentile models the expert has set up (predicates outputed by Percentile) for vehicle types 4, 6, and 9. These models consist of "update" and "accept" predicates from the final setup stage. For example, for vehicle type 6, Table 4 consists of the "update" predicates from Table 1—the final setup classification for Rectmix predicates. The middle column of Table 6 consists of the "update" predicates from Table 2—the final setup classification for Percentile predicates.

We use the model for anomaly detection and compute the resulting detection rate. We present plots for one vehicle type—type 6. The plots for type 4 and type 9 vehicles are similar except as indicated in the analysis that follows.

Figure 3 depicts a count of anomalous attributes flagged by the Rectmix predicates the expert chose for vehicle type 6. Similarly, Figure 4 depicts a count of anomalous attributes flagged by the Percentile predicates the expert chose for vehicle type 6.

Data subsets are time ordered; each has two thousand observations. The y-axis in a plot gives the total number of anomalies in one of the subsets, according to the criterion the plot specifies, e.g., length anomalies. Notice that the y-axis scale differs among plots. The x-axis is the sequential subset index. Figure 3's left-most plot summarizes the total number of anomalous attributes, out of eight thousand attribute observations (four attributes times two thousand observations for each). The other plots show the break-down of this total by attribute, out of two thousand observations.

The first column in Figure 4 summarizes the number of anomalies for each attribute. The plots in the second and third columns summarize the anomalies that are due to attribute values that are smaller or larger, respectively, than the range bounds. All are out of two thousand observations.

Table 7 summarizes the average detection rate over the subsets of each vehicle type. It gives the detection rate over all attributes and a break-down by attribute.

Looking at the detection rate over a number of subsets (Figures 3 and 4) is insightful. Patterns and changes become visually obvious.

The detection rate (anomalies) for type 9 vehicles is much lower than for the other types. The data of type 9 vehicles seem much cleaner than for the other types. The number of axles is absolutely clean (no anomalies). The weight is usually normal but in some of the subsets there is a very large number of over-weight vehicles (hundreds out of two thousand). This may be due to weight sensor problems in the scale or calibration problems on specific dates. Type 9 is by far the most common, so probably the scale and software are calibrated to best recognize this type.

Figure 4 draws our attention to a correlation between low speed (speed < 40 mph) and over-length (length > 39 feet)—the plots have a similar shape. This helps us to better understand how the length estimation works. The length is estimated from the time that passes between axles, assuming high-way speed. Therefore, if the speed is very low, the length will be over estimated.

Looking at the anomalies for axles in Figure 4, it appears there was a change in the WIM system starting subset number 54 (November 1999). The number of axles is very noisy in earlier observations and very clean in later observations. The same behavior occurs in type 4 vehicles. This may be due to a software update in the classification or filtering algorithms or a re-calibration of the WIM scale. Our expert was surprised to see this behavior. The expert was also surprised to learn that a large number of vehicles with one axle exist in the data; all commercial vehicles should have at least two axles, and the filtering algorithm should have detected such an anomaly.

Both Figure 3 and Figure 4 show that during the period of time in which the axle attribute is clean, the length is also cleaner (fewer anomalies). The same behavior occurs in type 4 vehicles. This may be due to the same change that resulted in a cleaner number of axles. Many of the type 6 length anomalies are due to the maximal length the WIM system can record: 99.9 feet. Our expert was unaware of the large number of exceptionally long and slow type 6 vehicles during the early data collection period. This may be due to problems in either the scale calibration or the software.

The total detection rate (Table 7) cannot be compared between Rectmix and Percentile because the attributes are not all the same and because these techniques describe different behavior: Rectmix finds correlations among common attribute values whereas Percentile simply finds common values for a single attribute. However, it is interesting to compare the detection rates for the identical attributes (length, axle, weight). Understanding differences helps in model understanding.

The axles anomaly detection rate is very different between Rectmix and Percentile because the predicates the expert chose differ. For example, Percentile predicates allow 3 axels for type 6 vehicles, but the Rectmix predicates allow 2–4 axles.

Small differences in the ranges for length and weight result in large differences in the detection rate, indicating that the values for these attributes are closely concentrated. The exact cut-off point between normal and anomalous is, therefore, not clear from the data. For example, due to small range differences, the Rectmix length-anomaly detection rate is about five times the Percentile detection rate, except for vehicle type 6 that has an exceptionally high length-anomaly rate. Type 4 Rectmix length anomalies are numerous compared to the other attributes, indicating this bound may be too tight. Due to small range differences, the Percentile weight-anomaly detection rate is about twice the Rectmix detection rate. Rectmix notices a correlation of weight and ESAL in light vs. heavy trucks. The type 6 upper weight bound is much higher for Rectmix, possibly because it also considers trucks with more axles.

CASE STUDY MISCLASSIFICATION RATE

The overall misclassification is $\frac{FP+FN}{Nor+Ab}$ (Runeson et al. 2001) (lower is better), where True Positives (TP) are correctly detected anomalous data, False Positives (FP) are normal data falsely detected as anomalous, False Negatives (FN) are undetected anomalous data, Normal (Nor=TN+FP) are data that are actually anomaly-free, and Abnormal (Ab=TP+FN) are data with anomalies.

Determining these measures is subjective even though documentation for the WIM system exists. This is because, on the one hand, the documentation is sometimes incomplete and imprecise, and on the other hand, it sometimes describes behavior that neither Rectmix nor Percentile can express.

To determine Ab, FP, and TP, our expert sets constraints based on analyzing anomalies flagged by the anomaly detector and differences between the inferred and documented models. Table 8 summarizes the resulting misclassification rate, averaged over data subsets of each vehicle type. The rates are reasonable for a human to handle. The slightly higher Rectmix rate for type 4 is due to the restrictive lower bound on length. Type 9 is the cleanest, so the techniques do best on it.

INFERRED MODEL VS. DOCUMENTED MODEL

We use the WIM system documentation of vehicle types (Chalkline et al. 2001) and attribute bounds (Mn Regulations 2000) as another indicator of what the system might do, and compare it to what the expert finds interesting.

The documentation concentrates on upper bounds. E.g., type 9 length $\leq$ 75 feet, type 4 or 6 length $\leq$ 40 feet. The techniques we use infer predicates about lower bounds as well (e.g., Tables 3–6). The expert found the lower bounds useful. For example, low speed correlates with over-length.

The classification is very noisy compared to the vehicle type documentation. For example, the documentation defines the number of axles per type, yet, except for type 9, the actual number of axles often differs. This led our expert to think about the way the system is calibrated and its effect on vehicle classification. The system seems to be physically tuned for the common type of trucks (type 9). Possible causes for anomalies in other types include: (1) inaccurate sensing, (2) unintended interaction effects among the algorithms (e.g., the filtering algorithms may not properly clean the output of the classification algorithm), and (3) boundary problems in the classification.

The class documentation often seems imprecise. Our expert chose predicates that are different from the documentation when they described vehicles the expert thought belonged in the same class. The documentation defines type 4 as traditional buses with at least two axles. The expert allowed only vehicles with 2–4 axles. The documentation defines type 6 vehicles as vehicles with a single frame having three axles. The expert allowed vehicles with 2–4 axles.

This comparison illuminates subtle expectation differences. The expert emphasizes equally all vehicle types and also data precision. The providers seem to emphasize most vehicle type 9 and avoiding over-estimation. The models reflect these different emphases.

SUMMARY OF EXPERT INSIGHTS

The major insights our expert gained from the analysis detailed above are as follows:

- The data behavior strongly suggests a system wide change in the Mn/ROAD WIM system starting November 1999.
- The system (both hardware and software) seems to be calibrated for the most common type of trucks. This, in turn, seems to adversely affect the accuracy of vehicle identification and classification of other types.
- The interaction of the various algorithms seems to occasionally have undesirable effects.

CONFIRMATION FROM PROVIDERS

The data providers confirmed the expert insights and cause analysis. They were unaware of the behavior that surprised our expert until recently, when they validated analyses that used these data. It turns out that the WIM scale has two different modes for weighing an axle. The various software algorithms made inconsistent assumptions about the weigh mode. As a result, they occasionally assigned values to the wrong attribute. The next algorithms in the chain did not recognize the problem and made calculations based on the incorrect data. Type 9 vehicles are cleaner because one of the many software providers recognized a problem and made an undocumented correction for type 9. In addition, the system is physically calibrated for this type.

This provides additional confirmation about the usefulness of our method. Moreover, it demonstrates the benefits of having automated anomaly detection. To set up the model, the expert invested less than ten hours. The anomaly detection was fully automated and quick (minutes). In comparison, it had taken the data providers several months to notice the same problems. Analyzing the anomalies requires expert time and this time depends on the number of anomalies and their nature. However, our method directs the expert's attention to problems, so expert time is invested efficiently.

CONCLUSIONS AND FUTURE WORK

We successfully applied our predicate inference framework to detect semantic anomalies in the Mn/ROAD data. Our template mechanism provides automated assistance to experts in setting up constraints for the behavior of monitoring data—it helps users to make their expectations for data behavior precise. The result is an analyzable model of proper behavior.

Our case study results support our hypothesis: (1) The model was useful for automatic anomaly detection over the Mn/ROAD data. It enabled detecting actual anomalies that the expert cared about: classification problems and unlikely vehicles. In addition, the misclassification rate was reasonable for a human to handle (usually less than 3%). (2) The expert gained insights about the WIM system. The data providers confirmed the expert insights.

Moreover, the case study results corroborate the benefits of interacting with the template mechanism to make expectations precise and of analyzing the resulting anomalies. Our method: (1) detected hardware and software problems from observed data only. It detected, for example, problems that were caused by mis-calibration, software modifications, or state changes, (2) promptly detected these problems. It had taken the data providers months to discover these independently. and (3) increased the understanding of existing documentation. For example, the exact cut-off point between normal and anomalous was not clear from the data though it was clear (for upper bounds) from the documentation, suggesting the documentation bounds may be too strict.

Many challenges remain in this area. We plan to extend our method to support updating predicated and time-correlated data, thus enhancing its applicability and usability. We believe our method is appropriate for any monitoring data. However, for every different data feed, a user would need to interact with our method to (1) set up the tool-kit techniques and (2) classify templates. This may require adding techniques to the tool kit; our method will provide a procedure for doing so. Once detection is in place, cleaning and mitigation/repair would be a natural next step. Automated support for analyzing the cause of anomalies would be a valuable aid for this purpose because it would greatly enhance the ability to automatically recover from or eliminate the detected anomalies.

ACKNOWLEDGMENTS

We thank the Minnesota Department of Transportation for their Mn/ROAD WIM data, Dan Pelleg for his Rectmix code and comments, the Auton Lab (Moore) for making their dataset processing and analysis software (SPRAT) available to us, the anonymous reviewers for their comments.

The authors wish to acknowledge support from the National Science Foundation under Grant ITR-0086003, by the Sloan Software Industry Center at Carnegie Mellon University, by the NASA High Dependability Computing Program under cooperative agreement NCC-2-1298, and by the General Motors Collaborative Research Laboratory at Carnegie Mellon. This material is based in part upon work supported by the National Science Foundation under Grant Number 9987871.

REFERENCES

Agrawal, R., Imielinski, T., and Swami, A. (1993). "Mining association rules between sets of items in large databases." *International Conference on Management of Data (SIGMOD 93).*

American Association of State Highway and Transportation Officials (1986). "AASHTO guide for design of pavement structures.

Ammons, G., Bodik, R., and Larus, J. (2002). "Mining specifications." *29th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages.*

Andrle, S., McCall, B., and Kroeger, D. (2002). "Application of weigh-in-motion (WIM) technologies in overweight vehicle enforcement." *3rd International Conference on Weigh-in-Motion (ICWIM3).*

T. Arciszewski and L. A. Rossman, eds. (1992). *Knowledge acquisition in civil engineering.* American Society of Civil Engineers (ASCE), New York.

Beshears, D. L., Muhs, J. D., Scudiere, M. B., Marlino, L., Taylor, B. W., Pratt, A., and Koenderink, R. J. (1998). "Advanced weigh-in-motion system for weighing vehicles at high speed." *Report No. C/ORNL-95-0364*, Lockheed Martin Energy Research Corporation.

Blum, A. and Mitchell, T. (1998). "Combining labeled and unlabeled data with co-training." *COLT: Workshop on Computational Learning Theory.*

Buchheit, R. (2002). “Vacuum: Automated procedures for assessing and cleansing civil infrastructure data,” PhD thesis, Carnegie Mellon University, Civil Engineering Dept.

Buchheit, R., Garrett Jr., J., and McNeil, S. (2003). “An automated procedure to assess civil infrastructure data quality: Method and validation. Submitted.

Buchheit, R., Garrett Jr., J., McNeil, S., and Chalkline, M. (2002). “Automated procedures for improving the accuracy of sensor-based monitoring data.” *7th Applications of Advanced Technologies in Transportation Conference (AATT-VII).*

Caldas, C. H. and Soibelman, L. (2003). “Automating hierarchical document classification for construction management information systems.” *Automation in Construction*, 12(4), 395–406.

Caldas, C. H., Soibelman, L., and J., H. (2002). “Automated classification of construction project documents.” *ASCE Journal of Computing in Civil Engineering*, 16(4), 234–243.

Chalkline, M. H., Dahlin, C., and Guan, R. (2001). *Documentation of Traffic Data Collection and Data Warehouse at Mn/ROAD*. Office of Materials and Road Research.

Clayton, A., Montufar, J., and Middleton, D. (2002). “Using weigh-in-motion data in a modern truck traffic information system.” *3rd International Conference on Weigh-in-Motion (ICWIM3).*

Cohn, D., Ghahramani, Z., and Jordan, M. (1996). “Active learning with statistical models.” *Journal of Artificial Intelligence Research*, 4, 129–145.

Dickinson, W., Leon, D., and Podgurski, A. (2001). “Finding failures by cluster analysis of execution profiles.” *23rd International conference on Software Engineering.*

Duda, R., Hart, P., and Stork, D. (2000). *Pattern Classification,*. John Wiley and Sons, 2nd edition.

Engler, D., Chen, D. Y., Hallem, S., Chou, A., and Chelf, B. (2001). “Bugs as deviant behavior: A general approach to inferring errors in systems code.” *18th ACM Symposium on Operating Systems Principles.*

Ernst, M., Cockrell, J., Griswold, W., and Notkin, D. (2000). “Dynamically discovering likely program invariants to support program evolution.” *IEEE Transactions on Software Engineering.*

Federal Highway Administration (2001). “Traffic monitoring guide. Technical report.

Fenton, N. E. and Pfleeger, S. L. (1997). *Software Metrics*, chapter 2. PWS Publishing Company, 2nd edition.

Hudson, W. R., Haas, R., and Uddin, W. (1997). *Infrastructure Management*. McGraw-Hill, New York.

Jolliffe, I. T. (1986). *Principal component analysis*. Springer-Verlag, New-York.

LAW PCS (1999). “Introduction to LTPP data. Prepared by LAW PCS, a LAWGIBB Group Member, Prepared for Office of Infrastructure Research and Development, Federal Highway Administration. URL: http://www.tfhrc.gov/pavement/ltpp/pdf/resguide.pdf. Accessed May 2002.

Lee, C. and Souny-Slitine, N. (1998). "Final research findings on traffic-load forecasting using weigh-in-motion data." *Report No. 987-7*, Texas University, Austin.

Maher, M. L. and Balachandran, B. (1994). "A multimedia approach to case-based structural design." *ASCE Journal of Computing in Civil Engineering*, 8(3), 359–376.

Melhem, H. G. and Cheng, Y. (2003). "Prediction of remaining service life of bridge decks using machine learning." *ASCE Journal of Computing in Civil Engineering*, 17(1).

Minnesota Department of Transportation (2000). *Minnesota Trucking Regulations*. Technical Report.

Minnesota Department of Transportation (2002). "Mn/DOT office of materials and road research. URL: http://mnroad.dot.state.mn.us. Accessed May 2002.

Minnesota Road Research Section (2003). "Mn/ROAD research projects. URL: http://www.mrr.dot.state.mn.us/research/MnROAD_Project/projects90.asp. Accessed Feb 2003.

Moore, A. "Auton Lab. URL: http://www.autonlab.org. Accessed April 2003.

Najafi, F. T. and Blackadar, B. (1998). "Analysis of weigh-in-motion truck traffic data." *40th Annual Meeting of the Transportation Research Forum*.

Pelleg, D. and Moore, A. (2001). "Mixtures of rectangles: Interpretable soft clustering." *8th International Conference on Machine Learning (ICML)*.

Raz, O., Buchheit, R., Shaw, M., Koopman, P., and Faloutsos, C. (2003). "Eliciting user expectations for data behavior via invariant templates." *Report no.*, CMU-CS-03-105.

Raz, O., Koopman, P., and Shaw, M. (2002). "Semantic anomaly detection in online data sources." *24th International Conference on Software Engineering*.

Reich, Y. (1997). "Machine learning techniques for civil engineering problems." *Microcomputers in Civil Engineering*, 12(4).

Runeson, P., Ohlsson, M., and Wohlin, C. (2001). "A classification scheme for studies on fault-prone components." *Product focused software process improvement*.

Simoff, S. J. and Maher, M. L. (1997). "Design education via web-based virtual environments." *Fourth Congress of Computing in Civil Engineering*, T. Adams, ed., Computing in Civil Engineering. 418–425.

Soibelman, L. and Kim, H. (2002). "Data preparation process for construction knowledge generation through knowledge discovery in databases." *ASCE Journal of Computing in Civil Engineering*, 16(1), 3–48.

List of Tables

Class	Length ∧	ESAL∧	Axles ∧	Weight
Update	20–42	0–.43	3–3	12–29
Update	23–44	0–1.2	2–3	26–47
Reject	13–100	0–.45	2–7	7–40
Update	23–29	0–6.7	2–4	27–71

TABLE 1. Example of Rectmix predicates classification

Class	Predicate	Template
Update	40≤speed≤88	#≤speed≤#
Update	17≤length≤39	#≤length≤#
Reject	.06≤ESAL≤.9	#≤ESAL≤#
Update	3≤axles≤3	#≤axles≤#
Update	12≤weight≤49	#≤weight≤#

TABLE 2. Example of percentile predicates with user classification and instantiated templates

Rectangle	Length ∧	ESAL∧	Axles ∧	Weight
1	32–43	0–.42	2–2	11–22
2	32–45	.1–1.2	2–2	18–29
3	31–49	0–1	3–4	21–43

TABLE 3. Rectmix predicates the expert chose for type 4

Rectangle	Length $\wedge$	ESAL$\wedge$	Axles $\wedge$	Weight
1	20–42	0–.43	3–3	12–29
2	23–44	0–1.2	2–3	26–47
3	23–29	0–6.7	2–4	27–71

TABLE 4. Rectmix predicates the expert chose for type 6

Rectangle	Length ∧	ESAL ∧	Weight
1	50–78	.1–2.2	37–77
2	51–77	0–.1	11–34
3	50–77	0–.2	30–41
4	52–78	2.4–6.3	74–101

TABLE 5. Rectmix predicates the expert chose for type 9 (Axles is always 5)

Type 4	Type 6	Type 9
45≤speed≤85	40≤speed≤88	39≤speed≤85
23≤length≤52	17≤length≤39	42≤length≤79
2≤axles≤3	3≤axles≤3	5≤axles≤5
13≤weight≤40	12≤weight≤49	16≤weight≤94

TABLE 6. Percentile predicates the expert chose for vehicle types 4, 6, and 9

Rectmix	Vehicle type	Average detection rate (%) Total	Length	ESAL	Speed	Axles	Weight
	4	15.5	42.5	7.7		4.4	7.4
	6	10.9	37.7	0.4		0.6	4.8
	9	2.3	5.0	3.4		0.0	0.9
Percentile	4	8.4	8.1		0.8	10.2	14.6
	6	20.2	30.5		22.2	17.0	11.3
	9	0.8	1.0		0.3	0.0	1.9

TABLE 7. Average detection rate

Vehicle type	Average misclassification rate (%)	
	Rectmix	Percentile
4	8.5	3
6	2.3	2.3
9	1	.8

TABLE 8. Average overall misclassification rate

List of Figures

1. Set up model of expected behavior by eliciting user expectations

    1. Identify appropriate techniques for the problem
    2. Use selected techniques from the technique tool kit to infer predicates that describe data behavior
    3. Interact, via the template mechanism, with the user to articulate expectations precisely using the predicates the techniques can output

4. Use the model (predicates) resulting from Item 1 as a proxy for missing specifications

    1. Detect semantic anomalies when a new observation falsifies a predicate

2. Tune the model to account for changing data behavior or user expectations

FIG. 1. Synopsis of our method

1. Run the techniques in the tool kit to infer predicates over subsets of the data.
2. Ask the user to classify each predicate as either “accept”, “update”, or “reject”.
3. Use the classification to instantiate templates.
4. Use the instantiated templates to filter the output of the tool kit techniques.
5. Give the filtered output to the anomaly detector and present to the user the resulting anomalies and their templates. Allow the user to change the classification.
6. Goto 1 or terminate when the user is happy with the classification.

FIG. 2. Synopsis of our template mechanism

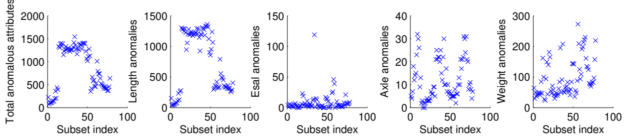

FIG. 3. Counts of anomalies detected using Rectmix predicates for vehicle type 6

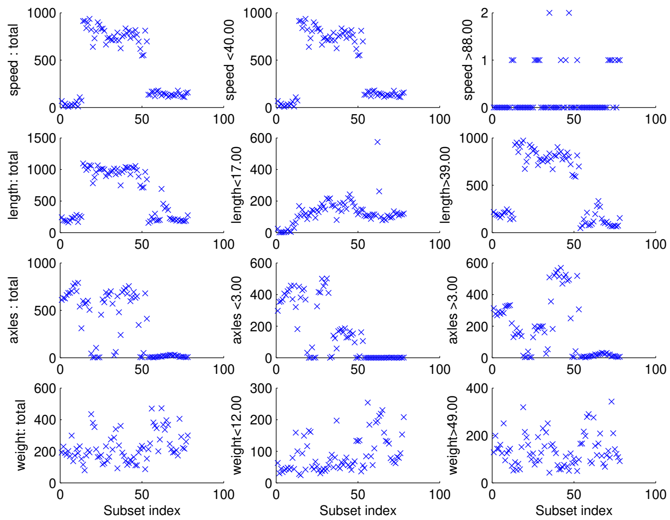

FIG. 4. Counts of anomalies detected using Percentile predicates for vehicle type 6