Outlier Detection with Space Transformation and Spectral Analysis [*]

Xuan Hong Dang, Barbora Micenková, Ira Assent[†]   Raymond T. Ng[‡]

Abstract

Detecting a small number of outliers from a set of data observations is always challenging. In this paper, we present an approach that exploits space transformation and uses spectral analysis in the newly transformed space for outlier detection. Unlike most existing techniques in the literature which rely on notions of distances or densities, this approach introduces a novel concept based on local quadratic entropy for evaluating the similarity of a data object with its neighbors. This information theoretic quantity is used to regularize the closeness amongst data instances and subsequently benefits the process of mapping data into a usually lower dimensional space. Outliers are then identified by spectral analysis of the eigenspace spanned by the set of leading eigenvectors derived from the mapping procedure. The proposed technique is purely data-driven and imposes no assumptions regarding the data distribution, making it particularly suitable for identification of outliers from irregular, non-convex shaped distributions and from data with diverse, varying densities.

1 Introduction

Outlier detection aims at discovering anomalous or inconsistent patterns from a dataset and it is one of the major tasks in data mining [9, 15]. Detecting outliers is particularly important in a number of practical domains ranging from fraud detection, network intrusion identification to public health applications [5]. For example, an insurance company that maintains a database of incoming invoices may automatically identify anomalous documents that are suspect of having been tampered. Such fraudulent invoices will be characterized by unusual features compared to other invoices coming from the same source. Although widely studied in the past [2], outlier detection remains challenging since not only the number of outliers is significantly smaller than that of normal data, but also measuring outlierness of an object in its original space is usually difficult and imprecise. That makes the task often under-specified.

Existing data mining approaches to outlier detection mostly rely on notions of *distance* and *density*. Distance-based approaches [9] define an instance to be an outlier in case that it is sufficiently far from the majority of other instances in the dataset (a *global* outlier). Density-based approaches [4] consider an instance to be an outlier if its density is sufficiently small *relative to* the average density of its neighbors. Techniques based on these approaches typically explore outliers in their *original* data space (or sometimes in projected subspaces) and have demonstrated to work well on linearly separable distributions with stable densities. However, they tend to underperform when data distributions are not convex Gaussian and when the densities vary.

In this paper, we present an algorithm that exploits space transformation and uses spectral analysis in the new transformed space for outlier detection. As in density-based approaches, the proposed algorithm is capable of detecting *local* outliers, yet unlike existing algorithms that rely on notions of (Euclidean) distance or density, our technique develops a novel concept of local quadratic entropy for assessing the similarity of a data object with its nearby instances. The advantage of this information theoretic approach is that it imposes no assumptions regarding the data distribution and thus can closely approximate the local properties of the data whose distribution can be complex. Such an appealing property enables our technique to adaptively regularize the closeness amongst instances and subsequently benefits the procedure of mapping data into a novel (eigen)space. Through spectral theory, we demonstrate that both global and local outliers can be effectively identified by analyzing the eigenspace spanned by the set of the leading eigenvectors derived from the transformation procedure.

The overall detection technique is completely data-driven, making no assumptions about data distributions, which makes it suitable for differentiation of outliers from non-convex shaped data distributions and even from data with great variation in density. We demonstrate the effectiveness of our algorithm via a number of experiments on synthetic and real world benchmark datasets and compare its performance against the state-of-the-art algorithms in the literature.

2 Related work

Outlier detection algorithms can be grouped into two large categories: (i) statistical and (ii) data mining

[*] Part of this work has been supported by the Danish Council for Independent Research - Technology and Production Sciences (FTP), grant 10-081972

[†] Aarhus University, Denmark ({dang,barbora,ira}@cs.au.dk).

[‡] University of British Columbia, Canada (rng@cs.ubc.ca).

algorithms. Most statistical methods assume that the observed data can be fitted to a probability distribution with appropriate parameters (e.g., Binomial, Gaussian, Poisson etc.). An object is considered an outlier based on how unlikely it could have been generated by that distribution [2]. Data mining techniques, on the other hand, attempt to avoid model assumptions, relying on the concepts of distance and density, as stated earlier. For most distance-based methods [9, 18], two parameters, a distance d and data fraction p, are required. An object is identified to be an outlier if at least fraction p of all instances lie farther than d from it [9]. As both d and p are parameters defined over the entire data, methods based on distance can only find *global* outliers.

In contrast, techniques relying on density go further by seeking *local* outliers whose outlying degrees ("local outlier factor"—LOF) are defined w.r.t. their neighborhoods rather than the entire dataset. LOF-based techniques [4, 19] can uncover local outliers effectively in low dimensionality but they have problems with density variations and the detection result degrades as dimensionality increases. This is aggravated by the fact that, in practice, data often have a lower intrinsic dimensionality than that in which they are represented. Thus, detecting outliers in the original space can be hard since many features may be irrelevant for this objective.

There are several recent studies that attempt to find outliers in spaces with reduced dimensionality. Some of them [16, 8] represent subspaces as sets of principal components extracted by PCA, however, these methods can only uncover *global* outliers. SOD [10] and OUTRES [13] explore outliers in different projections of the original feature space. An outlying degree for each object is computed in each subspace and then they are aggregated to derive the overall degree [13]. ABOD [11] pursues a different approach where variance of angles among objects is taken into account to compute outlierness. That makes the method suitable for high dimensional data. We provide experimental comparisons with state-of-the-art algorithms in Section 5.

3 Problem Definition

In this paper, we focus on the problem of identifying a set of global and local outliers from a single dataset. We are given N data observations $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ in $\mathbb{R}^D$ that contain both regular and outlying instances. The objective is to find a set of outliers $\mathcal{A} = \{\mathbf{x}_i\}_{i=1}^n$ ($n \ll N$) that deviate from the majority of the observations in $\mathcal{X}$. In our work, we further divide the outlying patterns into two types: (i) *global* outliers—those forming a small group of data instances that are isolated and occupy no more than $c\%$ of the size of the entire dataset $\mathcal{X}$; (ii) *local* outliers—those being inconsistent w.r.t. the nearest group of regular instances and occupy no more than $c\%$ of the size of that group.

4 Space Transformation for Outlier Detection

In real-life applications, data is often represented in a multidimensional space which may hide the underlying inherent data structures. A natural approach is to transform the original data into a new space, usually with lower dimensionality, in which the intrinsic structure of the data can be revealed. For the challenging problem of identifying *local* outliers, retaining the local properties of the data under such a transformation is much more important than preserving its global properties (e.g., variances in PCA). In this work we exploit an approach which is conceptually similar to Laplacian Eigenmaps (LE) [12]. They ensure instances that ere close in the original space to be mapped close in the transformed space. However, the ultimate goal of LE is to find a *single* manifold embedding in the original space while our task is fundamentally different. We search for a transformation that emphasizes the differences between objects—potential outliers (or small groups of outliers) and their neighbors. Moreover, the data distribution may also contain multiple groups of regular objects, not just a single manifold. Therefore, we need a new mapping objective function which ensures that outliers are clearly separated from the regular objects in the transformed space.

4.1 Preliminaries on Laplacian Eigenmaps

LE is a graph-based technique for dimensionality reduction. Specifically, given the set of N data instances $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ in $\mathbb{R}^D$, it defines an undirected graph over $\mathcal{X}$ as $G = \{V, E\}$, where $V = \{v_i\}_{i=1}^N$ is a set of vertices and $E = \{e_{ij}\}$ is a set of edges, each connecting a pair (v_i, v_j). A vertex v_i in the graph corresponds to an instance $\mathbf{x}_i$ in the dataset $\mathcal{X}$ and the edge e_{ij} between v_i and v_j exists if the respective $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other. The closeness between $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined using the concept of ℓ-nearest neighbors. In particular, $\mathbf{x}_i$ is close to $\mathbf{x}_j$ if it is among the ℓ-nearest neighbors of $\mathbf{x}_j$ or vice versa. In addition to the graph, LE needs a weight matrix $\mathbf{K}$ of size $N \times N$ whose element $\mathbf{K}_{ij}$ denotes closeness between v_i and v_j. $\mathbf{K}_{ij}$ can simply be 1 if $\mathbf{x}_i$ and $\mathbf{x}_j$ are nearest neighbors and 0 if not, or it can be inversely proportional to the Euclidean distance between the two instances, or, alternatively, a value computed from a kernel function between them.

Given G and $\mathbf{K}$ defined above, the goal in the Laplacian eigenmap is to map the original data instances $\{\mathbf{x}_i\}_{i=1}^N$ into a new set $\{\mathbf{y}_i\}_{i=1}^N$ in $\mathbb{R}^d$ (usually $d \ll D$) so that weights and connections in graph G are retained as much as possible. If Y is a $d \times N$ matrix and $\mathbf{y}_i$ its column vectors, the mapping objective is:

$$Y^* = \arg\min_Y \sum_{i=1}^{N}\sum_{j=1}^{N} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \mathbf{K}_{ij}. \tag{4.1}$$

For later discussion, we further use F as the transposition of Y (i.e., $F = Y^T$) to denote the feature matrix in the induced mapping subspace. A column $\mathbf{f}_j$ in F corresponds to the j-th dimension spanned in the new subspace, thus $\mathbf{y}_i^T = [\mathbf{f}_1(i), \ldots, \mathbf{f}_d(i)]$.

4.2 Closeness with Local Quadratic Entropy

Similar to most kernel-based methods, the mapping results by LE are very much dependent on the weight matrix, making the definition of closeness a sensitive issue. Even when a similarity function is appropriately selected, setting the parameters for that function is not an easy task. In this work, we develop a closeness measure function that is adaptive to the local distribution properties of the data instances. Particularly, we want a mapping function that not only keeps neighboring instances with similar properties close to each other, but that also intentionally maps instances that do not share the same distribution properties far apart in the transformed subspace. In other words, if two neighbors have different distributions, their closeness should be proportionally penalized. In this way, not only that groups with considerably different densities are more separable, but also outliers that are inconsistent with the data are amplified and thus they are more prominent in the mapping space. To achieve this goal, we adopt a Gaussian kernel to quantify closeness:

$$\mathbf{K}_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\times(\beta\sigma)^2}\right), & \text{if } v_i,\ v_j \text{ connected} \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

Parameter β is used to regulate the kernel width σ. $\mathbf{K}_{ij}$ is only defined for connected vertices and so $\mathbf{K}$ is symmetric and sparse. Moreover, it is crucial to see that values of both β and σ directly affect the closeness between $\mathbf{x}_i$ and $\mathbf{x}_j$. However, they are not parameters of our method since we compute them directly from the data (for σ see also Section 4.5). In this work, we propose to use entropy – an information measure based on local distribution properties – to compute β.

In information theory, entropy is a measure quantifying uncertainty or information of a random variable. Mathematically, let X be a continuous random variable characterized by the probability distribution $p(\mathbf{x})$, then the entropy of X defined by Shannon [3] is $H(X) = -\int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x}$. If $H(X)$ is high, X's purity is low and it contains less information. Conversely, if $H(X)$ is low, X's purity is large and it contains lots of information. Intuitively, we can make use of the entropy concept to evaluate how different the local distribution surrounding a data instance is. We would expect that for a regular object, its local distribution is stable and close to homogeneous, the corresponding entropy is thus large. In contrast, the data distribution surrounding an outlier, or a boundary object between different distributions, is usually skewed and, consequently, entropy is low. Notice that, compared to other measures relying on statistical moments (e.g. variance), a measure relying on entropy or mutual information is appealing [14] since it is a function defined over $p(\mathbf{x})$ and thus able to utilize full essential information contained in the data.

Unfortunately, without assumptions about the data distribution, it is hard to compute Shannon's entropy $H(X)$ as we do not have access to the true probability density function $p(\mathbf{x})$. Therefore, in this work, we explore a more general form of entropy, named Renyi's entropy [14]. Specifically, with α being an order, Renyi's entropy is defined as[1]:

$$H_{R_\alpha}(X) = \frac{1}{1-\alpha} \log \int p(\mathbf{x})^\alpha d\mathbf{x}, \text{ for } \alpha > 0,\ \alpha \neq 1$$

In this study, we use $\alpha = 2$ which allows us to proceed to computation based on the data straightforwardly. With this setting, the entropy is called quadratic entropy and has the following formula:

$$H_{R_2}(X) = -\log \int p(\mathbf{x})^2 d\mathbf{x} \tag{4.3}$$

The key idea behind this selection is that we can integrate the quadratic entropy with the well known Parzen window method [3] for kernel density estimation to enable its computation. Notice that the Parzen window is a non-parametric technique and thus imposes no assumptions on the data distribution. Essentially, by placing a kernel function at each data instance, we evaluate $p(\mathbf{x})$ via the sum of kernels

$$p(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} G(\mathbf{x} - \mathbf{x}_i, \sigma^2) \tag{4.4}$$

where $G(\mathbf{x} - \mathbf{x}_i, \sigma^2) = \frac{1}{(2\pi\sigma)^{D/2}} \exp\left\{-\frac{||\mathbf{x}-\mathbf{x}_i||^2}{2\sigma^2}\right\}$ is a Gaussian in $\mathbb{R}^D$ space. An important property associated with this Gaussian kernel is that the convolution of two kernels remains a Gaussian function:

$$\int_{\mathbf{x}} G(\mathbf{x} - \mathbf{x}_p, \sigma^2) G(\mathbf{x} - \mathbf{x}_q, \sigma^2) d\mathbf{x} = G(\mathbf{x}_p - \mathbf{x}_q, 2\sigma^2)$$

We now give a definition of local quadratic entropy (QE) for each data instance $\mathbf{x} \in \mathcal{X}$. Let $\mathbf{x}_p$, $\mathbf{x}_q$ be any two $\mathbf{x}$'s ℓ-nearest neighbors and the QE of $\mathbf{x}$:

$$QE(\mathbf{x}) = -\log \frac{1}{(\ell+1)^2} \sum_{p}^{\ell+1}\sum_{q}^{\ell+1} G(\mathbf{x}_p - \mathbf{x}_q, 2\sigma^2). \tag{4.5}$$

Using relative differences in local quadratic entropies, we adaptively regulate the closeness between two neighbors $\mathbf{x}_i$ and $\mathbf{x}_j$. Based on this idea, we set β to the following ratio:

[1] Shannon's entropy is a special case of Renyi's for $\alpha \longrightarrow 1$

$$\beta = \frac{\min\{QE(\mathbf{x}_i), QE(\mathbf{x}_j)\}}{\max\{QE(\mathbf{x}_i), QE(\mathbf{x}_j)\}}, \tag{4.6}$$

where $0 < \beta \leq 1$. Thus, if $\mathbf{x}_i$ and $\mathbf{x}_j$ have different local quadratic entropies, their closeness is penalized proportionally to this difference. The more different the two instances are, the smaller β is and subsequently the weaker the connection between two instances in the graph. If two instances have similar entropies, it implies that they likely belong to the same regular group. Their β is close to 1 and the connection unchanged.

This simple setting of β can be effective in most cases, especially if data distributions have similar densities. In practice, however, we may confront situations where data distributions are diverse or even show gradual changes. Equally applying β might not be a wise strategy since it may stretch out the distances amongst these instances and thus can further obscure the difference between outliers and true inliers in sparse distributions. We hence assume that two data instances belong to different distributions if the following inequality is satisfied:

$$\frac{\min\{QE(\mathbf{x}_i), QE(\mathbf{x}_j)\}}{\max\{QE(\mathbf{x}_i), QE(\mathbf{x}_j)\}} \leq \gamma, \tag{4.7}$$

where γ is a threshold between $(0, 1]$. Conversely, $\mathbf{x}_i$ and $\mathbf{x}_j$ are supposed to be taken of the same distribution if the above ratio is greater than γ. Furthermore, for instances that are in a sparse region and whose ratio is at the same time above γ, we increase their closeness so that they are mapped close in the new space. The average local quadratic entropy over the entire dataset is $QE(\mathcal{X}) = N^{-1} \sum_i QE(\mathbf{x}_i)$, and $\mathbf{x}_i$ and $\mathbf{x}_j$ are said to be in a sparse distribution if both $QE(\mathbf{x}_i)$ and $QE(\mathbf{x}_j)$ are smaller than $QE(\mathcal{X})$. In this case, their kernel width is widened proportionally by setting β to:

$$\beta = \frac{QE(\mathcal{X})}{\max\{QE(\mathbf{x}_i), QE(\mathbf{x}_j)\}}. \tag{4.8}$$

4.3 Exploring Eigenspace for Outliers

Using the above setup, connected instances with similar local entropy are expected to be mapped close in the transformed space whereas true outliers or objects with dissimilar entropies (despite being connected) are mapped far apart due to the loose connections amongst them. However, the question of how to determine the anomalous instances in the new transformed space remains. Fortunately, a clear advantage of our proposed scheme is that we are able to progressively explore each dimension of the transformed space to uncover outliers.

Let us define the degree (or volume) of a vertex v_i associated with $\mathbf{x}_i$ as $\mathbf{D}_{ii} = \sum_j \mathbf{K}_{ij}$. Thus, $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii}$'s being its diagonal elements. For ease of discussion, first assume that the data is mapped into a 1-dimensional space[2]. Then the summation in Eq.(4.1) can be reformulated as:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} (y_i - y_j)^2 \mathbf{K}_{ij} = 2\mathbf{f}^T(\mathbf{D} - \mathbf{K})\mathbf{f}. \tag{4.9}$$

Recall that $\mathbf{f} = [y_1, \ldots, y_N]^T$ is a feature vector in the induced space (as mentioned in Section 4.1) and let us denote $\mathbf{L} = \mathbf{D} - \mathbf{K}$ the Laplacian matrix of the graph. Then, searching for the first optimal dimension $\mathbf{f}$ in the transformed space can be rephrased as:

$$\arg\min{}_{\mathbf{f}}(\mathbf{f}^T\mathbf{L}\mathbf{f}), \text{ s.t. } \mathbf{f}^T\mathbf{f} = 1 \tag{4.10}$$

with the constraint $\mathbf{f}^T\mathbf{f} = 1$ which is added to remove the freedom of $\mathbf{f}$. Finding the solution for this constrained optimization is equivalent to solving $\mathcal{L}(\mathbf{f}, \lambda) = \mathbf{f}^T\mathbf{L}\mathbf{f} - \lambda(\mathbf{f}^T\mathbf{f} - 1)$ by the Lagrange multipliers method, with λ as a multiplier. As such, setting the 1st-order derivative of $\mathcal{L}(\mathbf{f}, \lambda)$ w.r.t. λ to zero gives us the satisfaction over the constraint whereas equating such derivative w.r.t. $\mathbf{f}$ to zero leads to $\mathbf{L}\mathbf{f} = \lambda\mathbf{f}$. Consequently, computing the optimal mapping subspace is equivalent to solving the eigenvalue problem. Since $\mathbf{L}$ is symmetric (due to the symmetry of $\mathbf{D}, \mathbf{K}$) and positive semi-definite (due to all $\mathbf{K}_{ij} \geq 0$), its eigenvalues are real and non-negative. Therefore, we can order its pairs of eigevalues/vectors $\{(\lambda_k, \mathbf{f}_k)\}_{k=1}^N$ such that $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$. The first optimal dimension turns out to be the first eigenvector of $\mathbf{L}$ with the corresponding smallest eigenvalue (and with the same reasoning, we can select the first d eigenvectors as our optimal d-dimensional mapping space for the general case).

From the perspective of spectral graph theory we assume that $f(.) : V \mapsto \mathbb{R}$ is a nonlinear function that maps the vertices of graph G into our first optimal dimension $\mathbf{f} \in \mathbb{R}$, i.e. $f(v_i) = \mathbf{f}(i) = y_i$. Then, we can view this mapping problem as a graph embedding problem. As a result, smoothing of the function is always desirable since if two vertices v_i and v_j are similar (i.e., strongly connected), their $\mathbf{K}_{ij}$ is large and thus the difference between two mappings $f(v_i)$ and $f(v_j)$ will be small if $f(.)$ is smooth. In short, the smoothness of the function ensures values of nearby similar objects to be mapped close to each other. We therefore rely on the smoothness property of the leading eigenvectors when searching for outliers and divide the process into two cases: searching for global outliers and searching for local outliers.

4.3.1 Global outlier search

Recall that when computing closeness amongst data instances, we penalize the weight for pairs of instances with different local

[2] The generalization to d dimensions is straightforward once we derive the solution for this case.

quadratic entropy. This is done to weaken connections between the graph components (or sets of connected vertices) if they correspond to different distributions, and especially for those vertices that correspond to truly isolated outliers. As a consequence, we observe as many zero eigenvalues of $\mathbf{L}$ as the number of loosely connected or disconnected components. Essentially, the spectrum of $\mathbf{L}$ is the union of spectra of each individual connected component where each component has the unique smallest eigenvalue equal to zero. Moreover, the eigenvector $\mathbf{f}_k$ associated with the zero eigenvalue λ_k will be the indicator vector for the k-th connected component. That means that this vector has non-zero values at the component's vertices and zero values otherwise in order to make $\mathbf{f}_k^T L\mathbf{f}_k$ vanished.

Hence, we can say with certainty that $\mathbf{f}_k$ is an indicator vector for *outliers* if there is only a small number of non-zero values among its elements which means that they correspond to an isolated small group of data instances. More specifically, if there are only $c\%$ of non-zero entries in $\mathbf{f}_k$, i.e. $|\mathbf{f}_k(.) \neq 0| \leq c \times |\mathbf{f}_k|$ (notice the cardinality $|\mathbf{f}_k|$ is equal to N), then we classify each $\mathbf{x}_i$ corresponding to $\mathbf{f}_k(i) \neq 0$ as a global outlier. Moreover, as the transformed space is spanned by the set of eigenvectors (corresponding to zero-eigenvalues) and as the norm of each eigenvector is bounded by 1 (see Eq. (4.10)), global outliers are those points that lie far apart from the origin and close to 1. For the groups of points that lie close to the origin, further analysis is needed. These instances possibly represent groups of regular objects yet loosely connected with (local) outliers in between. We address this case in the following section.

4.3.2 Local outlier search We view each eigenvector corresponding to a zero-eigenvalue not yet covered by the above case as an indicator vector for a connected subgraph and compute its eigenvalues and eigenvectors separately. Without loss of generality, let G_p denote the subgraph and assume it contains N_p vertices corresponding to $\{\mathbf{x}_i\}_{i=1}^{N_p}$ original instances. Likewise, we use $\mathbf{K}_p, \mathbf{D}_p, \mathbf{L}_p$ to denote the weight, degree and Laplacian matrices respectively for that subgraph. Obviously, $\mathbf{L}_p$ has a single eigenvalue $\lambda_1^{(p)} = 0$ and the corresponding constant eigenvector $\mathbf{f}_1^{(p)}$. There are three cases for this subgraph: (1) It may represent only a single regular group without any outliers. (2) It contains a single regular group and a few anomalous instances. (3) It comprises more than one regular group, yet some outliers in between make them weakly connected (this is the most general case).

Case 1: To distinguish this first case from the other two ones, we can explore the property of the second eigenvalue $\lambda_2^{(p)}$. Specifically, it can be proven [6] that the larger the value of the second eigenvalue is, the better the connection amongst the graph's vertices, which can be used as a measure of the graph's connectivity degree. Therefore, if the gap between $\lambda_2^{(p)}$ and $\lambda_1^{(p)}$ is much larger than the gap between $\lambda_3^{(p)}$ and $\lambda_2^{(p)}$ (at least 3 times in our experiments), we consider G_p tightly connected and without any local outliers.

Case 2: To see whether subgraph G_p comprises a single regular group and several outliers, we may check the smoothness of its second eigenvector $\mathbf{f}_2^{(p)}$. In particular, we sort its entries in ascending order and find the largest gap index q between two adjacent elements, i.e., $q = \arg\max\, (\mathbf{f}_2^{(p)}(q+1) - \mathbf{f}_2^{(p)}(q))$. Thus, if $q/N_p \leq c$, it indicates that $\mathbf{x}_i$'s corresponding to the q smallest values of $\mathbf{f}_2^{(p)}(i)$ are local outliers (w.r.t. to the rest instances in G_p). Practically, we can also directly exploit the degrees of vertices to identify anomalous instances in this case. Intuitively, since most vertices in G_p correspond to regular objects from the same group, those having the smallest degree will correspond to the most isolated objects. If the number of such isolated instances is less than $c\%$, they can be safely considered local outliers.

Case 3: Analyzing the third case involves more reasoning. Let's assume that $\widetilde{\mathbf{L}}_p$ is an *ideal* Laplacian matrix where all d normal groups can be perfectly separated using the first d eigenvectors of $\widetilde{\mathbf{L}}_p$. Following the perspective of perturbation theory [17], the *computed* $\mathbf{L}_p$ can be considered as the result of the ideal $\widetilde{\mathbf{L}}_p$ being *perturbed* by a matrix $\mathbf{A}$, representing a small set of anomalous points, i.e. $\mathbf{L}_p = \widetilde{\mathbf{L}}_p + \mathbf{A}$. Let F_p and $\widetilde{F_p}$ be the sets of the first d eigenvectors of $\mathbf{L}_p$ and $\widetilde{\mathbf{L}}_p$, respectively. Then, according to the Davis-Kahan theorem [17], the distance $\mathbf{d}(F_p, \widetilde{F_p})$ between F_p and $\widetilde{F_p}$ is bounded by $\mathbf{d}(F_p, \widetilde{F_p}) \leq \delta^{-1}\ \|\mathbf{A}\|_{\mathcal{F}}$, of which $\|\mathbf{A}\|_{\mathcal{F}}$ is $\mathbf{A}$'s Frobenius norm and $\delta = \min\{|\widetilde{\lambda^{(p)}} - s|\}$ with S being a real interval, $\widetilde{\lambda^{(p)}} \notin S, s \in S$, which should cover the first d eigenvalues of both $\mathbf{L}_p$ and $\widetilde{\mathbf{L}}_p$. Essentially, let $\widetilde{\lambda^{(p)}}_{d+1} > \widetilde{\lambda^{(p)}}_d$ and in our case we select $\widetilde{\lambda^{(p)}}_d$ as the upper bound of S, then δ receives a high value if the eigengap $|\widetilde{\lambda^{(p)}}_{d+1} - \widetilde{\lambda^{(p)}}_d|$ is large. This leads to a small value of $\mathbf{d}(F_p, \widetilde{F_p})$ given our assumption of a small set of outliers encoded in $\|\mathbf{A}\|_{\mathcal{F}}$. Consequently, for a small distance $\mathbf{d}(F_p, \widetilde{F_p})$, the first d eigenvalues/vectors of the ideal $\widetilde{\mathbf{L}}_p$ and of the perturbed $\mathbf{L}_p$ are close to each other.

Despite using this theorem for reasoning on $\widetilde{\mathbf{L}}_p$ for a selected range S, it still shows a close relationship between the computed and ideal cases, especially when the norm of $\mathbf{A}$ is small (which is true for our study

Algorithm 1: OutDST

```
Input: dataset $\mathcal{X}$; $c, \gamma$ parameters;
Output: Set of outliers $\mathcal{A}$ uncovered from $\mathcal{X}$;
Compute $QE(\mathbf{x}_i)$ according to Eq (4.5) and $QE(\mathcal{X})$
for each pair $(\mathbf{x}_i, \mathbf{x}_j)$ do
    $\beta \leftarrow \min\{QE(\mathbf{x}_i), QE(\mathbf{x}_j)\}/\max\{QE(\mathbf{x}_i), QE(\mathbf{x}_j)\}$
if  $\beta > \gamma \wedge \max\{QE(\mathbf{x}_i), QE(\mathbf{x}_j)\} < QE(\mathcal{X})$
$\beta \leftarrow QE(\mathcal{X})/\max\{QE(\mathbf{x}_i), QE(\mathbf{x}_j)\}$
Compute $\mathbf{K}_{ij}$ according to Eq (4.2)
then
Compute $\mathbf{D}$ with $\mathbf{D}_{ii} = \sum_j \mathbf{K}_{ij}$
Compute $\mathbf{L} = \mathbf{D} - \mathbf{K}$ and $\mathbf{L}$'s $\{(\lambda_k, \mathbf{f}_k)\}$
/*Global outliers */
if number of $\lambda_k = 0$ is larger than 1 then
    for each $\mathbf{f}_k$ having $\lambda_k = 0$ do
        if $|\mathbf{f}_k(.) \neq 0| \leq c \times |\mathbf{f}_k|$ then
            $\mathcal{A} \leftarrow \mathbf{x}_i \cup \mathcal{A}$ for each $\mathbf{f}_k(i) \neq 0$
        else
            Find eigengap $d$ and apply k-means
            Compute $\mathbf{L}_p$ corresponding to each graph
            component
else
    Compute $\mathbf{K}_p, \mathbf{D}_p, \mathbf{L}_p$ corresponds to $\mathbf{f}_2$
/*Local outliers */
for each $\mathbf{L}_p$ do
    Compute $\mathbf{f}_2^{(p)}$ of $\mathbf{L}_p$ and order its entries
    $q \leftarrow \mathrm{argmax}_q(\mathbf{f}_2^{(p)}(q+1) - \mathbf{f}_2^{(p)}(q))$
    $\mathcal{A} \leftarrow \mathbf{x}_i \cup \mathcal{A}$ for each $\mathbf{f}_2^{(p)}(i)$ with $i \leq q$ and $q \leq c.N_p$
return $\mathcal{A}$
```

of outliers) and G_p represents a set of well pronounced groups of regular objects. Therefore, we can use a large eigengap computed from $\mathbf{L}_p$ to estimate the number of normal groups d in the subgraph G_p. Given the set of corresponding $d-1$ top eigenvectors (excluding the first eigenvector), the k-means algorithm can be applied to divide F_p's rows into d disjoint groups, and we can view the corresponding graph components of these groups as either the first or second cases presented above.

The entire algorithm, OutDST (Outlier Detection with Space Transformation), is given in Algorithm 1.

4.4 Algorithm complexity OutDST requires the calculation of the Gaussian convolution amongst data instances and subsequently the $\mathbf{K}$ matrix. Both these steps take $O(DN \log N)$ with the implementation of a k–d tree. The size of $\mathbf{L}$ is the same as that of $\mathbf{K}$, so its eigendecomposition complexity is the most expensive. However, since $\mathbf{L}$ is sparse and we need to consider only the first few eigenvectors, the Lanczos method [7] can be employed to reduce the time to $O(DN \log N)$. Likewise, for each subgraph G_p, the eigendecomposition of the corresponding matrix $\mathbf{L}_p$ amounts to $O(DN_p \log N_p)$. Thus, the overall complexity of our algorithm is $O(DN \log N)$.

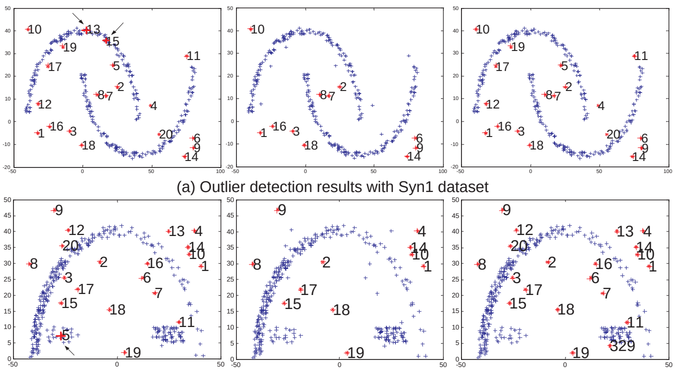
4.5 Kernel parameter setting For most practical applications where data is finite, the kernel size should be selected such that it balances bias and variance, which can essentially be derived from the optimization of the mean integrated squared error between an estimator $\widehat{p}(x)$ and the true density $p(x)$: $MISE\{\widehat{p}(x)\} = \int_x E\{[\widehat{p}(x) - p(x)]^2\}dx$. In our work, we chose $\sigma = \widehat{\sigma}\,(4/(N(2D+1)))^{\frac{1}{D+4}}$, [3] which is found by applying least square cross-validation and normal reference rule [20] to minimize the generalization error. As shown in the experiments, this selection works reasonably well for most examined datasets.

5 Experimental Results

In this section, we provide experimental results on synthetic and real datasets. We compare OutDST against the following algorithms: LOF (density-based technique) [4], ABOD (angle-based) [11] and SOD (axis-parallel subspaces) [10]. In addition to these algorithms, we implement the mutual ℓ-nearest neighbor graph (here called MGraph) which is known to capture well the differences in data densities. We will contrast it with our adaptive entropy-based approach. For OutDST algorithm, we select $c\%$ to be equal to the percentage of the ground truth outliers and vary γ from 0.5 to 0.8 until OutDST outputs such desired number of outliers. Recall that γ is the threshold that OutDST considers two objects being taken from potentially different distributions (Eq.(4.7)). If we set γ too small, the algorithm might be too sensitive and identify outliers that are not significant. For all techniques we vary the number of neighbors (*minPts* in LOF or the reference points in SOD) between 8 and 20 (step 2) and report the best results. For SOD, we further set $\alpha = 0.8$ as recommended in [10].

5.1 Synthetic Datasets We use two synthetic datasets to evaluate the performance. Syn1 contains two half-moon groups of regular data, each with 150 points plus 20 random points as outliers. Due to the randomness, outliers number 13 and 15 (marked by short arrows in Figure 1(a)), lie within the groups of regular points. The goal of using this dataset is to test whether our algorithm can distinguish outliers from non-convex, irregular shaped groups of regular points. Syn2 uses a more complicated setup which comprises a half-ring shaped distribution and two uniform distributions. To simulate different and varying densities, the half-ring's distribution is gradually fading from left to right while the right uniform distribution is three times denser than

[3] $\widehat{\sigma} = \sum_i \sigma_i/D$, σ_i's are diagonal elements of the sample covariance matrix after removing top 5% of the points with the largest Mahalanobis dist. (extreme points) from the sample mean.


(a) Outlier detection results with Syn1 dataset

(b) Outlier detection results with Syn2 dataset

Figure 1: Performance of OutDST on 2 synthetic datasets. Left figures show original data, middle ones show global outliers found by OutDST and right figures show its final results.

the left one (see Figure 1(b)). These two uniform distributions are intentionally located close to each tail of the half-ring shaped distribution. Similar to Syn1, 20 outliers are randomly added to Syn2. The purpose of this dataset is to verify whether our algorithm can differentiate outliers from data with different yet closely located and varying densities. We generate this data in a 2-dimensional space for the purpose of visualization. Concerning multiple-dimensional spaces, we will shortly present real world data.

The datasets and output of the OutDST algorithm are shown in Figure 1 (we omit outputs of the other algorithms due to space constraints). For comparison against other techniques, we use *F-measure* which is the harmonic mean between precision and recall. They are reported in Figure 2. Note that, unlike other methods, OutDST assigns binary labels (outlier or not) to data instances. In addition, it uses γ to adjust the number of output outliers which is not always equal to the number of outliers in ground truth. Thus, in Figure 2, we plot two different results for OutDST with the number of outliers closest to the desired number and denote the larger one by (*). The same situation is for MGraph where we report its best results only to save space. For other approaches, the results are based on the top-20 ranked outliers (20 is the number of true outliers in the ground-truth).

For Syn1 dataset, it can be seen from Figures 1(a) and 2(a) that OutDST can correctly distinguish true outliers from the non-convex shaped groups of regular instances with 18 out of 20 generated outliers being found. Increasing γ does not lead to any improvement since (as mentioned previously) 2 outliers lie inside the distribution and cannot be detected by any algorithm. MGraph and LOF also perform well on this dataset since they isolate outliers based on the differences in data densities. The performance of ABOD is slightly worse. SOD is also less successful because the approach is based on the axis-parallel projections that cannot handle non-convex shapes of data distributions.

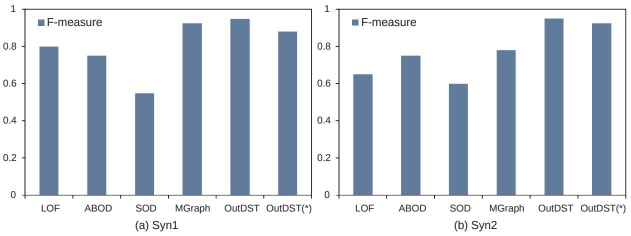
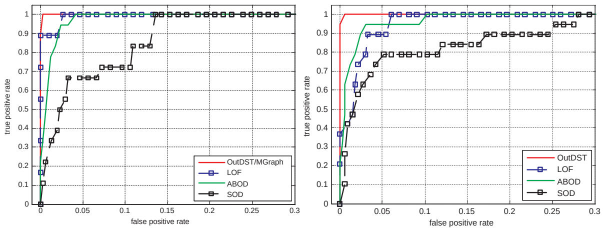


Figure 2: F-measure of all algorithms on two synthetic datasets. For OutDST, values with (*) corresponds to larger values of γ (i.e., with more outliers being reported).

For Syn2 dataset shown in Figure 1(b) and Figure 2(b), the performance of OutDST is also high. When γ is set to 0.75, it isolates 20 instances as outliers of which only one is falsely selected (point labeled 329 in Figure 1(b) on the right). It is interesting to compare this result to that of MGraph which lacks our adaptive entropy-based approach. Its performance is much worse due to the presentation of the varying density distributions. MGraph selects most instances in the half-ring's sparse area as outliers while it is also unable to discover outliers close to the half-ring (e.g., instances with labels 11 or 20). In terms of F-measure, its best performance is 78% compared to 95% of OutDST which clearly demonstrates the advantage of the local adaptive entropy-based approach. ABOD and SOD do not show high detection rates and LOF, which works well on Syn1 dataset, also performs worse on this more complex data. Across different *MinPts* values, we observe that LOF usually credits high outlying degree to the instances located at the boundary between the sparse uniform distribution and the half-ring shaped one, and instances located at the right-most tail of the half-ring distribution. This happens since LOF directly seeks outliers in the original space and its density computation tends to be sensitive to the distance of the farthest points among *MinPts* neighbors since they can be selected from different distributions. OutDST, in contrast, handles these challenges by penalizing the data instances using local quadratic entropies and subsequently seeking the most deviating outliers in the transformed space where distributions with varying densities can be well separated.

Notice that if we do not consider instances 13 and 15 in Syn1 and instance 5 in Syn2 as true outliers, then ROC curves can be plotted, which gives better comparison of all algorithms. Also, as plotting ROC requires outlier ordering, we keep the sequence of OutDST's outliers as they are naturally uncovered when analyzing the subgraphs. The ROC curves of all algorithms are plot-



(a) Syn1 (b) Syn2

Figure 3: ROC curves of all algorithms on two synthetic datasets computed by excluding instances 13 and 15 in Syn1 and instance 5 in Syn2 (best visualized in color).

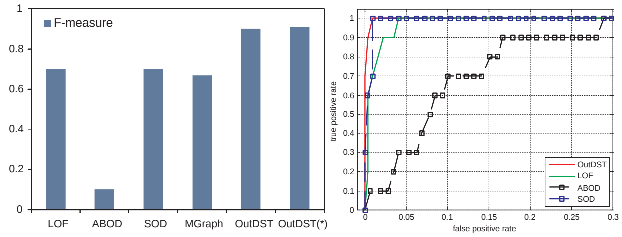
ted in Figure 3 (except MGraph's on Syn2 since it was unable to discover all true outliers). It is clear from the figure that other algorithms than OutDST ranked many regular points before true outliers.

5.2 CMU face data The first real-world dataset that we examine is the CMU face data from the UCI repository [1]. This dataset contains images of 20 people taken with different facial expressions (neutral, happy, sad, angry), head positions (left, right or straight), eye states (open or sunglasses), etc. Each person has 32 images captured in every combination of these aspects and at the resolution of 32×30 resulting in 960 dimensions. We select all images of 10 random people as regular data and instead of adding artificial outliers, we randomly pick one image from each remaining person's images as an outlier. Thus, we obtain a dataset of 330 images containing 10 outliers.

In Figure 4(a), we report the F-measure performance of all algorithms, and in Figure 4(b), their corresponding ROC curves are plotted. The experiment reveals that OutDST can also achieve good performance on this high-dimensional dataset. It successfully detects 9 out of 10 true outliers. Furthermore, when the value of γ is slightly increased, OutDST isolates two more outliers of which one is indeed the last truly anomalous image. It is also visible that the detection rate of SOD is reasonably good but LOF and MGraph are less successful (MGraph is unable to find all true outliers and thus its ROC curve could not be drawn). These results are consistent with those presented in [10] where SOD was also superior to LOF on various high dimensional datasets. On the other hand, the performance of ABOD is not as expected. Its approach relying on angles might be inappropriate for this image data where the data distributions may have non-convex shapes (especially in the full dimensional space). Overall, it is clear from Figure 4 that the performances of these techniques are all worse compared to that of the proposed OutDST algorithm.

5.3 Other real world data We further test the performance of all algorithms on three real-world datasets from the UCI repository [1]. The first one is the letter recognition data consisting of 20 000 instances characterizing 16 numerical features of 26 capital letters in the English alphabet. The second dataset is the vowel data that consists of 990 instances and is described by 14 variables of which the last one is the class label encoding 11 different vowels. The third dataset is the image segmentation data which includes 2 310 instances classified into 7 classes described by 19 attributes. For the letter recognition dataset, we select at random 3 letters along with all their corresponding instances as regular data and randomly select one instance from each remaining letter to form a dataset with 23 outliers. For the vowel dataset, we choose instances from one randomly selected vowel as regular objects and one from each remaining vowels as outliers, getting in total 10 outliers. Likewise, instances from two randomly selected classes of segmentation data are chosen as regular data and two instances from each remaining classes are selected as outliers, forming the dataset with 10 anomalous instances. These datasets were chosen since they are known to be hard for supervised learning as well due to the non-linear separation amongst classes.

We compare the performance of all algorithms via the F-measure reported in Table 1. For OutDST, we compute its F-measure such that the number of returned outliers is as close to the ground truth number as possible. Specifically, OutDST identified 26 instances as outliers for Letter, 13 for Vowel and 11 for Segmentation dataset. To make fair comparisons with the other techniques that provide outlier ranking, we report two values of the F-measure. The first one is based on the top-n points in their ranking where n is the number of



(a) (b)

Figure 4: F-measure and ROC curve on CMUFace data.

Data	LOF	ABOD	SOD	Mgraph	OutDST
Letter	0.57 (0.53)	0.52 (0.50)	0.57 (0.53)	0.46	**0.63**
Vowel	0.80 (**0.87**)	0.80 (0.70)	0.50 (0.61)	0.64	**0.87**
Segm.	0.60 (0.57)	0.40 (0.48)	0.60 (0.57)	0.40	**0.86**

Table 1: F-measure on three UCI datasets. Values in brackets for LOF, ABOD and SOD are based on the same number of outliers as OutDST outputs while the remaining values are computed from the number of outliers reported in the ground-truth.

true outliers, and the second one (in brackets) is based on top-n points where n is the number of points detected by OutDST.

It can be observed for the Letter dataset that LOF, ABOD and SOD perform competitively with almost the same F-measures computed from the 23 top-ranked outliers. While ABOD achieves 53%, LOF and SOD perform slightly better with 57% and all of them agree on the top 7 outliers. None of them uncovers any true outlier between rank 23 through 26, yielding slightly decreased F-measures due to the decrement of precision. Though better than MGraph technique, their detection rates are still lower compared to that of OutDST, which achieves 63%. Looking deeper, when we increase the γ threshold, OutDST further uncovers 19 true outliers out of 29 that it outputs which makes both its precision and recall better. Compared to the top 29 ranked outliers from the other methods, we see that only ABOD can uncover one more true outlier in this range. For the Vowel dataset, we observe that only LOF performs competitively to OutDST since both achieve 87% in their F-measures. Nonetheless, OutDST's performance on the Segmentation data is much better than LOF and the other techniques. Returning 11 outliers, 9 of them are true outliers whereas only 6 true outliers are found in the top-ranked data points of LOF. Similarly, the F-measures of ABOD and SOD are respectively 48% and 60%, which is much lower compared to that of the proposed OutDST algorithm.

6 Conclusions

In this paper, we have presented OutDST, an algorithm that applies space transformation and uses spectral analysis for outlier detection. We proposed a novel concept of local quadratic entropy to quantify data similarity and exploit it to adaptively regularize the closeness amongst nearby data instances. Through spectral graph theory, we show that both global and local outliers can effectively be uncovered by spectral analysis of the eigenspace spanned by the set of leading eigenvectors induced in the transformation step. The presented algorithm is purely data-driven, placing no assumptions on the observed data, which makes it work well on data with various non-Gaussian shaped distributions, even with great variation in densities. We demonstrated its effectiveness against other techniques via a number of experiments on synthetic and real-world benchmark datasets.

For future work, we plan to extend our approach to incremental fashions to process streaming data (like network flows) where distributions may change with time, i.e. concepts drifting, and also its extension to other nonlinear transformation techniques.

References

[1] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[2] V. Barnett and T. Lewis. *Outliers in statistical data.* John Wiley & Sons Ltd., 3rd edition edition, 1994.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer-Verlag., 2006.

[4] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *SIGMOD*, 2000.

[5] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 2009.

[6] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2), 1973.

[7] G. Golub and C. Loan. *Matrix Computations.* The Johns Hopkins University Press, 3rd edition, 1996.

[8] L. Huang, X. Nguyen, M. Garofalakis, M. Jordan, A. Joseph, and N. Taft. In-network pca and anomaly detection. In *NIPS*, pages 617–624, 2006.

[9] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *VLDB*, 1998.

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

[11] H. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *SIGKDD*, 2008.

[12] B. Mikhail and N. Partha. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2001.

[13] E. Müller, M. Schiffer, and T. Seidl. Statistical selection of relevant subspace projections for outlier ranking. In *ICDE*, pages 434–445, 2011.

[14] J. Principe, D. Xu, and J. Fisher. *Information Theoretic Learning.* John Wiley & Sons, 2000.

[15] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *SIGMOD*, 2000.

[16] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection schem based on pricipal component classifier. In *ICDM*, 2003.

[17] G. W. Stewart and J. Sun. *Matrix Perturbation Theory.* Academic Press, 1990.

[18] Y. Tao, X. Xiao, and S. Zhou. Mining distance-based outliers from large databases in any metric space. In *SIGKDD*, 2006.

[19] T. Vries, S. Chawla, and M. E. Houle. Finding local anomalies in very high dimensional space. In *ICDM*, 2010.

[20] M. P. Wand and M. C. Jones. *Kernel Smoothing-Monographs on Statistics and Applied Probability.* Chapman & Hall/CRC, 1994.