UCLA

UCLA Electronic Theses and Dissertations

Title

Semi-Supervised Outlier Detection Algorithms

Permalink

https://escholarship.org/uc/item/1f03f6hb

Author

Tun, Jason Sopheap

Publication Date

2018

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA

Los Angeles

Semi-Supervised Outlier Detection Algorithms

A thesis submitted in partial satisfaction

of the requirements for the degree Master of Science

in Statistics

by

Jason Sopheap Tun

2018

© Copyright by

Jason Sopheap Tun

2018

ABSTRACT OF THE THESIS

Semi-Supervised Outlier Detection Algorithms

by

Jason Sopheap Tun
Master of Science in Statistics
University of California, Los Angeles, 2018
Professor Yingnian Wu, Chair

Abstract: In this paper, I compared 6 semi-supervised point outlier detection algorithms: LOF, robust PCA, autoencoder, SOM, one-class SVM and isolation forest. In all experiments, I training the models on only normal data points. Then, I use the models to detect the outliers in the testing data sets basing on the fact that if a point is not a normal point, the point is an outlier. I do the experiment on both generated data and real data. I found each of the 6 algorithms has both advantages and disadvantages. I have described them in details and give some suggested solutions to the weak points.

The thesis of Jason Sopheap Tun is approved.

Mark Stephen Handcock

Nicolas Christou

Yingnian Wu, Committee Chair

University of California, Los Angeles

2018

I would like to thank my thesis committee members

for all of their guidance through this process;

your discussion, ideas, and feedback

have been absolutely invaluable.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Outliers

Outliers are patterns in data that do not conform to a well defined notion of normal or inlier behavior [1]. Outliers are called in different ways such as anomalies, novelties, exceptions, peculiarities, surprises, intrusions, frauds, etc.

1.2 Point Outliers

There are different types of outliers: point, contextual and structural. However, for this paper, I only focus on point outliers. If an individual data instance can be considered as an outlier with respect to the rest of data, then the instance is termed as a point outlier [1].

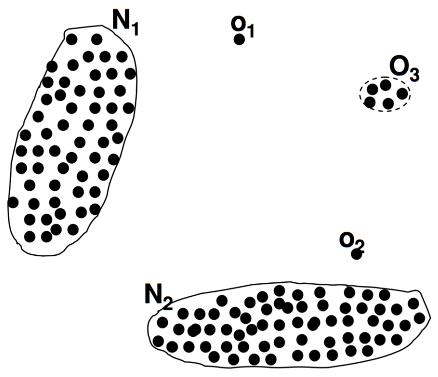

Figure 1.1: Example of Point Outliers

1.3 Semi-supervised Algorithm

There are 3 types of outlier detection techniques:

- Unsupervised detection: This is learning without knowing any labels. This type of algorithms assumes that the majority of the data set is normal data points and the data points that are different from the majority are outliers. In this learning, there is no training set and there is only unlabeled testing set.

- Supervised detection: This is learning with labels. In the training set, there are observations with inlier labels and outlier labels. This learning is the same as classification learning.

- Semi-supervised detection: In this learning, there are labels for both inliers and outliers. However, in the training set, there are only inlier data points. Unsupervised models are created for the training set. Then, the models are used to determine which data points in testing set are inliers and which are outliers basing on the fact that outliers are the observations that are not inliers.

Each type of detections has its own applications. When the data has no labels, only unsupervised detection can be used. When there are labels in the data, it is sometimes a good idea to use both inlier labels and outlier labels in order to maximize the prediction performance since both information from inlier labels and outlier labels are useful. However, semi-supervised detection is better than supervised detection in several situations. First, data does not give enough informations about the structure of outliers. This usually happens when the data is very unbalance and there are not enough outlier labels or there are too many different subgroups of outliers. Second, generators of outliers tend to generate outliers that are different from the previous outliers in order to avoid being caught; figure 1.2 is an example . This makes the structure of the outliers in the previous observations not only is useless but also leads to worse prediction performance. In these cases, it is better to use semi-supervised detection techniques to check which data points are not likely to be normal data points and should be considered as outliers.

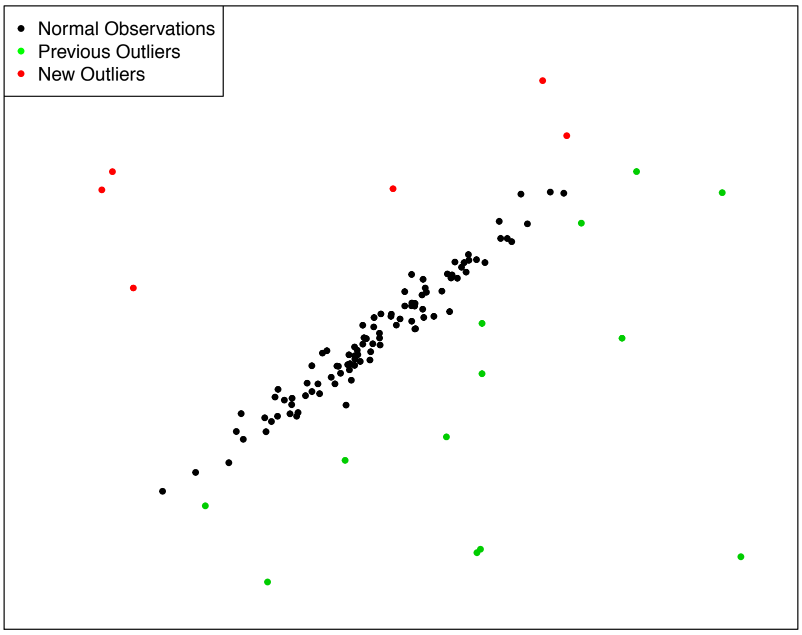

Figure 1.2: Previous outliers are completely different from new outliers.

In the above figure, if supervised techniques are used, the boundary decision will be between the black points and the green points. Basing on the boundary decision, the red points will be considered as normal points. The information from previous outliers leads to completely wrong predictions for the new outliers. However, if semi-supervised techniques are used the new outlier points might be all predicted as outliers since the deviations of red points from the normal points are quite big.

CHAPTER 2

Performance Measure

Accuracy is not a sufficient metric for evaluating outlier detection algorithms because most of the time, one false negative is way much more expensive than one false positive. In cancer detection, one false negative is one life. Thus, if there are 0.05% percent outliers, it is not a good idea to predict all normal to get 99.95% accuracy. Two common metrics for evaluating outlier detection algorithms are F-measure and AUC.

2.1 F-measure(F_β)

		Prediction outcome		
		n	**p**	**total**
Actual value	**n′**	True Negative	False Positive	N′
	p′	False Negative	True Positive	P′
	total	N	P	

To begin with,

$$\text{Precision}(P) = \Pr(\text{truely positive}|\text{predicted positive}) = \frac{TP}{TP+FP}$$

$$\text{Recall}(R) = \Pr(\text{predicted positive}|\text{truely positive}) = \frac{TP}{TP+FN}$$

Then,

$$\text{F-measure}(F_\beta) = \frac{(1+\beta^2)\cdot R \cdot P}{\beta^2 \cdot P + R}$$

where β is how important Recall(R) is and $\beta \in [0, +\infty]$

$F_\beta = (1+\beta^2)(\frac{1}{p}+\frac{\beta^2}{R})^{-1} = \frac{(1+\beta^2)TP}{(1+\beta^2)TP+\beta^2 \cdot FN+FP}$, which implies that TN is not important at all. On the other hand, accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$ depends on TN and when TN is extremely big comparing with the other values the accuracy will be almost 1. In outlier detection, TN usually is too big or not important; thus, F_β is a better measurement than accuracy.

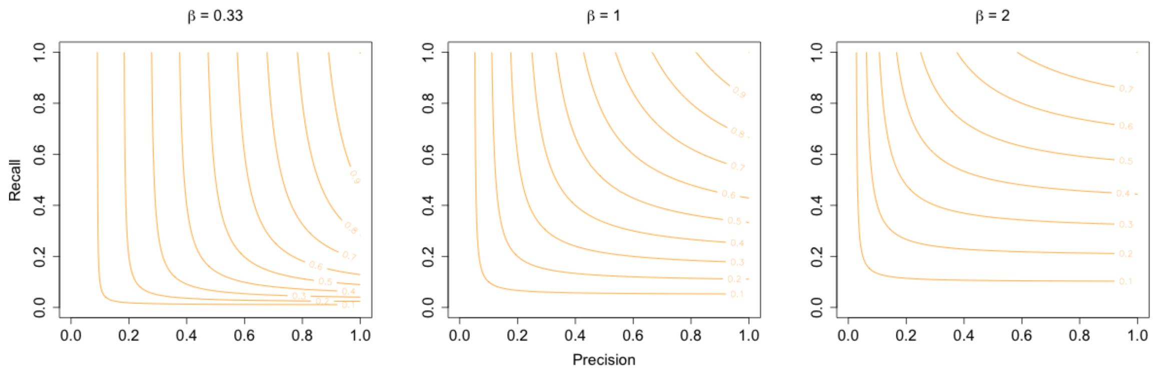

Figure 2.1: Contour Plots of F-measure with Different β Values

When precision or recall is almost zero, F_β is very low because of the imbalance as shown in figure 2.1. Beside that, small β means that precision has more influence on the F_β while large β means that recall has more influence on the F_β. $\beta = 1$ means that precision and recall are equally important and F_1 is symmetry with the reflection line of $R = P$.

2.2 Area Under ROC Curve(AUC)

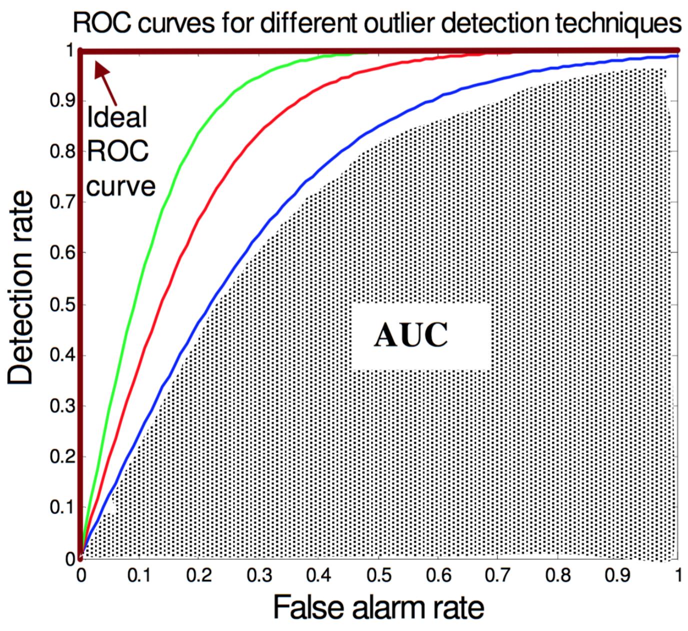

Figure 2.2: Illustration of ROC and AUC

Receiver Operating Characteristic curve (ROC) is the curve that shows the trade of between true positive rate and false positive rate.

True Positive Rate(TPR) = Recall(R) = Pr(predicted positive|truely positive) = $\frac{TP}{TP+FN}$

False Positive Rate(FPR) = Pr(predicted positive|truely negative) = $\frac{FP}{TN+FP}$

Then the formula for the Area Under Curve (AUC) is

$$\text{AUC} = \int_0^1 TPR(FPR)d(FPR) = E_{FPR\sim unif(0,1)}[TPR(FPR)]$$

AUC is the mean of TPR if FPR is set to be uniform between 0 and 1 [2]. In other words, AUC is the overall value of TPR over different FPR.

If it is the case that FPR needs to be between 0 and 0.2 only, we can use $\text{AUC}_{0-0.2}$ between 0 and 0.2 as an evaluation metric and we still have a similar property that $\text{AUC}_{0-0.2} = \int_0^{0.2} TPR(FPR)d(FPR) = 0.2E_{FPR\sim unif(0,0.2)}[TPR(FPR)]$.

CHAPTER 3

Methods

3.1 Local Outlier Factor

Local Outlier Factor(LOF) algorithm was introduced by [3]. This algorithm detects outlier basing on local density. If a data point has very small density comparing to its neighbors', the data point is considered as an outlier. In order to understand LOF algorithm, we need to know the terms: $k - distance(p)$, $k - distance$ neighborhood of p ($N_{k-distance(p)}(p)$), reachability distance of an object p w.r.t observation o ($reach - dist_k(p, o)$), and local reachability density of an object p ($lrd_k(p)$). Thus, I will define these terminologies and then define the LOF.

- $k - distance(p)$: is the maximum distance between p to the k neighbors closest to the p.
- $N_{k-distance(p)}(p)$: all the neighbors that have distances to p less than or equal to $k - distance(p)$. $|N_{k-distance(p)}(p)|$ can be more than k because of some of the distances might be equaled.
- $reach - dist_k(p, o)$: is equal to $\max\{k - distance(o), d(p, o)\}$. By using this formula, the fluctuation of $reach - dist_k(p, o)$ will be smoother when p is closed to o and the higher the k, the smoother it is. $reach - dist_k(p, o)$ represents the degree of not being able to reach p from o.
- $lrd_k(p)$: is equal to $1/(\frac{\sum_{o \in N_{k-distance(p)}(p)} reach-dist_k(p,o)}{|N_{k-distance(p)}(p)|})$. The denominator is the average degree of not being able to reach p from the p's neighbors. Thus, the inverse of the denominator is the local reachability score. $lrd_k(p)$ monotonically increases respect to the local density of point p.

- $LOF_k(p)$: is equal to $\frac{\sum_{N_{k-distance(p)}(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|N_{k-distance(p)}(p)|}$. This implies that $LOF_k(p)$ does not only depend on the local reachability of p but also the local reachabilities of all $k-distance$ neighbors of p. If the lrd_k of p is small comparing to their neighbors', p is more likely to be an outlier. LOF_k values of normal points are around 1 or less.

For this algorithm, there is only one tuning parameter which is k and it is usually called the $MinPts$. Usually, when $MinPts < 10$, $LOF_{MinPts}(p)$ fluctuates respect to the choice of $MinPts$. The fluctuation is caused by the unsmooth change of $reach - dist_{MinPts}(p, o)$ especially when o is getting closer to p. Choosing $MinPts$ is application-dependent. In general, setting the lower bound of $MinPts$ value between 10 and 20 is good enough. Beside that, a group of 30 close-by points can be considered as a normal cluster only when $MinPts$ is around 30 or less. Thus, the upper bound of $MinPts$ is set to be the maximum number of close-by objects that can potentially be local outliers.

3.2 Robust PCA Classifier

If all the original features are normally distributed the principle components(PCs) that we get from the PCA are independently normally distributed with variances equal to eigenvalues getting from the single value decomposition. Then, we have the fact that $\sum \frac{PC_j}{\lambda_j}$ should follow chi-square distribution with degree freedom of the number of terms in the summation.

According to [4], robust PCA is used to detect outliers by the following algorithm:

Step 1: Remove all the obvious outliers. This can be done by repeatedly remove the observations with high Mahalanobis distance for a number of times. Mahalanobis distance of point $i = \sqrt{(x_i - \bar{x})^T S^{-1}(x_i - \bar{x})}$, where S is the covariance matix, x_i is observation i's features and $\bar{x}$ is the means of all features [5].

Step 2: Do PCA on the training data set, find the corresponding PCs of the testing set and calculate outlier scores:

Score 1 $= \sum_{j=1}^{q} \frac{PC_j^2}{\lambda_j}$ (Major PC Outlier Score)

Score 2 $= \sum_{j=p-r+1}^{p} \frac{PC_j^2}{\lambda_j}$ (Minor PC Outlier Score)

Step 3: Make prediction basing on the score values

Score 1 $\geq c_1$ or Score 2 $\geq c_2$, the observation is an outlier

Score 1 $< c_1$ and Score 2 $< c_2$, the observation is not an outlier

where

- False positive rate from Score 1 detection is
  $\alpha_1 = P($ Score 1 $\geq c_1|$ obs is normal $)$
- False positive rate from Score 2 detection is
  $\alpha_2 = P($ Score 2 $\geq c_2|$ obs is normal $)$
- To make c_1 and c_2 robust to the normal assumption of principle components (PCs), we can estimate c_1 and c_2 by the empirical distribution of Score 1 and Score 2 respectively.
- Proportion of points that are falsely reject by both Score 1 and Score 2 is $\alpha_1\alpha_2$ since all PCs are orthogonal and Score 1 and Score 2 are independent. Thus, total false positive rate is $\alpha = \alpha_1 + \alpha_2 - \alpha_1\alpha_2$. Then, by using Cauchy-Schwartz and Bonferroni inequality,
  $\alpha_1 + \alpha_2 - \sqrt{\alpha_1\alpha_2} \leq \alpha \leq \alpha_1 + \alpha_2$
  This inequality can be used to set the desire false positive rate.

Robust PCA detects 2 types of outliers. They are dependency-oriented outliers and extreme-value outliers:

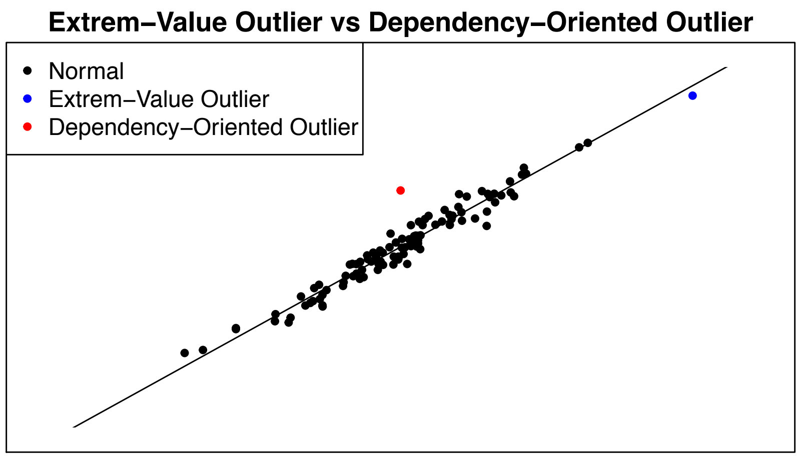

Figure 3.1: Dependency-Oriented Outliers do not have the right correlation between features. Extreme-Value Outliers might have the right correlation between features but the feature values are too extreme.

- **Dependency-oriented outliers:** This type of outliers is detected by Score 2. The detection is base on the assumption that all features of normal data points are generated from only r latent features which are the first r PCs. Thus, the $p-r$ PCA directions are all noise. If all data points are projected onto the first r-PC subspace, the inliers will have small errors while the outliers will have large errors since the outliers are not generated by the r latent features [6]. In the case that there are only two features in total and $r = 1$, the r-PC subspace is the regression line between the features. The regression line represents the correlation of the bivariate feature data while r-PC subspace represents the correlation of multivariate feature data in the case that $r > 1$. Thus, Score 2 outliers are the outliers that have strange correlation between their features. The red point in the plot above is a dependency-oriented outlier.

- **Extreme-value outliers:** This type of outliers is detected by Score 1. If data is assumed to be generated from q latent features which are the first q PCs. The data points that are closed enough to the q-PC subspace are considered to have the right correlation between features. However, these points are outliers if they have extreme coordinates and Score 1 can detect this [6]. The blue point in the plot about is an extreme-value outlier. Even though the blue point is very closed to the regression line, but its coordinates are too extreme.

Last but not the least, it sometimes is not a good idea to put the two scores together and detect both types of outliers at the same times since one of the scores in step 2 may be much larger than the other score. This makes the small value score almost has no votes when being putted together with the large score.

In order to achieve good prediction, the parameter q, r, c_1, and c_2 should be tune. q can be set to get the best Score 1 that when using the Score 1 alone as outlier score, the model gives the highest AUC or the best other performance measures as needed. Similarly, r should be chosen to get the best Score 2. Then, the right combination of c_1 and c_2 are chosen to get the right false positive rate and best performance.

Beside that, if all the PCs are put together and the outlier scores are $\sum_{i=1}^{p} \frac{PC_i^2}{\lambda_i}$, the robust PCA will become Mahalanobis distance method.

3.3 Autoencoder

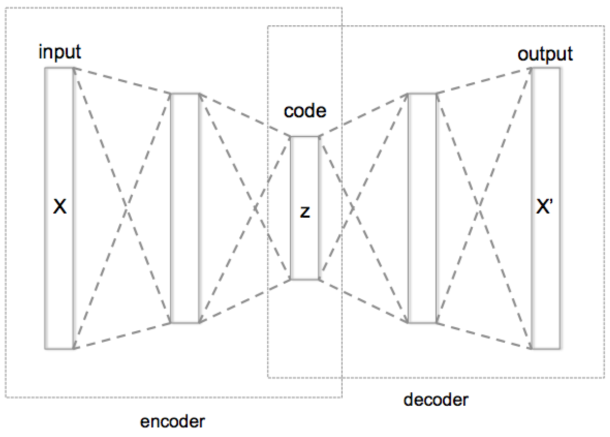

Figure 3.2: Autoencoder Structure

Autoencoder is a neural network that the input variables and the output variables are the same. The structure of autoencoder are divided into the encoder and decoder parts as shown in figure 3.2. From input, the encoder produces codes which can be used as dimension reduction. Then, the decoder reconstructs the input back. By assuming that all features of the inliers are generated from the reduced dimensions, the errors of input reconstruction can be used as outlier scores since the higher the errors, the more likely that the observations are outliers.

3.4 Self-organizing map (SOM)

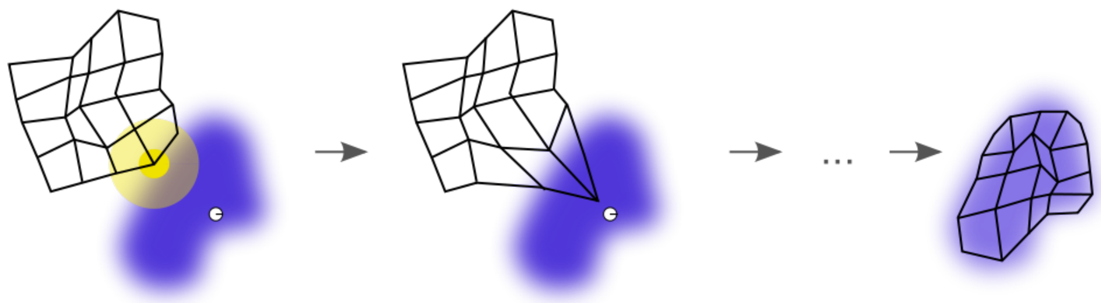

Figure 3.3: Training Procedure of Self-organizing Map

Self-organizing map (SOM) was introduced by [7]. SOM is also a type of dimension reduction that maps the original data points on to discrete number of points. The mapped points are called neurons. The index of the neurons can be 1-component, 2-component, 3-component, etc. In the 1-component index case, the neurons are lying on a flexible line. In the 2-component index case, the neurons are lying on a flexible towel. Usually, 2-component index neurons are used and I will describe the training procedure. First, we start with a towel with it initial position and shape that looks like the first picture in figure 3.3. On the towel there is a gill of neurons with 2-component index. For each original data point, we find the winning neuron that is closest to the point. Then, all the neurons are updated to get closer to the point that we are working on. How much each neuron is updated is depending on how closed it is to the winning neuron. The closer the neurons to the winning neuron, the more update it is and the wining neuron receives the most update. Also, the update is done in a way that the topological order of the neurons on the grill does not change. As shown in figure 3.3, after finding the yellow wining neuron for the white point, all the neurons are updated as described above. Then, we have the second picture in figure 3.3. Next, we will do the same thing for every single data point. After repeating the same action on all the data points for enough iterations, the towel will cover all the original data points very well as show in the last picture of figure 3.3. This makes the neurons to be able represent data points. The distance between the original points to the mapped neuron points are the errors. Like autoencoder, the errors are used as outlier scores.

3.5 One-Class SVM

One-Class SVM is a variation of SVM that can detect outliers. From [8], this algorithm tries to separate outliers from the origin with maximum margin. The observations inside the separating boundary are predicted as normal observations ($\hat{y}_i = 1$) and the observations outside the boundary are predicted as outliers ($\hat{y}_i = -1$) .
The Objective function for one-class SVM is

$$\min_{w \in Y, \xi_i \in \mathbb{R}, \rho \in \mathbb{R}} \frac{1}{2}||w||^2 + \frac{1}{vl}\sum_{i}^{l} \xi_i - \rho$$

subject to:

$$y_i(w \cdot x_i) \geq \rho - \xi_i \text{ for all } i = 1, 2, ..., n$$

$$\xi_i \geq 0 \qquad\qquad \text{for all } i = 1, 2, ..., n$$

where

- l : the number of data points
- w: the hyperplane's normal vector in the feature space
- ρ: the origin
- ξ_i: slack variables
- v: upper bound on the fraction of outliers and lower bound on the fraction of support vectors. In the semi-supervised case, the training data contains the normal observations only. Thus, v can be used to control the false alarm rate. The increase in v leads to the increase in false alarm rate.

Usually, Gaussian kernel is used in this algorithm in order to achieve the non-linear decision boundary and have the ability to create multiple decision boundaries for inliers with multi-mode distributions [9]. The Gaussian kernel is

$$K(X, Y) = e^{-\frac{|X-Y|^2}{2\sigma^2}}.$$

3.6 Isolation Forest

Isolation forest is a tree-base model that is developed to detect outliers. This algorithm was invented by [10]. To begin with, let define isolation tree, isolation forest and path length (h(x)).

Given a sample of data $X = \{x_1, ..., x_n\}$ of n instances from a d-variate distribution, an isolation tree is built by recursively dividing X by randomly select a feature and a split value until either: (i) the tree reaches a height limit, (ii) only one point left in the node or (iii) all data in the node have the same values.

Isolation forest is a group of isolation trees that are built for the same data.

In addition, path length of a point x (h(x)) is measured by the number of edges x traverses an isolation tree from the root node until the traversal is terminated at an external node. As shown in the figure below, outliers tend to have small value of h(x).

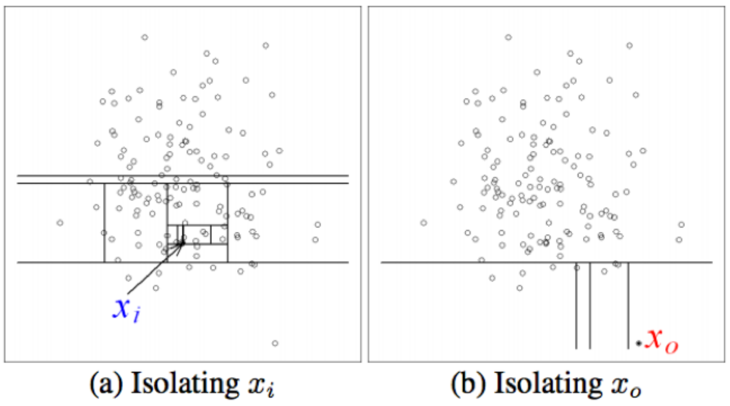

Figure 3.4: Outliers have smaller path length than normal observations.

The average value of h(x) from isolation forest usually converges when the number of trees is large enough. In this algorithm, the smaller the average of h(x), the more likely the point x is an outlier. Thus, the average of h(x) is used as an outlier score.

CHAPTER 4

Experiment on Generated Data sets

In this chapter, 5 data sets are generated to represent different outlier detection scenarios. Then, performances of different models for the 5 data are compared in the discussion part. Moreover, some suggestions for improving the algorithms' weak points are given.

4.1 Data Set Descriptions

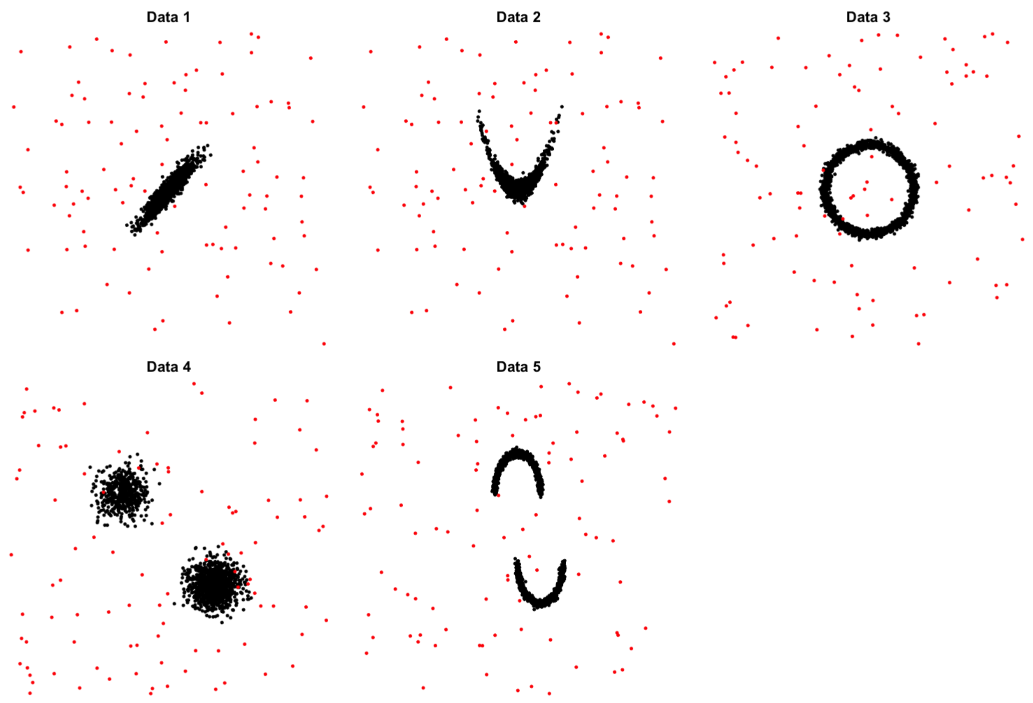

Figure 4.1: 5 Generated Data

In each of the 5 data, training set has 1045 data points which are all inliers and the testing set has 555 data points in which there are 455 inliers and 100 outliers. The outliers in all data sets are uniform points over the square: x between -7 to 7 and y

between -7 to 7 since it is assumed that the outliers can be anything. Besides, the normal points in the data sets are generated as described bellow:

- Data 1: The normal points are bivariate normal points.
- Data 2: The normal points are generated by the equation $y = x^2 + \epsilon$, where x and ϵ are normally distributed.
- Data 3: The normal points are points on a circle with uniform noises.
- Data 4: The normal points are two bivariate normal. The two bivariate normal distributions have the same covariance matrix and the lower-right bivariate normal is 2 times denser than the upper-left one.
- Data 5: The normal points are points on upside-down-U shape and U shape with uniform noises.

In data 1, data 2 and data 3, the normal points are only one cluster while in data 4 and data 5, there are two clusters being apart from each other.

4.2 Experiment Results

4.2.1 Outlier Score Contour Plots

The contour plots are grouped by data. The blue points in the SOMs are neurons.

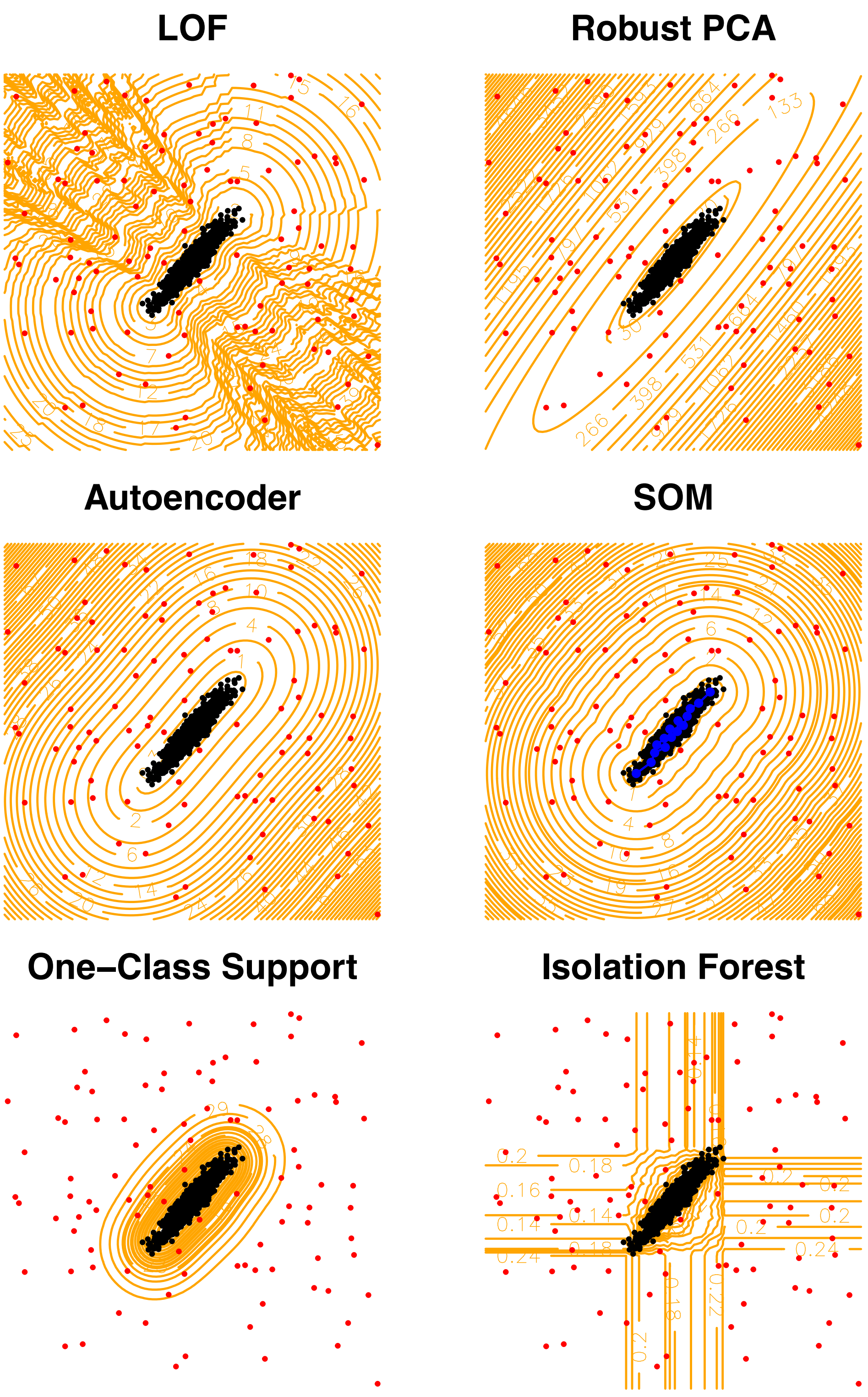

Figure 4.2: Data 1 Outlier Score Contour Plots

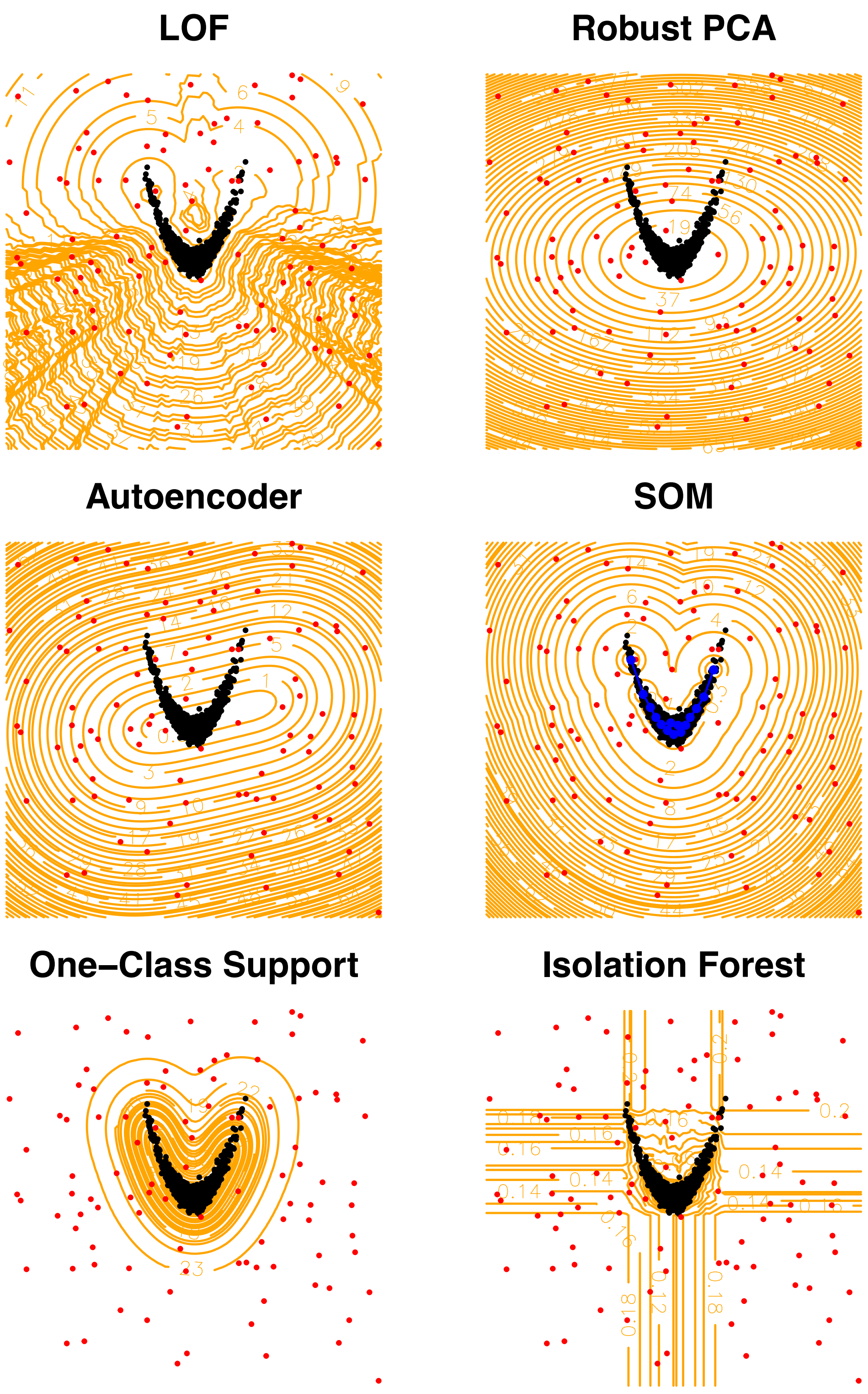

Figure 4.3: Data 2 Outlier Score Contour Plots

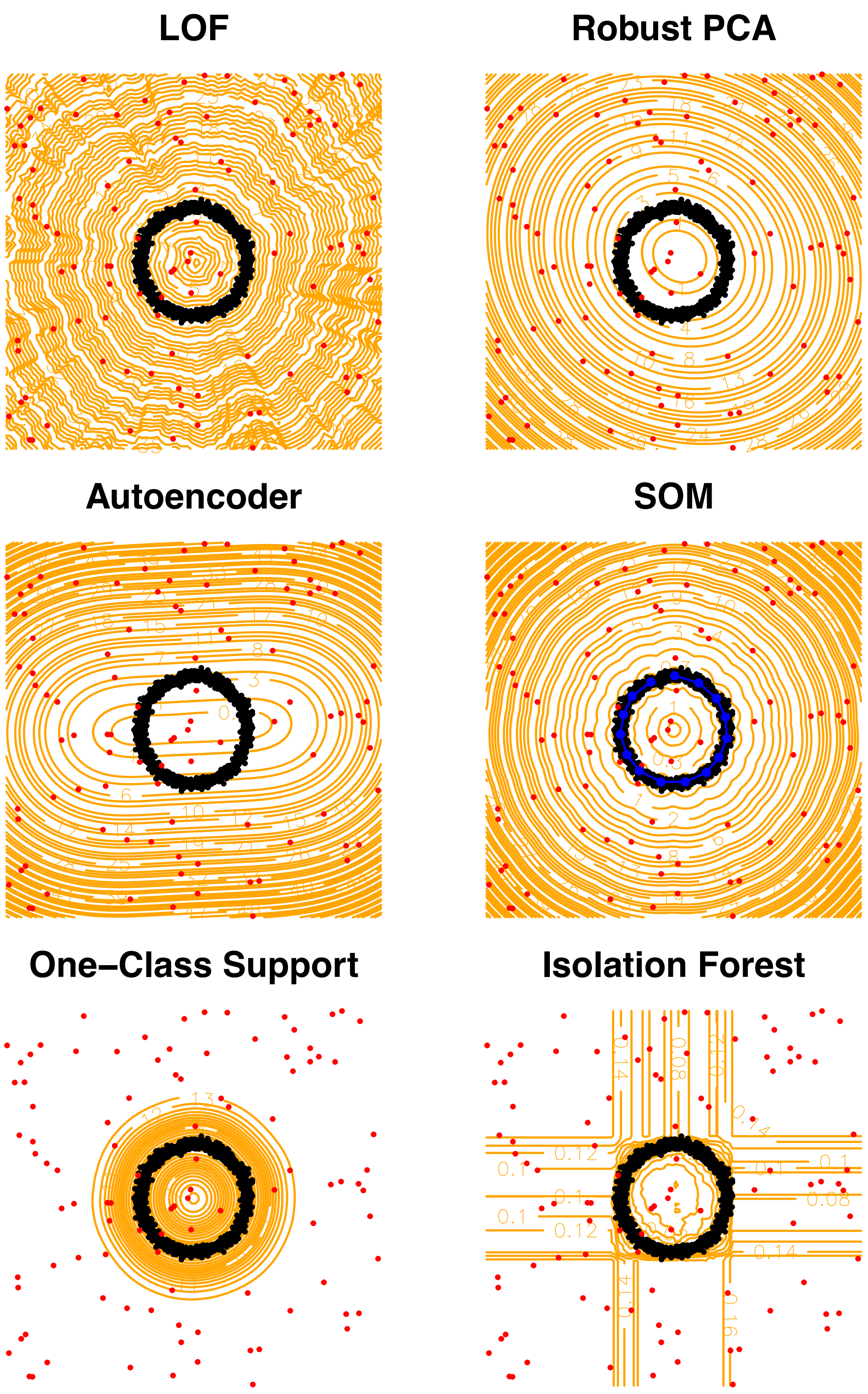

Figure 4.4: Data 3 Outlier Score Contour Plots

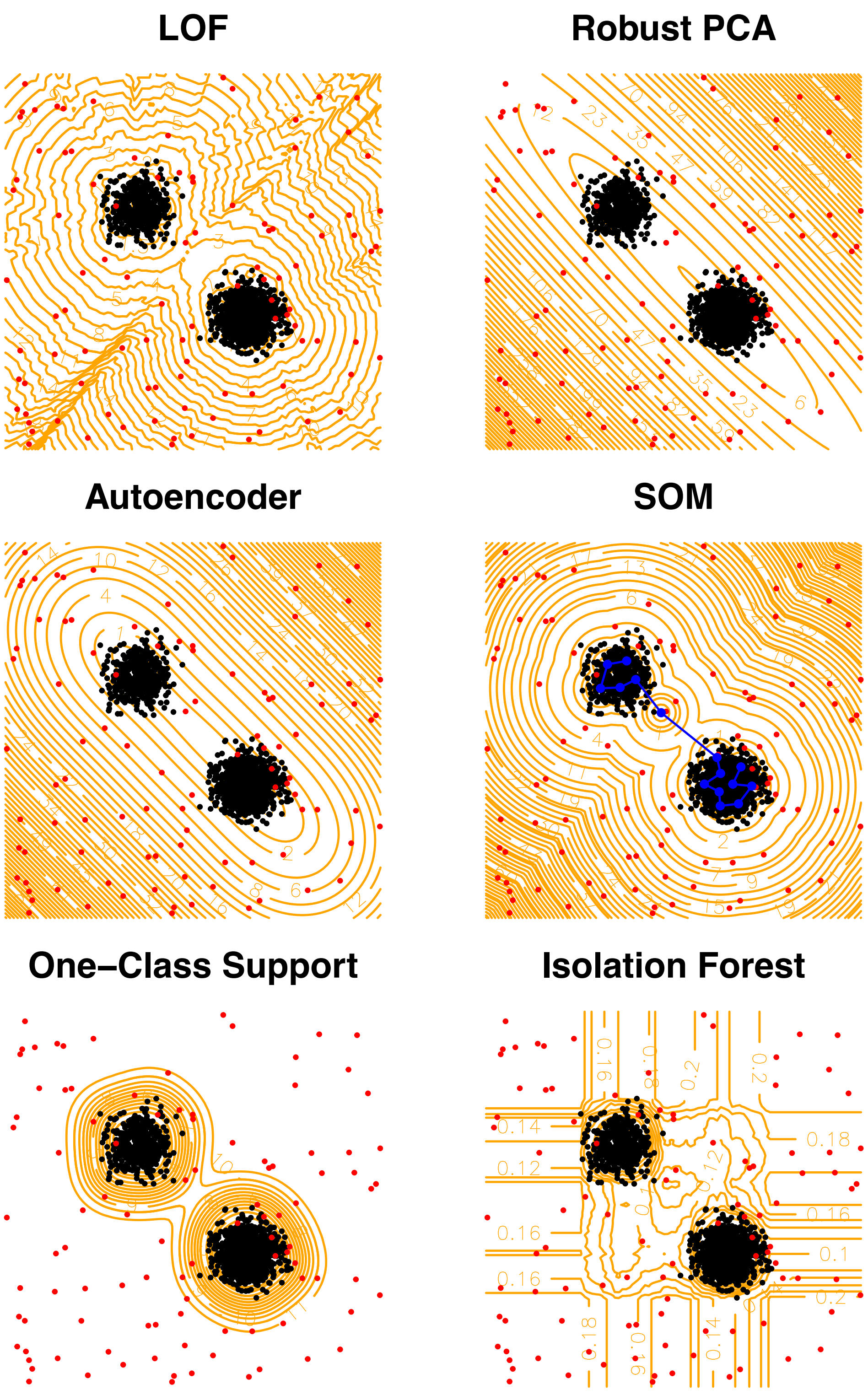

Figure 4.5: Data 4 Outlier Score Contour Plots

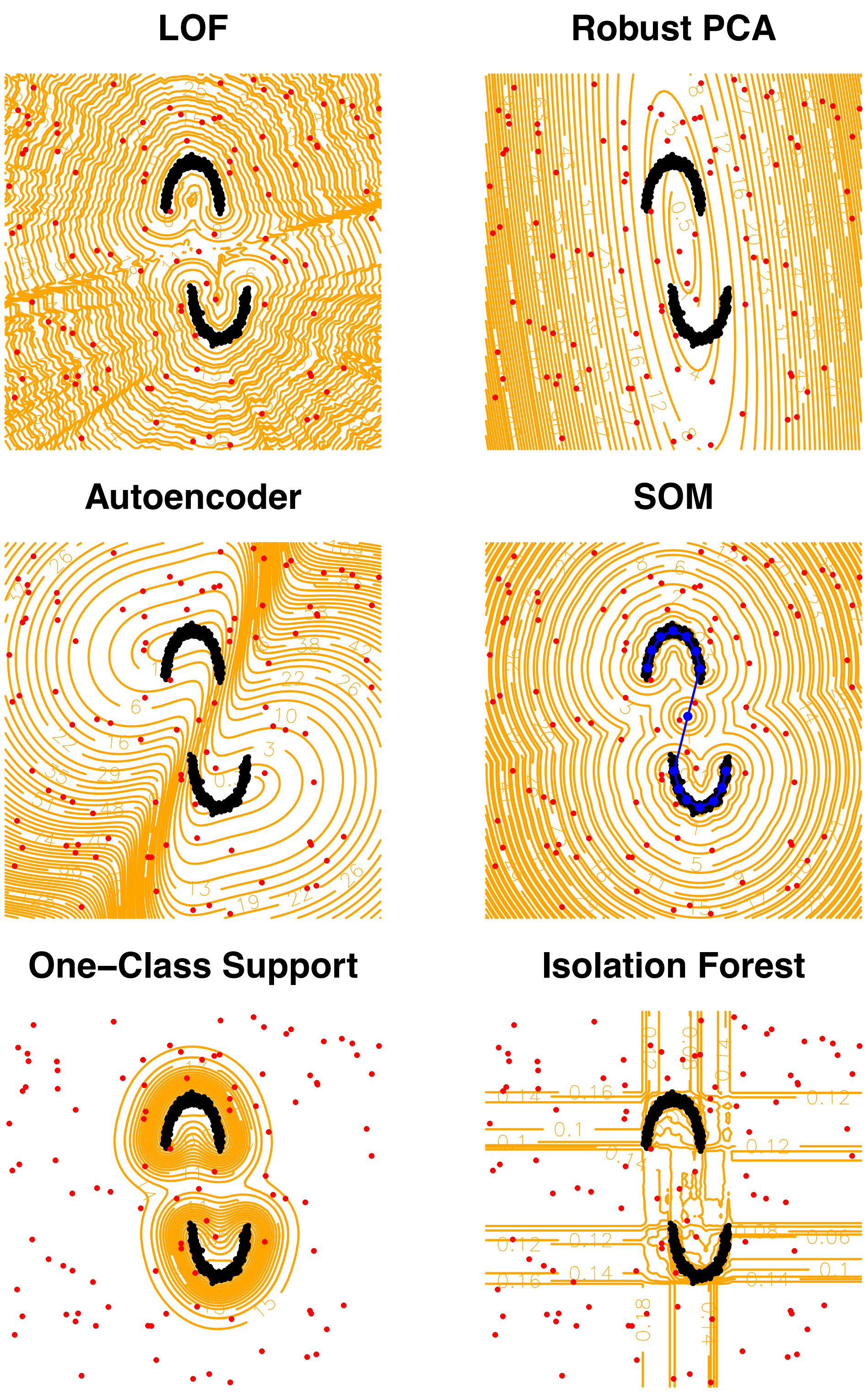

Figure 4.6: Data 5 Outlier Score Contour Plots

4.2.2 ROC Curves

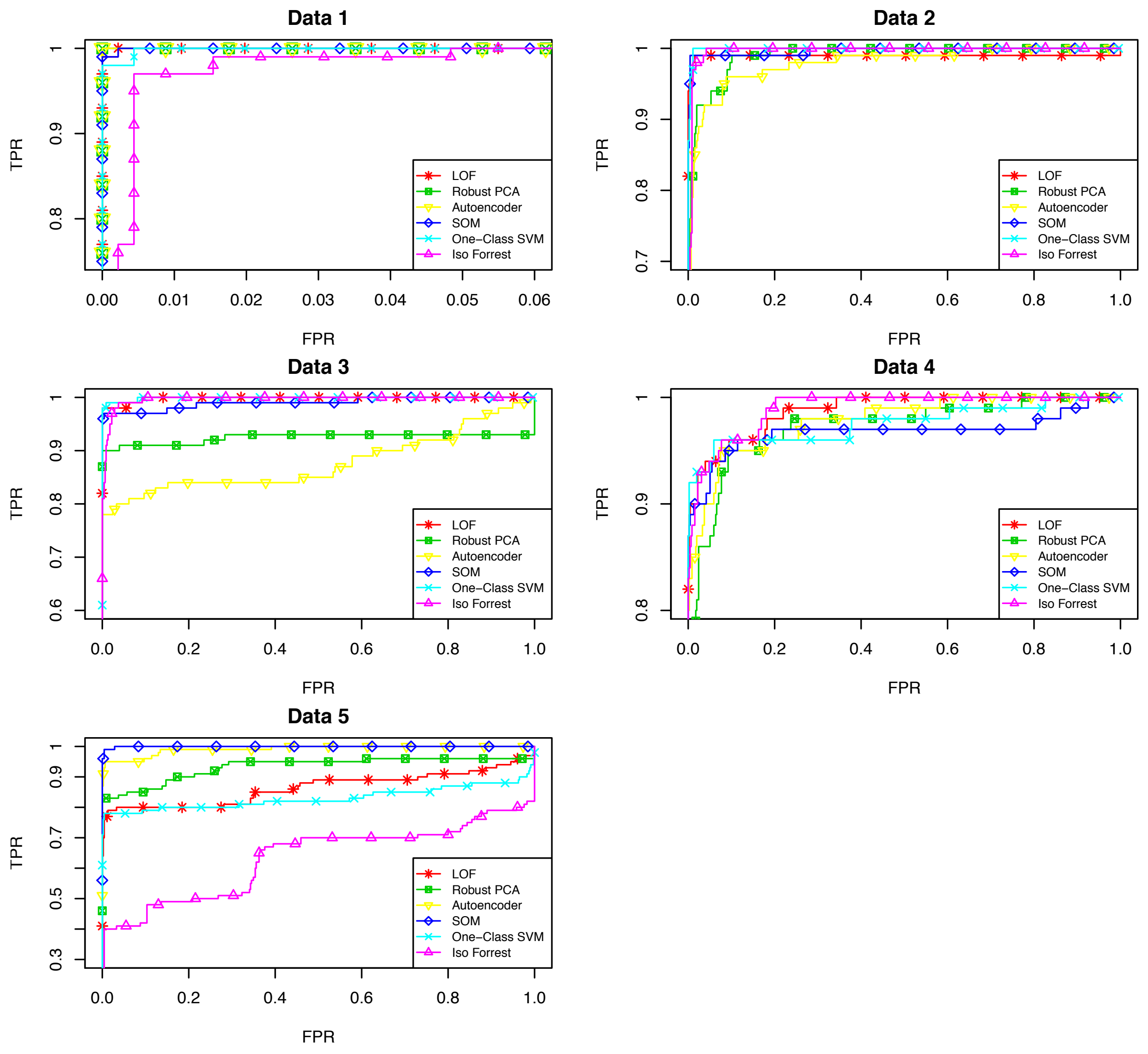

Figure 4.7: ROC for the 5 Data

4.2.3 F_1 and AUC

In the tables below, , the bold numbers are the best result in the columns. For F_1, I set the number of observations that are predicted as outliers to be 100, which is the total number of actual outliers. I do like this because I want to let all the algorithms have chance to do perfect prediction and detect the 100 outliers without making any mistakes. Since the total number of actual outliers equals the total number of observations that are predicted to be outliers, Precision = Recall = F_1. Thus, Precision, Recall columns are not shown. In addiction, because of the number of observations that are predicted as outliers are quite small, the performances of how good the detections of outliers that are not closed to the normal points can be measured by F_1 scores.

Table 4.1: Data 1 F_1 and AUC

Methods	F_1	AUC
LOF	0.99	0.9994
Robust PCA	**1.00**	**1.0000**
Autoencoder	**1.00**	**1.0000**
SOM	0.99	0.9994
One-Class SVM	0.98	0.9993
Isolation Forest	0.97	0.9977

Table 4.2: Data 2 F_1 and AUC

Methods	F_1	AUC
LOF	0.96	0.9895
Robust PCA	0.91	0.9895
Autoencoder	0.89	0.9793
SOM	**0.98**	0.9967
One-Class SVM	0.96	**0.9993**
Isolation Forest	0.95	0.9942

Table 4.3: Data 3 F_1 and AUC

Methods	F_1	AUC
LOF	**0.98**	0.9985
Robust PCA	0.90	0.9244
Autoencoder	0.80	0.8773
SOM	0.97	0.9904
One-Class SVM	**0.98**	**0.9991**
Isolation Forest	0.93	0.9967

Table 4.4: Data 4 F_1 and AUC

Methods	F_1	AUC
LOF	0.90	0.9878
Robust PCA	0.86	0.9717
Autoencoder	0.87	0.9781
SOM	0.90	0.9680
One-Class SVM	**0.92**	0.9762
Isolation Forest	0.90	**0.9896**

Table 4.5: Data 5 F_1 and AUC

Methods	F_1	AUC
LOF	0.80	0.8637
Robust PCA	0.83	0.9327
Autoencoder	0.95	0.9911
SOM	**0.98**	**0.9996**
One-Class SVM	0.78	0.8299
Isolation Forest	0.48	0.6316

4.3 Discussion

LOF: LOF detects outliers basing on local density. In the case that there are a couple of clusters of normal points with different densities, the clusters with low densities are still considered as normal clusters as long as they have at least $MinPts$ number of low density points. In data 4, LOF allows the low density bivariate normal to form a normal cluster; this power makes LOF rank number 2 in term of AUC.

On the other hand, because of local density detection, the sparse outliers in the tail of some distributions, such as bivariate normal, that have enough sparse members in their $k - distance$ neighborhoods are considered as normal points. As shown in the

LOF contour plots of data 1 and data 2, the outlier scores around the sparse tails are quite low.

This problem can be reduced by setting larger $MinPts$ to make the sparse points need to include some dense points in their $k - distance$ neighbors. As a result, the sparse points will have higher outlier scores. In spite of that, this solution can lead to higher value of $k - distance$ of all the points as well as dense points. Then, the outlier score penalty for the points that deviate from the dense inlier points will decrease quite a bit. This makes the contour curves of outlier scores around the dense inliers look less tight . In the left illustration in the figure bellow, I use $MinPts = 80$ for data 1 and I see that the contour curves near the dense inliers are looser than the one in figure 4.2. Also, the contour curves are even looser when I use $MinPts = 1000$ in the right illustration.

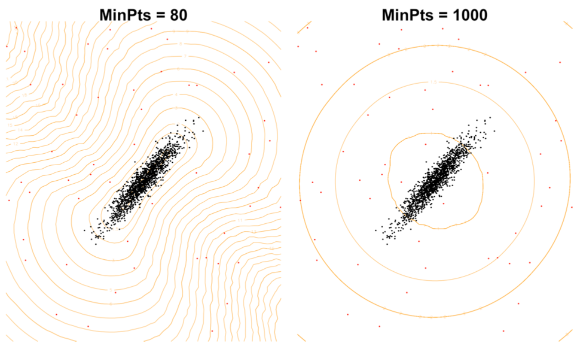

Figure 4.8: LOF: MinPts = 80 and MinPts = 1000

Robust PCA: In this part, I use all-principle-component robust PCA. Thus, robust PCA will draw an ellipsoid-shape decision boundary. The advantage is when the multivariate normal assumption is met, robust PCA can do extremely good. The perfect result is obtained in data 1 for robust PCA.

This method has 3 disadvantages. One is that when the multivariate normal assumption is not met, an ellipsoid-shape decision boundary is the best that robust PCA with all PCs can do. Two is that the center points of the ellipsoid have the lowest outlier scores. This makes robust PCA misses the outliers around the centers of the ellipses in data 2, 3, 4 and 5. Beside that, using Mahalanobis distance to remove the obvious outliers sometimes does not work when the distribution is not multivariate normal. For data 3, the lower-left part of the ring is removed after repeatedly excluding 1% of observations with the highest Mahalanobis distances for 15 times.

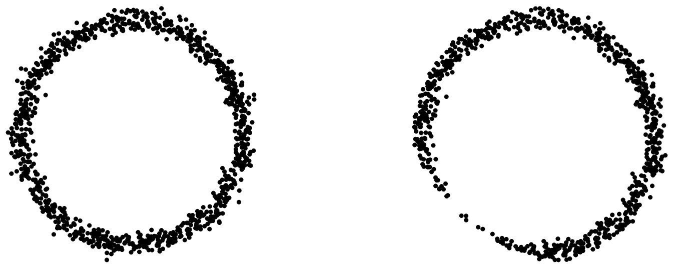

Figure 4.9: Normal Points of Data 3: Before vs After Removing 1 % For 15 Times

Autoencoder: Since the numbers of neurons in the layers are set to be 2-1-2, autoencoder is reducing the dimensions from 2 to 1 for the 5 data. The reduced dimension can be linear or none-linear since the activation functions are “Tanh.” Autoencoder achieves perfect result in data 1 since the bivariate normal data can be reasonably reduced the dimensions to one. However, the prediction of data 2, 3, 4 are not good since it is not reasonable to reduce the dimensions to one. Interestingly, because of non-linear dimension reduction, the autoencoder AUC for data 5 ranks number 2.

SOM: Since there are only 2 dimensions in all data sets, I use 1-component-index SOM with 15 neurons to make it fair to other algorithms. For all the 5 data, SOM does a good job in using its 15 neurons to represent the data. For two separated clusters in data 4 and 5, the two clusters are connected by a bridge with only 1 SOM neuron wasted on the bridge. Interestingly, for data 4, the cluster which is twice denser has twice more SOM neurons representing it. The above properties make the SOM's F_1 values at least get the second rank.

However, simply using errors as outlier score leads to 2 problems. First, the decision boundaries of SOM near inliers are just the connected circles with SOM neurons as their centers as shown in Figure 4.10. It is better to make the boundary look smoother. This problem does not affect the AUCs in the 5 data much since most outliers are not too close to the inliers. However, the effect is a bit more severe in chapter 5.

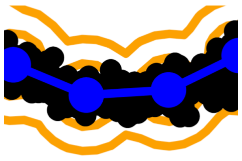

Figure 4.10: A fraction of contour plot of data 3 when SOM is used. The blue dots are neurons.

The solution for this problem is that the distance between the data points to the segment made by the 2 closest neurons to the data points should be used as outlier scores instead. Similarly, in the 2-component-index SOM, the distance between the data points to the planes made by the 3 closest neurons to the data points should be used as outlier scores. Doing like this, a better decision boundary will be obtained.

The second problem is the equality of neurons. The problem is that the outliers which are closed to the outer neurons as shown in the figure bellow cannot be detected since errors of the mapping are almost zero.

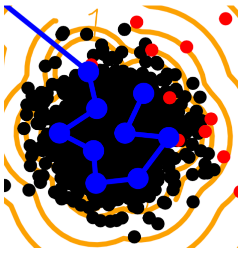

Figure 4.11: The Equality of Neurons Problem. 2 outliers are extremely close to the neurons.

This problem can be solved by giving a lower outlier scores to the points that is nearby the centers of clusters. It can be done by the following procedure. First, cluster the SOM neurons into clusters. Then, for each cluster, the outlier score for each data point is the SOM error weighted by how close the winning neuron to the center of its cluster is. The measurement of how close should be scaled to 0 and 1 since the clusters might have different sizes. However, this method depends so much on how the neurons are clustered, how the center of each cluster is defined and how to weight SOM errors. I hope that there will be a detail study on the procedure.

One-Class SVM: I use the Gaussian kernel for the 5 one-class SVMs. The one-class SVM does capture the multi-mode inlier points and gets the best F_1 in data 4. Furthermore, one-class SVM has slack variables(ξ_i). This allows the optimization of the objective function care less about the data points with non-zero slack variables. In other words, it allows one-class SVM to not care much about a few strange points in the training set. Because of the slack variables, one-class SVM have the best AUC in data 2 and 3.

A weak point of Gaussian one-class SVM is that even though the kernel allows one-class SVM to form variety shapes of decision boundaries, the shapes cannot be as complex as LOF's and SOM's and we cannot understand much about boundaries of one-class SVM.

Isolation Forest: For outliers that do not share similar vertical coordinates or horizontal coordinates with normal points', their isolation forest outlier scores are among the highest and are almost the same to each other. For outliers that have similar horizontal coordinates or vertical coordinates to the inliers, their outlier scores are lower since the cuts of isolation trees are done horizontally or vertically and this group of outliers need more cuts to be isolated from the normal points.

Finally, SOM achieves the overall best F_1 score while one-class SVM have the best overall AUC in the 5 data sets. This means that SOM is very good at detecting outliers that are a bit far from the inliers and one-class SVM's overall detections outperform the others'. Also, LOF interestingly has the second best overall F_1 and AUC.

CHAPTER 5

Experiment on Real Data Sets

5.1 Data Set Descriptions

5.1.1 Breast Cancer

This data is from University of Wiscosin Hospital [11]. This data set has 9 features, which are discrete number from 1 to 10. There are two classes, benign and malignant. The malignant class of this data set is considered as outlier, while the benign class is considered as inlier. In total, there are 699 data points: 458 inliers and 241 outliers.

70 % of the inliers are chosen to be training set while test set is the other 30% combining with all the outliers. As a result, there are 320 observations in the training set and 379 in the testing set.

5.1.2 Credit Card Fraud Detection

The datasets is gotten from Kaggle [12]. It contains transactions made by credit cards in September 2013 by European cardholders. In the data, there are 492 frauds out of 284,807 transactions. The data set is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. The frauds are considered as outliers while the others are inliers.

In total, there are 30 features. 28 are the result of a PCA transformation. Unfortunately, due to confidentiality issues, they cannot provide the original features and more background information about the data. Beside the PCA features, there are Time and

Amount. In this experiment the Time is removed. Thus, there are 29 features.

7000 data points are sampled from the inliers and are set to be the training set. Another 3000 data points are sampled from the inliers and are combined with all the outliers in order to make the testing set. As a result, there are 3492 observations in the testing set.

5.2 Experiment Results

5.2.1 ROC Curves

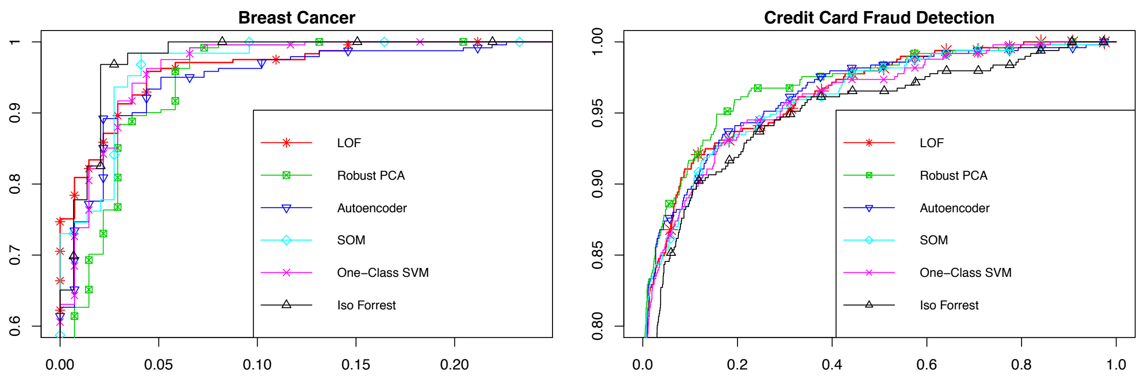

Figure 5.1: ROCs for Breast Cancer and Credit Card Fraud Detection Data

5.2.2 F_1 and AUC

In the tables below, the bold numbers are the best result in the columns.The Precision and Recall are given for reference.

Table 5.1: Breast Cancer F_1 and AUC

Methods	Precision	Recall	F_1	AUC
LOF	0.9512	0.9710	0.9610	0.9908
Robust PCA	0.9594	0.9793	0.9692	0.9897
Autoencoder	0.9431	0.9627	0.9528	0.9871
SOM	0.9594	0.9793	0.9692	0.9872
One-Class SVM	0.9634	0.9834	0.9733	**0.9912**
Isolation Forest	**0.9675**	**0.9876**	**0.9774**	0.9909

Table 5.2: Credit Card Fraud Detection F_1 and AUC

Methods	Precision	Recall	F_1	AUC
LOF	0.9543	0.6789	0.7934	0.9627
Robust PCA	0.9314	0.6626	0.7744	**0.9692**
Autoencoder	0.9457	0.6728	0.7862	0.9628
SOM	0.9486	0.6748	0.7886	0.9603
One-Class SVM	**0.9686**	**0.6890**	**0.8052**	0.9599
Isolation Forest	0.8829	0.6281	0.7240	0.9465

5.3 Discussion

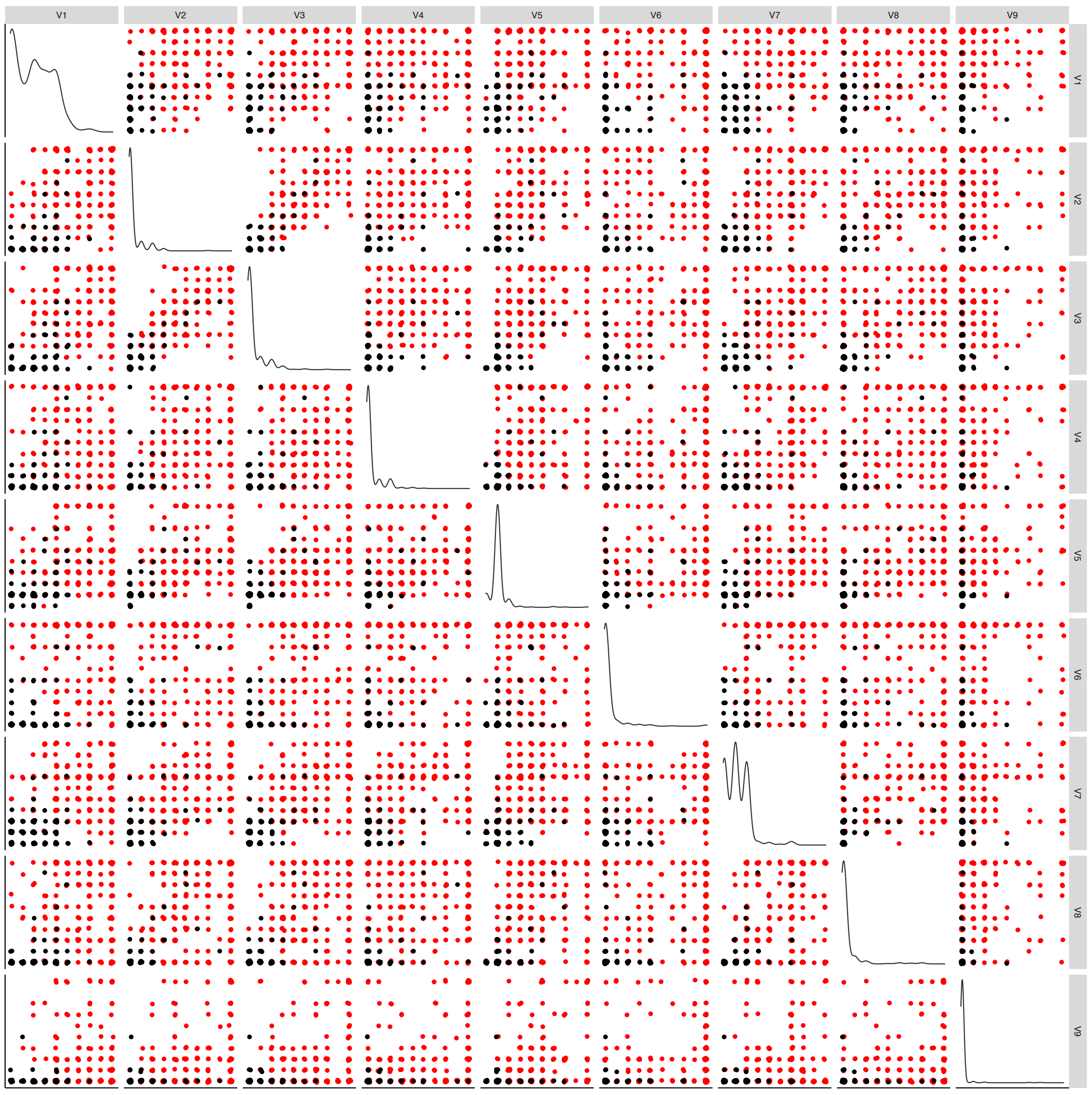

Figure 5.2: Pairwise Plots for Breast Cancer Features. I jitter the points since a lot of points are overlapping. The diagonal curves are the estimated inlier densities of the features.

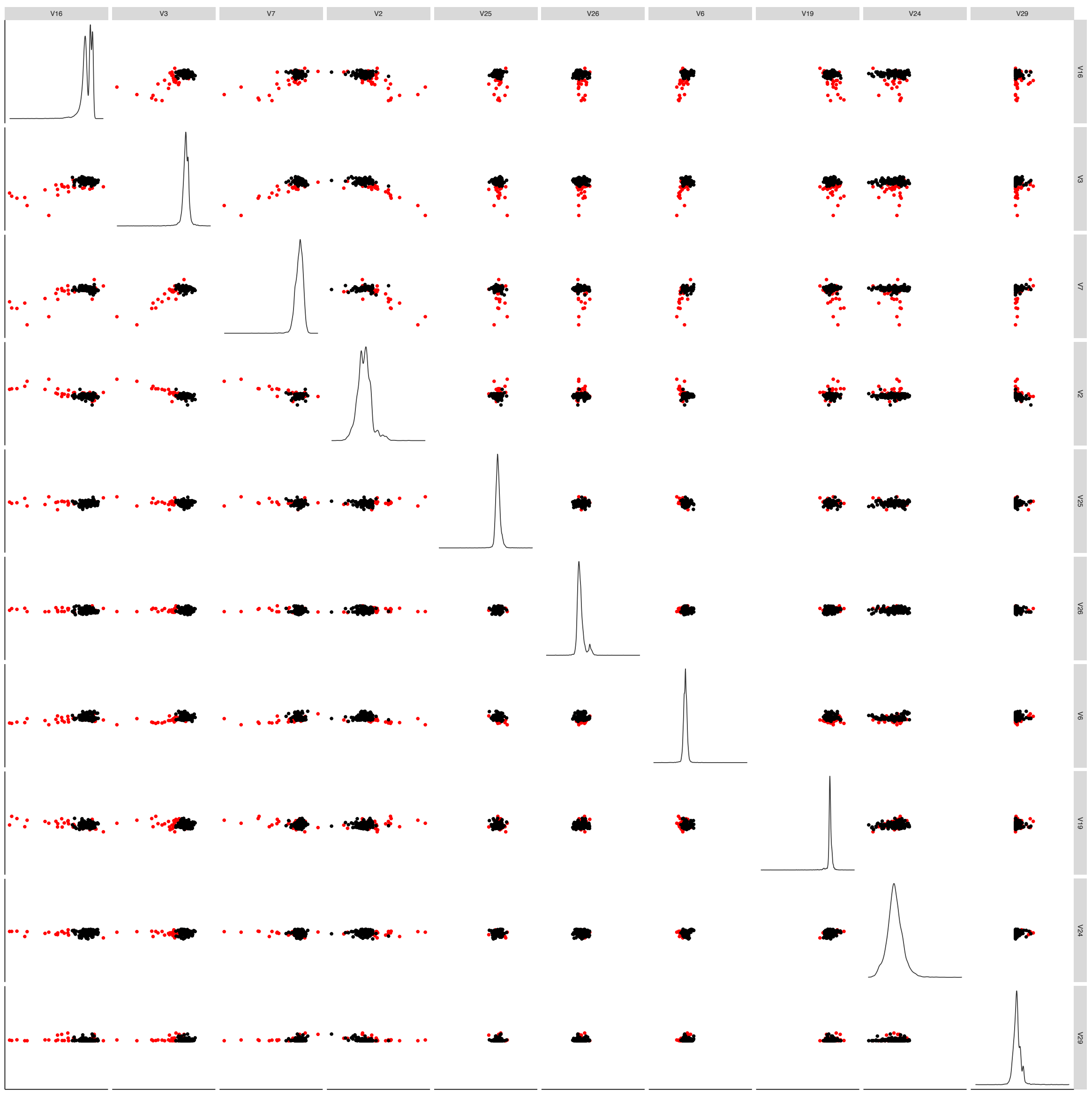

Figure 5.3: Pairwise Plots for Credit Card Fraud Detection Data Features. The first nine features are 9 randomly selected principle components out of the 29 . The last feature is the Amount feature. The diagonal curves are the estimated inlier densities of the features.

LOF: In Breast Cancer ROCs, LOF has the best startups. However, after FPR of around 0.025, the performance of LOF was not good anymore. This might be because LOF starts with detecting the outliers nearby the dense inliers and it does very well; then, sparse points in the tail of inlier distribution as shown in figure 5.3 cause the problem described in section 4.3. This is the best that LOF can do since I have tuned $MinPts$.

Robust PCA: As shown in the below figure, the performances of Score 1, Score 2 and the combination of Score 1 and 2 detections are depend on the nature of the outliers.

I have checked all of scores and choose the best.

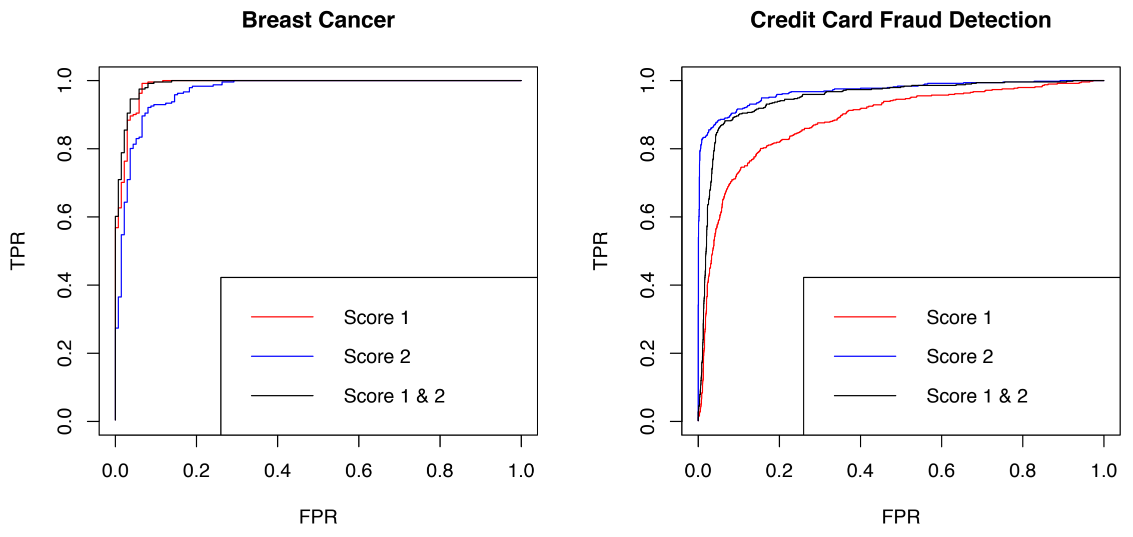

Figure 5.4: ROC of using Score 1, using Score 2 and using both Score 1 and Score 2 with appropriate c_1 and c_2 in Breast Cancer and Credit Card Fraud Detection

Beside that, comparing with other algorithms, robust PCA does not do so good in the Breast Cancer data. However, it has the best AUC in the Credit Card Fraud Detection. This is because of the normal assumption of the features. Figure 5.2 and 5.3 show that the features in Credit Card Fraud Detection are a lot more normal than the features in Breast Cancer.

Autoencoder: Autoencoder does not give good result in Breast Cancer. However, in Credit Card Fraud Detection data, autoencoder AUC is the second best.

SOM: The F_1 values of SOM in both data is not too bad. However, the AUCs of SOM in both are quit bad. This is because of the connected-circle boundary and equality of neurons problems.

One-Class SVM: The slack variable property is very helpful since there are a few strange inliers in both data sets as shown in figure 5.2 and 5.3. As a result, one-class SVM achieves best AUC in Breath Cancer data set and best F_1 in Credit Card Fraud Detection data.

Isolation Forest: In Breath Cancer data, only a small percentage of outliers have similar coordinates as inliers'; this makes isolation forest achieves the best F_1 and the second best AUC. However, in Credit Card Fraud Detection data, a lot of outliers have similar coordinates as inliers'; this makes isolation forest give the worst F_1 and AUC.

All in all, SOM's F_1 scores are not good as the ones in the generated data sets because the outliers are too closed to the normal points. One-class SVM achieves overall best F1 score while LOF has the overall best AUC in the two data sets.

CHAPTER 6

Conclusion

After the experiments on generated data and real data, I found that LOF allows inliers to have different densities clusters. However, LOF tends to give lower outlier scores to outliers around sparse tail regions of inlier clusters. Robust PCA can detect both dependence-oriented outliers and extreme-value outliers, but it cannot detect outliers around the center of a ring of inliers. Also, using Mahalanobis distance to remove the obvious outliers sometimes does not work when the distribution is not multivariate normal. Sometimes, autoencoder gives a very good result even though it is difficult to understand the non-linear dimension reduction mechanism that autoencoder does. SOM can detect the outliers that are a bit far away from SOM neurons very well. In spite of that, for outliers nearby inliers, SOM has connected-circle boundary and equality of neurons problems. Gaussian one-class SVM can capture multi-mode inlier distributions and slack variables allow boundaries to care less about some strange inliers. However, its variety of boundary shapes is limited and we cannot understand much about how the boundary shapes are structured. Finally, isolation forest detects the outliers that do not have similar coordinates with inliers' very well. The more similar the outliers' coordinates to the inliers', the lower the outlier scores are given by the isolation forest. The details of all of the described advantages and disadvantages of these algorithms can be found in section 4.3 and 5.3.

To sum up, gaussian one-class SVM and LOF have the best overall performance in term of both the detection of outliers that are far from the normal points and the overall detection. LOF can form extremely complex decision boundaries but the boundary is not smooth. Gaussian one-class SVM decision boundaries are flexible but not as complex as LOF. In spite of that, the boundaries of gaussian one-class SVM are very smooth. This makes gaussian one-class SVM has an advantage since most true decision boundaries are smooth. Besides that, for the sparse inliers that LOF is struggling with, one-class SVM can consider them as strange inliers and do not care about them. However, there is no perfect algorithms. In order to achieve good result, the characteristics of the data sets should be study in detail and the algorithms that have advantages should be chosen. I hope these pros and cons that I have found can help the readers in choosing the right models as needed. Last but not the least, if I have time, I study the suggested solutions to the cons of each algorithms as stated in section 4.3 more thoroughly.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[2] X.-H. Zhou, D. K. McClish, and N. A. Obuchowski, *Statistical methods in diagnostic medicine.* John Wiley & Sons, 2009, vol. 569.

[3] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[4] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, “A novel anomaly detection scheme based on principal component classifier,” MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING, Tech. Rep., 2003.

[5] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, “The mahalanobis distance,” *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[6] C. C. Aggarwal, “Outlier analysis,” in *Data mining.* Springer, 2016.

[7] T. Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, vol. 78, no. 9, 1990.

[8] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, “Support vector method for novelty detection,” in *Advances in neural information processing systems*, 2000, pp. 582–588.

[9] Y. Chen, X. S. Zhou, and T. S. Huang, “One-class svm for learning in image retrieval,” in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 1. IEEE, 2001, pp. 34–37.

[10] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on.* IEEE, 2008, pp. 413–422.

[11] W. H. Wolberg and O. L. Mangasarian, “Multisurface method of pattern separation for medical diagnosis applied to breast cytology.” *Proceedings of the national academy of sciences*, vol. 87, no. 23, pp. 9193–9196, 1990.

[12] Kaggle. (2017) Credit Card Fraud Detection kernel description. [Online]. Available: https://www.kaggle.com/dalpozz/creditcardfraud/data