sciendo

JAISCR, 2023, Vol. 13, No. 1, pp. 5 – 17
10.2478/jaiscr-2023-0001

A COMPARATIVE STUDY FOR OUTLIER DETECTION METHODS IN HIGH DIMENSIONAL TEXT DATA

Cheong Hee Park

Department of Computer Science and Engineering, Chungnam National University,
220 Gung-dong, Yuseong-gu
Daejeon, 305-763, Korea

**E-mail: cheonghee@cnu.ac.kr*

Submitted: 22nd June 2022; Accepted: 19th October 2022

Abstract

Outlier detection aims to find a data sample that is significantly different from other data samples. Various outlier detection methods have been proposed and have been shown to be able to detect anomalies in many practical problems. However, in high dimensional data, conventional outlier detection methods often behave unexpectedly due to a phenomenon called the curse of dimensionality. In this paper, we compare and analyze outlier detection performance in various experimental settings, focusing on text data with dimensions typically in the tens of thousands. Experimental setups were simulated to compare the performance of outlier detection methods in unsupervised versus semi-supervised mode and uni-modal versus multi-modal data distributions. The performance of outlier detection methods based on dimension reduction is compared, and a discussion on using k-NN distance in high dimensional data is also provided. Analysis through experimental comparison in various environments can provide insights into the application of outlier detection methods in high dimensional data.

Keywords: Curse of dimensionality, Dimension reduction, High dimensional text data, Outlier detection.

1 Introduction

An outlier is defined as an observation which deviates so much from other observations enough to arouse suspicions that it was generated by a different mechanism [1]. Outlier detection has been a hot research topic in recent years and has been applied to a variety of problems, such as fraud detection, intrusion detection in computer networks, system fault detection, and unexpected error detection in databases [2, 3, 4, 5, 6].

Outlier detection methods can be classified into three categories according to the learning environment. The first category is a supervised method that detects outliers by learning a binary classifier when training data consisting of normal data and outliers is given. However, there is a high probability of unbalanced learning where the amount of outliers and normal data is significantly different. The second category is unsupervised learning without data labels where it detects data samples that are highly likely to be outliers on the premise that most of the given data are normal and only a few outliers are included. The third category is semi-supervised learning, which detects whether the test data are normal or outliers given the training data consisting only of normal data. In situations where it is easy to collect data under normal conditions,

the approach of modeling the data distribution from the normal training data and detecting outliers in the test data based on it may be practical for some application problems.

On the other hand, according to computational methodologies employed in outlier detection methods, they can be roughly categorized to distance-based, density-based, tree-based, clustering-based, and neural network-based methods. A detailed survey of outlier detection methods can be found in several papers including [2, 7, 8]. The performance of the outlier detection method has been shown to be remarkable in many practical problems, but its application to high dimensional data is still difficult. In many cases, the experiments for high dimensional data were performed only for data with hundreds or fewer data dimensions, and not for data with more than tens of thousands of data dimensions such as text data [9, 10, 11].

Due to the curse of dimensionality in high dimensional space, the phenomenon referred to as data sparsity occurs where all pairs of data samples are almost equidistant in high dimensional data space [2]. This can be problematic in outlier detection methods which rely on distance computation such as in clustering or density estimation process. In this paper, we conduct comparative studies for outlier detection methods in high dimensional data. Focusing on text data with dimensions typically in the tens of thousands, the performance of outlier detection methods is compared in various experimental settings:

- Outlier detection in unsupervised mode of multi-modal normal data.
- Outlier detection in unsupervised mode of uni-modal normal data.
- Outlier detection in semi-supervised mode of multi-modal normal data.
- Outlier detection based on dimension reduction by feature selection or weighted feature combinations.

The remainder of the paper is organized as follows. In Section 2 we review outlier detection methods which have been applied in high dimensional data. In particular, a dimension reduction based outlier detection method is introduced [12]. Dimension reduction is performed by a transformation maximizing kurtosis which can be interpreted as the degree of presence of outliers in the distribution, and in the transformed space outlier detection is applied. In Section 3, using text data, experimental comparison and analysis are provided under various experimental setting of unsupervised or semi-supervised mode. The discussion follows in Section 4.

2 Outlier Detection methods

In this section, we review outlier detection methods that have shown good performance in various application problems.

2.1 Outlier Detection based on the Distance to k-Nearest Neighbors (KNN)

Distance-based outlier detection is simple and intuitive, but the outlier detection performance is often competitive to more complicated methods. In [13], the average or maximum of the distances to the k nearest neighbors is used as the outlier score of a data sample. The greater the distance to the k nearest neighbors, the more likely it is to be an outlier. Instead of computing outlier scores, a binary decision can be made that a data sample is determined as an outlier when less than k data samples lie within the radius R of the data sample [14].

The main challenge in distance-based methods is the scalability since distances between all pairs of data samples should be computed, and efforts to avoid high computational cost are being made. Partition-based pruning in [13] was used for speedup where data are first partitioned using a clustering algorithm and the partitions that cannot possibly contain the top n outliers are pruned. In [15], a sampling-based outlier detection method was proposed where a small set of samples are taken and an outlier score is measured by the distance from a data sample to its nearest neighbor in the sample set.

In various outlier detection methods, it is often necessary to calculate the distance between data samples during the process such as clustering or density estimation. However, in high dimensional space, the notion of distances may not work as in low dimensional data space. As the data dimension increases, the feature space becomes increasingly

sparse and the maximum and the minimum distance between the pairs of data samples become indiscernible compared to the minimum distance [16]. In the experiments using text data of Section 3, the performance of a distance-based outlier detection method is experimentally compared to other methods and the impact from the curse of dimensionality on outlier detection is evaluated.

2.2 Angle-based Outlier Detection (ABOD)

Angle-based outlier detection (ABOD) computes an outlier score using the variances of angles between the difference vectors to pairs of other data samples from a data sample [17]. The idea is motivated by the following intuition: For a data sample within a cluster, the angles between difference vectors to pairs of other points differ widely, but for outliers, the angles to the most pairs of data samples will be small since most data samples are clustered in some directions [17]. The angle-based outlier factor $ABOF(x)$ for a data sample x is computed as

$$ABOF(x) = variance_{z_1,z_2 \in D} \left(\frac{< \overline{xz_1}, \overline{xz_2} >}{\|\overline{xz_1}\|^2 \|\overline{xz_2}\|^2} \right), \tag{1}$$

where D is a given data set, $< \cdot, \cdot >$ denotes the scalar product, and $\overline{xz_i}$ is the difference vector $z_i - x$. The angle is weighted less if the corresponding data sample is far from the query data sample. In order to reduce the time complexity arising from dealing with all pairs of data samples for each data sample, FastABOD approximate ABOD by using only the pairs between k nearest neighbors instead of all pairs of data samples. In the experiments in Section 3, FastABOD was used.

2.3 Outlier Detection based on histograms (HBOS)

HBOS (histogram-based outlier score) assumes independence of the features which makes it fast at the cost of less precision [18]. For each feature, an univariate histogram is constructed by using static bin-width or dynamic bin-width histograms. The frequency of samples falling into each bin is used as an estimate of the density. After normalizing the histograms such that the maximum height is 1, the outlier score for a data sample x is calculated using the corresponding height of the bins where it is located such as

$$HBOS(x) = \sum_{i=1}^{d} log \left(\frac{1}{hist_i(x)} \right), \tag{2}$$

where d is the number of features.

2.4 Outlier Detection based on One-class SVM (OSVM)

Given a data set from an underlying probability distribution P, one-class SVM (Support vector machine) tries to estimate a simple subset S of input space such that the probability that a test point drawn from P lies outside of S is controlled by some pre-specified value ν between 0 and 1 [19]. It has been applied for outlier detection independently or in combination with other methods [20, 21].

Given data $\{x_1, \cdots, x_n\}$, one-class SVM maps the data into the feature space corresponding to the kernel and finds a hyperplane to separate them from the origin with maximum margin by solving the optimization problem

$$\begin{aligned} &\min_{w,\rho,\xi} \frac{1}{2}\|w\|^2 + \frac{1}{\nu n}\sum_i \xi_i - \rho \\ &subject\ to\ w \cdot \Phi(x_i) \geq \rho - \xi_i, \quad \xi_i \geq 0, \end{aligned} \tag{3}$$

where ξ_i's are slack variables penalizing data points on the negative side of a separating hyperplane $w \cdot \Phi(x) - \rho = 0$. ν is an upper bound on the fraction of data points outside the estimated region and also a lower bound on the fractions of support vectors [19].

2.5 Outlier Detection based on Local Outlier Factor (LOF)

LOF (Local Outlier Factor) measures the ratio of the peripheral density of a given data sample to that of neighboring data samples [22]. It is known that LOF works well when the regions of different densities exist. While local reachability density (lrd) of a data sample x is computed from the inverse of the average reachability distance to the k nearest neighbors of x, kNN(x), LOF is defined as

$$LOF(x) = \frac{\frac{1}{k}\sum_{z \in kNN(x)} lrd(z)}{lrd(x)}. \tag{4}$$

LOF provides an indication of whether x is in a denser or sparser region of the neighborhood than its neighbors [23, 6].

While LOF addresses with the problem of the local density variation, selecting the value for k is not trivial and the performance of LOF can be sensitive to the value of k. If groups of points might be close to one another by chance, a small k will increase the outlier scores of data samples in their locality [2]. On the other hand, a large k might cause to miss local outliers.

2.6 Outlier Detection using Isolation Forest (IF)

Among various outlier detection methods, Isolation Forest [24] is known to be computationally efficient and very effective in detecting outliers. Isolation Forest builds an ensemble of binary trees which are grown by randomly selecting a splitting feature and a random split value between the maximum and minimum values of the selected feature at each node. Under the premise that outliers are susceptible to isolation than normal data, outlier scores are computed by the average path length on Isolation trees. It has shown the competent outlier detection performance in various problems [25, 26, 27, 28].

Isolation Forest is a widely used outlier detection method, but it is difficult to apply to high dimensional data. Since tree height is limited by $ceiling(log_2\psi)$ for the sub-sampling size ψ, the total number of attributes which can be selected for node partitioning is also limited. Isolation Forest can also suffer from the sparse, irrelevant and noisy attributes in high dimensional data. In [24], the weighted selection of attributes by the kurtosis value was proposed for the application in high dimensional data. Kurtosis is a statistical measure for the thickness of the tail in the probability distribution of a real-valued random variable, which can be interpreted as the degree of presence of outliers in the distribution [29]. In [30] and [31], a method of partitioning data by a hyperplane with a random slope at each node of an isolation tree has been proposed. However, the hypothesis space of all hyperplanes with random slopes is too large in high dimensional space and the experiments were performed only for the data with the dimension below 40 and the performance for high dimensional data was not tested [31].

2.7 Outlier Detection based on Feature bagging (BLOF, BKNN)

Subspace outlier detection finds outliers in subspaces of the original data space. In [32], the subspace outlier score of a data sample is given by the degree of the deviation from the neighbors in an axis-parallel hyperplane spanned by the neighbors. Subspace outlier detection is often performed by combining outlier detection results in subspaces by random selection into an ensemble [33].

In ensemble construction of [33], the subsample size is always the same as the original input sample size, but the features are randomly sampled from half of the features to all features. The outlier score is computed by averaging or taking the maximum of all base detectors. In [33], LOF is used as the base outlier detection method. However, any detector such as KNN could be used as the base detector. In the experiments in Section 3, we test feature bagging based on LOF and KNN, denoted as BLOF and BKNN, respectively.

2.8 Outlier Detection based on Principal Component Analysis (PCA)

Principal component analysis (PCA) is a traditional linear dimension reduction method where the projection into the directions with the largest variance in data is pursued [34]. The covariance matrix of the data is decomposed to orthogonal vectors, called eigenvectors, associated with eigenvalues, and the eigenvectors with high eigenvalues which capture most of the variance in the data are used for dimension reduction. However, when it comes to outlier detection, outliers and normal data samples can be better distinguished in the hyperplane of eigenvectors with small eigenvalues.

In [35] and [2], an outlier score is computed by the weighted sum of the projected distance of a data sample to the centroid along the direction of eigenvectors as follows:

$$Score(x) = \sum_{i=1}^{d} \frac{|(x-\mu)\cdot e_i|^2}{\lambda_i}, \tag{5}$$

where λ_i is an eigenvalue corresponding to an eigenvector e_i. d usually denotes data dimension, but when the number of data samples is smaller than data dimension such as in high dimensional

text data, d can be set as the minimum value among the number of data samples and data dimension.

2.9 Outlier Detection based on AutoEncoder (AE)

Auto Encoder (AE) is a type of neural networks for learning useful data representations in an unsupervised manner. An autoencoder consists of two parts, the encoder ϕ and the decoder ψ, and it is trained so that the reconstruction error of data instances in a training set X

$$L = \sum_{x \in X} \|x - \psi(\phi(x))\|^2 \tag{6}$$

is minimized. The larger the reconstruction error of a test instance is, the greater the degree of its outlierness is [2, 6].

2.10 Outlier Detection based on dimension Reduction maximizing kurtosis (IF/DR)

The kurtosis is the fourth standardized moment of univariate random variable X, defined as

$$kurtosis(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] = \frac{\mu_4}{\sigma^4}, \tag{7}$$

where μ_4 is the fourth central moment and σ is the standard deviation. The meaning of kurtosis can be interpreted that data within one standard deviation of the mean contribute virtually little to kurtosis, since raising a number that is less than 1 to the fourth power makes it closer to zero. The only data values that contribute to kurtosis in any meaningful way are those outside the region of the peak, i.e., the outliers [36]. Kurtosis reflects the shape of a distribution and high kurtosis value means heavy tails than normal. High values of kurtosis can arise in the circumstances where the probability mass is concentrated in the tails of the distribution. In [37], $2p$ orthogonal directions maximizing or minimizing the kurtosis value are obtained by eigenvector computation, and in the projected space by the orthogonal directions outlier scores using a univariate measure of outlyingness are used for outlier detection. However, the experiments were performed only for very small data sets with the dimension below 5 and simulated data with the dimension below 20.

Recently an outlier detection method based on dimension reduction was introduced [12], where new features maximizing kurtosis are extracted by a transformation from the original feature space and in the transformed feature space Isolation Forest is modeled. Feature extraction by a transformation which maximizes kurtosis can be performed using a neural network with no hidden layers. Denoting the weight on the edge connecting the node j of an input layer and the node i ($1 \le i \le k$) of an output layer as w_{ij}, kurtosis on the output node i can be computed for a given input data $\{\mathbf{x_1}, \cdots, \mathbf{x_n} \,|\, \mathbf{x_j} \in R^d\}$ as

$$kurtosis_i = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{\mathbf{w_i x_j} + b_i - \mu_i}{\sigma_i}\right)^4, \tag{8}$$

where $\mathbf{w_i} = [w_{i1}, \cdots, w_{id}]$, and μ_i and σ_i are the mean and standard deviation of $\{\mathbf{w_i x_j} + b_i \,|\, 1 \le j \le n\}$ which is the mapping of $\mathbf{x_j}$'s to the node i of the output layer. Using the standardization $z_{ij} = \frac{\mathbf{w_i x_j} + b_i - \mu_i}{\sigma_i}$ and applying the activation function f on each output node, the objective function can be set as

$$minimize \frac{1}{kn}\sum_{i=1}^{k}\sum_{j=1}^{n} f(-z_{ij}^4). \tag{9}$$

In the implementation by a mini-batch stochastic method, batch normalization can be applied in place of the standardization process $z_{ij} = \frac{\mathbf{w_i x_j} + b_i - \mu_i}{\sigma_i}$. Also instead of using all the original features as input features, a subset of features with high kurtosis can be used as input features of the neural network. In the transformed data which is the output of the neural network, Isolation Forest is applied for outlier detection. We denote this method as IF/DR. The process for IF/DR can be summarized as follows:

1. Select s features with the highest kurtosis in the original feature space.
2. Construct a neural network with s input nodes and k output nodes.
3. Train the neural network to optimize the objective function in Eq. (9) by applying batch normalization.
4. Compute the transformed representation of data samples by the outputs of the trained neural network.
5. Perform Isolation Forest in the transformed data space.

Table 1. The description of text data sets used for performance comparison.

Data	classes	number of features (nf)	number of samples (ns)
bbc	5	17005	2225
reuter	3	15484	6656
20-ng	20	44713	18774
la12	6	21604	6279
sports	5	18324	8313
classic	4	12009	7094
ohscal	10	11465	11162
reviews	4	23220	3932

3 Experimental Comparison

In this section, we perform an experimental comparison for outlier detection methods reviewed in Section 2 in various experimental setups. Also a discussion on using k-NN distances in high dimensional data is provided.

3.1 Data Description

Eight text data sets were used for performance comparison, and detailed description is given in Table 1. The BBC News data consists of 2,225 news data which belong to five categories: business, entertainment, politics, sport, and tech [38]. It was preprocessed to have 17,005 terms by deleting special symbols and numbers and removing terms appearing in only one document. Reuters-21578 was downloaded from UCI machine learning repository and the documents belonging to 135 TOPICS categories were used. After preprocessing by stopwords removal, stemming, tf-idf transformation, and unit norm, and excluding documents belonging to two or more categories, there are 6,656 documents composed of 15,484 terms. The two largest categories of 1 and 36 and the collection of the remaining all the documents compose three classes. 20-newsgroup (20-ng) data contains about 20,000 articles in 20 news groups divided into 5 categories [1]. After preprocessing the 20news-bydate version, we constructed 18,774 text data with 44,713 terms. The remaining five data sets were downloaded from the site [2]. The final data sets were constructed removing classes with less than 200 texts and terms with frequencies less than or equal to 1.

3.2 Parameter setting for outlier detection methods

Parameter setting of the compared methods is summarized in Table 2. Most of the methods were implemented using PyOD [39] which is a python toolkit for outlier detection and all parameters were set as default values in PyOD with few exception. For example, in outlier detection based on autoencoders, the mini-batch size was set to 256 instead of the default of 32, because of the frequent interruptions caused by division-by-zero occurrences at batch size 32. The method, IF/DR, was implemented by PyTorch [40] and the parameter values to use as default values for all text data sets were determined by preliminary experiments on BBC data.

3.3 Unsupervised Mode: Multi-class Normal Data

The first experiment was to test the performance of outlier detection methods in unsupervised mode where a small percent of outliers are mixed with normal data with their true labels unknown. In particular, normal data were randomly selected from multiple classes to simulate a multi-modal distribution for normal data. For each data set in Table 1, 10% of data from one class was randomly selected as outliers, and data of all other classes were set as normal data. The performance of the outlier detection method was measured using Area Under the Curve (AUC), and the average AUC was computed by repeating the experiment about 20 times while using each class as an outlier class the same number of times. In 20-newsgroup data, the experiment was performed while repeating 20 times of setting all

[1] http://people.csail.mit.edu/jrennie/20Newsgroups

[2] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download

Table 2. Parameter setting in outlier detection methods

Methods	Parameters	Values
KNN	number of neighbors for k neighbors queries	k=5
	metric used for distance computation	Euclidean
	outlier score: the distance to the k-th neighbor	
ABOD	number of neighbors to use for k neighbors queries	k=10
HBOS	number of bins	10
OSVM	RBF kernel function: exp(-$\gamma\|\|x_1 - x_2\|\|^2$)	γ = 1/nf
	an upper bound on the fraction of training errors	ν =0.5
LOF	number of neighbors for local density estimation	k=20
	metric used for distance computation	Euclidean
IF	number of Isolation trees in the ensemble	100
	sub-sampling size for each Isolation tree	min(256, ns)
PCA	number of principal components to keep	min(ns, nf)
	standardization preprocessed	
BLOF	the base detector	LOF or KNN
BKNN	number of base estimators in the ensemble	10
	outlier score by the average of all detectors	
AE	the number of neurons per layers	[nf, 64, 32, 32, 64, nf]
	activation function in hidden layers	relu
	number of epochs to train the model	100
	mini batch size	256
	the percentage of data to be used for validation	0.1
	batch normalization, Adam optimizer, dropout rate=0.2	
	L2 regularizer of 0.1, standardization preprocessed	
IF/DR	input features: s features with the highest kurtosis	s = 0.15*nf
	number of nodes in the output layer	100
	activation function in the output layer	sigmoid
	number of epochs to train the model	10
	mini batch size	200
	batch normalization	
	Adam optimizer with learning rate 0.01, dropout rate=0.5	

(nf: number of features, ns: number of data samples)

Table 3. The performance comparison of outlier detection methods in unsupervised mode of multi-class normal and one-class outlier.

	KNN	ABOD	HBOS	OSVM	LOF	IF	BLOF	BKNN	PCA	AE	IF/DR
bbc	0.84	0.721	0.557	0.738	0.707	0.533	0.753	0.703	0.852	**0.899**	0.867
20-ng	0.791	0.774	0.525	0.653	0.607	0.483	0.759	0.782	0.802	**0.824**	**0.82**
reuter	0.66	0.619	0.522	**0.752**	0.569	0.539	0.617	0.619	0.613	0.611	0.565
la12	**0.749**	0.647	0.519	0.682	0.639	0.5	0.7	0.657	0.704	0.728	0.709
sports	0.811	0.687	0.568	0.767	0.511	0.525	0.801	0.739	**0.833**	**0.83**	0.791
classic	0.591	0.491	0.596	0.748	**0.781**	0.556	0.693	0.614	0.702	0.742	**0.782**
ohscal	0.645	0.57	0.529	0.599	0.617	0.514	0.633	0.572	**0.669**	**0.665**	0.626
reviews	0.704	0.651	0.537	0.716	0.513	0.532	0.673	0.634	0.707	0.739	**0.75**
average	0.724	0.645	0.544	0.707	0.618	0.523	0.704	0.665	0.735	**0.755**	0.739

Table 4. The performance comparison of outlier detection methods in unsupervised mode of one-class normal and one-class outlier.

	KNN	ABOD	HBOS	OSVM	LOF	IF	FLOF	BKNN	PCA	AE	IF/DR
bbc	0.91	0.795	0.591	**0.955**	0.754	0.613	0.828	0.841	0.897	0.903	0.937
20-ng	0.835	0.821	0.535	0.748	0.708	0.54	0.8	0.822	0.821	0.854	**0.88**
reuter	0.687	0.643	0.533	**0.844**	0.533	0.571	0.642	0.65	0.648	0.638	0.626
la12	0.854	0.74	0.565	**0.904**	0.666	0.566	0.784	0.751	0.737	0.746	0.794
sports	0.88	0.751	0.586	**0.92**	0.62	0.594	0.85	0.845	0.817	0.801	0.82
classic	0.853	0.761	0.535	**0.9**	0.785	0.542	0.855	0.773	0.723	0.758	0.847
ohscal	0.853	0.742	0.601	**0.891**	0.744	0.568	0.766	0.733	0.843	0.822	0.777
reviews	0.799	0.693	0.542	**0.888**	0.482	0.573	0.702	0.753	0.766	0.767	0.809
average	0.834	0.743	0.561	**0.881**	0.662	0.571	0.778	0.771	0.782	0.786	0.811

data in one category as normal data and randomly selecting one class from the other category as an outlier class.

Table 3 summarizes the average AUC for the compared methods. Three methods, AE, IF/DR, and PCA, obtained the higher performance than other methods. One common characteristic in those methods is that they utilize a weighted combination of features. One of the differences between AE and IF/DR is that IF/DR trains a simple neural network with no hidden layers, whereas AE models an encoder and decoder with two hidden layers. Also, the large performance difference between IF and IF/DR shows that kurtosis and neural network-based feature extraction can construct a transformed space which is effective in constructing an isolation forest.

3.4 Unsupervised Mode: One-class Normal Data

While the experiment in the previous section simulated the environment where normal data are drawn from multi-modal distribution, this experiment simulates the case when the normal data come from uni-modal distribution. For each data set in Table 1, 5% among data from one class was randomly selected as outliers, and all data of one class among remaining classes was set as normal data. The average AUC was measured by repeating the experiment for every pair of classes. In 20-newsgroup data, the experiment was repeated 20 times with random selection of normal and outlier classes, respectively, from each pair of categories.

Table 4 summarizes the average AUC for the compared methods. For each data set, the highest AUC is marked as a bold face. Unlike in the experiment of multi-class normal and one-class outlier, OSVM outperforms the other methods significantly and KNN showed the second best performance. It demonstrates that the outlier detection by OSVM can be a good choice when the normal data are considered to follow uni-modal distribution. The ensemble of one-class SVMs combined with a clustering algorithm has been applied for outlier detection or classification and showed better performance than single one-class SVM [41, 42]. Hence, for data with multi-modal distribution, we can expect the utility of one-class SVMs combined with a clustering method that can work well for high-dimensional data.

3.5 Semi-supervised Mode: When Multi-class Normal Data is Given as Training Data

Unsupervised outlier detection assumes that no class label is provided and that a small fraction of outliers may exist in the given data. However, since it is relatively easy to collect normal data compared to outliers, it may be practical to perform outlier detection when training data consisting only of normal data is given. In this experiment, one class was set as an outlier class and the remaining classes were set as normal classes. Of the data from each normal class, 50% was used as training data. The remaining data samples of normal classes constituted the test set together with data 50% from the outlier class. An outlier detection model is learned using normal training data and an outlier score on test data is computed by applying the model. The AUC (Area Under the Curve) is computed based on out-

Table 5. The performance comparison of outlier detection methods in semi-supervised mode.

	KNN	ABOD	HBOS	OSVM	LOF	IF	BLOF	BKNN	PCA	AE
bbc	**0.917**	0.69	0.501	0.74	0.886	0.484	0.744	0.253	0.734	0.706
20-ng	**0.822**	0.726	0.462	0.672	0.778	0.462	0.79	0.465	0.706	0.693
reuter	**0.872**	0.728	0.52	0.867	0.671	0.559	0.734	0.574	0.767	0.739
la12	**0.804**	0.64	0.507	0.674	0.766	0.515	0.693	0.423	0.665	0.642
sports	**0.946**	0.671	0.523	0.833	0.87	0.482	0.861	0.331	0.851	0.813
classic	0.885	0.424	0.539	0.795	**0.902**	0.511	0.75	0.286	0.669	0.647
ohscal	**0.659**	0.575	0.519	0.611	0.642	0.497	0.615	0.487	0.601	0.586
reviews	**0.775**	0.631	0.535	0.739	0.719	0.524	0.704	0.417	0.695	0.667
average	**0.835**	0.636	0.513	0.741	0.779	0.504	0.736	0.405	0.711	0.687

lier scores of test data. This process was repeated about 20 times while changing the outlier class.

Table 5 summarizes the average AUC for the compared methods. When the experimental results are analyzed, it should be considered that some of the methods are more suitable for outlier detection in unsupervised mode. Nevertheless, the highest detection performance by KNN in semi-supervised mode is surprising, considering the phenomenon by the curse of dimensionality in high dimensional space. In the next section, we check the validity of using k-NN distances for outlier detection in high-dimensional data.

3.6 Discussion on Using k-NN Distance in High Dimensional Space

It is known that due to data sparsity all pairs of data samples are almost equidistant in high dimensional data space and the difference between the maximum distance and minimum distance compared to the minimum distance vanishes as the dimensionality increases. However, as shown in the experiments of previous sections, outlier detection based on the distance to k nearest neighbors showed the performance of the high rank among the compared methods, especially in semi-supervised mode where normal training data are given.

We try to explain the reason why k nearest neighbors-based outlier detection worked competently in spite of the phenomenon from the curse of dimensionality. The research in [16] has shown that the concentration effect of the distance measure only holds in the artificial scenario when the one-dimensional distributions are independent and identically distributed, and the curse of dimensionality is not the main problem for outlier detection in high dimensional data. We conducted the experiment to compare the distance from an outlier or normal data to other data samples using text data in Table 1.

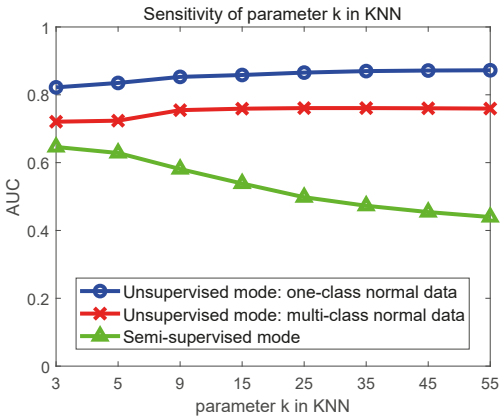

Figure 2. Comparison of outlier detection performance by KNN while changing the value of k. The average AUC in three experimental setups is shown.

Two figures in the top row of Figure 1 compare the maximum and minimum distances from normal or outliers of the test data set to normal data samples of the training set in semi-supervised mode of Section 3.5, and two figures in the bottom row compare the maximum and minimum distances between data samples in unsupervised mode of Section 3.3. The figures show that the average minimum distance from outliers is greater than that from normal data samples. On the other hand, the average maximum distance from outliers is almost equal to or smaller than that from normal data samples, implying that the distance concentration effect is stronger in outliers than in normal data samples. Hence, the distance to the nearest neighbors in high dimensional data can be used effectively for the discrimination

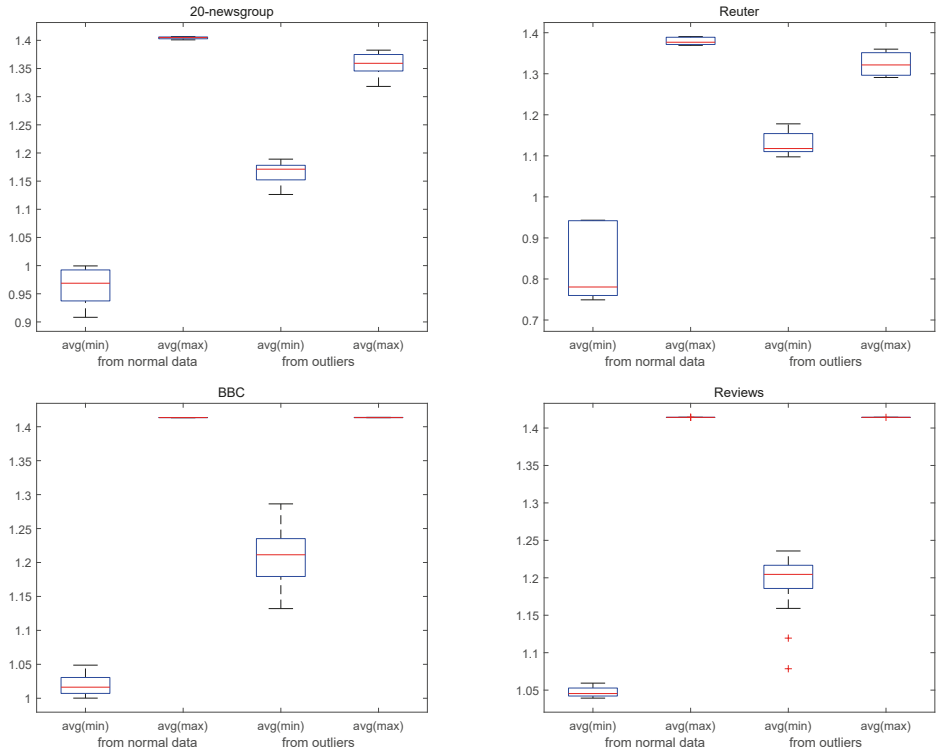

Figure 1. Comparison of the maximum and minimum distances from normal or outliers to other data samples. top row: measured in semi-supervised mode, bottom row: unsupervised mode

of outliers and normal data samples with the careful selection of the value k in k-NN search.

Next, to test the sensitivity of the parameter k in KNN, we compared outlier detection performance by KNN while changing the value of k. Figure 2 shows the average AUC by the KNN method in three experimental setups. As shown in Figure 2, in the unsupervised mode, stable performance was observed in the range of 25 to 45, while in the semi-supervised mode, high performance was obtained at small k values.

3.7 Running Time Comparison of Outlier Detection Methods

We measured the running time when performing the outlier detection method in unsupervised mode of Section 3.3. The computer used had a CPU Intel i9-9900X(3.50GHz), RAM 32GB. Figure 3(a) shows the measured CPU time in seconds while running on bbc and 20-newsgroup data respectively. The methods based on one-class SVM, PCA, and feature bagging showed a relatively high execution time compared to the other methods. Figure 3(b) shows the average AUC values by outlier detection methods copied from Table 3, 4, 5.

4 Discussions

In this paper, a comparative study for outlier detection methods in high dimensional data was performed and experimental results using text data were analyzed. In particular, experimental setups were simulated to compare the performance of outlier detection methods in unsupervised versus semi-supervised mode and uni-modal versus multi-modal data distributions. Experimental results can be summarized as follows:

– Outlier detection methods utilizing feature transformation such as autoencoder, PCA, or kurtosis-based dimension reduction achieved the highest performance in the unsupervised mode when normal data consisted of multiple classes. However, in the semi-supervised mode where the class label of normal data is given, outlier detection methods such as KNN, LOF, or one-class SVM were better than AE or PCA-based methods.

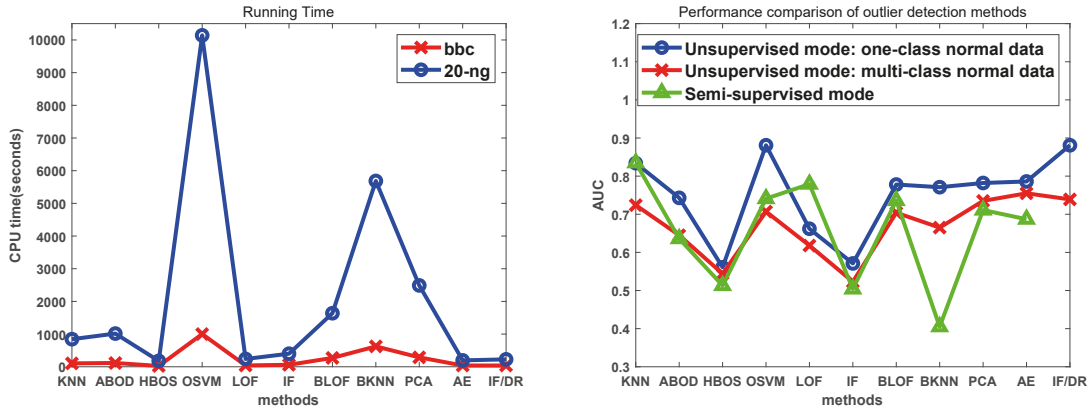

Figure 3. (a) Comparison of CPU time in seconds while running on bbc and 20-newsgroup data. (b) Performance comparison of outlier detection methods in three experimental settings by the average AUC.

– In unsupervised mode, bagging by feature selection using the base detector LOF achieved higher performance than using the single detector LOF, but it did not show the best performance among the compared methods. It is presumed to be caused by the characteristic that text data has many zero components.

– Outlier detection based on one-class SVM showed significantly higher performance when normal data consisted of one class. On the other hand, on multi-class normal data, the performance was lower than that of the KNN-based method.

– A dimension reduction method was introduced that maximizes kurtosis, which can be implemented using a simple neural network with no hidden layers. Experimental results have proven that the performance of the Isolation Forest built in a dimension reduced space is greatly improved.

– Outlier detection based on distance to k nearest neighbors worked well despite the curse of dimensionality in high dimensional space. Especially, it was prominent in semi-supervised mode where normal training data is given. The feasibility of using k-NN distances for outlier detection in high-dimensional data was experimentally examined.

Experimental comparison has the limitation that it requires optimization of parameter values for each method and data set. However, parameter optimization is not easy unless a validation set of outliers is not provided. Instead, for all the data sets we used default parameter values recommended in the PyOD package with very little exceptional cases. Regardless of that limitation, consistent findings can provide insight into the application of outlier detection methods in high dimensional text data.

Acknowledgments

This work was partly supported by Institute of Information & communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2022-00155857, Artificial Intelligence Convergence Innovation Human Resources Development (Chungnam National University)) and research fund of Chungnam National University.

References

[1] D. Hawkins. Identification of Outliers. Chapman and Hall, 1980.

[2] C. Aggarwal. Outlier analysis (2nd ed.) Springer, 2017.

[3] Caroline Cynthia and Thomas George. An outlier detection approach on credit card fraud detection using machine learning: A comparative analysis on supervised and unsupervised learning. In: Peter J., Fernandes S., Alavi A. (eds) Intelligence in Big Data Technologies-Beyond the Hype. Advances in Intelligent Systems and Computing, 1167, 2021.

[4] H. Mazzawi, G. Dalai, D. Rozenblat, L. Ein-Dor, M. Ninio, O. Lavi, A. Adir, E. Aharoni, and E. Ker-

many. Anomaly detection in large databases using behavioral patterning. In ICDE, 2017.

[5] T. Li, J. Ma, and C. Sun. Dlog: diagnosing router events with syslogs for anomaly detection. The Journal of Supercomputing, 74(2):845–867, 2018.

[6] C. Park. Outlier and anomaly pattern detection on data streams. The journal of supercomputing, 75:6118–6128, 2019.

[7] H. Wang, M. Bah, and M. Hammad. Progress in outlier detection techniques: A survey. IEEE Access, 7, 2019.

[8] A. Boukerche, L. Zheng, and O. Alfandi. Outlier detection: Methods, models, and classification. ACM Computing Surveys, 53:1–37, 2020.

[9] X. Zhao, J. Zhang, and X. Qin. Loma: A local outlier mining algorithm based on attribute relevance analysis. Expert Systems with Applications, 84, 2017.

[10] X. Zhao, J. Zhang, X. Qin, J. Cai, and Y. Ma. Parallel mining of contextual outlier using sparse subspace. Expert Systems with Applications, 126, 2019.

[11] F. Kamalov and H. Leung. Outlier detection in high dimensional data. Journal of Information and Knowledge Management, 19, 2020.

[12] C. Park. A dimension reduction method for unsupervised outlier detection in high dimensional data(written in korean). Journal of KIISE. In press.

[13] S. Damaswanny, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In Proceeding of ACM SIGMOD, pages 427–438, 2000.

[14] E. Knorr and R. Ng. Finding intensional knowledge of distance-based outliers. In Proceeding of 25th International Conference on Very Large Databases, 1999.

[15] M. Sugiyama and K. Borgwardt. Rapid distance-based outlier detection via sampling. In International Conference on Neural Information Processing Systems, 2013.

[16] A. Zimek, E. Schubert, and H. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining, 5:363–387, 2012.

[17] H. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In Proceeding of KDD, pages 444–452, 2008.

[18] M. Goldstein and A. Dengel. Histogram-based outlier score (hbos): a fast unsupervised anomaly detection algorithm. In Proceeding of KI, pages 59–63, 2012.

[19] B. Scholkopf, J. Platt, J. Shawe-Taylor, and A. Smola. Estimating the support of a high-dimensional distribution. Neural computation, pages 1443–1471, 2001.

[20] M. Amer, M. Goldstein, and S. Abdennadher. Enhancing one-class support vector machines for unsupervised anomaly detection. In Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, 2013.

[21] L. Ruff, R. Vandermeulen, N. Gornitz, L. Deecke, S. Siddiqui, A. Binder, E. Muller, and M. Kloft. Deep one-class classification. In Proceeding of international conference on machine learning, 2018.

[22] M. Breunig, H. Kriegel, R. Ng, and J. Sander. Lof: Identifying density-based local outliers. In Proceeding of the ACM Sigmod International Conference on Management of Data, 2000.

[23] P. Tan, M. Steinbach, and V. Kumar. Introduction to Data Mining. Addison Wesley, Boston, 2006.

[24] F. Liu, K. Ting, and Z. Zhou. Isolation forest. In Proceedings of the 8th international conference on data mining, 2008.

[25] G. Susto, A. Beghi, and S. McLoone. Anomaly detection through on-line isolation forest: An application to plasma etching. In the 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), pages 89–94, 2017.

[26] L. Puggini and S. MCLoone. An enhanced variable selection and isolation forest based methodology for anomaly detection with oes data. Engineering Applications of Artificial Intelligence, 67:126–135, 2018.

[27] J. Kim, H. Naganathan, S. Moon, W. Chong, and S. Ariaratnam. Applications of clustering and isolation forest techniques in real-time building energy-consumption data: Application to leed certified buildings. Journal of energy Engineering, 143, 2017.

[28] J. Hofmockel and E. Sax. Isolation forest for anomaly detection in raw vehicle sensor data. In the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018), pages 411–416, 2018.

[29] J. Livesey. Kurtosis provides a good omnibus test for outliers in small samples. Clinical Biochemistry, 40:1032–1036, 2007.

[30] F. Liu, K. Ting, and Z. Zhou. On detecting clustered anomalies using sciforest. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2010.

[31] S. Hariri, M. Kind, and R. Brunner. Extended isolation forest. IEEE transactions on knowledge and data engineering, 33:1479–1489, 2021.

[32] H. Kriegel, P. Kroger, E. Schubert, and A. Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In Proceedings of PAKDD, 2009.

[33] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In Proceedings of KDD, 2005.

[34] R. Duda, P. Hart, and D. Stork. Pattern classification (2nd ed.). Wiley-interscience, 2000.

[35] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. In Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, 2003.

[36] P. Westfall. Kurtosis as peakedness, 1905-2014. r.i.p. The American Statistician, 68(3):191–195, 2014.

[37] D. Pena and F. Prieto. Multivariate outlier detection and robust covariance matrix estimation. Technometrics, 43:286–310, 2001.

[38] D. Greene and P. Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In Proceeding of ICML, 2006.

[39] Y. Zhao, Z. Nasrullah, and Z. Li. Pyod: A python toolbox for scalable outlier detection. Journal of Machine Learning Research, 20:1–7, 2019.

[40] A. Paszke, S. Gross, F. Massa, and et. al A. Lerer. Pytorch: An imperative style, high-performance deep learning library. In NeurIPS, pages 8026–8037, 2019.

[41] L. Abdallah, M. Badarna, W. Khalifa, and M. Yousef. Multikoc: Multi-one-class classifier based k-means clustering. Algorithms, 14(5):1–10, 2021.

[42] B. Krawczyk, M. Wozniak, and B. Cyganek. Clustering-based ensemble for one-class classification. Information sciences, 264:182–195, 2014.

Cheong Hee Park received her Ph.D. in Mathematics from Yonsei University, Korea in 1998. She received the M.S. and Ph.D. degrees in Computer Science at the Department of Computer Science and Engineering, University of Minnesota in 2002 and 2004 respectively. She is currently in the Department of Computer Science and Engineering, Chungnam National University, Korea as a professor. Her research interests include machine learning, data mining, and pattern recognition.
https://orcid.org/0000-0002-8233-2206