Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

A Generalized Student-t Based Approach to Mixed-Type Anomaly Detection

Yen-Cheng Lu[1], **Feng Chen**[2], **Yang Chen**[1] and **Chang-Tien Lu**[1]

[1] Virginia Tech, VA, USA
[2] Carnegie Mellon University, PA, USA
{kevinlu, yangc10, ctlu}@vt.edu, fchen1@cmu.edu

Abstract

Anomaly detection for mixed-type data is an important problem that has not been well addressed in the machine learning field. There are two challenging issues for mixed-type datasets, namely modeling mutual correlations between mixed-type attributes and capturing large variations due to anomalies. This paper presents BuffDetect, a robust error buffering approach for anomaly detection in mixed-type datasets. A new variant of the generalized linear model is proposed to model the dependency between mixed-type attributes. The model incorporates an error buffering component based on Student-t distribution to absorb the variations caused by anomalies. However, because of the non-Gaussian design, the problem becomes analytically intractable. We propose a novel Bayesian inference approach, which integrates Laplace approximation and several computational optimizations, and is able to efficiently approximate the posterior of high dimensional latent variables by iteratively updating the latent variables in groups. Extensive experimental evaluations based on 13 benchmark datasets demonstrate the effectiveness and efficiency of BuffDetect.

Introduction

Anomaly detection is an important problem that has received much attention in recent years. The objective is to automatically detect abnormal patterns and identify unusual instances, so-called anomalies. For example, in signal processing, the anomalies could be caused by random hardware failures or sensor faults, whilst anomalies in a credit card transaction dataset could represent fraudulent transactions. Anomaly detection techniques have been widely applied in a variety of domains, including cyber security, health monitoring, financial systems, and military surveillance.

Approaches to anomaly detection include distance based, local density based, one-class classifier based, and statistical model based methods (Chandola, Banerjee, and Kumar 2009). Most of these approaches are designed for single-type datasets, whereas most real world datasets are composed of a mixture of different data types, such as numerical, binary, ordinal, nominal, and count. Direct application of these approaches to mixed-type data leads to the loss of significant correlations between attributes, and their extension to mixed-type data is technically challenging. For example, the distance based approach relies on well-defined measures to calculate the proximity between data observations, but there is no uniform measure that can be used for mixed-type attributes. The statistical model based approach relies on modeling the correlations between different attributes, but there is no uniform correlation measure available for mixed-type attributes. There are a limited number of methods designed for dealing with mixed-type data, including LOADED (Ghoting et al. 2004) and RELOADED (Otey, Parthasarathy, and Ghoting 2006), but these methods focus on computational efficiency and their correlation modeling between mixed-type attributes is heuristically driven, lacking a solid statistical foundation. There are two main challenges for mixed-type datasets, namely modeling mutual correlations between mixed-type attributes and capturing large variations due to anomalies. In the KDD panel discussion (Piatetsky-Shapiro et al. 2006) and the resulting position paper (Yang and Wu 2006), dealing with mixed-type data has been identified as one of the ten most important challenges in data mining for the next decade.

In this paper, we propose a statistical-based approach to address the above challenges. We begin by presenting a new variant of the generalized linear model that can capture the mutual correlations between mixed-type attributes. Specifically, the mixed-type attributes are mapped to latent numerical random variables that are multivariate Gaussian in nature. Each attribute is mapped to a corresponding latent numerical variable via a specific link function, such as logit function for binary attribute and log function for count attributes. Using link functions to model attributes of different types is one of the most popular strategies for modeling non-numerical data (Tyler 2008). Based on this strategy, the dependency between mixed type attributes is captured by the relationship between their latent variables using a variance-covariance matrix. We then incorporate an "error buffer" component based on Student-t distribution to capture the large variations caused by anomalies. Student-t distribution has been widely used in robust statistics to minimize the effects of anomalies in a variety of statistical models, e.g., multivariate regression, Kalman filtering, clustering, and independent component analysis. By fitting the data into the

model, the error buffer absorbs all of the errors. The detection process then revisits the error buffer and detects those abnormal instances with irregular magnitudes of error. The main contributions of our study are summarized as follows:

1. **Novel Anomaly Detection Model**: To the best of our knowledge, this is the first anomaly detection model that incorporates an "error buffering" scheme and is supported by solid statistical foundation.
2. **Integrated Framework**: A new type of integrated framework capable of performing general purpose anomaly detection on mixed-type data is proposed. It does not require a set of labeled training data.
3. **Enhanced Statistical Approximation**: A novel approximate inference process for anomaly detection is designed based on the INLA framework. The computational efficiency of the INLA method is further improved to enhance its ability to process high dimensional mixed-type datasets.
4. **Extensive experiments to validate the effectiveness and efficiency of BuffDetect**: Results on real benchmark datasets demonstrated that our proposed approach performed more effective than most existing approaches with comparable computational efficiency.

Model and Framework

This section first formalizes the problem, then discusses the modeling of mixed-attributes in the framework of generalized linear models, introduces an error buffering component to handle anomalous effects, and finally presents an integrated Bayesian hierarchical model.

Problem Definition

Assume that we are given N instances in a dataset $S = \{s_1 \cdots s_N\}$, in which each instance s has P response (or dependent) variables $\{y_1(s) \cdots y_P(s)\}$ and D explanatory (or independent) variables $\{x_1(s) \cdots x_D(s)\}$. The separation of response (e.g., house price) and explanatory (e.g., house size, number of rooms) variables is decided based on users' domain knowledge, and all the variables could be regarded as response (dependent) variables as a special case. The dependent variables could consist of different data types, combining numerical, binary, and/or categorical variables, whilst the explanatory attributes are typically set numerical. The objective is to model the data distribution and identify those instances that contain abnormal response variables or explanatory attributes.

Types of Anomalies: Since the attributes have been separated into two types, the anomalies can also be introduced as either abnormal response variables or unusual explanatory attributes. Thus, we define three types of anomalies based on their originating attribute groups.

1. **Type I Anomaly**: caused by abnormal values in response variables.
2. **Type II Anomaly**: caused by abnormal values in explanatory attributes.
3. **Type III Anomaly**: caused by abnormal values for both response variables and explanatory attributes.

Any object that has attributes behaving belong to one of the above three categories is defined as an anomaly. An anomalous object usually deviates far away from the normal trend and can hence be detected by using our statistical model.

Predictive Process: The first step utilizes numerical response variables, which are typically assumed to follow a Gaussian distribution model. Thus, the Gaussian predictive process can be applied here. The following regression formulation represents the behavior of the instances:

$$Y(s) = X(s)\beta + \omega(s) + \varepsilon(s) \tag{1}$$

This formulation implies that similar instances should have similar explanatory attributes. The regression effect β is a $P \times D$ matrix, which represents the weights of the explanatory attributes with regard to the response variables. The dependency effect $\omega(s)$ is a Gaussian process used to capture the correlation between the response variables and a local adjustment is provided for each response attribute. The error effect $\varepsilon(s)$ captures the difference between the actual instance behavior and normal behavior. The instances are assumed to be identical and independently distributed (*i.i.d.*), which introduces the Gaussian likelihood as

$$\pi(Y(s)|\eta(s)) \sim \mathcal{N}(Y(s)|\eta(s), \sigma^2_{num}), \tag{2}$$

where $\eta(s) = X(s)\beta + \omega(s) + \varepsilon(s)$, and σ^2_{num} is set to a small number in order to leave the random effects for ω and ε to be captured.

GLM and Robust Error Buffering

The idea of GLM (Generalized Linear Model) is to present the non-numerical type data in a continuous space by a link function, so non-numerical response variables must be assigned with proper distributions. Taking the binary response type as an example, each response variable is assumed to follow a Bernoulli distribution, such that $\pi(Y(s)|\eta(s)) \sim Bernoulli(g(\eta(s)))$, where g is a logit link function that converts the numerical likelihood value to the success probability of Bernoulli distribution. In this case, we use the sigmoid function for this conversion, i.e., $g(x) = \frac{1}{1+exp(-x)}$. GLM can handle not only binary data, but also count, categorical, and multinomial data types.

One of the major components in the proposed new algorithm is the robust error buffer. A latent random variable is designed to absorb the error effect caused by measurement error, noise, or abnormal behaviors. The purpose of this mechanism is to separate the expected normal behavior from the errors. Instead of a simple Gaussian distribution, Student-t distribution is considered to model the error variation ε. Student-t distribution has a heavier tail than Gaussian distribution, and has been widely used in robust statistics (Yang and Wu 2006). The tail heaviness is controlled by setting the degrees of freedom. When the degree of freedom approaches infinity, Student-t distribution becomes equivalent to Gaussian distribution. The probability density func-

tion of a Student-t distribution $st(0, df, \sigma)$ is defined as

$$p(\varepsilon) = \frac{\Gamma(\frac{df+1}{2})}{\Gamma(df/2)}(\frac{1}{\pi df\sigma})^{\frac{1}{2}}(1 + \frac{\varepsilon^2}{df\sigma})^{-\frac{df}{2}-\frac{1}{2}}, \quad (3)$$

where df is the degrees of freedom σ is the scale parameter, and Γ is the gamma function. Our model treats the error effect $\varepsilon(s)$ as a zero mean Student-t process, with a diagonal covariance matrix and a preset degree of freedom. There are two benefits to be gained by adding this error buffer in the model. First, the parameter estimation becomes robust and the normal behavior is modeled more accurately. Second, the errors are absorbed by this latent variable, making it possible to detect anomalies by checking the value of variables.

A Bayesian Hierarchical Model

Integrating the components introduced in the above subsections allows us to complete the design of the new algorithm. Figure 1 shows the graphical representation of our model.

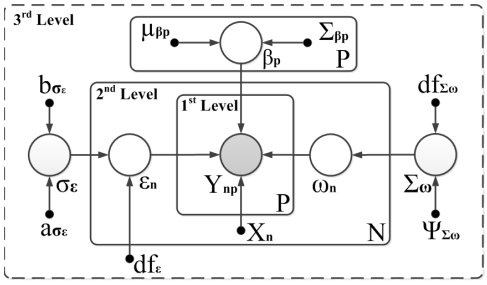


Figure 1: Graphical Model Representation

Our model is based on a Bayesian hierarchical model, which enables the parameters to be automatically learned while also reserving the option for users to set the hyper-parameters based on their prior knowledge.

The first (observation) level of the hierarchical model captures the relations between the response variables. This level refers to the predictive process and the GLM. This level models the relations between the latent effects and the response variables.

The second (latent variable) level is the latent variable level. This level contains the latent elements that refer to the effects of the error buffer and the correlation effect i.e., ω and ε. The main purpose of this level is to model the relations of the latent variables and the parameters. More specifically, we can form the following equations:

$$\omega(s) \sim \mathcal{N}(\omega(s)|0, \Sigma_\omega), \quad (4)$$
$$\varepsilon(s) \sim St(\varepsilon(s)|0, \sigma_\varepsilon, df), \quad (5)$$

where Σ_ω is the covariance matrix used to model the covariance between the response variables, σ_ε is a diagonal covariance matrix that indicates the variances of the error effects, and df denotes the degree of freedom parameter.

For convenience, the parameter representing the regression effect is placed in this level to facilitate the derivation because it is assigned a Gaussian distribution along with the dependency effect and is similar to the Student-t distribution of ε. The prior distribution of β can be represented by

$$\beta_p \sim \mathcal{N}(\beta|\mu_{\beta p}, \Sigma_{\beta p}), \quad (6)$$

where each β_p is the regression effect corresponding to the p-th response variable, and $\mu_{\beta p}$ and $\Sigma_{\beta p}$ are the hyper-parameters which define the Gaussian distribution of each β_p. We use the symbol ν to denote the latent variable set, such that $\nu = [\varepsilon, \omega, \beta]$.

The third (parameter) level defines conjugate priors for the model parameters, including the covariance matrix of ω and the covariance matrix of ε, i.e., Σ_ω and σ_ε. To reduce the dimensionality of θ, we only keep the variance of ω and ε in each response variable and the correlation between response variables:

$$\sigma^2_{\varepsilon p} \sim IG(a_{\varepsilon p}, b_{\varepsilon p}), \quad (7)$$
$$\Sigma_\omega \sim IW(\Phi, df_\omega), \quad (8)$$

The variance $\sigma^2_{\varepsilon p}$ at each response variable is assigned an inverse gamma distribution, and the covariance matrix of ω is assigned an inverse Wishart distribution. The symbols $a_{\varepsilon p}$,$b_{\varepsilon p}$,Φ, and df_ω denote the hyper-parameters of these prior distributions.

Approximate Bayesian Inference

Because the proposed Bayesian hierarchical model is analytically intractable, this section presents an approximate inference approach, which integrates Integrated Nested Laplace Approximation (INLA) (Rue, Martino, and Chopin 2009) and several computational optimizations, and efficiently approximates the posterior of high dimensional latent variables by iteratively updating the latent variables in groups.

The BuffDetect Framework

Algorithm 1 presents the framework of BuffDetect that is composed of three major components, including Laplace approximation, variable estimation, and anomaly detection.

Phase 1 - Laplace Approximation. Steps 1 to 10 show how the INLA framework is established by two Laplace approximations in a nested structure. The outer loop performs a maximum *a posteriori* (MAP) to θ. Since we can represent the posterior distribution of θ in the form:

$$\pi(\theta|Y) \propto \frac{\pi(\nu, Y, \theta)}{\pi(\nu|Y, \theta)}, \quad (9)$$

and our objective is to maximize $\pi(\theta|Y)$, we treat the posterior density function $\pi(\theta|Y)$ as an objective function and this becomes an optimization problem. The next step is to assign values for each input to this objective function $\pi(\theta|Y)$. Thus, the inner loop (steps 4-6) runs for the Laplace approximation to $\pi(\nu|Y, \theta)$. Applying a Taylor expansion to $\pi(\nu|Y, \theta)$, we can achieve an analytical formulation that restructures this density function into the quadratic form:

$$\pi(\nu|Y, \theta) = -\frac{1}{2}\nu^T Q\nu + \nu^T b. \quad (10)$$

Then, for each iteration at step 5, the latent variable set ν can be updated by $\nu = Q^{-1}b$. After a few iterations, ν will converge to a local optimum. This updating method, known as

Algorithm 1 BuffDetect

Require: The response variables Y and explanatory attributes X
Ensure: The anomalous instances
1: set $\theta = \theta_0$
2: **while** $\theta \neq argmax_\theta(\pi(\theta|Y))$ **do**
3: set$\nu = \nu_0$
4: **while** $\nu \neq argmax_\nu(\pi(\nu|Y,\theta))$ **do**
5: $\nu = update\_\nu$
6: **end while**
7: $\hat{\nu} = \nu$
8: $L = likelihoodof\pi(\theta|Y,\hat{\nu})$
9: $\theta = update\_\theta(L)$
10: **end while**
11: θ_{sample} = sample from neighborhood of θ
12: set $L_{\theta_{samples}}, \hat{\nu}_{\theta_{sample}} = \phi$
13: **for all** θ_s in $\theta_{samples}$ **do**
14: $\hat{\nu}_{\theta_s} = argmax_\nu(\pi(\nu|Y,\theta_s))$
15: $L_{\theta_s} = likelihoodof\pi(\theta|Y,\hat{\nu}_{\theta,s})$
16: put $\hat{\nu}_{\theta_s}$ into $\hat{\nu}_{\theta_{samples}}$
17: put L_{θ_s} into $L_{\theta_{samples}}$
18: **end for**
19: $weight = normalize(L_{\theta_{samples}})$
20: $\nu^* = \hat{\nu}_{\theta_{samples}} * weight$
21: $\varepsilon^* = getErrorBuffer(\nu^*)$
22: set $AnomalySet = \phi$
23: **for all** ε^*_s in ε^* **do**
24: **if** $\varepsilon^*_s > ErrorThreshold$ **then**
25: put s in $AnomalySet$
26: **end if**
27: **end for**
28: **return** $AnomalySet$

Iterative Reweighted Least-Squares (IRLS) (Gentle 2007), usually converges within 5 iterations. Steps 7-9 calculate the value of the objective function $\pi(\theta|Y)$ at the local optimum $\hat{\nu}$, and update θ according to this value. The iterations are continued until θ converges.

Phase 2 - Variable Estimation. After obtaining the mode of $\pi(\theta|Y)$, say $\hat{\theta}$, samples can be collected from the neighbors of $\hat{\theta}$ in the space of θ and used to estimate the optimum values of θ and ν. This is similar to the importance sampling (Press et al. 2007) approach often used for numerical analysis, the difference being that samples are only collected from the mode region in the space. Steps 11-19 demonstrate this process.

Phase 3 - Anomaly Detection. Finally, steps 20-28 show the process used to detect anomalies. Having identified the optimum ν, say ν^*, we are able to use the optimized latent variable set to perform anomaly detection. We begin by extracting the fitted error buffer ε^* from ν^*, and examining its contents. Step 24 indicates how the anomalies are detected in terms of a pre-determined threshold. This threshold is typically set to 3 times the standard deviation, i.e., Z-score equals to 3, just as labeling anomalies for a Gaussian distribution.

Computational Cost and Optimization

The computational cost is usually a concern for statistical modeling techniques, especially if the method is to be applied as an online method, the efficiency becomes more important. Here, the strategy is to approximate complex computations, accepting a slight drop in accuracy to gain a significant increase in efficiency. These optimizations were successfully tested experimentally, as described in the Experimental Results section.

Latent Computational Optimization: In Algorithm 1, step 5 is a major bottleneck in the framework shown . The high dimensionality of the latent variable set makes the computation of the matrix inversion very slow. To optimize this step, the update is approximated by separating ν into $\varepsilon, \omega, \beta$ and then updating these three variables iteratively as in Gibbs sampling method. Algorithm 2 demonstrates the idea behind the approximation process. Steps 1-9 show how the original process breaks into three smaller processes. Steps 2, 5, and 8 update the latent variables in the same sense as the original one. A Taylor expansion is performed on each of the three latent variables separately and inserted into the Gaussian quadratic form in equation (10), updated by IRLS, i.e. iteratively performing $\beta = Q_\beta^{-1} b_\beta, \varepsilon = Q_\varepsilon^{-1} b_\varepsilon$, and $\omega = Q_\omega^{-1} b_\omega$. In each call on $update\_\nu$, two variables are fixed and the remaining one updated. As the result, the computational cost is significantly reduced by adopting this modification. The complexity for the original INLA update is $O((P(2N+D))^3)$, which refers to the size of the latent variable set on the matrix inversion, while the complexity of the optimized update is reduced to $O(N^3)$. When the data size is large, we further reduce the complexity by sampling small portion of data and detect the anomalies by the model built by the samples. When the size of sampled instances and the number of sample batches are enough, the accuracy is maintained.

Algorithm 2 $update\_\nu$

Require: The original latent variable $\varepsilon, \omega, \beta$
Ensure: The updated latent variable $\varepsilon_{new}, \omega_{new}, \beta_{new}$
1: **while** $\beta \neq argmax_\beta(\pi(\beta|Y,\theta,\varepsilon,\omega))$ **do**
2: $\beta = update\_\beta(\varepsilon,\omega)$
3: **end while**
4: **while** $\varepsilon \neq argmax_\varepsilon(\pi(\varepsilon|Y,\theta,\beta,\omega))$ **do**
5: $\beta = update\_\varepsilon(\beta,\omega)$
6: **end while**
7: **while** $\omega \neq argmax_\omega(\pi(\omega|Y,\theta,\varepsilon,\beta))$ **do**
8: $\omega = update\_\omega(\varepsilon,\beta)$
9: **end while**
10: **return** $\varepsilon_{new} = \varepsilon, \omega_{new} = \omega, \beta_{new} = \beta$

Approximate Parameter Estimation: Another bottleneck in Algorithm 1 is that when the dimension of the parameter space is huge, sampling and evaluating the weight from the $\hat{\theta}$ neighborhood is computationally intensive. We approximate the optimum estimation by reducing the size of samples in step 11. Although the estimated parameters will not exactly match the optimum, the latent variable set still follows the trend if the estimated parameters are close to the optimum. We also observed that the approximated $\hat{\theta}$ was usually sufficiently close to the optimum solution of θ. Since the anomaly detection framework is only interested in the latent variables, having a minor bias on parameter estimation will not actually affect the detection results.

Correlation Parameter Reduction: Since the complexity of the process is proportional to the dimensionality of the parameters, one way to reduce the complexity is to reduce the number of parameters. For this optimization, we applied a Mutual Information (Steuer et al. 2002) method to calculate the scores of the dependency between each of the response attributes. By applying a user-defined parameter K, it is only necessary to consider the top K attribute correlations to be fitted. This approximation reduces the correlation parameter from $\binom{P}{2}$ to K. When P is a large number, this approximation significantly reduces the computational cost.

Experiments

This section evaluates the effectiveness and efficiency of the proposed BuffDetect framework on 13 real-life datasets. The experiments were conducted on a Windows 7 machine with a 2.4 GHz Intel Dual Core CPU and 4GB of RAM.

Experimental Design

Benchmark Approaches: Five benchmark approaches were evaluated, namely LOADED (Ghoting et al. 2004), RELOADED (Otey, Parthasarathy, and Ghoting 2006), KNN-CT, LOF-CECT, and SVM-PCT. LOADED and RELOADED are mixed-type anomaly detection methods and the remaining three methods are integrated single-type anomaly detection methods. The other three approaches are the combinations of six single-type anomaly detection methods, including three numerical anomaly detection methods (KNN, LOF (Breunig et al. 2000) and SVM (Ramaswamy, Rastogi, and Shim 2000)) and three categorical anomaly detection methods (CT, CECT and PCT, all from (Das 2007)). Das and Schneider (Das 2007) have shown that their methods outperformed other categorical methods, and thus we used their methods as the benchmark for categorical attributes. The integrated methods performed the detection procedures separately, and combined the scores into the same measure by a normalization process. For both LOADED and RELOADED, we tried popular settings of the model parameters (correlation threshold = [0.1, 0.2, 0.3, 0.5, 0.8, 1]; frequency threshold = [0, 10, 20]; τ = [1,2,3,5]), and reported the best results for each dataset based on true anomaly labels. For the other three approaches, the parameters were selected based on 10-fold cross validations.

Real Datasets: We validated our approach using 13 real datasets, all of which can be found from UCI machine learning repository (Frank and Asuncion 2010). Table 1 shows detailed information of these datasets.

Anomaly Labels: Because the above datasets do not provide true anomaly labels, we preprocessed the data in two different ways:

1. Rare Classes. For the first group of datasets (Abalone, Yeast, WineQuality, Heart and Autompg), we found that there are rare categorical classes in the datasets. Those rare class instances were defined as true anomalies.

2. Random Shifting. For the rest of datasets, we regarded all the data objects as normal objects, and followed the standard contamination procedure as used in (Riani, Atkinson, and Cerioli 2009; Cerioli 2009) to generate anomalies. For numerical attributes, we randomly selected 2.5 percent of objects and shifted the numerical values 3 standard deviations. For binary attributes, we randomly selected 2.5 percent of objects and switched the binary values to the alternative values. We preprocessed the data for each dataset with 20 different artificial anomaly combinations, and calculated the average of the 20 results for each test.

Table 1: Information in Real Datasets

Dataset	*Instances*	*Attrs*	*Type*
Abalone	4177	9	Categorical, Numerical
Yeast	1324	9	Categorical, Numerical
WineQuality	4898	12	Categorical, Numerical
Heart	163	11	Categorical, Binary
Autompg	398	8	Categorical, Numerical
Wine	178	13	Categorical, Numerical
ILPD	583	10	Binary, Numerical
Blood	748	5	Binary, Numerical
Concrete	103	10	Binary, Numerical
Parkinsons	197	23	Binary, Numerical
Pima	768	8	Binary, Numerical
KEGG	53413	23	Binary, Numerical
MagicGamma	19020	11	Binary, Numerical

Experimental Results

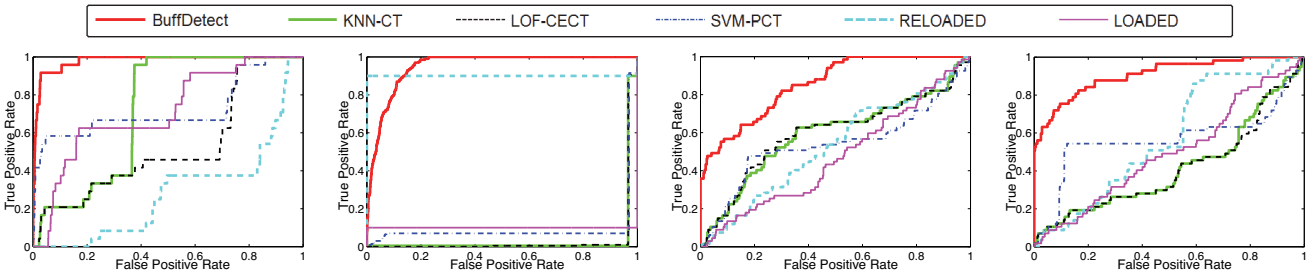
The first set of experiment tests the datasets for detection accuracy. Table 2 compares the precision, recall, and F-measure of the different approaches. The benchmark methods LOF-CECT and SVM-PCT failed to process large real datasets *KEGG* and *MagicGamma*.

Detection Accuracy: The average precision and recall show our approach outperformed the benchmark approaches on the real datasets (Table 2). The results demonstrated that in most cases the best performance on precision was obtained using the proposed method, indicating that the instances identified as anomalies by BuffDetect were mostly true positives. Our approach also achieved the highest recall on 8 datasets. Although SVM-PCT and KNN-CT had higher recall on the rest, it suffered from a high false positive rate. For example, for the *Concrete* dataset, SVM-PCT achieved a slightly better recall (78%), but with only 21% precision. This implies that SVM-PCT labeled around 30% of the instances in the whole dataset as positive, so it is not surprising it detected more true anomalies. In other words, those methods that combine a high recall with low precision will inevitably suffer from a large percentage of false alarms. In many situations, validating such false alarms can be expensive. The F-measure demonstrated consistent patterns.

Figure 2 shows the ROC curves of the different methods for 4 selected datasets. Interestingly, although LOADED and RELOADED had lower precision and recall, their ROC curves are reasonable, which implies that the anomalous scores generated by these two methods actually captured the anomalous patterns. For example, LOADED and RELOADED identified the top T instances in the rank of anomalous scores as anomalies even though the top ranked T instances were not identified correctly. The ROC curves demonstrate this phenomenon, as the curves of these two methods are usually raised when the false positive rate in-

Table 2: Detection Rate Comparison among Datasets (Precision, Recall, F-measure)

Dataset	BuffDetect	LOADED	RELOADED	KNN-CT	LOF-CECT	SVM-PCT
Abalone	**0.25**, **0.62**, **0.36**	0.00, 0.00, 0.00	0.00, 0.00, 0.00	0.16, 0.33, 0.22	0.02, 0.04, 0.03	0.20, 0.42, 0.27
Yeast	0.55, **0.67**, 0.60	**0.63**, 0.63, **0.63**	0.00, 0.00, 0.00	0.29, 0.57, 0.38	0.05, 0.10, 0.07	0.21, 0.44, 0.28
WineQuality	**0.33**, **0.65**, **0.44**	0.10, 0.10, 0.10	0.00, 0.00, 0.00	0.03, 0.06, 0.04	0.02, 0.04, 0.03	0.04, 0.07, 0.05
Heart	0.95, 0.75, **0.84**	0.51, 0.51, 0.51	**1.00**, 0.16, 0.28	0.46, **0.76**, 0.57	0.45, 0.75, 0.56	0.24, 0.43, 0.31
Autompg	**0.47**, **1.00**, **0.64**	0.29, 0.29, 0.29	0.33, 0.57, 0.42	0.00, 0.00, 0.00	0.00, 0.00, 0.00	**0.47**, **1.00**, **0.64**
Wine	**0.60**, **0.71**, **0.65**	0.10, 0.10, 0.10	0.41, 0.34, 0.37	0.35, 0.62, 0.45	0.34, 0.62, 0.44	0.31, 0.62, 0.41
ILPD	**1.00**, 0.28, **0.44**	0.12, 0.12, 0.12	0.01, 0.00, 0.00	0.26, **0.50**, 0.34	0.14, 0.26, 0.18	0.26, **0.50**, 0.34
Blood	**1.00**, 0.42, **0.59**	0.09, 0.09, 0.09	0.11, 0.39, 0.17	0.28, 0.55, 0.37	0.06, 0.11, 0.08	0.29, **0.56**, 0.38
Concrete	**0.98**, 0.72, **0.83**	0.13, 0.13, 0.13	0.23, 0.27, 0.25	0.37, 0.69, 0.48	0.37, 0.69, 0.48	0.21, **0.78**, 0.33
Parkinsons	**0.92**, **0.54**, **0.68**	0.06, 0.06, 0.06	0.13, 0.48, 0.20	0.27, 0.53, 0.36	0.26, 0.52, 0.35	0.28, 0.53, 0.37
Pima	**1.00**, 0.40, **0.57**	0.08, 0.08, 0.08	0.17, 0.47, 0.25	0.28, 0.54, 0.37	0.07, 0.13, 0.09	0.28, **0.55**, 0.37
KEGG	**0.65**, **0.76**, **0.70**	0.01, 0.01, 0.01	0.13, 0.28, 0.18	0.38, 0.75, 0.50	N/A	N/A
MagicGamma	**0.42**, **0.67**, **0.52**	0.01, 0.01, 0.01	0.03, 0.43, 0.06	0.33, 0.66, 0.44	N/A	N/A



(a) Abalone (b) WineQuality (c) ILPD (d) Blood Transfusion Service

Figure 2: ROC Curves of the anomaly detection in real datasets among approaches

creased. This implies that labeling the anomalies by setting a Z-score threshold on the error is a better discriminant.

The ROC curves also show that our approach always reached the 100% true positive rate faster than other benchmark approaches. This indicates that our error buffering mechanism has successfully captured the anomalous patterns. In all of the plots of Figure 2, the competitor methods show sharp increases in TPR at low FPR, and followed by flat curves afterward. This reveals that those methods had already labeled most of the correct positives at low FPR and then started reporting false alarms. Taking the test on *ILPD* as an example, BuffDetect obtained 100% precision and 28% recall, whilst SVM-PCT averaged 26% precision and 50% recall. The ROC curve in 2(c) actually shows that the TPR value for SVM-PCT at the point where FPR = 74% and precision = 26%, was higher than the TPR of BuffDetect at FPR = 0%. However, if we compare these results at the point where FPR = 74%, our approach had already satisfied TPR = 100%, which is higher than SVM-PCT's 60%. Thus, even where the same number of positives was achieved by other approaches, our anomalous score measure always delivered the highest detection rate.

Time Cost: This set of experiments compare the time costs between BuffDetect and the benchmark methods. We conducted the experiments on synthetic datasets, in which the normal instances were generated based on a GLM that models mixed-type attributes, and the anomalous instances were generated by random shifting. Table 3 shows the time cost comparison among the methods for datasets with different instance sizes. Although it suffers from a higher time cost than the benchmark methods, our approach delivers high detection accuracy in a reasonable time. We also demonstrated the effectiveness of our optimization. The non-optimized framework (Naive-BD) failed in processing the dataset of size 1000, whilst the optimized framework outperformed 3 of the benchmarks.

Table 3: Time Cost Comparison (Seconds)

Method \ Size	100	300	500	1000	10000	100000
BuffDetect	1.119	10.79	19.68	98.21	129.76	157.73
LOADED	0.024	0.071	0.118	0.235	2.38	23.203
RELOADED	0.081	0.102	0.159	0.193	0.543	6.9650
KNN-CT	0.004	0.010	0.024	0.074	6.19	322.137
LOF-CECT	0.003	0.010	0.032	0.131	N/A	N/A
SVM-PCT	0.009	0.017	0.038	0.139	N/A	N/A
Naive-BD	187.63	2541.4	9112.5	N/A	N/A	N/A

Conclusions

In this paper, we have proposed a statistical-based framework for general purpose anomaly detection on mixed-type data. The framework incorporates a novel statistical model for capturing abnormal behaviors based on improving the INLA method to approximate Bayesian inference. The empirical results demonstrated the good performance of our approach for anomaly detection in mixed-type data.

References

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. Lof: identifying density-based local outliers. *SIGMOD Rec.* 29(2):93–104.

Cerioli, A. 2009. Multivariate outlier detection with high-breakdown estimators. *Journal of the American Statistical Association* 105(489):147–156.

Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. *ACM Comput. Surv.*

Das, K. 2007. Detecting anomalous records in categorical datasets. In *KDD 07'*, 220–229.

Frank, A., and Asuncion, A. 2010. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/.

Gentle, J. 2007. *Solutions that Minimize Other Norms of the Residuals*. New York: Springer.

Ghoting, A.; Otey, M. E.; Parthasarathy, S.; and Ohio, T. 2004. Loaded: Link-based outlier and anomaly detection in evolving data sets. In *Proceedings of the 4th IEEE ICDM*, 387–390.

Otey, M. E.; Parthasarathy, S.; and Ghoting, A. 2006. Fast lightweight outlier detection in mixed-attribute data sets. *DMKD*.

Piatetsky-Shapiro, G.; Djeraba, C.; Getoor, L.; Grossman, R.; Feldman, R.; and Zaki, M. 2006. What are the grand challenges for data mining?: Kdd-2006 panel report. *SIGKDD Explor. Newsl.* 8(2):70–77.

Press, W.; Teukolsky, S.; Vetterling, W.; and Flannery, B. 2007. *Section 7.9.1 Importance Sampling*. New York: Cambridge University Press.

Ramaswamy, S.; Rastogi, R.; and Shim, K. 2000. Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.* 29(2):427–438.

Riani, M.; Atkinson, A. C.; and Cerioli, A. 2009. Finding an unknown number of multivariate outliers. *Journal of the Royal Stats Society Series B* 71(2):447–466.

Rue, H.; Martino, S.; and Chopin, N. 2009. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the Royal Stats Society Series B* 71(2):319–392.

Steuer, R. E.; Kurths, J.; Daub, C. O.; Weise, J.; and Selbig, J. 2002. The mutual information: Detecting and evaluating dependencies between variables. In *ECCB*, 231–240.

Tyler, D. E. 2008. Robust statistics: Theory and methods. *Journal of the American Statistical Association* 103:888–889.

Yang, Q., and Wu, X. 2006. 10 challenging problems in data mining research. *IJITDM* 5(04):597–604.