International Workshop on **Data Mining Methods for Anomaly Detection**

August 21, 2005
Chicago, Illinois, USA

Workshop Chairs:

Dragos Margineantu
Stephen Bay
Philip Chan
Terran Lane

KDD-2005 Workshop

on

Data Mining Methods for Anomaly Detection

Workshop Notes

Workshop Organizers

Dragos Margineantu, *The Boeing Company*
Stephen Bay, *PricewaterhouseCoopers*
Philip Chan, *Florida Institute of Technology*
Terran Lane, *University of New Mexico*

Workshop Program Committee

Naoki Abe, IBM TJ Watson
Carla Brodley, Tufts University
Vince Clark, University of New Mexico
Diane Cook, University of Texas, Arlington
Chris Drummond, The National Research Council of Canada
Wei Fan, IBM TJ Watson
Roman Fresnedo, The Boeing Company
Eamonn Keogh, University of California, Riverside
Adam Kowalczyk, National ICT Australia
Aleksandar Lazarevic, University of Minnesota
Wenke Lee, Gerogia Insititute of Technology
John McGraw, University of New Mexico
Ion Muslea, Language Weaver, Inc.
Raymond Ng, University of British Columbia
Galit Schmueli, University of Maryland, College Park
Mark Schwabacher, NASA, Ames Research Center
Salvatore Stolfo, Columbia University
Weng-Keen Wong, University of Pittsburgh
Bianca Zadrozny, IBM TJ Watson

Sponsors

The Boeing Company
and
PricewaterhouseCoopers

Table of Contents

An Empirical Bayes Approach to Detect Anomalies in Dynamic Multidimensional Arrays

Deepak Agarwal
AT&T Labs–Research
180 Park Avenue, Florham Park
New Jersey, United States
dagarwal@research.att.com

***Abstract*— We consider the problem of detecting anomalies in data that arise as multidimensional arrays with each dimension corresponding to the levels of a categorical variable. In typical data mining applications, the number of cells in such arrays are usually large. Our primary focus is detecting anomalies by comparing information at the current time to historical data. Naive approaches advocated in the process control literature do not work well in this scenario due to the multiple testing problem - performing multiple statistical tests on the same data produce excessive number of false positives. We use an Empirical Bayes method which works by fitting a two component gaussian mixture to deviations at current time. The approach is scalable to problems that involve monitoring massive number of cells and fast enough to be potentially useful in many streaming scenarios. We show the superiority of the method relative to a naive "per component error rate" procedure through simulation. A novel feature of our technique is the ability to suppress deviations that are merely the consequence of sharp changes in the marginal distributions. This research was motivated by the need to extract critical application information and business intelligence from the daily logs that accompany large-scale spoken dialog systems deployed by AT&T. We illustrate our method on one such system.**

I. Introduction

Consider a computational model of streaming data where a block of records are simultaneously added to the database at regular time intervals (e.g. daily, hourly etc) [15]. Our focus is on detecting anomalous behaviour by comparing data in the current block to some baseline model based on historic data. However, we are more interested in detecting anomalous patterns rather than detecting unusual records. A powerful way to accomplish this is to monitor statistical measures (e.g., counts, mean, quantiles) computed for combinations of categorical attributes in the database. Considering such combinations gives rise to a multidimensional array at each time interval. Each dimension of such an array corresponds to the levels of a categorical variable. We note that the array need not necessarily be complete i.e, only a subset of all possible cells might be of interest. A univariate measurement is attached to each *cell* of such an array. When the univariate cell measures are counts, such arrays are called contingency tables in Statistics. Henceforth, we also refer to such arrays as $cross\text{-}classified$ data streams. For instance, consider calls received at a call center and consider the two dimensional array where the first dimension corresponds to the categorical variable "caller intent"(reason for call) and the second dimension corresponds to the "originating location" (State where the call originates). A call center manager is often interested in monitoring daily percentages of calls that are attached to the cells of such an array. This is an example of a two dimensional cross-classified data stream which gets computed from call logs that are added to the database every day.

Some other examples are a) daily sales volume of each item sold at thousands of store locations for a retail enterprise. Detecting changes in cells might help for instance in efficient inventory management, provide knowledge of an emerging competitive threat. b) Packet loss among several source-destination pairs on the network of a major internet service provider (ISP). Alerts on cells in this application might help in identifying a network problem before it affects the customers. c) Emergency room visits at several hospitals with different symptoms. The anomalies in this case might point to an adverse event like a disease outbreak before it becomes an epidemic.

Apart from the standard reporting tasks of presenting a slew of statistics, it is often crucial to monitor a large number of cells simultaneously for changes that take place relative to expected behavior. A system that can detect anomalies by comparison to historical data provides information which might lead to better planning, new business strategies and in some cases might even lead to financial benefits to corporations. However, the success of such a system critically depends on having resources to investigate the anomalies before taking action. Too many false positives would require additional resources, false negatives would defeat the purpose of building the system. Hence, there is need to have sound statistical methods that could achieve the right balance between false positives and false negatives. This is particularly important when monitoring data classified into a large number of cells due to the well known multiple hypotheses testing problem.

Methods to detect changes in data streams have a rich literature in database and data mining. The primary focus of several existing techniques is efficient processing of data to compute appropriate statistics (e.g counts,quantiles,etc.), with change detection being done by using crude thresholds derived empirically or based on domain knowledge. For instance,[21] describe efficient streaming algorithms in the context of multiple data streams to compute statistics of interest (e.g.

pairwise correlations) with change being signalled using pre-specified rules. Non-parametric procedures based on Wilcoxon and Kolmogorov-Smirnov test statistics are proposed in [6] to detect changes in the statistical distribution of univariate data streams. In [20], the authors describe a technique to detect outliers when monitoring multiple streams by comparing current data to expected, the latter being computed using linear regression on past data. Our work, though related has important differences. First, we are dealing with cross-classified data streams which introduce additional nuances. Second, we adjust for multiple testing which is ignored by [20]. We are also close in spirit to [17] who use a Bayesian network for their baseline model and account for multiple testing using randomization procedures.

Adjusting for margins: When monitoring cells for deviations, it is prudent to adjust for sharp changes in the marginal statistics. Failure to do so may produce anomalies which are direct consequences of changes in a small number of marginals. For instance, it is not desirable to produce anomalies which indicate a drop in sales volume for a large number of items in a store merely because there was a big drop in the overall sales volume due to bad weather. We accomplish this by adjusting for the marginal effects in our statistical framework.

Multiple testing, also known as the *multiple comparisons* problem has a rich literature in Statistics dating back to the 1950s. Broadly speaking, if multiple statistical tests are simultaneously performed on the same data, it tends to produce false positives even if nothing is amiss. This can be very serious in applications. Thus, if a call center manager is monitoring repair calls from different states, he might see false positives on normal days and stop using the system. Much of the early focus in multiple testing was on controlling the family wise error rates (FWER) (probability of at least one false detection). If K statistical tests are conducted simultaneously at *per comparison error rate* (PCER) of α (probability of false detection for each individual test), the FWER increases exponentially with K. Bonferroni type corrections which adjust the PCERs to α/K achieving a FWER of α are generally used. However, such corrections may be unnecessarily conservative. This is especially the case in data mining scenarios where K is large. An alternate approach have been proposed in [5] which uses shrinkage estimation in a hierarchical Bayesian framework in combination with decision theory. Later, [19] proposed a method based on controlling the False Discovery Rate (FDR)(proportion of falsely detected signals) which is less strict than FWER and generally leads to gain in power compared to FWER approaches. In fact, controlling the FDR is better suited to high dimensional problems that arise in data mining applications and has recently received a lot of attention in Statistics, especially in genomics. Empirical and theoretical connections between Bayesian and FDR approaches have been studied in [11][9]. Another approach to tackle the curse of multiple testing is based on randomization [10] but might be computationally prohibitive in high dimensions. We take a hierarchical Bayesian approach in a decision theoretic framework similar in spirit to [5] but replace the normal prior with a two component mixture as in [14]. An added advantage of the hierarchical Bayesian approach over FDR is the flexibility it provides to account for additional features that might be present in some situations. For instance, if one of the dimension corresponds to spatial locations, correlations induced due to geographic proximity are expected and could be easily accounted for. For a detailed introduction to hierarchical Bayesian models, we refer the reader to [3].

A. Motivating application

This research was motivated by the need to build a data mining tool which extracts information out of spoken dialog systems deployed at call centers. The data mining tool built to accomplish this is called the *VoiceTone Daily News*(VTDN)[7] and supplements AT&T's call center service called VoiceTone by *automatically* extracting critical service information and business intelligence from records of dialogs resulting from a customer calling an automated help desk. The *Daily News* uses the spoken dialog interaction logs to automatically detect interesting and unexpected patterns and presents them in a daily web-based newsletter intended to resemble on-line news sites such as CNN.com or BBC.co.uk. Figure1 shows an example of the front page of such a newsletter. The front page news items are provided with links to precomputed static plots and a drill down capability, powered by a query engine and equipped with dynamic visualization tools that enables a user to explore relevant data pertaining to news items in great detail. The data mining task in this application involves three challenging steps, viz., a) extraction of relevant features from dialogues b) detect changes in these features and c) provide a flexible framework to explore the detected changes. Our focus in this paper is on task b), for complete details on a) and c) we refer the reader to [7].

To end this section, we briefly summarize our contributions below.

- We present a framework to detect anomalies in cross-classified data streams with potentially large number of cells. We correct for multiple testing using a hierarchical Bayesian model and suppress redundant alerts caused due to changes in the marginal distributions.
- Empirically illustrate the superiority of our method by comparison to a PCER method and illustrate it on a novel application that arise in speech mining.

The roadmap is as follows - section II describes the theoretical setup for our problem followed by a brief description of the hierarchical Bayesian procedure called *hbmix*. Sections III and IV describe our data in the context of the VTDN application. Section V compare *hbmix* to a PCER method through simulation followed by an illustration of *hbmix* on actual data in section VI. We end in section VII with discussion and scope for future work.

II. Theoretical framework

For ease of exposition, we assume the multidimensional array consists of two categorical variables with I and J levels

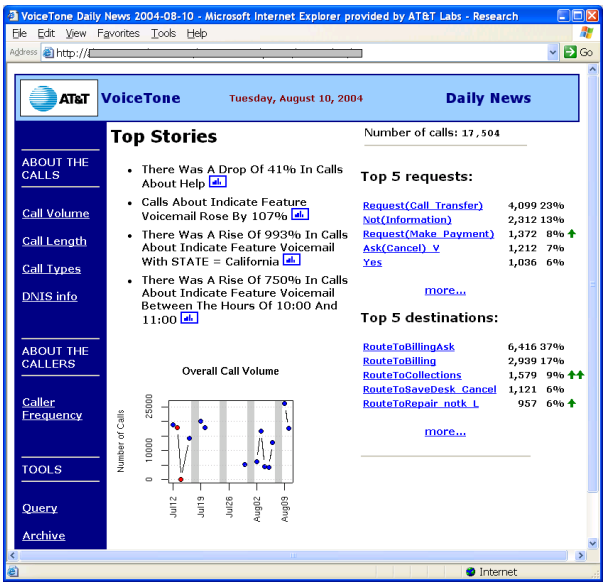

Fig. 1. *The front page for VTDN: a simulated example.*

respectively and note that generalization to higher dimensions is similar. In our discussion, we assume the array is complete. In practice, this is usually not the case but the theory still applies. Let the suffix ijt refer to the i^{th} and j^{th} levels of the first and second categorical variables respectively at time t. Let y_{ijt} denote the observed value which is assumed to follow a gaussian distribution. Often, some transformation of the original data might be needed to ensure this is approximately true. For instance, if we observe counts, a square root transformation is adequate, for proportions arc sine ensures approximate normality. In general, the Box-Cox transformation $((y+m)^p - 1)/p$ with parameters m and p chosen to 'stabilize' variance if it depends on the mean is recommended. Usually, p is constrained to lie between 0 and 1, and $p \to 0$ implies a log transformation. In fact, one could choose reasonable values of these parameters using some initial training data.

For time interval t, we may want to detect anomalies after adjusting for changes in the marginal means. We show the difference between adjusting and not adjusting the margins by using a toy example. Consider a 2×2 table, the levels of the row factor being A,B and the column factor being a,b respectively. We denote the 4 cell entries corresponding to (Aa,Ab,Ba,Bb) by a vector of length 4. Let the expected values be (50,50,50,50) and the observed values be (25,25,75,75). Then the raw changes are (-25,-25,25,25) which are all large. The deviations after adjusting for the changes in the row and columns means are (0,0,0,0) producing no anomalies. Note that the significant values in the non-adjusted changes can be ascribed to a drop in the first row mean and a rise in the second row mean. Hence, non-adjusted cell changes contain redundant information. In such situations, adjusting for margins is desirable.

However, marginal adjustments are not guaranteed to produce a parsimonious explanation of change in all situations. For instance, consider a second scenario where the observed values are (50,0,50,100). The raw and adjusted changes are (0,-50,0,50) and (25,-25,-25,25) respectively. The raw changes in this case produce two alerts which pinpoint the culprit cells that caused deviations in the row means, the adjusted changes would alert all four cell entries. To summarize, adjusting the margins work well when changes in the marginal means can be attributed to some common cause affecting a large proportion of cells associated with the margins. Also, one byproduct is the automatic adjustment of seasonal effects, holiday effects, etc., that affect the marginals, commonplace in applications. However, if the marginal drops/spikes could be attributed to a few specific cells and the goal is to find them, the unadjusted version is suitable. In our application, we track changes in the margins separately(using simple process control techniques) and run both adjusted and unadjusted versions but are careful in interpreting the results. In fact, the adjusted version detects changes in interactions among the levels of categorical variables which might be the focus of several applications. For instance, in the emergency room example it is important to distinguish an anthrax attack from the onset of flu season. Since an anthrax attack is expected to be localized initially, it might be easier to identify the few culprit hospitals by adjusting for margins. Also, in higher dimensions one might want to adjust for higher order margins, which is routine in our framework. For instance, adjusting for all two-way margins in a three dimensional array would detect changes in third order interactions.

Let H_{t-1} denote historical information upto time $t-1$. Deviations at time t are detected by comparing the observed values y_{ijt}'s with the corresponding *posterior predictive* distributions (expected distribution of data at time t based on historic data until $t-1$) which in our set up are gaussian with means $\mu_{ijt} = E(y_{ijt}|H_{t-1})$ and variances $\sigma^2_{ijt} = Var(y_{ijt}|H_{t-1})$ (known at t from historic data). Strategies to compute the posterior predictive distributions are discussed in section II-A.

Letting $y_{ijt} \sim N(\mu_{ijt} + u_{ijt}, \sigma^2_{ijt})$ ($X \sim N(m, \sigma^2)$ denotes the random variable X has a univariate normal distribution with mean m and variance σ^2), the goal is to test for zero values of u_{ijt}'s. For marginal adjustment, write $u_{ijt} = u_t + ur_{it} + uc_{jt} + \Delta_{ijt}$ (u_t, ur_{it} and uc_{jt} are overall, row and column effects respectively at t which are unknown but plugged-in by their best linear unbiased estimates) and the problem reduces to testing for zero values of Δ_{ijt}'s. More formally, with $e_{ijt} = y_{ijt} - \mu_{ijt}$ and $\delta_{ijt} = e_{ijt} - u_t - ur_{it} - uc_{jt}$, $\delta_{ijt} \sim N(\Delta_{ijt}, \sigma^2_{ijt})$ and we want to test multiple hypotheses $\Delta_{ijt} = 0$ ($i = 1, \cdots, I$, $j = 1, \cdots, J$). For the unadjusted version, $\delta_{ijt} = e_{ijt}$. We note that adjusting for higher order interactions is accomplished by augmenting the linear model stated above with the corresponding interaction terms. For a detailed introduction to linear model theory for k-way tables, we refer the reader to [4].

A naive PCER approach generally used in process control[2] is to estimate Δ_{ijt} with δ_{ijt} and declare the ij^{th} cell an

anomaly if

$$|\delta_{ijt}/\sigma_{ijt}| > M_1(3 \text{ is a common choice}). \quad (1)$$

The central idea of the hierarchical Bayesian method *hbmix* is to assume Δ_{ijt}'s are random samples from some distribution G_t. The form of G_t may be known but depend on unknown parameters. For instance, [8] assumes G to be $N(\theta_{0t}, \tau_t^2)$ and discuss the important problem of eliciting prior probabilites for the unknown parameters. In [13], a non-parametric approach which assigns a Dirichlet process prior to G_t is advocated but not pursued here due to computational complexity. Following [14] and [9], we take a semi-parametric approach which assumes G_t to be a mixture $P_t 1(\Delta = 0) + (1 - P_t)N(0, \tau_t^2)$ i.e. a proportion P_t of cells don't change at time t while the remainder are drawn from a normal distribution. We assume a log-logistic prior for τ_t^2 centered at the harmonic mean of σ_{ijt}^2's as in [8] and a half-beta prior($\pi(x) \propto x^m, m > 0$) centered around $\hat{P_{t-1}}$ for P_t ($\hat{P_{t-1}}$ is the estimated value of P_{t-1} at time $t-1$.) At time $t = 0$, we assume a uniform prior for P_0.

Conditional on the hyperparameters (P_t ,τ_t^2), δ_{ijt}'s are independently distributed as a two-component mixture of normals $P_t N(0, \sigma_{ijt}^2) + (1 - P_t)N(0, \sigma_{ijt}^2 + \tau_t^2)$. The joint marginal likelihood of δ_{ijt}'s are the product of the individual two-component mixture densities and from Bayes rule the posterior distribution of (P_t, τ_t^2) is proportional to the joint likelihood times the prior. The posterior distribution of Δ_{ijt} conditional on (P_t, τ_t^2) is degenerate at 0 with probability Q_{ijt} and with probability $1 - Q_{ijt}$ it follows $N(b_{ijt}, v_{ijt}^2)$ where

$$\begin{aligned} \frac{Q_{ijt}}{(1 - Q_{ijt})} &= \frac{P_t N(\delta_{ijt}; 0, \sigma_{ijt}^2)}{(1 - P_t)N(\delta_{ijt}; 0, \sigma_{ijt}^2 + \tau_t^2)} \\ b_{ijt} &= \tau_t^2 \delta_{ijt}/(\tau_t^2 + \sigma_{ijt}^2) \\ v_{ijt}^2 &= (\tau_t^2 \sigma_{ijt}^2)/(\tau_t^2 + \sigma_{ijt}^2) \end{aligned}$$

($N(x; m, s^2)$ denotes density at x for a normal distribution with mean m and variance s^2.) An Empirical Bayes approach makes inference about Δ_{ijt}'s by using plug-in estimates of the hyperparameters (P_t, τ_t^2) which are obtained as follows - compute the mode $(\tilde{P}_t, \tilde{\tau_t^2})$ by maximizing the posterior of (P_t, τ_t^2) (for very large values of K, we use a data squashing technique [16]) and define the estimates as $(\hat{P}_t, \hat{\tau_t^2}) = \lambda(\tilde{P}_t, \tilde{\tau_t^2}) + (1 - \lambda)(\hat{P_{t-1}}, \hat{\tau_{t-1}^2})$, where the smoothing constant is chosen in the interval $[.95, .99]$. At time $t = 0$, $\lambda = 1$. This exponential smoothing allows hyperparameters to evolve smoothly over time. In a fully Bayesian approach, inference is obtained by numerically integrating with respect to the posterior of (P_t, τ_t^2) using an adaptive Guass Hermite quadrature. Note that the posterior distribution of Δ_{ijt} depends directly on δ_{ijt} and indirectly on the other δ's through the posterior of the hyperparameters. Generally, such "borrowing of strength" makes the posterior means of Δ_{ijt}'s regress or "shrink" toward each other and automatically builds in penalty for conducting multiple tests.

A natural rule is to declare the ij^{th} cell anomalous when the posterior odds $\frac{Q_{ijt}}{1-Q_{ijt}} < c$, which yields (after simplication) $|\delta_{ijt}/\sigma_{ijt}| > A_{ijt}$ where

$$A_{ijt} = \sqrt{(}2(1 + e^{\kappa_{ijt}})(\eta + .5log(1 + e^{-\kappa_{ijt}}) - log(c))) \quad (2)$$

($\kappa_{ijt} = log(\sigma_{ijt}^2/\tau_t^2)$ (log of the variance ratio) and $\eta_t = log(P_t/(1 - P_t))$ (prior log odds)) with A_{ijt} in (2) being monotonically increasing in both κ_{ijt} and η_t. Thus, the cell penalty increases monotonically with predictive variance. Also, the overall penalty of the procedure at time t depends on the hyperparameters which are estimated from data. In fact, replacing σ_{ijt}^2's by their harmonic mean σ_t^2 in (2) gives us a constant A_t which provides a good measure of the global penalty imposed by *hbmix* at time t. However, the loss assigned to false negatives by (2) does not depend on the magnitude of deviation of Δ's from zero. Motivated by [5] and [14], we use a loss function

$$L(a, \Delta) = 1(\Delta = 0)1(a = C) + c|\Delta|^p 1(\Delta \neq 0)1(a = N) \quad (3)$$

where $p \geq 0$, $c(> 0)$ is a parameter which represents the cost of a false negative relative to a false positive, C denotes change and N denotes no change. With $p = 0$, we recover (2) and $p = 1$ gives us the loss function in [14]. In fact, $p = 1$ is a sensible choice for the VTDN application where missing a more important news item should incur a greater loss. In our application we assume $c = 1$ but remark other choices elicitated using domain knowledge are encouraged. Having defined the loss function, the optimal action(called the Bayes rule) minimizes the posterior expected loss of Δ. In our setup, we declare a change if $E(L(C, \Delta)) - E(L(N, \Delta)) < 0$ noting that the expression is a known function of hyperparameters and could be computed either by using plug-in estimates or numerical integration.

A. Calculating posterior predictive means and variances

Two popular approaches used to capture history H_t are *sliding window* and *exponential smoothing*. In the former, a window size w is fixed a-priori and the distribution at t is assumed to depend *only* on data in the window $[t-1-w, t-1]$. Extensive research on fast computational approaches to maintain summary statistics under this model have been done (see [1] for an overview). In an exponential smoothing model, a decay parameter $\lambda \in (0, 1)$ is used to downweight historic data with the weights dropping exponentially in the past.

In principle, any statistical model that could provide an estimate of posterior predictive means and variances could be used to obtain μ_{ijt}'s and σ_{ijt}^2's. For instance, [20] use a linear model, [18] use an AR model, [12] provide a general framework using state space models, the possibilities are numerous and depends on the application at hand. However, elaborating on appropriate models is not the focus of this paper, we assume it has been chosen and trained judiciously by the user. Also, to be useful in streaming scenarios, the chosen model should easily adapt to new data.

System: Hello, this is AT&T, how may I help you?
User: I want to talk to a human $\longrightarrow$ *Request(Call_Transfer)*
System: Would you like to speak to an agent?
User: yes . $\longrightarrow$ *Yes*
System: Okay, I will transfer your call. Is there anything else I can help you with today?
User: No thanks $\longrightarrow$ *No*
System: Thank you for calling AT&T. Goodbye.

Fig. 2. An simulated example of a VoiceTone dialog

For the VTDN application illustrated in this paper, we use a sliding window to capture H_t. We assume the cells are uncorrelated and for the ij^{th} cell, $y_{ijk}, k = t-w-1, \cdots, t-1$ are iid normal with mean μ_{ij} and variance σ^2_{ij}. Then, the posterior predictive mean μ_{ijt} is the sample mean of y_{ijk}'s and the posterior predictive variance σ^2_{ijt} is $((w+1)/w)\sigma^2_{ij}$. Since σ^2_{ij} is unknown, it is replaced by its estimator s^2_{ij}, the sample variance of y_{ijk}'s. In order to adjust for seasonal effects, a separate sliding window is maintained for each season.

III. VoiceTone Daily News

We illustrate and evaluate *hbmix* on a customer care(BCC) application supported by VoiceTone(client's identity not disclosed due to reasons of confidentiality). Before we describe the data, a high level description of the features extracted are given below(see [7] for complete details).

A dialog is a stream of events (in XML) which is divided into a sequence of turns. A turn consists of a system prompt, the user response as recognized by the system, and any records associated with the system's processing of that response. Each turn is mapped to one of a set of *call types* using BoosTexter - a member of the AdaBoost family of large-margin classifiers. A dialog ends when a goal is achieved by completing a transaction, for instance, or routing the user to an appropriate destination. A simulated example is shown in Fig 2, illustrating the system's classifications (*Request(Call_Transfer)*, *Yes*, *No*). The features that are currently extracted include the originating telephone number for the call (**ANI**), the number of turns in a dialog (**NTURNS**), the length of the call (**DURATION**), any final routing destination the call gets routed to (**RD**) and the **final actionable call type(FACT)**. This is the last call type the classifier obtained in the course of the system's dialog with the user before routing. For instance, in figure 2 the value of FACT is "Request(Call_Transfer)" and that of RD (not shown in the figure but computed based on the location the call gets routed to) is "Repair" if the call gets routed correctly. FACT and RD are primary features tracked by the "Daily News" alert system. The FACT is our closest approximation to the caller's intent. This is of particular interest to VoiceTone's clients (banks, pharmacies, etc.), who want to know what their customers are calling about and how that is changing. The RD, particularly together with time of day information and geographic information derived from the ANI, provides information on call center load to support decision-making about provisioning and automation.

IV. Data description for business customer care

Due to proprietary nature of the data, all dates were translated by a fixed number of days i.e. acutual date $=$ date used in the analysis $+ x$, where x is not revealed. The news page for this application is updated on a daily basis. The system handles approximately $15K - 20K$ care calls per day. Features tracked by *hbmix* include average call duration cross-classified by FACT X STATE (STATE where the calls originate are derived using ANI), RD X STATE, FACT X Hourofday, RD X STATE. The system is flexible enough to accept any new combination of variables to track. We present an analysis that tracks proportions for FACT X STATE.

There are about 100 categories in FACT, 50 states we're interested in. At time t, we only include cells that have occured at least once in the historic window of length w which, for a window size of 10 days (we choose this by a using predictive loss criteria on initial training data) results in about 2900 categories being monitored on average. The system went live last week of January, 2004. We use data ending April, 2004 as our training set to choose an appropriate window size and to choose parameters for a simulation experiment discussed later. Finally, we run *hbmix* on data from May, 2004 through Jan 2005.

Our cell measurements are proportions p_{ij} computed from the block that gets added to the database every day. For the ij^{th} cell, p_{ij} = number of calls in ij^{th} cell/Total number of calls. This multinomial structure induces negative correlations among cells. Under a multinomial model, the negative correlation between any pair of cells is the geometric mean of their odds ratio. This is high only if both odds ratio are large, i.e., if we have several big categories. From the training data we compute the 95^{th} percentile of the distribution of p's for each cell. The top few cells have values $.07, .05, .04, .03$ which means the correlation is approximately bounded below by $-.06$. To ensure symmetry and approximate normality, we compute the score $y_{ij} = Sin^{-1}\sqrt{(p_{ij})} / \sum_{ij} Sin^{-1}\sqrt{(p_{ij})}$ with the normalization meant to preserve the multinomial structure. The top few cells after transformation have 95^{th} percentile values of $.012, .009, 0.009, 0.008$ which gives a lower correlation bound of about $-.01$. Hence, the assumption of cell *independence* seems reasonable in this case.

V. Simulation to evaluate hbmix

Here, our goal is to compare the performance of *hbmix* with a naive PCER approach for the BCC application. We take a simulation based approach, i.e., we generate data whose statistical properties are close to that of our actual data during the training period, artificially inject anomalies and then score the two methods under consideration.

We compare the methods based on performance at a single time interval. We simulate K streams (K is the number of cells in our stream, we ignore the issue of adjusting for margins since it is not relevant for this experiment) at $w+1$ time points introducing anomalies only at the last time point and compare the FDR and false negative rates based on several repititions of the experiment. Since the difference between FDR and false

negative rate is not symmetric, we tweak the value of M_1 so that the false negative rate for PCER matches the one obtained for *hbmix* with $c = 1$. The tweaking is done using a bisection algorithm due to the monotonic dependence of false negative rate on M_1. Simulation details are given below.

- Generate $(\mu_1, \cdots, \mu_K)$ such that μ_i's are iid from some distribution F. The cell means computed from training data fitted a log-normal distribution well(original arcsine scores were multiplied by 1000), hence we choose F =lognormal with location parameter = -1.36 and scale parameter = $\sqrt{(}1.46)$.
- The cell variances σ_i^2 were generated from the following log-linear model (which fitted the training data well) $log(\sigma_i^2) = -2.37 + .64log(\mu_i) + N(0, .43)$
- For each i, simulate $w+1$ observations as iid $N(\mu_i, \sigma_i^2)$.
- At time $w+1$, randomly select 100 streams, add "anomalies" generated from $N(0, 3)$.
- Detect anomalies at $w + 1$ using *hbmix*(we choose $w = 10$, $p = 1, c = 1$) with both empirical Bayes and full Bayes methods, tweak M_1 to match the false negative rate as discussed earlier.
- The above steps are repeated 100 times, results are reported in Table I

TABLE I

Results comparing hbmix and PCER with 100 *true anomalies based on* 100 *replications of each experiment*

K	false neg rate(%)	M_1	FDR(%) (hbmix)	FDR(%) (PCER)	t-stat
500	18.4	3.1	4.4	7.4	6.8
1000	19.7	3.5	5.7	9.2	7.4
2000	20.9	3.8	7.7	12.4	8.3
5000	21.9	4.0	13.8	21.4	9.9

TABLE II

Comparing time(in secs) for Full and Empirical Bayes procedures

K	EB	FB
500	.53	5.8
1000	2.2	10.6
2000	3.6	18.0
5000	14.1	47.0

The FDR for *hbmix* is consistently smaller than PCER. Moreover, the difference increases with K. Also, the difference is statistically significant indicated by the significant t-statistics(p-values were all close to 0.00) obtained using a two-sample t-test. For *hbmix*, we obtained similar results for both the Empirical Bayes and full Bayes methods. Table II compares the computational time for the two methods using our non-optimized code. The full Bayes method is roughly 10 times slower and hence we recommend Empirical Bayes if the main goal is inference on Δs'.

VI. Data Analysis

In this section, we present results of our analyses on customer care from May,2004 to January, 2005 for the combination FACT X State. We apply *hbmix* both adjusting and not adjusting for the marginal changes (call them *adjusted hbmix* and *non-adjusted hbmix* respectively). In figure 3, the top panel shows time series plots of A_t for both versions of *hbmix* (horizontal gray line shows the constant threshold of 3 for PCER). As noted earlier, A_t provides an estimate of the penalty built into *hbmix* at each time interval. The bottom panel shows the number of alerts obtained using the three procedures. The figure provide insights into the working of adjusted and non-adjusted *hbmix* relative to the PCER method. Large values of A_t correspond to periods when the system is relatively stable producing a few alerts. (e.g., mid June through mid July.) In general, the PCER produces more alerts compared to *hbmix*. On a few days (the ones marked with dotted lines on the bottom panel of figure 3), adjusted *hbmix* drastically cuts down on the number of alerts relative to non-adjusted *hbmix*. These are days when a system failure caused a big increase in HANGUP rate triggering several related anomalies. The adjusted version always gives smaller number of alerts compared to PCER and it never produces more than a couple of extra alerts compared to the unadjusted version. In fact, there are about 30 days where the adjusted version produces one or two alerts when the unadjusted version produce none. These represent subtle changes in interactions. To illustrate the differences between adjusted and unadjusted *hbmix*, we investigate the alerts obtained on Sept 3^{rd} (we had other choices as well but believe this is sufficient to explain our ideas).

Sept 3rd, 2004: This is an interesting day. Our univariate alert procedures don't point to anything for FACT, we notice a couple of spikes in the STATE variable for Maryland (3.2% to 7.4%) and Washington D.C.(.6%to 2.1%). There are 8 alerts common to both versions of *hbmix*. Interestingly, these alerts are spatially clustered, concentrated to states that are geographically close to each other. There is one alert (an increase) that appear only with the unadjusted *hbmix*, viz., about *Indicate(Service_Line)* in Maryland. One alert indicating increase in *Ask(Cancel)* in Connecticut is unique to the adjusted version. Figure 4 shows the difference in the *Indicate(Service_Line)* alert in Maryland using the adjusted and non-adjusted *hbmix*. The broken lines are the appropriate control limits about the historic mean. (For the marginals, the control limits are computed using PCER.) It provides an illustrative example of how the adjusted version works, the spike in Maryland when adjusted for reduce severity and the alert is dropped. Figure 5 shows an example where adjusted *hbmix* produce the alert missed by the unadjusted one on Sept 3^{rd}. Although marginal changes are well within their respective control limits, drops in *Ask(Cancel)* and *connecticut* increase severity of the alert with the adjusted version.

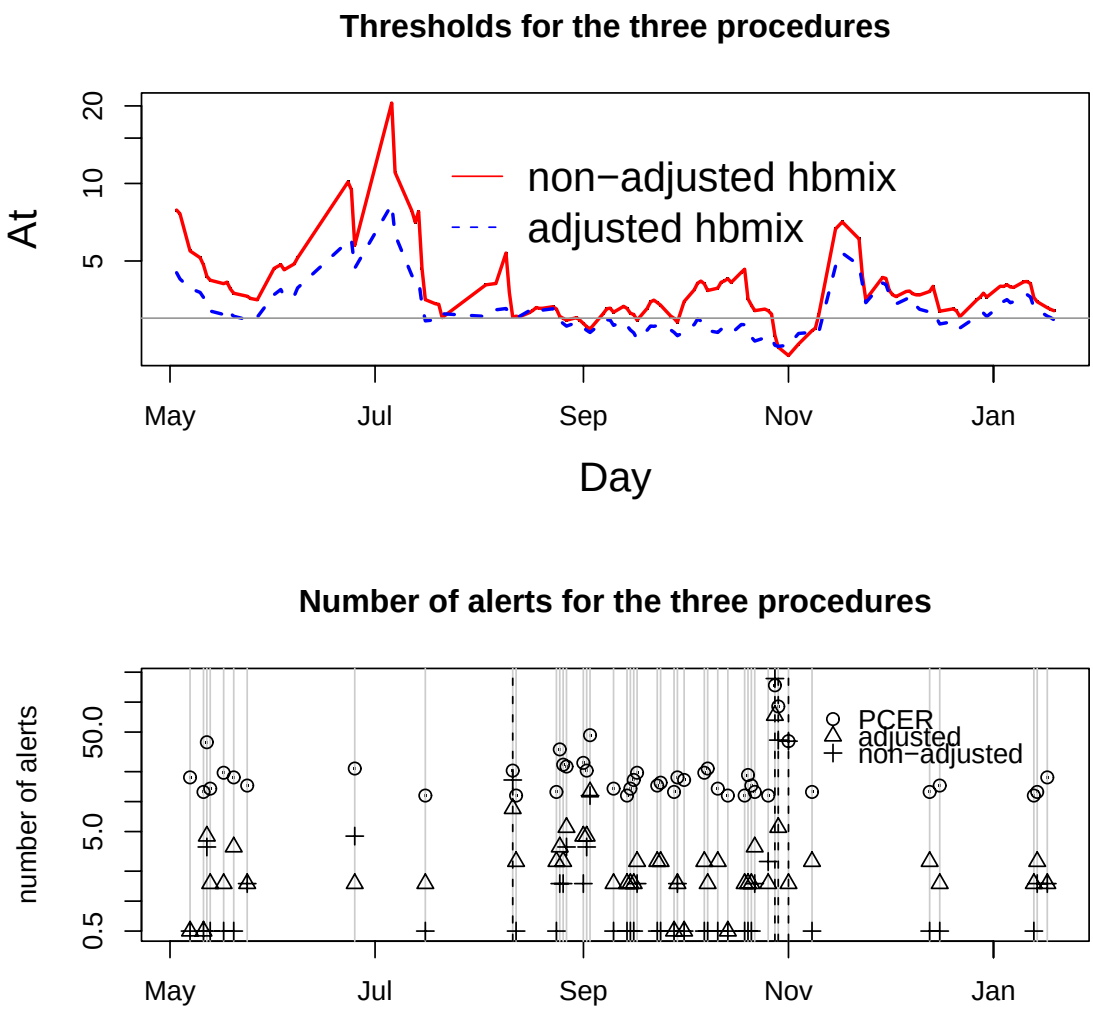

Fig. 3. *Top panel give values of A_t over time for the adjusted and non-adjusted hbmix, bottom panel gives number of alerts for the three procedures. The y-axes are on the log_e scale for both figures with .5 added to the number of alerts.*

VII. Discussion

We proposed a framework for detecting anomalies in massive cross-classified data streams. We described a method to reduce redundancy by adjusting for marginal changes. We solve the multiple testing problem using a hierarchical Bayesian model within a decision theoretic framework and prove the superiority of *hbmix* to a naive PCER method through simulation. We illustrate *hbmix* on a new speech mining application.

Ongoing work includes relaxing the gaussian assumption for δ's to the one-parameter exponential family. We are also working on methods to combine adjusted and unadjusted *hbmix* to automatically produce a parsimonious explanation of anomalies. For instance, in 2-d, this could be done by testing for mean shifts in the distribution of individual row and column vectors using non-parametric quantile based tests that are robust to outliers. Rows and columns that are subject to shifts relative to historic behaviour would be the only ones that get adjusted.

Acknowledgements

I thank Divesh Srivastava and Chris Volinsky for useful discussions.

References

[1] B.Babcock, S.Babu, M.Datar, R.Motwani, and J.Widom. Models and issues in data stream systems. In *PODS*, Madison,Wisconsin,USA, 2002.

[2] G. E. Box. *Time series analysis : forecasting and control*. Holden-Day, 1970.

[3] B.P.Carlin and T.A.Louis. *Bayes and Empirical Bayes methods for data analysis 2nd Ed.* Chapman and Hall/CRC Press, 2000.

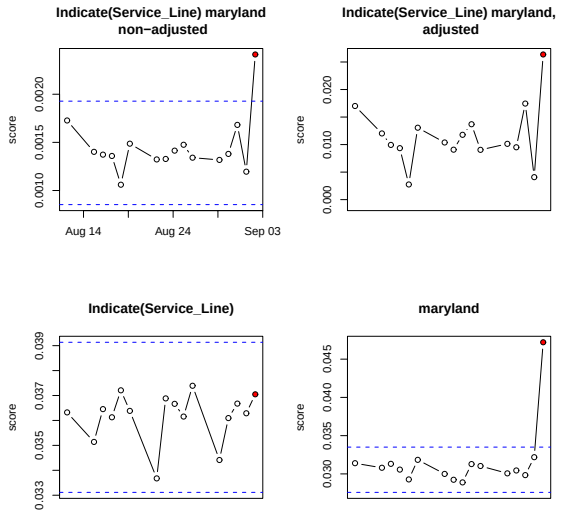

Fig. 4. *Example on sept 3^{rd} where the adjusted version drops an alert caused due to a spike in one of the marginal means. Absence of control lines in one of the plot indicate all points are within the control limits.*

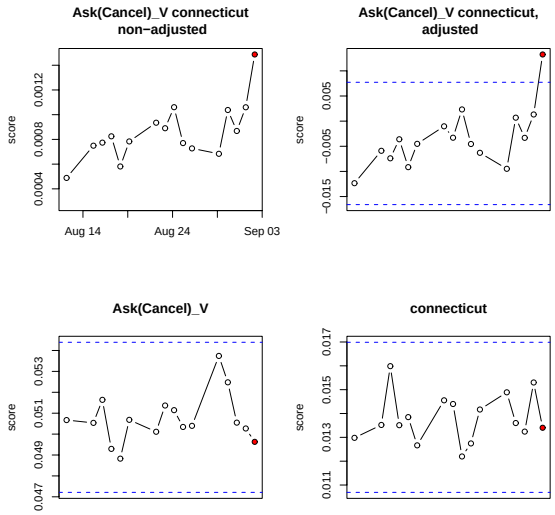

Fig. 5. *Example on sept 3^{rd} where adjusted version detects an alert missed by the unadjusted version. Absence of control lines in one of the plot indicate all points are within the control limits.*

[4] C.R.Rao. *Linear Statistical Inference and Its Applications 2nd Ed.* Wiley, 2002.

[5] D.B.Duncan. A bayesian approach to multiple comparisons. *Technometrics*, 7:171–222, 1965.

[6] D.Kifer, S.Ben-David, and J.Gehrke. Detecting change in data streams. In *Proc. of the 30th VLDB conference*, pages 180–191. Toronto, Canada, August 2004.

[7] S. Douglas, D. Agarwal, T. Alonso, R. Bell, M. Rahim, D. F. Swayne, and C. Volinsky. Mining Customer Care Dialogs for "Daily News". In *INTERSPEECH-2004*, Jeju, Korea, 2004.

[8] W. DuMouchel. A bayesian model and graphical elicitation procedure for multiple comparisons. In *J.M.Degroot, M.H.Lindley, D.V.Smith, A.F.M.(Eds.), Bayesian Statistics 3*. Oxford University Press. Oxford, England, 1988.

[9] C. Genovese and L.Wasserman. Bayesian and frequentist multiple testing. In *Bayesian Statistics 7 – Proc. of the 7th Valencia International Meeting*, pages 145–162, 2003.

[10] P. Good. *Permutation tests - a practical guide to resampling methods for testing hypotheses*. Springer-Verlag, 2nd edition, New York, 2000.

[11] J.P.Shaffer. A semi-bayesian study of duncan's bayesian multiple comparison procedure. *Journal of statistical planning and inference*, 82:197–213, 1999.

[12] M.West and J.Harrison. *Bayesian Forecasting and Dynamic Models*. Springer, 1997.

[13] R.Gopalan and D.A.Berry. Bayesian multiple comparisons using dirichlet process priors. *Journal of the American Statistical Association*, 93:1130–1139, 1998.

[14] J. Scott and J. Berger. "an exploration of aspects of bayesian multiple testing". Technical report, Institute of Statistics and Decision Science, 2003.

[15] V.Ganti, J.E.Gehrke, and R.Ramakrishnan. Mining data streams under block evolution. *Sigkdd explorations*, 3:1–10, january 2002.

[16] W.DuMouchel, C.Volinsky, T.Johnson, C.Cortes, and D.Pregibon. Squashing flat files flatter. In *Proc. of the 5th ACM SIGKDD conference*, pages 6–15. San Diego, California,USA, August 1999.

[17] W.Wong, A.Moore, G.Cooper, and M.Wagner. Bayesian network anomaly pattern detection for disease outbreaks. In *Proc. of the 20th International Conference on Machine Learning*, pages 808–815. Washington, DC, USA, 2003.

[18] K. Yamanishi and J. ichi Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proc. of the 8th ACM SIGKDD conference*, pages 676–681. Edmonton,Canada, August 2002.

[19] Y.Benjamini and Y.Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the royal statistical society, series B*, 57:289–300, 1995.

[20] B.-K. Yi, N.Sidiropoulos, T.Johnson, H.V.Jagadish, C.Faloutsos, and A.Biliris. Online data mining for co-evolving time sequences. In *Proc. of the 16th International Conference on Data Engineering*, pages 13–22. San Diego, California,USA, March 2000.

[21] Y.Zhu and D.Shasha. Statstream: Statistical monitoring of thousands of data streams in real time. In *Proc. of the 28th VLDB conference*, pages 358–369. HongKong,China, 2002.

Discovering hidden association rules

Marco-Antonio Balderas†, Fernando Berzal⋆, Juan-Carlos Cubero⋆, Eduardo Eisman†, Nicolás Marín⋆
Department of Computer Science and AI
University of Granada
Granada 18071 Spain
⋆{fberzal|jc.cubero|nicm}@decsai.ugr.es,†{mbald|eeisman}@correo.ugr.es

Abstract

Association rules have become an important paradigm in knowledge discovery. Nevertheless, the huge number of rules which are usually obtained from standard datasets limits their applicability. In order to solve this problem, several solutions have been proposed, as the definition of subjective measures of interest for the rules or the use of more restrictive accuracy measures. Other approaches try to obtain different kinds of knowledge, referred to as peculiarities, infrequent rules, or exceptions. In general, the latter approaches are able to reduce the number of rules derived from the input dataset. This paper is focused on this topic. We introduce a new kind of rules, namely, anomalous rules, which can be viewed as association rules hidden by a dominant rule. We also develop an efficient algorithm to find all the anomalous rules existing in a database.

1. Introduction

Association rules have proved to be a practical tool in order to find tendencies in databases, and they have been extensively applied in areas such as market basket analysis and CRM (Customer Relationship Management). These practical applications have been made possible by the development of efficient algorithms to discover all the association rules in a database [11, 12, 4], as well as specialized parallel algorithms [1]. Related research on sequential patterns [2], associations varying over time[17], and associative classification models [5] have fostered the adoption of association rules in a wide range of data mining tasks.

Despite their proven applicability, association rules have serious drawbacks limiting their effective use. The main disadvantage stems from the large number of rules obtained even from small-sized databases, which may result in a second-order data mining problem. The existence of a large number of association rules makes them unmanageable for any human user, since she is overwhelmed with such a huge set of potentially useful relations. This disadvantage is a direct consequence of the type of knowledge the association rules try to extract, i.e, frequent and confident rules. Although it may be of interest in some application domains, where the expert tries to find *unobserved* frequent patters, it is not when we would like to extract *hidden* patterns.

It has been noted that, in fact, the occurrence of a frequent event carries less information than the occurrence of a rare or hidden event. Therefore, it is often more interesting to find surprising non-frequent events than frequent ones [7, 27, 25]. In some sense, as mentioned in [7], the main cause behind the popularity of classical association rules is the possibility of building efficient algorithms to find all the rules which are present in a given database.

The crucial problem, then, is to determine which kind of events we are interested in, so that we can appropriately characterize them. Before we delve into the details, it should be stressed that the kinds of events we could be interested in are application-dependent. In other words, it depends on the type of knowledge we are looking for. For instance, we could be interested in finding infrequent rules for intrusion detection in computer systems, exceptions to classical associations for the detection of conflicting medicine therapies, or unusual short sequences of nucleotides in genome sequencing.

Our objective in this paper is to introduce a new kind of rule describing a type of knowledge we might me interested in, what we will call anomalous association rules henceforth. Anomalous association rules are confident rules representing homogeneous deviations from common behavior. This common behavior can be modeled by standard association rules and, therefore, it can be said that anomalous association rules are hidden by a dominant association rule.

2. Motivation and related work

Several proposals have appeared in the data mining literature that try to reduce the number of associations obtained in a mining process, just to make them manageable

by an expert. According to the terminology used in [6], we can distinguish between user-driven and data-driven approaches, also referred to as subjective and objective interestingness measures, respectively [21].

Let us remark that, once we have obtained the set of *good rules* (considered as such by any interestingness measure), we can apply filtering techniques such as eliminating redundant tuples [19] or evaluating the rules according to other interestingness measures in order to check (at least, in some extent) their degree of surprisingness, i.e, if the rules convey new and useful information which could be viewed as unexpected [8, 9, 21, 6]. Some proposals [13, 25] even introduce alternative interestingness measures which are strongly related to the kind of knowledge they try to extract.

In user-driven approaches, an expert must intervene in some way: by stating some restriction about the potential attributes which may appear in a relation [22], by imposing a hierarchical taxonomy [10], by indicating potential useful rules according to some prior knowledge [15], or just by eliminating non-interesting rules in a first step so that other rules can automatically be removed in subsequent steps [18].

On the other hand, data-driven approaches do not require the intervention of a human expert. They try to autonomously obtain more restrictive rules. This is mainly accomplished by two approaches:

a) Using interestingness measures differing from the usual support-confidence pair [14, 26].

b) Looking for other kinds of knowledge which are not even considered by classical association rule mining algorithms.

The latter approach pursues the objective of finding surprising rules in the sense that an informative rule has not necessary to be a frequent one. The work we present here is in line with this second data-driven approach. We shall introduce a new kind of association rules that we will call *anomalous rules*.

Before we briefly review existing proposals in order to put our approach in context, we will describe the notation we will use henceforth. From now on, X, Y, Z, and A shall denote arbitrary itemsets. The support and confidence of an association rule $X \Rightarrow Y$ are defined as usual and they will be represented by supp$(X \Rightarrow Y)$ and conf$(X \Rightarrow Y)$, respectively. The usual minimum support and confidence thresholds are denoted by $MinSupp$ and $MinConf$, respectively. A frequent rule is a rule with high support (greater than or equal to the support threshold $MinSupp$), while a confident rule is a rule with high confidence (greater than or equal to the confidence threshold $MinConf$). A *strong rule* is a classical association rule, i.e, a frequent and confident one.

[7, 20] try to find non-frequent but highly correlated itemsets, whereas [28] aims to obtain *peculiarities* defined as non-frequent but highly confident rules according to a nearness measure defined over each attribute, i.e, a peculiarity must be significantly *far* away from the rest of individuals. [27] finds *unusual sequences*, in the sense that items with low probability of occurrence are not expected to be together in several sequences. If so, a surprising sequence has been found.

Another interesting approach [13, 25, 3] consists of looking for *exceptions*, in the sense that the presence of an attribute interacting with another may change the consequent in a strong association rule. The general form of an exception rule is introduced in [13, 25] as follows:

$$\begin{gathered} X \Rightarrow Y \\ XZ \Rightarrow \neg Y \\ X \not\Rightarrow Z \end{gathered}$$

Here, $X \Rightarrow Y$ is a *common sense* rule (a strong rule). $XZ \Rightarrow \neg Y$ is the *exception*, where $\neg Y$ could be a concrete value E (the $\underline{E}$xception [25]). Finally, $X \not\Rightarrow Z$ is a *reference* rule. It should be noted that we have simplified the definition of exceptions since the authors use five [13] or more [25] parameters which have to be settled beforehand, which could be viewed as a shortcoming of their discovery techniques.

In general terms, the kind of knowledge these exceptions try to capture can be interpreted as follows:

X strongly implies Y (and not Z).
But, in conjunction with Z, X does not imply Y
(maybe it implies another E)

For example [24], if X represents `antibiotics`, Y `recovery`, Z `staphylococci`, and E `death`, then the following rule might be discovered: with the help of `antibiotics`, the patient usually tends to `recover`, unless `staphylococci` appear; in such a case, `antibiotics` combined with `staphylococci` may lead to `death`.

These exception rules indicate that there is some kind of interaction between two factors, X and Z, so that the presence of Z alters the usual behavior (Y) the population have when X is present.

This is a very interesting kind of knowledge which cannot be detected by traditional association rules because the exceptions are hidden by a dominant rule. However, there are other exceptional associations which cannot be detected by applying the approach described above. For instance, in scientific experimentation, it is usual to have two groups of individuals: one of them is given a placebo and the other one is treated with some real medicine. The scientist wants to discover if there are significant differences in both populations, perhaps with respect to a variable $\mathbb{Y}$. In those cases,

where the change is significant, an ANOVA or contingency analysis is enough. Unfortunately, this is not always the case. What the scientist obtains is that both populations exhibit a similar behavior except in some rare cases. These infrequent events are the interesting ones for the scientist because they indicate that something happened to those individuals and the study must continue in order to determine the possible causes of this unusual change of behavior.

In the ideal case, the scientist has recorded the values of a set of variables $\mathbb{Z}$ for both populations and, by performing an exception rule analysis, he could conclude that the interaction between two itemsets X and Z (where Z is the itemset corresponding to the values of $\mathbb{Z}$) change the common behavior when X is present (and Z is not). However, the scientist does not always keep records of all the relevant variables for the experiment. He might not even be aware of which variables are really relevant. Therefore, in general, we cannot not derive any conclusion about the potential changes the medicine causes. In this case, the use of an alternative discovery mechanism is necessary. In the next section, we present such an alternative which might help our scientist to discover behavioral changes caused by the medicine he is testing.

3. Defining anomalous association rules

An anomalous association rule is an association rule that comes to the surface when we eliminate the dominant effect produced by a strong rule. In other words, it is an association rule that is verified when a common rule fails.

In this paper, we will assume that rules are derived from itemsets containing discrete values.

Formally, we can give the following definition to anomalous association rules:

Definition 1 *Let X, Y, and A be arbitrary itemsets. We say that $X \rightsquigarrow A$ is an anomalous rule with respect to $X \Rightarrow Y$, where A denotes the Anomaly, if the following conditions hold:*

a) $X \Rightarrow Y$ is a strong rule (frequent and confident)

b) $X\neg Y \Rightarrow A$ is a confident rule

c) $XY \Rightarrow \neg A$ is a confident rule

In order to emphasize the involved consequents, we will also used the notation $X \rightsquigarrow A|\neg Y$, which can be read as: "X is associated with A when Y is not present"

It should be noted that, implicitly in the definition, we have used the common minimum support ($MinSupp$) and confidence ($MinConf$) thresholds, since they tell us which rules are frequent and confident, respectively. For the sake of simplicity, we have not explicitly mentioned them in the definition. A minimum support threshold is relevant to condition *a)*, while the same minimum confidence threshold is used in conditions *a)*, *b)*, and *c)*.

The semantics this kind of rules tries to capture is the following:

X strongly implies Y,
but in those cases where we do not obtain Y,
then X confidently implies A

In other words:

When X, then
we have either Y (usually) or A (unusually)

Therefore, anomalous association rules represent homogeneous deviations from the usual behavior. For instance, we could be interested in situations where a common rule holds:

```
if symptoms-X then disease-Y
```

Where the rule does not hold, we might discover an interesting anomaly:

```
if symptoms-X then disease-A
              when not disease-Y
```

If we compare our definition with Hussain and Suzuki's [13, 25], we can see that they correspond to different semantics. Attending to our formal definition, our approximation does not require the existence of the *conflictive* itemset (what we called Z when describing Hussain and Suzuki's approach in the previous section). Furthermore, we impose that the majority of exceptions must correspond to the same consequent A in order to be considered an anomaly.

In order to illustrate these differences, let us consider the relation shown in Figure 1, where we have selected those records containing X. From this dataset, we obtain $\text{conf}(X \Rightarrow Y) = 0.6$, $\text{conf}(XZ \Rightarrow \neg Y) = \text{conf}(XZ \Rightarrow A) = 1$, and $\text{conf}(X \Rightarrow Z) = 0.2$. If we suppose that the itemset XY satisfies the support threshold and we use 0.6 as confidence threshold, then "$XZ \Rightarrow A$ is an exception to $X \Rightarrow Y$, with reference rule $X \Rightarrow \neg Z$". This exception is not highlighted as an anomaly using our approach because A is not always present when $X\neg Y$. In fact, $\text{conf}(X\neg Y \Rightarrow A)$ is only 0.5, which is below the minimum confidence threshold 0.6. On the other hand, let us consider the relation in Figure 2, which shows two examples where an anomaly is not an exception. In the second example, we find that $\text{conf}(X \Rightarrow Y) = 0.8$, $\text{conf}(XY \Rightarrow \neg A) = 0.75$, and $\text{conf}(X\neg Y \Rightarrow A) = 1$. No Z-value exists to originate an exception, but $X \rightsquigarrow A|\neg Y$ is clearly an anomaly.

The table in Figure 1 also shows that when the number of variables (attributes in a relational database) is high, then the chance of finding spurious Z itemsets correlated with

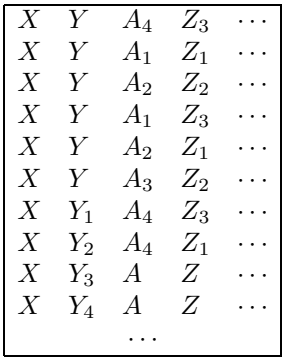

Figure 1. A is an exception to $X \Rightarrow Y$ when Z, but that anomaly is not confident enough to be considered an anomalous rule.

$\neg Y$ notably increases. As a consequence, the number of rules obtained can be really high (see [25, 23] for empirical results). The semantics we have attributed to our anomalies is more restrictive than exceptions and, thus, when the expert is interested in this kind of knowledge, then he will obtain a more manageable number of rules to explore. Moreover, we do not require the existence of a Z explaining the exception.

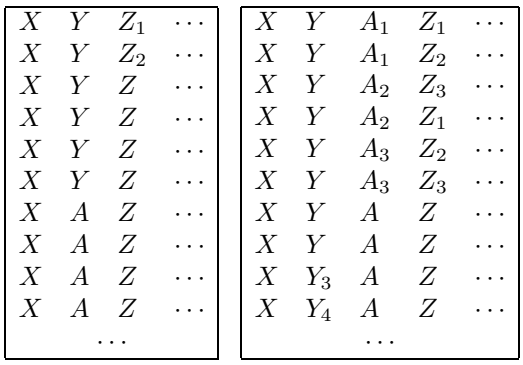

Figure 2. $X \rightsquigarrow A|\neg Y$ is detected as an anomalous rule, even when no exception can be found through the Z-values.

In particular, we have observed that users are usually interested in anomalies involving one item in their consequent. A more rational explanation of this fact might have psychological roots: As humans, we tend to find more problems when reasoning about negated facts. Since the anomaly introduces a negation in the rule antecedent, experts tend to look for 'simple' understandable anomalies in order to detect unexpected facts. For instance, an expert physician might directly look for the anomalies related to common symptoms when these symptoms are not caused by the most probable cause (that is, the usual disease she would diagnose). The following section explores the implementation details associated to the discovery of such kind of anomalous association rules.

4. Discovering anomalous association rules

Given a database, mining conventional association rules consists of generating all the association rules whose support and confidence are greater than some user-specified minimum thresholds. We will use the traditional decomposition of the association rule mining process to obtain all the anomalous association rules existing in the database:

- Finding all the relevant itemsets.
- Generating the association rules derived from the previously-obtained itemsets.

The first subtask is the most time-consuming part and many efficient algorithms have been devised to solve it in the case of conventional association rules. For instance, Apriori-based algorithms are iterative [16]. Each iteration consists of two phases. The first phase, candidate generation, generates potentially frequent k-itemsets (C_k) from the previously obtained frequent (k-1)-itemsets (L_{k-1}). The second phase, support counting, scans the database to find the actual frequent k-itemsets (L_k). Apriori-based algorithms are based on the fact that that all subsets of a frequent itemset are also frequent. This allows for the generation of a reduced set of candidate itemsets. Nevertheless, it should be noted that the there is no actual need to build a candidate set of potentially frequent itemsets [11].

In the case of anomalous association rules, when we say that $X \rightsquigarrow A|\neg Y$ is an anomalous rule, that means that the itemset $X \cup \neg Y \cup A$ appears often when the rule $X \Rightarrow Y$ does not hold. Since it represents an anomaly, by definition, we cannot establish a minimum support threshold for $X \cup \neg Y \cup A$, in the same sense than a strong rule. In fact, an anomaly is not usually very frequent in the whole database. Therefore, standard association rule mining algorithms, exploiting the classical *Apriori* support pruning, cannot be used to detect anomalies without modification.

Given an anomalous association rule $X \rightsquigarrow A|\neg Y$, let us denote by R the subset of the database that, containing X, does not verify the association rule $X \Rightarrow Y$. In other words, R will be the part of the database that does not verify the rule and might host an anomaly. The anomalous association rule confidence will be, therefore, given by the following expression:

$$conf_R(X \rightsquigarrow A|\neg Y) = \frac{supp_R(X \cup A)}{supp_R(X)}$$

When we write $supp_R(X)$, it actually represents $supp(X \cup \neg Y)$ in the complete database. Although this value is not usually computed when obtaining the itemsets, it can be easily computed as $supp(X) - supp(X \cup Y)$. Both values in this expression are always available after the conventional association rule mining process, since both X and $X \cup Y$ are frequent itemsets.

Applying the same reasoning, the following expression can be derived to represent the confidence of the anomaly $X \rightsquigarrow A|\neg Y$:

$$conf_R(X \rightsquigarrow A|\neg Y) = \frac{supp(X \cup A) - supp(X \cup Y \cup A)}{supp(X) - supp(X \cup Y)}$$

Fortunately, when somebody is looking for anomalies, he is usually interested in anomalies involving individual items. We can exploit this fact by taking into account that, even when $X \cup A$ and $X \cup Y \cup A$ might not be frequent, they are extensions of the frequent itemsets X and $X \cup Y$, respectively.

Since A will represent individual items, our problem reduces to being able to compute the support of $L \cup i$, for each frequent itemset L and item i potentially involved in an anomaly.

Therefore, we can modify existing iterative association rule mining algorithms to efficiently obtain all the anomalies in the database by modifying the support counting phase to compute the support for frequent itemset extensions:

- Candidate generation: As in any Apriori-based algorithm, we generate potentially frequent k-itemsets from the frequent itemsets of size $k-1$.
- Database scan: The database is read to collect the information needed to compute the rule confidence for potential anomalies. This phase involves two parallel tasks:
  - Candidate support counting: The frequency of each candidate k-itemset is obtained by scanning the database in order to obtain the actual frequent k-itemsets.
  - Extension support counting: At the same time that candidate support is computed, the frequency of each frequent $k-1$-itemset extension can also be obtained.

Once we obtain the last set of frequent itemsets, an additional database scan can be used to compute the support for the extensions of the larger frequent itemsets.

Using a variation of an standard association rule mining algorithm as TBAR [4], nicknamed ATBAR (Anomaly TBAR), we can efficiently compute the support for each frequent itemset as well as the support for its extensions.

In order to discover existing anomalies, a tree data structure is built to store all the support values needed to check potential anomalies. This tree is an extended version of the typical itemset tree used by algorithms like TBAR [4]. The extended itemset tree stores the support for frequent itemset extensions as well as for all the frequent itemsets themselves. Once we have these values, all anomalous association rules can be obtained by the proper traversal of this tree-shaped data structure.

5. Pruning and summarizing rules

Deriving anomalous association rules without imposing some constraints is meaningless. We introduce some general criteria which can be divided into two groups: *a priori* and *a posteriori*.

A priori pruning criteria. (Restrictions imposed before proceeding to the construction of the itemset tree)

- Do not allow an attribute with only two different values to appear in the anomalous consequent part of the rule. In general, attributes appearing in the anomalous consequents, should have at least three or four distinct values.
- Null values should not appear in the anomalous consequent part of a rule, but they could appear in the strong part. A strong rule with a null consequent but a non-null anomalous consequent could provide useful information to the user.

A posteriori pruning criteria. (Criteria imposed once the set of anomalous rules is construted)

- Eliminate those rules sharing the same strong and anomalous consequent, and having more antecedents. In this case, the simplest rule is included and the others are pruned.

  If there exists an anomalous rule $X \rightsquigarrow A|\neg Y$, then every anomalous rule $XH \rightsquigarrow A|\neg Y$ is pruned.
- Do not allow anomalies supported by just one or two records. Thus, a support threshold for the anomaly should be considered. A minimum support of three records might be choosen.

  If supp$(XA) < 3$, then $X \rightsquigarrow A|\neg Y$ is pruned.

DataBase	Ant. Size	MinSupp	Confidence 90%				Confidence 75%			
			Anom. Prun.	Anom.	Assoc.	Reduct.	Anom. Prun.	Anom.	Assoc.	Reduct.
HEPATITIS	1	10%	4	61	131	97%	57	229	398	86%
		5%	4	63	137	97%	70	253	427	84%
		1%	4	63	238	98%	70	398	561	88%
	2	10%	11	901	1639	99%	222	3029	3820	94%
		5%	11	1806	3249	99%	310	7017	7352	96%
		1%	11	1806	12406	100%	310	13496	18836	98%
BREAST-CANCER	1	10%	0	0	9	100%	1	2	43	98%
		5%	0	2	12	100%	2	5	61	97%
		1%	0	2	24	100%	2	35	89	98%
	2	10%	3	11	62	95%	27	50	265	90%
		5%	3	55	146	98%	44	146	485	91%
		1%	3	85	736	100%	50	574	1423	96%
WISCONSIN-BREAST-CANCER	1	10%	1	2	29	97%	1	2	80	99%
		5%	1	13	43	98%	4	19	117	97%
		1%	1	47	70	99%	15	121	170	91%
	2	10%	7	63	183	96%	33	100	427	92%
		5%	16	163	313	95%	71	248	688	90%
		1%	19	600	936	98%	117	1634	1811	94%
POSTOPERATIVE	1	10%	0	0	14	100%	3	5	29	90%
		5%	0	0	14	100%	3	6	30	90%
		1%	0	0	43	100%	3	6	59	95%
	2	10%	0	11	87	100%	2	37	206	99%
		5%	0	11	123	100%	2	57	310	99%
		1%	0	11	586	100%	2	64	792	100%
CONTRACEPTIVE	1	10%	0	0	32	100%	3	3	76	96%
		5%	0	0	34	100%	3	3	84	96%
		1%	0	0	36	100%	3	3	87	97%
	2	10%	4	7	132	97%	9	34	253	96%
		5%	16	32	311	95%	49	131	612	92%
		1%	17	65	527	97%	106	314	1114	90%
PIMA DIABETES	1	10%	0	0	36	100%	0	0	49	100%
		5%	0	0	36	100%	0	0	49	100%
		1%	0	0	36	100%	0	0	49	100%
	2	10%	0	2	45	100%	4	12	54	93%
		5%	1	25	185	99%	17	124	232	93%
		1%	1	141	543	100%	77	691	834	91%

Table 1. Number of rules obtained after pruning

These pruning methods should be applied to eliminate spurious and trivial anomalous rules. The application of these simple criteria can dramatically decrease the number of outputs as Table 1 shows (see description of Table 1 in next section). Once the reduced set of rules is obtained, summarizing and ranking measures could also be applied. Such measures should be applied once the whole set of pruned rules are discovered. The particular measures used for a particular problem might depend on specific domain knowledge. Some criteria are:

Summarizing criteria help us to merge several rules into a single one.

For instance, we can merge several rules with the same pair of strong and anomalous consequents in the following way:

> All the anomalous rules $X_i \rightsquigarrow A|\neg Y$, could be merged into one single rule $(\vee_i X_i) \rightsquigarrow A|\neg Y$, where $\vee$ stands for the logical or.

This summarizing method is aimed at presenting a simple set of rules to the user. Obviously, the confidence and support values can not be merged and, therefore, the individual rules should still be stored in case the user wanted to analyze them.

Let us note that the greater the number of different X_i are merged, the more confident we are that the negative association between Y and A, is not related to those X_i. For instance, Y could stand for *Less than 18 years* and A for *Has the car licence*.

On the other hand, the first a posteriori pruning method we introduced before could be rewritten as a summarizing one, but following Occam's razor we prefer to consider the simplest rule, and thus eliminate (not summarize) unnecessarily complex rules.

Ranking measures give a numerical value to the interest of each rule. Some examples are:

- If an anomalous rule involves the same numerical attribute in the strong and in the anomalous consequent part, then a ranking measure could give more importance to those rules where such intervals are not closed, because such rule would detect very opposite behaviors.
- The more confident the rules $X\neg Y \Rightarrow A$ and $XY \Rightarrow \neg A$ are, the stronger the $X \rightsquigarrow A|\neg Y$ anomaly is. This fact could be useful in order to define a degree of strength associated to the anomaly.

6. Experimental results

Table 1 presents some results obtained with ATBAR using datasets from the UCI Machine Learning Repository (we focused our experimentation on medical datasets). As motivated in Section 3, we only consider associations with one consequent value. Numerical attributes are a priori clustered in 5 intervals by using a classical equi-depth partitioning algorithm. `Ant.Size` represents the number of antecedents. We restrict our experimentation to the case of one and two antecedents. `MinSupp` is the support threshold (as a percentage) for the strong rule. `Confidence` is the confidence of the strong rule (as well as the confidence of the anomaly), as stated in definition 1. `Anom` is the number of anomalous rules. `Anom.Prun.` is the number of pruned rules obtained by using the basic methods introduced in Section 5 with four distinct values in each attribute (we do not apply any ranking measure or summarizing criteria). `Assoc` is the number of association rules satisfying the support (row) and confidence (column) thresholds. `Reduct` is the reduction percentage of `Anom Pruned` with respect to `Assoc`. It is worth mentioning that this percentage is included only as a reference to the problem complexity, because anomalies and associations are not the same concept.

The need to obtain the support for frequent itemset extensions obviously incurs in some overhead, although it is reasonable even for large datasets. The overhead in time is about 20% in the experiments we have performed.

7. Conclusions and future work

In this paper, we have studied situations where standard association rules do not provide the information the user seeks. Anomalous association rules have proved helpful in order to represent the kind of knowledge the user might be looking for when analyzing deviations from normal behavior. The normal behavior is modeled by conventional association rules, and the anomalous association rules are association rules which hold when the conventional rules fail.

We have also developed an efficient algorithm to mine anomalies from databases. Our algorithm, ATBAR, is suitable for the discovery of anomalies in large databases. Our approach could prove useful in tasks such as fraud identification, intrusion detection systems and, in general, any application where the user is not really interested in the most common patterns, but in those patterns which differ from the norm.

We intend to apply our technique to huge datasets as well as to contrast the results with experts in order to evaluate the false positive rate and analyze summarizing criteria in depth, so more rules can be pruned.

Acknowledgments

Marco-Antonio Balderas was supported by a scholarship grant from Mexican National Research Council on Science and Technology and the "Universidad Autonoma de Tamaulipas" (CONACyT-PROMEP).
Eduardo Eisman was supported by a scholarship grant from the University of Granada in Spain.
During the preparation of this paper, Fernando Berzal was a visiting research scientist at the research group led by prof. Jiawei Han at the University of Illinois at Urbana-Champaign.

References

[1] R. Agrawal and J. Shafer. Parallel mining of association rules. *IEEE Transactions on Kowledge and Data Engineering*, 8(6):962–969, 1996.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In P. S. Yu and A. S. P. Chen, editors, *Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.

[3] Y. Aumann and Y. Lindellt. A statistical theory for quantitative association rules. *Journal of Intelligent Information Systems*, 20(3):255–283, 2003.

[4] F. Berzal, J. Cubero, J. Marin, and J. Serrano. An effi cient method for association rule mining in relational databases. *Data and Knowledge Engineering*, 37:47–84, 2001.

[5] F. Berzal, J. Cubero, D. Sanchez, and J. Serrano. Art: A hybrid classifi cation model. *Machine Learning*, 54(1):67–92, 2004.

[6] D. Carvalho, A. Freitas, and N. Ebecken. A critical review of rule surprisingness measures. In N. Ebecken, C. Brebbia, and A. Zanasi, editors, *Proc. Data Mining IV - Int. Conf. on Data Mining*, pages 545–556. WIT Press, December 2003.

[7] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. D. Ullman, and C. Yang. Finding interesting associations without support pruning. *IEEE Transactions on Knowledge and Data Engineering*, 13(1):64–78, 2001.

[8] A. Freitas. On Rule Interestingness Measures. *Knowledge-Based Systems*, 12(5-6):309–315, October 1999.

[9] A. A. Freitas. On objective measures of rule surprisingness. In *Principles of Data Mining and Knowledge Discovery*, pages 1–9, 1998.

[10] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *Proceedings of the VLDB Conference*, pages 420–431, 1995.

[11] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of Data*, pages 1–12, 2000.

[12] C. Hidber. Online association rule mining. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of Data*, pages 145–156, 1999.

[13] F. Hussain, H. Liu, E. Suzuki, and H. Lu. Exception rule mining with a relative interestingness measure. In *Pacifi c-Asia Conference on Knowledge Discovery and Data Mining*, pages 86–97, 2000.

[14] Y. Kodratoff. Comparing machine learning and knowledge discovery in DataBases: An application to knowledge discovery in texts. In *Machine Learning and its Applications*, volume 2049, pages 1–21. Lecture Notes in Computer Science, 2001.

[15] B. Liu, W. Hsu, S. Chen, and Y. Ma. Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems*, pages 47–55, 2000.

[16] R. S. R. Agrawal. Fast algorithms for mining association rules. In *Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile*, 1994.

[17] S. Ramaswamy, S. Mahajan, and A. Silberschatz. On the discovery of interesting patterns in association rules. In *The VLDB Journal*, pages 368–379, 1998.

[18] S. Sahar. Interestingness via what is not interesting. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 332–336, 1999.

[19] D. Shah, L. V. S. Lakshmanan, K. Ramamritham, and S. Sudarshan. Interestingness and pruning of mined patterns. In *1999 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 1999.

[20] J.-L. Sheng-Ma, Hellerstein. Mining mutually dependent patterns. In *Proceedings ICDM'01*, pages 409–416, 2001.

[21] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. On Knowledge And Data Engineering*, 8:970–974, 1996.

[22] R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. In D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining, KDD*, pages 67–73. AAAI Press, 14–17 1997.

[23] E. Suzuki. Scheduled discovery of exception rules. In *Discovery Science*, volume 1721, pages 184–195. Lecture Notes in Artifi cial Intelligence, 1999.

[24] E. Suzuki. In pursuit of interesting patterns with undirected discovery of exception rules. In *Progress Discovery Science*, volume 2281, pages 504–517. Lecture Notes in Artifi cial Intelligence, 2001.

[25] E. Suzuki. Undirected discovery of interesting exception rules. *International Journal of Pattern Recognition and Artifi cial Intelligence*, 16(8):1065–1086, 2002.

[26] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right objective measure for association analysis. *Information Systems*, 29:293–313, 2003.

[27] J. Yang, W. Wang, and P. Yu. Mining surprising periodic patterns. *Data Mining and Knowledge Discovery*, 9:1–28, 2004.

[28] N. Zhong, Y. Yao, and M. Ohshima. Peculiarity oriented multidatabase mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):952–960, 2003.

Learning to Live with False Alarms

Chris Drummond
Institute for Information Technology
National Research Council Canada
Ottawa, Ontario
Canada, K1A 0R6
Chris.Drummond@nrc-cnrc.gc.ca

Rob Holte
Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada, T6G 2E8
holte@cs.ualberta.ca

ABSTRACT

Anomalies are rare events. For anomaly detection, severe class imbalance is the norm. Although there has been much research into imbalanced classes, there are surprisingly few examples of dealing with severe imbalance. Alternative performance measures have superseded error rate, or accuracy, for algorithm comparison. But whatever their other merits, they tend to obscure the severe imbalance problem. We use the relative cost reduction of a classifier over a trivial classifier that chooses the less costly class. We show that for applications that are inherently noisy there is a limit to the cost reduction achievable. Even a Bayes optimal classifier has a vanishingly small reduction in costs as imbalance increases. If events are rare and not too costly, the unpalatable conclusion is that our learning algorithms can do little. If the events have a higher cost then a large number of false alarms must be tolerated, even if the end user finds that undesirable.

1. INTRODUCTION

An anomalous event is, by definition, unusual, but how unusual is an important question. At last year's workshop, Bay [2] equated anomalous to "extremely rare and unusual", Fawcett [7] stated that "positive activity is inherently rare". This is certainly true of one of the authors's experience applying data mining algorithms to the maintenance of complex equipment. With aircraft engines, for instance, component failure is fortunately far from common. In anomaly detection, we should expect an imbalance in excess of 10:1 and often 100:1 or 1000:1 or even larger.

One obvious source of ideas to help with anomaly detection is the community researching class imbalance, and the difficulties that result [8, 4]. Unfortunately, the sort of severe imbalance seen in anomaly detection is not commonplace in this research, an issue we return to later in this paper. On the occasions when imbalance has been severe, the measures used to verify success have obscured the problem. One original motivation for this area of research was that, when classes were imbalanced, many people observed that learning algorithms often produced classifiers that did little more than predict the most common class. It seemed intuitive that a practical classifier must do much better on the minority class, often the one of greater interest, even if this meant sacrificing performance on the majority class. This was our belief as well, earlier work by one of the authors stated [9] "A classifier that labels all regions as [the majority class] will achieve an accuracy of 96% .... a system achieving 94% on [the minority class] and 94% on [the majority class] will have worse accuracy yet be deemed highly successful".

Provost and Fawcett [13] introduced ROC curves to the data mining community, which seemed the solution to such concerns. ROC curves made clear the inherent trade-off between performance on the positive and negative examples. We could choose a point on this curve and make whatever trade-off we thought appropriate. If costs and class distribution were known, this point could be determined by using an iso-performance line, but this decision was best left to the end user of the classifier in the particular application. From a research prospective then we should focus on developing algorithms that produce better ROC curves. An attractive metric for comparing ROC curves that has become popular recently is area-under the curve (AUROC) [10]. This approach encourages the development of algorithms that are effective over a range of costs and class distributions.

For anomaly detection, however, we know that the class distribution is severely imbalanced, we also know the direction of imbalance. We are not interested in performance of the whole curve only its lower left hand corner. Using partial AUROC [12] or DET curves [11] would at least concentrate on the important region. But we have found it difficult to determine the actual performance gains achieved by one classifier over another using ROC curves and these variants are unlikely to help. We introduced an alternative representation called cost curves [6] which makes performance gains explicit.

In the rest of the paper, we show that even a Bayes optimal classifier does only marginally better than a trivial classifier with severe imbalance. Real classifiers will do worse than Bayes optimal and often even worse than the trivial classifier. If events are rare and not too costly, our learning algorithms can do little. If the events have a higher cost then it is better to have a large number of false alarms, even if the end user finds that undesirable, rather than miss an occurrence. We then continue by defending this viewpoint against various arguments we think might be forthcoming.

2. SEVERE IMBALANCE

To be useful, a classifier must appreciably outperform a trivial solution, such as choosing the majority class. Many people have observed that for extreme imbalances the majority classifier's error rate is so small that it seems little can be done to improve on it. Even classifiers with good performance when classes are balanced fare badly for severe imbalance [1]. Here, we make the stronger claim that a "relative reduction" in the majority classifier's error rate is often unachievable. We focus on "relative reduction" because we think it important to consider what success means when a trivial classifier gets only say 1% wrong. Error rate reduction is the fraction of the majority classifier's error rate that the new classifier removes. The classifier could, in principle, achieve a value of one, removing all existing error. If the majority classifier's error rate is 1%, a classifier with a 0.4% error rate would have an error rate reduction of 0.6, still a respectable value. This would be equivalent to achieving a 20% error rate when the classes are balanced and the majority classifier has an error rate of 50%. This idea seems even more intuitive when considering misclassification costs. The success of a classifier is how much it reduces the costs that occur when using a trivial classifier. We will use the phrase "relative cost reduction" to indicate this and a decrease in error rate if misclassification costs are not used.

Figure 1 shows cost curves for the Bayes optimal classifier for two univariate normal distributions, one representing the positive class, the other the negative. Drummond and Holte [6] discuss cost curves in detail, here we give a very brief sketch hopefully sufficient for the reader to understand the argument. The bold continuous curves are cost curves for 3 different values of distance between the means of the two normal distributions. The curves give the error rate (the y-axis, ignore the axes' labels in parentheses for the moment) for each possible prior probability of an instance belonging to the positive class (the x-axis). The dashed triangle is the majority classifier. It has an error rate of zero when the instances are all positive or all negative, $x = 0$ or $x = 1$, and an error rate of 0.5 when there are an equal number of positives and negatives, $x = 0.5$.

We can include costs simply by relabeling the axes, as shown by the text in parentheses. The curves are unchanged, but now give the expected cost, normalized between zero and one, (the y-axis) and the probability times the cost, normalized between zero and one, (the x-axis). There is still a triangular trivial classifier, but it now represents the classifier that labels instances according to which class produces the smaller expected cost (for simplicity we will still call it the majority classifier).

The distances between the means of the normal distributions were chosen to make the relative cost reduction when the classes are balanced 0.2, 0.5 and 0.8 (from top to bottom). The series of progressively smaller triangles in Figure 1, made of dotted lines, we call cost reduction contours. Each cost reduction contour indicates a specific fraction of the cost of using the majority classifier. The continuous curves cross multiple contours indicating a decreasing relative cost reduction as imbalance increases.

If we focus on the lower left hand corner of Figure 1, where the negative instances are much more common than the positives, or more costly to misclassify. The upper two curves have become nearly indistinguishable from the majority classifier for ratios about 20:1. The lowest cost curve

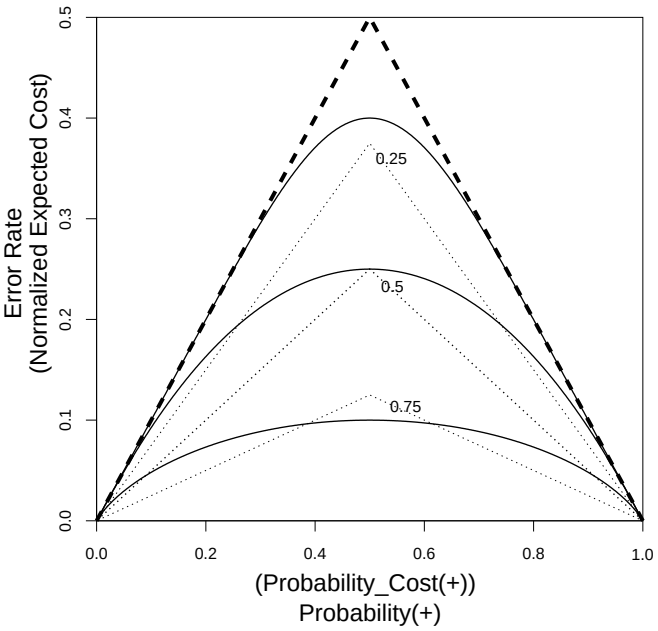

Figure 1: Different Distances

has crossed the 0.5 cost reduction contour at an imbalance of about 10:1 and crossed the 0.25 cost reduction contour at about 50:1. So even a Bayes optimal classifier with good performance, say a normalized expected cost of 0.1 with no imbalance, fares a lot worse when imbalance is severe. With imbalances as low as 10:1, and certainly for imbalances of 100:1 and greater, the performance gain over the majority classifier is minimal.

Figure 2 shows examples using non-normal distributions. The problem is made worse when distributions have heavier tails than the normal, the top two curves. With lighter tails the problem is reduced. But only in the case of two overlapping uniform distributions, the lower continuous triangle, is the relative cost reduction, when balanced, maintained for all degrees of imbalance. These results are for Bayes optimal classifiers. For practical algorithms any gain will be reduced and possibly disappear altogether.

Introducing misclassification costs will improve the situation, but they should not simply be used as a device to correct class imbalance. They must exist in the application. In some situations, such as safety critical operations, missing a true alarm may have major consequences. Adding a large misclassification cost to represent this would, at least somewhat, offset the severe imbalance. But the inclusion of such a cost inevitably produces a high rate of false alarms which users often find unacceptable.

3. ARGUMENTS AGAINST THE CONCLUSIONS

In this section, we try to anticipate the arguments that might be raised against the conclusions we have drawn in this paper.

A small performance gain is worth having. In some situations a small performance gain is the difference between success and failure. But we believe this is by no means the norm. One might argue that if a company's costs are

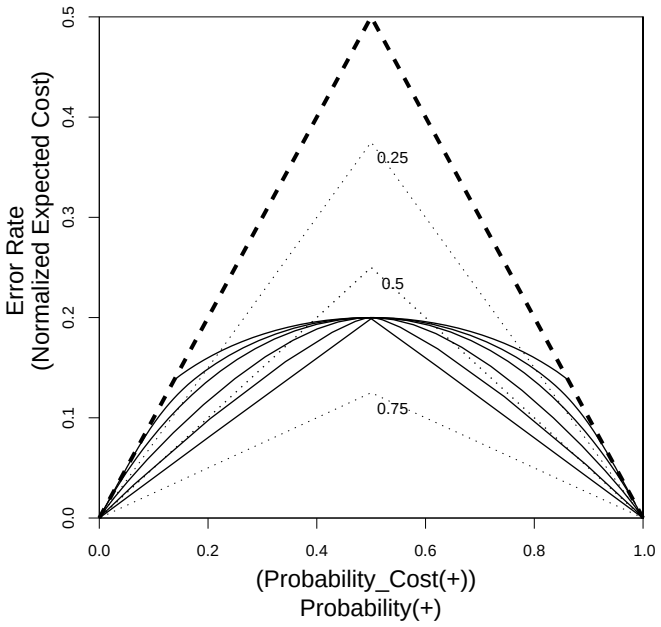

Figure 2: Different Distributions

very large even a small percentage represents a large sum of money and therefore well worth saving. Our response is that effort spent on the cost reduction must equate to the savings and this must be viewed in terms of a percentage of total cost to have any meaning to the company.

Some performance measures don't have this problem. Costs are a very general way of measuring performance. So if alternative measures don't exhibit this problem one might ask why not. We have, however, assumed that costs are linear (3 errors costs 3 times as much as 1 error). In information retrieval, where precision-recall is the preferred measure, often one is only interested in retrieving a small sample with high precision. This sample may contain only a very small percentage of the total number of documents on a particular topic. This is an example of highly non-linear costs, which we have not addressed in this paper. For anomaly detection, it is unlikely to be of much value if only a very small percentage of anomalies are found, so the simple linear model is relevant.

An extremely imbalanced application was a success. One often cited paper, from high energy physics [5], had an imbalance of 1000,000:1. If one can cope with such an extreme imbalance, more modest imbalances such 10,000:1 should be easy. But in this application, as in the above paragraph, precision for a small number of positives was all that was required, the vast majority of positives were ignored. In many other examples in the literature imbalance was not severe, less than 10:1. Of the few examples of severe imbalance, tables of true positives and false alarms, or ROC curves, were typically used to compare algorithms. These did not address any possible performance advantage the majority classifier.

Real data sets don't suffer from this problem. Our argument would be weakened if real data sets typically had very low noise. We can only speculate on how much noise is intrinsic. Figure 3 shows cost curves for C4.5 (with the defaults settings) applied to three UCI data sets [3]. All three curves cross the lines for the majority classifiers for some degree of imbalance. For the hepatitis data, the topmost curve, this occurs when the positive class has a probability of about 0.2 very close to the actual class frequency in the data set. The middle curve for glass2 fares little better. Its expected cost when everything is balanced is lower, about 0.2. But at quite moderate imbalances of less than 10:1, it is also worse than the majority classifier. The lowest curve for the vote data fares the best, with better than 0.05 normalized expected cost when balanced. But even in this case with imbalances greater than 100:1 the majority classifier is better. Some of this might, of course, be due to algorithmic deficiencies but we suggest that some is due to noise inherent to the problem.

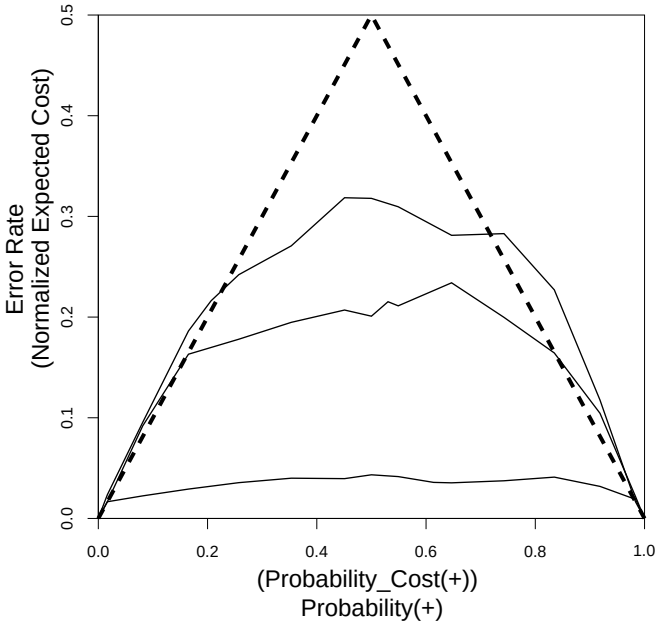

Figure 3: Three UCI Data Sets

Improving the algorithm will eliminate noise. Our analysis used a Bayes optimal classifier, real algorithms will fare worse. But better algorithms would be effective if the problem we have with existing algorithms are due to representational or search issues rather than inherent noise. Then a Bayes optimal classifier might achieve almost perfect classification, allowing much room for algorithmic improvement. But for this problem to disappear, extremely large regions of instance space without any noise are needed. Whether or not this is likely in practice we leave this to the intuitions of the reader.

4. CONCLUSIONS

The point of this paper is to raise awareness of the difficulty of dealing with rare events. If events are rare and not too costly, the unfortunate conclusion is that our learning algorithms can do little. We should just wait for the event to occur. If the events have a much higher cost then a large number of false alarms should be tolerated. If the end user is unhappy with the number of false alarms the only real answer may be to demonstrate that cost calculations show that capturing a real event is worth any costs assciated with false alarms.

5. ACKNOWLEDGMENTS

This work was supported by the National Research Council of Canada and the Alberta Ingenuity Fund through the Alberta Ingenuity Centre for Machine Learning.

6. REFERENCES

[1] S. Axelsson. The base-rate fallacy and its implications for the difficulty of intrusion detection. In *Proceedings of 6th ACM Conference on Computer and Communications Security*, pages 1–7, 1999.

[2] S. Bay. A framework for discovering anomalous regimes in multivariate time-series data with local models. http://cll.stanford.edu/symposia/anomaly/abstracts.html, 2004.

[3] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, University of California, Irvine, CA . www.ics.uci.edu/∼mlearn/MLRepository.html, 1998.

[4] N. V. Chawla, N. Japkowicz, and A. Kolcz, editors. *Proceedings of ICML'2003 Workshop on Learning from Imbalanced Data Sets*, 2003.

[5] S. H. Clearwater and E. G. Stern. A rule-learning program in high energy physics event classification. *Computational Physics Communications*, 67:159–182, 1991.

[6] C. Drummond and R. C. Holte. Explicitly representing expected cost: An alternative to ROC representation. In *Proceedings of 6th International Conference on Knowledge Discovery and Data Mining*, pages 198–207, New York, 2000. ACM.

[7] T. Fawcett. Activity monitoring: Anomaly detection as on-line classification. http://cll.stanford.edu/symposia/anomaly/abstracts.html, 2004.

[8] N. Japkowicz, editor. *Proceedings of AAAI'2000 Workshop on Learning from Imbalanced Data Sets*, 2000. AAAI Tech Report WS-00-05.

[9] M. Kubat, R. C. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30:195–215, 1998.

[10] C. X. Ling, J. Huang, and H. Zhang. AUC: a statistically consistent and more discriminating measure than accuracy. In *Proceedings of 18th International Joint Conference on Artificial Intelligence*, pages 519–524, 2003.

[11] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The det curve in assessment of detection task performance. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 1895–1898, 1997.

[12] S. H. Park, J. M. Goo, and C.-H. Jo. Receiver operating characteristic (roc) curve: Practical review for radiologists. *Korean Journal of Radiology*, 5(1), 2004.

[13] F. Provost and T. Fawcett. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 43–48, 1997.

Multivariate dependence among extremes, abrupt change and anomalies in space and time for climate applications

Auroop R. Ganguly
Oak Ridge National Laboratory

Tailen Hsing
Ohio State University

Rick Katz
Nat'l Ctr. for Atmospheric Research

David J. Erickson III
Oak Ridge National Laboratory

George Ostrouchov
Oak Ridge National University

Thomas J. Wilbanks
Oak Ridge National Laboratory

Noel Cressie
Ohio State University

ABSTRACT

This paper discusses multivariate spatio-temporal dependence between extremes or abrupt change and unusual values or anomalies in the context of climate dynamics and climate change. In climate, as in many other applications, anomalies (or extremes) in one variable like sea surface temperature may be a precursor for extremes (or abrupt change) in another variable like regional precipitation. In addition, this multivariate dependence may be spatially or temporally lagged, owing to climate "teleconnections". However, the anomalies may not be easily detectable and their dependence with extremes and rapid change may be difficult to quantify. This paper provides a brief review of the literature, which is followed by a description of critical gaps, both in the data or computational sciences as well as in the climate sciences. The quantification and visualization of multivariate dependence among extreme values and anomalies in highly nonlinear or stochastic systems is an emerging research area in theoretical statistics, with limited development in application areas and/or for massive or disparate space-time data. Further development is needed in these areas for multiple domains ranging from climate sciences and geography to sensor networks and national security.

Categories and Subject Descriptors

I.5.4 [**Pattern Detection**]: Applications; G.3 [**Probability & Statistics**]: Multivariate statistics; Statistical computing; J.2 [**Physical Sciences & Engineering**]: Earth/atmospheric sciences.

General Terms

Algorithms.

Keywords

Extremes, Anomalies, Dependence, Spatio-temporal, Climate.

1. INTRODUCTION

Multivariate dependence among extremes and anomalies in climate variables is important for a number of reasons. Climate anomalies like unusual sea surface temperatures in one part of the globe may cause imbalances in the atmospheric system leading to phenomena like the El Nino Southern Oscillation, which in turn may cause extremes in precipitation at regional scales. One consequence of global warming may be a preponderance of regional climate extremes like heat waves or intense storms. The inherently nonlinear climate system may be triggered to a different behavioral mode or equilibrium state due to sudden or gradual disruptions induced by human activities. There are several aspects to this problem. Extremes may be defined in terms of static or dynamic thresholds (e.g., exceedence over fixed threshold or thresholds that depend on location and time) and/or may be context-specific (e.g., temperature for human comfort levels, precipitation for optimal agricultural productivity). Large anomalies like the El Nino phenomena may be relatively easily detectable from data, while other anomalies may be more elusive. In certain situations, the anomalies may not be directly detectable from data and may have to be defined ex-post or from historical data analysis, for example, as a set of conditions among multiple variables that lead to extremes in another variable of interest. Thus, the problem of multivariate dependence among extremes, abrupt change, anomalies and unusual values is arguably more important in climate than the detection of anomalies from data.

2. PROBLEM STATEMENT

The ability to understand and eventually predict climate extremes and abrupt change is critical for science [1-2, 12-13, 16-17] and policy [17-20], and can help answer questions like the following: (a) Are heat waves or precipitation extremes likely to grow more intense in the next century? (b) Is an increase in Atlantic hurricane activity caused by warming of the earth at global or regional scales? (c) Is Sahel Africa likely to experience more extensive droughts in the next few decades? (d) Does an observed strong anomaly in sea surface temperatures in the Eastern Pacific imply extreme rainfall in South America within a few months? (e) Can abrupt change in historical (paleo-) climates be related to leading indicator variables, and can these be used to assign likelihoods of future change? (f) What are the likelihoods, risks and impacts of global or regional scale abrupt change or extremes in climate, and how can the adverse impacts be mitigated?

3. STATE OF THE ART

The current generation of general circulation models (GCM) yields precise, but not necessarily accurate, simulations of future climate scenarios. The Oak Ridge National Laboratory (ORNL) has vast quantities of climate simulations based on "IPCC runs" (*IPCC: Inter-governmental Panel on Climate Change*): the latest runs are global 3-hourly outputs for the atmosphere available from 2000 to 2100 at roughly 100 km spatial resolutions. Government agencies like NASA and NOAA have significant amount of observed data based on remote or in-situ sensors of climate variables. Our understanding of climate extremes and abrupt change can dramatically improve if the vast quantities of observed and simulated data can be mined using focused methodologies. The climate system involves potentially nonlinear dependence among multiple variables dimensioned by space and time, and exhibits strong teleconnections, or geographically dispersed dependence. Thus, methodologies for multivariate and potentially nonlinear dependence in space and time, geared towards extreme values, abrupt change or anomalous behavior, are key requirements. However, the literature in climate extremes [16; 1-2, 17] rely on simplistic statistics. Time series anomaly detection methods have been developed in statistics [5] and nonlinear dynamics [8], and applied to environmental problems [11-14]. However, methods for multivariate extremal and nonlinear dependence, especially in space and time [4, 9], are not well developed. Extreme value theory in statistics [3], and its environmental applications [6, 12-13], are well-established. However, the literature on multivariate dependence among extreme values and their visualization [15, 7, 10] is beginning to emerge. Novel methods are needed for massive, space-time data.

4. NEW DIRECTIONS

The focus of our ongoing research at ORNL, with collaborators at NCAR and OSU, is to develop new approaches for multivariate dependence among extremes, abrupt change and anomalies for potentially large data sets that are dimensioned by space and time, and then implementing these new approaches in the context of regional and global climate change and climate teleconnections. We are further developing and implementing recent advances in statistical theory of extreme values, including specialized probabilistic models for temporal and spatial extremal patterns and lagged dependence, as well as approaches to relate extremes in the dependent variable with temporally or spatially lagged anomalies or extremes in the independent variables. In addition, we are developing new theories and measures for multivariate dependence among extreme values and anomalies. The theories developed for extremes are applicable to abrupt change, following the differencing operation. The extensions and new formulations are being designed for easy visualization and quantification of the multivariate dependence among extremes, abrupt changes and anomalies, as well as for applications to massive data. The approaches are expected to yield the uncertainty associated with the anticipated extremes or unusual events at multiple scales, and relate these uncertainties to risks and economic/societal impacts.

ACKNOWLEDGMENTS

Research sponsored by the Laboratory Directed Research and Development Program (SEED funds) of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U. S. DOE under Contract No. DE-AC05-00OR22725. Workshop participation sponsored by ORNL's SensorNet® R&D program and Knowledge Discovery initiative.

REFERENCES

[1] American Institute of Physics, *Rapid climate change*, http://www.aip.org/history/climate/rapid.htm, 2004.

[2] Alley, R.B. et al, Abrupt climate change, *Science*, 299, 2005-10, 2003.

[3] Coles S., *An Introduction to Statistical Modeling of Extreme Values*, Springer-Verlag, 2001.

[4] Cressie, N., *Statistics for Spatial Data*, Rev. Edition, Wiley, 1993.

[5] Downing, D.J., V.V. Fedorov, W.F. Lawkins, M. D. Morris, and G. Ostrouchov, Large data series: Modeling the usual to identify the unusual, *Computational Statistics & Data Analysis*, 32, 245, 2000.

[6] Erickson, D.J. III, and J.A. Taylor, Non-Weibull behavior observed in a model-generated global surface wind speed frequency distribution, *Journal of Geophysical Research*, 94, 12, 693-8, 1989.

[7] Gilleland, E., R. Katz, and G. Young, *Extremes toolkit (extRemes): weather and climate applications of extreme value statistics*, National Center for Atmospheric Research, 2004.

[8] Heffernan, J.E., and J.A. Tawn, A conditional approach for multivariate extreme values, *Journal of the Royal Statistical Society*, Serial B, 66(3): 497-546, 2004.

[9] Hively, L.M., P.C. Gailey, and V. Protopopescu, Detecting Dynamical Change in Nonlinear Time Series, *Physics Letters A*, 258, 103, 1999.

[10] Hoffman, F.M., W.W. Hargrove, D.J. Erickson, III, and R. Oglesby, Using clustered climate regimes to analyze and compare predictions from fully coupled general circulation models, to appear in *Earth Interactions*, 2005.

[11] Hsing, T., Kluppelberg, C., and Kuhn, G., Dependence estimation and visualization in multivariate extremes with applications to financial data, to appear in *Extremes*, 7, 99-121, 2005.

[12] Jin, Y.-H., A. Kawamura, K. Jinno, and R. Berndtsson, Nonlinear multivariable analysis of SOI and local precipitation and temperature, *Nonlinear Processes in Geophysics*, 12: 67-74, 2005.

[13] Katz, R.W., G.S. Brush, and M.B. Parlange, Statistics of extremes: Modeling ecological disturbances, *Ecology*, 86: 1124-1134, 2005.

[14] Katz, R.W., M.B. Parlange, and P. Naveau, Statistics of extremes in hydrology, *Advances in Water Resources*, 25: 1287-1304, 2002.

[15] Khan, S., A.R. Ganguly, S. Saigal, Detection and predictive modeling of chaos in finite hydrological time series, *Nonlinear Processes in Geophysics*, 12(1): 41-53, 2005.

[16] Ledford, A.W., and J.A. Tawn, Diagnostics for dependence within time series extremes, *Journal of the Royal Statistical Society*, Serial B, 65(2): 521-543, 2003.

[17] Meehl, G.A., and C. Tebaldi, More intense, more frequent, & longer lasting heat waves in the 21st century, *Science*, 305: 994-7, 2004.

[18] NRC, *Abrupt Climate Change: Inevitable Surprises*, Committee on Abrupt Climate Change, National Research Council, Washington, DC, USA, 230 pp., 2002.

[19] Perrings, C., The economics of abrupt climate change, *Philosophical Transactions of the Royal Society*, London, A 361, 2043-59, 2003.

[20] Schneider, S. H., B. L. Turner, and H. Morehouse Gariga, Imaginable Surprise in Global Change Science, *Journal of Risk Research*, 1 (2): 165-185, 1998.

[21] Schneider, S. H., K. Kuntz-Duriset, and C. Azar, Costing Nonlinearities, Surprises, and Irreversible Events, *Pacific and Asian Journal of Energy*, 10 (1): 81-106, 2000.

Provably Fast Algorithms for Anomaly Detection

D. Hush, P. Kelly, C. Scovel and I. Steinwart
Modeling, Algorithms and Informatics Group, CCS-3
Los Alamos National Laboratory
Los Alamos, NM 87545
{dhush,kelly,jcs,ingo}@lanl.gov

ABSTRACT

We consider one of the most common anomaly detection formulations and describe a solution method that is proven to be computationally efficient, universally consistent, and to guarantee near optimal finite sample performance for a large class of (practical) distributions [23, 21]. We also describe an algorithm for this method that accepts the desired accuracy ϵ as an input and produces an approximate solution that is guaranteed to satisfy this accuracy in low order polynomial time. Experimental results are presented to demonstrate the actual run times for a *typical* problem.

1. INTRODUCTION

In a recent paper we describe a new solution method for one of the most common anomaly detection formulations [23]. This method is unique in that it is proven to be computationally efficient, universally consistent, and to guarantee near optimal finite sample performance for a large class of (practical) distributions [23, 21]. Since this method solves a *density level detection* (DLD) problem using a *support vector machine* (SVM) approach (both described below) it is called the *density level detection support vector machine* (DLD–SVM). The DLD–SVM was recently compared with several popular methods [1] using real data from a cybersecurity problem and found to perform very well [23]. Indeed it gave the best overall performance and was far superior to some methods. In this paper we describe a provably fast algorithm for the DLD–SVM.

In practice most SVM algorithms produce *approximate* solutions and consequently they introduce a trade-off between computation and accuracy that is not well understood. The accuracy, as measured by the difference between the criterion value of the approximate solution and the optimal criterion value, is important for learning because it has a direct influence on the generalization error. The accuracy of the approximate solution produced by existing SVM algorithms is often unknown. In addition the computational requirements of existing SVM algorithms are largely unknown. However in this paper we describe a DLD–SVM algorithm that accepts the desired accuracy ϵ as an input and produces an approximate solution that is guaranteed to satisfy this accuracy in low order polynomial time. Our analysis reveals the effect of the accuracy on the run time, thereby allowing the user to make an informed decision regarding the trade–off between computation and accuracy. In addition this analysis provides a worst case bound on the number of iterations that is typically linear in the number of samples. We present experimental results which validate this linear relation, but also show that the actual number of iterations for a typical problem can be much smaller than the worst case bound.

[1]These popular methods included schemes based on Parzen density estimates, Gaussian density estimates determined by maximum likelihood parameter estimates, Mixture of Gaussians density estimates determined by the EM algorithm, and the 1-CLASS SVM [19].

KDD 2005 Chicago, IL USA

2. PROBLEM FORMULATION

Anomalies are often described as rare or unusual events. This notion can be represented mathematically by defining anomalies to be points with low probability density value. In particular the set of points with density value below a threshold ρ comprise the *anomalous set*, while the complement of this set is called the *normal set*. Our goal is to design a binary function (an anomaly detector) that assigns the value -1 to points in the anomalous set and $+1$ to points in the normal set.

To formalize these notions we first recall the basic concept of *density*. *Density* is a (local) valuation of the relative concentration of two measures. In particular, for two measures Q and μ on a space X where Q is absolutely continuous with respect to μ (i.e. every μ–negligible set is a Q–negligible set) the density h of Q with respect to μ is the Radon–Nikodym derivative $h = dQ/d\mu$. In the anomaly detection problem Q is an (unknown) probability measure that describes the data and μ a (known) reference measure. For example when $X \subseteq \mathbb{R}^d$ the reference μ is usually taken to be the Lebesgue measure (i.e. the standard volume). In principle however the reference measure is chosen by the user in a way that establishes a definition of anomalies relevant to the application. Given a density level $\rho > 0$, the normal set $\{h > \rho\}$ is called the *ρ-level set*. The goal of the *density level detection* (DLD) problem is to find an estimate of the ρ-level set of h and therefore an estimate of the anomalous set (by taking the complement). To find this estimate we use information

given to us by a training set $T = (x_1, \ldots, x_n) \in X^n$ that is i.i.d. drawn from Q. With the help of T a DLD algorithm constructs a function $\hat{f} : X \to \mathbb{R}$ for which the set $\{\hat{f} > 0\}$ is an estimate of the ρ-level set $\{h > \rho\}$. A standard performance measure that quantifies how well $\{\hat{f} > 0\}$ approximates the set $\{h > \rho\}$ is (see e.g. [1])

$$\mathcal{S}(f) \ := \ \mu \ \{f > 0\} \triangle \{h > \rho\} \quad ,$$

where $\triangle$ denotes the symmetric difference. The goal of the DLD problem is to find $\hat{f}$ such that $\mathcal{S}(\hat{f})$ is close to zero.

Now let μ be a probability measure and define the risk

$$\mathcal{R}(f) \ := \ \frac{1}{1+\rho} \ Q(f \leq 0) + \frac{\rho}{1+\rho} \ \mu(f > 0).$$

Steinwart et al. [23] show that any function that minimizes $\mathcal{R}$ also minimizes $\mathcal{S}$. Furthermore they prove a very tight relation between $\mathcal{R}$ and $\mathcal{S}$ for all functions f. This establishes $\mathcal{R}$ as a bona fide risk function for the DLD problem. Therefore $\mathcal{R}$ *is a legitimate performance measure for anomaly detection.* Consequently our goal of choosing $\hat{f}$ to (approximately) minimize $\mathcal{S}$ can be revised to choosing $\hat{f}$ to (approximately) minimize $\mathcal{R}$.

It turns out that $\mathcal{R}$ is also a performance measure for a supervised classification problem. Indeed let $Y \ := \ \{1, -1\}$ be the label set and let $x \in X$ and $y \in Y$ denote values of the random variables $\mathbf{x}$ and $\mathbf{y}$. The supervised classification problem is formed by identifying Q and μ with the conditional distributions $P_{\mathbf{x}|\mathbf{y}=1}$ and $P_{\mathbf{x}|\mathbf{y}=-1}$ respectively and defining the class marginals $P(\mathbf{y} = 1) \ := \ 1/(1+\rho)$ and $P(\mathbf{y} = -1) \ := \ \rho/(1+\rho)$. To form a data set for this classification problem we *collect* n_1 i.i.d. samples $(x_1, \ldots, x_{n_1})$ from Q and assign each of them the label $y = +1$, and we *synthesize* n_{-1} i.i.d. samples $(x_{n_1+1}, \ldots, x_{n_1+n_{-1}})$ from μ and assign each of them the label $y = -1$. This gives a training set $\mathcal{T} = ((x_1, y_1), ..., (x_n, y_n))$ of size $n = n_1 + n_{-1}$. The goal is to use $\mathcal{T}$ to choose a function $\hat{f}$ so that $\mathcal{R}(\hat{f})$ is as small as possible. The only difference between this problem and a standard classification problem is that the class marginal probabilities are *known.*

We now describe the DLD–SVM solution method. Let $k : X \times X \to \mathbb{R}$ be a kernel function, i.e. there exists a Hilbert space H and a map $\phi : X \to H$ such that $k(x_1, x_2) = \phi(x_1) \cdot \phi(x_2), \forall x_1, x_2 \in X$. SVM functions f take the form

$$f_{\psi,b}(x) = \psi \cdot \phi(x) + b.$$

The DLD–SVM determines the parameters $\hat{\psi}$ and $\hat{b}$ by (approximately) solving the *primal* QP problem

$$\begin{array}{ll} \min_{\psi,b,\xi} & \lambda\|\psi\|^2 + \sum_{i=1}^n u_i \xi_i \\ \text{s.t.} & y_i(\phi(x_i) \cdot \psi + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \quad i = 1, 2, ..., n \end{array} \tag{1}$$

where $\lambda > 0$ and

$$u_i = \begin{cases} \frac{1}{(1+\rho)n_1}, & y_i = 1 \\ \frac{\rho}{(1+\rho)n_{-1}}, & y_i = -1 \end{cases} \quad .$$

Since this QP problem can be prohibitively large (e.g. the dimension of ψ may be infinite) and its dual QP problem is considerably smaller we employ a two–stage process where the first stage produces an approximate solution to the dual QP problem and the second stage maps this approximate dual solution to an approximate primal solution. The *canonical dual* QP problem is

$$\begin{array}{ll} \max_\alpha & -\frac{1}{2}\alpha \cdot Q\alpha + \alpha \cdot w + w_0 \\ \text{s.t.} & 1 \cdot \alpha = c \\ & 0 \leq \alpha_i \leq u_i \quad i = 1, 2, ..., n \end{array} \tag{2}$$

where

$$Q_{ij} = k(x_i, x_j)/2\lambda, \ \ c = l \cdot 1, \ \ w = Ql + y, \ \ w_0 = -l \cdot y - \frac{1}{2} l \cdot Ql. \tag{3}$$

and

$$l_i = \begin{array}{ll} 0 & y_i = 1 \\ u_i & y_i = -1 \end{array} \quad . \tag{4}$$

We denote the canonical dual criterion by

$$R(\alpha) \ := \ -\frac{1}{2}\alpha \cdot Q\alpha + \alpha \cdot w + w_0.$$

We define the set of ϵ–optimal solutions to the canonical dual QP problem to be $\{\alpha \in \mathcal{A} : |R^* - R(\alpha)| \leq \epsilon\}$ where $\mathcal{A}$ is the set of feasible points and R^* is the optimal criterion value. We use a similar definition for the set of ϵ_p–optimal solutions to the primal QP problem.

Our approach is to compute an ϵ–optimal canonical dual solution $\hat{\alpha}$ and then map it to an ϵ_p–optimal primal solution $(\hat{\psi}, \hat{b}, \hat{\xi})$. Let $K \geq \max_i k(x_i, x_i)$. For a dual solution $\hat{\alpha}$ with accuracy $\epsilon = (2\sqrt{2K} + 8)^{-2}\lambda\epsilon_p^2$ the map

$$\hat{\psi} = \frac{1}{2\lambda} \sum_{i=1}^n (\hat{\alpha}_i - l_i)\phi(x_i)$$

$$\hat{b} \in \arg\min_b \sum_{i=1}^n u_i \max(0, 1 - y_i(\hat{\psi} \cdot \phi(x_i) + b)) \tag{5}$$

and

$$\hat{\xi}_i = \max(0, 1 - y_i(\hat{\psi} \cdot \phi(x_i) + \hat{b})), \ \ i = 1, .., n$$

has been shown to produce a primal solution with accuracy ϵ_p [20]. Thus if we let $\hat{\gamma}_i = \frac{\hat{\alpha}_i - l_i}{2\lambda}$ the corresponding SVM anomaly detector takes the form

$$f_{\hat{\psi},\hat{b}}(x) = \sum_{i=1}^n \hat{\gamma}_i k(x_i, x) + \hat{b}.$$

Pseudocode for the main routine which produces the values $\hat{\gamma}$ and $\hat{b}$ corresponding to an ϵ_p–optimal primal solution is shown in Procedure 1. This routine forms an instance of the canonical dual QP according to (3), sets the desired accuracy of the canonical dual solution $\epsilon = (2\sqrt{2K}+8)^{-2}\lambda\epsilon_p^2$, uses the routine `Composite` to compute an ϵ–approximate canonical dual solution, determines the expansion coefficients $\hat{\gamma}$, and uses the routine `Offset` to compute the offset parameter $\hat{b}$ according to (5). A simple $O(n \log n)$ algorithm for the routine `Offset` is described in Hush et al. [6]. Our focus here is on efficient algorithms for the routine `Composite`.

The routine `Composite` solves the canonical dual QP problem by solving a sequence of smaller QP problems where each of the smaller QP problems is obtained by fixing a subset of the variables and optimizing with respect to the

Procedure 1 The main algorithm for the DLD–SVM.

1: `INPUTS:` A data set $\mathcal{T} = ((x_1, y_1), ..., (x_n, y_n))$, a density level ρ, a kernel function k, and values λ and ϵ_p
2: `OUTPUTS:` Parameter values $\hat{\gamma}$ and $\hat{b}$
3:
4: Form canonical dual parameters:
5: $Q_{ij} = \frac{k(x_i,x_j)}{2\lambda}$, $l_i = \frac{(1-y_i)u_i}{2}$, $w = Ql + y$, $c = l \cdot 1$,
6: $\epsilon = \frac{\lambda\epsilon_p^2}{2\sqrt{2K}+8}$, and $u_i = \begin{cases} \frac{1}{(1+\rho)n_1}, & y_i = 1 \\ \frac{\rho}{(1+\rho)n_{-1}}, & y_i = -1 \end{cases}$
7: $\hat{\alpha} \leftarrow$ `Composite`(Q, w, c, u, ϵ)
8: Compute expansion coefficients: $\hat{\gamma}_i \leftarrow (\hat{\alpha}_i - l_i)/2\lambda$
9: $\hat{b} \leftarrow$ `Offset`$(\hat{\gamma}, \mathcal{T})$
10: Return$(\hat{\gamma}, \hat{b})$

remaining variables. A number of these so–called *decomposition* algorithms have been developed for SVMs [3, 4, 5, 7, 8, 10, 11, 14, 15, 16, 17, 18, 22]. The key to developing a successful decomposition algorithm is in the method used to determine the *working sets*, which are the subsets of variables to be optimized at each iteration. To guarantee stepwise improvement each working set must contain a *certifying pair* [7]. Stronger conditions are required to guarantee convergence [2, 3, 7, 9, 12, 13, 14] and even stronger conditions appear necessary to guarantee rates of convergence [7, 12]. Indeed, although numerous decomposition algorithms have been proposed few are known to possess polynomial run time bounds. However by restricting to working sets of size 2 and augmenting the working set selection algorithm introduced by Simon [22] we have constructed a decomposition algorithm called `Composite` whose worst case run time is a low order polynomial given by the following theorem. The proof of this theorem is obtained by Hush et al. [6] through a slight modification of the analysis of List and Simon [15].

THEOREM 1. *Consider the DLD–SVM canonical dual QP problem in (2) with criterion function R. Let $K \geq \max_i k(x_i, x_i)$, $r = (n_1 + n_{-1} - 1)/n_1$, and assume that the number of synthetic samples n_{-1} is chosen large enough so that $\frac{1}{(1+\rho)n_1} \geq \frac{\rho}{(1+\rho)n_{-1}}$. Define*

$$\beta := \frac{2Kr}{\lambda(1+\rho)^2 n_1}.$$

Then the `Composite` *decomposition algorithm in [6] achieves $R^* - R(\alpha^m) \leq \epsilon$ after $m = \acute{m}$ iterations of the main loop where*

$$\acute{m} = \begin{cases} 2rn_1 \ln \frac{1}{\epsilon}, & \epsilon \geq \beta \\ 2rn_1 \left(\frac{\beta}{\epsilon} - 1 + \ln \frac{1}{\beta} \right), & \epsilon < \beta \end{cases} \quad (6)$$

Furthermore the overall run time of this algorithm is

$$O\left(r^2 n_1^2 \log(1/\epsilon)\right) \text{ for } \epsilon \geq \beta \text{ and } O\left(\frac{Kr^3 n_1}{\lambda\epsilon(1+\rho)^2}\right) \text{ for } \epsilon < \beta.$$

For typical parameter values these run time bounds are on the order of n_1^2. This result is significant for two reasons. First, if we applied the fastest known algorithm for the general convex QP problem the run time bound would be $O(n^3/\log n)$. Thus by developing an algorithm for a specific class of QP problems we have obtained a significant improvement. Second, non–asymptotic run time guarantees with this type of efficiency are extremely rare for anomaly detection algorithms, especially for algorithms which also guarantee near optimal performance for such a large class of practical distributions.

To achieve the run time guarantees described by this theorem the `Composite` algorithm must be terminated properly. The simplest stopping rule that guarantees an ϵ–optimal solution is to stop after $\acute{m}$ iterations. However for a typical problem instance the algorithm may reach the accuracy ϵ in far fewer iterations. For this reason Hush et al. [6] introduce a rule that computes an upper bound on $R^* - R(\alpha)$ *adaptively* and then stops the algorithm when this upper bound falls below ϵ. With this rule we are able to achieve run times for *typical* problem instances that are much faster than the worst case bound.

3. EXPERIMENTAL RESULTS

To illustrate the run-time performance of the `Composite` training algorithm, we made use of the cybersecurity data set introduced in [23]. This data was derived from network traffic collected from a single computer over a 16-month period. Each vector in the data set contains 12 feature values, each representing some measurement of network activity over a one-hour window (e.g. "average number of bytes per session"). All values are normalized to fall in the interval $[0, 1]$. This collected data was used to represent samples from our unknown data distribution Q, and our goal was to build a detector that would recognize anomalous behavior from the machine. We used a uniform distribution over $[0, 1]^{12}$ for the background distribution μ. The kernel function used in our DLD-SVM problem was the Gaussian RBF kernel

$$k(x, x') = e^{-\sigma^2 \|x - x'\|^2}.$$

A grid search over λ and σ^2 was used to determine values that provided the best performance on a set of hold-out data [23]. This resulted in parameter values $\lambda = 10^{-7}$ and $\sigma^2 = 0.1$ and a solution that separates the training data. The associated hold-out value of $\mathcal{R}$ is 0.00025. The corresponding *alarm rate* (i.e. the rate at which anomalies are predicted by the classifier once it is placed in operation) is 0.0005. This corresponds to approximately one alarm every three months. This rate can be adjusted by training with a different value of ρ. It was also noted during the grid search that some parameter value selections (e.g. $\lambda = 0.05$ and $\sigma^2 = 0.05$) provided partial separation of the training data while exhibiting markedly different run-time behavior. We decided to repeat our experimental analysis using these parameter values as well.

In our experiments, we focus only on the run-time characteristics of the main loop in the `Composite` algorithm. We have purposefully omitted the setup time for each experiment from our plots. This non-trivial amount of work included: (A) drawing random samples from the base data sets; (B) setting up internal variables for the canonical dual formulation of the DVD-SVM problem; (C) initialization of the algorithm to a feasible solution; and (D) pre-computing

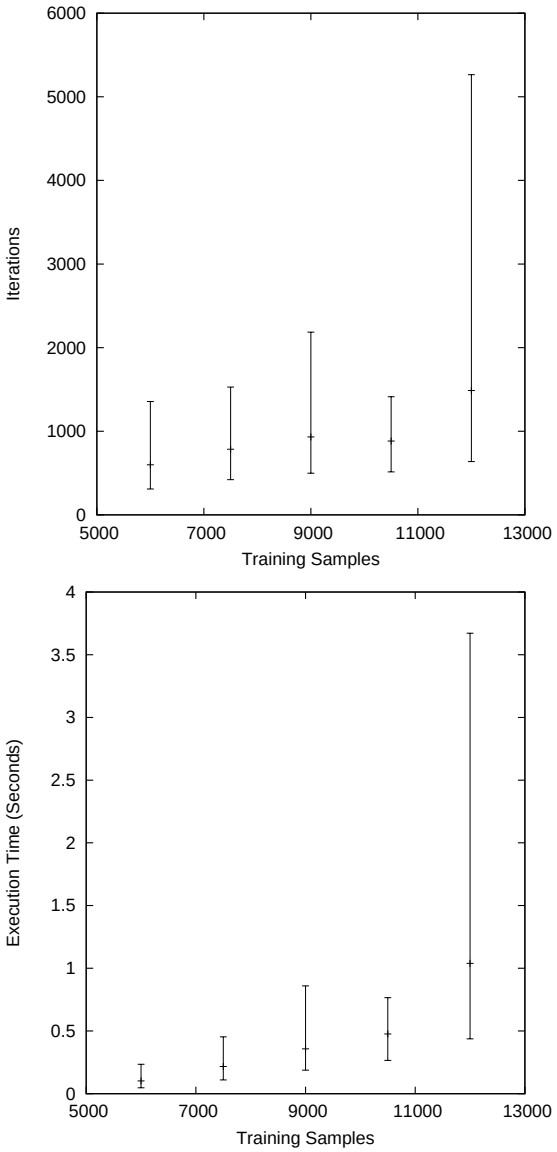

Figure 1: Training with ($\lambda = 10^{-7}$, $\sigma^2 = 0.1$).

all kernel values $k(x_j, x_k)$.

To study the effect of training set size on the run-time properties of `Composite`, we used five different problem sizes. For simplicity, we always chose the number of background data samples (drawn from μ) to be twice the number of actual network data samples (drawn from Q). The five problem sizes of data drawn from $Q : \mu$ were 2000:4000, 2500:5000, 3000:6000, 3500:7000, and 4000:8000. For each of these problem sizes, we performed ten different random samplings of our base data sets, and trained our DLD-SVM classifier on each. The density level ρ was always fixed at 1, and accuracy ϵ was fixed at 10^{-6}. Tabulated results include the min, max, and average number of main loop iterations. Results when using parameter values of $\lambda = 10^{-7}$ and $\sigma^2 = 0.1$ are given in Figure 1. For all of these experiments, our adaptive stopping criterion was able to terminate the main processing loop after a total number of iterations that was about 9 orders of magnitude smaller than the theoretical worst-case given by Equation 6. Wallclock execution times for the main processing loop were also tabulated, as shown in Figure 1. It is noteworthy that there was a significant variance in number of iterations across different random samplings of the same problem size (3x-8x).

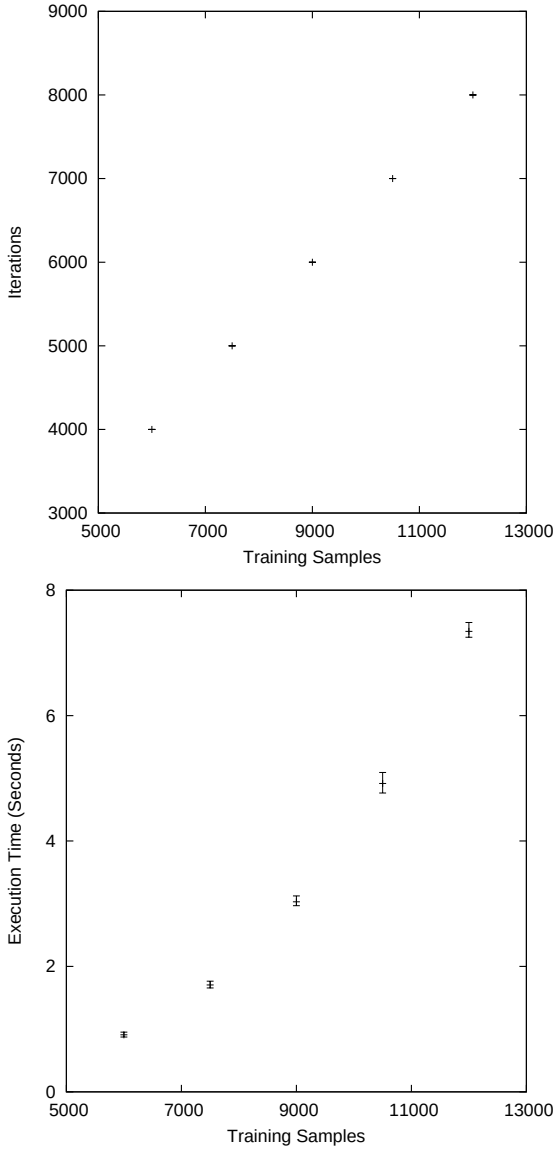

Figure 2: Training with ($\lambda = 0.05$, $\sigma^2 = 0.05$).

Results from our second set of experiments, using the same randomly sampled data sets that we used before, are given in Figure 2. This time we selected algorithm parameters ($\lambda = 0.05$ and $\sigma^2 = 0.05$) that gave solutions that did not separate the training data. This larger value of λ, which corresponds to strong regularization, caused our DLD-SVM to always produce a simple solution that discriminates based on a difference in means. All variables were forced to one of the extreme values defined by the canonical dual's inequality constraints (see Equation 3), and every randomly sampled subset of a given size required the same number of iterations for convergence. The relationship between the

number of main loop iterations and the training set size is demonstrably linear in this case, and the corresponding wallclock processing time indeed appears quadratic. Training always required more effort when using these parameter values than with those used in our first set of experiments.

4. CONCLUSIONS

We have proposed a solution method for training a DLD–SVM. This method is guaranteed to satisfy a user-provided accuracy ϵ in low order polynomial time. Experimental results suggest that actual run times can be many orders of magnitude smaller than our theoretical worst-case bounds.

5. REFERENCES

[1] S. Ben-David and M. Lindenbaum. Learning distributions by their density levels: a paradigm for learning without a teacher. *J. Comput. System Sci.*, 55:171–182, 1997.

[2] C. Chang, C. Hsu, and C. Lin. The analysis of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, 11(4):1003–1008, 2000.

[3] P.-H. Chen, R.-E. Fan, and C.-J. Lin. A study on SMO-type decomposition methods for support vector machines. Technical report, 2005. http://www.csie.ntu.edu.tw/ cjlin/papers.html.

[4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Canbridge ; United Kingdom, 1st edition, 2000.

[5] C.-W. Hsu and C.-J. Lin. A simple decomposition algorithm for support vector machines. *Machine Learning*, 46:291–314, 2002.

[6] D. Hush, P. Kelly, C. Scovel, and I. Steinwart. QP algorithms with guaranteed accuracy and run time for support vector machines. Technical report 05-5165, Los Alamos National Laboratory, 2005. submitted for publication.

[7] D. Hush and C. Scovel. Polynomial-time decomposition algorithms for support vector machines. *Machine Learning*, 51:51–71, 2003.

[8] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA, 1998.

[9] S. Keerthi and E. Gilbert. Convergence of a generalized SMO algorithm for SVM classifier design. *Machine Learning*, 46:351–360, 2002.

[10] S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13:637–649, 2001.

[11] P. Laskov. Feasible direction decomposition algorithms for training support vector machines. *Machine Learning*, 46(1–3):315–349, 2002.

[12] C.-J. Lin. Linear convergence of a decomposition method for support vector machines. Report, 2001. http://www.csie.ntu.edu.tw/ cjlin/papers.html.

[13] C.-J. Lin. On the convergence of the decomposition method for support vector machines. *IEEE Transactions on Neural Networks*, 12:1288–1298, 2001.

[14] N. List and H. Simon. A general convergence theorem for the desomposition method. In J. Shawe-Taylor and Y. Singer, editors, *17th Annual Conference on Learning Theory, COLT 2004, volume 3120 of Lecture Notes in Computer Science*, pages 363–377, 2004.

[15] N. List and H. Simon. General polynomial time decomposition algorithms. Report, 2005. submitted for publication.

[16] O. Mangasarian and D. Musicant. Lagrangian support vector machines. *Journal of Machine Learning Research*, 1:161–177, 2001.

[17] E. Osuna, R. Freund, and F. Girosi. Support vector machines: training and applications. Technical Report AIM-1602, MIT, 1997.

[18] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 41–64. MIT Press, Cambridge, MA, 1998.

[19] B. Schölkopf, J. Platt, J. Shawe-Taylor, and A. Smola. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

[20] C. Scovel, D. Hush, and I. Steinwart. A computation–performance bound for SVM classifiers. *in preparation*, 2005.

[21] C. Scovel, D. Hush, and I. Steinwart. Learning rates for density level detection. *Analysis and Applications*, 2005. to appear.

[22] H. Simon. On the complexity of working set selection. In *Proceedings of the 15th International Conference on Algorithmic Learning Theory*, 2004.

[23] I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6:211–232, 2005.

Trajectory Boundary Modeling of Time Series for Anomaly Detection

Matthew V. Mahoney and Philip K. Chan

Computer Science Dept.
Florida Institute of Technology
Melbourne FL 32901

{mmahoney,pkc}@cs.fit.edu

ABSTRACT

We address the problem of online detection of unanticipated modes of mechanical failure given a small set of time series under normal conditions, with the requirement that the anomaly detection model be manually verifiable and modifiable. We specify a set of time series features, which are linear combinations of the current and past values, and model the allowed feature values by a sequence of minimal bounding boxes containing all of the training trajectories. The model can be constructed in $O(n \log n)$ time. If there are at most three features, the model can be displayed graphically for verification, otherwise a table is used. Test time is $O(n)$ with a guaranteed upper bound on computation time for each test point. The model compares favorably with anomaly detection algorithms based on Euclidean distance and dynamic time warping on the Space Shuttle Marrotta fuel control valve data set.

Keywords

Time series anomaly detection, Machine health monitoring, Path model, Box model, Rule Learning, NASA.

1. INTRODUCTION

In 1996 an Ariane 5 rocket self destructed during launch because the primary and backup flight control units had identical software errors. In each processor, a 64 bit floating point number was assigned to a 16 bit integer, raising an unhandled Ada overflow exception and halting it [1]. In 1999 the Mars Climate Orbiter was lost when engineers sent navigation commands using English units, while the spacecraft was expecting metric units [2]. In 2004, half of the data sent by the Huygens probe to Titan was lost because one of two receiver channels on the Cassini mother craft orbiting Saturn was not turned on due to a software error [14].

We are given the task of automating the detection of mechanical failures in the Marrotta fuel control valves used in the space shuttle. Because not all failure modes can be anticipated, this is an ideal task for time series anomaly detection: train a model on known good data, estimate the probability distribution, and assign a likelihood-based score to new sensor data. However, NASA is keenly aware of the consequences of software errors on a manned spacecraft. Therefore a requirement of our project is that the model be transparent. It is not enough that we demonstrate the ability to detect anomalies caused by simulated failures in the lab. Engineers also want to know *what* the modeler learned, and if necessary, manually update the model using domain specific knowledge. Unfortunately, many good time series anomaly detection algorithms produce opaque models that are difficult to analyze.

Our goal is to produce an anomaly detection system whose model is transparent. In addition, testing must be online, fast, and generalize when given more than one training series. By online, we mean that each test point receives an anomaly score, with an upper bound on computation time. We accept that there is no "best" anomaly detection algorithm for all data, and that many algorithms have *ad-hoc* parameters which are tuned to specific data sets. Therefore our subgoal is to provide tools to make this tuning easier on a given data set. The software that allows this capability is not directly discussed in this paper.

Our approach is to offer a set of models based on feature trajectory paths, because these models can be visualized in two or three dimensions, or coded as rules which can be edited in higher dimensions. A *feature* is defined as a linear combination of present and past values (a digital filter), for example, a time lagged copy, a derivative, or a smoothed signal. Thus, a feature is also a time series. Given d features, a signal traces a path or trajectory through d-dimensional feature space. The idea is that a test series should follow a similar trajectory to that of a known good training signal, or at least be near the training trajectory at all times. An engineer may choose to approximate the trajectory using straight line segments or a sequence of boxes for performance reasons. There may also be more than one training series, in which case we can construct a model which encloses all of the trajectories.

Our main contributions include:

- we propose two anomaly detection methods based on models that are transparent/editable, generalizable from multiple training time series, efficient during testing, and provide online scoring during testing;
- our empirical results from the NASA shuttle valve data indicate that our methods can detect similar or more abnormal time series than three existing methods.

The rest of the paper is organized as follows. In Section 2 we discuss related work. In Sections 3 and 4 we introduce path and box modeling respectively, along with efficient algorithms for generating approximations. In Section 5 we present experimental results with the NASA valve data set. In Section 6, we conclude.

2. RELATED WORK

One view of time series anomaly detection is that of a machine learning or modeling task. Given a training set X of time series with an unknown probability distribution P, the task is to estimate P. Then given a new time series y, we assign an anomaly score inversely related to $P(y)$. Ypma [15] surveys some important techniques, such as Bayesian models, neural networks, and support vector machines, and applications to the detection of failures in rotating machinery using vibration sensors

Dasgupta and Forrest [4] uses an immunological approach. A time series is quantized and chopped into fixed length strings of several symbols. A random set of strings is generated. Any strings which match the training data are removed. The remaining strings form an anomaly model. If a test signal matches any strings in the model, then an alarm is signaled. This technique was shown to detect simulated failures in a milling machine.

Keogh approaches the problem as that of finding a dissimilarity function $D(x, y)$ between a (normal or good) training series x and a test series y [7]. Viewed this way, we avail ourselves of the vast body of research in related data mining topics such as classification, clustering, and search. The simplest measure is Euclidean distance:

$$D_{EUCLID}(x, y)^2 = \sum_{i=1}^{N} (x_i - y_i)^2 \qquad (1)$$

where both series have length N and $x_1, x_2, ..., x_N$ are the N values of x. In some applications, we normalize x and y to have zero mean and unit standard deviation. Two disadvantages of this measure are that the series must have equal length and it is sensitive to shifts in time. Dynamic time warping (DTW) overcomes these problems by finding the minimum Euclidean distance when the data points of both series may be shifted arbitrarily in time (but maintained in order). DTW is defined recursively as follows:

$$DTW(x, y) = \sqrt{D(x_1^m, y_1^n)} \qquad (2)$$

where

$$D(x_1^i, y_1^j) = (x_i - y_j)^2 + \min[D(x_1^{i-1}, y_1^j), D(x_1^i, y_1^{j-1}), D(x_1^{i-1}, y_1^{j-1})]$$

and x_1^i means the sequence $x_1, x_2, ..., x_i$ and $D(x, y)$ is infinite if either x or y is empty. A *warp path* is the set of (i,j) from (1,1) to (m,n) such that if all x_i are aligned with y_j by shifting them in time, then $DTW(x, y) = D_{EUCLID}(x, y)$.

A disadvantage of DTW is that computation time is $O(mn)$. Various fast approximations have been proposed. For example, Salvador [10] describes FastDTW, an approximation to DTW in which the warp path is estimated as successively higher resolutions and the search is constrained within a radius of the previous estimate.

Many other distance measures have been proposed. In an exhaustive test, Keogh and others at UCR implemented about 50 proposed distance measures published over a 10 year period and evaluated them on a variety of data mining tasks on a large corpus of time series from diverse domains [7]. The rather surprising finding is that while many of the proposed measures improve over existing techniques on the specific data sets on which they were tested, none did better than normalized Euclidean distance over the entire data set.

Keogh also proposes a very general method which does outperform Euclidean distance on this diverse set: a compression dissimilarity measure, or CDM [8], defined as:

$$CDM(x, y) = \frac{C(xy)}{C(x) + C(y)}$$

where $C(x)$ is the compressed size of a symbolic (SAX) representation of x, saved as a file and compressed with an off-the-shelf compressor such as *gzip*. The idea is that CDM estimates the information shared by x and y. If the two series are identical, then a compressor can store y as a reference to x, so $C(xy) \approx C(x)$ and $CDM(x, y) \approx 0.5$. If x and y are unrelated, then the compressor cannot use knowledge from x to model y, so $C(xy) \approx C(x) + C(y)$ and $CDM(x, y) \approx 1$.

2.1 Feature Trajectory Models

Some proven and broadly applicable techniques such as CDM and neural networks suffer from opacity. It is not at all clear from the state of a data compression program or the trained weights of a neural network exactly what has been learned. Our work is based on trajectory modeling in feature space as described by Povinelli et. al. [9]. Povinelli extracted d features of a time series, which are simply time-lagged copies of the data delayed by t, $2t$, $3t$, ... dt, and d and t are parameters. The density in d-dimensional feature space is modeled by clustering the training points and using a Gaussian mixture model to approximate the clusters. A test point is evaluated by its distance (in standard deviations) from the nearest cluster. The model was shown to classify phonemes in speech, detect arrhythmias in ECG traces, and detect mechanical failures in a motor simulation.

Generating a Gaussian mixture model requires a slow, iterative process. Vlachos et. al. [13] describe a minimum bounding rectangle (MBR) clustering algorithm that runs in $O(n \log n)$ time that is nearly identical to the one used in our system. A sequence of n points in feature space is first approximated by a sequence of $n - 1$ boxes, each enclosing a pair of adjacent points. Then pairs of adjacent boxes are merged by greedily selecting the pair that minimizes the increase in volume after merging. The algorithm for modeling the sequence of n points $x_1, x_2, ..., x_n$ using k boxes is as follows:

```
MBR(x1...n, k)
  For each i in [1, n-1] do
    xi := merge(xi, xi+1)
  Delete xn
  While n > k do
    Find i minimizing ΔV =
       V(i, i+1) - V(i) - V(i+1)
       (minimize increase in volume)
    xi := merge(xi, xi+1)
    Delete xi+1
  Return x = x1...k
```

Fig. 1. MBR Algorithm.

In the MBR algorithm, `merge(x, y)` means to replace points or boxes *x* and *y* with the smallest box that encloses both, `V(i)` means the volume of x_i, and `V(i, i+1)` means the volume of `merge(`x_i`, `x_{i+1}`)`. ΔV is the increase in volume that would result from merging. Deleting an element x_i implicitly decrements *n*.

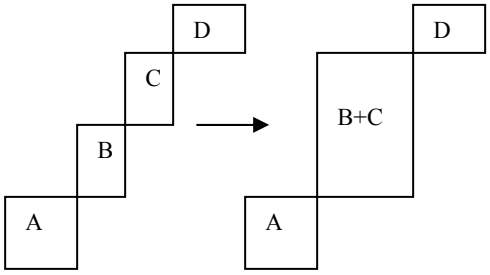

Fig. 2. Merging boxes B and C in the MBR algorithm.

MBR can run in $O(n \log n)$ time by storing the boxes in a heap ordered by ΔV, the increase in volume that would result from merging it with the next box. In a heap, the elements are stored in a balanced binary tree such that at each node the parent is smaller than the two children. Each node x_i also stores pointers to x_{i-1} and x_{i+1} to form a doubly linked list. When the box at the root of the heap is merged with its neighbor, the two old boxes are removed from the heap, the merged box is inserted, and ΔV of the two neighbors of the new box are updated, requiring them to be sifted up or down the heap. Each of the heap operations takes $O(\log n)$ time.

2.2 Gecko

In our earlier work on the NASA valve data [5], we used the Gecko algorithm [11] to create a bounded rectangle model. The Gecko model is more complex and less efficient than MBR in the training phase, but our interest is in the correctness of the model and efficiency in the testing phase. Gecko uses 3 dimensions of feature space: the original signal and the first and second derivatives, each of which is smoothed by a low pass filter. The trajectory is then segmented in feature space using a bottom-up clustering algorithm. Next, RIPPER [3] is used to generate a minimal rule set which separates the clusters. Each rule corresponds to one surface of one box, for example "*if segment = 3 then feature2 < 2.5*". It is possible to define one segment by several boxes, and some boxes may be open on some sides. Gecko, like MBR, satisfies our criteria that the model be comprehensible. The feature space can either be visualized in three dimensions, or expressed as a set of *if...then* rules.

During testing, a state machine is constructed such that each state corresponds to one trajectory segment, plus one error state. A transition to the next state occurs if the number of consecutive points satisfying the rules for the new state (falling within one of the bounding boxes) exceeds a threshold. An error occurs if the number of consecutive points satisfying neither the current nor next state exceeds a second threshold. Both thresholds are user defined parameters.

Gecko has been extended to handle multiple training series. First, the series are aligned by DTW or FastDTW. Next, the aligned series are averaged. Then the averaged series is segmented as before. Finally RIPPER is applied to separate the points in the original series that align with different segments in the merged series.

3. PATH MODELING

Our work in time series modeling falls between two extremes. At one end, we have a single training series, and we compute the distance from it using some function. At the other extreme, we have a large set of training sequences (or a single series with thousands of cycles) which we model using a probability distribution in a feature space and then estimate the probability of the test series. The NASA valve data set is one example of a data set that falls in the middle. We have one to four "normal" training series from which we generalize to a model. Our approach is to construct a model that encloses all of the training trajectories and the space "between" them.

We describe two representations that approximate this space, *path modeling* and *box modeling*. For path modeling, we store the training trajectories and test whether the sensor data falls between or near these paths. For box modeling, we construct a sequence of boxes enclosing all of the training paths, and test whether a test point falls within or near these boxes. We describe path modeling in this section, and box modeling in Section 4.

For the case of a single training path *x* in *d*-dimensional feature space, and a point y_i in a test series *y*, we could assign an anomaly score $D(x, y_j)$ equal to the square of the Euclidean distance between y_j and the nearest point in *x*.

$$D(x, y_j) = \min_{i \in [1,n]} \sum_{k=1}^{d} (x_{ik} - y_{jk})^2 \qquad (3)$$

where x_{ik} denotes the value of the *k*'th feature of the *i*'th point in *x*. This measure would have two problems. First it is inefficient because the testing time would be $O(dn)$ per test point (or $O(n)$ best case if we test the nearest points first). Second, the score would be nonzero even for the case of a test path following the training path exactly, because *x* is sampled and y_i could fall between the sample points in *x*. Addressing the latter problem by increasing the number of samples would make the first problem worse.

Our approach is to model *x* using a piecewise linear approximation of $k - 1$ straight line segments defined by *k*

vertices, where k is a parameter. Then we define $D(x, y_j)$ to be the square of the Euclidean distance between y_j and the nearest point in the approximation of x. The computation time is now $O(kd)$, where $k << n$. Computing the distance between a point and a line segment is more complex than computing the distance between two points, but is still $O(d)$.

Depending on the domain, we might require that the test signal follow the same trajectory as the training data in the same order. This restriction, which we call *sequential testing*, is used by Gecko and is appropriate when we require the training and test series to have the same overall shape, while still allowing time shifts. Suppose that the line segment (x_i, x_{i+1}) is the closest segment to test point y_j. Then it is only necessary to test the next point, y_{j+1} by computing the distance to the current and next segments, (x_i, x_{i+1}) and (x_{i+1}, x_{i+1}). We maintain i as a state variable and set it to the index of the closest segment. The time to compute $D(x, y_j)$ is now $O(d)$. Other variations are possible, such as also testing the previous segment to allow backwards movement.

Path modeling can be extended to multiple training series in a number of ways. For example, we could use 1-nearest neighbor modeling, in which the anomaly score is the square of the distance to the nearest path. If our training set is limited, it may be desirable to test whether a point lies "between" the training paths. Depending on how we define "between", this can lead to difficult calculations. We use the following definition, which is an easy to compute approximation. Given p paths and a test point y_j, we find the nearest point on each path, and then find the smallest box that will enclose all p nearest points. If y_j is inside this box, its anomaly score is zero. Otherwise its score is the square of the Euclidean distance to the box (Fig. 3). For a single path, this reduces to finding the minimum Euclidean distance from the test point to the path.

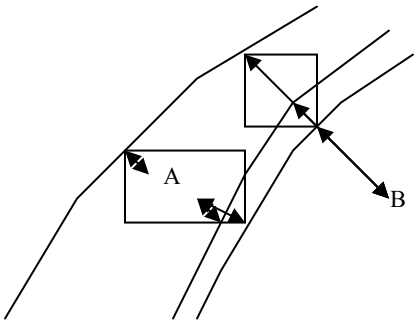

Fig. 3. Computing distance to multiple paths. Point A is inside the box enclosing the three points nearest to it, so its score is 0. Point B is outside the box enclosing the three points nearest to it, so its score is the distance squared to that box.

The test time complexity of multiple path modeling is $O(pkd)$ given p paths, k segments per path, d dimensions and stateless modeling (testing all path segments). Run time improves to $O(pd)$ using sequential testing and maintaining a nearest segment state for each path. Later in Section 4, we will eliminate the $O(p)$ penalty by approximating the training paths with a sequence of boxes that enclose them.

3.1 Path Model Generation

To approximate x with $k - 1$ line segments defined by k vertices, we use a greedy bottom-up approach. The *vertex removal* algorithm removes $n - k$ vertices. Referring to Figure 4, the effect of removing vertex B in the sequence ABC is to replace the two line segments AB and BC with the line segment AC. This induces an error, which we define to be $|AC||BB'|^2$, where $|AC|$ is the length of segment AC, and $|BB'|$ is the distance from B to B', the nearest point on segment AC. The justification for this definition is that if we were to test the training data on itself, then the measured anomaly score would be proportional to our proposed measure, while the true anomaly score should be zero.

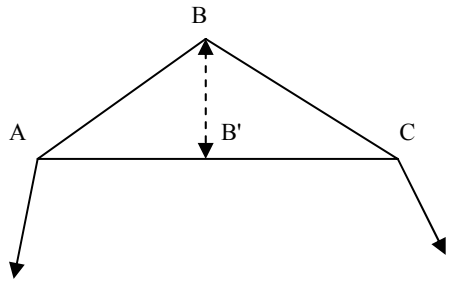

Fig. 4. Removing vertex B induces an error approximated by $|AC||BB'|^2$

An improvement to vertex removal is *path fitting*, in which, after removing B, we shift A and C a distance of $|BB'|/4$ in the direction from B' to B (Fig. 4). If the path is smooth with a gradual curve, then this has the effect of reducing the error because the new segment A'C' is a better fit to ABC than the original AC in the vertex removal algorithm. An optimal shift for AC alone would be $|BB'|/2$, but this would induce too much error in the segments adjacent to A'C'.

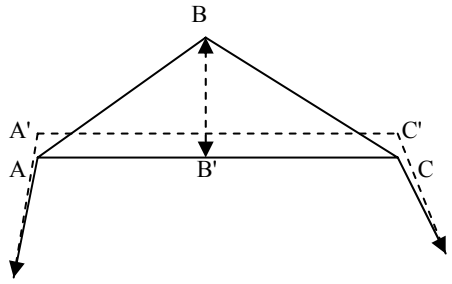

Fig. 5. Path fitting. After removing vertex B, A and C are shifted 1/4 the distance from B' to B to reduce the induced error.

The algorithm for path fitting is given in Fig. 6. The input is the sequence of n vectors $\mathbf{x_i}...\mathbf{x_n}$ in d-dimensional feature space and the desired number of vertices, k. The algorithm runs in $O(n \log n)$ time by storing the vertices in a doubly linked heap, as in the MBR algorithm. The vertices are sorted by error with the smallest at the root. When a vertex is removed, the stored errors of the two nearest neighbors on each side are updated, and they are sifted up or down to restore the heap property. The vertex removal algorithm is identical to path fitting except that the shift

is zero and only one neighbor on each side of the removed vertex needs to be updated.

```
path_fit(x1...xn , k)
  while n > k do
    find i minimizing error(xi)
    b := point on (xi-1,xi+1) nearest xi
    shift := (xi - b)/4
    xi-1 := xi-1 + shift
    xi+1 := xi+1 + shift
    (xi...xn-1) := (xi+1...xn)
    n := n - 1
  return (x1...xk)
```

Fig. 6. Path fitting algorithm

4. BOX MODELING

Building a box model follows the MBR algorithm described in Section 2 with two modifications. First, instead of merging two boxes into one, we merge three boxes into two. Second, we model multiple paths by first constructing a box model of one path, then expanding the boxes to enclose the other paths. In addition, testing differs from MBR in that the test series is not also converted to a box model. This allows us to assign an anomaly score to each test point online.

Box merging is shown in Fig. 7. We first find the box whose removal results in the smallest increase in volume (ignoring overlap between nonadjacent boxes). Then to remove the box, we expand the two neighboring boxes just enough to include the center of the removed box. We call this algorithm MBR3.

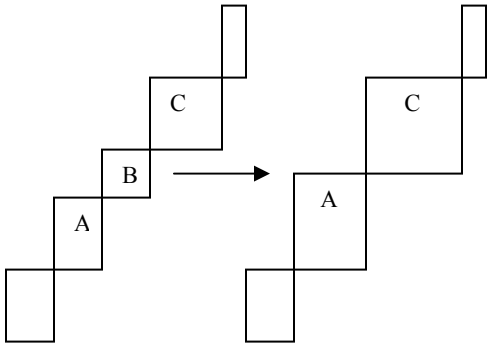

Fig. 7. When box B is removed in MBR3, boxes A and C are grown to enclose the center of B.

The intent of MBR3 is to produce a more uniform distribution of box sizes than MBR. (However we did not test this, nor claim that we succeeded). However MBR3 has the disadvantage that the original path is no longer guaranteed to be enclosed by the new boxes. This occurs when the original path does not pass exactly through the center of the removed box.

The second modification is to expand the *k* boxes that approximate the first training path to contain all of the remaining *p* - 1 paths. This is done in two passes for each path. First, we label each point in the path with the box that is closest to it. In the second pass we expand the boxes to enclose the points with matching labels. We do this one path at a time to reduce the space complexity between passes from $O(pk)$ to $O(k)$. Two passes are required because consecutive points in a path tend to be close together, which could result in a pathological model in which a single box grows in small steps to enclose the entire data set. The algorithm is given in Fig. 8..

```
box_expand(x1...xk, y1...yn)
    (x: sequence of k boxes)
    (y: sequence of n points)
    (output: x expanded to enclose y)
  for each yj
    lj = i: xi is closest box to yj
  for each yj
    expand xlj to enclose yi
```

Fig. 8. Expanding box sequence *x* to enclose path *y*.

We recommended that the first path (the input to MBR3) be included in the box expansion step, even if it is the only path. This solves the problem mentioned earlier in which the path may lie slightly outside the box model.

Note that the box model depends on the order in which the paths are presented. We recommend that the most "average" path be used as the initial input to MBR3, and to present the outlier cases last

5. EXPERIMENTAL RESULTS

In this section, we compare path and box modeling with Euclidean distance, DTW and Gecko+RIPPER on the NASA valve data set. The purpose of the experiments is to show that it is possible to construct working anomaly detection systems based on path or box modeling for this data set.

5.1 NASA Valve Data Set

The NASA valve data set [5] consists of solenoid current measurements recorded on Marrotta series MPV-41 valves as they are remotely opened and closed in a laboratory. These small valves are used to actuate larger, hydraulic valves that control the flow of fuel to the space shuttle engines. Sensor readings were recorded using either a shunt resistor or a Hall effect sensor under varying conditions of voltage, temperature, or blockage or forced movement of the poppet to simulate fault conditions.

There are several data subsets, of which two are suitable for testing anomaly detection systems. These are the TEK and VT1 (voltage test 1) sets. The TEK set contains 4 normal and 8 abnormal time series. The four normal traces are labeled TEK 0 through TEK 3, and vary slightly in the degree of background noise, duration of the "on" cycle, and average current during both the "on" and "off" portions. The abnormal series (TEK 10 through 17) were generated by restricting or forcing the movement of the poppet, which has the effect of changing the shape of the rising and falling edges of the waveform. All of the waveforms consist of 1000 samples at a rate of 1 ms per sample. The trace begins at time -0.1s. The valve is actuated at time 0, and deactivated at various times, typically around time 0.2s to 0.3s. The "on" current is approximately 4 in unspecified units.

The "off" current is approximately 0. Measurements are quantized with a resolution of 0.04. In our experiments we do not use TEK 4 through TEK 9 because these are partial waveforms with different sampling rates.

Figure 9 shows three typical waveforms, TEK 0, 10, and 16. TEK 0 is normal. The spikes on the rising and falling edges of the waveform are due to induced voltage caused by movement of the solenoid magnet during opening and closing of the poppet. In TEK 10, the poppet is blocked, so these spikes are absent. In TEK 16, the poppet is initially blocked, then released during the middle of the "on" cycle, causing a temporary dip in the current. It lacks a spike on the rising edge, but has a normal spike on the falling edge.

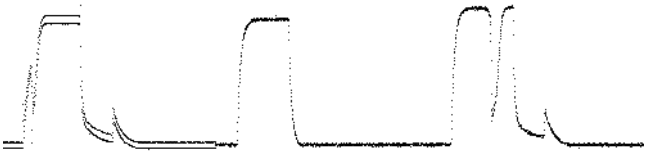

Fig. 9. Concatenation of TEK 0, 10, and 16.

In addition to these differences, there are also differences unrelated to valve failure. TEK 0, 1, and 15 have a 500 Hz signal with amplitude 0.24 as a background signal, visible in the first waveform as a double line. TEK 0 also has a large 2 ms alternating current spike at the start of the falling edge (not visible at this scale) that is absent in the other traces.

The second data set is the VT1 set. This consists of 27 time series recorded under varying conditions of voltage, temperature, and poppet blockage. Each series is 20,000 samples over a period of 2 seconds. In all cases the valve is actuated at time 0.5 sec. and deactivated at time 1.3 sec. For each series there are two readings, the first with a shunt resistor and the second with a Hall effect sensor. In our experiments we use the Hall effect measurement because it is less noisy but otherwise identical. The "off" current is approximately 0 A. The "on" current ranges from 0.42 to 1.08 A, increasing with voltage and decreasing with temperature (due to increased resistance of the solenoid coil). The voltage ranges from 14 V to 32 V in steps of 2 V at room temperature (21C or 22C). At 4 V steps (16, 20, 24, 28, 32) there is an additional recording for high temperature (69C to 71C) and one recording each for a poppet impedance of 4.5 and 9 mils. There are three runs under normal conditions at 32 V, but only one run for all other test conditions. The poppet fails to open at 14 V and at 16 V at high temperature.

In this paper we use the following notation to refer to VT1 traces: *V* for voltage, *T* for high temperature, i45 or i90 for 4.5 or 9.0 mil impedance. For example, *V24i45* denotes 24 V and 4.5 mil impedance. *V32T* denotes 32 V and high temperature.

5.2 Experimental Procedures and Evaluation Criteria

We test each proposed anomaly detection algorithm on the TEK and data sets. In each case we train the model on a proper subset of the training data, assign anomaly scores to all of the traces, and compare the normal and abnormal scores.

We say that an abnormal trace is detected if it has a higher score than all of the normal traces, whether those traces were included in the training set or not. We evaluate an anomaly detection system by the number of detections.

We evaluate the following algorithms.

- Euclidean model (equation (1)), with and without normalization.
- DTW (equation (2)), with and without normalization.
- Gecko with default parameters (tuned to TEK 0-1).
- Path modeling with parameters tuned for best results.
- Box modeling with parameters tuned for best results.

The VT1 set does not label the data as normal or abnormal. In our experiments we define "normal" to be the set of traces at low temperature with no impedance in the range 18 V to 30 V. Thus, there are 7 normal traces: V18, V20, V22, V24, V26, V28 and V30. We use the VT1 set to test the capability of Gecko, path and box modeling to generalize to unseen voltages given a subset of the normal voltages, and to detect temperature and impedance anomalies at unseen voltages. This test arrangement is not suitable for testing Euclidean distance or DTW because they cannot generalize.

By adjusting the threshold on the anomaly scores, different detection and false alarms rates can be obtained. For this study, we choose a threshold that yields no false alarms. That is, the threshold is set to be higher than the anomaly scores obtained from the normal traces (including those that are not used in training). In practice this is reasonable because normal traces are readily available for tuning the threshold and unforeseen bad traces are not available.

5.2.1 Euclidean Distance and DTW

Euclidean modeling requires that the time series be aligned. Recall that only the rising edge of the TEK waveforms are aligned. We test two solutions to the TEK alignment problem.

- Test the rising edge only.
- Manually align the falling edge.

To test the rising edge only, the series are truncated at time 0.1s, at which point the "on" current has stabilized. To align the falling edge, we insert copies of or remove samples at time 0.1s to align the falling edge to 0.2s and then truncate at time 0.78s.

5.2.2 Gecko+RIPPER

We tuned the Gecko parameters to produce the best results we could find on the TEK data set: a consecutive error threshold of 5, a consecutive next state threshold of 1, a smoothing window of size 2, and a derivative window of size 11 (5 before and 5 after). Although a Gecko model can be edited, we did not do so.

Gecko is designed to give a pass/fail result. The test data determines the transitions in a sequential state machine, which either goes to an accepting state or an error state. However, the current version will also produce an anomaly score using a rather complex algorithm which we outline here; see [12] for details. The modification is to run as a "nondeterministic" state machine, in which the state is the set of segments for which the test point satisfies the rules. When a point fails to satisfy the rules of either the current or next segment, that segment is removed from the set. When the set is empty, Gecko goes into a recovery mode in which it tests segments in an exponentially growing window starting at

the last known matching segment. Gecko outputs an anomaly score as a time series which increases by 1 at each step when the set is empty and decreases by 1/3 otherwise. The final score is the sum of these outputs.

5.2.3 Path and Box Modeling

We used the same feature set for path and box modeling. For features, we used the smoothed signal, and the smoothed first and second differences to create a 3-D feature space. We chose the first and second differences because they are intuitive (each test point should match the level, slope, and curvature of a training point), but it is actually the time lag in the smoothing filters that makes the model work. The smoothing is also necessary because the valve data is quite noisy. We selected the filters based largely on visual inspection of the output, and found that additional filtering is needed after each difference operation.

Specifically, we built the filters from two primitive elements, a two tap low pass infinite impulse response filter, *F*, and a two tap finite impulse response difference filter, D. *F* is defined:

$$F(x_i) = \frac{(T-1)F(x_{i-1}) + x_i}{T}$$

where T is the filter time constant and x_i is the input at time i. $F(x_0)$ is initialized to 0. *D* is defined:

$$D(x_i) = x_i - x_{i-1}$$

The three features are:

$$current = F(F(x))$$

$$d\_current = F(F(D(current)))$$

$$d2\_current = F(F(D(d\_current)))$$

To make a distance measure meaningful, each of the features should play a role. In this experiment, we scale the three features to fit a unit cube, so that the training data always ranges from 0 to 1. Other approaches are certainly possible, such as normalizing to unit standard deviation, or specifying the scaling as parameters.

Smoothing allows the output to be subsampled at the rate $1/T$ to speed processing with little loss of information. We do this for all of our experiments.

Figure 10 shows a 3-D view of a path model. Our software allows the user to rotate the image with the mouse, making it easier to visualize. In the figure, the three closely spaced loops are the trajectory path approximations of TEK 0, 2, and 3, each segmented by the path fitting algorithm with k = 25 segments. The outer loop of connected dots is the test path of TEK 16, which has not been approximated. As can be seen, the points on TEK 16 lie far from the three training paths. This model uses a filter time constant of T = 4 ms with subsampling at the same rate.

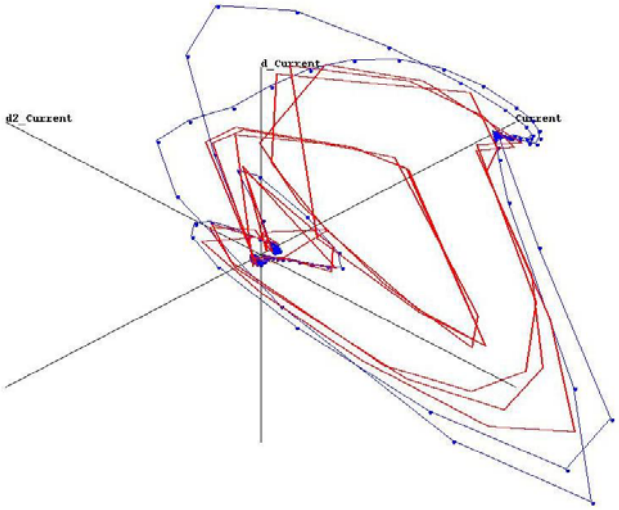

Figure 10. Path model of TEK 0, 2 and 3 with abnormal test path TEK 16.

Figures 11 and 12 shows the equivalent box model with k = 25 boxes. Figure 11 shows a normal test trace, TEK 1, which closely follows the model. Figure 12 shows the same abnormal test trace, TEK 16 as Figure 10, which again deviates from the model.

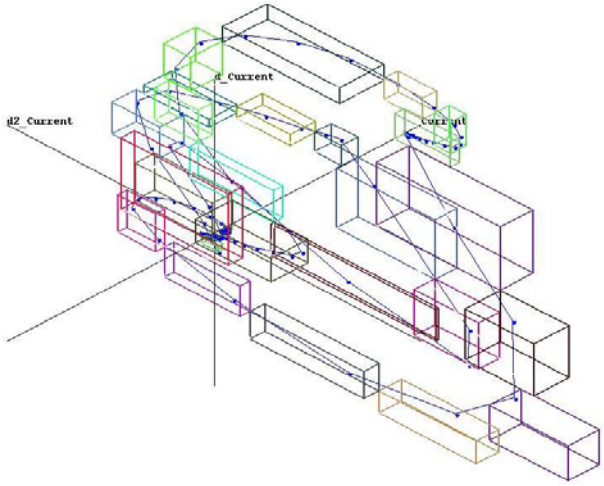

Fig. 11. Box model of TEK 0, 2, 3 with normal test path TEK 1.

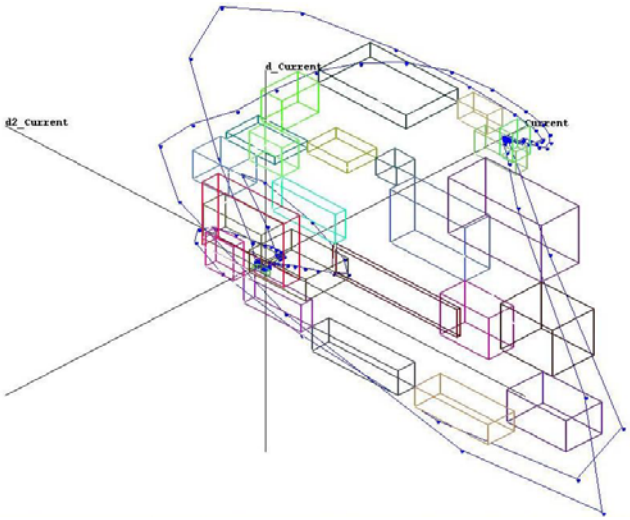

Fig. 12. Box model of TEK 0, 2, 3 with abnormal test path TEK 16.

Path and box modeling allow testing with or without a sequential constraint. The parameter R selects the number of path segments or boxes tested. Segments/boxes are tested in the following order: current, next, previous, second from next, or chosen at random. Thus, $R = 2$ constrains the test data to proceed forward, $R = 3$ allows backwards movement, $R = 5$ allows escape from local minima. $R = k$ tests all boxes or segments. Note that test time complexity is $O(Rpd)$ for path modeling and $O(Rd)$ for box modeling, where p is the number of training paths and d is the number of features.

5.3 TEK Results

For the TEK data set, we label TEK 0 through 3 as normal and TEK 10 through 17 as abnormal. Results are given in Table 1. Recall that an abnormal trace is detected if its score is higher than all of TEK 0-3. The column *Pct 1* gives the percent detected out of the 32 tests with one training trace (8 for each training trace). The column *Pct 2* gives the percent detected out of the 48 tests with two training traces for Gecko and path modeling, or 96 tests for box modeling. The number is higher for box modeling because the training order is significant. N/A means not applicable.

Table 1. TEK test results using 1 or 2 training traces.

Algorithm	**Pct 1**	**Pct 2**
Euclidean, raw, rising edge only	69	N/A
Euclidean, normalized, rising edge only	66	N/A
Euclidean, raw, edited full waveform	69	N/A
DTW, not normalized	41	N/A
DTW, normalized	44	N/A
Gecko + RIPPER	47	65
Path T=5ms, k=25, R=4 or k	100	100
Box, T=5ms, k=20, R=2,3,4,5,k	100	100

Table 2 lists the abnormal traces not detected by each algorithm when trained on one trace.

Table 2. TEK misses with one training trace

Method	TEK 0	TEK 1	TEK 2	TEK 3
Euc raw rise	15,17	14,15	11,14, 15,17	15,17
Euc norm rise	14,15,17	14,17	11,13,15	15,17
Euc raw edit	11,15,17	12,14, 15, 17	11,14,17	14,15, 17
DTW raw	15	15	10-15,17	10-15, 17
DTW norm	15	15	10-15	10-15
Gecko		16	10-17	10-17
Path				
Box				

Table 3 lists the abnormal traces not detected when Gecko is trained on two traces. The results only apply to Gecko because Euclidean distance and DTW allow only one training trace, and because path and box modeling do not miss any anomalies.

Table 3. Gecko misses with two training traces

Training	Missed detections among TEK 10-17
TEK 0, 1	TEK 16
TEK 0, 2	
TEK 0, 3	
TEK 1, 2	
TEK 1, 3	TEK 10-17
TEK 2, 3	TEK 10-17

To be fair, Gecko gives better results (83% detected) when trained on TEK 0, 1, or both, for which it was tuned. The other missed detections are due mainly to a very high false alarm score assigned to TEK 0 when trained on TEK 2 or 3. We did not attempt to tune Gecko for these other training sets.

Path and box modeling generally give good results on the TEK data using a filter time constant of T from about 4 to 10 ms, subsample interval $S \leq T$, $k \geq 25$ path segments or 20 boxes, whether testing with or without sequential constraints.

5.4 VT1 Results

As we mentioned, the VT1 set lacks baselines when used with only one training series, so it is not possible to test Euclidean and DTW on this data set. Instead, we test the generalization capabilities of Gecko, path and box modeling. To do this, we arbitrarily define the range 18 to 30 V, low temperature and no impedance as our normal set. There are 7 traces in this range, allowing us to train on a subset and use the remainder as a baseline. The 20 anomalies consist of low voltage, high voltage, high temperature and impedance.

In this experiment we train on V18, V22, V26 and V30. The order is irrelevant for Gecko and path modeling. For box modeling the training order is V22, V18, V22, V26, V30, following the recommendation of starting in the middle and

repeating the first trace (V22). An abnormal trace is counted as detected if the score is higher than all normal traces including the three normal traces not used in training, V20, V24 and V28. Results are shown in Table 4.

Table 4. VT1 test results.

Algorithm	Pct
Gecko + RIPPER	95% (misses V20i45)
Path, T=5ms, k=20, R=k	100%
Box, T=5ms, k=20, R=k	90% (misses V28T, V32T)

The missed detections by box modeling are higher voltage, high temperature anomalies such as V32T. These are hard to detect because the effects of high voltage and high temperature cancel out to produce a normal looking waveform.

The same range of path and box model parameters that work well on the TEK data also work well on the VT1 data, except that models with a sequential constraint ($R < k$) tend to do poorly.

6. CONCLUSIONS AND FUTURE WORK

We introduced two time series anomaly detection algorithms that that are accurate, not opaque, editable, score each data point (online), efficient, and generalizable from multiple time series. We first extended feature trajectory path models by introducing an efficient but approximate method of testing whether a data point lies between the trained paths. Then we eliminated the test time penalty for multiple paths by extending the MBR model to approximate the set of paths with a sequence of boxes in feature space. A box model is not quite as accurate as a path model, but is faster.

We evaluated our two methods (path and box modeling) against three existing methods (Euclidean, DTW, Gecko) with the shuttle valve data from NASA. For the TEK data, compared to existing algorithms, our methods detected more abnormal traces. For the VT1 data, our methods detected similar or more abnormal time series.

We do not pretend that path or box models are appropriate for all time series. Some work is required to tune parameters to a data set, but this is no different than most other anomaly detection systems. However these models have the nice property that they can be visualized, which should aid in verifying their correctness or modifying them manually to add domain specific knowledge. We did not directly test this capability, however.

In addition to the valve data, path and box modeling have been tested on spring-mass and battery charger simulations with good results. Future work will include online testing to identify anomalous points within a time series, comparison with other algorithms such as CDM, and testing on other data sets, such as arrhythmia detection in ECG traces.

7. ACKNOWLEDGMENTS

This work is supported by NASA (NAS10-02044). Bob Ferrell and Steve Santuro at NASA provided the valve data set. Walter Schefele at ICS developed the visualization software used in this project and provided screenshots for this paper. Stan Salvador and Chris Tanner at Florida Tech. provided test results for Gecko. Eamonn Keogh of UCR provided helpful comments on this paper.

8. REFERENCES

[1] "Inquiry Board Traces Ariane 5 Failure to Overflow Error", SIAM News, 29 (8), October 1996, http://www.siam.org/siamnews/general/ariane.htm

[2] Greg Clark, Alex Canizares, "Navigation Team Was Unfamiliar with Mars Climate Orbiter", space.com, Nov. 10, 1999, http://www.space.com/news/mco_report-b_991110.html

[3] W. Cohen, "Fast Effective Rule Induction", *Proc. ICML*, 1995.

[4] D. Dasgupta and S. Forrest, Artificial Immune Systems in Industrial Applications, *Proc. International Conference on Intelligent Processing and Manufacturing Material (IPMM)*, Honolulu, HI, 1999.

[5] B. Ferrell, S. Santuro, NASA Shuttle Valve Data. http://www.cs.fit.edu/~pkc/nasa/data/ (2005)

[6] Marios Hadjieleftheriou, George Kollios, Vassilis J. Tsotras, Dimitrios Gunopulos, "Efficient Indexing of Spatiotemporal Objects". EDBT 2002: 251-268.

[7] E. Keogh and S. Kasetty, On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration, Proc. *Proc. SIGKDD*, 2002.

[8] E. Keogh, S. Lonardi, C. A. Ratanamahatana, Towards Parameter-Free Data Mining, *Proc. ACM SIGKDD*, 2004.

[9] Richard J. Povinelli, Michael T. Johnson, Andrew C. Lindgren, Jinjin Ye, "Time Series Classification using Gaussian Mixture Models of Reconstructed Phase Spaces," IEEE Transactions on Knowledge and Data Engineering, 16 (6), June 2004, pp. 779-783.

[10] S. Salvador, P. Chan, "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space", *KDD Workshop on Mining Temporal and Sequential Data*, 2004.

[11] S. Salvador, P. Chan, J. Brodie, Learning States and Rules for Time Series Anomaly Detection, *Proc. 17th Intl. FLAIRS Conf*, pp. 300-305, 2004.

[12] Stan Salvador, "*Learning States for Detecting Anomalies in Time Series*", MS Thesis, Florida Tech, 2004.

[13] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos,. & E. Keogh, "Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures", *Proc. SIGKDD*, 2003

[14] James Watson, "Veteran software makes it to Titan", Personal Computer World, Jan. 26, 2005, http://www.pcw.co.uk/analysis/1160783

[15] A. Ypma, "Learning Methods for Machine Vibration Analysis and Health Monitoring", Dissertation, Delft University of Technology, Netherlands, 2001.

Anomalous Spatial Cluster Detection

Daniel B. Neill
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
neill@cs.cmu.edu

Andrew W. Moore
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
awm@cs.cmu.edu

ABSTRACT
We describe a general statistical and computational framework for the detection of anomalous spatial clusters, based on the *spatial scan statistic* [1]. Much of this material has been adapted from [2], to which we refer the reader for a more detailed discussion. We focus here on the purely spatial cluster detection task; for extensions to space-time cluster detection, the reader is referred to [3] and the references contained therein.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Apps—Data Mining

General Terms
Algorithms

Keywords
Cluster detection, anomaly detection, spatial scan statistics.

1. INTRODUCTION
Spatial cluster detection has two main goals: to identify the locations, shapes, and sizes of potentially anomalous spatial regions, and to determine whether each of these potential clusters is more likely to be a "true" cluster or simply a chance occurrence. In other words, we wish to answer the questions, is anything unexpected going on, and if so, where? This task can be broken down into two parts: first figuring out what we expect to see, and then determining which regions deviate significantly from our expectations. For example, in the application of disease surveillance, we examine the spatial distribution of disease cases (or some related quantity, such as the number of emergency department visits or over-the-counter drug sales of a specific type), and our goal is to determine whether any regions have sufficiently high case counts to be indicative of an emerging disease epidemic in that area. Thus we first infer the expected case count for each spatial location (e.g. zip code), typically based on historical data (though simpler approaches, such as assuming that number of cases is proportional to census population, can also be used). Then the next step is to determine which (if any) regions have significantly more cases than expected. One simple possibility would be to perform a separate statistical test for each spatial location under consideration, and report all locations that are significant at some level α. However, there are two main problems with this simple approach. First, we cannot use information about the spatial proximity of locations: for example, while a single zip code with count two standard deviations higher than expected might not be sufficiently interesting to trigger an alarm, we would probably be interested in a cluster of adjacent zip codes where each zip code's count is two standard deviations higher than expected. Second, *multiple hypothesis testing* is a problem: because we are performing a separate hypothesis test for each spatial location, where each hypothesis test has some fixed false positive rate α, the total number of false positives that we expect is $Y\alpha$, where Y is the total number of locations tested. For large Y, we are almost certain to get huge numbers of false alarms; alternatively, we would have to use a threshold α so low that the power of the test would be drastically reduced.

To deal with these problems, Kulldorff [1] proposed the *spatial scan statistic*. This method searches over a given set of spatial regions (where each region consists of a set of locations), finding those regions which are most likely to be generated under the "alternative hypothesis" of clustering rather than the "null hypothesis" of no clustering. A likelihood ratio test is used to compare these hypotheses, and randomization testing is used to compute the p-value of each detected region, correctly adjusting for multiple hypothesis testing. Thus, we can both identify potential clusters and determine whether each is significant. Our recent work on spatial scanning has two main emphases: first, to generalize the statistical framework to a larger class of underlying models, making the spatial scan applicable and useful for a wide variety of application domains; and second, to make these methods computationally tractable, even for massive real-world datasets. In this paper, we present an outline of our *generalized spatial scan* framework. We then consider each of the steps in more detail, giving some idea of the relevant decisions that need to be made when applying the spatial scan to a new domain. In [8], we present our experiences in one such domain (outbreak detection using over-the-counter drug sales data); here we discuss the method more generally, considering those issues which apply to any domain.

2. THE GENERALIZED SPATIAL SCAN
Our *generalized spatial scan* framework consists of the following six steps:

1) Obtain data for a set of spatial locations s_i.

2) Choose a set of spatial regions to search over, where each spatial region S consists of a set of spatial locations s_i.

3) Choose models of the data under H_0 (the null hypothesis of no clusters) and $H_1(S)$ (the alternative hypothesis assuming a cluster in region S).

4) Derive a "score function" $F(S)$ based on $H_1(S)$ and H_0.

5) Find the "most interesting" regions, i.e. those regions S with the highest values of $F(S)$.

6) Determine whether each of these regions is "interesting," either by performing significance testing or calculating posterior probabilities.

We now consider each step of this framework in detail.

1) Obtain data for a set of spatial locations s_i.

For each spatial location s_i, we are given a *count* c_i and optionally a *baseline* b_i. For example, each s_i may represent a zip code, with location (latitude and longitude) assumed to be at the centroid of the zip code; c_i may represent the number of respiratory disease cases in that zip code, and b_i may represent the at-risk population. In any case, the goal of our method is to find regions where the counts are higher than expected, given the baselines. Two typical approaches are the *population-based* method, where b_i represents the underlying population of location s_i, and we expect each count to be proportional to its population under the null hypothesis, and the *expectation-based* method, where b_i represents the expected count of location s_i, and thus we expect each count to be equal to its expectation under the null. In either case, the b_i for each location may either be given (e.g. census population) or may be inferred from the time series of past counts. For example, one simple expectation-based approach would be to estimate today's expected count in a zip code by the mean daily count in that zip code over the past d days. For many datasets, more complicated methods of time series analysis should be used to infer baselines; for example, in the over-the-counter drug sales data, we must account for both seasonal and day-of-week effects. We consider various methods of inferring baselines in [3].

2) Choose a set of spatial regions to search over, where each spatial region S consists of a set of spatial locations s_i.

We want to choose a set of regions that corresponds well with the shape and size of the clusters we are interested in detecting. In general, the set of regions should cover the entire space under consideration (otherwise we will have no power to detect clusters in non-covered areas) and adjacent regions should overlap (otherwise we will have reduced power to detect clusters that lie partly in one region and partly in another). We typically consider the set of all regions of some fixed shape (e.g. circle, ellipse, rectangle) and varying size; what shape to choose depends on both statistical and computational considerations. If we search too few regions, we will have reduced power to detect clusters that do not closely match any of the regions searched; for example, if we search over square or circular regions, we will have low power to detect highly elongated clusters. On the other hand, if we search too many regions, our power to detect any particular subset of these regions is reduced because of multiple hypothesis testing. Additionally, the runtime of the algorithm is proportional to the number of regions searched, and thus choosing too large a set of regions will make the method computationally infeasible.

Our typical approach in epidemiological domains is to map the spatial locations to a grid, and search over the set of all rectangular regions on the grid. Additionally, non-axis-aligned rectangles can be detected by searching over multiple rotations of the data. The two main advantages of this approach are its ability to detect elongated clusters (this is important in epidemiology because disease clusters may be elongated due to wind or water dispersion of pathogens) and also its computational efficiency. Use of a grid structure allows us to evaluate any rectangular region in constant time, independent of the size of the region, using the well-known "cumulative counts" trick [4]. Additionally, we can gain huge computational speedups by applying the "fast spatial scan" algorithm [4-6], as we discuss below.

3) Choose models of the data under H_0 (the null hypothesis of no clusters) and $H_1(S)$ (the alternative hypothesis assuming a cluster in region S).

4) Derive a "score function" $F(S)$ based on $H_1(S)$ and H_0.

These are perhaps the most difficult steps in our method, as we must choose models which are both efficiently computable and relevant to the application domain under consideration. For our models to be *efficiently computable*, the score function $F(S)$ should be computable as a function of some additive sufficient statistics of the region S being considered (typically these statistics are the total count of the region, $C(S) = \sum_S c_i$, and the total baseline of the region, $B(S) = \sum_S b_i$). If this is not the case, the model may still be useful for small datasets, but will not scale well to larger sources of data. For our models to be *relevant*, any simplifying assumptions that we make must not reduce our power to distinguish between the "cluster" and "no cluster" cases, to too great an extent. Of course, any efficiently computable model is very unlikely to capture all of the complexity of the real data, and these unmodeled effects may have either small or large impacts on detection performance. Thus we typically use an iterative design process, beginning with very simple models, and examining their detection power (ability to distinguish between "cluster" and "no cluster") and calibration (number of false positives reported in day-to-day use). If a model has high detection power but poor calibration, then we have a choice between increasing model complexity and artificially recalibrating the model (i.e. based on the empirical distribution of scores); however, if detection power is low, then we have no choice but to figure out which unmodeled effects are harming performance, and deal with these effects one by one. Some such effects (e.g. missing data) can be dealt with by pre-processing, and others (e.g. clusters caused by single locations) can be dealt with by post-processing (filtering the set of discovered regions to remove those caused by known effects), while others must actually be included in the model itself. In [8], we discuss several of these effects present in the over-the-counter sales data, and how we have dealt with each; here we focus on the general framework and then present two simple and efficiently computable models.

The most common statistical framework for the spatial scan is a frequentist, hypothesis testing approach. In this approach, assuming that the null hypothesis and each alternative

hypothesis are point hypotheses (with no free parameters), we can use the likelihood ratio $F(S) = \frac{\Pr(Data \mid H_1(S))}{\Pr(Data \mid H_0)}$ as our test statistic. A more interesting question is what to do when each hypothesis has some parameter space Θ: let $\theta_1(S) \in \Theta_1(S)$ denote parameters for the alternative hypothesis $H_1(S)$, and let $\theta_0 \in \Theta_0$ denote parameters for the null hypothesis H_0. There are two possible answers to this question. In the more typical, *maximum likelihood* framework, we use the estimates of each set of parameters that maximize the likelihood of the data: $F(S) = \frac{\max_{\theta_1(S) \in \Theta_1(S)} \Pr(Data \mid H_1(S), \theta_1(S))}{\max_{\theta_0 \in \Theta_0} \Pr(Data \mid H_0, \theta_0)}$. In many cases, such as in Kulldorff's statistic [1], this will lead to an *individually most powerful* statistical test under the given model assumptions. We then perform randomization testing using the maximum likelihood estimates of the parameters under the null hypothesis, as discussed below. In the *marginal likelihood* framework, on the other hand, we instead average over the possible values of each parameter:

$$F(S) = \frac{\int_{\theta_1(S) \in \Theta_1(S)} \Pr(Data \mid H_1(S), \theta_1(S)) \Pr(\theta_1(S))}{\int_{\theta_0 \in \Theta_0} \Pr(Data \mid H_0, \theta_0) \Pr(\theta_0)}$$

This, however, makes randomization testing very difficult. A third alternative (discussed in detail in [7]) is a Bayesian approach, in which we use the marginal likelihood framework to compute the likelihood of the data under each hypothesis, then combine these likelihoods with the prior probabilities of an cluster in each region S. Thus our test statistic is the posterior probability of a cluster in each region: $F(S) = \frac{\Pr(Data \mid H_1(S)) \Pr(H_1(S))}{\Pr(Data)}$. The marginal likelihood of the data is typically difficult to compute, but in [7], we present an efficiently computable Bayesian statistic using Poisson counts and conjugate Gamma priors. Here we instead focus on the simpler, maximum likelihood frequentist approach, and give an example of how new scan statistics can be derived.

Let us first consider the expectation-based scan statistic discussed above, under the simplifying assumption that counts are independently Poisson distributed (i.e. counts are not spatially correlated, and neither overdispersed nor underdispersed). In this case, we are given the *baseline* (or expected count) b_i and the observed count c_i for each spatial location s_i, and our goal is to determine if any spatial region S has counts significantly greater than baselines. Furthermore, let us consider a simple cluster model, where we assume a uniform multiplicative increase in counts inside the cluster (the amount of increase is unknown). Thus we test the null hypothesis H_0 against the set of alternative hypotheses $H_1(S)$, where:

H_0: $c_i \sim \text{Poisson}(b_i)$ for all spatial locations s_i.

$H_1(S)$: $c_i \sim \text{Poisson}(qb_i)$ for all spatial locations s_i in S, and $c_i \sim \text{Poisson}(b_i)$ for all spatial locations s_i outside S, for some constant $q > 1$.

Here, the alternative hypothesis $H_1(S)$ has one parameter, q (the *relative risk* in region S), and the null hypothesis H_0 has no parameters. Computing the likelihood ratio, and using the maximum likelihood estimate for our parameter q, we obtain the following expression:

$$F(S) = \frac{\max_{q>1} \prod_{s_i \in S} \Pr(c_i \sim \text{Poisson}(qb_i)) \prod_{s_i \notin S} \Pr(c_i \sim \text{Poisson}(b_i))}{\prod_{s_i} \Pr(c_i \sim \text{Poisson}(b_i))}$$

We find that the value of q that maximizes the numerator is $q = \max(1, C / B)$, where C and B are the total count $\sum c_i$ and total baseline $\sum b_i$ of region S respectively. Plugging in this value of q, and working through some algebra, we obtain:

$F(S) = \left(\frac{C}{B}\right)^C \exp(B - C)$, if $C > B$, and $F(S) = 1$ otherwise.

Because $F(S)$ is a function only of the sufficient statistics $C(S)$ and $B(S)$, this function is efficiently computable: we can calculate the score of any region S by first calculating the aggregate count and baseline (in constant time, as noted above) and then applying the function F.

Kulldorff's spatial scan statistic [1] is a population-based method commonly used in disease surveillance, which also makes the simplifying assumption of independent, Poisson distributed counts. However, this statistic assumes that counts (i.e. number of disease cases) are distributed as $c_i \sim \text{Poisson}(qb_i)$, where b_i is the (known) census population of s_i and q is the (unknown) underlying disease rate. We then attempt to discover spatial regions where the underlying disease rate q is significantly higher inside the region than outside. Thus we wish to test the null hypothesis H_0 ("the underlying disease rate is spatially uniform") against the set of alternative hypotheses $H_1(S)$: "the underlying disease rate is higher inside region S than outside S." More precisely, we have:

H_0: $c_i \sim \text{Poisson}(q_{all} b_i)$ for all locations s_i, for some constant q_{all}.

$H_1(S)$: $c_i \sim \text{Poisson}(q_{in} b_i)$ for all locations s_i in S, and $c_i \sim \text{Poisson}(q_{out} b_i)$ for all locations s_i outside S, for some constants $q_{in} > q_{out}$.

In this case, the alternative hypothesis has two free parameters (q_{in} and q_{out}) and the null hypothesis has one free parameter (q_{all}). Computing the likelihood ratio, and using maximum likelihood parameter estimates, we obtain:

$$F(S) = \frac{\max_{q_{in} > q_{out}} \prod_{s_i \in S} \Pr(c_i \sim \text{Poisson}(q_{in} b_i)) \prod_{s_i \notin S} \Pr(c_i \sim \text{Poisson}(q_{out} b_i))}{\max_{q_{all}} \prod_{s_i} \Pr(c_i \sim \text{Poisson}(q_{all} b_i))}$$

We can compute the maximum likelihood estimates $q_{in} = C_{in} / B_{in}$, $q_{out} = C_{out} / B_{out}$, and $q_{all} = C_{all} / B_{all}$, where "in", "out", and "all" represent the aggregates of counts and baselines for s_i inside region S, for s_i outside region S, and for all s_i respectively. Plugging in these values and performing some algebra, we obtain: $F(S) = \left(\frac{C_{in}}{B_{in}}\right)^{C_{in}} \left(\frac{C_{out}}{B_{out}}\right)^{C_{out}} \left(\frac{C_{all}}{B_{all}}\right)^{-C_{all}}$ if $\frac{C_{in}}{B_{in}} > \frac{C_{out}}{B_{out}}$, and $F(S) = 1$ otherwise. Again, the score function can be computed efficiently from the sufficient statistics of region S.

We have also used this general framework to derive scan statistics assuming that counts c_i are generated from Normal distributions with mean (i.e. expected count) μ_i and variance σ_i^2; these statistics are useful if counts might be overdispersed or underdispersed. In this case, the score function is still

efficiently computable, as a function of the sufficient statistics $B = \sum \frac{\mu_i}{\sigma_i^2} \mu_i$ and $C = \sum \frac{\mu_i}{\sigma_i^2} c_i$. Many other likelihood ratio scan statistics are possible, including models with simultaneous attacks in multiple regions and models with spatially varying (rather than uniform) rates. We believe that some of these more complex model specifications may have more power to detect relevant and interesting clusters, while excluding those potential clusters which are not relevant to the application domain under consideration.

5) Find the "most interesting" regions, i.e. those regions *S* with the highest values of *F(S)*.

Once we have decided on a set of regions *S* to search, and derived a score function *F(S)*, the "most interesting" regions are those that maximize *F(S)*. In the frequentist spatial scan framework, these are the most significant spatial regions; in the Bayesian framework, these are the regions with highest posterior probabilities. The simplest method of finding the most interesting regions is to compute the score function *F(S)* for every region. An alternative to this naïve approach is to use the *fast spatial scan* algorithms of [4-6], which allow us to reduce the number of regions searched, but without losing any accuracy. The idea is that, since we only care about the most significant regions, i.e. those with the highest scores *F(S)*, we do not need to search a region *S* if we can prove that it will not have a high score. Thus we start by examining large regions *S*, and if we can show that none of the smaller regions contained in *S* can have high scores, we do not need to actually search each of these regions. Thus, we can achieve the same result as if we had searched all possible regions, but by only searching a small fraction of these. Further speedups are gained by the use of multiresolution data structures, which allow us to efficiently move between searching at coarse and fine resolutions; we discuss these methods in detail in [4-6].

6) Determine whether each of these regions is "interesting," either by performing significance testing or calculating posterior probabilities.

For the frequentist approach, once we have found the highest scoring region S^* and its score $F^* = F(S^*)$, we must still determine the statistical significance of this region by randomization testing. To do so, we randomly create a large number R of replica grids by sampling under the null hypothesis, given our maximum likelihood parameter estimates for the null. For example, for the expectation-based approach given above, we generate counts independently from $c_i \sim \text{Poisson}(b_i)$, and for the population-based approach given above, we generate counts independently from $c_i \sim \text{Poisson}(q_{all} b_i)$, using the maximum likelihood estimate $q_{all} = C_{all} / B_{all}$. We then find the highest scoring region and its score for each replica grid: the *p*-value of S^* is $\frac{R_{beat} + 1}{R + 1}$, where R_{beat} is the number of replicas with F^* higher than the original grid. If this *p*-value is less than some threshold (e.g. 0.05), we can conclude that the discovered region is unlikely to have occurred by chance, and is thus a significant spatial cluster; we can then examine secondary clusters. Otherwise, no significant clusters exist.

For the Bayesian approach, on the other hand, no randomization testing is necessary. Instead, we can compute the posterior probability of each potential cluster by dividing its score $\Pr(Data \mid H_1(S)) \Pr(H_1(S))$ by the total probability $\Pr(Data) = \Pr(Data \mid H_0) \Pr(H_0) + \sum_S \Pr(Data \mid H_1(S)) \Pr(H_1(S))$. We can then report all clusters with posterior probability greater than some predetermined threshold, or simply "sound the alarm" if the total posterior probability of all clusters *S* is sufficiently high. Because we do not need to perform randomization testing in the Bayesian method, we need only to search over all regions for the original grid, rather than the original grid and a large number (typically $R = 1000$) of replicas. Thus the Bayesian approach is approximately 1000x faster than the (naïve) frequentist approach, as we show empirically in [7]. However, we can apply the fast spatial scan described above to achieve similar speedups for the frequentist approach: in this case, we still have to search over all replica grids, but can do a much faster search on each. As a result, the fast frequentist approach is faster than the Bayesian approach for sufficiently large grid sizes ($N > 256$) but slower for smaller grids. Either method can search a 256×256 grid, and calculate significance (*p*-values or posteriors respectively) in 10-12 hours, as compared to months for the standard (naïve frequentist) approach. Thus we now have two ways to make the spatial scan computationally feasible for large datasets: to apply the fast spatial scan of [4-6] or to use the Bayesian framework of [7]. For even larger grid sizes, it may be possible to extend the fast spatial scan to the Bayesian framework: this would give us the best of both worlds, searching only a single grid, and using a fast algorithm to do so. We are currently investigating this potentially useful synthesis.

3. REFERENCES

[1] M. Kulldorff. A spatial scan statistic. *Communications in Statistics: Theory and Methods* **26**(6), 1481-1496, 1997.

[2] D.B. Neill and A.W. Moore. Methods for detection of spatial and spatio-temporal clusters. In M. Wagner et al., eds., *Handbook of Biosurveillance*, 2005.

[3] D.B. Neill, A.W. Moore, M. Sabhnani, and K. Daniel. Detection of emerging space-time clusters. Accepted to *11th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2005.

[4] D.B. Neill and A.W. Moore. Rapid detection of significant spatial clusters. *Proceedings of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 256-265, 2004.

[5] D.B. Neill, A.W. Moore, F. Pereira, and T. Mitchell. Detecting significant multidimensional spatial clusters. *Advances in Neural Information Processing Systems* **17**, 969-976, 2005.

[6] D.B. Neill and A.W. Moore. A fast multi-resolution method for detection of significant spatial disease clusters. *Advances in Neural Information Processing Systems* **16**, 651-658, 2004.

[7] D.B. Neill, A.W. Moore, and G.F. Cooper. A Bayesian spatial scan statistic. Submitted for publication.

[8] M.R. Sabhnani, D.B. Neill, A.W. Moore, F.-C. Tsui, M.M. Wagner, and J.U. Espino. Detecting anomalous patterns in pharmacy retail data. *KDD Workshop on Data Mining Methods for Anomaly Detection*, 2005.

An Empirical Comparison of Outlier Detection Algorithms

Matthew Eric Otey, Srinivasan Parthasarathy, and Amol Ghoting
Department of Computer Science and Engineering
The Ohio State University
Contact: srini@cse.ohio-state.edu

Abstract

In recent years, researchers have proposed many different techniques for detecting outliers and other anomalies in data sets. In this paper we wish to examine a subset of these techniques, those that have been designed to discover outliers quickly. The algorithms in question are ORCA, LOADED, *and* RELOADED. *We have performed an empirical evaluation of these algorithms, and here present our results as guide to their strengths and weaknesses.*

1. Introduction

A common problem in data mining is that of automatically finding outliers in a database, since they can be indicative of bad data or malicious behavior. Examples of bad data include skewed data values resulting from measurement error, or erroneous values resulting from data entry mistakes. A common example of data indicating malicious behavior occurs in the field of network traffic analysis, where anomalous IP packets may indicate either a possible intrusion or attack, or a failure in the network [21]. Efficient detection of such outliers reduces the risk of making poor decisions based on erroneous data, and aids in identifying, preventing, and repairing the effects of malicious or faulty behavior. Additionally, many data mining and machine learning algorithms, and techniques for statistical analysis may not work well in the presence of outliers. Outliers may introduce skew or complexity into models of the data, which may make it difficult, if not impossible, to fit an accurate model to the data in a computationally feasible manner. Accurate, efficient removal of outliers may greatly enhance the performance of statistical and data mining algorithms and techniques [5]. As can be seen, different domains have different reasons for discovering outliers: They may be noise that we want to remove, since they obscure the true patterns we wish to discover, or they may be the very things in the data that we wish to discover. As has been said before, "One person's noise is another person's signal" [14].

Effective outlier detection requires the construction of a model that accurately represents the data. Over the years, a large number of techniques have been developed for building such models for outlier and anomaly detection. However, real-world data sets and environments present a range of difficulties that limit the effectiveness of these techniques. Among these problems is the fact that the data sets may be dynamic, the fact that the data may be distributed over various sites, and the fact that the data sets may contain a mixture of attribute types (i.e. continuous and categorical attributes).

In this paper we empirically compare and contrast a set of outlier detection algorithms. This set includes ORCA and variations on the LOADED algorithm. These algorithms are designed to minimize execution times, by means of minimizing the number of passes of the data that must be made. Each addresses one or more of the problems mentioned above in a different manner, and each has their own strengths and weaknesses with respect to execution time, memory usage, and detection quality.

2. Related Work

There are several approaches to outlier detection. One approach is that of statistical model-based outlier detection, where the data is assumed to follow a parametric (typically univariate) distribution [1]. Such approaches do not work well in even moderately high-dimensional (multivariate) spaces, and finding the right model is often a difficult task in its own right. Simplified probabilistic models suffer from a high false positive rate [16, 17]. Also, methods based on computational geometry [10] do not scale well as the number of dimensions increase. To overcome these limitations, researchers have turned to various non-parametric approaches including distance-based approaches [2, 11], clustering-based approaches [7, 21], and density-based approaches [4, 20]. Here we consider these methods in more detail.

An approach for discovering outliers using distance metrics was first presented by Knorr *et al.* [12, 13, 11]. They define a point to be a *distance outlier* if at least a user-defined fraction of the points in the data set are further away than some user-defined minimum distance from that point. In their experiments, they primarily focus on data sets containing only continuous attributes. Related to distance-based methods are methods that cluster data and find outliers as part of the process of clustering [9]. Points that do not cluster well are labeled as outliers. This is the approach used by the ADMIT intrusion detection system [21].

Recently, density-based approaches to outlier detection have been proposed [4]. In this approach, a local outlier factor

(LOF) is computed for each point. The LOF of a point is based on the ratios of the local density of the area around the point and the local densities of its neighbors. The size of a neighborhood of a point is determined by the area containing a user-supplied minimum number of points ($MinPts$). A similar technique called LOCI (Local Correlation Integral) is presented in [20]. LOCI addresses the difficulty of choosing values for $MinPts$ in the LOF-based technique by using statistical values derived from the data itself. Both the LOF- and LOCI-based approaches do not scale well with a large number of attributes and data points, and so are not considered in this evaluation.

A comparison of various anomaly detection schemes is presented in [15]. Its focus is on how well different schemes perform with respect to detecting network intrusions. The authors used the 1998 DARPA network connection data set to perform their evaluation, which is the basis of the KDDCup 1999 data set used in our experiments [8]. They found detection rates ranging from a low of 52.63% for a Mahalanobis distance-based approach, to a high of 84.2% for an approach using support vector machines.

3. Algorithms

In this section we give a brief overview of each of the algorithms we evaluate. For a more in-depth discussion, the reader is referred to the papers in which the algorithms were originally proposed [2, 6, 18, 19].

3.1 ORCA

Most distance-based methods for detecting outliers take time that is at least quadratic in the number of points in the data set, which may be unacceptable if the data set is very large or dynamic. Bay and Schwabacher [2] present a method called ORCA for discovering outliers in near linear time. The central idea is to perform pruning by keeping a monotonically decreasing score for each point in the data set. If the score falls below a certain threshold, then further processing on the data point is not necessary. In the worst case (when there are no outliers), the algorithm still takes quadratic time, but in practice the algorithm runs very close to linear time. Such an approach assumes that the data set is randomized, and randomization is performed on disk prior to running the algorithm. ORCA handles mixed-attribute data sets by using the Euclidean distance for the continuous attributes and the Hamming distance for the categorical attributes.

3.2 LOADED

The LOADED (Link-based Outlier and Anomaly Detection in Evolving Data sets) algorithm was first presented in [6]. It is designed explicitly for dynamic data with heterogeneous attributes. For dynamic data sets, it can process the data in one pass, and for static data sets it can make a second pass for increased accuracy.

The original version of LOADED presented in [6] is a centralized algorithm for detecting outliers in dynamic mixed-attribute data. The central data structure used to model the data is an augmented lattice of all itemsets formed from the categorical attributes of the data. Each node in the lattice is augmented with the support count of the corresponding itemset, and the correlation matrix computed from the continuous attributes of all data points in the data set containing that itemset. Such a data structure ensures that the dependencies between all attributes, regardless of type, can be modeled. Each data point is assigned an anomaly score based on the support of all its itemsets and how well the continuous attributes agree with the relevant correlation matrices. The basic algorithm makes a single pass of the data, incrementally updating the lattice for each data point processed. The algorithm is also able to make a second pass of the data, which allows for better detection rates. Finally, it is also possible to constrain the size of the lattice to conserve memory, at a small cost to accuracy.

The LOADED algorithm has also been extended to handle distributed data sets [18]. The algorithm is essentially the same as the centralized version of LOADED presented above, but by default it performs two passes of the data set. In the first pass, each site constructs a local augmented lattice from its local portion of the data set. The lattices are then exchanged and combined to form a global augmented lattice which is used to detect outliers at each site during the second pass. There is also a variation that allows the global model to be computed incrementally, which allows for a single-pass approach. To avoid repeatedly exchanging lattices, only local outliers are exchanged between nodes. If all nodes agree that a given point is an outlier, then it is marked as a global outlier.

3.3 RELOADED

The RELOADED [19] algorithm is designed to address the memory usage problems of LOADED. As it stands, LOADED's space complexity is exponential in the number of categorical attributes, since it must maintain an augmented itemset lattice. RELOADED dispenses with the lattice and uses set of classifiers to model dependencies between the categorical attributes. Each classifier is trained to predict the value of a given categorical attribute based on the values of the remaining categorical and continuous attributes. Incorrect predictions of a categorical attribute for a data point increase the point's anomaly score. To further model dependencies between the categorical and continuous attributes, a covariance matrix is maintained for each unique value of each categorical attribute. Each covariance matrix is computed from those data points in which the given categorical attribute has the given value. A data point's anomaly score is also based on how well the data point's continuous attributes adhere to the relevant covariance matrices.

4. Evaluation

4.1 Setup

For evaluating the centralized algorithms, we use a machine with a 2.8 GHz Pentium IV processor and 1.5 GB of memory, running Mandrake Linux 10.1. Our implementations are in C++ and are compiled using gcc with O2 optimizations. We evaluate the distributed version of LOADED using an eight-node cluster, where each node has dual 1 GHz Pentium III

processors and 1 GB of memory, running Red Hat Linux 7.2. Our implementation uses MPI for message passing. Unless otherwise noted, we use the default values of the parameters for each algorithm (see [6, 18, 19, 2]). We use the following data sets.

4.1.1 KDDCup 1999 Intrusion Detection Data

The 1999 KDDCup data set [8] contains a set of records that represent connections to a military computer network where there have been multiple intrusions and attacks. This data set was obtained from the UCI KDD archive [3]. The training data set has 4,898,430 data instances with 32 continuous attributes and 9 categorical attributes. The testing data set is smaller and contains several new intrusions that were not present in the training data set. Since these data sets have an unrealistic number of attacks, we preprocess them such that intrusions constitute 2% of the data set, and the proportions of different attacks is maintained. Since packets tend to occur in bursts for some intrusions, intrusion instances are not randomly inserted into the data, but occur in bursts that are randomly distributed in the data set. The processed training data set contains 983,561 instances with 10,710 attack instances, while the processed testing data set contains 61,917 instances with 1,314 attack instances.

4.1.2 Adult Data

The Adult data set [3], contains 48,842 data instances with 6 continuous and 8 categorical attributes. Since the algorithms we test differ in their abilities to handle missing data, we removed all records containing missing data, leaving 32,561 records. The data was extracted from the US Census Bureau's Income data set. Each record contains an individual's demographic attributes together with a class label indicating whether person made more or less than 50,000 dollars per year.

4.1.3 Synthetic Data

Since there are very few publicly available large mixed-attribute data sets, we wrote a synthetic data set generator to produce data to compare performance with existing algorithms, and with varying data set characteristics. The generator can produce data sets with a user-supplied number of continuous attributes and categorical attributes. The data points are generated according to a user-supplied multi-modal distribution. The exact details can be found in [18]. To create actual data sets for our experiments, we first generate a set of normal points from one distribution, and then separately generate a much smaller set of outliers from another distribution The two sets are then randomly mixed to produce the final data set. However, the synthetic data set is designed for benchmarking the memory and execution time scalability of the detection algorithms, and so it is not fair to use it to make detection quality comparisons. In our experiments, we consider a synthetic data set containing a 1% mixture of outliers.

4.2 Detection Quality

Our first set of experiments compares the detection rates of the various algorithms. In particular, we examine the performance of the two-pass versions of RELOADED and LOADED, and ORCA. Both the two-pass centralized and distributed versions of LOADED have the same accuracies, so we only present the former here. Also, since ORCA is designed to find the top k outliers, we set k equal to the number of outliers in the data sets.

Detection rates for all the different algorithms are reported in Table 1 (Note that "n/a" indicates that the attack was not present in that particular data set). Since the intrusion packets tend to occur in bursts in our data set, we mark an intrusion as detected if at least one instance in a burst is flagged as an outlier. This is realistic, since a network administrator needs to be alerted only once that an intrusion is underway. Consequently, the detection (true positive) rates in the table are in terms of the number of intrusion bursts detected. We report false positive rates in terms of the number of normal packets marked as outliers (in the case of ORCA, this is the percentage of normal packets that are marked as top-k outliers). *Overall, the detection rates for* LOADED *are very good, better than those of* RELOADED *and much better than those of* ORCA. LOADED has a false positive rate of 0.35%, which is extremely good for anomaly detection schemes, especially considering its high detection rates for many of the intrusions. ORCA has a false positive rate of 0.43%, but this is not as significant considering its low detection rates. RELOADED has detection rates comparable to LOADED on many intrusions, and does very well on a handful of intrusions (e.g. IP sweep and smurf) on which both LOADED and ORCA do poorly. RELOADED has higher false positive rates of 1.5% for the testing data set and 3.6% for the training data set, which is to be expected since it builds a less intricate model in order to save on memory. Finally, we note that as the single-pass distributed version of LOADED scales to more nodes, its detection rate decreases slightly, as can be seen in the experimental results presented in [18]. This is can be attributed to data points that are flagged as local normals when they are in fact global outliers.

4.3 Memory Usage

Algorithm	KDDCup (Test)	KDDCup (Train)	Adult
RELOADED	623	852	291
LOADED	49,328	595,280	58,316
ORCA	599	n/a	390

Table 2. Peak heap usage in kilobytes.

We first compare the memory usage of ORCA, LOADED, and RELOADED when they are run on the KDDCup and Adult data sets. For RELOADED and LOADED we use single-pass approaches, as the amount of memory used does not vary with the number of passes. For LOADED, we use just 4 lattice levels, and we set ORCA to find the top 1,314 outliers in the KDDCup testing data set, and the top 30 outliers in the Adult data set. We also note that ORCA cannot finish processing the KDDCup training data set in a reasonable amount of time. We measure memory usage by looking at the peak heap usage measured in kilobytes. The results can be seen in Table 2. Both RELOADED and ORCA consume less than one megabyte of memory, while LOADED uses *two to three orders of magnitude more memory*, even when we constrain the lattice to 4 levels. RELOADED manages to keep its memory usage low by using a compact model of the data, while ORCA processes the data in blocks.

Unlike ORCA, LOADED and RELOADED have greater than

Attack	KDDCup Testing			KDDCup Training	
	RELOADED	LOADED	ORCA	RELOADED	LOADED
Apache2	100%	100%	0%	n/a	n/a
Back	n/a	n/a	n/a	0%	98%
Buffer Overflow	72%	90%	100%	0%	91%
FTP Write	n/a	n/a	n/a	0%	33%
Guess Password	50%	100%	0%	34%	100%
Imap	n/a	n/a	n/a	50%	100%
IP Sweep	100%	28%	0%	90%	37%
Land	n/a	n/a	n/a	100%	100%
Load Module	n/a	n/a	n/a	0%	100%
Multihop	63%	70%	75%	0%	94%
Named	67%	100%	40%	n/a	n/a
Neptune	n/a	n/a	n/a	100%	98%
Nmap	n/a	n/a	n/a	64%	91%
Perl	n/a	n/a	n/a	0%	100%
Phf	80%	20%	100%	0%	0%
Pod	96%	100%	18%	81%	54%
Port Sweep	100%	100%	3%	93%	100%
Root Kit	n/a	n/a	n/a	0%	33%
Saint	100%	100%	1%	n/a	n/a
Satan	n/a	n/a	n/a%	80%	72%
Sendmail	17%	50%	50%	n/a%	n/a%
Smurf	98%	21%	0%	78%	22%
Snmpgetattack	0%	52%	0%	n/a	n/a
Spy	n/a	n/a	n/a	0%	100%
Teardrop	n/a	n/a	n/a	40%	30%
Udpstorm	0%	0%	0%	n/a	n/a
Warez Client	n/a	n/a	n/a	4%	43%
Warez Master	n/a	n/a	n/a	0%	25%
Xlock	50%	50%	66%	n/a	n/a
Xsnoop	100%	100%	100%	n/a	n/a

Table 1. Detection rates for the KDDCup data sets.

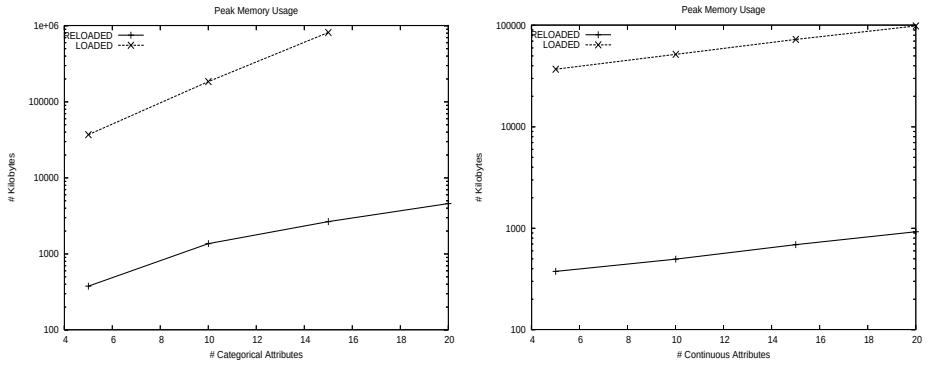

Figure 1. Log of peak memory usage with increasing numbers of (a) categorical and (b) continuous attributes.

linear space complexity with respect to the number of categorical and continuous attributes, and so we empirically test how they scale in terms of peak memory usage as the number and types of attributes vary. Again, in all cases, we set LOADED to use only 4 lattice levels. In Figure 1(a) we plot the log of peak memory usage versus increasing numbers of categorical attributes, while setting the number of continuous attributes equal to 5. As is evident from the graph, the memory requirements of LOADED are very large and grow exponentially as the number of categorical attributes increase, while those of RELOADED grow much more slowly. Note that we cannot run LOADED on data sets with more than 15 categorical attributes, as our machines do not have sufficient memory. In Figure 1(b) we set the number of categorical attributes equal to 5 and then vary the number of continuous attributes. The peak memory requirements of both RELOADED and LOADED increase at about the same rate, which is expected since they both use space that is quadratic in the number of continuous attributes. Note that even for 5 categorical attributes, LOADED requires significantly more memory to maintain the itemset lattice.

4.4 Execution Time

In our next set of experiments we compare the execution times of the ORCA, LOADED and RELOADED algorithms. Again, for LOADED we only use 4 lattice levels. Also, we use the single-pass versions of both RELOADED and LOADED. In our first experiment, we measure the execution times on the KDDCup testing data set. ORCA takes 303 seconds to complete, compared with 109 seconds for LOADED, and 47 seconds for RELOADED. In our next experiment, we examine how execution time scales with the number of data points processed. We use synthetic data with 10 categorical and 5 continuous attributes. The results can be seen in Figure 2(a). Note that we use a log scale on both axes. For small data sets, ORCA out-performs both LOADED and RELOADED, but since it does not scale linearly, this advantage is lost for larger data sets. As we expect, both RELOADED's and LOADED's execution times scale linearly with the number of points, though LOADED does not scale as well as RELOADED.

While ORCA's time complexity is linear with respect to the number of categorical and continuous attributes, LOADED's and RELOADED's complexity is not, and so in our next two experiments we compare how the execution of times of both LOADED and RELOADED scale for data sets with varying numbers of categorical and continuous attributes. In our first experiment, we set the number of continuous attributes equal to 5 and vary the number of categorical attributes between 1 and 15. The results can be seen in Figure 2(b). Note that we only use a log scale on the execution time axis. Though we limit LOADED to using only 4 lattice levels, its execution time still increases exponentially with the number of categorical attributes, while RELOADED increases quadratically. In our second experiment we examine the scalability of both algorithms with respect to the number of continuous attributes. In this experiment we set the number of categorical attributes equal to 5 and vary the number of continuous attributes between 1 and 25. The results can be seen in Figure 2(c). For smaller numbers of continuous attributes (less than 15), LOADED is more efficient than RELOADED, but since it appears that RELOADED scales near linearly while LOADED scales quadratically with respect to the number of continuous attributes, RELOADED is more efficient if there are larger numbers of continuous attributes.

As we noted above, LOADED does not scale well for large numbers of categorical attributes, since it maintains the entire itemset lattice in memory. However, LOADED uses an approximation scheme in which it only uses a partial lattice. The primary benefit of the approximation scheme is that LOADED achieves far better execution times if fewer lattices levels are maintained, as can be seen in Figure 3(a). Empirically, it appears from Figure 3(a) that execution time grows quadratically with the number of lattice levels. The detection rates decrease as the number of lattice levels decrease, as can be seen in Figure 3(b). The affect of the number of lattice levels on the false positive rates can be seen in Figure 3(c). Note that the false positive rate axis uses a log scale. The false positive rates are not affected significantly with changing lattice levels. On the other hand, detection rates seem to increase as the number of lattice levels increase to 3, after which they stabilize.

4.4.1 Distributed LOADED

In our last set of experiments, we explore the benefits gained from using the distributed versions of LOADED. We first explore the speedup obtained when running the two-pass distributed LOADED algorithm on two, four, and eight sites. The KDDCup 1999 training and synthetic data sets are split evenly between the sites for this experiment. Figure 4(a) shows the speedup obtained on the two data sets. As there is only one round of communication, the overall message passing overhead is minimal. Most of the time is spent in the two phases: 1) building the local model in the first pass; and 2) finding outliers in the second pass. Consequently, each node works independently, and we see up to 7.7-fold speedup on 8 sites. The slight reduction in speedup with increasing number of sites is due to increasing communication overhead associated the local model exchange. Next, we vary the link bandwidth between the sites in our controlled environment in order to simulate a network spanning larger distances. As shown in Figure 4(b), for a wide area setting consisting of eight nodes, efficiency varies from a low of 69% to a high of 96%, for link bandwidths equal to 1 MB/s and 100 MB/s respectively (note the log scale of the bandwidth axis). Even when link bandwidth is equal to 10 MB/s, LOADED achieves an efficiency of 95%, suggestive of good scalability for outlier detection within an organization.

Finally, we explore the speedup obtained when running the LOADED one-pass outlier detection algorithm on two, three, and four sites. The KDDCup 1999 and synthetic data sets are evenly split between the nodes for this experiment. Figure 5(a) shows speedup obtained on the two data sets. Since there are relatively few outliers in the data set, and we have a low false positive rate, there is very little communication overhead, resulting in minimal synchronization between the sites. Therefore each site is able to work independently. As the number of nodes increases, the communication overhead also increases, as more nodes are involved in the local outlier exchange. As a result we see a slight reduction from the ideal speedup, and efficiency falls to 95% on the two data set when using four

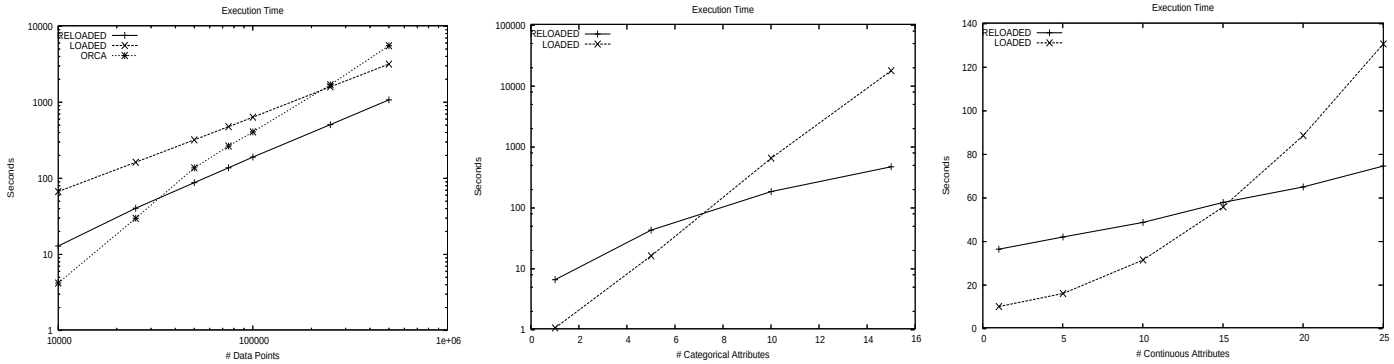

Figure 2. Plots of execution time versus (a) data set size; (b) increasing categorical attributes; (c) increasing continuous attributes.

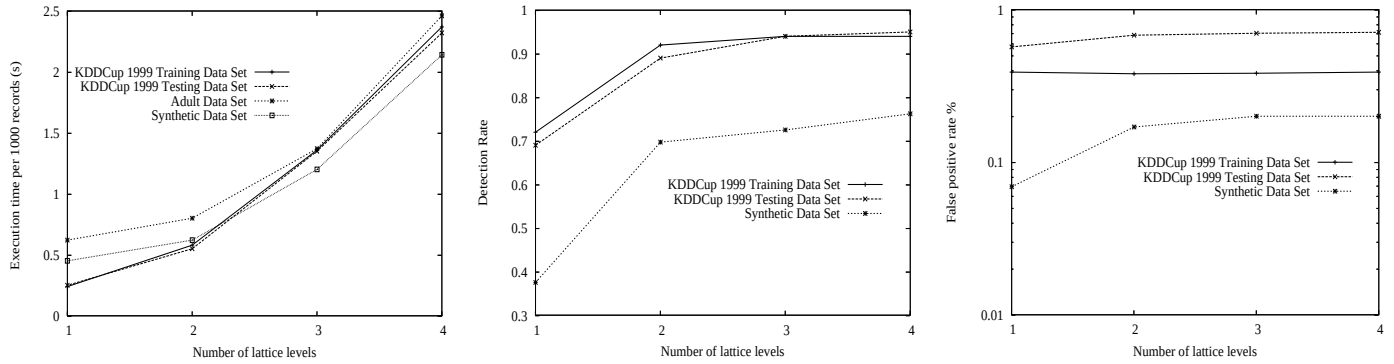

Figure 3. Effect of the number of lattice levels on (a) Execution time, (b) Detection rates, and (c) False positive rates.

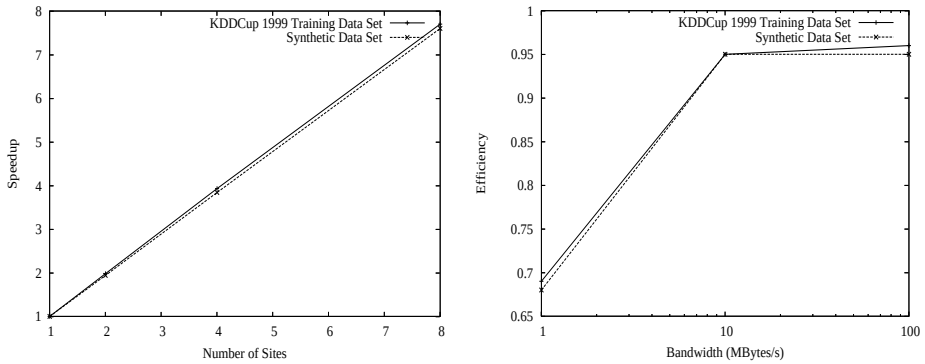

Figure 4. (a) Speedup for a varying number of sites, and (b) Expected efficiency in a wide area network.

sites.

As nodes primarily communicate by exchanging outliers, which are small messages, link latency will be the primary performance bottleneck in a wide area setting. We vary the average link latency between the nodes in our controlled environment to simulate a wide area network spanning larger distances. As shown in Figure 5(b), efficiency falls to 90% for the KDDCup 1999 data set when using 4 sites, with an average link latency of 25ms. This is representative of networks spanning across several states and excellent scalability.

5. Conclusions

It is clear from our evaluation that ORCA, LOADED, and RELOADED have different strengths and weaknesses. ORCA has relatively poor detection rates for mixed-attribute data sets such as the KDDCup data set, due to its use of an ad-hoc combination of the Euclidean and Hamming distance metrics to account for both the continuous and categorical attributes. Also, ORCA performs randomization and multiple passes of data sets, which make it unsuitable for incremental outlier detection. However, it shows excellent memory usage properties and good scalability with respect to the number of data points. LOADED, on the other hand, outperformed all the other algorithms with respect to detection rates, and is the only algorithm currently able to process distributed data sets. However, it scales very poorly with respect to memory usage, even if the number of lattice levels are restricted. RELOADED, like ORCA, scales very well with respect to memory usage, and achieves better detection rates than ORCA, at the cost of an increased false positive rate. Both LOADED and RELOADED are capable of detecting outliers in a single pass of the data, unlike ORCA. Though its detection rate is lower than LOADED's, RELOADED's small memory footprint and small execution times make it a good candidate for embedded outlier detection systems, such as might be found in network interface card-based intrusion detection [17], or sensor networks.

6. Acknowledgments

This work is supported in part by NSF grants (CAREER-IIS-0347662) and (NGS-CNS-0406386).

7. References

[1] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley, 1994.

[2] Stephen D. Bay and Mark Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proc. of 9th annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[3] C. Blake and C. Merz. UCI machine learning repository, 1998.

[4] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jorg Sander. LOF: Identifying density-based local outliers. In *ACM SIGMOD Intl. Conf. Management of Data*, 2000.

[5] D. Gamberger, N. Lavrač, and C. Grošelj. Experiments with noise filtering in a medical domain. In *ICML*, 1999.

[6] Amol Ghoting, Matthew Eric Otey, and Srinivasan Parthasarathy. Loaded: Link-based outlier and anomaly detection in evolving data sets. In *Proceedings of the IEEE International Conference on Data Mining*, 2004.

[7] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. ROCK: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366, 2000.

[8] S. Hettich and S. Bay. KDDCUP 1999 dataset, UCI KDD archive, 1999.

[9] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[10] Theodore Johnson, Ivy Kwok, and Raymong T. Ng. Fast computation of 2d depth contours. In *ACM SIGKDD*, pages 224–228, 1998.

[11] E. Knorr and *et al*. Distance-based outliers: Algorithms and applications. *VLDB Journal*, 2000.

[12] E. Knorr and R. Ng. A unified notion of outliers: Properties and computation. In *ACM SIGKDD*, 1997.

[13] E. Knorr and R. Ng. Finding intentional knowledge of distance-based outliers. In *VLDB*, 1999.

[14] E. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. Int'l Conf. on VLDB*, 1998.

[15] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A comparative study of outlier detection schemes for network intrusion detection. In *SIAM Data Mining*, 2003.

[16] Matthew V. Mahoney and Philip K. Chan. Learning nonstationary models of normal network traffic for detecting novel attacks. In *ACM SIGKDD*, 2002.

[17] M. Otey, S. Parthasarathy, A. Ghoting, G. Li, S. Narravula, and D. Panda. Towards nic-based intrusion detection. In *Proceedings of 9th annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[18] Matthew Eric Otey, Amol Ghoting, and Srinivasan Parthasarathy. Fast distributed outlier detection in mixed-attribute data sets. Technical Report OSU-CISRC-6/05-TR42, Department of Computer Science and Engineering, The Ohio State University, 2005.

[19] Matthew Eric Otey, Srinivasan Parthasarathy, and Amol Ghoting. Fast lightweight outlier detection in mixed-attribute data. Technical Report OSU-CISRC-6/05-TR43, Department of Computer Science and Engineering, The Ohio State University, 2005.

[20] Spiros Papadimitriou, Hiroyuki Kitawaga, Phillip B. Gibbons, and Christos Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In *ICDE*, 2003.

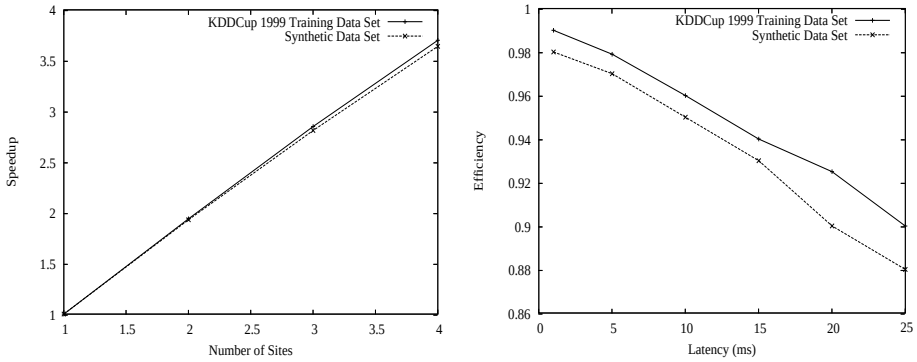

Figure 5. (a) Speedup for the single-pass approach, and (b) Expected efficiency in a wide area network.

[21] Karlton Sequeira and Mohammed Zaki. Admit: Anomaly-based data mining for intrusions. In *ACM SIGKDD 02*, pages 386–395, 2002.

A Comparison of Generalizability for Anomaly Detection

Gilbert L. Peterson Robert F. Mills Brent T. McBride Wesley C. Allred
Department of Electrical and Computer Engineering
Air Force Institute of Technology
{gilbert.peterson, robert.mills}@afit.edu

ABSTRACT
In security-related areas there is concern over the novel "zero-day" attack that penetrates system defenses and wreaks havoc. The best methods for countering these threats are recognizing "non-self" as in an Artificial Immune System or recognizing "self" through clustering. For either case, the concern remains that something that looks similar to self could be missed. Given this situation one could logically assume that a tighter fit to self rather than generalizability is important for false positive reduction in this type of learning problem.

This article shows that a tight fit, although important, does not supersede having some model generality. This is shown using three systems. The first two use sphere and ellipsoid clusters with a *k*-means algorithm modified to work on the one-class/blind classification problem. The third is based on wrapping the self points with a multidimensional convex hull (polytope) algorithm capable of learning disjunctive concepts via a thresholding constant. All three of these algorithms are tested on an intrusion detection problem and a steganalysis problem with results exceeding published results using an Artificial Immune System.

Categories and Subject Descriptors
I.5.3 [Pattern Recognition]: Classifier design and evaluation, feature evaluation and selection.

General Terms
Algorithms, Security

Keywords
Anomaly detection, clustering, intrusion detection systems.

1. INTRODUCTION
The development of computer and network intrusion detection systems has been conducted along two paths. The first development thrust identifies signature elements of attacks, and includes them in an attack database. The database is then compared with incoming samples looking for matches, and if a match occurs, the user, packet, or file is blocked from the internal network. This is the approach taken by the majority of commercial intrusion detection and steganalysis products, with the capability of catching most known attacks with very few false alarms. A limitation of this approach is that the attack must be known before it can be given a signature and blocked. Subtle, stealthy probes will most likely not be picked up by this type of system (Williams et al, 2001). Additionally, due to the sample arrival rate and database matching procedure, the speed at which attacks can be blocked will be limited.

An alternative attack matching method is based on anomaly detection. In this approach, a machine learning algorithm learns a model of normal operating behavior so that abnormal conditions can be identified. The advantage of this approach is that novel attacks (for which signatures have not been identified) may be identified and blocked. Additionally, the approach may be much quicker, because maintenance of an online signature database for matching purposes is not required. A disadvantage is that an attacker with knowledge of which attributes are used for detection could construct stealthy attacks that avoid using or manipulate the attributes used by the machine learning algorithm to appear normal.

In order to detect attacks from an attacker trying to blend in to normal traffic, we examine fitting the normal "self" data more closely. Figure 1 shows the results of applying the modified *k*-means sphere, ellipse, and the convex polytope algorithms to each class separately for a simple two class problem. As can be seen from just this simple example, the generalizability of the model decreases as the model improves its tightness to the data points. One could also imagine that if these classes were more interspersed that the convex polytope which provides the closest fit to the data would perform the best. Given a domain in which the attackers attempt to craft an attack that appears as close to normal (self) as possible, a learning approach which fits the model closely could be seen as important.

In the following sections we discuss related work on anomaly detection for the intrusion detection and steganalysis domains used for testing. This is followed by a discussion of how we have modified *k*-means and the thresholding element required for the convex polytope to learn disjunctive concepts. The test results are then presented showing that a tight fit is important but that generalizability is still necessary given the sampling of the normal/self space.

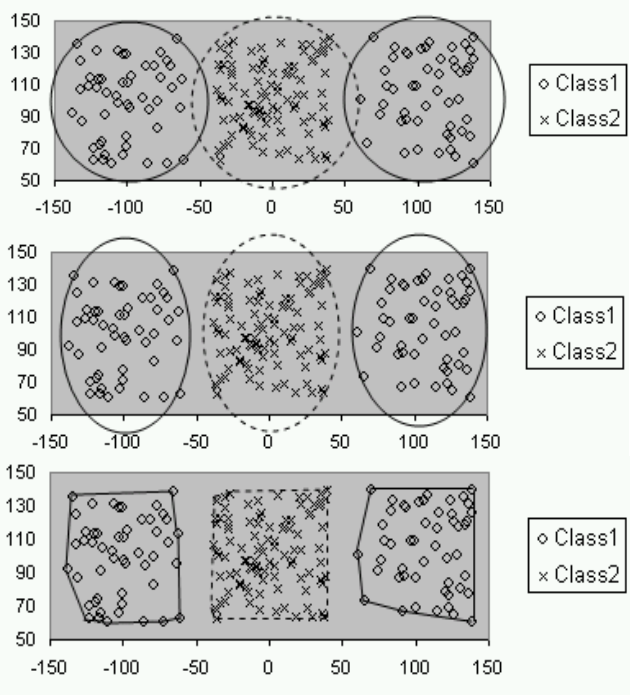

Figure 1. The 2-Class Problem with Sphere, Ellipse and Convex Polytope

2. RELATED WORK

In this section we discuss related work on anomaly detection for the intrusion detection and steganalysis domains.

2.1 Intrusion Detection Systems

Anomaly detection systems have been built making use of rule learning, neural networks, Artificial Immune Systems (AIS), and clustering methods. The clustering methods and Artificial Immune Systems are most closely related to this work in that the systems can be trained using only normal traffic. Artificial Immune Systems train on normal data by enclosing non-self space with randomly generated immune system cells. These cells then take part in an evolutionary algorithm evolution process until as much of the non-self space as possible is covered with none of the cells impinging on self space (Harmer, et al, 2002). We compare our results with an AIS technique (Dasgupta and Gonzales, 2002) in a later section. Researchers have also made use of system call activity as another source of data for anomaly based intrusion detection (Hofmeyr, et al, 1998; Nguyen, et al, 2003; and Tan, et al, 2003).

The application of clustering to intrusion detection groups network traffic into subclasses such that the members in one subclass are similar, while members of different subclasses are distinct. Several techniques have been studied, such as *k*-means, Self Organizing Maps (SOM), Neural-Gas, and Mixture-of-Spherical Gaussians (MOSG) to name a few. Clustering has been shown to produce very good results as an unsupervised IDS technique (Zhong, et al, 2004) and for data reduction prior to categorization (Zanero and Savaresi, 2004). In addition, there is a variation of *k*-means that also contains a stochastic element which behaves like an AIS (Guan, Ghorbani and Belacel, 2003).

2.2 Steganography

Steganography refers to hiding information in an innocuous place so that it may be transmitted without notice. In the digital realm, specifically digital images, the message is hidden within a cover image. The hiding or steganograpy process varies the image's pixels in such a way that the changes are virtually undetectable to the human eye. The cover images that provide the most difficulty for message detection are JPEG images.

JPEG compression is a lossy image compression technique that exploits the fact that the eye cannot detect small changes in an image. In a JPEG image, a message is stored using the least significant bit (LSB) or even through rounding errors on the quantized discrete cosine transform (DCT) coefficients representing 8x8 blocks of the image.

For the lossy steganography problem there have only been a few applications of learning models for normal images, and none have used any type of clustering. Approaches which make use of both self and non-self data have used Fisher's linear discriminant, Support Vector Machines with image quality metrics, and wavelet statistics calculated from the suspect images (Farid and Lyu, 2002; Lyu and Farid, 2002; and Avcibas, et al, 2002). A survey of the metrics available and their utility is provided in (Kharazzi, et al, 2004).

Blind or one-class learning methodologies have consisted of Artificial Immune Systems (Jackson, 2003) and single class Support Vector Machines (Lyu and Farid, 2004).

3. METHODS

In this section, we discuss how we have modified *k*-means and the thresholding element required for the convex polytope to learn disjunctive concepts.

3.1 *k*-means

The *k*-means algorithm is a clustering algorithm which assigns points to clusters by attempting to minimize the sum of squared errors within groups, or the sum of the distance squared between each point and the centroid of its assigned cluster. The algorithm then iteratively updates the cluster centroids moving the centroid toward the center of the cluster's points. This is followed by reassigning points to different clusters until it can no longer reduce the sum of squared within group errors. The time complexity of the *k*-means algorithm is O(*knr*) for *k* clusters, *n* points, and *r* iterations (Wong, Chen and Yeh, 2000).

As *k*-means is being used as a classifying algorithm, a class is described by a set *S* of *k* hyper-spheres. First, the *k*-means clustering algorithm partitions the self data into *k* different clusters, where *k* acts as a tolerance parameter for the hyper-sphere classification algorithm by controlling the partitioning of the self data. For the spherical version, a radius for each cluster is calculated from the distance between the corresponding centroid and the most distant point in the cluster. A new sample is declared part of self if it falls within one of the cluster radii.

A good IDS or steganalysis detection system should have a high probability of detection (P_D) and small probability of false alarm (P_F). The challenge is finding the appropriate balance between these opposing objectives. For example, decreasing the volume

of the training class reduces the number of missed detections, thereby improving P_D, but at the expense of more false alarms and a higher P_F. As a method to create a tradeoff between P_D and P_F, a tolerance parameter, $0 < \delta \leq 1$, applied to each cluster's radius provides a simple method to constrain the clusters from covering too much non-self space.

An ellipsoid model was also used to strike a balance between the loose fitting spherical *k*-means representation of self space and the very tight fitting convex polytope described in the next section. An ellipsoid in *d* dimensions is represented by three parameters defining its location size (*s*: a scalar value), (μ: a *d*-vector specifying the center point), and shape (Σ^{-1}: a *d*-by-*d* matrix describing the shape of the ellipsoid). Any point *x* on the ellipsoid boundary (locus) satisfies

$$(x-\mu)^T \sum^{-1} (x-\mu) = s$$

The ellipsoid model in *k*-means minimizes to find not only a cluster center μ but the shape Σ^{-1} as well. This increases the creation time complexity to $O(kn^2d^2)$.

3.2 Convex Polytope

A *d*-polytope is a closed geometric construct bounded by the intersection of a finite set of hyperplanes, or halfspaces, in *d* dimensions (Coxeter, 1973). The polytope is convex if all points in a line segment between any two points on the polytope boundary lie either within the polytope or on its boundary. A convex hull of a set of points *S* in *d* dimensions is the smallest convex *d*-polytope that encloses *S* (O'Rourke, 1998). Each vertex of this enclosing polytope is a point in *S*.

For classification purposes, the convex polytope for a class *C* is built from the set *T* of *d*-vectors from the sample space. If the desired geometric shape is a convex *d*-polytope, then the convex hull *H* of *T* is computed. There are several algorithms for computing convex hulls in higher dimensions (Avis, et al, 1997). This research uses the *qhull* program (Barber, et al. 1997), which has a time complexity of $O(n^{\lfloor d/2 \rfloor})$ for *n* points in *d*-space. A distinct test point *p* is declared to be a match (member of class *C*) if and only if it is bounded by the polytope defined by *H*.

To account for class disjunction, we define $0 \leq \beta \leq 1$ as a tolerance parameter to control the creation of smaller convex hulls. With $\beta = 1$, the algorithm creates a single convex polytope around all training points. As β decreases, the potential number of smaller polytopes increases, and their combined hyper-volume in the attribute space decreases. For the extreme case $\beta = 0$, no convex hull models are created and all test points are subsequently rejected. The method for constructing the smaller convex hulls is described in (McBride and Peterson 2004).

Selecting different values of β allows us to achieve the desired balance between false positive and false negative error probabilities. If instances of all possible testing classes are available when creating the class model, then the value of β that best fits the training data (i.e., provides an appropriate balance between false positives and false negatives) can be found through experimentation.

4. TESTING

The flexibility of these classifiers allows for uses in many possible domains. Our research focuses on evaluating anomaly classification as applied to the problems of detecting suspicious computer network activity and steganography, both of which may accompany an attack against a computer network by an outsider. These domains also show the classification capabilities on windowed time series data (IDS) as well as discrete sampled data (steganalysis).

4.1 IDS Experiment

The dataset used for this experiment was obtained from the Lincoln Laboratory of the Massachusetts Institute of Technology. MIT maintains data sets with normal and abnormal information collected in a test network (Haines, et al, 1999). Although this data set has been shown to be statistically different from normal traffic (Mahoney and Chan, 2003), its many uses by the research community allow for comparison with other approaches. For this experiment, we used the 1999 data set, with week 1 (normal traffic) to train our classifiers, and week 2 (normal traffic mixed with attacks) for testing. Abnormal activity includes both internal (misuse) and external (hacking or denial of service) attacks, but not the external use of operating system or application exploits, as shown in Table 1.

Table 1. Week 2 Attack Profile

Day	Attack	Attack Type	Start Time	Duration
1	Back	DOS	9:39:16	00:59
2	Portsweep	Probe	8:44:17	26:56
3	SATAN	Probe	12:02:13	2:29
4	Portsweep	Probe	10:50:11	17:29
5	Neptune	DOS	11:20:15	04:00

We follow the same data preparation methodology as (Dasgupta and Gonzalez 2002) and collect statistics on the number of bytes per second, number of packets per second, and number of Internet Control Management Protocol (ICMP) packets per second for classification features. These features were sampled each minute from the raw *tcpdump* data files. Dasgupta and Gonzalez showed that while none of these features alone could reliably detect the five attacks, combining the features was quite effective. They also explored overlapping the time series as a means of detecting temporal patterns, with their best results generated using a sliding window of three seconds.

False positive and true positive probabilities were calculated by comparing the classifier output with the Week 2 attack data. Table 2 shows the results of testing the *k*-means sphere and ellipse classifiers, the convex polytope, and the AIS results (Dasgupta and Gonzalez, 2002) on the MIT IDS dataset. Multiple tests for each algorithm were run, and the table contains the best results found for P_F and P_D of each algorithm with the exception of the AIS which includes the results for 1 and 3 time slices from (Dasgupta and Gonzales, 2002).

Table 2. IDS Results

	Sphere		Ellipse		Polytope		AIS	
	k=75 δ = 1.0	k=100 δ = 0.9	k=30 δ = 1.0	k=75 δ = 1.0	$\beta > 0.3$	$\beta = 0.1$	1 time slice	3 time slices
P_D (%)	1.82	5.45	98.2	100.0	98.2	100.0	92.8	98.0
P_F (%)	0.0	1.02	0.0	0.2	0.27	0.35	1.0	2.0

During testing of the *k*-means variations, *k*-values ranged from 1-100 in steps of 5 and δ=0.9, 0.95, and 1.0 were used to determine classifier sensitivity as a function of the number of ellipsoids used to fit the training data. As shown, the ellipsoid model with its added capability of generalizing beyond the strict sampling is able to better fit the training data over the convex polytope which was trained using several values of β for $0 \leq \beta \leq 1$. In addition, the results show that the sphere version of *k*-means performs very poorly predominantly because it inaccurately covers the training attribute space by also enclosing space including anomalous data points. This continues even as *k* increases and each cluster decreases in size. The reason the sphere does not perform as well as the other two geometric constructs is that the *k*-means classifier uses the point furthest from the mean for each cluster to estimate the size of the hyper-sphere, resulting in an over-generalization. This contrasts with the ellipse and convex polytopes which try to maintain a closer fit to the training data.

These results imply that the convex polytope and the ellipse *k*-means had little trouble fitting the training data, and that their ability to more tightly fit the self space improves their overall performance for classification based on these three statistical attributes. Additionally this shows that although both models fit the data closely that the added generality of the ellipse *k*-means assists in reducing the false positives which is counter to the assumption that one would want the closest fit to the training data for anomaly detection.

4.2 Steganalysis Experiment

For this domain we test using the wavelet coefficient statistics (Farid and Lyu, 2003) derived from a database of 1,100 grayscale images. The best three of the 36 coefficients determined by J-score are extracted from each image. In addition to clean images, the testing set includes steganographic images created with Jsteg, and Outguess with and without statistical correction. For each of these three steganography methods, images are created using 100%, 50%, 25%, and 12.5% of the cover image's embedding capacity.

Figure 2 shows the results from the steganography testing compared with the results using the same testing domain and an AIS as the classifier from (Jackson 2003). As seen with the IDS problem, the closer fit to the self space provided by both the convex polytope and ellipse *k*-means outperforms the more general sphere *k*-means. However, it is also shown that striving for the closest fit possible, i.e. the convex polytope, is also not the direction that should be pursued. Specifically, the lack of generality, especially on the Jsteg dataset, is detrimental to the convex polytope over the ellipse *k*-means.

5. CONCLUSIONS

For security anomaly detection domains, a concern prior to fielding the system is whether it can be spoofed by an attacker manipulating their attack to appear similar to normal traffic. In order to combat such an event we proposed that a model of self should fit the normal self sample tightly. This theory has been tested on two security domains, namely intrusion detection and steganalysis.

This paper shows that while the convex polytope algorithm provides the tightest fit to self, the ellipsoid *k*-means provides the best balance between a tight fit and sufficient generality. The small amount of generality provided by the ellipse resulted in a better ability to detect novel events that may otherwise go undetected in a classifier with a tight fit. This is especially worrisome in a network intrusion scenario in which the attack pattern appears as close to normal as possible. The results have demonstrated that a tight fit is important but does not obviate the need for generality.

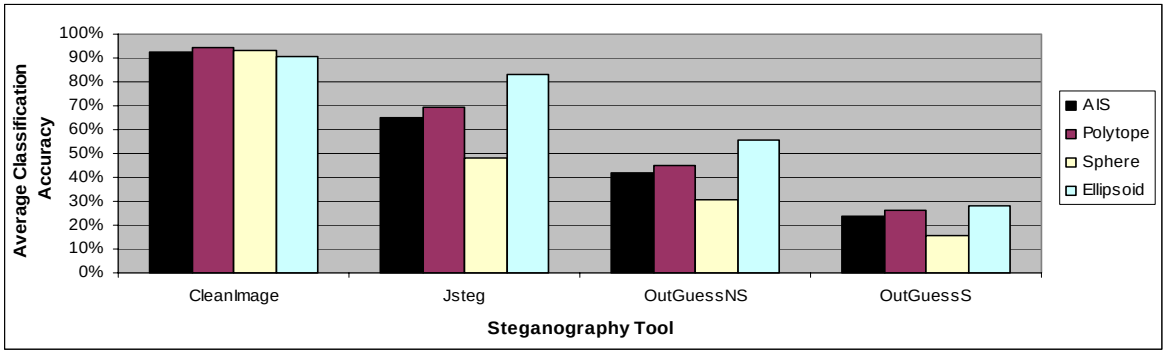

Figure 2. Steganography Results

6. REFERENCES

[1] Avcibas, I., Memon, N., and Sankur, B. "Image Steganalysis With Binary Similarity Measures", *International Conference on Image Processing*, Rochester, NY, September 2002.

[2] Avis, D., Bremner, D., and Seidel, R. "How Good are Convex Hull Algorithms?" *ACM Symposium on Computational Geometry*, Nice, France, 1997.

[3] Barber, C., Dobkin, D., and Huhdanpaa, H. "The Quickhull Algorithm For Convex Hulls", ACM Trans. on Mathematical Software, 22, 469-483, 1997.

[4] Coxeter, H. S. M. *Regular Polytopes, 3rd ed.* New York: Dover, 1973.

[5] Dasgupta, D., and Gonzales, F. "An Immunity-Based Technique to Characterize Intrusions in Computer Networks", *IEEE Trans. on Evolutionary Computation, Vol 6*, June 2002.

[6] Faird, H. and Lyu, S. "Higher-order Wavelet Statistics and their Application to Digital Forensics", *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, Wisconsin, June 2003.

[7] Guan, Y., Ghorbani, A., and Belacel, N. "Y-Means: A Clustering Method for Intrusion Detection", *IEEE Canadian Conference on Electrical and Computer Engineering CCECE*, Montréal, Canada, May 2003.

[8] Haines, J., Lippmann, R. , Fried, D., Tran, E., Boswell, S., and Zissman, M. "1999 DARPA Intrusion Detection System Evaluation: Design and Procedures", MIT Lincoln Laboratory Technical Report.

[9] Harmer, P., Williams, P., Gunsch, G., and Lamont, G. "An artificial immune system architecture for computer security applications", *IEEE Transactions on Evolutionary Computation, Vol 6*, June 2002.

[10] Hofmeyr, S., Forrest, S., and Somayaji, A., "Intrusion Detection Using Sequences of System Calls", *Journal of Computer Security, Vol. 6*, pp. 151-180 (1998).

[11] Jackson, J. *Targeting Covert Messages: A Unique Approach For Detecting Novel Steganography*, Masters Thesis, Air Force Institute of Technology, Wright Patterson Air Force Base, Ohio, 2003.

[12] J-Steg Steganograpy software for Windows, http://members.tripod.com/steganography/stego/software.html.

[13] Kharrazi, M., Sencar, T., and Memon, N. "Benchmarking Steganographic And Steganalysis Techniques", *EI SPIE San Jose*, CA, January 16-20, 2005.

[14] Lyu, S., and Farid, H. "Detecting Hidden Messages Using Higher-Order Statistics And Support Vector Machines," *Information Hiding: 5th International Workshop, IH 2002*, Noordwijkerhout, The Netherlands, October 7-9, 2002.

[15] Lyu, S., and Farid, H. "Steganalysis Using Color Wavelet Statistics And One-Class Support Vector Machines," *SPIE Symposium on Electronic Imaging*, San Jose, CA, 2004.

[16] McBride, B. and Peterson, G. "Blind Data Classification using Hyper-Dimensional Convex Polytopes", *Proceedings of the 17th International FLAIRS Conference*, Miami Beach, FL, 2004.

[17] Mahoney, M., and Chan, P., "An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection", *Proceedings of the Recent Advances in Intrusion Detection, RAID 2003*, Pittsburgh, PA, USA, September 8-10, 2003.

[18] Nguyen, N., Reiher, P., and Kuenning, G. "Detecting Insider Threats by Monitoring System Call Activity", *2003 IEEE Workshop on Information Assurance*, United States Military Academy, West Point, NY, June 2001.

[19] Provos, N., "Defending Against Statistical Steganalysis", *Proceedings of the 10th USENIX Security Symposium*, Washington, DC, 2001.

[20] O'Rourke, J. *Computational Geometry in C, 2nd ed.* Cambridge, England: Cambridge University Press, 1998.

[21] Tan, K., McHugh, J., and Killourhy, K. "Hiding Intrusions: From the Abnormal to the Normal and Beyond", *Information Hiding: 5th International Workshop, IH 2002*, Noordwijkerhout, The Netherlands, October 7-9, 2002.

[22] Williams, P., Anchor, K., Bebo, J., Gunsch, G., and Lamont, G. "Warthog: Towards a Computer Immune System for Detecting 'Low and Slow' Information System Attacks", *Proceedings of the Recent Advances in Intrusion Detection Symposium, RAID 2001*, Davis, California, 2001.

[23] Wong C., Chen, C., and Yeh, S. "K-Means-Based Fuzzy Classifier Design", *Proceedings of the Ninth IEEE International Conference on Fuzzy Systems, Vol. 1*, pp. 48-52, 2000.

[24] Zanero, S., and Savaresi, S. M. "Unsupervised Learning Techniques for an Intrusion Detection System", *Proceedings of the 19th Annual ACM Symposium on Applied Computing*, Nicosia, Cyprus, 2004.

[25] Zhong, S., Khoshgoftaar, T., and Seliya, N., "Clustering-Based Network Intrusion Detection", To appear in *International Journal of Reliability, Quality, and Safety Engineering (IJRQSE)*, 2005.

Detecting Anomalous Patterns in Pharmacy Retail Data

Maheshkumar R. Sabhnani
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
sabhnani+@cs.cmu.edu

Daniel B. Neill
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
neill@cs.cmu.edu

Andrew W. Moore
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
awm@cs.cmu.edu

ABSTRACT
This paper describes a bio-surveillance system designed to detect anomalous patterns in pharmacy retail data. The system monitors national-level over-the-counter (OTC) pharmacy sales on a daily basis. Fast space-time scan statistics are used to detect disease outbreaks, and user feedback is incorporated to improve system utility and usability.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Apps—Data Mining

General Terms
Algorithms

Keywords
Cluster detection, space-time scan statistics, biosurveillance.

1. INTRODUCTION
Bio-surveillance systems have recently gained a lot of attention and are growing more and more complex. Multiple sources of data (pharmacy sales, emergency department visits, weather indicators, census information, etc.) are now available, and these sources can be used to identify both natural disease outbreaks (e.g. influenza) and outbreaks resulting from bio-terrorist attacks (e.g. anthrax release). The bio-surveillance research community is actively developing intelligent algorithms to detect outbreaks in a timely manner, in order to save lives and costs. However, though many of these algorithms show impressive results under simulated environments, their performance tends to degrade when applied to real-world datasets. Seasonal and day-of-week trends, missing data, lack of known disease outbreaks, difficulties in designing test beds, and high costs associated with processing false positives are some of the many reasons that hinder development of a successful practical bio-surveillance system. We believe that incorporating expert knowledge from public health officials will provide valuable insight to this complex process of disease outbreak detection. An immediate goal is to provide a tool that not only shows the alarms to the expert users, but also allows them to provide feedback on the alarms. This feedback loop is essential for iterative refinement of outbreak detection tools. This paper highlights our experiences with developing such a bio-surveillance system that currently monitors national level pharmacy sales of over-the-counter (OTC) drugs on a daily basis.

Our system searches for spatio-temporal patterns in the OTC data from pharmacies, grocers and other stores that sell OTC products throughout the United States. Given some search region (which can be a city, county, state, or even the entire country), the algorithm first maps this search region to a uniform, rectangular $N \times N$ grid. It then searches over all axis-aligned rectangular regions on the grid, in order to find regions that have shown a recent anomalous increase in sales. The regions that show high deviation in sales from the estimated baselines are labeled as alerts–clusters of OTC sales that may indicate disease outbreaks. A detailed description of the algorithm is available in [1-3]. Given our limited ability to distinguish clusters caused by outbreaks from clusters with other causes, we present selected alerts to public health officials only after they have been filtered by some simple rules to remove unimpressive anomalies. Their feedback is then used to improve the performance of the algorithm. The following sections describe this system in detail.

2. SYSTEM OVERVIEW
The National Retail Data Monitor, developed and operated by the RODS (Real-time Outbreak and Disease Surveillance) Laboratory at the University of Pittsburgh, receives the OTC data from the national and local vendors [4, 5]. The data consists of daily store level sales of 9000 OTC products used for the symptomatic treatment of infectious diseases. The NRDM groups individual product sales into 18 groups of similar products (e.g., Baby/Child electrolytes, Cough/Cold, Thermometers, Stomach remedies, and Internal analgesics). We process the past three months of data (around 5.5 million records) to estimate recent baselines (i.e. the number of sales we would expect to see in each store). Each record includes the store ID, its corresponding zip code, date of sale, and units sold for a particular syndrome. There are more than 10,000 unique stores present in the data. This data is received on a daily basis, with one-day delay from the date of sale. There are various challenges with estimating the store baseline sales. First, there are strong seasonal and weekly trends in the OTC data. Figure 1 shows a sample weekly trend in Baby/Child electrolyte sales. Sales on a typical Monday and Tuesday tend to be higher than on Friday and Saturday. This trend depends on many factors: region location, urban or rural community, etc. Figure 2 shows the seasonal trends in Cough/Cold sales. Average daily sales in the month of March were ~5000 units higher than in April. We have also noticed a

sudden rise in sales for days following a national holiday. We address the seasonal and day-of-week trends by incorporating them into the baseline time-series analysis. The current data storage schema does not differentiate between missing data (i.e. stores that have not reported sales for a specific date by the time of analysis) and zero counts (i.e. stores that sold zero units on that date). To deal with this limitation, we assume that data are missing only if a store reports no sales for all product categories; if a store has zero counts for some product categories and non-zero counts for others, the zero counts are assumed to result from zero sales rather than from missing data. We infer all missing data points from the time series of counts for that location, using an exponentially weighted moving average technique. Once the time series has no missing data, any reasonable univariate time series algorithm that accounts for day-of-week and seasonal trends can be applied to estimate recent baseline sales.

After we receive the past three months of national OTC data, we define multiple search regions with differing resolution (some states, some counties, and others that cover the entire country). This ensures that we detect large-scale anomalies, and not just daily fluctuations at the store or zip code level. As noted above, the search region is mapped to a rectangular two-dimensional grid of size $N \times N$. We need to know the store locations in order to map them onto the grid cells; however, due to data privacy concerns, we do not have access to the exact longitude and latitude of each store. Instead, we are given the zip code containing each store, and use the longitude and latitude of the zip code centroid to populate the grid cells. The search algorithm then scores every possible axis-aligned rectangular region using the recent baselines (expected counts) and observed counts in the region. Baseline values can be aggregated either for individual stores (the "building-aggregated time series" method, or BATS) for individual grid cells (the "cell-aggregated time series" method, or CATS), or on-the-fly for an entire search region (the "region-aggregated time series" method, or RATS). Additionally, a variety of methods are used for time-series analysis. For details on aggregation techniques and time series algorithms tested on the OTC data, please refer to [3]. The scoring function assumes that baseline sales follow a Poisson distribution. We also perform significance testing on the score of each region by randomization. This helps us remove anomalous regions that could be explained as being generated by chance. The *k*-best regions (i.e. those significant regions with the highest scores, and therefore the lowest *p*-values) are reported as possible disease outbreaks.

3. SYSTEM EVOLUTION

The primitive versions (version 1.X) of the current spatial scan statistics (SSS) system involved reporting significant regions via e-mail. Each day, a set of states and counties was scanned for anomalous regions, and the alert results for each state/county were sent as an e-mail attachment to the appropriate public health officials. Though the users were given the latitude, longitude, syndrome, score, and *p*-value of each alert region, it was difficult for them to get a feel of where exactly the outbreak occurred, or to interpret the probable cause of the alert (i.e. whether it was a real outbreak or a false positive). To deal with these issues, we developed a SSS viewer application tool with a dual purpose. First, it allows end users to browse the data that led to an alert. Second, it provides easy feedback opportunities in which they can tell us which alerts were genuine and which were uninteresting or due to non-outbreak reasons. Figures 4 and 5 show sample screen shots of our viewer tool. Salient features of this tool include showing alert-region time series, showing store-level data in the region, and navigating in and around the alert region on the GIS map to help further investigate the alert. We released this tool during our version 2 release. In this version, all alerts were displayed on the website rather than via e-mails. The current version 3.0 (to be released in June 2005) has enhanced capabilities on the web. Now users can not only view alerts, but they can also rank them, add feedback comments, and give suggestions. Users can also search for alerts using different criteria, such as zip code, score, observed counts, expected counts, etc. We are trying to extract user expertise in identifying features of the clusters that may discriminate between clusters likely due to disease outbreaks and clusters likely due to other causes. Another powerful tool that we have given to users is to add their custom-defined input scripts to the pool of scripts that run daily. Users can set their own grid resolution, change baseline evaluation time series method, set aggregation level, etc. By enabling users to create their own input scripts, we can learn what results and settings are most relevant to real users in the surveillance task. This feedback will help us better manage these alerts and distinguish true outbreaks more efficiently. Figure 3 shows a sample screen shot of the user home page. In the future, we also plan to provide more features (e.g. providing store locations, tracking of previously reported alerts for post analysis purposes, etc.) to the end users so that they can give better feedback.

We have been running this system daily on OTC data for over one year. Initially the algorithm reported a large number of false positives: regions that were statistically significant according to our model but clearly did not correspond to actual outbreaks. Some of these false positives resulted from "single store" anomalies: individual stores with large spikes in sales on a given day. Two possible explanations for these single store anomalies are bulk purchases by a single buyer (e.g. restocking by a hotel, clinic, etc.) or promotional sales. We address this issue by only reporting those regions that have shown increased counts due to multiple stores: in other words, we filter out a region if removing any single store from that region would cause its score to become insignificant. In order to make a simple adjustment for potentially unmodeled fluctuations in day-to-day counts, we also apply a conservative "threshold" filter, which assumes that the baselines were underestimated by some amount (e.g. 15%). If both the "single-store" adjusted score and the "threshold" adjusted score are still significant, we report the region as a potential outbreak. Figure 4 shows a recent potential Baby/Child Electrolyte disease outbreak at the border of Alabama and Georgia. There are 16 stores in this area, and at least five of these stores have shown high deviations from baseline in electrolyte sales. The alert region is not shown in the figure due to data privacy concerns.

We have already observed a number of unique and interesting trends in the OTC data using this system. For example, people tend to buy some products just before inclement weather (such as snowstorms or hurricanes), presumably to stockpile them. There is also typically a rise in OTC sales immediately after a national holiday. Another interesting effect recently observed was increased sales in tourist destinations during long weekends. Figure 5 illustrates this trend during the recent Memorial Day weekend. Since the NRDM has highest coverage in the eastern

United States, a large number of tourist destinations (gray highlighted regions on the map) produced alerts resulting from the change in population distribution around these areas. Again, due to data privacy concerns, we have not shown the location of the region whose time series is shown below the map. Although these are interesting results, they underscore the difficulty of determining which increases in sales are due to real outbreaks, and which increases are due to a variety of other unmodeled factors. In the near future, we intend to increase the number of outbreak indicators: adding more algorithms and data sources to the system. We are planning to add emergency department data and include more independent univariate time-series algorithms to improve our confidence when alerting outbreaks. This system is helping us to understand the real-world OTC data and to improve our detection models and methods. Continued feedback from public health users will increase our ability to differentiate true outbreaks from yet unknown natural causes for increased OTC sales, thus enabling us not only to find "significant" regions, but also to determine which of these clusters are most relevant for public health investigation.

4. ADDITIONAL AUTHORS

Fu-Chiang Tsui, Michael M. Wagner, and Jeremy U. Espino (RODS Laboratory, University of Pittsburgh, Pittsburgh, PA 15213). E-mail: {tsui, mmw, jue}@cbmi.pitt.edu.

5. REFERENCES

[1] D.B. Neill and A.W. Moore. Rapid detection of significant spatial clusters. *Proceedings of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 256-265, 2004.

[2] D.B. Neill, A.W. Moore, F. Pereira, and T. Mitchell. Detecting significant multidimensional spatial clusters. *Advances in Neural Information Processing Systems* **17**, 969-976, 2005.

[3] D.B. Neill, A.W. Moore, M. Sabhnani, and K. Daniel. Detection of emerging space-time clusters. Accepted to *11th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2005.

[4] M.M. Wagner, F.-C. Tsui, J. Espino, W. Hogan, J. Hutman, J. Hersh, D.B. Neill, A.W. Moore, G. Parks, C. Lewis, and R. Aller. A national retail data monitor for public health surveillance. *Morbidity and Mortality Weekly Report, Supplement on Syndromic Surveillance* **53**, 40-42, 2004.

[5] M.M. Wagner, J.M. Robinson, F.-C. Tsui, J.U. Espino, W.R. Hogan. Design of a national retail data monitor for public health surveillance. Journal of the American Medical Informatics Association 10/5 (Sept/Oct), 409-418, 2003.

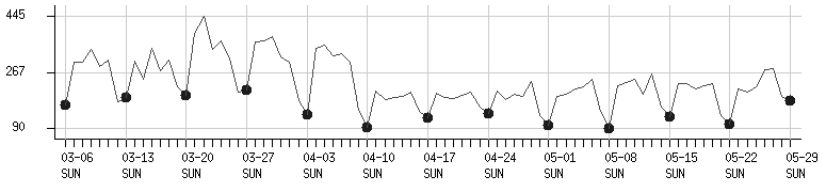

Figure 1. Weekly trend in Baby/Child electrolyte sales

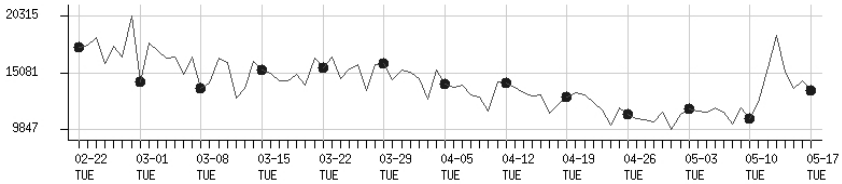

Figure 2. Seasonal trend in Cough/Cold sales

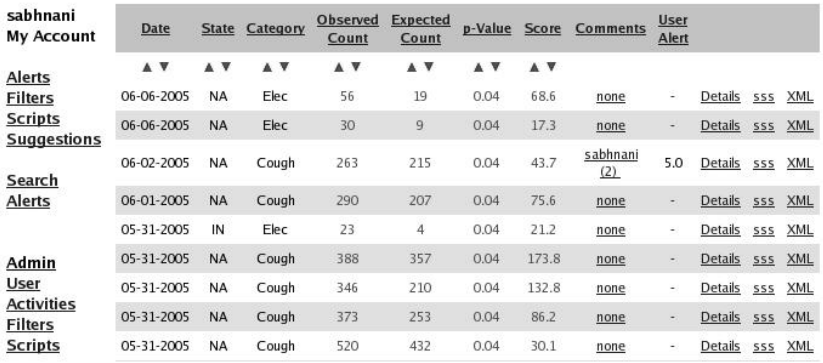

Figure 3. Screen shot of SSS user home page on the Web

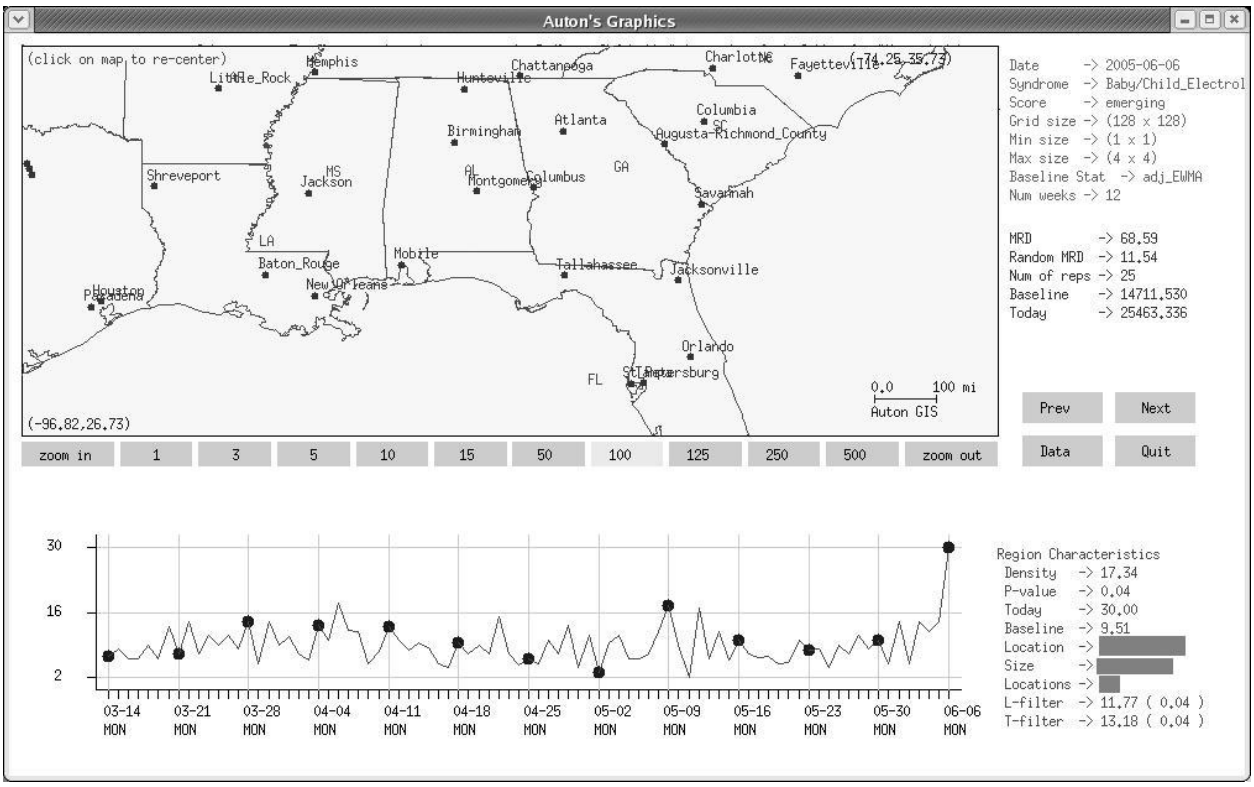

Figure 4. Potential disease outbreak at the border of Alabama and Georgia

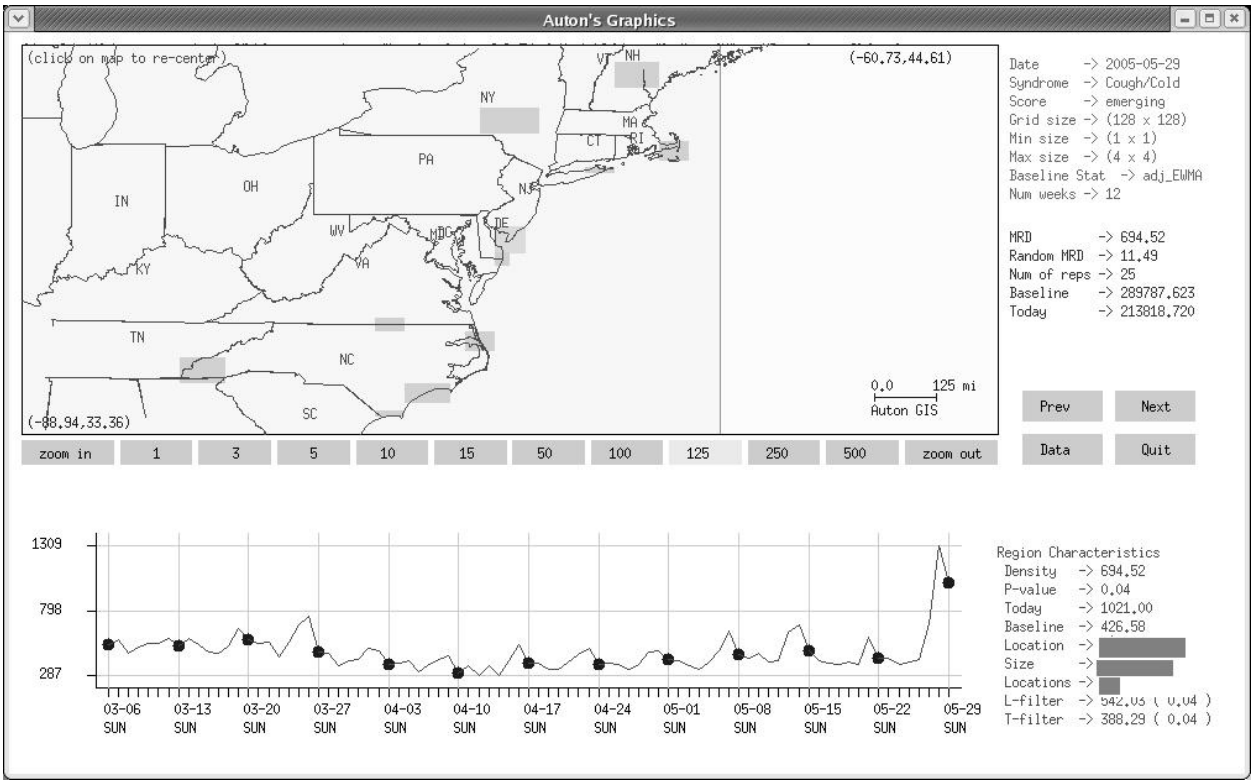

Figure 5. Long weekend trend showing the tourist spots in the country

Filtering Search Engine Spam based on Anomaly Detection Approach

Kazumi Saito
NTT Communication Science Laboratories, NTT Corporation
Kyoto 619-0237, Japan
saito@cslab.kecl.ntt.co.jp

Naonori Ueda
NTT Communication Science Laboratories, NTT Corporation
Kyoto 619-0237, Japan
ueda@cslab.kecl.ntt.co.jp

ABSTRACT

In this paper, we address the problem of filtering search engine optimization (SEO) spam websites that have directly irrelevant contents to some search terms. Focusing on hyperlink structure of a given network, we present a method for detecting the spam websites that have *anomalously* dense connections within some cores in the network. Based on the graph spectral approach, the method calculates the principal eigenvector of the network adjacency matrix, ranks the nodes according to the element values of the eigenvector, and detects a set of densely connected nodes as a spam core. By repeatedly performing the procedure after removing the links in the detected core, we can detect several spam cores. Using a real traceback network data in a *blogspace*, we experimentally evaluate the detection performance of our method by comparing with the conventional methods based on fundamental metrics on a network. We also demonstrate that our method could separately detect SEO spam entries according to their types.

1. INTRODUCTION

Search engine optimization (SEO) is the process of increasing the amount of visitors to a Web site by ranking high in the search results of a search engine [1]. However, there are often SEO spam websites that contain little or no relevant content and whose aim is solely to increase their position in the search engine rankings. Such spamming involves obtaining more exposure for a website than it really deserves for a given search term, leading to unsatisfactory search experiences. Hence, it is an important research issue to develop a technique to detect SEO spam websites.

By contrast, considerable attention has recently been devoted to investigating weblogs (or *blogs*) [9],[7]. Here, blogs are personal on-line diaries managed by easy-to-use software packages, and they have spread rapidly through the World Wide Web. Someone who keeps a blog is called a *blogger*, and a collection of blogs with their links is referred to as *blogspace*. A blog consists of entries that include text, images, hyperlinks, and *trackbacks*. Compared to ordinary websites, one of the most important features of blogs is the existence of trackbacks. Unlike a hyperlink, one blogger can construct a link from an entry of another blogger to his/her entry by creating a trackback on the entry. Thus, one can more easily create SEO spam entries in the trackback networks. In this paper, we explore a method for detecting SEO spams.

Considering the generating process of the SEO spam websites, we can assume that in a hyperlink network the SEO spams form several clusters consisting of *anomalously* dense node connections in each cluster. Thus, we try to detect cores each of which is defined as a set of nodes in which each member node has more links to nodes within the core than to nodes outside the core. For a large network, it becomes hard to detect such cores due to the combinatorial explosion. In this paper, we present a method for efficiently detect cores based on the graph spectral relaxation.

More specifically, in the proposed method the principal eigenvector of the adjacency matrix of the network is calculated, the nodes are ranked according to the element values of the eigenvector, and a set of densely connected nodes are detected as a spam core. By repeatedly performing the procedure after removing the links in the detected core, several *different* types of spam cores are also detected.

The graph spectral relaxation [3] has already been employed in the context of graph-based clustering [11]. However, since clustering tries to divide *all* nodes into sets, the conventional graph spectral relaxation algorithms (eg., *normalized cut* algorithm) are inappropriate for our purpose in which we want to detect spam *subsets* in a network. In other words, in this paper, we first derive another relaxation algorithm for *detecting* subsets of nodes in a graph. That is, one should note that detecting subsets of nodes in a network is essentially different from clustering nodes in a network [4] and [5].

In addition, from the viewpoint of the calculation of the principal eigenvector of the adjacency matrix of the network, the "HITS" algorithm [8] and the "PageRank" algorithm [2] are well known. However, these algorithms are for node (webpage) ranking algorithms and cannot directly detect cores.

[1] see, `http://www.webopedia.com/TERM/S/SEO.html`

Therefore, these algorithms are also inappropriate for our task.

We experimentally evaluate the detection performance of our method by using a real traceback network data in a *blogspace* and show that the proposed method could significantly outperform naive methods based on fundamental metrics on a network, Furthermore, we demonstrate that our method could actually detect different types of SEO spam entries.

2. DETECTION METHOD

In this section, we describe a method for detecting some core portions of a given network, having anomalously dense connections. This method calculates the principal eigenvector of the network adjacency matrix, ranks nodes (websites) according to the values of eigenvector elements, and then detects a core portion of densely connected nodes; after removing links in the detected core, the method repeatedly detects the other portions.

2.1 Basic problem

For a given network (graph), let $S = \{1, \cdots, N\}$ be a set of nodes (vertices), and $\mathbf{A}$ be its adjacency matrix. Namely, the (i, j) component of the adjacency matrix, denoted by $A(i, j)$, is set to 1 if there exists a link (edge) between nodes i and j; otherwise 0. In this paper, for simplicity we focus on undirected graphs without self-connections, i.e., $A(i, j) = A(j, i)$ and $A(i, i) = 0$. However, note that we can easily extend our framework to coping with directed graphs.

For any subset of nodes, $T \subset S$, we can define its average number of links as follows:

$$G(T) = \frac{1}{2} \sum_{i \in T} \sum_{j \in T} \frac{A(i, j)}{|T|}, \tag{1}$$

where $|T|$ stands for the number of elements in T. As described earlier, we consider detecting a subset of nodes T that maximizes Equation (1). However, in the case of a larger network, any straightforward method based on an exhaustive search is likely to suffer from combinatorial explosion. To cope with such a larger network, we focus on the following relaxation problem whose optimal solution is obtained as the principal eigenvector of the adjacency matrix.

2.2 Relaxation problem

For a subset of nodes T, we define an N dimensional indicator vector $\mathbf{q}$ by setting $q(i) = 1$ if $i \in T$; otherwise $q(i) = 0$. Then we can rewrite Equation (1) as follows:

$$G(\mathbf{q}) = \frac{1}{2} \frac{\mathbf{q}^T \mathbf{A} \mathbf{q}}{\mathbf{q}^T \mathbf{q}}, \tag{2}$$

where $\mathbf{q}^T$ stands for a transposed vector of $\mathbf{q}$. Now we consider a relaxation problem by letting $\mathbf{q}$ take continuous values. Then, according to the Rayleigh-Ritz theorem [6], the solution of maximizing $G(\mathbf{q})$ is given by the principal eigenvector $\mathbf{q}^*$ of the adjacency matrix $\mathbf{A}$.

In order to obtain the eigenvector $\mathbf{q}*$, we employ the following procedure based on the power iteration [6].

E1. Initialize $\mathbf{q}^{(0)} = (1, \cdots, 1)^T$, and set $t = 1$;

E2. calculate $\widetilde{\mathbf{q}} = \mathbf{A}\mathbf{q}^{(t-1)}$ and $\mathbf{q}^{(t)} = \widetilde{\mathbf{q}} / \max_i \widetilde{q}_i$;

E3. Terminate if $\max_i |q^{(t)}(i) - q^{(t-1)}(i)| < \epsilon$;

E4. Set $t = t + 1$, and return to **E2.**.

Here a small positive parameter ϵ controls the termination condition, and we can obtain the final solution as $\mathbf{q}^* = \mathbf{q}^{(t)}$ after its termination. Since all elements in $\mathbf{A}$ and $\mathbf{q}^{(0)}$ have non-negative values, we can guarantee that all values in $\widetilde{\mathbf{q}}$ are non-negative after any number of iterations. Moreover, due to the scaling operation in **E2**, we can guarantee that $0 \leq q^{(t)}(i) \leq 1$. Thus we consider that the above formulation gives one of desirable relaxation solutions to the original problem.

We note the computational complexity of the above procedure. Let L be the total number of links in the network. As for one-iteration, $\widetilde{\mathbf{q}}$ can be calculated within L additions because $A(i, j) \in \{0, 1\}$; the scaling operation in **E2** can be done within N multiplications.

2.3 Quantization problem

By ranking nodes according to the values of eigenvector elements, we can obtain a list of nodes, $R = [r(1), \cdots, r(N)]$, where $r(i)$ stands for a mapping function from ranks to nodes. Note that $q^*(r(i)) \geq q^*(r(i + 1))$. By considering a set of the top k nodes,

$$T(k) = \{r(i) : i \leq k\}. \tag{3}$$

we can calculate its average number of links as follows:

$$G(k) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{A(r(i), r(j))}{k}. \tag{4}$$

In our method, instead of directly solving Equation (1), we compute a node set $T(k^*)$ where k^* maximizes Equation (4).

In order to efficiently calculate k^*, we utilize the following update formula:

$$G(k + 1) = G(k) + \frac{\Delta(k + 1) - G(k)}{k + 1}, \tag{5}$$

where $\Delta(k + 1)$ stands for the increment by adding node $r(k + 1)$, calculated by

$$\Delta(k + 1) = \sum_{j=1}^{k} A(r(j), r(k + 1)). \tag{6}$$

Note that $G(1) = 0$. The above procedure can be summarized as follows.

F1. Compute $r(i)$ by sorting elements in $\mathbf{q}*$;

F2. Calculate $G(2), \cdots, G(N)$ by using Equations (5) and (6);

F3. Output $T(k^*)$ such that $k^* = \arg\max_k G(k)$;

Here we note the major computational complexity. In **F1**, sorting $\{q_i\}$ can be performed with $O(N \log N)$ computation. In **F2**, $G(2), \cdots, G(N)$ can be calculated by using Equation (6) within L additions, and by using Equation (5) within N multiplications.

2.4 Detection algorithm

By repeatedly performing the above procedures, M times, we can detect M core portions of a given network, having anomalously dense connections as follows.

G1. Repeat the following steps for m = 1 to M;

G2. Calculate $\mathbf{q}_m^*$ using **E1** to **E4**;

G3. Calculate T_m^* using **F1** to **F3**;

G4. Set $A(i,j) = 0$ if $i, j \in T_m^*$.

Here the number of cores, M, is determined by a user, and we can obtain the final result as $T_1^*, \cdots T_M^*$.

3. EXPERIMENTAL EVALUATION

Using real data of a trackback network in blogspace, we experimentally evaluate the performance of our method in comparison with a number of methods based on fundamental metrics on a network. We also demonstrate that our method is potentially able to separately detect SEO spam entries according to their types.

3.1 Data acquisition

Although there are a large number of blog entries in blogspace, many of them have no trackbacks. Namely, it is hard to obtain random samples of large connected trackback networks. Then, we exploited the blog "Theme salon of blogs [2]", where blog users can recruit trackbacks of other bloggers by registering interesting themes. By tracing ten steps ahead the trackbacks from the blog entries for a theme in the "Theme salon of blogs", we collected a large connected trackback network. Note that the entries in the network were not restricted to the theme first chosen due to frequent topic drifts, and thus had a variety of topics. Namely, we might consider that this collection procedure could produce a reasonably random sampling of a large connected trackback network from blogspace.

We treated blog entries that had been participants in certain well-known SEO contests in Japan as SEO spam entries. In these SEO contests, SEO devotees compete for search engine rankings in a search for a specified keyword such as "Goggole" or "Deskedgar", where these words are artifacts for the contests. We defined an entry as an SEO spam entry if it has the banner (link farm) "Trackback OK, Goggole", the banner "Trackback OK, Deskedgar", or one of the following keywords in the blogger name, entry name, or description section: "Goggole", "Deskedgar", "Nama sanargi", "Yahhyoi", "Ponesonic", and "Den-nou Purion".

3.2 Comparison methods

In order to evaluate the performance of our method, we consider a number of node ranking methods based on fundamental metrics on a network introduced in recent studies of complex network theory [12], [1], [10]. The *degree* d_i of a node i in a network is defined as the number of links attached to node i [1]. One naive strategy for raising the rankings of blog entries on search engines is to create many trackbacks to those blog entries. Thus, we can naively consider that SEO spam entries should have high degrees.

By contrast, we can also consider that the blog entries with which an SEO spam entry connects should have high degrees. Thus, as studied in [10], we investigate the average degree $\bar{k}_i$ among the nearest neighbors of an entry i in a trackback network. We call $\bar{d}_i$ the *average NN degree* of entry i. Let $\mathcal{N}_i$ be the neighborhood of a node i in a network, that is, the set of nodes that have links to node i. Then, $\bar{d}_i$ is defined by

$$\bar{d}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} d_j \ , \tag{7}$$

where $|\mathcal{N}_i|$ denotes the number of elements in the set $\mathcal{N}_i$.

The *clustering coefficient* C_i of a node i in a network is defined by

$$C_i = \frac{2b_i}{k_i(k_i - 1)} \ , \tag{8}$$

where b_i is the number of direct links connecting the nodes in the neighborhood $\mathcal{N}_i$ of node i [12]. Note that C_i reflects the probability that two friends of node i are friends themselves. We can naively consider that SEO spam entries should have high clustering coefficients in a trackback network.

We consider extracting the SEO spam entries from a given trackback network by ranking the entries in the network according to the level of SEO. Namely, we employ the following four ranking methods based on the metrics explained above. Let $e_1(i), e_2(i), e_3(i)$ and $e_4(i)$ be the evaluation functions of Methods 1, 2, 3 and 4, respectively, for measuring the SEO level of each node i. Then these functions are defined by $e_1(i) = C_i$, $e_2(i) = d_i$, $e_3(i) = C_i \log d_i$ and $e_4(i) = \bar{d}_i$.

Figure 1 shows the kind of node in a network that is regarded as having a high SEO level for each method, i.e., a node with a high clustering coefficient for Method 1 (see, Fig. 1 (1)); a node with a high degree for Method 2 (see, Fig. 1 (2)); a node that has both a high clustering coefficient and a high degree for Method 3 (see, Fig. 1 (3)); and a node such that its nearest neighbors have high degrees for Method 4 (see, Fig. 1 (4)). Note that since the magnitude of the degree is generally much larger than that of the clustering coefficient, we performed a logarithmic transformation on the degree for Method 3. Note also that we can regard our proposing method as a ranking method defined by $e(i) = q_1^*(i)$.

We quantified the performance of the proposed methods in terms of *F-measure* and *precision*, which are widely used in information retrieval. Let U denote the set of SEO spam entries in a trachback network. We fix a method for extracting the SEO spam entries from the network. For any positive integer K, let Z_K denote the set of the top K entries extracted by the method. Then, the F-measure $F(K)$ and the precision $P(K)$ of the method for ranking K are defined by

$$F(K) = \frac{2|Z_K \cap U|}{|Z_K| + |U|} \ , \quad P(K) = \frac{|Z_K \cap U|}{|Z_K|} \ . \tag{9}$$

Note that $F(K)$ quantifies how close the sets Z_K and U are. Note also that the higher the value $P(K)$ is, the lower the detection error is.

[2] http://blog.goo.ne.jp/usertheme/

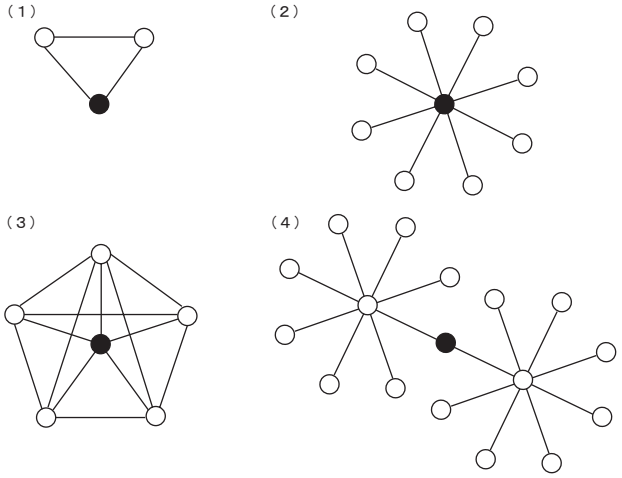

Figure 1: Examples of nodes (filled circles) with high SEO levels for the comparison methods.

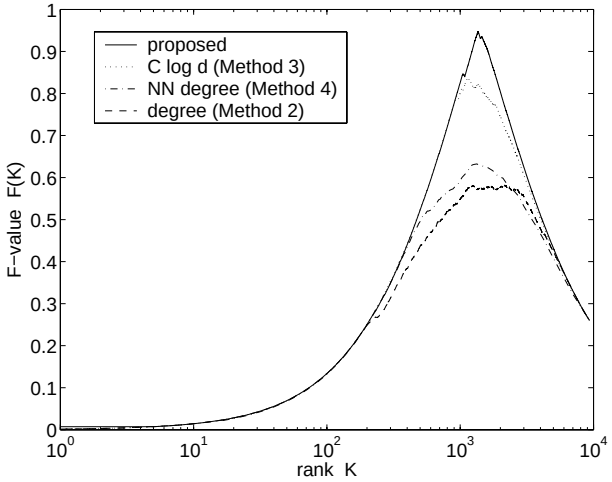

Figure 2: Performance comparison on F-measure.

Table 1: Measurement of the means of C_i, d_i and $\bar{d}_i$ for the set of SEO spam entries and the set of non-SEO spam entries.

	$\langle C_i \rangle$	$\langle d_i \rangle$	$\langle \bar{d}_i \rangle$
SEO spam	0.50317	63.833	176.97
Non-SEO spam	0.27830	6.4297	24.267

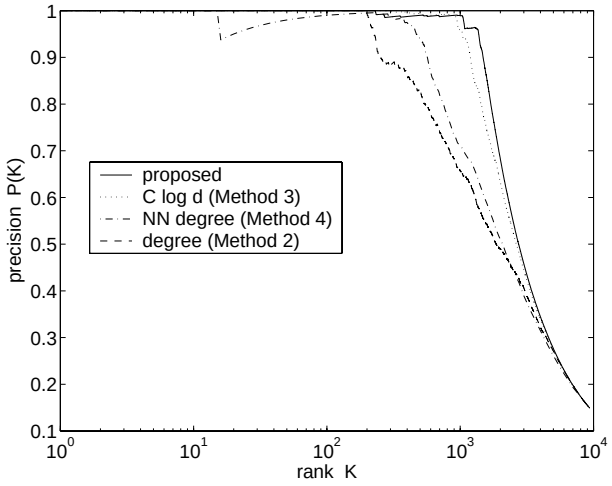

Figure 3: Performance comparison on the precision.

3.3 Performance evaluation

We describe our experimental results using data collected from the theme "Introduction of Special Sites" in the "Theme salon of blogs". Similar results were obtained by using data collected from other themes like "News for Smiling". Then, the total numbers of blog entries and trackbacks were 9,338 and 187,128, respectively. By our definition described above, the number of SEO spam entries was 1,395. Table 1 shows the fundamental statistics related to the proposed methods. Namely, the means of C_i, d_i and $\bar{d}_i$ are respectively displayed for the set of SEO spam entries and the others. Table 1 implies that the clustering coefficient, degree, and average NN degree of an SEO spam entry are generally larger than those of a non-SEO spam entry. Namely, these results justify applying the comparison methods.

Figures 2 and 3 respectively display F-measure $F(K)$ and precision $P(K)$ with respect to ranking K for the proposed method. Here we omitted the results of Method 1 because its performance was too bad. Recall that we applied our proposing method as a ranking method defined by $e(i) = q^*(i)$, where $\mathbf{q}_1^*$ means the principal eigenvector of the original adjacency matrix. Figure 2 shows that our proposing method provided the highest level of performance followed by Methods 3, 4 and 2. In particular, the F-measure of our method was extremely high with a value of over 90% around $K = 1,395$ (the number of SEO spam entries). Moreover, Figure 3 shows that our method was extremely precise. In particular, the value was 100% at $K = 289$ and over 90% around $K = 1,000$.

3.4 Multi-part detection

As mentioned earlier, we can detect multiple cores by using our method. In our experiment by setting the number of cores to 5 ($M = 5$), our method could separately detect SEO spam entries according to their types. More specifically, the entries in the first core exclusively had the keywords, "Nama sanargi" or "Den-nou Purion", which are closely relating to an ASCII-art community; those in the second and third cores only had the SEO keyword, "Deskedgar", used in the second Japaneses SEO contest; and those in the fourth one had the keywords, "Goggole", "Yahhyoi" or "Ponesonic", which are closely relating to the first Japaneses SEO contest. Incidentally, our method detected "adult blog entries" as the fifth core.

In order to evaluate how precisely our method can detect the types of spam entries, we performed further experiments by adding each entry a detailed label. More specifically, we separately examined the ranking performance of $\mathbf{q}_1^*$, $\mathbf{q}_2^*$ and $\mathbf{q}_4^*$,

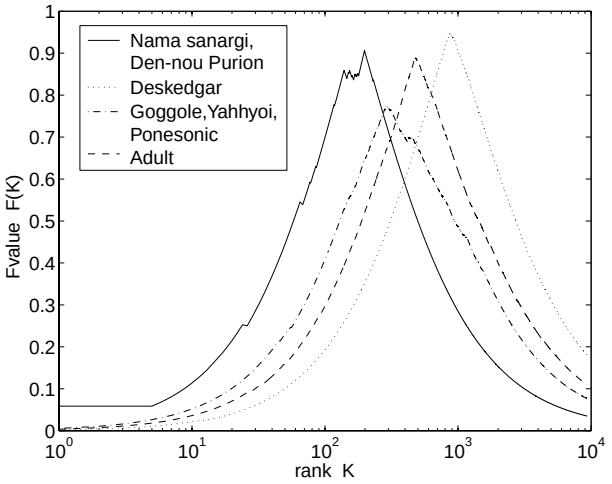

Figure 4: Performance evaluation on F-measure.

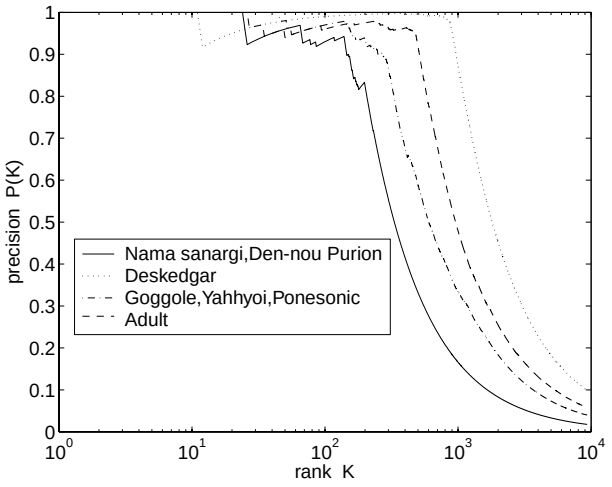

Figure 5: Performance evaluation on the precision.

with respect to three types of the entry labels, "Nama sanargi" or "Den-nou Purion", "Deskedgar", and "Goggole", "Yahhyoi" or "Ponesonic", respectively. Furthermore, by manually adding another label "adult blog entries", we also evaluated the ranking performance of $\mathbf{q}_5^*$. Figures 4 and 5 respectively display F-measure $F(K)$ and precision $P(K)$ with respect to ranking K for each pair of the eigenvector and its corresponding label. These figures show that our method could separately detect the types of spam entries with the high levels of performance. These results imply that our method can be a very promising approach for detecting the SEO spam entries from a trackback network.

4. CONCLUSION

We present a method based on an anomaly detection approach for filtering the SEO spam entries from a given trackback network. Using a connected trackback network collected by tracing ten steps ahead of the trackbacks from the blog entries for a theme in the "Theme salon of blogs", we experimentally demonstrated that the method can be a promising tool to filter SEO spam entries. We also demonstrated that our method could separately detect SEO spam entries according to their types.

By contrast, the next important task is to undertake an extensive verification of our method with various real blog data. To this end, we will need more sophisticated data collection processes. However, we have already made substantial progress, and we are encouraged by our initial results.

5. ADDITIONAL AUTHORS

Masahiro Kimura (Department of Electronics and Informatics, Ryukoku University), Kazuhiro Kazama (NTT Network Innovation Laboratories, NTT Corporation) and Shin-ya Sato (NTT Network Innovation Laboratories, NTT Corporation).

6. REFERENCES

[1] A.-L. Barabási and R. Albert, Emergence of scaling in random networks, *Science*, **286** (1999) 509–512.

[2] S. Brin and L. Page, The anatomy of a large scale hypertextual Web search engine, In *Proceedings of the Seventh International World Wide Web Conference* (1998) 107–117.

[3] F. Chung, Spectral Graph Theory, Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society (1997).

[4] G.W. Flake, S. Lawrence and C.L. Giles, Efficient identification of Web communities, In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2000) 150–160.

[5] M. Girvan and E.J. Newman, Community structure in social and biological networks, *Proceedings of the National Academy of Sciences of the United States of America*, **99** (2002) 7821–7826.

[6] G. H. Golub and C.F. Van Loan, Matrix Computations, Johns Hopkins University Press, Balimore, MD (1989).

[7] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, Information diffusion through blogspace, In *Proceedings of the 13th International World Wide Web Conference* (2004) 491–501.

[8] J. Kleinberg, Authoritative sources in a hyperlinked environment, In *Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithms* (1998) 668–677.

[9] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, On the bursty evolution of Blogspace, In *Proceedings of the 12th International World Wide Web Conference* (2003) 568–576.

[10] R. Pastor-Satorras, A. Vázquez, and A. Vespignani, Dynamical and correlation properties of the Internet, *Physical Review Letters*, **87** (2001) 258701.

[11] J. Shi and J. Malik, Normalized cuts and image segmentation, *IEEE Trans. PAMI*, **22(8)**, (2000) 888–905.

[12] D. J. Watts, and S. H. Strogatz, Collective dynamics of 'small-world' networks, *Nature*, **393** (1998) 440–442.

Multi-Stage Classification

Ted E. Senator
*DARPA/IPTO**
tsenator@darpa.mil

Abstract

While much research has focused on methods for evaluating and maximizing the accuracy of classifiers either individually or in ensembles, little effort has been devoted to analyzing how classifiers are typically deployed in practice. In many domains, classifiers are used as part of a multi-stage process that increases accuracy at the expense of more data collection and/or more processing resources as the likelihood of a positive class label increases. This paper systematically explores the tradeoffs inherent in constructing these multi-stage classifiers from a series of increasingly accurate and expensive individual classifiers, considering a variety of metrics such as accuracy, cost/benefit ratio, and lift. It suggests architectures appropriate for both independent instances and for highly linked data.

1. Introduction

The use of classifiers to detect extremely rare events is subject to the well-known and commonly pointed out pitfall that even with a highly accurate classifier, almost all positive classifications will be false positives. (For example, a 99.9% accurate classifier applied to a population of 300,000,000 entities containing only 3000 true positives – or 0.001 % – would yield 299,997 false positives and only 2997 true positives, corresponding to over 100 times more false positives than true positives, while failing to detect 3 actual positives.) Many potential applications of data mining have been criticized as infeasible because of this fact. These criticisms have been made by knowledgeable people in respected publications, in both the popular press and in scientific journals. (See [14] and [17] for two particular examples.) However, these criticisms are based on the assumption that such an application would consist of a single classifier operating on a single database and result in an unacceptably severe action for the entities that are classified "positive" – an assumption that is not valid or even close to valid for realistic examples of useful and deployed detection systems and that would not be employed by any reasonable designer or accepted by any reasonable user of such an application.

In contrast, real detection systems apply multiple stages of classification to a carefully selected corresponding series of databases, with each stage providing both evidence and justification for additional data collection, access and/or analysis in the subsequent stage. At each stage in the process, only those instances that have a positive classification are considered for the next stage. This drastically reduces the overall false positive rate, while opening up the possibility of additional false negatives.

However, a second characteristic of real domains mitigates this problem. Real domains of interest are characterized by strong linkages between entities. Following these linkages both enables the missed positive entities to be recovered and classified correctly through their connections to the correctly classified positive entities and also provides further reduction in false positives. (See [1], [7], and [12].) Finally, in the frequently occurring situations in which the phenomena of interest is characterized by combinations of entities, often with some amount of structure, these combined phenomena can be detected with adequate accuracy despite the rarity of the phenomena.

The data mining community and related communities have devoted much effort to techniques for creating better classifiers and, more recently, to techniques for combining individual classifiers to produce a more accurate combined classifier. (See [13] for reports on a series of workshops devoted to this subject and [11] for a recent book that is a comprehensive survey of the field.) However, this work has almost entirely focused at ensemble methods that combine classifiers in parallel, rather than at sequential combination of classifiers. (Minor exceptions are found in [20] and in the proposal for multilayered cascaded machines in [16]; however, these exceptions are still aimed at classifier construction rather than use, and do not consider use of additional data.) This existing research concentrates on the problem of efficiently constructing better classifiers that operate on a single shared data space rather than on the problem of developing effective applications given a set of classifiers

* The views and conclusions expressed in this paper are solely those of the author and should not be interpreted as representing the official policies, either expressed or implied, of DARPA, the Department of Defense, or the US Government. The research described in this paper was not conducted as part of any DARPA funded research project. The author's affiliation is provided for identification purposes only.

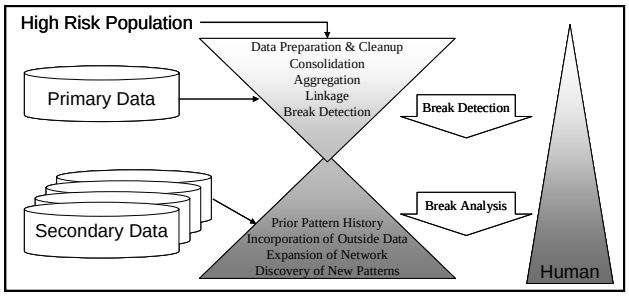

Figure 1: Break Detection Systems

and databases with specified performance characteristics and discriminative information content.

There has also been much recent work on constructing classifiers that use relational data. (See, for example, [3], [4], [9].) These collective classification techniques typically outperform classifiers that use only propositional data. The algorithms typically are iterative in so far as they propagate evidence of class labels between connected entities; however, these iterations are a convergence process that seeks a self-consistent labeling of a single dataset rather than a multi-stage classification process of the type described and analyzed in this paper. Other recent work has addressed the problem of data mining of graphical data. (See [2] for an overview.) Active learning is a technique that uses data incrementally; however, it is also aimed at constructing a classifier rather than at using a classifier to detect instances, and it selects examples from a single data source. Its incremental acquisition of specific labeled examples is distinct from the multi-stage classification of all data instances discussed in this paper.

The most thoughtful and thorough consideration of detection in linked data is [8]; it considers the implications of relational versus propositional data, ranking versus classification, and multi-pass versus single-pass inference. However, the form of multi-pass inference it considers – as it clearly notes – is one in which the predictions of one pass are used to inform the results of the next, until convergence occurs and a joint consistent probability model is achieved. This form of multi-pass inference is distinct from the form discussed in this paper, in which a series of independent classifiers, typically operating on different data, are used to filter a set of entities.

This paper presents, models, and evaluates architectures for constructing multi-stage classifiers to detect rare phenomena. It is based on architectures used in real detection systems in several domains. These architectures consist of various design alternatives for combining classifiers in series and for selecting population subsets to which they should be applied. It models alternative methods of choosing the classifiers and combining these classifiers in series and it evaluates tradeoffs between alternative approaches. It shows that feasible design alternatives exist for the construction of useful and practical detection systems for real phenomena of interest.

2. Detection Systems

Real detection systems consist of multiple stages. Based on the results of each stage, a decision is made regarding the further disposition of each entity under review. Two typical actions are 1) the acquisition and analysis of additional data about the entity to enable a reduction in its classification uncertainty and 2) the use of a more accurate and correspondingly more expensive classification test. While early stages may consist of entirely automated processes, later stages are typically characterized by an increased level of human involvement. The overall process consists not just of automated data analysis components, but also of human analyses, processes and procedures controlled by policies, and the overall controlling legal authorities. Human judgment – governed by management policies and legal authorities – is applied not only to the classification steps but to decisions about relative resource allocation between the steps and to issues such as thresholds and justifications for taking particular actions, such as acquiring additional information. These stages are highly interdependent – decisions made at one stage affect other stages. For example, the human resources available to conduct investigations limits the number of investigative leads desired to be produced by an automated classifier, thereby affecting the classification threshold that is employed and effectively choosing a particular setting on the classifier's possible ROC curve. This generic characterization describes many domains, including for example, public health, fraud detection, intelligence and law enforcement, to name but a few. A general model of these multi-stage detection systems was introduced in [5] and [19] and is depicted in figure 1.

2.1. Examples of Real Detection Systems

Many reports of real detection systems are available in the literature. The FinCEN Artificial Intelligence System (FAIS) [18] and NASD Regulation's Advanced Detection System (ADS) [10] are two. The commonality of their design is discussed in [19]. Primary data, i.e., the data that are used for initial classification, are prepared and cleaned. Entity consolidation is performed and models of the entities activities are created though aggregation. Initial break detection is performed by various classification techniques. After initial classification, additional data about positively classified entities are obtained from secondary sources. Secondary sources can only be queried with specific entity identifiers. Once the secondary source data are available, link analysis

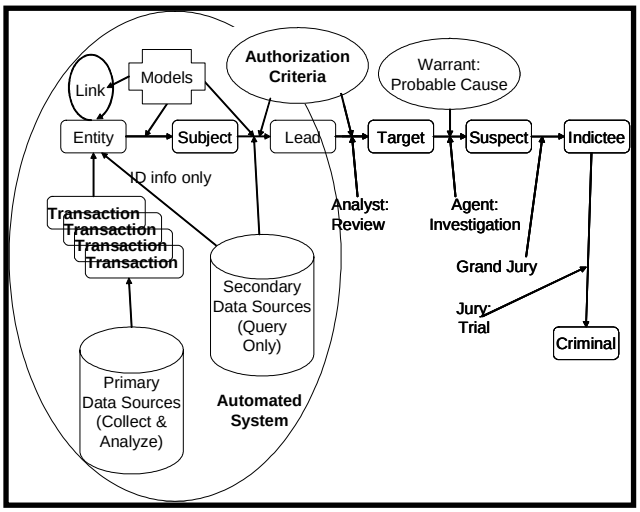

Figure 2: End-to-End Detection System

techniques are used to perform the final classification. The systems can be viewed as a series of filters (or classifiers), each of which provides increased accuracy on a correspondingly smaller set of entities, at a constant classification cost per stage. (Classification can be very expensive when it consists of complex pattern matching on a massive data stream.) Other detection systems, both existing [6] and proposed [15] also exhibit this structure.

2.2. Models of Detection Systems

Figure 2 illustrates a model of an end-to-end detection system in a law enforcement domain. It was developed for FAIS and it illustrates the design principles of multi-stage classification. It begins with "primary" data sources. These data sources are those that are considered to have a signal-to-noise ratio that can yield useful starting points for an investigation. Entities are inferred from the identification information provided with the transactions and from the linkages between transactions. Secondary data sources may be used for entity identification but not for classification at this point in the process. Two stages of automated classification are used. Stage 1 classifies some entities as "subjects. Based on this initial classification, additional information from the secondary sources may be queried for these specific entities and a stage 2 classification classifies some of the subjects as leads. These leads are reviewed by analysts; those that appear suspicious are regarded as targets and investigations are opened. It is at this point that the criterion of "probable cause" must be satisfied. Note that at all earlier stages other authorization criteria had to be met in order to "promote" an entity to the next level of suspicion. Some investigations lead to grand jury indictments, trials, and ultimately convictions.

3. Architectures

3.1. Motivation

As motivation we consider not only the examples presented in section 2 but also a completely different domain that exhibits the same general characteristics: the overall system used in the US for HIV detection.* This process is illustrative because it uses the same abstract strategy of multi-stage classification, even though the classification tests are based on biological samples rather than data analyses.

First, a high-risk population is identified. Routine HIV screening is a recommended procedure for individuals in this population. Screening is a low-cost procedure. The most common screening test is the enzyme-linked immunosorbent assay (ELISA). Individuals who test positive are then given a second, confirmatory test, typically the Western Blot. The screening test has a high sensitivity (few false negatives), while the confirmatory test has a high specificity (few false positives) but less sensitivity. Testing positive on the confirmatory test means one is infected with HIV. Notification of positive results from the screening test is not recommended in many circumstances because of the high false-positive rate. Note that membership in the high-risk population is determined by behavioral factors, while both the screening and confirmatory tests are performed on biological data; i.e., the data used by the classifiers (i.e., the screening and confirmatory tests) is independent of the criteria used to select individuals who are part of the high-risk group. It is also noteworthy that testing can be anonymous or confidential up through the confirmatory test, maintaining individual privacy.

Once HIV is confirmed in an individual, and only after it is confirmed, he is notified that he is infected. At this point, no more testing is needed to classify the individual as HIV-positive. However, the detection process does not end. Counselors encourage the HIV-positive individual to notify other individuals who he may have infected; this notification process leads to other infected individuals even if they are not included in the high-risk population or if they are one of the rare individuals who was falsely classified negative so they can be informed that they should be tested. It corresponds to the link analysis component of detection systems in other domains.

3.2. Architectural Models

We consider several increasingly complex architectures for a multi-stage detection system. The overall model structure is depicted in figure 3. Each specific instantiation is a different selection of components. We limit modeling and discussion to a high-

* Information regarding HIV testing is taken from www.cdc.gov.

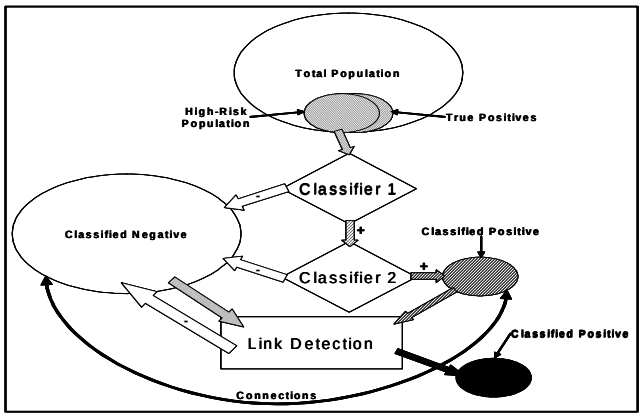

Figure 3: Model Components

risk population selection, two stages of classification, and one group detection calculation. Each classifier evaluates only the entities that have been classified as positive by the previous classifier. The errors from each stage are taken to be uncorrelated.

Extensions to additional classification stages would be straightforward. More important, two stages of classification is adequate in many single-source detection systems – once two classifications are performed, the entities that have been identified are few enough that the appropriate next stage is link analysis. Another view of this claim is that two stages of entity-based classification typically exhaust the information content of a single data source; to obtain more accurate classification then requires additional data sources rather than more sophisticated analyses.

The baseline architecture is a single stage classifier applied to an entire population. This baseline is the standard architecture on which individual classifiers are typically evaluated. Our first refinement of the baseline architecture is called the two-stage architecture. This two stage architecture employs two classifiers in series. If the errors in one of the classifiers arise from random events rather than from available data features, there might be reason to repeat an initial test, or, in our architecture, to employ the first stage classifier a second time on the same data, with no additional accuracy or cost. We call this the "two-first stage architecture." Likewise, given the availability of a more accurate second-stage classifier, perhaps there are advantages to using it twice. We call this the "two-second stage architecture." (Note that these two architectures make sense only if the classifiers operate on the same data.) We also consider using the more accurate classifier only once but on the entire population. This is referred to as the "all second-stage architecture." All of the above architectures can be preceded by the preprocessing step of selecting a high-risk population.

Finally, and perhaps most important, we consider the problem of detecting phenomena that can occur or be recognized only when groups are acting together. We imagine a group of size N and consider the simplest situation – that a group will be detected and its plans thwarted if at least one of its members is detected.

The architectures that are analyzed are summarized in Table 1. The X's in a box indicate that a model listed in the row includes the component named in the column.

Table 1: Architectural Model Components

Model	High-Risk Group	Stage 1 Classifier	Stage 2 Classifier	Group Detection
Baseline		X		X
Two-Stage		X	X	X
High-Risk	X	X		X
All Stage 2			X	X
Two Stage 1		X X		X
Two Stage 2			X X	X
High-Risk Two Stage	X	X	X	X

3.2.1. Model Parameters. Model input parameters are:

- Population Size
- True Positive Percentage
- High-Risk Percentage
- High-Risk Lift
- Stage 1 Test Sensitivity
- Stage 1 Test Specificity
- Stage 1 Cost
- Stage 2 Test Improvement/Cost Ratio
- False Negative Cost (= True Positive Benefit)
- Group Size

Most of the parameters are straightforward. High-Risk percentage is the percentage of the overall population in the high-risk group. High-risk lift is the additional likelihood that a member of the high-risk group is positive relative to the likelihood of an entity in the overall population. Care must be taken in selecting values for these parameters that they not be chosen so as to result in more positive entities in the high-risk group than in the whole population. The Stage 2 Test Improvement/Cost Ratio models the additional accuracy provided by the stage 2 test as well as the additional cost it is assumed to entail. To avoid introducing yet another parameter, we assume that the ratio of accuracy to cost is linear. The stage 1 cost is an arbitrary choice; other costs are expressed as a factor relative to it.

3.2.2. Computations. For each architecture included in table 1, the parameters described in section 3.2.1 are used to compute the expected values of the number of true positives and negatives in the population and, if it exists, in the high-risk group. Next, the expected number of true positives, true negatives, false positives, and false negatives is computed for the stage 1 classifier. Note that

in the architectures that incorporate a high-risk group, only the high-risk group is subject to the stage 1 classifier. Any positive exemplars not contained in the high risk group are counted as false negatives. Using the number of true positives and false positive as input to the stage 2 classifier, the expected number of true positives, true negatives, false positives, and false negatives is computed. The negatives resulting from this second stage of classification are added to those resulting from the first stage and those not tested at all to yield the final true positive, false positive, true negative, and false negative numbers for the entire population.

3.2.3. Cost-Benefit Modeling. The overall evaluation of a set of classifiers in an operational setting is performed as a cost-benefit analysis. Costs include the cost of performing the classification itself combined with the cost of any incorrect classifications. Benefits include not only the benefit of correct classifications but potentially the benefits of deterrence; i.e., changes in the behavior of potential adversaries, but modeling this effect is beyond the scope of this paper. Costs and benefits depend on distribution in population as well as on classifier performance.

3.2.4. Assumptions and Limitations. The analyses of alternative architectures for multi-stage classification described in this paper are limited by the assumptions and limitations of the models, which include:

- Particular choices for parameter settings
- Calculations are performed for expected values only; variances are not evaluated.
- The high-risk population criteria and the stage 1 and stage 2 classifiers are conditionally independent (i.e., errors are uncorrelated)
- Classifier cost and accuracy improvements scale linearly.
- Groups are treated as homogeneous collections of entities
- In-group connectivity is treated as 100% existing and observable, no cross-group connectivity is modeled.

3.3. Metrics

Different fields use different metrics for describing classifier quality. While related, these metrics are not identical. Information retrieval is typically evaluated by precision and recall. Pattern recognition or target detection systems are measured by Receiver-Operating Characteristics (ROC curves). Public health professionals report positive predictive value and negative predictive value of a diagnostic test as a function of sensitivity and specificity and the characteristics of a population. The relationship between these measures is depicted in table 2 and by the following equations:

PPV = Precision = TP / (TP + FP)
NPV = TN / (FN + TN)
P(detection) = Sensitivity = Recall = TP / (TP + FN)
P(false alarm) = FP / (FP + TN)
Specificity = TN / (FP + TN) = 1 – P(false alarm)

Table 2: Metrics

		ACTUAL (Population)	
TEST (CLASSIFIER)		**Positive**	**Negative**
	Positive	True Positives (TP)	False Positives (FP)
	Negative	False Negatives (FN)	True Negatives (TN)

These are point metrics. Because one often thinks of tuning a classifier to reflect a tradeoff between False Positives and False Negatives, and because many point metrics depend not only on classifier characteristics but also on the distribution of classes in the population, which may not be known *a priori*, ROC curves and, in particular, the area under the (ROC) curve, are considered better methods of comparing classifiers. However, in practice, tuneable classifiers are rarely available to application developers, especially when each classifier is designed for different datasets. Rather, they embody a particular decision rule operating at a particular point on an ROC curve. For the purposes of this paper, we specify our classifiers' performance in terms of Sensitivity and Specificity, metrics that are independent of the distribution of class labels. We compute PPV, NPV, P(detection) and P(false alarm) for each of the architectural alternatives. We also compute lift, defined as the improvement in signal/noise ratio obtained through classification, or more precisely as the ratio of true positives to total positives resulting from the classification process compared to the fraction of true positives in the overall population. Finally, we report the total error (overall % misclassified) and the computed benefit/cost ratio. For groups we compute the % of groups that would be detected under the assumption that detecting at least one group member results in the group's being detected, the expected number of groups that are not detected (i.e., false negatives), and the associated benefit/cost ratio.

Even though a classifier can be viewed as a combination of a scorer and a threshold, in a multi-stage architecture a binary decision must be made as to whether or not a particular entity is carried forward to a subsequent stage. Therefore, we consider only binary classifiers in this paper with no loss of generality.[*]

[*] An interesting architectural alternative that we did not analyze here is one that is governed by an externally specified limit on the number of positively classified examples that could be handled. This would reflect the common real-world situation in which downstream human analytical processes govern the number of leads that can be pursued.

4. Experimental Results

Alternative parameter settings were chosen and the various architectures described in section 3 were compared under a variety (over 25) of parameter settings. A population of 300,000,000 (roughly the population of the US, to one significant figure) was used for all the experiments. Somewhat realistic parameters chosen to characterize counter-terrorism detection (line T0) and HIV detection (line HIV0) are presented in table 3, with corresponding results in table 4.

Table 3. Input Parameters

Label	True % Positive	High-Risk %	High-Risk Lift	Stage 1 Sensitivity	Stage 1 Specificity	Stage 2 Improvement / Cost Factor	False Negative Cost	Group Size
T0	0.001	5	10	75%	95%	10	100	6
HIV0	0.4	8	10	99%	95%	10	1	6

Table 4. Results: T0

ARCHITECTURE / **OUTPUTS**	**Baseline**	**Two-Stage**	**High-Risk**	**All Stage 2**	**Two Stage 1**	**Two Stage 2**	**High-Risk 2 Stage**
PPV %	0.015	2.84	0.15	0.19	0.22	27.55	22.63
NPV %	99.99						
P(detection) %	75	73	38	98	56	95	37
P(false alarm)	5	0.025	0.25	0.5	0.25	0.0025	0.00125
Lift	15	2842	150	195	224	27550	22632
Total Error %	5	0.025	0.25	0.50	0.25	0.0025	0.0019
Benefit/Cost	0.0012	0.0048	0.012	0.0009	0.0043	0.009	0.044
GROUP DETECTION							
% Groups Detected	99.98	99.96	94.04	100.0	99.30	100.0	93.48
Number of Groups Missed	0	0	30	0	4	0	33
Group Benefit/Cost	4095	2653	16	4M	142	69M	14

Table 5. Results: HIV0

ARCHITECTURE / **OUTPUTS**	**Baseline**	**Two-Stage**	**High-Risk**	**All Stage 2**	**Two Stage 1**	**Two Stage 2**	**High-Risk 2 Stage**
PPV %	7.37	94.1	45.2	44.5	61.2	99.4	99.4
NPV %	99.99	99.99	99.92	99.99	99.99	99.99	99.92
P(detection) %	99.0	98.9	79.2	99.9	98.0	99.8	79.1
P(false alarm)	5	0.025	0.39	0.50	0.25	0.0025	0.002
Lift	18	235	113	111	153	248	248
Total Error %	5.0	0.029	0.47	0.50	0.26	0.0033	0.086
Benefit/Cost	0.66	2.46	2.44	0.38	2.84	3.63	3.21
GROUP DETECTION							
% Groups Detected	100	100	99.99	100	100	100	99.99
Number of Groups Missed	0	0	16	0	4	0	17
Group Benefit/Cost	999B	568B	12k	Inf.	16B	Inf.	12k

We next vary selected parameters from these base cases to examine the effect of the parameter choices. Because of the large number of degrees of freedom, even in this simplified model, it is not feasible to explore the entire state space at once.

The most important result is the number of groups missed. Using the T0 parameters and varying group size, the number of undetected groups is shown in Table 6. Even small increases in group size almost guarantee that at least one member of every group will be detected.

Table 6: Undetected Groups

group size	**Baseline**	**2-Stage**	**High-Risk**	**All Stage 2**	**Two Stage 1**	**Two Stage 2**	**High-Risk 2 Stage**
1	750	806	1875	75	1313	148	1903
2	94	108	586	1	287	4	604
3	16	19	244	0	84	0	255
4	3	4	114	0	27	0	121
5	1	1	57	0	10	0	62
6	0	0	30	0	4	0	33
7	0	0	16	0	1	0	18
8	0	0	9	0	1	0	10
9	0	0	5	0	0	0	6
10	0	0	3	0	0	0	3
11	0	0	2	0	0	0	2
12	0	0	1	0	0	0	1
13	0	0	1	0	0	0	1
14	0	0	0	0	0	0	0

The results of even a high-risk two stage detection system would be used as the basis for further analysis in a sensitive domain such as counter-terrorism detection, in which the goal of these stages is to enrich the data to maximize productivity of the downstream processes. (This is similar to the law enforcement model presented in figure 2.) Figure 4 depicts results obtained from examining the effect of varying stage 1 specificity with sensitivity fixed at 75%; figure 5 depicts the result of varying sensitivity with specificity fixed at 95%. Figure 5 examines the effect of the Stage 2 Improvement/Cost Factor. All other parameters are as in T0.

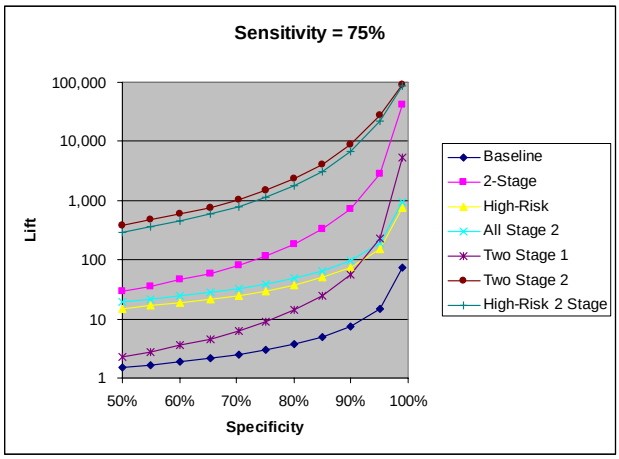

Figure 4

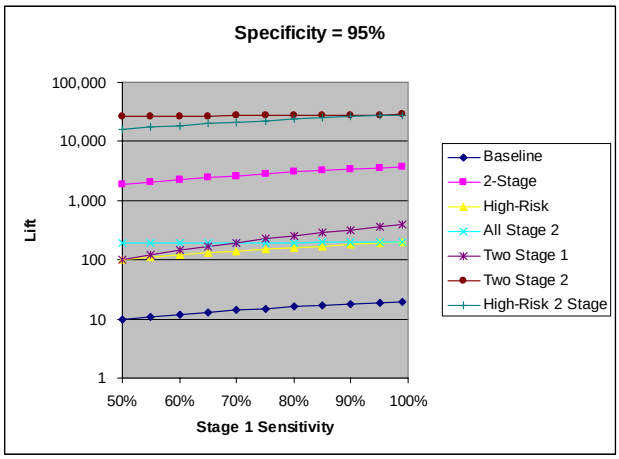

Figure 5

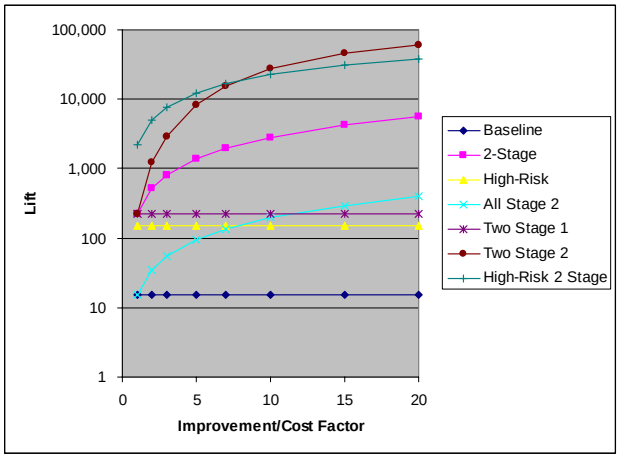

Figure 6

5. Conclusions

Comparisons of the different architectures suggest that the full multi-stage classification architecture that takes advantage of the high-risk population and two classification stages is especially advantageous for extremely rare phenomena. Significant contributions to lift arise from the use of the more accurate classifier in the second stage. Most important, for groups whose actions depend on having all participants available, it is well within the realm of feasibility to disrupt them. However, detecting an entity in isolation would be difficult even with the proposed multi-stage classification techniques. Initial segmentation into a high-risk population risks missing some smaller groups but provides an even greater reduction in false positives. Maximum lift occurs over a range of sensitivity and specificity for the two-stage 2 and the high-risk 2 stage architectures.

The initial results presented here suggest that multi-stage classification is a feasible design for the initial stages of detecting rare events, especially when there are strong and observable links between entities that can be used to compensate for false negatives. Multi-stage classification techniques can significantly reduce – to acceptable levels – the otherwise unacceptably large number of false positives that would result from even the most optimistically accurate single-stage classifier applied to very rare phenomena. It can eliminate most entities from suspicion early on, at a low cost in data collection and in testing, with acceptable impacts on overall detection effectiveness. For complex phenomena characterized by the necessity of all participants' not being detected, multi-stage classification may by itself be sufficient to disrupt, and therefore prevent, the phenomena that is the subject of the detection process. And for phenomena for which links are not readily observable or that require a larger proportion of the participating entities to be detected, multi-stage classification can provide the initial leads for collective inference and link analysis techniques. Starting from known examples and following the links is a feasible approach to complex event detection, but so is multi-stage classification applied to individual entities. In combination it is likely that a wide range of practical and desirable applications can be designed and constructed.

To return to the example that introduces this paper, multi-stage classification for a phenomena present only 0.001% of the time, using two independent stages at 99% and 99.9% accuracy and with 5% of the population in a high-risk group that was 10 times more likely to be positive, would be expected to detect almost all the groups with minimal false positives.

6. Future Work

This work could be extended in many directions. Perhaps the most interesting would be a comprehensive examination of the parameter space, providing an empirical sensitivity analysis. Related to this would be the use of Monte Carlo type simulations instead of Expected

Value Models to explore the variance of the system characteristics in addition to the mean. Exploring alternative models of the cost of additional accuracy in the second stage classifier has been suggested. Allowing more than two classification stages – i.e., modeling the availability of multiple data sources that can be accessed only when previous analyses suggest some likelihood of a positive classification – would be a natural extension. The conditional independence assumption between the classifiers could be weakened. More sophisticated group detection models would also be of interest, as would consideration of a distribution of group sizes and explicit modeling of link observability and link existence probabilities. Explicitly considering resource constraints at each stage of classification would model the reality of many organizations. Finally, combining the multi-stage classification architectures described in this paper with the relational classification models evaluated in [8] would be a major step towards a comprehensive understanding of a complete detection system for extremely rare events in complex relational domains.

7. Acknowledgements

I thank David Jensen, Foster Provost, and Raghu Ramakrishnan for their suggestions and ideas and for encouraging me to pursue this research and prepare this paper. I also thank my former colleagues at NASD and at FinCEN for helping to develop some of the ideas expressed in this paper over the course of many years of developing, deploying, and using break detection systems.

8. References

[1] Adibi, J., Chalupsky, H., Grobelnik, M., Milic-Frayling, N., and Mladenic, D. (Eds.) *Second International Workshop on Link Analysis and Group Detection (LinkKDD-2004)*. (Seattle, WA, USA, August 22, 2004).

[2] Dzeroski, S., De Raedt, L. Multi-relational data mining: the current frontiers. *ACM SIDKDD Explorations Newsletter, 5, 1* (July 2003), 1-16.

[3] Friedman, N., Getoor, L., Koller, D., and Pfeffer, A. Learning Probabilistic Relational Models. In *Relational Data Mining*, S. Dvzeroski and N. Lavrac (Eds.), Springer-Verlag, 307-337, 2001.

[4] Getoor, L. and Jensen, D. (Eds.) *Learning Statistical Models from Relational Data: Papers from the AAAI 2000 Workshop*, AAAI Press, Menlo Park, CA, 2000.

[5] Goldberg, H.G., and Senator, T.E. Break Detection Systems. In *AI Approaches to Fraud Detection and Risk Management: Collected Papers from the 1997 Workshop* Technical Report WS-97-07 AAAI Press, Menlo Park, CA.

[6] Goldberg, H., Kirkland, D., Lee, D., Shyr, P., and Thakker, D. The NASD Securities Observation, News Analysis & Regulation System (SONAR). In Proceedings of the Fifteenth Innovative Applications of Artificial Intelligence Conference (IAAI-2003). (Acapulco, MX, August 12-14, 2003.) AAAI Press, Menlo Park, CA, 2003. 11-18.

[7] Jensen, D. and Goldberg, H. *Artificial Intelligence and Link Analysis: Papers from the 1998 AAAI Fall Symposium*, AAAI Press, Menlo Park, CA, 1998.

[8] Jensen, D., Rattigan, M., and Blau, H. Information Awareness: A Prospective Technical Assessment. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*. (Washington, DC, USA, August 24-27, 2003). ACM Press, New York, NY, 2003, 378-387.

[9] Jensen, D. and Neville, J. Data Mining in Social Networks. *Papers of the Symposium on Dynamic Social Network Modeling and Analysis.* (National Academy of Sciences. November 7-9, 2002). National Academy Press, Washington, DC, 2002.

[10] Kirkland, J., Senator, T., Hayden, J., Dybala, T., Goldberg, H., and Shyr, P. The NASD Regulation Advanced Detection System (ADS). *AI Magazine, 20, 1* (Spring 1999), 55-67.

[11] Kuncheva, Ludmila I. *Combining Pattern Classifiers: Methods and Algorithns.* John Wiley & Sons, Inc, Hoboken, NJ, 2004.

[12] Mladenic, D., Grobelnik, M., Milic-Frayling, N., Donoho, S., and Dybala, T. (Eds.) *Workshop on Link Analysis for Detecting Complex Behavior (LinkKDD2003)* KDD2003, (Washington, DC, USA, August 2003).

[13] Multiple Classifier Systems: First/Second/Third/Fourth/Fifth International Workshop (MCS-2000/2001/2002/2003/2004) Springer-Verlag GmbH, 2000-2004.

[14] Paulos, J., Do the Math: Rooting Out Terrorists is Tricky Business. *Los Angeles Times*, January 23, 2003.

[15] Popp, R., Armour, T., Senator, T., and Numrych, K. Countering Terrorism Through Information Technology. *Communications of the ACM, 47, 3* (March 2004), 36-43.

[16] Sebestyen, George S. *Decision-Making Processes in Pattern Recognition*. The Macmillan Company, New York, NY, 1962.

[17] Scientific American (editorial). Total information overload. *Scientific American*, March 2003, 12.

[18] Senator, T. E., Goldberg, H., Wooton, J., Cottini, A., Khan, A. U., Klinger, C., Llamas, W., Marrone, M., and Wong, R. The FinCEN Artificial Intelligence System: Identifying Potential Money Laundering from Reports of Large Cash Transactions. *AI Magazine, 16, 4* (Winter 1995), 21-39.

[19] Senator, T. E., and Goldberg, H. Break Detection Systems. In *Handbook of Data Mining and Knowledge Discovery*, W. Klosgen and J. Zytkow (eds.), Oxford University Press. 863-873, 2002.

[20] Yang, S., Browne, A., Picton, P. D., Multistage Neural Network Ensembles, In *Multiple Classifier Systems: Third International Workshop (MCS 2002) Lecture Notes in Computer Science, 2364*. (Cagliari, Italy, June 24-26, 2002). Springer-Verlag, 91-97

Current and Potential Statistical Methods for Anomaly Detection in Modern Time Series Data: The Case of Biosurveillance

Galit Shmueli
Department of Decision & Information Technologies
Robert H Smith School of Business
University of Maryland, College Park
+1 301 405 9679
gshmueli@umd.edu

ABSTRACT
Modern biosurveillance is concerned with the early detection of natural disease outbreaks and those following bio-terror attacks by monitoring syndromic data. We describe the various challenges that arise in detecting disease outbreaks of unknown patterns in temporal data that come from multiple sources. Current practice is to use statistical monitoring tools that make assumptions about the data structure and anomalous patterns that are hard to justify. Similar problems in other fields (e.g. geophysics, chemical engineering) have led to more advanced monitoring methods. We survey some of these methods and their potential for biosurveillance.

Categories and Subject Descriptors
A.0, C.3, C.4, G.1.2, I.2.1, I.5.1, J.3

General Terms
Algorithms, Measurement, Performance.

Keywords
Disease outbreaks, bioterrorism, syndromic data

1. BIOSURVEILLANCE
The goal of modern bio-surveillance systems is the rapid detection of a disease outbreak related to a "natural cause" or a bio-terror attack. To achieve this goal, data are routinely collected from multiple sources on multiple data streams that are considered to carry early signals of an outbreak. Data range from traditional sources such as lab results and hospital visits, to non-traditional "syndromic" sources such as pharmacy medication sales and medical website activity. Such data tend to be frequent (at least daily), and can be vary widely within a data source, and even more so across data sources. There are various statistical issues that are related to the collection, transfer, and storage of such data (see [6] for details], but here we focus on the data analysis stage.

1.1 Structure of Data and Anomalies
Current surveillance methods rely mostly on traditional statistical monitoring methods such as statistical process control and autoregressive time series models. However, these methods are not always suitable for monitoring non-traditional bio-surveillance data. Assumptions such as normality, independence, and stationarity are the backbone of such methods, whereas the types of data that are monitored in bio-surveillance almost always violate these assumptions. Furthermore, outbreak signatures in such data are of unknown patterns, and therefore methods that are tuned to particular anomaly patterns (such as classic control charts) become much less powerful. This situation is now becoming common in many other fields, where data have become much more accessible, frequent, and diverse.

1.2 Evaluating Performance
Another main issue is the absence of clear "gold standards": (1) It always remains unclear whether a natural disease outbreak is present in a dataset, and even if it is believed to exist, it is hard to determine its exact dates. (2) There are no datasets of this type that contain true bioterrorist attacks. Thus, it is hard to establish performance measures and to compare algorithms.
Finally, since multiple data streams (from within and across multiple sources) are being monitored concurrently, it is likely that multivariate methods will outperform univariate methods for detecting in earliness of detection.

2. CURRENT PRACTICE
Temporal monitoring is typically done at the univariate level, using classical statistical control charts such as Shewhart charts, Cumulative Sum (CuSum) charts and Exponentially Weighted Moving Averages (EWMA) charts. These are used to monitor series such as daily counts of visitors to emergency departments with upper respiratory complaints; daily sales of over the counter cough medications, etc. These monitoring tools assume that the data are temporally uncorrelated, are stationary, and follow a normal distribution. Obviously, these assumptions can be shown to be violated. The result is increased false alarms and generally deteriorated performance. Empirical studies show alarm rates that are much higher than the level the charts are set to achieve. Researchers therefore tend to choose alarm thresholds according to the performance of historic data [Wagner]. However, because of the non-stationary nature of the data, this is a risky practice.

There have been attempts to use modified versions of classical control charts that account for autocorrelation through fitting a time series model to the data and monitoring the residuals (e.g., [16]) and/or non-normality. Adjustment for seasonality is typically done through a regression-type model introduced by Serfling ([17]). However, the stationarity assumption really limits the usefulness of such methods. One attempt to tackle the stationarity issue is through "adaptive" charts, where the alarm thresholds are based not on the usual "phase I" stage, a historic

period that is assumed to be clean of anomalies, but instead on a moving window of the most recent month ([12]).

Another feature of these classical charts is that they are most efficient at detecting an anomaly pattern of a certain type ([3]). For instance, simple Shewhart types are best for detecting a large spike. A CuSum chart is best at detecting a small/medium step change. A EWMA chart is best for detecting an exponential increase/decrease. In almost all cases we do not know how the disease outbreak will manifest itself in these non-traditional data: what will be the shape and magnitude of a local anthrax attack in sales of flu-like remedies? In addition to the physical progression of the disease in the body and in the population other factors such as mass psychology, level of bio-agent, and patterns of sales will affect what we see. Therefore, methods that are "general detectors" appear to be more attractive if we are interested in finding patterns of unknown nature.

3. ADVANCED MONITORING

Monitoring methods have been developed and used in fields such as machine learning, computer science, geophysics, and chemical engineering. Also forecasting, which is related to monitoring, has had advances in fields such as finance and economics. In these fields there exist a wealth of very frequent autocorrelated data, the goal is the rapid detection of abnormalities or forecasting, and the developed algorithms are flexible and computationally efficient.

We give a short description three methods used in other fields that can be adapted for monitoring, and explain why they are potentially useful for biosurveillance. For a detailed description of the different methods and other methods see [18].

3.1 Exponential Smoothing

Exponential smoothing (ES) is a class of methods that is very widely used in practice (e.g., for production planning, inventory control, and marketing ([14])) but not so in the biosurveillance field. ES has gained popularity mostly because of its usefulness as a short-term forecasting tool. Empirical research by [11] has shown simple exponential smoothing (SES) to be the best choice for one-step-ahead forecasting, from among 24 other time series methods and using a variety of accuracy measures. Although the goal in biosurveillance is not forecasting, ES methods are relevant because they can be formulated as models ([4]). Non-traditional biosurveillance data include economic series such as sales of medications, health-care products, and grocery items. Since trends, cycles, and seasonality are normally present in sales data, more advanced ES models have been developed to accommodate non-stationary time series with additive multiplicative seasonality and/or linear/exponential/dampened trend components. The advantage of these models is their simplicity of implementation and interpretation, their flexibility for handling many types of series, and their suitability for automation ([5]).

Although research and application of univariate exponential smoothing is wide-spread there is a surprising scant number of papers on multivariate exponential smoothing, as a generalization of the univariate exponential smoothing methods.
The main challenge in moving to a multivariate setting is the specification of the smoothing matrices and initial values for the different components. This requires a distributional assumption or prior subjective judgments (which are much harder in a multivariate setting). Once specified, this process need not be repeated. Furthermore, once specified, the estimated smoothing matrices can shed light on the cross-relationships between the different time series in terms of seasonal, trend, and level components.

3.2 Wavelet-Based Methods

A promising method that is suitable for detecting aberrations of unknown nature is based on wavelet decomposition of data streams. Such methods are also advantageous because they are suitable for series that exhibit autocorrelation, and even non-stationarity. The idea behind wavelet methods is to compose a series into multiple scales and then monitor the different scales for aberrations.

Wavelets are a method for representing a time series in terms of coefficients that are associated with a particular time and a particular frequency ([13]). The wavelet decomposition is widely used in the signal processing world for denoising signals and recovering underlying structure. Unlike other popular types of decompositions, like the Fourier transform, the wavelet decomposition yields localized components. Wavelet decompositions have proven especially useful in applications where the series of interest is not stationary. This includes long-range dependent processes which include many naturally occurring phenomena such as river flow, atmospheric patterns, telecommunications, astronomy, and financial markets ([7]). Also, the wavelet decomposition highlights inhomogeneity in the variance of a series. Furthermore, data from most practical processes are inherently multiscale due to events occurring with different localizations in time, space, and frequency ([1]).

DWT can be used for both retrospective monitoring and prospective monitoring. In both cases there are several issues and challenges that must be addressed such as downsampling (that causes delays in detection), boundary extrapolations, and multiple testing. Goldenberg et al. [6] used a wavelet-based method to monitor the sales of OTC cough medications at a large grocery chain, and showed the timeliness of detection using simulated outbreak patterns. A few others have also used wavelets in some form for bio-surveillance (e.g., [20]). A survey of wavelet-monitoring issues and some recommendations can be found in [18].

A multivariate wavelet based monitoring method was suggested by [2]. They perform DWT on each series and apply principal components analysis (PCA) to the coefficients of a certain scale across series. These principal components are then monitored using a T^2 chart and Q chart for detecting abnormal coefficients.

There is room for developing other multivariate DWT-based monitoring tools that will address the challenges and issues that arise in the biosurveillance context.

3.3 Singular Spectral Analysis

Singular Spectral Analysis (SSA) is used in the geosciences for monitoring climatic time series. It is suitable for decomposing a short, noisy time series into a (variable) trend, periodic oscillations, other aperiodic components, and noise ([15]). The method uses principal components analysis (PCA) of the M-lag autocorrelation matrix, and then reconstructs the signal from a

subset of principal components. SSA is used mostly for revealing the underlying components of a time series and separating signal from noise. However, it can be used for forecasting by using low-order autoregressive models for the separate reconstructed series ([15]). This means that SSA can be used for biosurveillance and monitoring in general by computing one-step-ahead forecasts and comparing them to the actual data. If the distance between a forecast and an actual observation is too large, a signal is triggered.

A generalization of SSA to multivariate time series, called multichannel-SSA (M-SSA), was described by [8] and applied to several climate series. The idea is similar to the univariate SSA, except that now the lag-covariance matrix is a block-Toeplitz matrix T, where Tij is an MxM lag-covariance matrix between series i and series j.

SSA's good performance in monitoring climatic data for unusual events (associated with El Nino) encourages its consideration for biosurveillance because of the core similarities in data and tasks: climatic and syndromic data share components such as weekly, seasonal, and annual patterns; the method is relatively insensitive to the stationarity assumption; and finally, the ability to generalize it to the analysis of multiple time series (although computationally challenging) is useful not only for monitoring purposes but also for shedding light on the cross-relationship between different biosurveillance series, both within a data source and perhaps even across data sources. The SSA-MTM toolkit is a software package for applying M-SSA (and other techniques), and is freely available at http://www.atmos.ucla.edu/tcd/ssa/.

4. ACKNOWLEDGMENTS

This paper is based on joint work with Prof. Stephen E. Fienberg, Carnegie Mellon University.

5. REFERENCES

[1] Aradhye, H. B., Bakshi, B. R., Strauss, R. A., and F, D. J. Multiscale statistical process control using wavelets - theoretical analysis and properties. *AIChE Journal*, 49(4):939-958. 2003.

[2] Bakshi, B. R. Multiscale pca with application to multivariate statistical process monitoring. *AIChE Journal*, 44:1596-1610, 1998.

[3] Box, G. and Luceo, A. Statistical Control: By Monitoring and Feedback Adjustment. Wiley-Interscience, 1st edition, 1997.

[4] Chatfield, C., Koehler, A., Ord, J., and Snyder, R. A new look at models for exponential smoothing. *The Statistician, Journal of the Royal Statistical Society – Series D*, 50(2):147-159. 2001.

[5] Chatfield, C. and Yar, M. Holt-winters forecasting: Some practical issues. *The Statistician*, 37:129-140. 1988

[6] Fienberg S. E.., and Shmueli, G. Statistical Issues and Challenges Associated with Rapid Detection of Bio-terrorist Attacks *Statistics in Medicine.* 24 (4), 513-529, 2005.

[7] Gencay, R., Selcuk, F., and Whitcher, B. An Introduction to Wavelets and Other Filtering Methods in Finance and Economics. Academic Press, 2001.

[8] Ghil, M. and Yiou, P. Decadal Climate Variability: Dynamics and Predictability, chapter in *Spectral methods: What they can and cannot do for climatic time series*, pages 445-482. Elsevier, Amsterdam. 1996.

[9] Goldenberg, A., Shmueli, G., Caruana, R. A., and Fienberg, S. E. Early Statistical Detection of Anthrax Outbreaks by Tracking Over-the-Counter Medication Sales. *Proceeding of the National Academy of Sciences*, vol. 99, Issue 8, pp. 5237-5240, 2002.

[10] Ivanov, O., Gesteland, P. H., Hogan, W., Mundorf M. B., and Wagner, M. M. Detection of pediatric respiratory and gastrointestinal outbreaks from free-text chief complaints. In *AMIA Annual Symposium Proceedings*, number 318-322. 2003.

[11] Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., and Winkler, R. The accuracy of extrapolative (time series methods): Results of a forecasting competition. *Journal of Forecasting*, 1(2):111-153. 1982.

[12] Murphy, S., Burkom, H., and Shmueli, G. Data-Adaptive Multivariate Control Charts for Routine Health Monitoring, *Work in Progress*.

[13] Percival, D. and Walden, A. *Wavelet Methods for Time Series Analysis.* Cambridge University Press, Cambridge, U.K. 2000.

[14] Pfeffermann, D. and Allon, J. Multivariate exponential smoothing: Method and practice. *International Journal of Forecasting*, 5(1):83-98. 1989.

[15] Plaut, G., Ghil, M., and Vautard, R Interannual and interdecadal variability in 335 years of central England temperatures. *Science*, 268(5211):710-713. 1995.

[16] Reis, B. and Mandl, K. Time series modeling for syndromic surveillance. *BMC Medical Informatics and Decision Making*, 3(2). 2003.

[17] Serfling, R. E. Methods for current statistical analysis for excess pneumonia-influenza deaths. *Public Health Reports*, 78:494-506. 1963.

[18] Shmueli, G. Wavelet-Based Monitoring in Biosurveillance, *Working Paper*, Smith School of Business, University of Maryland, College Park. 2005.

[19] Shmueli, G., and Fienberg, S. E., Current and Potential Statistical Methods for Monitoring Multiple Data Streams for Bio-Surveillance. In *Statistical Methods in Counter-Terrorism*, Eds: A Wilson and D Olwell, *to appear*.

[20] Zhang, J., Tsui, F., Wagner, M., and Hogan, W. Detection of outbreaks from time series data using wavelet transform. In *AMIA Annual Symposium Proceedings*, pages 748-752. 2003.

Outlier Detection in High-Dimensional Data - Using Exact Mapping to a Relative Distance Plane

R.L. Somorjai*, A. Demko, M. Mandelzweig

Institute for Biodiagnostics, National Research Council Canada
435 Ellice Avenue, Winnipeg MB R3B 1Y6

Outlier/novelty/anomaly detection (OD) is an important aspect of data analysis, data mining in particular. Classifying numerically highly unbalanced classes can also be formulated as OD; in extremis, it is called one-class classification.

Outlier detection in high-dimensional (HD) spaces presents specific challenges, mostly related to the curse of dimensionality. In particular, masking effects may be troublesome in HD spaces. Exact and reliable detection and low-dimensional visualization of outliers would be especially relevant.

We show below that for distance-driven OD, the distance (similarity) – based mapping we have developed for visualizing high dimensional patterns and their relative relationships (Somorjai et al., 2004) could be particularly useful. This mapping only requires a single computation of a distance matrix in some metric. The mapping's most important characteristics is that certain distances in the original, high-dimensional space are *exactly* preserved in a special 2D coordinate system (S, T), the *relative distance plane* (RDP). Thus, all points of the dataset can be displayed without any distortion of their original distances *to two reference patterns*, say, $\mathbf{R_1}$ and $\mathbf{R_2}$. The RDP mapping is a version of projection pursuit, using directions defined by pairs of patterns actually in the dataset.

Since the reference patterns can be any pair in the dataset, this provides not only an immediate visualization of a putative set of outliers, but a powerful approach to confirm that the set really does represent likely outliers. One simply cycles through all possible reference pairs and records the frequency of occurrence of a sample's "outlyingness". A sample's likely outlier status can be further confirmed by using different distance metrics.

We demonstrate the use of the RDP software for outlier detection on publicly available gene microarray data, and on mass spectra from proteomics experiments.

Population-wide Anomaly Detection

Weng-Keen Wong, Gregory F. Cooper
RODS Laboratory,
University of Pittsburgh
Forbes Tower, Suite 8084
200 Lothrop Street
Pittsburgh, PA 15213
412-647-7113
{wwong,gfc}@cbmi.pitt.edu

Denver H. Dash
Intel Research
3600 Juliette Lane
Santa Clara, CA 95054
408-765-0410
denver.h.dash@intel.com

John D. Levander, John N. Dowling, William R. Hogan, Michael M. Wagner
RODS Laboratory,
University of Pittsburgh
550 Cellomics Building
100 Technology Drive
412-383-8134
jdl@cbmi.pitt.edu, dowling+@pitt.edu, {wrh,mmw}@cbmi.pitt.edu

ABSTRACT
Early detection of disease outbreaks, particularly an outbreak due to an act of bioterrorism, is a critically important problem due to the potential to reduce both morbidity and mortality. One of the most lethal bioterrorism scenarios is a large-scale release of inhalational anthrax. The Population-wide Anomaly Detection and Assessment (PANDA) algorithm [1] is specifically designed to monitor health-care data for the onset of an outbreak caused by an outdoor, airborne release of inhalational anthrax. At the heart of the PANDA algorithm is a causal Bayesian network which models the effects of the outbreak on a population. The most unique aspect of the PANDA algorithm is an approach we will refer to as *population-wide anomaly detection* in which each individual in the population is represented as a subnetwork of the overall causal Bayesian network. This paper will describe the benefits of the population-wide approach used by PANDA, which include a coherent way to incorporate background knowledge as well as different types of evidence, the ability to combine multiple data sources indicative of an outbreak, and the capability to identify the evidence that contributes the most to the belief that an anthrax outbreak is occurring.

Keywords
Anomaly Detection, Syndromic Surveillance, Biosurveillance, Bayesian Networks

1. INTRODUCTION
Early detection of disease outbreaks is a critically important problem due to the potential to reduce both morbidity and mortality. Disease outbreaks can either occur naturally or they can be caused by acts of bioterrorism. One of the most lethal bioterrorism scenarios is a large-scale release of inhalational anthrax, which is estimated to kill as many as 30,000 people per day and to have a long-term economic cost of as much as $200 million per hour of the outbreak according to an analysis done by [2]. The Population-wide Anomaly Detection and Assessment (PANDA) algorithm [1] is specifically designed to monitor health-care data for the onset of an outbreak caused by an outdoor, airborne release of inhalational anthrax. At the heart of the PANDA algorithm is a causal Bayesian network[1] which models the effects of the outbreak on a population. The most unique aspect of the PANDA algorithm is an approach we will refer to as *population-wide anomaly detection* in which each individual in the population is represented as a subnetwork of the overall Bayesian network.

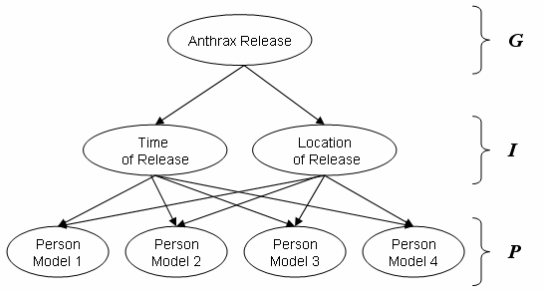

Figure 1: The causal Bayesian network structure used in PANDA.

Figure 1 provides an illustration of the causal Bayesian network structure used in PANDA to detect an outdoor release of inhalational anthrax. This model consists of three sets of nodes which we have labeled ***G***, ***I*** and ***P*** in the diagram. The nodes in the set ***G*** consist of global features that are common to all people. Included in the set ***G*** is a target node *T* which is the node that is actively monitored. In Figure 1, the target node is *Anthrax Release* and we monitor the probability that *Anthrax Release* equals true. In our example, *Anthrax Release* is the only global node. In general, the set ***G*** could include other global features such as the national terror alert level or information about local events such as major sports events or political conventions. At the next layer down, the interface nodes in set ***I*** are the nodes which contain the factors that significantly influence the status of an outbreak disease in individuals in the population. Inhalational anthrax is an infectious but non-contagious disease; the bio-agent can only infect people through the spores and not through person-to-person contact. As a result, the state of the disease in the population can be reasonably modeled with the nodes *Time of*

[1] A causal Bayesian network is a Bayesian network in which the arcs have a causal interpretation in addition to indicating a probabilistic relationship between the nodes.

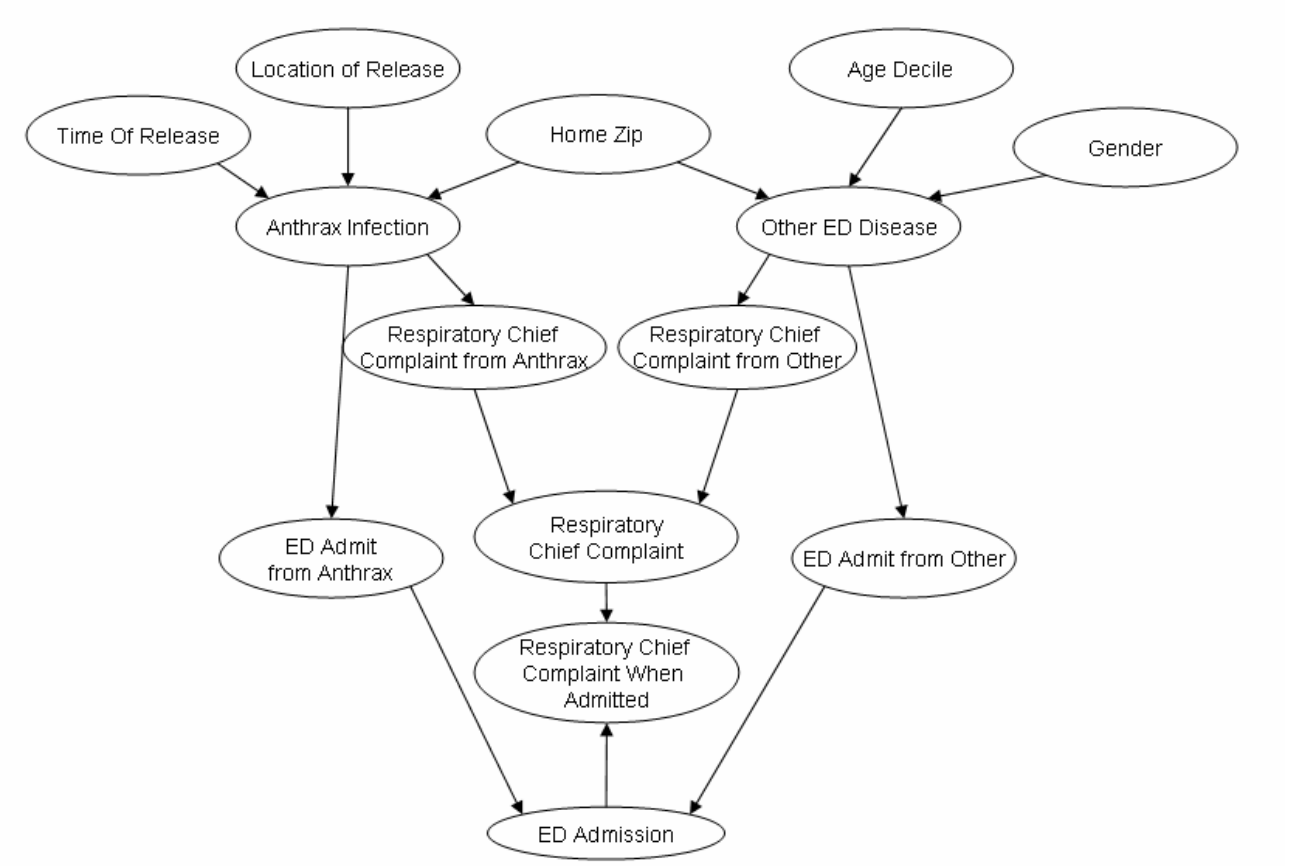

Figure 2: The person model used by PANDA to model each individual in the population.

Release and *Location of Release*. In the future, we plan to refine the model to include other interface nodes such as the amount of the release, the type of anthrax powder and meteorological information. Often the variables in $\boldsymbol{I}$ will be unmeasured. It is legitimate, however, to have measured variables in $\boldsymbol{I}$. For example, the wind direction might be a measured variable that influences the disease status of people in the population, and thus it would be located in $\boldsymbol{I}$. The last set of nodes $\boldsymbol{P}$ consists of the *person models* P_i *ie.* $\boldsymbol{P} = \{P_1, \ldots, P_n\}$ which form the core of the population-wide approach. Although we refer to these subnetworks as a *person* model, it can be generalized to entities that provide information about disease outbreaks, such as biosensors and livestock. Each person model P_i represents an individual in the population. In general, each person model can be different but for simplicity, we will use the model shown in Figure 2 for each P_i. Evidence observed on an individual basis will be entered at the person model level. In our implementation of PANDA, the information observed for each individual consists of Emergency Department (ED) records. Each ED record contains the attributes Home Zip, Age (by decile), Gender, Respiratory Chief Complaint When Admitted to the ED, and ED Admission.

The structure of the Bayesian network used by PANDA is designed by expert judgment rather than learned from data. The parameters of our model are obtained from a combination of census data, training data consisting of one year's worth of ED records, and expert assessments informed by the literature. With the structure and parameters of the model in place, we can perform inference on the Bayesian network to calculate the probability of an anthrax release. We emphasize that the population-wide anomaly detection approach is only used in designing the structure of the model and in the inference phase. We do not estimate parameters for an individual in the population. Rather, we observe evidence regarding an individual from an ED record and then propagate the effects of that evidence through the Bayesian network in order to update our belief that an anthrax attack is occurring.

At first glance, this approach appears to be intractable since the resulting model will consist of millions of nodes. For example, a typical surveillance region such as Allegheny County, Pennsylvania consists of 1.4 million people. Inference on a network of this scale seems intractable. However, in a previous paper [1], we have shown that such an approach is indeed feasible for a real-time bio-surveillance application that monitors Emergency Department (ED) data. In our initial prototype in [1], we exploit the conditional independence structure of the causal Bayesian network to produce two optimizations: incremental updating and equivalence classes. Incremental updating dramatically reduces inference time by allowing us to calculate probabilities for the entire population incrementally rather than from scratch whenever new data arrives. We also exploit the fact that individuals with exactly the same evidence are indistinguishable under the PANDA model. Individuals with the same values for the *Home Zip*, *Age Decile*, *Gender*, *Respiratory Chief Complaint When Admitted*, and *ED Admission* nodes are placed into the same equivalence class. In our surveillance of

Allegheny County, this optimization reduces the population of 1.4 million people to 24,240 equivalence classes. On a Pentium 4, 3 Gigahertz processor with 2 Gigabytes of RAM, the PANDA algorithm takes approximately 45 seconds of initialization time; after initialization, each hour of ED data can be processed in about 3 seconds.

Grouping individuals in equivalence classes may seem to contradict our claim of modeling each individual in the population. However, the use of equivalence classes is purely for computational convenience. We are indeed representing each person in the population and we are still capable of doing so without equivalence classes, albeit at a higher computational price. In future work, we intend to incorporate more information regarding the symptoms exhibited by patients in the ED. Adding this information will increase the number of features that define an equivalence class and consequently increase the number of equivalence classes beyond the number of people in the population. We plan to replace the use of equivalence classes with other optimizations such as approximate inference in order to make future extensions of the PANDA algorithm tractable.

Having addressed the most obvious downfall to population-wide anomaly detection, we will now discuss its advantages. Intuitively, it is the individuals in the population that generate the observed evidence. Thus, the most logical unit in the model is the individual, which is the finest level of granularity permitted by the data. With the modeling unit of an individual, we can exploit the conditional independence between individuals for a non-contagious disease to make inference tractable. As shown in Figure 1, if we condition on the time and location of the anthrax release, then the person models in the population are independent of each other. Another advantage gained by modeling each individual in the population is the ability to distinguish arbitrary groups from each other. This ability buys us a tremendous about of representational flexibility and power. In particular, we can coherently incorporate various forms of background knowledge and evidence into the model. Modeling at the individual level also facilitates combining information between multiple data sources, especially if the interaction between these data sources is much easier to model at an individual level than at a population level. Finally, the population-wide approach allows us to determine the contribution of each individual to the overall probability that an anthrax attack is occurring. We can determine the individuals that most influence this belief and in doing so, produce an explanation for why we believe an attack has occurred. The remainder of this paper will address these merits of a population-wide anomaly detection approach. We intend to provide an overview of this approach while leaving the details in previous papers on PANDA [1, 3].

2. INCORPORATING BACKGROUND KNOWLEDGE

One of the main advantages of a population-wide approach is the ability to coherently represent different types of background knowledge in the model. This background knowledge is particularly useful for disease outbreak detection algorithms that monitor for a specific disease; we will refer to these detection algorithms as *specific detectors*. In contrast, a *non-specific detector* such as WSARE [4] searches for any irregularities from the normal behavior. A strategy that works well for *non-specific detectors* is to model the baseline behavior of the data and signal an alert when the deviation from this baseline exceeds some threshold. However, since this strategy raises alarms for any irregularities rather than those caused by the disease being monitored, it can result in many false positives for a specific detector. We can improve the performance of specific disease detectors by building models of the data during non-outbreak periods and building models of the effects of the specific disease during outbreak periods.

Data during non-outbreak conditions are often available and in some cases abundant. The most common approach to building a baseline model is to use standard machine learning techniques such as Bayesian network structure learning [4] to learn the structure and/or the parameters of the model. Another option is to incorporate background knowledge of this baseline behavior into our model; for instance, in PANDA we use census information to model the demographics of the population. In contrast to data during non-outbreak periods, data during outbreak periods are scarce or completely non-existent. In the case of anthrax, there are only two commonly known anthrax outbreaks – an accidental leak in Sverdlovsk, Russia [5] and the 2001 postal anthrax attacks [6-9], although the postal attacks are clearly not representative of the large-scale outdoor release of anthrax that the PANDA algorithm is intended to detect. We cannot learn a model of an anthrax outbreak from data because do not have training data available from both of these incidents.

Nevertheless, we can incorporate the assessments of domain experts who are informed by their experience and the literature. In addition to studies performed on the two known outbreaks, there is a vast body of medical literature that allows us to model what we know about the likely patterns of presentation of inhalational anthrax [10-13]. In particular, we can model the known progression of symptoms that occur after a person has inhaled anthrax spores. We can also represent the incubation period, which is the earliest period of time after infection that a person begins to physically manifest the symptoms of anthrax (the incubation period varies depending on the concentration of spores released and the amount inhaled by an individual). Finally, in the case of an airborne release of anthrax, we can model the spatial dispersion pattern of the spores as in [14, 15], enabling the detection algorithm to know that a person standing downwind in the dispersion region is more likely to be infected than someone who is standing upwind. We can coherently incorporate all of this information in the parameters of the causal Bayesian network as background knowledge. Most importantly, the majority of the background knowledge about inhalational anthrax is at an individual level and it is precisely this background knowledge that we intend to use to improve our detector.

3. INCORPORATING DIFFERENT TYPES OF EVIDENCE

Besides the power in representing different forms of prior knowledge, modeling each individual allows the model to combine spatial, temporal, demographic, and symptomatic evidence to derive a posterior probability of a disease outbreak. For instance, if many people are admitted to the ED with symptoms consistent with those of inhalational anthrax and their home locations follow roughly the spatial dispersal pattern of an

airborne anthrax release, then the posterior probability of an anthrax attack should be high. Furthermore, individual modeling permits new types of knowledge and evidence to be readily incorporated into the model. We had previously assumed that the person models are identical for the purpose of simplicity but we can easily incorporate different person models into our framework. If we know more information about one person or group of people than another, we can represent that difference. As an example, if we gain access to radiology reports for a group of individuals, and we find that radiology reports are especially useful indicators of an anthrax attack, we can then readily add this new evidential variable to the person model representing those individuals.

4. DATA FUSION

Modeling each individual in the population also facilitates fusion of different data sources, because such data originate from the individuals in the population that are being explicitly modeled. In [3] we extended the PANDA model to incorporate evidence from both ED data and from over-the-counter (OTC) data. By jointly monitoring both data sources, the combined information could reinforce our belief that an anthrax outbreak is happening and improve the detection algorithm's performance. However, the correlation between OTC and ED data during outbreak conditions cannot be learned because no training data exists that captures the effects of a large-scale anthrax attack on these data sources during the same time period. Although training data do not exist, we do have some background knowledge at the individual level about the plausible relationship between OTC and ED data during an anthrax outbreak. Our approach to combining multiple data streams relies on using this background knowledge and explicitly modeling the actions of individuals that result in the interaction between OTC medication purchases and ED admissions.

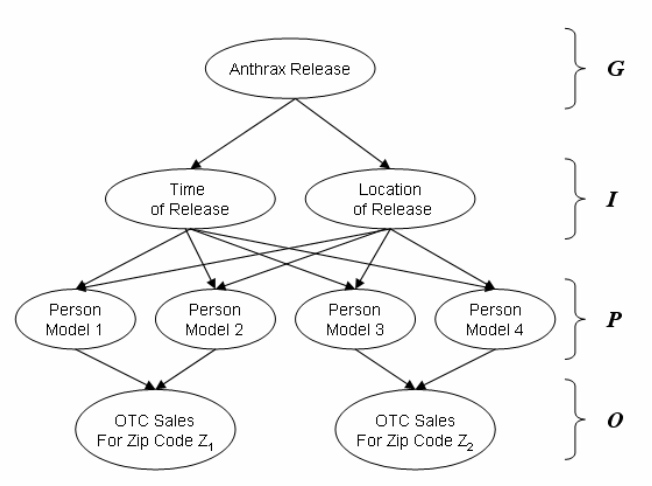

Figure 3: The causal Bayesian network used to combine ED data and OTC data.

Another concern in data fusion is incorporating data sources of different spatial and temporal granularity. For example, ED data is available in real-time (although we process it as if it were available hourly) with each record corresponding to an individual. On the other hand, the OTC data is available at the end of each day and each record aggregates the OTC sales over a zip code. The population-wide approach models the data at the level of an individual, which is the finest granularity that makes sense and that is permissible though the data. With this level of granularity, we can always aggregate individuals to form a coarser level of granularity while taking full advantage of all the information available.

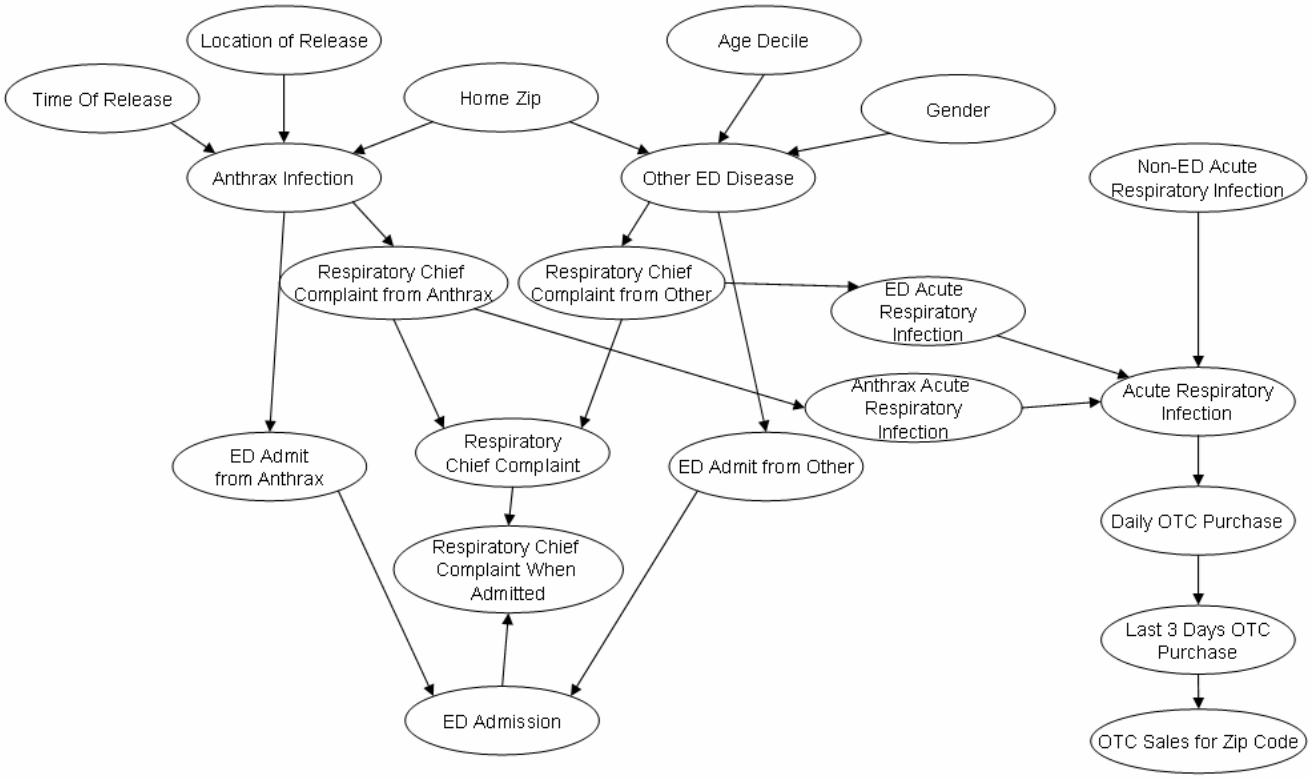

Figure 4: The person model for the PANDA algorithm that combines both ED data with OTC data.

Figure 3 illustrates the extension to the model in Figure 1 while Figure 4 depicts the modifications to the person model in Figure 2. The new causal Bayesian network incorporates the OTC evidence in the set of population-wide evidence nodes ***O***. The set ***O*** represents evidence that is aggregated over a particular group of people, such as the daily OTC sales of cough medication sales over a zip code.

5. EXPLANATION

With a population-wide anomaly detection algorithm, we can not only detect anomalies but also explain why they are anomalies. Using the Bayesian network framework, we can find the subset of evidence E^* that most influences the target node T. Once this subset of evidence is found, we can trace the pathways between E^* and T that contribute the most to the belief that an attack is occurring. In the current PANDA model, E^* corresponds to evidence about individuals. We can determine the individuals whose evidence most supports the hypothesis of an anthrax attack. Once these individuals have been identified, we can determine the relationships between them, which can potentially identify the origin and subsequent spread of the anthrax release. In our current prototype, we group the individuals into equivalence classes defined by the evidence observed in the data. Thus, we can identify the equivalence class that most supports the hypothesis of an anthrax attack. We have also used this explanation method to identify the zip code that is the most likely location of the release and the day that is the most likely time of release.

6. CONCLUSION

We have approached the task of detecting a large-scale airborne release of inhalational anthrax with a population-wide anomaly detection algorithm. This method has been ideally suited for this task due to the various forms of background knowledge and evidence that need to be incorporated into the model. In addition, if an alert is raised over a possible anthrax release, we gain the capability to explain why the alarm was triggered. The results reported in [1] have been promising and indicate that modeling each individual is feasible for a real-time bio-surveillance system. We also believe that the merits of this approach can benefit anomaly detection tasks in other domains.

7. ACKNOWLEDGEMENTS

This research was supported by grants IIS-0325581 from the National Science Foundation, F30602-01-2-0550 from the Department of Homeland Security, and ME-01-737 from the Pennsylvania Department of Health.

8. REFERENCES

[1] Cooper, G.F., et al. *Bayesian Biosurveillance of Disease Outbreaks.* in *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence.* 2004. Banff, Canada: AUAI Press.

[2] Kaufmann, A., M. Meltzer, and G. Schmid, *The economic impact of a bioterrorist attack: Are prevention and postattack intervention programs justifiable?* Emerging Infectious Diseases, 1997. 3(2): p. 83-94.

[3] Wong, W.-K., et al. *Bayesian Biosurveillance Using Multiple Data Streams.* in *Proceedings of the Third National Syndromic Surveillance Conference*. 2004. Boston, MA.

[4] Wong, W.-K., et al. *Bayesian Network Anomaly Pattern Detection for Disease Outbreaks*. in *Proceedings of the Twentieth International Conference on Machine Learning.* 2003: AAAI Press.

[5] Meselson, M., et al., *The Sverdlovsk anthrax outbreak of 1979.* Science, 1994. 266(5188): p. 1202-1208.

[6] Barakat, L.A., et al., *Fatal inhalational anthrax in a 94-year-old Connecticut woman.* Journal of the American Medical Association, 2002: p. 287-868.

[7] Inglesby, T.V., et al., *Anthrax as a biological weapon, 2002: updated recommendations for management.* Journal of the American Medical Association, 2002. 287(17): p. 2236-2252.

[8] Jernigan, J.A., et al., *Bioterrorism-related inhalation anthrax: the first 10 cases reported in the United States.* Emerging Infectious Diseases, 2001. 7(6): p. 933-944.

[9] Mayer, T.A., et al., *Inhalational anthrax due to bioterrorism: would current Centers for Disease Control guidelines have identified the 11 patients with inhalational anthrax from October through November 2001?* Clinical Infectious Diseases, 2003. 36(10): p. 1275-1283.

[10] Brachman, P.S., *Inhalation anthrax.* Annals of the New York Academy of Science., 1980. 353: p. 83-93.

[11] Franz, D.R., et al., *Clinical recognition and management of patients exposed to biological warfare agents.* Journal of the American Medical Association, 1997. 278: p. 399-411.

[12] Penn, C.C. and S.A. Klotz, *Anthrax pneumonia.* Seminars in Respiratory Infections, 1997. 12(1): p. 28-30.

[13] Shafazand, S., et al., *Inhalational anthrax: epidemiology, diagnosis, and management.* Chest, 1999. 116: p. 1369-1376.

[14] Hogan, W.R., G.F. Cooper, and M.M. Wagner, *A Bayesian anthrax aerosol release detectors.* RODS Technical Report, 2004.

[15] Wein, L.M., D.L. Craft, and E.H. Kaplan, *Emergency response to an anthrax attack.* Proceedings of the National Academy of Sciences USA, 2003. 100: p. 4346-4351.

Strip Mining the Sky: The CTI-II Transit Telescope Survey

Peter Zimmer
University of New Mexico
Dept. of Physics and Astronomy
800 Yale NE
Albuquerque, NM 87131
(505) 277-5127
zimm@as.unm.edu

John T. McGraw
University of New Mexico
Dept. of Physics and Astronomy
800 Yale NE
Albuquerque, NM 87131
(505) 277-5127
jmcgraw@as.unm.edu

The CTI-II Computing Collective
University of New Mexico
800 Yale NE
Albuquerque, NM 87131
(505) 277-5127

ABSTRACT

In observational astronomy, data anomalies are either exciting new discoveries of heretofore unknown classes of phenomena, or noise. We describe an upcoming large astronomical dataset consisting of approximately 200 gigapixels per night of operation over a period of seven years. In addition, a database of parameters derived from the images, as well as extensive metadata, will be collected and stored. We believe that this dataset will provide fertile ground for novel anomaly detection methods to provide not just error detection and correction, but more importantly to discover new and interesting astronomical objects.

1. ASTRONOMICAL ANOMALIES

Generically, astronomical anomalies manifest themselves in one of several broad categories: angular, spectral, temporal and bungle. Because astronomers deal with objects at large distances, spatial phenomena become angular phenomena. The spectral domain is often represented as broadband colors of objects, and for the CTI-II, the temporal sampling occurs at one day intervals. The category of *bungle* includes extreme noise fluctuations, data artifacts such as cosmic ray events or bright stars flooding our optical detectors, or processing errors.

Most astronomical objects have well understood shapes. Statistically, they are either stars - point sources at a large distance - or galaxies, which are made up of hundreds of billions of stars at vastly larger distances. Telescope optics limit the spatial frequencies that can be detected, and as long as the highest spatial frequencies are well sampled in the images, objects containing frequencies higher than this cut-off are of the bungle variety. Objects that have other anomalous shapes, when compared to the ensemble of well understood shapes of astronomical objects, are certainly of interest. Similarly, anomalous groupings of objects are also of considerable interests.

The astrophysics of the vast majority of astronomical objects is also relatively well understood and encompasses a narrow range of physical mechanisms such that the range of spectral signatures (colors) of astronomical objects is fairly well constrained. Objects with anomalous colors are of particular interest in that they indicate uncommon physics.

The time-domain variations of astronomical objects, especially objects with rapid and dramatic changes which can manifest themselves as shape, intensity or color changes, are high priority targets for further study. Objects that move between observations or appear and disappear over the course of several observations are referred to as transients. The field of transient astronomy has been enabled by large scale surveys with automated data processing.

Dangerously, from the discovery perspective, the most common sort of astronomical anomaly is the bungle: one or more errors confused as a legitimate measurement. Such anomalies can occur due to instrumental artifacts, stray light entering the telescope, cosmic rays, image edges, and many other sources.

2. CTI-II

The University of New Mexico is currently implementing the CCD/Transit Instrument Version II (CTI-II), a 1.8m meridian-pointing telescope, and equipping it with a modern focal plane array and wide-field optics for deployment at McDonald Observatory. The current design of CTI-II is expected to generate over two hundred gigapixels of image data per night of operation from a one degree wide strip of the sky observed in five bandpasses. These data will feed both near real-time (detection and classification within 15 minutes of observation) and time-critical (detection and classification within 12 hours of observation) analysis pipelines, the design of which is driven by the principal science projects of CTI-II. However, the goals of these analysis systems are common to many sky surveys: precision astrometry, precision photometry, and the ability to facilitate rapid follow-up observations. It is this last goal that falls squarely in the domain of anomaly detection.

Given the potential of new optics and detectors coupled with a unique, dedicated observing mode, several key science projects have been chosen as drivers of the ultimate design of CTI-II: Red Star Astrometry, AGN Reverberations, and Supernova

Detection. Each of these science drivers places strong constraints on the data reduction and analysis pipeline:

2.1 Red Star Astrometry

M dwarfs constitute 70% by number and 40% by mass of the stars in the solar neighborhood. Despite previous large-scale sky surveys, much of the nearby M dwarf population remains undetected. Long-term astrometric monitoring of these objects will enable milliarcsecond parallax and proper motion measurements. The resulting three dimensional motions support a probe of the gravitational properties of our Galaxy and of the scale-heights of low-mass stellar populations.

2.2 Active Galactic Nuclei Reverberations

When an Active Galactic Nucleus (AGN) outburst occurs, there is a lag between when the UV/optical continuum brightens and when the broad emission lines react. This light travel time lag allows the inner structure of AGNs in the vicinity of the central supermassive black holes to be dissected with an absolute scale size. CTI-II observations will provide a basis for photometric variability investigations for more than 1000 nearby to distant AGNs ($15 < B < 20$). Using intermediate band continuum filters, CTI-II will monitor the variability of AGNs, providing a "trigger" for AGN outbursts to be followed up with spectroscopic observations by other telescopes worldwide.

2.3 Supernova Detection

CTI-II is well-suited to discovering supernovae – the explosive deaths of stars – in distant galaxies. In each nightly survey strip, CTI-II will observe more than 300,000 galaxies and discover several new supernovae per night to $m_B < 22.5$. Detection of these supernovae will occur in near real-time and provide targets-of-opportunity for immediate follow-up. Therefore, the near real-time portion of the data analysis pipeline must identify supernova candidates, characterize them, and alert the astronomer on-call in less than fifteen minutes, with a very low false positive rate, under varying seeing and background conditions.

3. DATA MINE

The cumulative CTI-II pixel archive and object database will provide a fertile ground for new and existing anomaly detection routines. Over the course of its planned seven years of operation, it will generate over 100 TB of image data from the same one degree wide strip across the sky, observing each part of the strip over 700 times in each of five colors. The database of object observations generated from this image data will contain millions galaxies, over two million stars, and over a hundred thousand quasars. Because of varying observing conditions, noise in the observations, and changing background levels, few pixels (and therefore no object) will be the same from night to night.

Anomaly detection and characterization is key to the success of this and other astronomical surveys. Discovering the "right" anomaly won't be just a curiosity, it will be a discovery of universal proportions.