Local Kernel Density Ratio-Based Feature Selection for Outlier Detection

Fatemeh Azmandian FAZMANDI@ECE.NEU.EDU
Jennifer G. Dy JDY@ECE.NEU.EDU
Javed A. Aslam JAA@CCS.NEU.EDU
David R. Kaeli KAELI@ECE.NEU.EDU
Northeastern University, 360 Huntington Ave, Boston, MA 02115 USA

Editor: Steven C.H. Hoi and Wray Buntine

Abstract

Selecting features is an important step of any machine learning task, though most of the focus has been to choose features relevant for classification and regression. In this work, we present a novel non-parametric evaluation criterion for filter-based feature selection which enhances outlier detection. Our proposed method seeks the subset of features that represents the inherent characteristics of the normal dataset while forcing outliers to stand out, making them more easily distinguished by outlier detection algorithms. Experimental results on real datasets show the advantage of this feature selection algorithm compared to popular and state-of-the-art methods. We also show that the proposed algorithm is able to overcome the small sample space problem and perform well on highly imbalanced datasets.

Keywords: Feature Selection, Outlier Detection, Anomaly Detection, Imbalanced Data

1. Introduction

An integral part of any machine learning task is having a good set of features that can be used to accurately model the inherent characteristics of the data. In practice, the best set of features is not known in advance. Therefore, a pool of candidate features are collected and processed to removed irrelevant and redundant features. This can improve both the memory and computational cost of the machine learning algorithm, as well as the accuracy of the learner. Reducing the space of possible features is done in two ways: feature transformation and feature (subset) selection. In the former, the original space of features is transformed into a new, typically lower-dimensional, feature space. Examples include Principal Components Analysis (PCA), Kernel PCA (Schölkopf et al., 1998), and Kernel Dimensionality Reduction (Fukumizu et al., 2004).

In the latter approach to reducing the size of the feature space, the original set of features remain unchanged and a subset of those features are selected. A simple way to perform feature selection is to use a feature evaluation function, such as relevance (Kira and Rendell, 1992), to rank the features on an *individual* basis. Then, a subset of the features are selected by taking the top-ranked features (for example, the top m features). Another feature selection methodology is to search the space of feature subsets and select the subset that optimizes a criterion function. For a dataset with d features, there are 2^d possible subsets of features. For even a moderate value of d, an exhaustive search would be too computationally expensive, so it is common to use a greedy search strategy such as sequential forward selection or backward elimination (Liu and Motoda, 1998).

In addition to the search strategy, the other important component of feature selection is the criterion to be optimized. A brute force way to evaluate features is to utilize the classifier that will ultimately be used. Such a method is called a wrapper approach (Kohavi and John, 1997). Another, less computationally expensive, way is to evaluate features based on some criterion function. This is referred to as a filter method. Existing criteria include measures based on distance, information, dependency, and consistency (Dash and Liu, 1997). A third approach is to perform feature selection as part of the learning task, known as an embedded method, such as a decision tree.

In this work, we present a novel optimization criterion inspired by outlier detection problems where the data is highly imbalanced and outliers comprise a small portion of the dataset. *The criterion tries to find the set of features that maximizes the density*[1] *of normal data points while minimizing the density of outliers.* In other words, it seeks feature subsets wherein normal data points fall in high-density regions and outliers fall in low-density regions of the feature space. The goal is to make outliers stand out more prominently from normal data points, which allows outlier detection algorithms to more easily identify them.

Most of the work on dimensionality reduction for outlier detection have tackled the problem using a feature transformation approach. In these methods, a projection of the feature space is sought which can be used to detect outliers. While this approach subsumes feature selection (i.e., projecting onto the original feature axes is equivalent to selecting a subset of the features), there is a case to be made for the understandability that comes with feature subset selection as opposed to linear or non-linear combinations of the features. Retaining a subset of the original features goes a long way towards understanding the nature of the underlying data and the features that contribute to an outlier's deviant behavior. This can be advantageous in domains such as fraud detection and intrusion detection, in which case anomalous activity can be narrowed down to a subset of the collected features.

The remainder of the paper is organized as follows. In section 2, we discuss related work and in section 3, we describe the details of our proposed local kernel density ratio feature selection algorithm. In section 4, we present the results of our feature selection algorithm on several real-word datasets. Finally, section 5 concludes the paper and presents directions for future work.

2. Related Work

Feature selection is a very important and well-studied problem in machine learning. Most of the work have focused on the area of feature selection for classification and regression (Dash and Liu, 1997; Kohavi and John, 1997; Guyon and Elisseeff, 2003; Tibshirani, 1994; Song et al., 2007), and little has been done to create feature selection algorithms that cater to outlier detection problems. Chen and Wasikowski (2008) developed a ranking-based feature selection algorithm for classification of high-dimensional datasets that suffer from the "small sample space" problem and whose class labels are highly imbalanced, the latter being a characteristic inherent in outlier detection.

[1] Here, the term *density* refers to how dense and closely situated the data points are in the feature space, not to be confused with the probability density function.

Recent work that look for outliers in high-dimensional datasets deal with the issue of high dimensionality in different ways. Aggarwal and Yu (2005) use an evolutionary search technique to find multiple lower dimensional projections of the data which are locally sparse in order to detect outliers. Other methods perform feature transformation, rather than feature selection, for outlier detection (Nguyen and Gopalkrishnan, 2010).

There has also been some work done that use the idea of "a ratio of densities" to directly perform outlier detection. Hido et al. (2011) use the ratio of the density of a data point in the training set to its density in the test set as a measure of the degree to which the point is an *inlier*, as opposed to an outlier. Their training set consists of only normal points and the test set consists of both normal points and outliers. To deal with high-dimensional data, Sugiyama et al. (2011) use a projection matrix to find the low-dimensional subspace in which the two densities are significantly different from each other. In (Smola et al., 2009), the novelty of data points in one distribution are assessed relative to another distribution based on the log-likelihood ratio of the two distributions. In our work, we use a ratio of densities to perform feature selection, with the distinction that our method uses the notion of *local* neighborhoods to measure densities. In the denominator, we utilize the density of only outliers. This ensures that we pick features in which outliers become even more conspicuous as they will be represented in low-density regions of the feature space. For the outlier detection, we take a one-class learning approach and distinguish outliers using well-established methods. In this regard, once features are chosen by our feature selection technique, the outlier detection algorithms proposed in (Sugiyama et al., 2011) and (Smola et al., 2009) can be used as alternative methods of identifying outliers. In the next section, we present the details of our feature selection algorithm.

3. Local Kernel Density Ratio Feature Selection

The inspiration for our feature selection algorithm came from an approach taken to solve the outlier (or anomaly) detection problem. Hawkins (1980) describes an outlier as "an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism." In outlier detection problems, outliers typically comprise a small portion of the dataset. Examples include intrusion detection data, where malicious attacks are less frequent than normal activity, and certain tumor datasets where patients with a particular type of cancer are few compared to the number of healthy patients. In some cases, obtaining samples from the outlier class may be difficult or expensive, such as a fault detection system where it can be expensive to obtain outlier data representing the faulty situations of a machine. Therefore in the outlier detection domain, learners must deal with highly imbalanced data. Next, we describe our criterion function for feature selection which caters to outlier detection problems and is insensitive to the degree of imbalance in the data as it is based on a ratio of *average* normal to outlier densities.

3.1. Local Kernel Density Ratio Criterion

In this work, we propose a novel feature selection criterion for outlier detection which tries to find the set of features that best describes the normal class while ensuring that outliers "stand out". While most outlier detection techniques take an unsupervised approach, it is not uncommon to have samples that belong to the outlier class. For example, in intrusion detection there may be many instances of malware attacks and malicious executions that can provide guidance as to how normal executions differ from malicious ones. In our case, we utilize the supervised information to select features which are intrinsically suitable for outlier detection. Normal data points come from the same distribution while outliers can be any point, from a completely different distribution. Taking advantage of information from both normal and outlier points (if available) is more powerful than normal alone. The aim is to have something to compare the normal distribution against and find out what separates outliers from the normal data. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$ represent a dataset with n data points where each data point $\mathbf{x} \in \mathbb{R}^d$ has d features and is associated with a class label $y \in \{-1, +1\}$, where -1 denotes the normal class and $+1$ denotes an outlier.

The Local Outlier Factor (LOF) algorithm (Breunig et al., 2000) solves the outlier detection problem using a ratio of densities. In the LOF algorithm, the density of a data point is compared to that of its neighbors and based on this, the point is assigned a *degree* of being an outlier, known as its *local outlier factor.* The LOF of a data point is calculated as the average density of data points within its neighborhood divided by its own density. When a data point has a low density compared to points in its neighborhood, it is more likely to be an outlier. Conversely, outliers should have a lower density compared to its neighbors. Therefore, it stands to reason that a feature set which emphasizes this phenomenon would facilitate the detection of outliers.

To test this hypothesis, we developed a criterion that measures the quality of features based on the density induced for normal and outlier data points. More specifically, to maximize the density of normal data points while minimizing the density of outliers, the criterion function takes the ratio of the two, with a focus on the *local* neighborhood density of each data point. To measure density, we make no assumptions about the form of the underlying distribution of the data. Instead, we take a non-parametric approach and calculate the kernel density estimate of the data points with a Gaussian kernel. The objective is to find the optimal set of features $\mathbf{w}^*$ that maximizes the described criterion function $\mathcal{J}(\mathbf{w})$:

$$\mathbf{w}^* = arg\,max_{\mathbf{w}\in\{0,1\}^d} \mathcal{J}(\mathbf{w}) \tag{1}$$

We formally define the criterion function as:

$$\mathcal{J}(\mathbf{w}) = \frac{\frac{1}{|\mathbf{X}_-|}\sum\limits_{\mathbf{x}_-\in\mathbf{X}_-}\frac{1}{|N_k(\mathbf{x}_-)|}\sum\limits_{\mathbf{x}\in N_k(\mathbf{x}_-)} K(\mathbf{w}\circ\mathbf{x}_-, \mathbf{w}\circ\mathbf{x})}{\frac{1}{|\mathbf{X}_+|}\sum\limits_{\mathbf{x}_+\in\mathbf{X}_+}\frac{1}{|N_k(\mathbf{x}_+)|}\sum\limits_{\mathbf{x}\in N_k(\mathbf{x}_+)} K(\mathbf{w}\circ\mathbf{x}_+, \mathbf{w}\circ\mathbf{x})} \tag{2}$$

In the above equation, K is a kernel function. In our experiments, we use the Gaussian (or Radial Basis Function) kernel:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\parallel \mathbf{x} - \mathbf{x}' \parallel^2}{2\sigma^2}\right) \tag{3}$$

The vector $\mathbf{w} = (w_1, w_2, ..., w_d)$ is a binary vector signifying which features are selected; for $(j = 1, ..., d)$, $w_j = 1$ denotes the presence of feature j and $w_j = 0$ denotes its absence. We use $\mathbf{x}_- \in \mathbf{X}_-$ and $\mathbf{x}_+ \in \mathbf{X}_+$ to represent data points from the normal and outlier class, respectively. The parameter k determines the size of the local neighborhood of a data point, σ is the width of the Gaussian kernel, and the symbol $\circ$ represents the Hadamard product (Horn and Johnson, 1985). The Hadamard (or Schur) product of two matrices is their element-wise product.

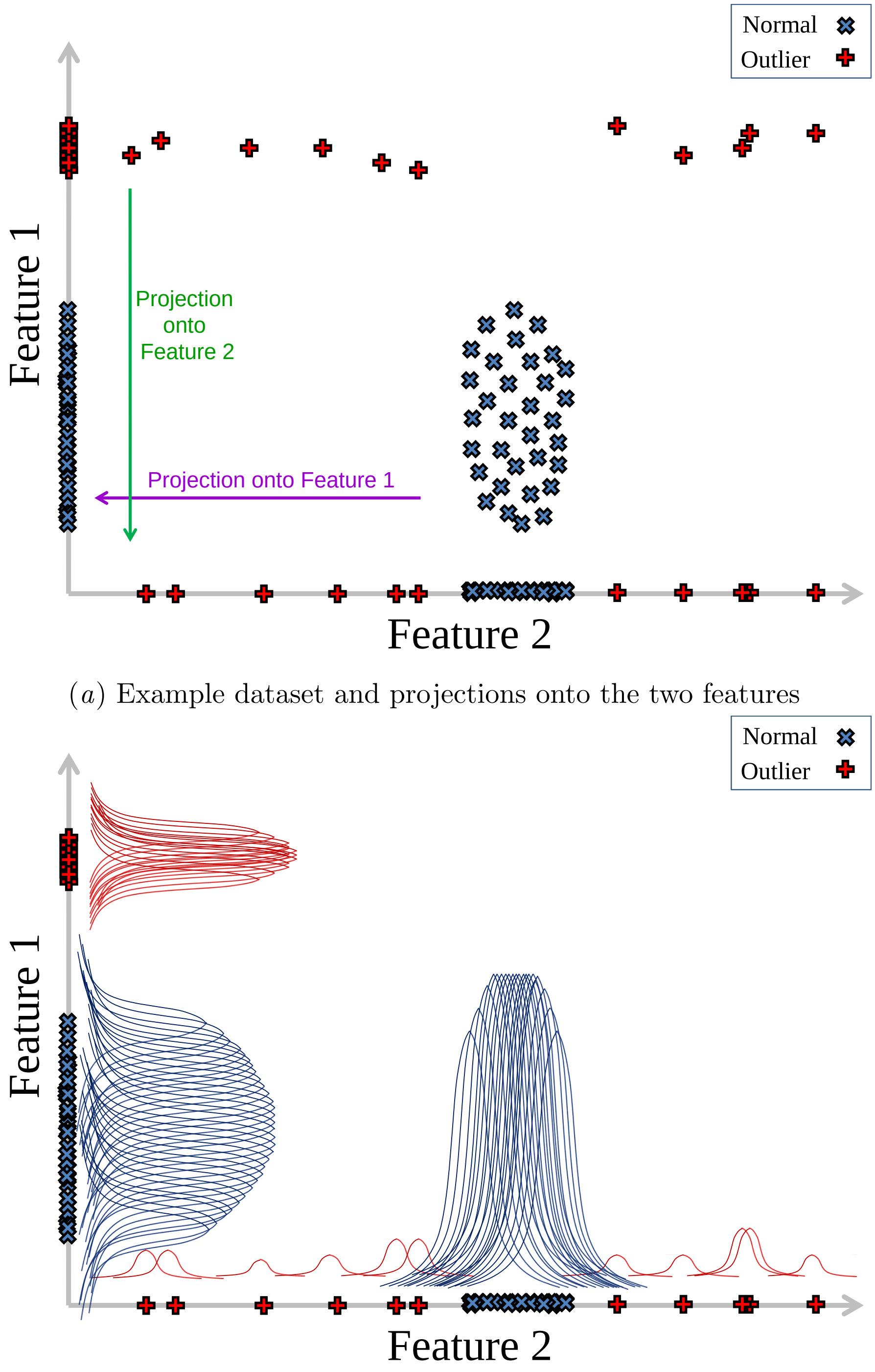
The size of the local neighborhood of a data point $\mathbf{x}$ is determined by the distance to its k^{th}-nearest neighbor, referred to as its k-distance. All of the data points whose distance to $\mathbf{x}$ is less than this distance comprise its k-distance neighborhood, $N_k(\mathbf{x})$. The number of data points in the k-distance neighborhood of a point may exceed k, due to possible ties in distance. Therefore in the criterion function, once we sum up the contributions to the density by the k-distance neighbors, we divide it by the number of points in the k-distance neighborhood, $|N_k(\mathbf{x})|$. The local neighborhood *density* of a data point $\mathbf{x}$ can be thought of as a measure of the *similarity* of points within that neighborhood to $\mathbf{x}$. Since a kernel function can be used as a measure of the similarity between two data points (Balcan and Blum, 2006), in Equation 2 other kernel functions can be used in place of the Gaussian kernel. We use the Gaussian kernel and effectively perform kernel density estimation (KDE), also referred to as the Parzen-Rosenblatt Window method (Parzen, 1962; Rosenblatt, 1956). This provides a standard, non-parametric notion of density for the data points.[2]

Our criterion function tries to optimize the ratio of local kernel density estimates for normal and abnormal points (outliers). In the numerator, we sum the local kernel density of all normal data points and in the denominator, we sum the local kernel density of all outliers. By maximizing the ratio of the two, the goal is to find the subset of features that maximizes the density of normal data points and simultaneously minimizes the density of outliers. Intuitively, we would like to find a lower dimensional subspace that corresponds to a subset of the features wherein normal data points are in closely compacted regions of the space while outliers are dispersed, allowing them to be more easily distinguished as anomalous with respect to the normal data. By using a *local* density approach, our criterion can aid outlier detection algorithms in detecting local, as well as global, outliers (cf. Breunig et al., 2000). In particular, we are already thinking in terms of local neighborhoods, as reflected in the KDE calculations. This notion can be carried over to the outlier detection phase, especially in the case of a local density-based outlier detection algorithm such as LOF which calculates the density of each point within a local neighborhood. This allows the detection of data points that seem to be *outlying* when considered within the scope of its local neighborhood, not just on a global scale.

To illustrate the advantage of using our proposed criterion function, we show a simple two-dimensional example in Figure 1. Assume we have a dataset with two features and we would like to select the single best feature for outlier detection. In the figure, we use × to represent normal points, + for outliers, and we show the projections of the data points onto each of the feature axes. In Figure 1(b), for each data point we use a bell-shaped curve to indicate the magnitude of its density. By projecting onto Feature 1, the two classes will have high separability, yet the outliers will have high density which makes it difficult for an outlier detection algorithm to identify them. Projection along Feature 2 gives the normal points high density and the outliers low density, facilitating their detection. A method that tries to best separate the classes, such as Linear Discriminant Analysis (LDA) which maximizes the trace of the between-class to within-class scatter ratio, would fail to select

[2] We use a single density estimator for the entire data, rather than having one per class.

Feature 2 over Feature 1. Our criterion will correctly select Feature 2 as it maximizes the density ratio of normal points to outliers.



(*a*) Example dataset and projections onto the two features

(*b*) Density of the data points

Figure 1: Example of feature selection for outlier detection

3.2. Forward Search Strategy

Using the described criterion function to evaluate features, the next component of our feature selection algorithm is its search strategy. There are many approaches to searching the space of possible feature subsets, from the naïve exhaustive search to more sophisticated search strategies such as genetic algorithms. In this work, we apply sequential forward selection (SFS), also referred to as sequential forward search (Devijver and Kittler, 1982). This is a greedy search technique that begins with an initially empty set and adds features one at a time such that the feature added at each round is the one that best improves the criterion. Note that one can utilize other search strategies, such as backward selection, sequential forward floating search, or applying sparse optimization (Tibshirani, 1994; Masaeli et al., 2010). For the purposes of this paper, whose goal is to test the appropriateness of the newly introduced criterion for outlier detection, we find it sufficient to utilize a simple search strategy that takes feature interaction into account, such as sequential forward search. We call our proposed method Local Kernel Density Ratio (LoKDR) feature selection.

3.3. Computational Complexity Analysis

One of the main components of calculating the criterion function is the k-nearest neighbor (k-NN) search. A simple brute force approach is to calculate the distance between all pairs of points, requiring $\frac{n(n-1)}{2} \times O(d)$ calculations, and then sort the distances to find the k-nearest neighbors of each point using $n \times O(n \log n)$ comparisons, for a total computational complexity of $O(n^2(d + \log n))$. Other k-NN algorithms have been proposed to reduce the computation time, where the main idea is to reduce the number of distances computed. For example, some algorithms partition the data points using an indexing structure, such as a kd-tree, and only compute distances within nearby volumes. This method has been shown to be faster than the brute force approach by up to a factor of 10 (Garcia et al., 2008).

Once the k-NNs of a point is found, the k-nearest neighbor distances are used to calculate its kernel density estimate. These KDE values are then summed up and averaged for the normal points and outliers ($O(n)$), after which the ratio is taken ($O(1)$). This produces the criterion value for one set of features. Assuming feature set F_i has $d_i \leq d$ features, the computational complexity of calculating the criterion function for F_i is: $O(n^2(d_i + \log n)) + O(n) + O(1) = O(n^2(d_i + \log n))$. It is interesting to note that all of the k-NN queries and feature subset evaluations are independent of one another. Thus, in an ideal situation where they are all performed simultaneously, round i of the algorithm would only require $O(d_i + n)$ time, with $O(d_i)$ time for a distance calculation consisting of d_i features and $O(n)$ time for the KDE summation. Since there are at most d rounds of the algorithm (corresponding to the addition of every feature), the best-case computational complexity of a parallel implementation of the LoKDR algorithm is $O(d^2 + nd)$. In practice, the forward search is cut off after a certain constant number of features $c << d$ are selected, yielding a ideal run-time of $O(c(d + n)) = O(d + n)$, making the algorithm linear in the size of the features and sample space. In future work, we will utilize the power of graphics processing units (GPUs) to perform many of the computations concurrently, pushing us closer to the realization of the ideal linear run-time.

Name	Features	Samples	Description
CNS	7129	90	Central Nervous System Embryonal Tumor Data: 60 samples have medulloblastomas and 30 have other tumors or no cancer.
LYMPH	7129	77	Lymphoma Data: 58 samples are diffuse large B-cell lymphomas and 19 are follicular lymphomas.
OVARY	6000	66	Ovarian Cancer Data: 50 samples are benign tumors and 16 are malignant tumors.
PROST	6000	89	Prostate Cancer Data: 63 samples have no evidence of cancer and 26 have prostate cancer.
ARRHY	276	450	Cardiac Arrhythmia Data: 244 samples are from class 01 and 206 are from classes 02-16.

Table 1: Overview of datasets

4. Experimental Evaluation

In this section, we describe the datasets and outlier detection algorithms used in this study to evaluate the quality of features selected by the LoKDR algorithm. We also briefly describe other feature selection techniques with which we compare our results. We then present the outlier detection results for each of the feature selection methods.

4.1. Datasets

The datasets used to evaluate our feature selection algorithm are shown in Table 1. CNS and LYMPH are microarray gene expression datasets and OVARY and PROST are mass spectrometry datasets provided by Chen and Wasikowski (2008). These datasets are examples of real-world problems that consist of a small set of samples and imbalanced class labels. We also evaluate our feature selection algorithm on the ARRHY dataset[3] (Güvenir et al., 1998) from the UCI Machine Learning Repository, a dataset which has neither highly imbalanced data nor a small sample space. The goal is to distinguish between the presence and absence of cardiac arrhythmia, where class 1 is the normal class and classes 2 to 16 are outliers. Table 1 provides the number of features, number of samples, and a summary description on each of these datasets.

4.2. Outlier Detection Algorithms

For outlier detection, we use one-class classifiers which are trained on only normal data (inliers). For each data point, the classifier produces a *decision value* that represents its confidence in that point being an outlier. We apply a threshold on the decision value as a cutoff point for decision making. A data point is flagged as an outlier if the decision value exceeds a threshold. Varying the threshold varies the number of correctly classified outliers (true positives) and incorrectly classified normal data (false positives). Using this information, we plot a curve of the true positive rate versus the false positive rate, known

[3] Preprocessing was done to account for missing values, resulting in the removal of three features and two instances.

as the Receiver Operating Characteristic (ROC) curve. In section 4.4, we perform an evaluation of several feature selection techniques in terms of the area under the ROC curve (AUC) achieved by the outlier detection algorithms on different feature subsets chosen by the feature selection techniques.

The classifiers used to evaluate the feature subsets are Nearest Neighbor (NN), Local Outlier Factor (LOF), and One-Class Support Vector Machines (OCSVM). The (one-class) Nearest Neighbor classifier is a distance-based outlier detection algorithm wherein a data point's decision value is the distance to its nearest neighbor. The greater the distance, the more likely that point is an outlier. The LOF algorithm takes a density-based approach to detect outliers; the greater the density of a point's nearest neighbors compared to its own density, the more outlying the data point. The decision value of a data point is its local outlier factor.

The OCSVM classifier (Manevitz et al., 2001) uses a kernel function to map the data into a feature space H with the goal of capturing most of the data vectors within a "small" region. It then tries to separate the mapped vectors from the origin with a hyperplane that has the maximum margin. The origin and data points "close enough" to it are assumed to be outliers. The decision value of a data point is $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) - \rho$ where α_i are the support vector coefficients, K is a kernel function, and ρ represents the margin. The parameter ρ is effectively the threshold that determines whether points are "close enough" to the origin to be considered outliers. For our OCSVM experiments, we use LIBSVM (version 3.11) (Chang and Lin, 2011) with the standard parameters for one-class SVMs.

4.3. Feature Selection Algorithms

To compare our results with other popular filter-based feature selection algorithms, we evaluate the features selected by RELevance In Estimating Features (RELIEF), Feature Assessment by Sliding Thresholds (FAST), Least Absolute Shrinkage and Selection Operator (LASSO), and BAckward elimination with the Hilbert Schmidt Independence Criterion (BAHSIC). RELIEF (Kira and Rendell, 1992) is a filter feature selection method that evaluates individual features based on how well they differentiate between neighboring instances from different classes versus from the same class. We use the Weka toolbox (Hall et al., 2009) to select features with RELIEF. FAST (Chen and Wasikowski, 2008) is a feature selection algorithm for small sample and imbalanced data classification problems. The main idea is to rank features based on the area under the ROC curve generated by each feature.

LASSO (Tibshirani, 1994) solves the linear regression problem formulation with an added constraint on the sum of the regression coefficients. This drives the coefficients of less relevant features towards zero. We use the logistic regression variant provided in the GLMNet (Friedman et al., 2010) Matlab toolbox and rank features based on the absolute value of the coefficients. BAHSIC (Song et al., 2007) uses the backward elimination search strategy on features evaluated using the Hilbert Schmidt Independence Criterion (HSIC). We use the Python code provided by Song et al. (2007) to perform feature selection with BAHSIC.

4.4. Results

For the training phase of the outlier detection algorithms, we take a one-class (or semi-supervised) learning approach and train only on normal data points.[4] During testing, both normal and outlier data points are used to see how well the model is able to detect outliers. We perform 10-fold cross validation by dividing the normal data points into ten folds and training on nine of them while testing on the tenth. Since no outliers are used during the training phase, we use all outliers during the testing phases of the cross validation.

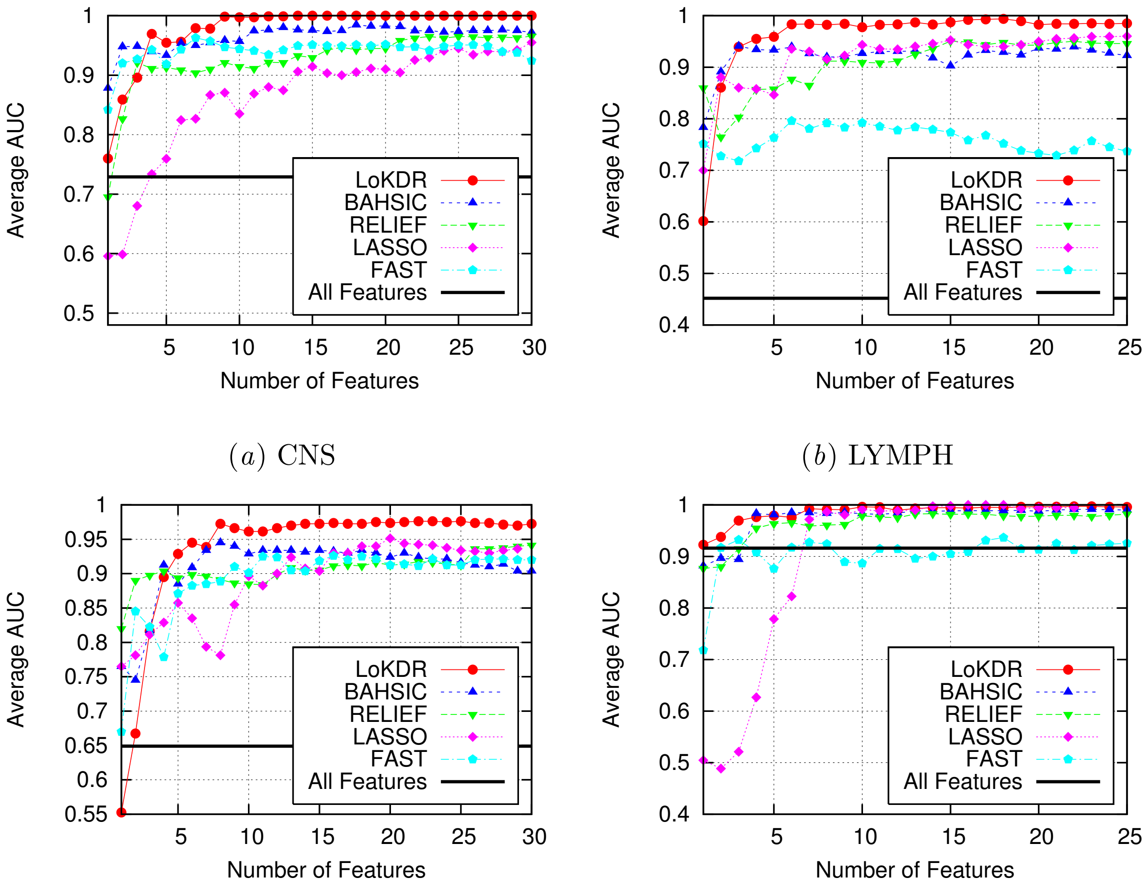
We evaluate the quality of the outlier detection results using the area under the ROC curve (AUC) and the balanced error rate (BER). The ROC curve is a plot of the true positive rate (fraction of outliers correctly detected) versus the false positive rate (fraction of normal points misclassified as outliers). It represents the behavior of a classifier across a range of thresholds on the decision values. The BER, on the other hand, represents the behavior of the classifier at a particular operating point and is the average of the false positive rate and the false negative rate (fraction of outliers misclassified as normal) at that threshold.

For the LoKDR feature selection algorithm, there are two main parameters that can be tuned: k which determines the size of the local neighborhood and σ, the Gaussian kernel width. By varying the value of k in $[1, n-1]$ and σ in [1, 5], we observed that most of the results were not drastically sensitive to the choice of these parameters, though some values produced slightly better results than others. The benefits of this are two-fold; first, it shows the stability of the criterion function, as it is not extremely sensitive to these parameters. Second, with smaller values of k, we can achieve similar (if not better) results than larger values, thereby reducing the computational cost. For our experiments, we set the values of the parameters based on 10-fold cross validation.

The results of our experiments using NN, LOF, and OCSVM show that in general,[5] the classifiers perform comparably across the various feature selection algorithms and datasets with no clear winner. As the goal is not to compare the outlier detection algorithms themselves, but rather to compare the proposed feature selection algorithm (LoKDR) with previous feature selection methods, in the figures of this section we shall present the outlier detection results using the LOF classifier. For the full set of results, please refer to (Azmandian, 2012). In Figure 2, we show the area under the ROC curve (AUC) results of the feature selection algorithms on the microarray and mass spectrometry datasets. On the x-axis, we vary the number of selected features and on the y-axis, we plot its corresponding average AUC. With a horizontal line, we show the AUC obtained when using all of the features. This displays the importance of performing feature selection, as all of the feature selection algorithms are able to surpass the AUC achieved with the entire feature set. The figure also highlights the strength of the LoKDR algorithm in selecting features for outlier detection. Across the datasets, the features chosen by LoKDR enable the outlier detection algorithm to identify outliers with a high detection rate and few false positives, as reflected in the high average AUC. For the CNS, LYMPH, and OVARY datasets, as the number of selected features increases, the average AUC for LoKDR rapidly exceeds that of the other methods.

[4] This is also the approach taken by Hido et al. (2011).

[5] There were a couple of exceptions to this with the NN classifier.



(*a*) CNS

(*b*) LYMPH

(*c*) OVARY

(*d*) PROST

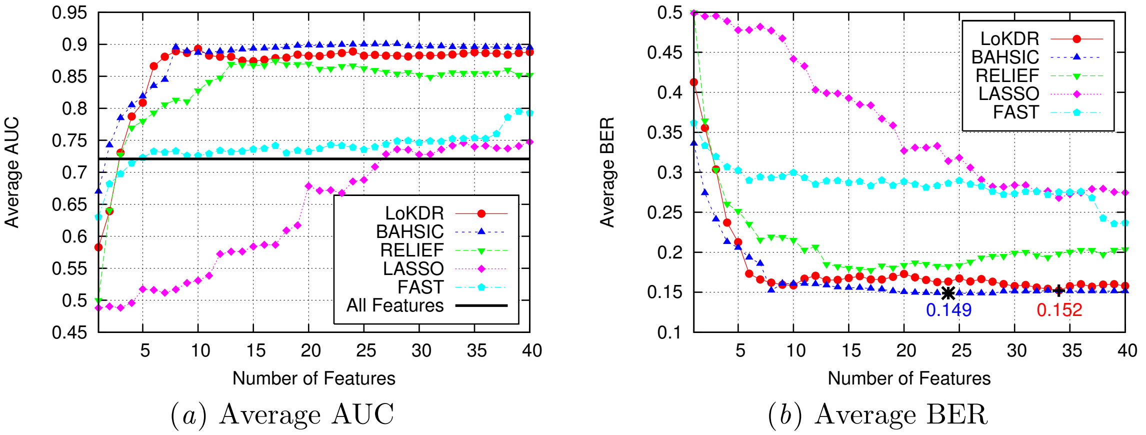
Figure 2: Average AUC results for microarray and mass spectrometry datasets

For the PROST dataset, the performance of LoKDR starts out the strongest and continues to be competitive with the other methods.

To see how well our feature selection algorithm performs on a more general dataset that does not have imbalanced data with a small sample size, we also ran experiments on the ARRHY dataset and present results in Figure 3(*a*). The results show that while features selected by BAHSIC produce the highest average AUC for most of the feature subsets, the LoKDR algorithm still performs well and produces AUC values that are comparable with those of BAHSIC.

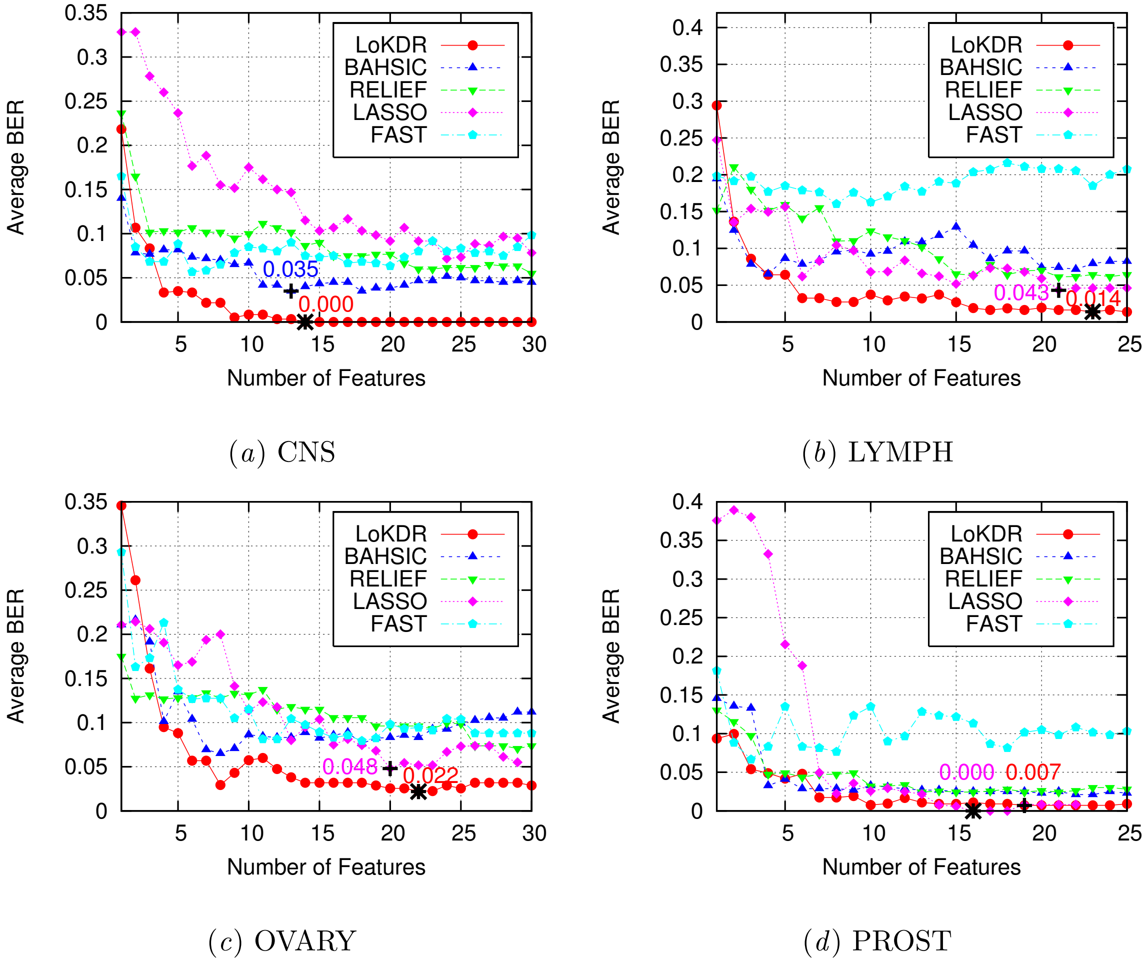
In addition to the area under the ROC curve which provides a picture of the outlier detection results across a range of thresholds on the decision values, we also provide a "snapshot" of the outlier detection results using the average minimum balanced error rate (BER).[6] These results are presented for the various feature selection algorithms in Figure 3(*b*) and Figure 4. On the x-axis, we have the number of selected features and on

[6] We find the minimum BER for different thresholds and then take the average across the cross-validation runs.



(a) Average AUC (b) Average BER

Figure 3: AUC and BER results for ARRHY dataset



(a) CNS (b) LYMPH

(c) OVARY (d) PROST

Figure 4: Average BER results for microarray and mass spectrometry datasets

the y-axis, the average minimum BER. For each dataset, the two lowest BER values are highlighted, with an asterisk at the lowest BER and a plus sign at the second lowest BER. The corresponding values are shown in the color of the feature selection algorithm that produced it. The results show that LoKDR consistently produces low values for the BER and in most cases, outperforms all of the other feature selection methods. For the CNS dataset, it is even able to achieve perfect detection (100% true positives and 0% false positives) with only 14 features. For the LYMPH and OVARY datasets, LoKDR achieves the lowest BER at a cost of a few more features than LASSO (which came in second). For the PROST and ARRHY datasets, LoKDR places a very close second to LASSO and BAHSIC, respectively.

We also vary the number of selected features up to 100 and present the lowest average BER results for the various feature selection algorithms (FS) and outlier detection algorithms (OD) in Table 2. For each FS/OD pair, we show the number of selected features next to the balanced error rate. For each dataset and outlier detection algorithm, the two top feature selection algorithms with the lowest BER are shown in boldface, with an asterisk next to the winner. These results show that LoKDR always achieves very low BER values and in every case, either outperforms all of the other methods or places a close second.

Dataset	**OD** / **FS**	NN		LOF		OCSVM	
		BER	Features	BER	Features	BER	Features
CNS	LoKDR	**0.000***	14	**0.000***	14	**0.000***	14
	BAHSIC	**0.035**	27	**0.035**	13	**0.042**	18
	RELIEF	0.047	41	0.048	37	0.055	51
	LASSO	0.060	37	0.050	36	0.050	37
LYMPH	LoKDR	**0.008***	42	**0.008***	41	**0.008***	39
	BAHSIC	0.183	9	0.065	4	0.082	3
	RELIEF	0.043	17	0.059	33	0.077	21
	LASSO	**0.041**	21	**0.043**	21	**0.043**	21
OVARY	LoKDR	**0.010***	42	**0.019***	35	**0.016***	36
	BAHSIC	0.060	21	0.065	8	0.056	9
	RELIEF	0.045	92	0.064	38	0.064	56
	LASSO	**0.039**	23	**0.048**	20	**0.045**	20
PROST	LoKDR	**0.007**	37	**0.007**	19	**0.007**	19
	BAHSIC	0.019	48	0.017	45	0.027	40
	RELIEF	0.023	97	0.017	93	0.025	64
	LASSO	**0.000***	18	**0.000***	16	**0.000***	16
ARRHY	LoKDR	**0.164***	34	**0.152**	34	**0.151**	34
	BAHSIC	**0.172**	31	**0.149***	24	**0.148***	29
	RELIEF	0.186	17	0.178	17	0.179	14
	LASSO	0.263	33	0.268	34	0.269	34

Table 2: Lowest average BER results for the feature selection algorithms on all the datasets

Using the paired Student's t-test, we confirmed that our experimental results showing the superiority of LoKDR over the other feature selection methods are statistically significant at the 95% confidence level with respect to all methods, except BAHSIC on the PROST and ARRHY datasets, where they perform comparably. From our results, we conclude that the

LoKDR feature selection algorithm chooses features that enable outlier detection algorithms to do consistently well across all the datasets, from those which are very high-dimensional with imbalanced class labels and that suffer from the small sample space problem, to a more general dataset without these properties.

4.5. Discussion

One of the characteristics of outlier detection problems is the class imbalance of the data, as outliers occur less frequently than normal points. Our proposed feature selection algorithm is robust to the degree of imbalance present in the data as it seeks features which maximize the average density of normal points while minimizing the average density of outliers. Binary classification is often performed on imbalanced class data problems and this inspired us to evaluate the quality of the features selected by our algorithm when the subsequent data mining task to perform is not outlier detection, but rather classification using two-class Support Vector Machines. The results (not shown here due to space limitations) were very encouraging as the area under the ROC curve values was consistently high, and out of the five feature selection algorithms tested, LoKDR was always one of the top-performing algorithms on all of the datasets.[7] This is a testament to the quality of the selected features and their ability to, in particular, identify characteristics that do well in distinguishing members of the larger class.

Binary classifiers provide supervised learning by using information about the class to which some data points belong. Knowledge about class labels can improve the accuracy of a learner by hinting at the characteristics of each class through the use of paradigms. Outlier detection, on the other hand, is typically performed as unsupervised or semi-supervised (one-class) learning. In cases where no labels for outliers exist, it is still desirable to select informative features for outlier detection. Therefore in future work, we will explore the notion of using the density of unlabeled data points in the denominator of our criterion function, thereby providing a one-class learning variant of our feature selection algorithm. While the use of some labeled outlier data can enhance the quality of the final solution, a one-class feature selector may be a better fit for the subsequent task of outlier detection.

5. Conclusions and Future Work

In this work, we presented a novel feature selection criterion catered to outlier detection problems. The proposed method is non-parametric and makes no assumptions about the distribution of the underlying data other than the fact that outliers are *different* than the normal data. It selects features that best capture the behavior of normal data while making outliers more distinct from the normal. We applied a forward search strategy to our evaluation criterion and compared its ability to detect outliers with other popular feature selection methods. Experiments on real datasets showed that our local kernel density ratio feature selection algorithm does very well to discern features that facilitate the detection of outliers.

[7] It was the best-performing algorithm on ARRHY and PROST, second-best on CNS and LYMPH, and a close third on OVAR.

In future work, we will incorporate other search techniques using our novel feature selection criterion and exploit the parallelism that exists in calculating the criterion to achieve computation time speedup, by implementing the algorithm on a graphics processing unit (GPU). We will also investigate the quality of the features chosen by our algorithm for real-world intrusion detection datasets. The selected features should capture the behavior of normal patterns while enhancing the detection of the malicious activity of malware attacks.

References

C. Aggarwal and S. Yu. An effective and efficient algorithm for high-dimensional outlier detection. *The VLDB Journal*, 14:211–221, 2005.

F. Azmandian. *Learning at the Virtualization Layer: Intrusion Detection and Workload Characterization from within the Virtual Machine Monitor*. PhD thesis, Northeastern University, August 2012.

M.-F. Balcan and A. Blum. On a theory of learning with similarity functions. In *International Conference on Machine Learning (ICML)*, pages 73–80, 2006.

M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. *ACM SIGMOD Record*, 29(2):93–104, 2000.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27, 2011.

X.-W. Chen and M. Wasikowski. FAST: A ROC-based feature selection metric for small samples and imbalanced data classification problems. In *KDD*, pages 124–132, 2008.

M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1: 131–156, 1997.

P. A. Devijver and J. Kittler. *Pattern recognition: A statistical approach.* Prentice Hall, 1982.

J. H. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *JMLR*, 5:73–99, 2004.

V. Garcia, E. Debreuve, and M. Barlaud. Fast k nearest neighbor search using gpu. In *CVPR Workshop on Computer Vision on GPU*, Anchorage, Alaska, USA, June 2008.

H. A. Güvenir, B. Acar, G. Demiröz, and A. Çekin. A supervised machine learning algorithm for arrhythmia analysis. In *Computers in Cardiology Conference*, pages 433–436, 1998.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, 11:10–18, 2009.

D. M. Hawkins. *Identification of outliers.* Chapman and Hall, London; New York, 1980.

S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Statistical outlier detection using direct density ratio estimation. *Knowledge and Information Systems*, 26(2): 309–336, 2011.

R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge; New York, 1985.

K. Kira and L. A. Rendell. A practical approach to feature selection. In *International Conference on Machine Learning (ICML)*, pages 249–256, 1992.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97 (1-2):273–324, 1997.

H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining.* Kluwer Academic Publishers, 1998.

L. M. Manevitz, M. Yousef, N. Cristianini, J. Shawe-Taylor, and B. Williamson. One-class svms for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001.

M. Masaeli, G. Fung, and J. G. Dy. From transformation-based dimensionality reduction to feature selection. In *ICML '10*, pages 751–758, 2010.

H. V. Nguyen and V. Gopalkrishnan. Feature extraction for outlier detection in high-dimensional spaces. *Journal of Machine Learning Research*, 10:66–75, 2010.

E. Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

M. Rosenblatt. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

A. Smola, L. Song, and C. H. Teo. Relative Novelty Detection. In *Artificial Intelligence and Statistics (AISTATS), JMLR W&CP 5*, 2009.

L. Song, J. Bedo, K. M. Borgwardt, A. Gretton, and A. Smola. Gene selection via the BAHSIC family of algorithms. *Bioinformatics*, 23:490–498, 2007.

M. Sugiyama, M. Yamada, P. von Bünau, T. Suzuki, T. Kanamori, and M. Kawanabe. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Networks*, 24(2):183–198, 2011.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.