Anomaly Detection in Streaming data from Air Quality Monitoring System

Author: Cong YUE
Supervisor: Dr. Masud MOSHTAGHI

A thesis submitted in fulfilment of the requirements for the degree of Master of Information Technology in the Department of Computing and Information System

October 29, 2015
Declaration of Authorship

I, Cong YUE, declare that this thesis titled, “Anomaly Detection in Streaming data from Air Quality Monitoring System” and the work presented in it are my own. I confirm that:

- This thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.

- Where necessary I have received clearance for this research from the University’s Ethics Committee and have submitted all required data to the Department

- The thesis is 7715 words in length (excluding text in images, table, bibliographies and appendices)

Signed:                                                                                                         

Date:
“In the cave we fear lies the treasure we seek.”

Joseph Campbell
Detection of abnormalities is an important aspect of air quality monitoring. Wireless Sensor Networks (WSNs) provide a flexible and low-cost solution for air quality monitoring. However, considering the limited resources available in these networks in terms of power, memory and computational resources, obtaining a high anomaly detection rate while prolonging the life span of these networks is a challenging task. In recent years, both parametric and non-parametric algorithms are put forward to tackle this challenge. In order to save energy and memory, researchers have been investigating the iterative detection algorithms. In this thesis, we proposed a new efficient parametric iterative algorithm, in which the cumulative sum of likelihood ratio is calculated then we compare the cumulative sum with a manually defined control limit. We also evaluate effectiveness of our proposed algorithms both on synthetic data and real sensor data and compare it with a recently proposed algorithm. In evaluation on synthetic data, we design different experimental cases with respect to real environment and point out principles in selection of the two algorithms in practise. In evaluation on real data, we analyse and discuss the result and compare the effectiveness and efficiency of the two algorithms.
Acknowledgements

It has not been too long but a deeply fulfilling journey. A journey makes my study so special in the University of Melbourne. A journey let me get acquainted with so many wonderful people, without whose support I would not acquire the passion and knowledge I enjoy.

I would like to firstly say thank you to my superior Dr. Masud Moshtaghi. For his guidance and contributions in helping me achieve substances and methodologies from what started as a superficial idea in my head and his considerable patience in answering me so much questions.

I am also want to express my thanks to Prof. James Bailey who gave me guidance at the start of the research project.

Personally, I feel indebted to my parents. As their son, I am so regretful that I cannot return them much. I feel grateful to their willingness love.
# Contents

Declaration of Authorship i  
Abstract iii  
Acknowledgements iv  

1 Introduction 1  
   1.1 A statement of the problem 1  
   1.2 Motivation of the research 4  
   1.3 Contributions 4  
   1.4 Description of the remaining chapters 5  

2 Literature Review 6  
   2.1 Nature of environmental monitoring 6  
   2.2 Input and output 6  
   2.3 Anomaly detection techniques 7  
      2.3.1 Non-parametric approaches 8  
      2.3.2 Parametric approaches 10  

3 Problem Statement 12  
   3.1 General description of the problem 12  
   3.2 Definition of terms of the problem 12  
      3.2.1 Definitions of terms in IMCUSUM 13  
      3.2.2 Definitions of terms in FFIDCAD 13  

4 Methodology 15  
   4.1 Control chart based algorithm - IMCUSUM 15  
   4.2 Ellipsoid based algorithm - FFIDCAD 17  

5 Implementation of algorithm FFIDCAD and IMCUSUM 19  

6 Results 21  
   6.1 Testing on Synthetic data 21  

v
6.1.1 Evaluation on synthetic data with different noise levels and magnitudes ........................................... 22
6.1.2 Evaluation on synthetic data with different data change speed .................................................. 23
6.2 Testing on Air Quality Sensor data .................................................. 23
6.2.1 Correlation Analysis in real data ........................................ 24
6.2.2 Comparison of FFIDCAD and IMCUSUM on Real Air Quality Data ........................................... 25
6.2.3 Efficiency of FFIDCAD and IMCUSUM ................................. 26

7 Conclusions and Future work .................................................. 28
7.1 Conclusion ................................................................. 28
7.2 Future works ........................................................... 29

Bibliography ................................................................. 30
List of Figures

1.1 WSN-Example .................................................. 1
1.2 Anomaly-Example ........................................... 2

4.1 Example of ellipses in FFIDCAD .......................... 17

6.1 ROC curves on synthetic data with slow change speed .... 23
6.2 ROC curves on synthetic data with fast change speed ....... 24
6.3 Run Chart of No2 and O3 with labelled anomalies ........... 25
6.4 Run Chart of CO and CO2 with labelled anomalies ......... 25
6.5 ROC curves of real dataset ................................... 26
6.6 Comparison on efficiency of FFIDCAD and IMCUSUM ....... 26
List of Tables

1.1 Constraints of environmental monitoring in WSNs . . . . . . . . . 3
2.1 Parametric and Non-Parametric techniques in recent years . . . . 8
6.1 Detection rates and false alarm rates with different noise level and magnitude in FFIDCAD (p value=0.999) . . . . . . . . . . . . . 22
6.2 Detection rates and false alarm rates with different noise level and magnitude in IMCUSUM (δ = 2, L=0.2) . . . . . . . . . . . . . 22
6.3 Correlation Coefficient between printer jobs and gas attribute . . 25
List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFIDCAD</td>
<td>Forgetting Factor Iterative Data Capture Anomaly Detection</td>
</tr>
<tr>
<td>IMCUSUM</td>
<td>Iterative Multivariate Cumulative SUM</td>
</tr>
<tr>
<td>CUSUM</td>
<td>Cumulative SUM</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>MGDF</td>
<td>Multi Granularity Deviation Factor</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 A statement of the problem

Air quality monitoring is an important task in Environmental monitoring. It has been studied in the context of agriculture, industry, ecology and many other fields for a long time. Techniques used for environmental monitoring experienced a change from manual detection to automatic detection using Wireless Sensor Networks (WSNs) (Oliveira and Rodrigues, 2011).

WSNs are consist of many multi-functional tiny sensor devices that can react to environment change then automatically record and process data. An important data processing task in these networks is anomaly detection. An example of a sensor device and a WSN is displayed in Figure 1.1.

"Anomalies are patterns in data that do not confirm to a well-defined notion of normal pattern" (Chandola, Banerjee, and Kumar, 2009). Figure 1.2 is an example of anomalies in a 2 dimensional dataset. Red points in the figure

\[\text{Figures are modified based on libelium gases 2.0 guide:}\]
\[\text{http://www.libelium.com/uploads/2013/02/gases-sensor-board_2.0_eng.pdf}\]
are anomalies because they are far away from the majority of all data points. Anomalies are caused either by an event of interest in the environment or a fault in the system. If anomalies are related to events of interest, like network intrusion (Northcutt and Novak, 2002), they are usually the aim of the monitoring project. If anomalies are faulty noise (Oh, Hong, Choi, et al., 2012), we want to clean the data from them. Anomaly detection can be performed in Batch mode, in which all data points are collected and then anomalies are identified in them. (Rajasegarar et al., 2010; Moshtaghi et al., 2009). They can also be performed in Iterative mode, in which data are processed sequentially and anomalies are identified in each iteration (Tan, Ting, and Liu, 2011; Moshtaghi et al., 2011).

![Figure 1.2: Example of a anomaly in Temperature-Humidity Data set.](image)

Considering evaluation of anomaly detection algorithms, two criteria are important. One is detection rate which is a ratio of number of detected anomalies to number of all anomalies. It measures how many anomalies are detected; Another one is false alarm rate which is a ratio of number of mislabelled normal data to all normal data. It measures how many normal data are mislabelled as anomalies. In different application domains, to obtain a good detection rate, we have to tolerate some degree of false alarm rate. But generally, we prefer the techniques with low false alarm rate and high detection rate. While in WSNs, besides detection rate and false alarm rate, we care about how to minimise energy and storage consumption as well. The reason is as follows: In most conditions, sensor devices are distributed in unattended areas. In order to conveniently deploy them, compact central processing unit (CPU) and memory chips are integrated to reduce the size and price of sensor devices. The small
size of low price sensor devices determine the limitation of their energy source (battery) and memory and hence limit their lifetime and reliability. Accuracy is important, while a reliable sensor device with long lifetime is important as well. The main **Constraints** in environmental monitoring using WSNs are displayed in Table 1.1. Designing an integratable anomaly detection algorithm that satisfy these constraints is a challenge in WSNs.

Data in WSNs are observed sequentially, forming a time series. Considering the characteristics of data, iterative methods are appropriate for environmental monitoring. The main advantages of iterative methods compared to batch methods are as follows:

- Iterative methods require fewer data to update detection model. Iterative methods either use the latest data point or a window size of previous \( k \) data points, where \( k \) is a small constant. While batch methods usually need all historical data points to update detection model.

- Iterative methods are more efficient and energy-saving for model updating. Iterative methods usually require one latest data point to update model, so they are efficient hence save energy for computation in model updating. However, batch methods update detection model based on all observed data points, they need more time to update model. The long intensive calculation for model updating consumes more energy.

### Table 1.1: Constraints of environmental monitoring in WSNs

<table>
<thead>
<tr>
<th>Perspective</th>
<th>Constraint</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>No labels for normal and anomalous data</td>
<td>Only unsupervised techniques available</td>
</tr>
<tr>
<td></td>
<td>Multivariate in most cases</td>
<td>More complex than univariate</td>
</tr>
<tr>
<td></td>
<td>Streaming mode</td>
<td>Data points are autocorrelated in time series</td>
</tr>
<tr>
<td>Sensor</td>
<td>Sensors inside the device may have different response time (sensitivity) and lifetime</td>
<td>Faulty data may occur</td>
</tr>
<tr>
<td>Compact Structure</td>
<td>Small size of battery</td>
<td>Limitation of energy for computation and commun-</td>
</tr>
<tr>
<td></td>
<td>Small size of memory</td>
<td>ication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Limitation of data size for computation</td>
</tr>
</tbody>
</table>
Iterative methods have better performance. Models that update based on more data perform better. Iterative methods update the model after each observation and apply the model on the new data point. But batch methods update model less frequently, they have to trade off between model updating frequency and energy consumption. So a model that applied on a new data point may be built based on less data points compared to iterative methods.

Iterative methods are more adaptive. Iterative methods update detection model more frequently (usually once there is a new observation) than batch methods. So, they are more sensitive to the environment change.

We propose a iterative anomaly detection algorithm IMCUSUM, and compare our algorithm with another iterative algorithm - Forgetting Factor Iterative Data Capture Anomaly Detection (FFIDCAD) provided by Moshtaghi et al. (Moshtaghi et al., 2011). The comparison includes experiments to test effectiveness and efficiency of the two algorithms.

1.2 Motivation of the research

WSNs are a hot topic in both research and industry, especially under the notion of Internet of Things (IoT) (Evans, 2011). In applications of IoT, most of electrical equipments in our surroundings are integrated with sensors. At present, WSNs are deployed in many application domains and provide convenience due to their automatic operation manner. Research on anomaly detection techniques, as a indispensable part of monitoring applications, is important of the new development in IoT.

There are many known anomaly detection algorithms that work in batch mode (Silva et al., 2005; Rajasegarar et al., 2007; Rajasegarar et al., 2010; Moshtaghi et al., 2009). However, considering the characteristics of data in WSNs and advantages of iterative algorithms as we discussed, iterative methods are more appropriate for WSNs. So we explore a known iterative detection algorithm in details, and also we proposed a new control chart based iterative detection algorithm.

1.3 Contributions

We summarise here the main contributions of this thesis:
1) We proposed a iterative anomaly detection algorithm - IMCUSUM. Cumulative Sum (CUSUM) Chart is a change detection technique in quality control. But it relies on the known mean and variance of a normally distributed variable. Combining known multivariate CUSUM with iterative parameter updating methods, we get a new iterative anomaly detection algorithm - IMCUSUM.

2) We evaluate and compare the proposed technique with a recently proposed method. In chapter 6, we show that IMCUSUM can perform one data instance in around 0.25 ms and FFIDCAD can do that in less than 0.25 ms. We also evaluate their effectiveness. For example, on one synthetic data that generated in slow change speed with low noise level and magnitude, IMCUSUM has 55% detection rate and FFIDCAD has 35% detection rate with false alarm rate around 5%.

1.4 Description of the remaining chapters

The main chapters of this thesis are as follows:

- In Chapter 2, we introduce related works on environmental monitoring, input and output of anomaly detection and anomaly detection techniques both in general notion and in WSN.

- in Chapter 3, we describe the problem and define some formal mathematical terms of the problem that the later chapters will use.

- in Chapter 4, we elaborate the details of the FFIDCAD and IMCUSUM algorithms.

- in Chapter 5, we discuss the pipeline for implementation and provide the complete FFIDCAD and IMCUSUM algorithms.

- in Chapter 6, we provide the results of both algorithms on synthetic data and real sensor data. We also evaluate their effectiveness and efficiency.

- in Chapter 7, we conclude this thesis and describe open issues for future work.
Chapter 2

Literature Review

2.1 Nature of environmental monitoring

Environmental monitoring experienced a long history. Today, Wireless Sensor Networks (WSNs) make this task easier than before. They provide high density information from the monitoring area. This information can be used to detect abnormal sensor readings as alert, for example in earthquake prediction (Saradjian and Akhoondzadeh, 2011). Considering the nature of sensor nodes, low price devices with limited energy source and memory storage, it becomes a key challenge to effectively identify anomaly with minimised energy cost. With the knowledge that intercommunication between sensors consume much more energy than computation (Oliveira and Rodrigues, 2011; Rajasegarar, Leckie, and Palaniswami, 2008), a light-weight scheme that performs computation inside each sensor node can expand the lifetime of WSNs.

2.2 Input and output

Anomaly detection is a general technique that applied in many application domain like financial fraud detection (Srivastava et al., 2008), health-care (Wong et al., 2003) and cyber security (Northcutt and Novak, 2002). It has a long history and can be traced back to hundreds years ago (Edgeworth, 1887). When choosing an anomaly detection method, characteristics of input data need to be taken into consideration. For example, in WSNs, the data mainly forms a multivariate and temporally correlated sequence of observations.

In a survey paper (Chandola, Banerjee, and Kumar, 2009), the author categorised input data to point data, sequence data and spatial data in terms of relationship between each data instance.
1) *Point data* suggests data instance has no relationship between each other. Authors of (Fawcett and Provost, 1999) proposed a cell-phone fraud detection technique for point data;

2) *Sequence data* suggests data instances are temporally related. Lane et al. (Lane, 1999) and Yamanishi et al. (Yamanishi and Maruyama, 2005) use Hidden Markov Model to detect anomalies in system-call sequence and network failure respectively for sequence data;

3) *Spatial data* suggests data instances are geographically related. Hazel et al. (Hazel, 2000) use Random Markov Fields to identify anomalous region from images, where the input image is regarded as spatial data.

In terms of dimensions of input data, data instance can also be classified as *univariate* and *multivariate*. Take a WSN that monitors *Temperature* and *Humidity* as an example, if temperature or humidity are analysed separately, each of them is univariate as an input data instance; if they are analysed as a whole, the data instance is two-dimensional multivariate.

Generally, there are two kinds of output of anomaly detection techniques. The one is the *label* that explicitly identifies anomaly, the other is the *score* that measures to what extent a data instance is anomalous.

In this thesis, considering the environmental monitoring sensor generating data sequentially, we take *sequence data* as input. In air quality monitoring, multiple gases are measured by sensors in most cases and which resulting in multivariate data instances. Considering output, we use *label* to denote anomaly data and normal data.

### 2.3 Anomaly detection techniques

Researchers put forward many anomaly detection techniques. In general, these techniques can be categorised into three classes (Chandola, Banerjee, and Kumar, 2009; Rajasegarar, Leckie, and Palaniswami, 2008; Xie et al., 2011): *supervised detection*, *semi-supervised detection* and *unsupervised detection*. *Supervised detection* subjects to the assumption of prior knowledge known to the model, which means detection model need to be trained on training data that has known label (normal or anomalous). The classifier flags the test data as anomaly or normal based on trained model; *semi-supervised detection* requires normal
Chapter 2. Literature Review

<table>
<thead>
<tr>
<th>Category</th>
<th>Techniques</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>Bayesian Network</td>
<td>Hill, Minsker, and Amir, 2007; Silva et al., 2005;</td>
</tr>
<tr>
<td></td>
<td>Rule-based</td>
<td>Moshtaghi et al., 2009; Moshtaghi et al., 2011; Rajasegarar et al., 2009;</td>
</tr>
<tr>
<td></td>
<td>Elliptical boundary based</td>
<td>Moshtaghi et al., 2011; Rajasegarar et al., 2009; Ross, Tasoulis, and Adams, 2009;</td>
</tr>
<tr>
<td></td>
<td>CUSUM</td>
<td></td>
</tr>
<tr>
<td>Non-parametric</td>
<td>SVM</td>
<td>Rajasegarar et al., 2007; Rajasegarar et al., 2010;</td>
</tr>
<tr>
<td></td>
<td>Cluster based</td>
<td>Rajasegarar et al., 2006; Subramaniam et al., 2006;</td>
</tr>
<tr>
<td></td>
<td>Density based</td>
<td></td>
</tr>
</tbody>
</table>

or abnormal label are known for training data; unsupervised detection requires no labelling. Rajasegarar et al. (Rajasegarar, Leckie, and Palaniswami, 2008) categorised anomaly detection techniques into parametric approach and non-parametric approach. The parametric approach requires prior knowledge in form of an assumption about a specific underlying model of the data. Then it learns the parameters of the model from the observed data. The non-parametric approach is on the contrary, prior knowledge free. Table 2.1 briefly describes the two kinds of approaches and corresponding details and references.

### 2.3.1 Non-parametric approaches

Considering non-parametric model, authors (Rajasegarar et al., 2007; Rajasegarar et al., 2010) put forward SVM-based classifier to detect outliers in WSN. The proposed SVM classifier maps input data to a higher dimensional feature space and identifies anomaly by building a hypersphere in that feature space: data points lying outside the hypersphere are labeled as anomalies. The limitation of SVM-based technique is that using historical data to build hypersphere increases the computational complexity and need for large enough storage. Furthermore, the parameters that control the volume of hypersphere is hard to adjust in fast changing environment (Rajasegarar, Leckie, and Palaniswami, 2008).

Rajasegarar et al. (Rajasegarar et al., 2006) put forward a cluster-based algorithm, where they use Euclidean distance between data vectors to form clusters. The width $\omega$ of each cluster is fixed and manually set, the distance of any data...
vector exceeds $\omega$ will be labeled as anomaly. The complexity of this algorithm is $O(N^2c)$ where $N_c$ is the number of clusters and this algorithm also need $O(N_c)$ space to store clusters. This algorithm performs in distributed manner, therefore a large number of communications between sensors are saved. The authors claim they save 85% to 98% communication overhead, which is a significant reduction.

In another approach, the authors (Subramaniam et al., 2006) suggest a density-based approach, where they define “density” in terms of distance and local-metric respectively. Under the distance-based definition, a data point $p_{tr}$ will be identified as anomaly with respect to dataset $T$ if no more than $D$ points in $T$ lie within distance $r$ from $p$. Here, $D$ is determined by estimated density distribution. $r$ is a threshold. If different regions in the data space have different density, then a single threshold for all regions cannot offer good performance. Under this circumstance, the local-metric definition can be applied, where they defined Multi Granularity Deviation Factor(MGDF) to measure the count of neighbours. A data vector is flagged as anomaly if its MDEF is significantly different from the local average.

The advantage of non-parametric techniques is that they are more flexible to generalise in unfamiliar environment due to the prior knowledge free property. But they are less efficient, because parameters lies in the data distribution and grows as size of data grows, which means they learn parameters from the joint distribution, so in most approaches they have to consider all data points to get detection model, such as the SVM-based approach, cluster approach and density-based approach we discussed. In the view of iterative approaches, if non-parametric methods can learn distribution iteratively and dynamically adjust itself to fit different distributions, it can be more applicable and efficient. Another thing about non-parametric approaches is they have to trade off between energy consumption and the loss of detection accuracy (Rajasegarar, Leckie, and Palaniswami, 2008).
2.3.2 Parametric approaches

In the WSN, considering parametric approaches, authors (Hill, Minsker, and Amir, 2007) proposed a Dynamic Bayesian Network method to track wind-speed and corresponding measurements. They estimate the posterior distribution by the observed data and then use the learned posterior distribution to construct a Bayesian credible interval \( p \% \). The \( p \) is actually the confidence interval of posterior probability of observed data. Measurements that fall outside \( p \% \) are classified as anomaly. Like general probabilistic graph model, the network parameters are learned by Expectation Maximisation (EM) algorithm. This technique process streaming data very fast, but the problem of this technique is the expectation calculation phase among all sensor nodes in the network. This phase consumes considerable amount of energy. A rule-based approach is proposed by Silva et al. (Silva et al., 2005), where large number of rules are generated for network attack detection and this approach shows very high performance. But this technique depends much on prior knowledge set manually by domain experts and the performance is closely linked to the topology of the routing tree. This makes it hard to generalise to other WSN fields.

Authors from (Moshtaghi et al., 2009; Rajasegarar et al., 2009) designed a parametric statistical approach in which an ellipsoidal boundary is formed under the assumption that input data fit multivariate Gaussian distribution. Data instances outside the elliptical boundary are labeled as anomalies. This method is only suitable for specific application domain, because it requires all the historical data in each sensor node, which is a burden for computation and storage. Recently, the same authors put forward a new iterative elliptical boundary model - Forgetting Factor Iterative Data Capture Anomaly Detection (FFIDCAD) (Moshtaghi et al., 2011) that addresses the problem of offline training of the previous method.

Ross et al. (Ross, Tasoulis, and Adams, 2009) proposed a Parametric Cumulative Sum (CUSUM) methods to perform online anomaly detection, they use an auto-regressive model \( y = X^T \cdot \Theta + \epsilon \) to deal with streaming data and then use CUSUM to detect the change of \( \Theta \), if the change exceeds a threshold \( H \), the latest observed data can be labeled as anomaly. The drawback of CUSUM based algorithm is the threshold, the different value of threshold affects effectiveness.
Generally, in terms of detection rate and false alarm rate, parametric models are very effective if the prior knowledge is accurate. With the knowledge of the data distribution, parametric models perform anomaly detection efficiently. In WSNs, sensor data are mainly generated sequentially, it leaves room for optimisation in terms of computation and memory. Iterative parametric models estimate parameters and perform detection iteratively. They only require to store current data from observation and parameters from the previous one observation. Batch parametric models estimate parameters considering all observed data points, so they have to store all the historical data points to generate the detection model. Therefore, iterative models are normally more efficient than batch models and usually applied in real-time or online detection environment. In addition, like discussed, iterative model saves sensor energy and storage. Parametric models, no matter the iterative one or the batch one, have drawbacks - If the assumptions of prior knowledge is wrong, the result can be incorrect.

In this thesis, we focus on iterative parametric techniques because they are efficient due to limited parameters and iterative updating manner. These techniques hence save energy for computation and memory for data storage. As we discussed, this is a key challenge in WSN.
Chapter 3

Problem Statement

3.1 General description of the problem

Our objective is detecting anomaly data instances from a sequence of data instances. Each data instance is in multivariate form. In each iteration, we label the data instance anomalous or normal by using FFIDCAD and IMCUSUM algorithms.

In the following parts of this chapter, we give formal definitions of terms in FFIDCAD and IMCUSUM.

3.2 Definition of terms of the problem

Let $X_k = \{x_1, x_2, ..., x_k\}$ be the first $k$ samples in times $\{t_1, t_2, ..., t_k\}$. Our aim is to classify a new sample at time $t_{k+1}$ as normal or anomalous without keeping all the previous $k$ samples. Each sample in $X$ is a $d \times 1$ vector in $\mathbb{R}^d$. Each one of these $d$ elements signifies an attribute that sensor measures in environments. For example, the amount level of NO2 and O3 are attributes that sensor measures. Under the assumption of Multivariate Gaussian distribution, the density function of $x$ is defined in Eqn 3.1, The sample mean $m_k$ can be calculated by Eqn 3.2, sample covariance matrix $S_k$ can be calculated by Eqn 3.3. The iterative formulas of $m_k$, $S_k$ and inverse of $S_k$ can be calculated by Eqn 3.4, Eqn 3.5 and Eqn 3.6. Eqn 3.5 and Eqn 3.6 are derived from (Duda, Hart, and Stork, 2000) (p.65).

$$f(x) = (2\pi)^{-\frac{d}{2}}|\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right) \quad (3.1)$$

$$m_k = \frac{1}{k} \sum_{i=1}^{k} x_i \quad (3.2)$$
Chapter 3. Problem Statement

$$S_k = \frac{1}{k-1} \sum_{i=1}^{k} (x_i - m_k)(x_i - m_k)^T$$  \hspace{1cm} (3.3)

$$m_{k+1} = m_k + \frac{1}{k+1}(x_{k+1} - m_k)$$  \hspace{1cm} (3.4)

$$S_{k+1} = \frac{k-1}{k}S_k + \frac{1}{k+1}(x_{k+1} - m_k)(x_{k+1} - m_k)^T$$  \hspace{1cm} (3.5)

$$S_{k+1}^{-1} = \frac{kS_k^{-1}}{k-1}[I - \frac{x_{k+1} - m_k)(x_{k+1} - m_k)^T S_k^{-1}}{\frac{k}{k-1} + (x_{k+1} - m_k)^T S_k^{-1}(x_{k+1} - m_k)}$$  \hspace{1cm} (3.6)

### 3.2.1 Definitions of terms in IMCUSUM

In IMCUSUM, we measure the change of mean by log-likelihood ratio defined in Eqn 3.7

$$\log \frac{f_{H_0}(x)}{f_{H_1}(x)}$$  \hspace{1cm} (3.7)

Where $f(x)$ is the density function as defined in Eqn 3.1. $H_0$ is the null hypothesis where mean changes to a pre-defined scale that set by parameters, $H_1$ is the alternative hypothesis where mean stays same as in previous iteration. The bigger the ratio, the more the mean tend to shift.

**IMCUSUM Definition** - We define a data vector $x_k$ as anomalous by Eqn 3.8

$$C_k = max \{0, C_{k-1} + \log \frac{f_{H_0}(x_k)}{f_{H_1}(x_k)} \} > L$$  \hspace{1cm} (3.8)

Where $C_k$ is cumulative sum of $k$ samples, $L$ is the manually defined control limit.

### 3.2.2 Definitions of terms in FFIDCAD

In FFIDCAD, the hyper-ellipsoid is formed by Eqn 3.9

$$ell_k(m_k, S_k^{-1}; t) = \{x \in \mathbb{R}^d | (x - m_k)^T S_k^{-1}(x - m_k) \leq t^2 \}$$  \hspace{1cm} (3.9)
Where $t^2 = (\chi^2_{d})^{-1}$, the hyper-ellipsoid is actually the confidence space of the data distribution. $p$ is $p$ value as same defined in statistics. By assigning different $p$ values, this equation forms different confidence spaces. For example, if $p = 0.95$, the hyper-ellipsoid will cover 95% of the data.

**FFIDCAD Definition** - In FFIDCAD, they define a single data vector $x$ as anomalous by Eqn 3.10

\[
(x - m_k)^T S_k^{-1} (x - m_k) > t^2
\]  

(3.10)
Chapter 4

Methodology

4.1 Control chart based algorithm - IMCUSUM

In statistics, the likelihood ratio is used to test if the data distribution changes from null hypothesis to alternative hypothesis or vice versa. Under the definition in Eqn 3.7, the bigger the ratio, the more likely that the data distribution will change to null hypothesis. In this approach, we use log-likelihood ratio to measure the change of mean. If mean changes in large scale, we are believed the new data point locates far away from the average of all data points, therefore, it is more likely to be anomaly. Both null hypothesis, $f_{H_0}(x)$ and alternative hypothesis, $f_{H_1}(x)$ are likelihood functions, they are also the density function of Multivariate Gaussian distribution (Cheng and Thaga, 2005a). For first $k + 1$ samples, By apply Eqn 3.1, Eqn 3.2 and Eqn 3.3 to Eqn 3.7, we have Eqn 4.1

\[
\log \frac{f_{H_0}(x_{k+1})}{f_{H_1}(x_{k+1})} = \log \frac{(2\pi)^{-\frac{d}{2}} |S_k|^{-\frac{1}{2}} \exp(-\frac{1}{2}(x_{k+1} - m_{k+1}')^T S_k^{-1}(x_{k+1} - m_{k+1}'))}{(2\pi)^{-\frac{d}{2}} |S_k|^{-\frac{1}{2}} \exp(-\frac{1}{2}(x_{k+1} - m_k)^T S_k^{-1}(x_{k+1} - m_k))}
\]

\[
= \log \frac{\exp[-\frac{1}{2}(x_{k+1} - m_{k+1}')^T S_k^{-1}(x_{k+1} - m_{k+1}')] \exp[-\frac{1}{2}(x_{k+1} - m_k)^T S_k^{-1}(x_{k+1} - m_k)]}{\exp[-\frac{1}{2}(x_{k+1} - m_k)^T S_k^{-1}(x_{k+1} - m_k)]}
\]

(4.1)

In Eqn 4.1, $f_{H_0}(x_{k+1})$ indicates the probability of $x_{k+1}$ to be anomalous. In $f_{H_0}(x_{k+1})$, it assumes mean changes from $m_k$ to $m_{k+1}'$ that is a boundary value of $x_{k+1}$ to be anomalous. We define $m_{k+1}' = m_k + \delta \sigma_k$ where the $\sigma_k$ is a $d \times 1$ vector of standard deviations with respect to attributes in $x_k$, it can be derived from diagonal line of $S_k$. The $\delta$ is a manually defined coefficient that defines how many times of standard deviation the mean shifts to support the null hypothesis. $f_{H_1}(x_{k+1})$ indicates the probability of $x_{k+1}$ to be normal. In $f_{H_1}(x_{k+1})$,
the mean keep the same as in previous round, \( m_k \). The \( m_k \) can be calculated iteratively by Eqn 3.4, the \( S_k^{-1} \) can be calculated by Eqn 3.6.

The IMCUSUM model accumulates the the log-likelihood ratio in each iteration and decides whether a data vector is anomaly or not by Eqn 3.8. For the CUSUM test in statistics, authors from (Cheng and Thaga, 2005a) apply Eqn 3.1 to Eqn 3.7 and break it down into a simpler form. Here we use the same linear transformation method and break down Eqn 4.1 into Eqn 4.2

\[
\log \frac{f_{H_0}(x_{k+1})}{f_{H_1}(x_{k+1})} = (m'_{k+1} - m_k)^T S_{k}^{-1}(x_{k+1} - m_k) - \frac{1}{2} (m'_{k+1} - m_k)^T S_{k}^{-1}(m'_{k+1} - m_k)
\]

(4.2)

By applying Eqn 4.2 into Eqn 3.8 and standard scaling, we get the IMCUSUM detection model in Eqn 4.3.

\[
C_{k+1} = \max \{0, C_k + \alpha_{k+1}(x_{k+1} - m_k) - \frac{1}{2} \beta_{k+1}\} > L
\]

(4.3)

where \( k > 1 \), \( L > 0 \) and

\[
\beta_{k+1} = \sqrt{(m'_{k+1} - m_k)^T S_{k}^{-1}(m'_{k+1} - m_k)} = \sqrt{\sigma_{k}^T S_{k}^{-1} \sigma_{k}}
\]

\[
\alpha_{k+1} = \frac{(m'_{k+1} - m_k) S_{k}^{-1}}{\beta_{k+1}} = \frac{\sigma_{k} S_{k}^{-1}}{\sqrt{\sigma_{k}^T S_{k}^{-1} \sigma_{k}}}
\]

The \( m_k \) and \( S_k^{-1} \) are iteratively updated by following Eqn 3.4 and Eqn 3.6. Another thing about IMCUSUM is the initialisation. Because \( S_1 \) does not exist and if \( S_2 \) is singular, then there is no \( S_2^{-1} \). So, we define the first two data vectors normal, \( C_1 = 0, C_2 = 0 \), \( m_2 \) can be calculated by Eqn 3.2 and

\[
S_2^{-1} = \begin{pmatrix}
c_{1,1} & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & c_{d,d}
\end{pmatrix}
\]

where each \( c_{i,j} \) in \( S_2^{-1} \) is constant, this definition can address the deficient matrix inverse problem.
When choosing parameters, we usually set $\delta \in [1, 3]$ based on the “Three Sigma Rule” (Pukelsheim, 1994). Generally, reducing control limit $L$ increases false alarm rate as well as detection rate. When $L$ is fixed, reducing $\delta$ increases the likelihood ratio hence increases false alarm rate and detection rate. In practice, we can set $\delta = 2$ and differ $L$ to control false alarm rate.

### 4.2 Ellipsoid based algorithm - FFIDCAD

The FFIDCAD algorithm is actually a algorithm called Iterative Data Capture Anomaly Detection (IDCAD) by applying Forgetting Factor $\lambda$. Except for updating mechanism by using $\lambda$, FFIDCAD and IDCAD share the same detection model.

In FFIDCAD and IDCAD, under the assumption of Eqn 3.9 covering a majority of data. A ellipsoidal boundary as defined in Eqn 4.4 is formed in each iteration. A 2-dimensional example is displayed in Figure 4.1, where 3 boundaries with p-value=0.98 are formed in corresponding iterations. The boundary determines the data point of the same color anomaly or not. Like the figure shows, the red point in iteration $m$ is anomaly, the blue point in iteration $n$ is normal and the black point in iteration $q$ is anomaly.

$$\delta_k(m_k, S_k^{-1}; t) = \{ x \in \mathbb{R}^d | (x - m_k)^T S_k^{-1} (x - m_k) = t^2 \}$$  \hspace{1cm} (4.4)
Unlike IMCUSUM, the FFIDCAD algorithm does not directly use Eqn 3.4 and Eqn 3.6 to update mean and inverse of covariance matrix. It introduces the forgetting factor - $\lambda \in (0, 1)$ which apply a weight $\lambda$ on the previous $k$ samples. By applying $\lambda$ into Eqn 3.4 and Eqn 3.6, FFIDCAD updates mean and inverse of covariance matrix by Eqn 4.5 and Eqn 4.6 (Moshtaghi et al., 2011).

$$m_{k+1,\lambda} = \lambda m_{k,\lambda} + (1 - \lambda)x_{k+1}$$ (4.5)

$$S_{k+1,\lambda}^{-1} = \frac{kS_{k,\lambda}^{-1}}{\lambda(k-1)} \times [I - \frac{(x_{k+1} - m_{k,\lambda})(x_{k+1} - m_{k,\lambda})^T S_{k,\lambda}^{-1}}{\frac{k-1}{\lambda} + (x_{k+1} - m_{k,\lambda})^T S_{k,\lambda}^{-1} (x_{k+1} - m_{k,\lambda})}]$$ (4.6)

The reason to introduce $\lambda$ is: For very large $k$ in Eqn 3.4, $\frac{1}{k+1}(x_{k+1} - m_k)$ will be very close to zero, then $m_{k+1}$ will be approximately equal to $m_k$. Updating of mean will be meaningless. It is the same case of covariance matrix as shown in Eqn 3.5. The forgetting factor $\lambda$ addresses this problem by forming an effective window for previous $3\tau$ samples before sample $k + 1$, where $\tau = \frac{1}{1-\lambda}$. It only uses previous samples in the window for updating instead of all historical samples. Authors from (Moshtaghi et al., 2011) suggests $\lambda$ to be a constant that close to 1 to stabilise the algorithm.

When choosing parameters, if p value is more close to 1, the false alarm rate will be smaller. In practise, we can control false alarm rate by different p-value.
Chapter 5

Implementation of algorithm
FFIDCAD and IMCUSUM

The input of both algorithms contains a data sequence $ds$, each element in $ds$ is a $d \times 1$ data vector. Besides $ds$, the FFIDCAD algorithm adds a manually defined $p-value \in (0, 1)$ as a input, the IMCUSUM algorithm adds manually defined coefficient $\delta$ and a control limit $L$ as inputs. The output of both algorithms are binary labels that denotes data vectors anomalous or not. In addition, we set the forgetting factor $\lambda = 0.99$ in the implementation of FFIDCAD. We implement both algorithms following 3 steps:

1) **Initialization** - Initialise mean and inverse of covariance matrix from the second sample, $m_2 = \frac{x_1 + x_2}{2}$, $S_2^{-1} = c \times I$ where $c$ is a constant and $I$ is $d \times d$ identity matrix.

2) **Detection** - Calculate detection model and label the new coming data vector.

3) **Updating** - Update $m_k$ and $S_k^{-1}$ in iteration $k$ where $k > 2$.

The implementation details of FFIDCAD and IMCUSUM are shown in Algorithm 1 and 2 respectively. From line 5-14 in Algorithm 1 and line 6-16 in Algorithm 2, we can see both algorithms finish in one traverse to all data points and perform detection in each iteration, so the complexity of both algorithms is $O(n)$. 
Chapter 5. Implementation of algorithm FFIDCAD and IMCUSUM

20

input: data sequence \(X = \{x_1, x_2, \ldots, x_N\}\), p value \(p\)
output: label of the sample, anomalous or normal

ALGORITHM FFIDCAD

\[
\begin{align*}
// & \text{ } m_i \text{ and } S_i \text{ are the mean and covariance matrix for first } i \text{ samples} \\
\text{Label } x_1, x_2 \text{ as normal} \\
m_2 & \leftarrow \frac{x_1 + x_2}{2} \\
S_2^{-1} & \leftarrow 100 \times I \quad // I \text{ is identity matrix} \\
\text{for } i \leftarrow 3 \text{ to } N \text{ do} \\
& \text{if } (x_i - m_{i-1})^T S_{i-1}^{-1} (x_i - m_{i-1}) > (\chi^2_d)^{-1} \\
& \quad \text{Label } x_i \text{ as anomalous} \\
& \text{else} \\
& \quad \text{Label } x_i \text{ as normal} \\
& m_i \leftarrow \lambda m_{i-1} + (1 - \lambda)x_i \quad // \lambda = 0.99 \text{ is a constant} \\
S_i^{-1} & \leftarrow \frac{(i-1)S_{i-1}^{-1}}{i\lambda} \times \left[ I - \frac{(x_i - m_{i-1})(x_i - m_{i-1})^T S_{i-1}^{-1}}{\frac{\sigma^T_i S_{i-1}^{-1} \sigma_i}{\sigma^T_i S_{i-1}^{-1} \sigma_i}} \right] \\
\text{end}\end{align*}
\]

Algorithm 1: Algorithm FFIDCAD

ALGORITHM IMCUSUM

\[
\begin{align*}
\text{Label } x_1, x_2 \text{ as normal} \\
m_2 & \leftarrow \frac{x_1 + x_2}{2} \\
S_2^{-1} & \leftarrow 100 \times I \quad // I \text{ is identity matrix} \\
C_1, C_2 & \leftarrow 0 \\
\text{for } i \leftarrow 3 \text{ to } N \text{ do} \\
& \text{if } \max(0, C_{i-1} + \frac{\sigma_{i-1} S_{i-1}^{-1}}{\sqrt{\sigma^T_{i-1} S_{i-1}^{-1} \sigma_{i-1}}} (x_i - m_{i-1}) - 0.5\delta \sqrt{\sigma^T_{i-1} S_{i-1}^{-1} \sigma_{i-1}}) > L \\
& \quad \text{Label } x_i \text{ as anomalous} \\
& \text{else} \\
& \quad \text{Label } x_i \text{ as normal} \\
& m_i \leftarrow m_{i-1} + \frac{1}{i} (x_i - m_{i-1}) \\
S_i^{-1} & \leftarrow \frac{(i-1)S_{i-1}^{-1}}{i^2} \times \left[ I - \frac{(x_i - m_{i-1})(x_i - m_{i-1})^T S_{i-1}^{-1}}{\frac{\sigma^T_i S_{i-1}^{-1} \sigma_i}{\sigma^T_i S_{i-1}^{-1} \sigma_i}} \right] \\
& \sigma_i \text{ can be derived from } S_i \text{'s diagonal line.} \\
\text{end}\end{align*}
\]

Algorithm 2: Algorithm IMCUSUM
Chapter 6

Results

6.1 Testing on Synthetic data

We generate 1000 sample data points following 2 variate Gaussian distribution. Each data point is a 2 dimensional vector. In order to simulate the characteristics of sensor data, we produce the dataset sequentially. Let us say the dataset \( X = \{ x_1, x_2, \cdots, x_i, \cdots, x_n \} \) maps times \( \{ t_1, t_2, \cdots, t_i, \cdots, t_n \} \). Each \( x_i \) in \( X \) is produced by manually defined mean \( m_i \) and covariance matrix \( S_i \). To get the sample \( x_{i+1} \), we manually drift mean \( m_i \) and covariance matrix \( S_i \) with a specific scale into \( m_{i+1} \) and \( S_{i+1} \). Here we set the initial mean and covariance matrix like the below. We will talk about the drift scale of them later.

\[
\mu_1 = (5, 5)^T \quad S_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}
\]

We evaluate IMCUSUM and FFIDCAD on two perspectives, when we evaluate one perspective, we set the other fixed. One perspective is **Change Speed** - the drift scale of mean and covariance matrix. The other one is the **Noise level and magnitude**. We list the two perspectives and describe in details:

1. **Change speed** - In practise, sensors may generate data by different time intervals, or the environment may change fast or slow resulting in sensors’ readings change fast or slow. We manually set the drift scale of mean to be fast or slow to simulate these conditions. We define slow drift of mean is \((0.1, 0.1)^T\), the fast drift of mean is \((0.5, 0.5)^T\). So we produce two datasets, *DS-slow* and *DS-fast*. *DS-slow* denotes slow drift of mean and *DS-fast* denotes fast drift of mean. We set covariance matrix \( S \) randomly drift between 0 to 1. The following is an example of possible drifts in *DS-slow*.

\[
m_1 = (5, 5)^T \quad \rightarrow \quad m_2 = (5.1, 5.1)^T
\]
2. Noise level and magnitude - Two characteristics of anomaly lies in the data sequence. They can affect the effectiveness of anomaly detection algorithms. One is noise level, the ratio of number of anomalies to the number of all data points. The other is noise magnitude which suggests how far away anomalies are from the mean, we usually use $\sigma$, the standard deviation as unit. We perform tests on DS-slow and label anomalies based on different combinations of noise level and magnitude.

6.1.1 Evaluation on synthetic data with different noise levels and magnitudes

We vary the noise level of 2 values, 1% indicates the dataset with small size of anomalies and 10% indicates the dataset with big size of anomalies. We choose noise magnitude in variations of 3 values, $\sigma$, $2\sigma$ and $3\sigma$. Under the fact of “Three sigma rule” in Gaussian distribution (Pukelsheim, 1994), a labelled point has the probability of 68%, 95% or 99.7% to be shifted outside the coverage of Gaussian distribution regarding $\sigma$, $2\sigma$ or $3\sigma$ shift scale. We select parameters that perform best and display the result of FFDICAD on table 6.1 and IMCUSUM on table 6.2.

<table>
<thead>
<tr>
<th>Table 6.1: Detection rates and false alarm rates with different noise level and magnitude in FFDICAD (p value=0.999)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Detection Rate</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>10%</td>
</tr>
<tr>
<td><strong>False Alarm Rate</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>10%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6.2: Detection rates and false alarm rates with different noise level and magnitude in IMCUSUM ($\delta = 2$, L=0.2)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Detection Rate</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>10%</td>
</tr>
<tr>
<td><strong>False Alarm Rate</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>10%</td>
</tr>
</tbody>
</table>
From table 6.1 and table 6.2, we can see: 1) For both methods, the detection rate increases when noise magnitude increases; 2) For both methods, the false alarm rate in dataset with 10% noise level is smaller than that in dataset with 1% noise level; 3) On the same dataset, IMCUSUM has bigger detection rate, while FFIDCAD has smaller false alarm rate.

In some application domain, we know noise level and magnitude or we can get them through historical labelled data. When choosing from these two algorithms, we can take these 4 tables as references.

### 6.1.2 Evaluation on synthetic data with different data change speed

In this section, we set 10% noise level and $3\sigma$ magnitude on DS-slow and DS-fast. By applying different parameters, $p$ value for FFIDCAD and $\delta$, $L$ for IMCUSUM, we plot the ROC curves. Figure 6.1 shows the curves in DS-slow. Figure 6.2 displays the curves in DS-fast.

![ROC curves on DS-slow](image)

We can see in DS-slow, on the same false alarm rate, IMCUSUM has bigger detection rate than FFIDCAD. The two algorithms operate similarly on DS-fast.

### 6.2 Testing on Air Quality Sensor data

We intend to monitor the gas emissions from printer job in an office room (about 10$m^2$). Our interesting gases are NO2, CO, CO2 and O3. We logged
all printer jobs and their execution time in 2 weeks, meanwhile we logged sensors’ readings during the same time period. Our objective is to detect anomalies caused by big printer jobs. Let us say, \( Y = \{y_1, y_2, \cdots, y_i, \cdots, y_n\} \) on times \( \{t_1, t_2, \cdots, t_i, \cdots, t_n\} \) is the sequential data of page numbers that jobs printed; \( X = \{x_1, x_2, \cdots, x_i, \cdots, x_n\} \) on times \( \{t_1, t_2, \cdots, t_i, \cdots, t_n\} \) is the sequential data of sensors’ readings, \( y_i \) is a scalar that indicates the number of pages printed between time \( t_{i-1} \) and \( t_i \). \( x_i \) is a 4 dimensional vector recording values of readings.

We sorted \( Y \) and define all \( y_i > 50 \) as anomalies, under this definition, some facts are gathered:

- There are 1339 data points, \( n = 1339 \)
- Non zero printer jobs constitute 7.8\% of all data points: \( \frac{|\{y_i > 0\}|}{n} = 7.8\% \)
- Anomalies constitute 34.6\% of all non-zero printer jobs: \( \frac{|\{y_i > 50\}|}{|\{y_i > 0\}|} = 34.6\% \)
- Anomalies constitute 27\% of all data points: \( \frac{|\{y_i > 50\}|}{n} = 27\% \)

### 6.2.1 Correlation Analysis in real data

We firstly analyse the correlation between \( Y \) (printer jobs) and 4 attributes (NO2, O3, CO, CO2) in \( X \). The reason for correlation analysis is that we want to test if gas emissions of printer jobs will impact the amount of corresponding gases in the air. Then we can focus on the principle gas that dominated by printer jobs. Table 6.3 shows the correlation coefficient between \( Y \) and \( X \).
<table>
<thead>
<tr>
<th>Printer Job</th>
<th>Gases</th>
<th>NO2</th>
<th>O3</th>
<th>CO</th>
<th>CO2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation Coefficient</td>
<td>-0.0419</td>
<td>-0.0104</td>
<td>-0.0443</td>
<td>-0.0976</td>
<td></td>
</tr>
</tbody>
</table>

Printer jobs shows weak correlation with monitored gases, the reasons are: 1) The office room is a public area, air exchange may affect the result. 2) The number of printer jobs is too small compared to all data points, even they have linear correlation, it is hard to reveal. We plot each attribute (gas) with labelled points in Figure 6.3 and Figure 6.4.

![Figure 6.3: Run Chart of NO2 and O3 with labelled anomalies](image)

![Figure 6.4: Run Chart of CO and CO2 with labelled anomalies](image)

6.2.2 Comparison of FFIDCAD and IMCUSUM on Real Air Quality Data

By applying different parameters of FFIDCAD and IMCUSUM on real dataset. We get the ROC curves in Figure 6.5.
Chapter 6. Results

Generally, under a tolerable false alarm rate below 10%, effectivenesses of both methods are not good. The reason is that no gas or combinations of gases we monitored has strong relationship with big printer jobs as shown in Table 6.3. Specifically, notice the cross point around (35%, 8%), with false alarm rate (False positive rate) above 8%, FFIDCAD is more effective, while with that below 8%, IMCUSUM is more effective.

6.2.3 Efficiency of FFIDCAD and IMCUSUM

We compare the two algorithms by a series of dataset with linear addition on size. We implement FFIDCAD algorithm 1 and IMCUSUM algorithm 2 on Python 2.7.1 and execute them on a Linux platform with a 2-core 2.0 GHz CPU
and 8 G memory. The test results are shown in Figure 6.6, where we can see both algorithms can perform detection for a sample in less than 0.25 ms and they all show linear time cost. In theory, from Algorithm 1 and 2, FFIDCAD and IMCUSUM have $O(n)$ time complexity. Comparing line 7 in Algorithm 2 with line 6 in Algorithm 1, we find IMCUSUM has constant numbers of more calculations than FFIDCAD, it is the reason that FFIDCAD is faster than IMCUSUM in Figure 6.6.
Chapter 7

Conclusions and Future work

7.1 Conclusion

In this thesis, we proposed a iterative parametric algorithm - IMCUSUM to address the challenges in WSNs. This algorithm is based on two algorithms: Multivariate CUSUM chart (Cheng and Thaga, 2005b) and another iterative algorithm FFIDCAD (Moshtaghi et al., 2011). We also compared our algorithm with FFIDCAD and provided formal pipelines to implement the two algorithms.

We elaborated the derivation of IMCUSUM and FFIDCAD and then provided implementation. Then, we designed experiments both on synthetic data and real data to test their efficiency and effectiveness. Our proposed method outperform FFIDCAD in terms of accuracy while having a slightly higher processing time.

For efficiency, both algorithms follows linear complexity in theory and can process one input sample in less than 0.25 ms.

Our evaluation results in synthetic datasets showed that IMCUSUM performs better in slow changing environment and the two algorithms performs similarly in fast changing environment; In real dataset, with false alarm rate bigger than 8%, FFIDCAD has higher detection rate than IMCUSUM. With false alarm rate smaller than 8%, IMCUSUM has higher detection rate.
7.2 Future works

We propose IMCUSUM under the assumption of the data following the Gaussian distribution. Actually, by using different density function in likelihood ratio, IMCUSUM can follow other distributions. In the next stage, we will study if IMCUSUM can learn the joint distribution from data points by in iterative fashion.
Bibliography


Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Cong, Yue

Title:
Anomaly detection in streaming data from air quality monitoring system

Date:
2015

Persistent Link:
http://hdl.handle.net/11343/92233

File Description:
Anomaly detection in streaming data from air quality monitoring system