Proceedings

2014 Workshop on Optimization Methods for Anomaly Detection

OMAD 2014

Held as part of the 2014 SIAM International Conference on Data Mining in Philadelphia, PA

24-26 April 2014

Edited By

Sanjay Chawla, Kamalika Das, Aris Gionis

Foreward

Anomaly Detection techniques are playing an increasingly important role in the analysis of large data sets across many application domains. In particular, the use of anomaly detection techniques for monitoring digital and physical infrastructure is growing rapidly, Other important application domains include health and climate informatics. The aim of the 2014 workshop on Optimization Methods for Anomaly Detection (OMAD) is to bring together researchers to study the detection of anomalies in large data sets in a systematic optimization framework. The workshop consists of four papers and two keynote addresses.

In *ParitoSVR: Parallel Iterated Optimizer for Support Vector Regression in the Primal,* the authors will present a distributed algorithm for support vector regression using the ADMM framework. The SVM model is then used to identify anomalies as those data points, which have a large residual value vis-à-vis the model. The authors apply paritoSVR on a real (and large) data set consisting of fuel consumption patterns in airline flights.

In *Anomaly Detection Using Tripoint Arbitration Similarity Method,* the author, proposes a tripoint similarity function to identify outliers. The similarity function is used in a MinMaxCut optimization framework. The proposed method is validated on an application related to monitoring computing infrastructure.

A new measure of anomalousness (called q-value) is proposed in *Measuring Anomalousness in Statistical Models*. The measure, which is related to p-value, provides a natural way to find anomalies in clustered data.

In *Identifying Precursors to Anomalies Using Inverse Reinforcement Learning*, the authors propose a method for determining pre-cursor signals just before the advent of an anomalous event. The application domain includes the monitoring of airplane flight data.

The workshop will also host two keynote talks. The first by Professor Vipin Kumar from the University of Minnesota will highlight the role of anomaly detection in getting a better understanding of climate data. The second, by Dr. Dragos Margineantu, from Boeing, will focus on the application of anomaly detection techniques in the airline industry.

The organizing committee would like to thank (i) the authors for participating in the workshop, (ii) the organizers of the SDM 2014, especially Professor Tina Elliasi-Rad, the SDM workshops chair, (iii) colleagues who served in the program committee

Finally we would like to thank the sponsors of the workshop: NASA Ames and National ICT Australia (NICTA) for their generous support.

Thanks!

Sanjay Chawla, Kamalika Das and Aris Gionis

Workshop Committee

Program Chairs:

Sanjay Chawla, University of Syndey
Kamalika Das, UARC, NASA Ames Research Center
Aris Gionis, Aalto University

Program Committee:

Leman Akoglu SUNY, Stonybrook
Arindam Banerjee, University of Minnesota
Kanishka Bhaduri, Netflix Inc.
Varun Chandola, SUNY, Buffalo
Tina Eliassi-Rad , Rutgers University
Jing Gao, SUNY, Buffalo
Manish Gupta, Microsoft India
Aditya Menon, NICTA
Emmanuel Müller , Karlsruhe Institute of Technology
Khoa Nguyen, HCMUT
Nikunj Oza, NASA Ames Research Center
Aditya Prakash, Virginia Tech
Eric Schubert, Ludwig-Maximilians-Universität München
Nikolaj Tatti, Aalto University
Matthijs van Leeuwen, KU Leuven
Hamed Valizadegan , UARC, NASA Ames
Jilles Vreeken, University of Antwerp
Arthur Zimek, Ludwig-Maximilians-Universität München

2014 Workshop on Optimization Methods for Anomaly Detection

OMAD 2012
Table of Contents

Invited Talk Abstracts

Accepted Abstracts

Understanding Global Change: Opportunities and Challenges for Data Driven Research

Vipin Kumar

The world's population is growing steadily and many countries are simultaneously industrializing, developments that have been ongoing at varying rates for two centuries but have accelerated over the past several decades. These processes are increasingly straining already scarce natural and food resources, which must scale up to keep pace with growing demand. The consequences of such large-scale changes include tremendous stresses on the environment that would be calamitous at the current rate of change if they are not managed sustainably. As a result, scientists are tasked with providing answers to challenging questions such as: What is the effect of urbanization on regional land use and ecology? What is the impact of climate change on global water resources? How does deforestation affect the net carbon balance? How does increased biofuel production impact crop patterns and food availability? Addressing these interconnected, societally-relevant questions requires development of new computational methods that enable monitoring, analysis and understanding of changes in the Earth system, interactions between different processes, and their impacts on factors such as the carbon cycle, hydrology, air quality, and biodiversity.

This talk will present an overview of research being done in a large interdisciplinary project on the development of novel data driven approaches that take advantage of the wealth of climate and ecosystem data now available from satellite and ground-based sensors, the observational record for atmospheric, oceanic, and terrestrial processes, and physics-based climate model simulations. These information-rich datasets offer huge potential for monitoring, understanding, and predicting the behavior of the Earth's ecosystem and for advancing the science of global change. This talk will discuss some of the challenges in analyzing such data sets and our early research results.

User-in-the-loop Learning and Optimization for Anomalous Action Detection

Dragos Margineantu

An increasing number of users collect transaction data and need scalable tools that assist them in identifying abnormalities. This talk will present an interactive user-in-the-loop approach based on inverse reinforcement learning and linear optimization methods for detecting anomalies and intent in data. We implemented and tested our algorithms on real-world GMTI and AIS sensor data.

ParitoSVR: Parallel Iterated Optimizer for Support Vector Regression in the Primal

Kamalika Das* Kanishka Bhaduri† Nikunj Oza‡

Abstract

Regression problems on massive data sets are ubiquitous in many application domains including the Internet, earth and space sciences, and aviation. Support vector regression (SVR) is a popular technique for modeling the input-output relations of a set of variables under the added constraint of maximizing the margin, thereby leading to a very generalizable and regularized model. However, for a dataset with m training points, it is challenging to build SVR models due to the $O(m^3)$ cost involved in building them. In this paper we propose ParitoSVR — a parallel iterated optimizer for Support Vector Regression in the primal that can be deployed over a network of machines, where each machine iteratively solves a small (sub-)problem based only on the data observed locally and these solutions are then combined to form the solution to the global problem. Experiments on real datasets demonstrate the accuracy and scalability of our algorithm. As a real application, we use ParitoSVR to detect flights having abnormal fuel consumption from a fleet-wide commercial aviation database.

1 Introduction

In many application domains, it is important to predict the value of one feature based on certain other measured features. For example, in commercial aviation, it is very important to model the fuel consumption based on input parameters such as aircraft speed, wind speed, control surfaces, engine power, pitch, roll, yaw etc. This is because according to the Air Transportation Association (ATA), fuel is an airline's largest expense at a staggering 17.5 billion gallons per year[1]. Identifying flights with abnormal fuel consumption may help the airlines to do proper maintenance of these aircrafts and save operating costs. For such problems, a regression model can be learned that predicts the fuel flow based on these input parameters. One such popular regression method is Support vector machines (SVM) [1] which is a class of maximum margin classifiers, that demonstrates good generalization performance. SVM's can also exploit the kernel trick, thereby making them suitable for non-linear model learning as well. SVMs however are computationally expensive for large datasets.

In this paper we propose Parallel Iterated Optimizer for Support Vector Regression in the Primal (ParitoSVR), a new support vector regression algorithm that can be deployed over a network of machines, where each machine solves a small (sub-)problem based only on the data observed locally and these solutions are then combined to form the solution to the global problem. Our proposed method is based on the Alternating Direction Method of Multipliers (ADMM) optimization technique [2][3], which is parallelizable for separable convex problems, and converges to the exact solution as the centralized version with theoretical guarantees.

2 Background

Our ParitoSVR algorithm uses as a building block two components: (1) Alternating Direction Method of Multipliers (ADMM), and (2) SVR. In this section, we discuss these two topics.

ADMM: ADMM [3] is a decomposition algorithm for solving separable convex optimization problems of the form:

$$\begin{aligned} (2.1) \qquad & \min_{\mathbf{x},\mathbf{y}} \quad G_1(\mathbf{x}) + G_2(\mathbf{y}) \\ & \text{subject to} \quad A\mathbf{x} - \mathbf{y} = 0, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{y} \in \mathbb{R}^m \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$ and G_1 and G_2 are convex functions. ADMM is an iterative technique and the update equations are:

$$\begin{aligned} \mathbf{x}^{t+1} &= \min_{\mathbf{x}} \left\{ G_1(\mathbf{x}) + \rho/2 \left\| A\mathbf{x} - \mathbf{y}^t + \mathbf{p}^t \right\|_2^2 \right\} \\ \mathbf{y}^{t+1} &= \min_{\mathbf{y}} \left\{ G_2(\mathbf{y}) + \rho/2 \left\| A\mathbf{x}^{t+1} - \mathbf{y} + \mathbf{p}^t \right\|_2^2 \right\} \\ \mathbf{p}^{t+1} &= \mathbf{p}^t + A\mathbf{x}^{t+1} - \mathbf{y}^{t+1} \end{aligned}$$

where $\mathbf{p} = (1/\rho)\mathbf{z}$. ADMM effectively decouples the $\mathbf{x}$ and $\mathbf{y}$ updates such that parallel execution becomes possible. In a distributed computing framework, this becomes even more interesting since each computing node can now solve a (smaller) subproblem in $\mathbf{x}$ inde-

*UARC, NASA Ames Research Center. kamalika.das@nasa.gov

†Netflix Inc. kanishka.bh@gmail.com

‡NASA Ames Research Center. nikunj.c.oza@nasa.gov

[1] http://www.airlines.org/Energy/Fuels101/Pages/AirlineEnergyQA.aspx

pendently, and then, these solutions can be efficiently gathered to compute the consensus variable $\mathbf{y}$ and the dual variable $\mathbf{p}$. ADMM converges within a few iterations when moderate precision is required. This can be particularly useful for many large scale problems, similar to what we consider here.

SVR: Give m data tuples (training set) $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^m$, where $\mathbf{x}_i \in \mathbb{R}^n$ is the input and $y_i \in \mathbb{R}$ is the corresponding output or target, SVR solves the following optimization problem:

$$\min_{\mathbf{w},b} \left[\lambda ||\mathbf{w}||^2 + \sum_{i=1}^{m} \ell_\varepsilon(\mathbf{w} \cdot \mathbf{x}_i + b - y_i) \right] \tag{2.2}$$

where λ is a constant and ℓ_ε is the *ε-insensitive loss function* defined as, $\ell_\varepsilon(r) = \max(|r| - \varepsilon, 0)$. This is a convex optimization problem which can be solved using convex optimization solvers such as CVX[2].

In the next section we show how to build SVR models for very large datasets using distributed computing via the ADMM technique.

3 ParitoSVR formulation

For the linear ParitoSVR algorithm setup, we assume that the training data is distributed among N client processors (nodes) $P_1, \ldots, P_N$ with a central machine P_0 acting as the server or collector. The dataset at machine P_j, denoted by D_j, consists of m_j data points *i.e.* $D_j = \left\{ \mathbf{x}_i^{(j)}, y_i^{(j)} \right\}_{i=1}^{m_j}$. It is assumed that the datasets are disjoint: $D_i \bigcap D_j = \emptyset$ and $\bigcup_{j=1}^{N} D_j = D$, where D is the total (global) data set. The goal is to learn a linear support vector regression model on D without exchanging all of the data among all the nodes.

Given Eqn. 2.2, the optimization problem is now:

$$\min_{\mathbf{w}} \left[\sum_{i=1}^{m} \ell_\varepsilon(\mathbf{w} \cdot \mathbf{x}_i - y_i) + \lambda ||\mathbf{w}||^2 \right]$$

$$\Leftrightarrow \min_{\mathbf{w}} \left[\sum_{j=1}^{N} \sum_{i=1}^{m_j} \ell_\varepsilon \left(\mathbf{w} \cdot \mathbf{x}_i^{(j)} - y_i^{(j)} \right) + \lambda \|\mathbf{w}\|^2 \right]$$

The inner sum can be computed by each node independently (assuming that $\mathbf{w}$ is known). We next write it in a form such that it is decoupled across the nodes:

$$\min_{\mathbf{w}_1, \ldots, \mathbf{w}_N, \mathbf{z}} \left[\sum_{j=1}^{N} \sum_{i=1}^{m_j} \ell_\varepsilon \left(\mathbf{w}_j \cdot \mathbf{x}_i^{(j)} - y_i^{(j)} \right) + \lambda \|\mathbf{z}\|^2 \right] \tag{3.3}$$

$$\text{subject to} \quad \mathbf{w}_j = \mathbf{z}$$

In the ADMM decomposition, each node can solve its local problem using its own data and optimization variable and then coordinate the results across the nodes to drive them into consensus. The nodes update the consensus variable $\mathbf{z}$ iteratively, based on their local data and scatter-gather operations on $\mathbf{z}$ until they converge to the same result.

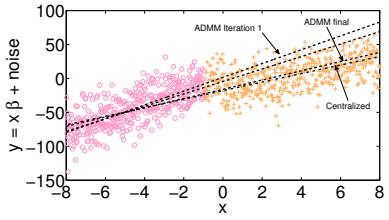


Figure 1: Models formed by node 1 on synthetic dataset as the algorithm progresses.

THEOREM 3.1. *The ADMM update rules for the linear support vector regression primal optimization are:*

$$\mathbf{w}_j^{t+1} = \min_{\mathbf{w}_j} \left\{ \sum_{i=1}^{m_j} \ell_\varepsilon \left(\mathbf{w}_j \cdot \mathbf{x}_i^{(j)} - y_i^{(j)} \right) + \frac{\rho}{2} \left\| \mathbf{w}_j - \mathbf{z}^t - \mathbf{u}_j^t \right\|_2^2 \right\}$$

$$\mathbf{z}^{t+1} = \min_{\mathbf{z}} \left\{ \lambda \|\mathbf{z}\|_2^2 + \frac{N\rho}{2} \left\| \mathbf{z} - \overline{\mathbf{w}}^{t+1} - \overline{\mathbf{u}}^t \right\|_2^2 \right\}$$

$$\mathbf{u}_j^{t+1} = \mathbf{u}_j^t + \mathbf{w}_j^{t+1} - \mathbf{z}^{t+1}$$

where $\mathbf{u} \in \mathbb{R}^n$ is the (scaled) dual variable and $\overline{\mathbf{w}}^{t+1}$ and $\overline{\mathbf{u}}^{t+1}$ are the averages of the variables over all the nodes.

Proof. We omit the proof here due to shortage of space.

The $\mathbf{w}$ update can be executed in parallel for each machine. It involves solving a convex optimization problem in $n+1$ variables at each node. This solution depends only on the data available at that partition. The $\mathbf{z}$ update step involves computing the average of the $\mathbf{w}$ and $\mathbf{u}$ vectors in order to combine the results from the different partitions. Critical to the working of ADMM is the convergence criteria. The primal and dual residuals can be written as: $r_p^t = \left\| \mathbf{w}^t - \mathbf{z}^t \right\|_2^2, r_d^t = \left\| \rho(\mathbf{z}^t - \mathbf{z}^{t-1}) \right\|$ Also, given the thresholds ϵ_{pri} and ϵ_{dual}, the primal and dual thresholds can be written as, $\epsilon_{pri} = \epsilon_{abs}\sqrt{m} + \epsilon_{rel} \max(\|\mathbf{w}\|, \|\mathbf{z}\|)$ and and $\epsilon_{dual} = \epsilon_{abs}\sqrt{m} + \rho\epsilon_{rel} \|\mathbf{u}\|$. The iterations terminate when $r_p^t < \epsilon_{pri}$ and $r_d^t < \epsilon_{dual}$.

4 Experiments

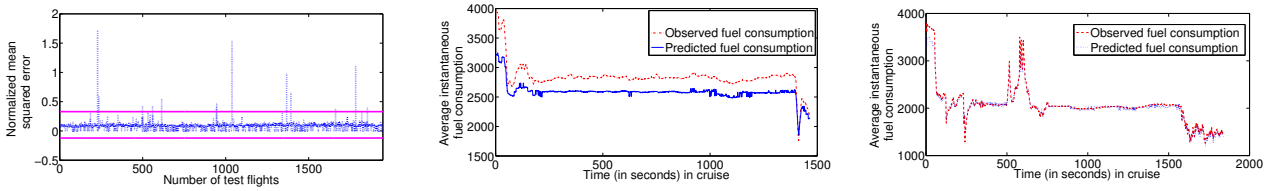
In this section we demonstrate the performance of the ParitoSVR algorithm.

ParitoSVR has been implemented in MATLAB 2011b. The experiments have been executed in NASA Pleiades supercomputer facility[3]. For solving the convex problems at each iteration, we have used the convex optimization toolbox CVX for Matlab[4].

[2] http://cvxr.com/cvx/

[3] http://www.nas.nasa.gov/hecc/resources/pleiades.html

[4] http://cvxr.com/cvx/



(a) Squared error for all flights (b) Outlier flight fuel consumption (c) Normal flight fuel consumption

Figure 2: Fuel flow study on CarrierX dataset. Fig. (a) shows squared error for all test flights, the 3-σ bound and flights which cross the threshold. Fig. (b) shows the observed and predicted fuel flow of top ranked anomalous flight. Fig. (c) shows the same for a normal flight.

Fig. 1 shows the sample dataset generated from a linear model following $y = \mathbf{w} \times \mathbf{x} + noise$, where $\mathbf{w}$ is the weight of the regression model. We have used 2 nodes in this experiment and, for each node, chosen a different $\mathbf{w}$ vector so that each node sees a different data distribution. The data of the two nodes are shown in two different colors (circle and plus markers). Also shown in the figure are the models (straight lines) formed by node 1 at different iterations of linear ParitoSVR algorithm.

4.1 Anomaly detection on CarrierX dataset We use the linear ParitoSVR algorithm to detect anomalous fuel consumption in a commercial aircraft. We model the average fuel flow as a function of 29 different parameters that measure system parameters such as lateral and longitudinal acceleration, roll and pitch angle, air pressure, and velocity, as well as external parameters such as wind speed and direction. We have used all 1500 flights (≈ 4.5 million training instances) for a specific tail number for a particular year for training, and tested subsequent years' flights for predicting fuel consumption. Flights for which the mean squared errors of the predicted instantaneous fuel consumption fall outside the 3-σ boundary of the average mean squared error, are tagged anomalous (σ is the standard deviation of the mean predictions). Out of approximately 1800 flights for a test year, 14 flights were determined to be anomalous. Figure 2(a) shows the mean squared errors for each of the flights in blue and the 3-σ bounds in green. The instantaneous fuel flow for the top ranked anomalous flight among these 14 flights is shown in Figure 2(b). The red graph depicting observed fuel flow is significantly higher than the predicted fuel consumption, shown in blue.

5 Conclusion

In this paper we have proposed ParitoSVR — a parallel iterated optimizer which solves support vector regression in the primal. Our formulation is parallelizable among a number of computing nodes connected to a central computing node. Empirical study show that our algorithm is accurate and scalable, ideal for large scale deployment. As future work, we plan to develop asynchronous version of this problem for peer-to-peer architectures.

Acknowledgements

The material is based upon work supported by the ARMD seedling fund from the National Aeronautics and Space Administration under Prime Contract Number NAS2-03144 awarded to the University of California, Santa Cruz, University Affiliated Research Center.

References

[1] V. V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995.

[2] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Found. and Trends in Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

Anomaly Detection Using Tripoint Arbitration Similarity Method

Aleksey Urmanov*

Abstract

The tripoint arbitration similarity method uses every point in a sample as an observer to evaluate the similarity of a pair of points of the sample. The similarity of the pair is aggregated over all observers in the sample. The resulting pairwise similarity matrix captures information about clusters of similar points. An anomalous point is defined as an observer point for which all points in the sample are pairwise similar. The method is independent of the underlying joint distribution of the sample points and does not require the user to tune any parameters other than selecting the appropriate distance function and setting the admissible false detection rate. The proposed method handles heterogeneous data by computing a combined similarity score which is of interest for many industrial, social, scientific, web, retail, finance and health sciences applications. The work in progress on anomaly detection using the tripoint arbitration similarity method is reported.

1 Introduction

Anomaly/outlier detection is one of the practical problems of data analysis. Applications range from cleansing of data in statistical hypothesis testing and modeling, performance degradation detection in systems prognostics, workload characterization and performance optimization for computing infrastructures, intrusion detection in network security applications, medical diagnosis and clinical trials, social network analysis and marketing, optimization of investment strategies, filtering financial market data, fraud detection in insurance and e-commerce applications. Methods for anomaly detection utilize statistical approaches such as hypothesis testing [1] and machine learning approaches such as one-class classification and clustering [2]. See [3] for a review.

An anomaly is defined qualitatively as an observation that significantly deviates from the rest of the sample. To quantify significant deviation a model is created that represents nominal observations and allows to compute deviation from it with a given false detection rate (type I error). In rare cases when instances of actual outliers are available in quantities sufficient to create a model describing the outlier observations, likelihood ratio-based statistical tests and two-class classification can be used with a specified missed detection rate (type II error).

Distributional and possibly other data-generating assumptions and tuning of various critical parameters are required to use existing anomaly detection methods. For example, when using the Mahalanobis distance, a mutivariate Gaussian assumption is made for the data generating mechanism. When using clustering, a number of clusters must be specified and a specific cluster formation mechanism must be assumed.

The analysis is becoming more laborious when observations are represented by heterogeneous data. For instance, a health monitoring system of a computing infrastructure that provides cloud services must continuously monitor diverse types of data about thousands of targets. The monitored data may include physical sensors, soft error rates of communication links, data paths, memory modules, network traffic patterns, internal software state variables, performance indicators, log files, workloads, user activities etc, all combined into a heterogeneous observation describing a target within a time interval. An anomaly detection system must consume all this data and alert the system administrator about anomalously behaving targets. In such environments it is unpractical to expect that the system administrator will possess sufficient skills to set and tune various anomaly detection parameters.

The tripoint arbitration similarity-based anomaly detection system is developed to address these new challenges. It has the following features designed-in:

- Makes no distributional or other assumptions about the data-generating mechanism.
- Operates without tuning of any parameters by the user. The appropriate distance function is selected based on the type of the data.
- Detects anomalies with a desired false detection rate. The user may specify the admissible false detection rate, otherwise $< 1\%$ is used by default.
- Handles seamlessly observations composed of heterogeneous components (numeric, text, categorical, time series, other) as long as an appropriate distance function is available for each data type.

*Oracle Labs, San Diego. Email: aleksey.urmanov@oracle.com

2 Tripoint Similarity and Clustering

Tripoint arbitration similarity method is based on a novel definition of similarity of data points. Consider a collection of samples $x_1, x_2, \ldots, x_n$ in R^m with the Euclidian distance $d_{ij} = d(i,j) = d(x_i, x_j)$ as the closeness measure for two points. Given a pair of points, (x_i, x_j), and an arbiter point $a \in R^m$, the *tripoint arbitration similarity* is defined as

$$S_a(x_i, x_j) = \frac{min(d(i,a), d(j,a)) - d_{ij}}{max(min(d(i,a), d(j,a)), d_{ij}))} \tag{2.1}$$

S_a takes values between -1 and $+1$ with the following interpretation. $S_a(x_i, x_j) = -1$ means that points x_i and x_j are completely dissimilar for the arbiter point a. $S_a(x_i, x_j) = +1$ means the points are completely similar. $S_a(x_i, x_j) = 0$ means the arbiter point cannot decide whether the points similar or dissimilar. All other non-zero values reflect the degree of similarity (positive values) or dissimilarity (negative values) of the pair for the arbiter.

For a set of arbiter points $A = \{a_1, a_2, \ldots, a_l\}$ the similarity is aggregated over all arbiters

$$S_A(x_i, x_j) = \frac{1}{l} \sum_{k=1}^{l} S_{a_k}(x_i, x_j) \tag{2.2}$$

Let $D = \{x_1, x_2, \ldots, x_n\}$ be a random sample from an unknown population. The *empirical pairwise tripoint arbitration similarity matrix* of sample D is defined as

$$S_D = [S_{D \setminus \{x_i, x_j\}}(x_i, x_j)] \tag{2.3}$$

Since similarity (2.1) ranges from -1 to +1 for any type of data, it is possible to combine similarities of different modalities of multimodal data into a single overall similarity. One rule for combining modal similarities is to follow common sense (analogously to [4]). For modal similarities with the same sign, the overall similarity becomes bigger than either of the modal similarities but still remains ≤ 1.

$$\begin{aligned} S_a(x_i, x_j) = {} & S_{a^{(1)}}(x_i^{(1)}, x_j^{(1)}) + S_{a^{(2)}}(x_i^{(2)}, x_j^{(2)}) \\ & - S_{a^{(1)}}(x_i^{(1)}, x_j^{(1)}) \cdot S_{a^{(2)}}(x_i^{(2)}, x_j^{(2)}) \end{aligned} \tag{2.4}$$

When modal similarities have different signs, the overall similarity is determined by the maximum absolute value but the degree of similarity or dissimilarity weakens.

$$S_a(x_i, x_j) = \frac{S_{a^{(1)}}(x_i^{(1)}, x_j^{(1)}) + S_{a^{(2)}}(x_i^{(2)}, x_j^{(2)})}{1 - \min(\mid S_{a^{(1)}}(x_i^{(1)}, x_j^{(1)}) \mid, \mid S_{a^{(2)}}(x_i^{(2)}, x_j^{(2)}) \mid)} \tag{2.5}$$

Using this definition of the pairwise similarity matrix for a data set, the clustering problem can be formulated as follows. Given a set of points $D = \{x_1, x_2, \ldots, x_n\}, x_i \in R^m$, the problem is to partition the set into an unknown number of clusters $C_1, C_2, \ldots, C_L$ so that points in the same cluster are similar and points in different clusters are dissimilar. This clustering problem can be casted into an optimization problem that can be efficiently solved using matrix spectral analysis methods

$$\min J(C_1, C_2, \ldots, C_L) \tag{2.6}$$

$$S_D(C_p, C_p) \geq 0, 1 \leq p \leq L \tag{2.7}$$

$$S_D(C_p, C_q) \leq 0, 1 \leq p < q \leq L \tag{2.8}$$

$S_D(C_p, C_q)$ is the average of all pairwise similarities of points from clusters C_p and C_q

$$S_D(C_p, C_q) = \frac{1}{\mid C_p \mid \mid C_q \mid} \sum_{i: x_i \in C_p} \sum_{j: x_j \in C_q} S_D(x_i, x_j) \tag{2.9}$$

and the objective function J is constructed to simultaneously satisfy $\min S_D(C_p, C_q)$ for $1 \leq p < q \leq L$ and $\max S_D(C_p, C_p)$ for $1 \leq p \leq L$. One such objective function is [5]

$$J = \sum_{1 \leq p < q \leq L} \frac{S_D(C_p, C_q)}{S_D(C_p, C_p)} + \frac{S_D(C_p, C_q)}{S_D(C_q, C_q)} \tag{2.10}$$

Dropping the constraints in (2.6) leads to the problem similar to the MinMaxCut formulation in [5] with pairwise associations given by the tripoint similarity.

To deal with the constraints in (2.6) an iterative approach was adopted. At the initial iteration the original problem is solved by partitioning the data set into two clusters using an appropriate objective function, for example, the MinMaxCut objective function in (2.10). At the next iteration each of the two clusters is partitioned in two and so forth. At each iteration the constraints are checked. Violation of the constrains serves as a stopping criterion for the iterations. The process of splitting clusters is stopped when no more clusters can be split without violating the inter-cluster dissimilarity constraint. This iterative procedure automatically produces the appropriate number of clusters. The tripoint arbitration clustering method uses matrix spectral analysis results to iteratively find the appropriate number of clusters by solving the problem (2.6).

3 Anomaly Detection System

The proposed anomaly detection system relies on tripoint clusters to determine a possible global structure in the data. Tripoint clustering finds automatically the

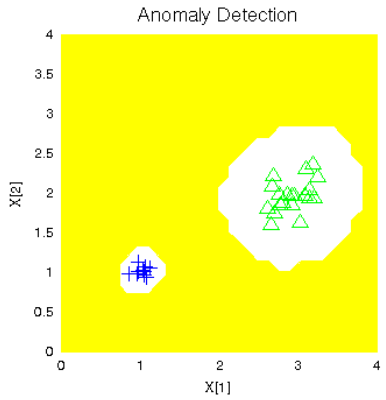


Figure 1: Outlier detection on artificial data with FAR $< 1\%$. All observations that lie in the yellow region will be detected as outliers given the nominal data that comprise two clusters.

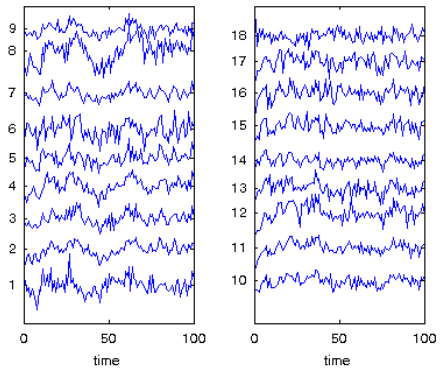


Figure 2: Anomalous target detection using time series data representing operating targets in a computing infrastructure. Target #18 is correctly detected as anomalously behaving compared to nominally behaving targets #1-17.

appropriate number of clusters and labels the observations with a cluster label $l = 1, 2, \ldots, L$. The resulting clusters $C_1, C_2, \ldots, C_L$ constitutes the nominal model based on the sample.

An *anomaly* is defined as an observation z for which all cluster-average similarities are positive or all points from clusters $C_1, C_2, \ldots, C_L$ are pairwise similar on average, i.e.

(3.11)
$$S_z(C_l) = \frac{1}{\mid C_l \mid} \sum_{i,j} S_z(x_i, x_j) > t_\alpha, i, j : x_i, x_j \in C_l,$$

where α is the desired false detection rate. The threshold t_α is determined from $\text{Prob}(S_z > t_\alpha) < \alpha$.

Figure 1 illustrates the area (in yellow color) points

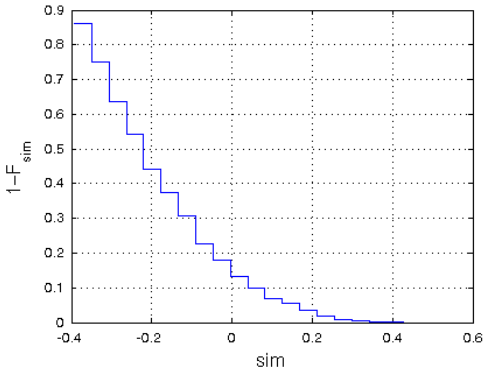


Figure 3: Estimated sampling distribution of S_z.

from which are considered as outliers with $< 1\%$ FAR for the data set compose of two clusters shown with blue crosses and green triangles. Tripoint clustering finds automatically two clusters in the data set and assigns cluster labels to the corresponding observations. Any new observation z for which $S_z(C_1) > 0.5$ or $S_z(C_2) > 0.5$ is a detected outlier or anomaly.

Figure 2 shows the results of clustering and anomaly detection in time series which represent certain attributes of monitored targets in a computing infrastructure. Tripoint clustering found 2 clusters in a pool of 17 targets shown on the left and right sides of the plot. A new target (#18) when presented to the anomaly detection system was correctly detected as anomalous.

And finally Figure 3 shows the estimated sampling distribution of S_z for a multivariate Gaussian data set with $n = 500$ points. By varying n, the values of t_α can be tabulated for different α for use in (3.11). For more rigorous calculation of t_α, the exact sampling distribution of S_z can be determined through Monte-Carlo simulations or asymptotic distribution theory.

References

[1] P.J. Rousseeuw and A. Leroy, *Robust regression and outlier detection*. New York: Wiley (1987).

[2] B. Schlkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson, *Estimating the support of a high-dimensional distribution*, Neural computation, 13(7), pp. 1443-1471 (2001).

[3] V.J. Hodge and J. Austin, *A survey of outlier detection methodologies*, Artificial Intelligence Review 22.2, pp. 85-126 (2004).

[4] I. B. Sirodza, *Quantum models and methods of artificial intelligence for decision-making and control*, Nauchnaya Mysl (2002), pp. 92.

[5] C. Ding, et al., *A minmaxcut spectral method for data clustering and graph partitioning*, Lawrence Berkeley National Laboratory, Tech. Rep 54111 (2003).

Measuring Anomalousness in Statistical Models

Thomas Veasey* Stephen Dodson*

1 Introduction.

Broadly speaking, unsupervised anomaly detection techniques fall into two categories: distance based approaches, which look at the distance between points or local density, for example [4, 7, 8], and statistical approaches, which are usually based on robust estimators and hypothesis testing, see for example [2]. In this paper, we study a statistical technique that can be used for unsupervised anomaly detection, based on a variation of the concept of the p-value of the value of a test statistic, see for reference [12].

We show how this quantity, which we will refer to henceforth as the q-value of an event, is well defined and generates an intuitive measure of the events' anomalousness in the presence of distribution modes, which would correspond to anomaly detection on clustered data. Furthermore, unlike many distance based approaches for anomaly detection, such as kNN proposed in [10], it naturally captures the relationship between the number of items in a cluster and their anomalousness. We discuss how to compute the q-value for some specific distributions and also numerical approaches that can be used to compute it for arbitrary distributions.

Finally, we study a class of high dimensional anomaly detection problems where the events of primary interest are statistically significant deviations in one, or a small number of, the dimensions. For such problems, many of the difficulties associated with high dimensional anomaly detection, data sparsity, choice of distance metric [1], runtime [6], and model size, can be circumvented for the proposed measure of anomalousness, without using dimension reduction techniques. In particular, we show how applying the q-value to order statistics on the individual dimension values leads to a natural measure for solving exactly this problem.

The authors' primary interest is in anomaly detection for application performance monitoring and network security. The data sets are nearly always collections of time series, and a common requirement for anomaly detection in this context is to provide alerts about anomalous system behaviour. We discuss a decision criterion, which we use to identify anomalies corresponding to unusual system or user behaviour given the ordering defined by the q-value.

Various characteristics are ubiquitous in the data sets we work with, and we highlight those which we have found are particularly important to capture in the statistical model in order to get accurate anomaly detection. Specifically, non-Gaussian distribution tails, proper handling of integer data, brakes and/or highly variable data rates and seasonality. Furthermore, the data sets are typically very large: monitoring data for large computer networks can comprise tens or even hundreds of thousands of performance metrics; common data related to network security, such as proxy logs have transaction rates in the thousands of events per second. Any time series model must be highly compact, and fast enough to compute on these data volumes. We have found that summarising the time series by small numbers of statistics, such as the mean, minimum and maximum of n metric values, allows us to scale to enormous data volumes with little loss in detection performance. In fact, varying the resolution, varying n for our example, effectively provides different insight into the data: different types of anomaly emerge for different choices.

We give results for a set of performance metrics generated monitoring an internet banking system over a three day period. This is around 12 GB and comprises around 33,500 distinct metric time series.

2 A Definition of Anomalousness.

The q-value is defined for any statistical model for which a distribution function exists. Specifically, the model must be some random variable from a probability space $(\Omega, \mathscr{F}, P)$ to some measure space $(X, \mathscr{A})$ and there must exist a measurable function $f : X \to \mathbb{R}^+$, where $\mathbb{R}^+$ denotes the non-negative real line with Borel algebra, which recovers the probabilities of the measurable sets of X. We define the q-value of an event $x \in X$ as:

$$q(x) = P(\{y : f(y) \leq f(x)\})$$

This is clearly well defined, since the closed interval $[0, f(x)]$ is Borel measurable and so its preimage is $\mathscr{A}$ measurable. Since $q(x)$ is a probability it takes values in the interval $[0, 1]$. Any subset of $[0, 1]$ has the usual strict total ordering of the reals, and so we

*Prelert Ltd., 156 Blackfriar's Road, London, UK. E-mail: {tveasey,steve}@prelert.com

can define a strict weak ordering of events by their anomalousness, i.e. $x >_a y$ if and only if $q(x) < q(y)$. In particular, anomalousness can be defined as some monotonic decreasing function of the q-value, for example $-\log(q(x))$.

It is interesting to compare this definition with the p-value, which is always defined in terms of some test statistic, say $T(x)$, of the event. In particular, the p-value is the probability of the test statistic exceeding its observed value given the null hypothesis: $P(T(y) > T(x)|H_0)$. In the case that the null hypothesis is the data set is Gaussian distributed with known mean m and covariance V, and the test statistic is $T(x) = \|x - m\|_2$, the so called two-tailed test, then the definitions coincide, since the probability density function is less than $f(x)$ in exactly the region where the test statistic is greater than $T(x)$. Note, however, that this case, a single mode symmetric model, is one of the few cases where the values coincide. Furthermore, the q-value makes no specific appeal to a null hypothesis. The idea is, given a statistical model of a data set, to define a quantity that naturally relates to the anomalousness of observed events, in much the same way as say then mean distance to k-nearest neighbours does for distance based anomaly detection.

If the data set to be analysed for outliers has been clustered then the corresponding statistical model will be multimodal. In particular, each cluster will typically generate a mode of the distribution, or local maximum in the density function. The q-value for an item from such a data set is therefore the (Lebesgue) integral of the density function over some region that (usually) contains multiple holes, corresponding to modes of the distribution.

The exact value for an item depends on the choice of statistical model. However, for reasonable models the value should be close to the fraction of items in lower density regions. In particular, we show that for any mixture of uniform random variable, $\sum_i f_i \times U(B_i)$ where f_i denotes the fraction of items in an m-dimensional cuboid $B_i = [a_{1,i}, b_{1,i}] \times [a_{2,i}, b_{2,i}] \times ... \times [a_{m,i}, b_{m,i}]$ and $U(B_i)$ is a uniform random variable on that cuboid, then the q-value of an item x is exactly the fraction of items for which the density $f(y) = {}^{f_{i_y}}/_{V(B_{i_y})}$ is less than or equal to $f(x)$, where i_z denotes the index of the box which contains item z and V denotes the volume function defined as $V(B_i) = \prod_j |b_{j,i} - a_{j,i}|$.

Proof. By definition we have that

$$\begin{aligned} q(x) &= \int 1\{f(y) \leq f(x)\} f(y) dy \\ &= \sum_i 1\left\{ \frac{f_i}{V(B_i)} \leq \frac{f_{i_x}}{V(B_{i_x})} \right\} \frac{f_i}{V(B_i)} V(B_i) \\ &= \sum_{\{i: d_i \leq d(x)\}} f_i \end{aligned}$$

where $1\{\cdot\}$ denotes the indicator function, and in the last line we have defined $d_i = {}^{f_i}/_{V(B_i)}$. We can interpret this summation as the fraction of items for which the density is less than or equal to the density at the item x.

We note, also, that any random variable can be approximated in distribution by a mixture of uniforms, since we may approximate the cumulative density function by a sequence of piecewise constant functions. So this result can be used, in conjunction with binary space partitioning, to estimate q-values for arbitrary smallish dimensional multivariate models. Also, the density function for any reasonable mixture model describing clustered data will be proportional to the fraction of items in a mode, in the vicinity of that mode, so items from clusters with fewer items will naturally have lower q-values.

3 Numerical Schemes for Calculating q-values.

For many univariate distributions the q-value can be evaluated in closed form. Otherwise, efficient numerical methods often exist for finding the density function level sets. However, for general multivariate distribution and mixture models numerical methods must be used. We review a couple of approaches. Perhaps the simplest scheme for computing the q-value, if the statistical model can be sampled is the following: generate n independent samples of the distribution, say $Y_n = \{y\}$, and define:

$$q_n(x) = \frac{|\{y \in Y_n : f(y) \leq f(x)\}|}{n}$$

We show that $q_n(x) \xrightarrow{a.s.} q(x)$ as $n \to \infty$. In fact, we can compute the asymptotic error distribution.

Proof. As before, define our system model to be a random variable Y with probability density function f. Let, $A(x)$ denote the $\mathscr{A}$ measureable set $f^{-1}[\{z : z \geq 0, z \leq f(x)\}]$, and $1\{A(x)\}$ denote the indicator function of $A(x)$. Given a random sample y of Y then, by definition, $1\{A(x)\}(y) = 1$ with probability $q(x)$ and 0 otherwise. Therefore, $1\{A(x)\}(Y)$, which we understand as $1\{A(x)\} \circ Y$, is a Bernoulli random variable

with success probability $p = q(x)$. By definition,

$$\frac{|\{y \in Y_n : f(y) \leq f(x)\}|}{n} \sim \frac{1}{n}\sum_{i=1}^{n} 1\{A(x)\}(Y)$$

Furthermore, $\sum_{i=1}^{n} 1\{A(x)\}(Y) \sim B(n,p)$, i.e. it is a binomial random variable with number of trials n and probability of success p. Noting that $B(n,p) \xrightarrow{a.s.} N(np, np(1-p))$ as $n \to \infty$ it follows that

$$\frac{1}{n}\sum_{i=1}^{n} 1\{A(x)\}(Y) \xrightarrow{a.s.} N\left(q(x), \frac{q(x)(1-q(x))}{n}\right)$$

In particular, it is normally distributed with mean $q(x)$ and variance $q(x)(1-q(x))/n$. The variance is maximised when $q(x) = 1/2$, and so $q_n(x) \xrightarrow{a.s.} q(x)$ as $n \to \infty$ for all x and we are done.

This is just a particular Monte Carlo scheme for evaluating the integral $\int 1\{f(y) \leq f(x)\} f(y) dy$. It has one important advantage over the other schemes we discuss: the same set of samples can be used for evaluating $q(x)$ for any x. This means it is particularly well suited to Sequential Monte Carlo methods for estimating the statistical model. Otherwise, the following methods will have lower error for a given running time.

A recursive stratified sampling, such as the MISER algorithm [9], will yield lower variance for the same sample size. Further speedup can be obtained by importance sampling. In particular, the integrand is identically zero where $f(y) > f(x)$. Therefore, if we are using a mixture model, with density function $f(y) = \sum_i \pi_i f_i(x)$ and can compute the regions $\{R_i\}$, bounded by $\left\{z_i : z_i = f_i^{-1}\left(f(x)/\pi_i\right)\right\}$, we should only sample outside the region $\bigcup_i R_i$. Finally, we note for a mixture of uniforms approximation, then storing the region densities in a red-black tree augmented with the fraction of items in each subtree means it is possible to compute the q-value for a new item in $O(\log(N))$ and update the data structure in $O(\log(N))$ where N is the number of uniforms, see [5] for details.

4 Statistically Significant Anomalies in Low Dimensional Subspaces.

For many problems in application performance monitoring and network security, anomalies of particular interest are statistically significant deviations in a small number of the raw measurement dimensions. For example, if a system has ten thousand performance metrics, which might comprise average response times of different database queries, responses per interval, errors per interval and so on, a system problem is likely to manifest itself as highly unusual values in a subset of the performance metrics. Similarly, many types of network attack amount to a small set of users doing highly unusual things at a given instant. For example, a port scan attack would correspond to a client sending requests to an unusually large range of server port addresses on a host in a relatively short period of time. For these problems, it is not necessary to try and model the distribution on the full space. Instead, we can accurately model its marginals, for example the individual performance metrics, or the distribution of a population for particular attributes, such as unique server port address requests in a fixed time interval. Then compute q-values on these, and aggregate the individual q-values to get an effective measure for overall anomalous at any given instance.

To understand why we must account for the number of dimensions in this aggregation process, consider the simple case that each marginal is a Gaussian. If there is no anomaly, we expect N independent samples from a Gaussian, where N is the number of dimensions. In the case that $N = 10,000$ the most extreme sample we expect to see is about 4 standard deviations, where as in the case that $N = 10$ the most extreme sample we expect to see is around 1.5 standard deviations. These correspond to q-values of $1 - \text{erf}\left(4/\sqrt{2}\right) = 6.3 \times 10^{-5}$ and $1 - \text{erf}\left(1.5/\sqrt{2}\right) = 0.13$, respectively.

Given a collection of N q-values, $\{q_i\}$, a heuristic we have found to be useful for computing an aggregate q-value is to compute the q-value on the order statistic

$$q^{(N)}(x) = P\left(\{y : f_{X^{(N)}}(y) \leq f_{X^{(N)}}(x)\}\right)$$

Here, $f_{X^{(N)}}(y) = \frac{N!}{1!(N-1)!}\left(F(y) - F(-y)\right)^{N-1} f(y)$ denotes the distribution of the most extreme sample as a function of y, from a collection of N independent identically distributed samples from a symmetric single mode distribution, and x is the value of the most extreme sample. In our case, we are not interested when the smallest q-value is too large, given the sample size. Therefore, we evaluate $P\left(\{y : |y| \geq |x|\}\right)$. This is equal to

$$\frac{2N}{2N}\left[(2t-1)\right]_{F_X(x)}^{1} = 1 - \left(1 - \min_i\{q_i\}\right)^N$$

In particular, we set our aggregate q-value to be $1 - (1 - \min_i\{q_i\})^N$. Note, $1 - F_X(x) = \min_i\{q_i\}/2$ follows from the assumptions about sample distributions and the definition of x. This result can be generalized to compute the aggregate q-value from the M most extreme samples for N dimensions under the same assumptions.

5 Test Data and Methodology.

The data set we analysed was gathered by the CA APM product monitoring three servers of an internet

banking site. Every performance metric is reported at 60s intervals, although some record transactions and are not necessarily available at this granularity. It contains 33,159,939 distinct records and 33,456 distinct time series. The data cover a period of 72 hours and so the total data rate is around 500,000 values per hour. There are 38 categories of metric; these include responses per interval, average response time, errors per interval, stall counts and average result processing. Note that various categories are split out by SQL command, host and so on, which accounts for the total number of distinct time series.

For this data set, we found it was sufficient to assume that the series were stationary. For other problems, capturing diurnal and weekly variation is important, for which we use radial basis function interpolation to fit the periodic temporal patterns. We chose to fit either a Gaussian distribution with unknown mean and precision, a gamma distribution with unknown shape and rate, or a log-normal distribution with unknown location and scale to the (assumed) stationary distribution of each time series. We use standard Bayesian techniques to estimate the parameters, and Bayesian model selection to choose among the models, see [3] for details on Bayesian model selection. Finally, on this data set we found it was very important to accurately account for time series comprising integer data with low variation, in particular, series for which particular integer values have significant probability. Such data are generally badly modelled by continuous distributions. We automatically detect this case, and model these data using a latent variable. In particular, we assume that the observed values are described by $X + U([0, 1])$, where we estimate X, and $U([0, 1])$ denotes a uniform random variable on the interval $[0, 1]$.

A large anomaly manifested itself as system performance degradation during the interval 32 to 35 hours after the start of the data set. In terms of the raw anomaly scores, which were obtained by aggregating individual time series q-values, this corresponded to a signal-to-noise ratio of around 330dB, where the noise level was taken as the median aggregate q-value. If the time series are ordered by their q-values at that time, then 560 of the 33,456 time series are significantly anomalous. These results indicated there was an operational issue with a specific component of the backend, which resulted in the response time of a subsection of the website (6 JSPs) having dramatically increased response times. In addition, there was a precursor to the main anomaly, at 27 hours after the start of the data set, which provided the system administrators with early warning of the specific problem before the main failure. This was detected in the performance metrics with a signal-to-noise ratio of around 65dB; however, this was not significant enough to result in user noticeable system performance degradation.

We generated two alerts, corresponding to these two incidents for this data set. Our algorithm to generate alerts from the raw aggregate q-values is based on both the signal-to-noise and the historical quantiles of the aggregate q-value, for which we use the data structure proposed in [11].

References

[1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, *On the Surprising Behavior of Distance Metrics in High Dimensional Space*, Proc. Int. Conf. on Database Theory, (2001), pp. 420–434.

[2] V. Barnett, and T. Lewis, *Outliers in Statistical Data*, Third Edition, John Wiley & Sons, Chichester, 2006.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[4] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, *LOF: Identifying Density-Based Local Outliers*, SIGMOD00: Proc. ACM SIGMOD Int. Conf. on Management of data, (2000), pp. 93–104.

[5] T. H. Cormen, C. E. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, Third Edition, The MIT Press, 2009.

[6] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, *Locality-Sensitive Hashing Scheme Based on p-Stable Distributions*, Proc. ACM Symp. on Computational Geometry, (2004), pp. 253–262.

[7] H. Fan, O. Zaïane, A. Foss, and J. Wu, *A Nonparametric Outlier Detection for Efficiently Discovering Top-N Outliers from Engineering Data*, Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD), (2006), pp. 557–566.

[8] E. M. Knorr, and R. T. Ng, *Algorithms for Mining Distance-Based Outliers in Large Datasets*, Proc. of the 24th International Conference on Very Large Data Bases, (1998), pp. 392–403.

[9] W. H. Press, and G. R. Farrar, *Recursive stratied sampling for multidimensional Monte Carlo integration*, Computers in Physics, vol. 4, (1990), pp. 190–195.

[10] S. Ramaswamy, R. Rastogi, and K. Shim. *Efficient Algorithms for Mining Outliers from Large Data Sets*, Proc. ACM SIGMOD Int. Conf. on Management of data, (2000), pp. 427–438.

[11] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri. *Medians and Beyond: New Aggregation Techniques for Sensor Networks*, Proc. of the 2nd Int. Conf. on Embedded Network Sensor Systems, (2004), pp. 239–249

[12] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability and Statistics for Engineers and Scientists*, Ninth Edition, Prentice Hall, 2011.

Identifying Precursors to Anomalies Using Inverse Reinforcement Learning*

Vijay Manikandan Janakiraman† Santanu Das‡ Bryan Matthews§ Nikunj Oza¶

Abstract

In this paper, we consider the problem of discovering candidate precursors to anomalies in a set of time sequenced data. Typical scenarios involving time sequential data include dynamical systems and general monitoring systems. In such scenarios, a precursor could be any event that frequently precedes a given event of interest. Anomalies are rare but significant events in time series data and identifying precursors to anomalies is vital in proactive management. In this work, an inverse reinforcement learning (IRL) based method is formulated to succinctly represent the nominal behavior and identify sequences that preceded the anomalous events. A preliminary evaluation is performed on flight recorded data identifying challenges and future directions for application.

1 Introduction

In many applications including finance, study of natural calamities and extreme weather, network security [1] etc., finding precursors to an event of interest (a phenomenon) is a task of high importance. The knowledge about precursors to these phenomena can be vital to proactive management of risk. If precursor events could be identied, appropriate alarming mechanisms can be designed to either prevent or at least minimize the deleterious consequences of the phenomenon. Anomalous events are rare but significant events which in many cases, lead to an abnormal behavior or a risky situation. In such cases, it is important to analyze and identify precursors that lead to anomalies for proactive risk management. This paper considers anomalies in time sequenced data and attempts to discover candidate precursors to the anomalous events.

2 Discovering Precursors to Anomalies

In this section, an algorithm using inverse reinforcement learning is proposed to identify candidate precursors to anomalies in time series data. The section proceeds by introducing some background in inverse reinforcement learning, using its solution to perform value function estimation and using the optimal value function, discover precursors.

2.1 Inverse Reinforcement Learning The goal of inverse reinforcement learning (IRL) is to determine the underlying reward function using observed behavior of the agent making decisions in a Markov Decision Process (MDP). A finite MDP is a tuple ($\mathcal{S}$, $\mathcal{A}$, $P_{s,a}$, γ and $R(s)$) where $\mathcal{S}$ is a state space with n states, $\mathcal{A}$ is an action space with k actions, $\{P_{s,a}\}$ are the state transition probabilities corresponding to an action a at state s, $\gamma \in [0, 1)$ is the discount factor, $R(s) \in \mathbb{R}$ is the underlying reward function. A policy π can be defined as any map $\pi : \mathcal{S} \mapsto \mathcal{A}$ and the corresponding value function at any state s_1 can be given by

$$V^{\pi}(s_1) = E[R(s_1) + \gamma R(s_2) + \gamma^2 R(s_3) + ...|\pi] \tag{2.1}$$

where the expectation is over the distribution of state sequences $(s_1, s_2, s_3, ...)$ following the policy π starting from s_1.

Given the setting above, the goal of standard reinforcement learning is to determine a policy π^* that maximizes $V^{\pi}(s)$ among all policies for all $s \in \mathcal{S}$. When the agent's reward function is known, this task can be achieved using existing techniques for value function estimation [2]. However, in several situations, the agent's behavior is not completely known, i.e., the reward function cannot be defined easily. In such situations, the expert's observed behavior can be used to either reconstruct the underlying reward function as in the case of inverse reinforcement learning [3] or construct optimal policies directly as in the case of apprenticeship learning [4].

Assuming availability of sampled trajectories (relevant to the problem involving time series in this paper), the IRL problem can be posed as in [3]. The sampled trajectories can be considered as demonstrations of both the expert and non-expert acting in the MDP. Using the trajectories, the value functions of the expert and non-expert policies can be determined as follows. Let the unknown reward function be parameterized as

$$R(s) = \alpha_1 \phi_1(s) + \alpha_2 \phi_2(s) + .. + \alpha_d \phi_d(s) \tag{2.2}$$

*Supported by the NASA System-wide Safety and Assurance Technologies (SSAT) Project.

†UARC, Nasa Ames Research Center, Moffett Field, CA
‡Verizon, Palo Alto, CA
§SGT Inc., Nasa Ames Research Center, Moffett Field, CA
¶Nasa Ames Research Center, Moffett Field, CA

where the ϕ_i represent the features of the reward function. The expert value function following policy π_E at state s_1 can be given by

$$\begin{aligned}(2.3)\quad V^{\pi_E}(s_1) &= E[R(s_1)+\gamma R(s_2)+...|\pi] \\ &= E[\alpha_1\phi_1(s_1)+\alpha_2\phi_2(s_1)+..+\alpha_d\phi_d(s_1) \\ &+\gamma\alpha_1\phi_1(s_2)+\gamma\alpha_2\phi_2(s_2)+..+\gamma\alpha_d\phi_d(s_2)+..|\pi_E] \\ &= E[\alpha_1(\phi_1(s_1)+\gamma\phi_1(s_2)+..)+\alpha_2(\phi_2(s_1)+\gamma\phi_2(s_2)+..) \\ &+...+\alpha_d(\phi_d(s_1)+\gamma\phi_d(s_2)+..)|\pi_E] \\ &= \alpha_1 E[(\phi_1(s_1)+\gamma\phi_1(s_2)+..)|\pi_E]+ \\ &+\alpha_2 E[(\phi_2(s_1)+\gamma\phi_2(s_2)+..)|\pi_E] \\ &+...+\alpha_d E[(\phi_d(s_1)+\gamma\phi_d(s_2)+..)|\pi_E] \\ &= \alpha_1\lambda_1+\alpha_2\lambda_2+..+\alpha_d\lambda_d\end{aligned}$$

where λ_i represent the feature expectations, i.e., the value function if the reward function is composed of $\phi_i(s)$ only. After calculating the feature expectations knowing the state sequences, the value function can be defined as a function of the unknown $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, .., \alpha_d]^T$ as follows.

$$(2.4)\qquad V^{\pi_E}(\boldsymbol{\alpha}) = \sum_{i=1}^{d}\alpha_i\lambda_i$$

Similarly, by knowing the sequence of states (trajectories) for J sub-expert/non-optimal policies, the $V^{\pi_j}(\boldsymbol{\alpha})$ can be calculated. The objective of IRL is to determine the coefficients α_i so that $V^{\pi_E}(\boldsymbol{\alpha}) \geq V^{\pi_j}(\boldsymbol{\alpha})$ for $j = 1, 2, ..J$. A linear programming problem can be solved for α_i as follows

$$(2.5)\qquad \min_{\alpha}\sum_{j=1}^{J}\zeta_j$$

$$(2.6)\qquad \text{subject to}\begin{cases} V^{\pi_j}(\boldsymbol{\alpha}) - V^{\pi_E}(\boldsymbol{\alpha}) - \zeta_j \leq 0 \\ \zeta_j \geq 0, j = 1, 2, .., J \\ |\alpha_i| \leq 1, i = 1, 1, .., d \end{cases}$$

2.2 Value Function Estimation The IRL problem gives an optimal $\boldsymbol{\alpha}$ which gives a model of the underlying expert's reward function. The reward function can then be used to determine the expert's value function using a regular reinforcement learning algorithm. Any of the methods described in [2] such as dynamic programming, monte carlo or temporal difference depending on availability of the system model, ability to sample etc. In this paper, considering sample time series from a policy as monte carlo samples, the value function is approximated as follows. For each policy π_j including the expert policy π_E, a sample trajectory is used to identify the state sequences and using the reward function obtained above, the values of every state in $\mathcal{S}$ is updated. This is repeated for several trajectories from the selected policy and the average returns are stored as state values.

2.3 Precursor identification The value function of the expert policy π_E obtained above can be used to compare a non-expert behavior to identify a possible precursor sequence. V^{π_E} can be thought of as the expert's value function and any action that is greedy with respect to the expert's value function gives the optimal policy π_* [2]. Let the greedy action at s be $a^*(s)$ and the corresponding value be $V^{\pi_*}(s)$. In our problem involving time series, the time sequence and the physics of the problem can be used to restrict the state space for searching optimal actions in some cases. A given test time series can be analyzed as follows. Using the state sequences of the test data and the obtained reward function, the state values $V^{\pi_{test}}$ can be estimated. By comparing the $V^{\pi_{test}}$ with V^{π_*}, we can indirectly evaluate the actions taken by the agent in the test trajectory. Let

$$(2.7)\qquad \Delta V = V^{\pi_{test}}(s) - V^{\pi_*}(s)$$

and if $\Delta V \leq 0$, then it would mean that a sub-optimal action has been taken by the agent executing the test policy and by comparing over the state sequence, we can identify a sequence of bad actions by the agent. As defined earlier, an optimal action is one that corresponds to a nominal time series while a non-optimal action would correspond to an anomalous sequence as defined in the IRL problem. It should however be noted that the test policy is evaluated just based on one time series and hence not an expectation. However, the goal is to identify the level of sub-optimality in the state sequences specifically executed by the test trajectory to identify the precursor and not for the policy in general. This assumption needs to be analyzed more in detail and will be considered in the future. Further, if the action space is well defined, instead of comparing the value functions as above, the actions of the test agent can be directly compared against the optimal actions of the expert and precursors can be identified by noting their difference.

3 Application to Flight Anomalies

In this section, the IRL based precursor discovery algorithm is evaluated on flight time series data sets ob-

tained from a FOQA (Flight Operations Quality Assurance) archive. Typical FOQA parameters consist of both continuous and discrete (categorical) data from the avionics, propulsion system, control surfaces, landing gear, the cockpit switch positions, and other critical systems. Each flight record can have up to 500 parameters in the form of time sequences and are sampled at 1 Hz.

Flight anomalies are of significant interest within the NASA System-wide Safety and Assurance Technologies (SSAT) project to assess the health of large commercial fleets of aircraft. In this paper, flights that violated exceedance thresholds on computed air-speed are considered as operational anomalies. A specific exceedance defined as computed air-speed above a certain threshold (in knots) at an altitude of 1000 feet is considered an operationally significant high-energy approach. The goal of this study is to discover precursors to such high-energy approach flights [5] for use in proactive flight management. The data set consists of about 20000 nominal flights (flights that did not violate the exceedance and considered optimal with respect to the exceedance) and about 250 anomalous flights.

3.1 Discovery of candidate precursor sequences

The FOQA raw data consists of more than 400 parameters recorded as time sequences during the flight. However, to overcome the curse of dimensionality in solving the Markov decision process in the IRL, the FOQA data is abstracted to represent the various events happening in a flight using a high level parameter such as the aircraft energy. With the given definition of an anomaly, the flight data is considered as a sample from an expert policy (π_E) if it doesn't flag the exceedance or a sample from a non-expert policy (π_j) if it flags the exceedance. A reward function $R(s)$ can be defined as a linear combination of several Gaussian functions defined with respect to the states s. It has to be noted that the state definition is given by $s = [E \quad D]^T$ where E represents the kinetic energy of the aircraft while D represents the distance in nautical miles to touchdown. The reward function $R(s)$ can be represented as in equation (2.2) where ϕ_i could represent Gaussian functions with mean μ_i and spread σ_i and d represents the total number of Gaussian functions in the state space. Using the reward function with unknown coefficients α_i the value function of each trajectory is calculated and the IRL problem is solved as in section 2.1. The optimal value of $\boldsymbol{\alpha}$ gives a model of the underlying reward function that when used to solve the associated MDP, results in maximum possibility of avoiding the given exceedance. The model hyper-parameters including d, μ_i, σ_i are determined based on cross-validating the learned model on a hold-out data set. Following section 2.1, the ΔV for a given test flight is calculated. A negative value for ΔV indicates that the given test flight performs inferior to the optimal policy π_* and a negative rate of ΔV indicates a sequential inferior behavior. These two features are used in defining precursor candidates for the given test flight. It should be noted that the problem in hand uses FOQA data that only records the state of the flight and no explicit information about the intentions/actions of the agent (a pilot) is available and hence we were restricted to comparing the value functions as mentioned in section 2.3

Using the identified precursor sequences of a given test flight in terms of the states s, the FOQA historical data can be used to identify the flight parameters that are abnormal. The identified precursor sequence points to a section of the flight prior to the adverse event where interesting precursor events can be discovered. By modeling a nominal distribution of the FOQA parameters, any abnormality can be detected by comparison against the nominal. The identified abnormal parameters may contain information about possible factors that lead to the adverse event. This is algorithmically analyzed and validated by a domain expert.

4 Results and Discussion

In this section, a high-energy approach flight is analyzed for precursors from 35 nautical miles until touchdown. Figure 1 shows the evolution of the fight in terms of the parameters reported by the algorithm as candidate precursors. The blue shaded region represents the nominal distribution of that parameter (99 percentile of the non-exceedance flights) The green shaded regions of the figure represent the sequence of precursors (a precursor window) as identified by comparing the flight's state values to V^{π_*}.

It can be observed from Figure 1 that the computed air-speed of the flight is high compared to the nominal distribution of the non-exceedance flights indicating that the test flight is indeed an example of a high-energy approach. Further, out of the 56 chosen parameters from the FOQA list, only 11 were listed as possible precursor parameters as these parameters were out of the nominal distribution in the precursor window. The algorithm also reported ground speed which is correlated with the computed air-speed, vertical speed, stabilizer position, engine speed, flight director specified speed etc. However, a close look at the discrete parameters reported by the algorithm gives a clear picture of the actions responsible for the anomalies, i.e., the landing gear has been deployed a little earlier compared to nominal flights and the flaps were deployed very late causing the aircraft to slow down late leading

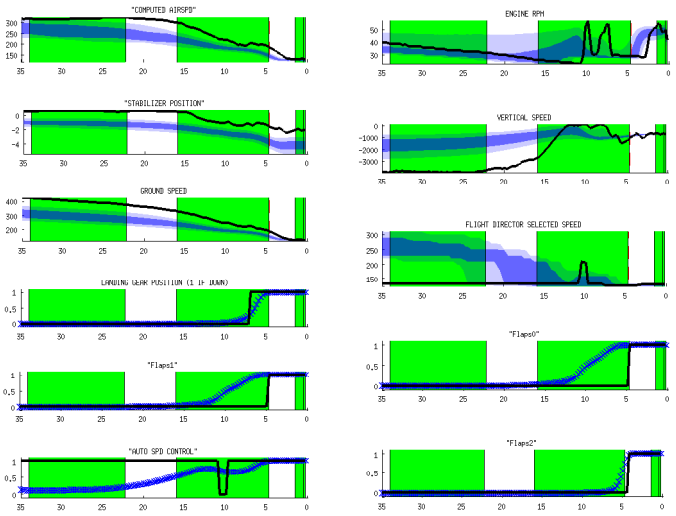


Figure 1: Figure showing the test flight trajectory (black curve) along with the precursor window (green region) as identified by the IRL algorithm. The nominal distribution of the continuous parameters such as computed air-speed, engine RPM, stabilizer position, vertical speed and ground speed are shown in blue - light blue region represents 0 - 99 percentile while dark blue region represents 25 - 75 percentiles. The nominal distribution of discrete variables including landing gear, flaps, auto speed control are shown by blue curve with markers indicating the probability of the variable having a value of 1.

to the exceedance (In the discrete plots, the marked blue curve represents the probability that a nominal discrete event takes a value of 1). Both these factors were validated by domain experts as probable precursors to the high speed exceedance. The initial high computed air-speed followed by a lack of optimal action (which is to deploy landing gear and flaps on time) in this case can be concluded as a valid precursor for flights violating the high-speed exceedance at 1000 feet altitude.

5 Conclusions

In this paper, a novel method to discover precursors to anomalies has been formulated using inverse reinforcement learning. It is argued that a value function of a non-expert, if compared against the optimal value function of an expert, can be used to identify instances of a "bad" or sub-optimal actions/situations in time series data. A high dimensional FOQA time series data has been abstracted and used for preliminary evaluation of the algorithm. The results indicate that the algorithm indeed finds the precursors that were validated by a domain expert. Although the analysis on a couple of flights gave us promising results, the algorithm is at infancy and requires extensive validation for which data sets with ground truth information about the precursors and anomalies are required. Also, for precursor identification, an appropriate performance metric will be identified for evaluation of this algorithm in future. Finally, some of the underlying hypotheses/assumptions of the algorithm will be tested in future.

References

[1] J. B. D. Cabrera, L. Lewis, X. Qin, W. Lee, R. Prasanth, B. Ravichandran, and R. Mehra, "Proactive detection of distributed denial of service attacks using mib traffic variables-a feasibility study," in *Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on*, 2001, pp. 609–622.

[2] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[3] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *in Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, 2000, pp. 663–670.

[4] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the Twenty-first International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 1–.

[5] S. Das, L. Li, A. Srivastava, and R. J. Hansman, "Comparison of algorithms for anomaly detection in flight recorder data of airline operations," in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*. American Institute of Aeronautics and Astronautics, 2012.