Scalable Outlying-Inlying Aspects Discovery via Feature Ranking

Nguyen Xuan Vinh[1], Jeffrey Chan[1], James Bailey[1], Christopher Leckie[1],
Kotagiri Ramamohanarao[1], and Jian Pei[2]

[1]The University of Melbourne, Australia
{vinh.nguyen | jeffrey.chan | baileyj | caleckie | kotagiri}
@unimelb.edu.au; [2]Simon Fraser University, Canada, jpei@cs.sfu.ca

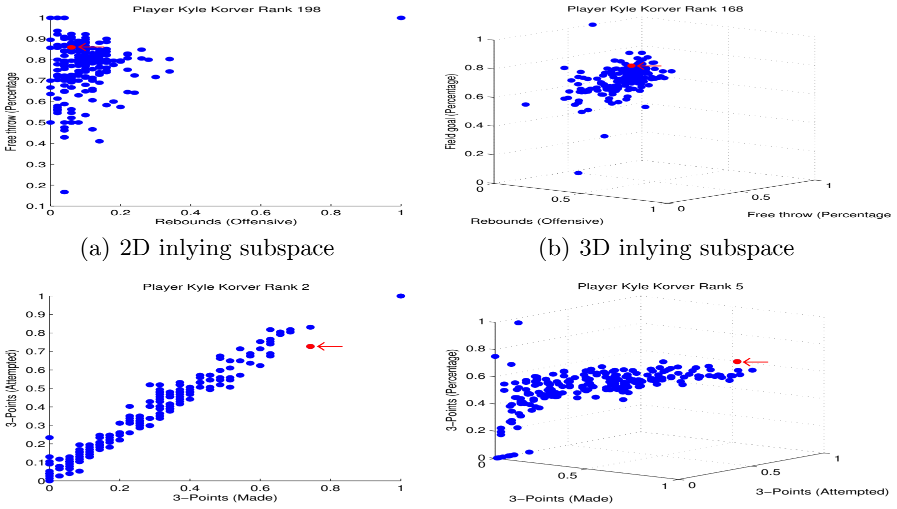
Abstract. In outlying aspects mining, given a query object, we aim to answer the question as to what features make the query most outlying. The most recent works tackle this problem using two different strategies. (i) Feature selection approaches select the features that best distinguish the two classes: the query point vs. the rest of the data. (ii) Score-and-search approaches define an outlyingness score, then search for subspaces in which the query point exhibits the best score. In this paper, we first present an insightful theoretical result connecting the two types of approaches. Second, we present **OARank** – a hybrid framework that leverages the efficiency of feature selection based approaches and the effectiveness and versatility of score-and-search based methods. Our proposed approach is orders of magnitudes faster than previously proposed score-and-search based approaches while being slightly more effective, making it suitable for mining large data sets.

Keywords: Outlying aspects mining, feature selection, feature ranking, quadratic programming

1 Introduction

In this paper, we are interested in the novel and practical problem of investigating, for a particular query object, the aspects that make it most distinguished compared to the rest of the data. In [5], this problem was coined *outlying aspect mining*, although it has also been known as *outlying subspaces detection* [15], *outlier explanation* [10], *outlier interpretation* [2], and *object explanation* [12]. Outlying aspects mining has many practical applications. For example, a home buyer would be highly interested in finding out the features that make a particular suburb of interest stand out from the rest of the city. A recruitment panel may be interested in finding out what are the most distinguishing merits of a particular candidate compared to others. An insurance specialist may want to find out what are the most suspicious aspects of a certain insurance claim. A natural complementary task to outlying aspects mining is *inlying aspects mining*, i.e., what features make the query most usual.

A practical example of outlying aspects mining is given in Fig. 1, where we present the outlying-inlying aspects returned by our proposed approach–**OARank**–for player Kyle Korver in the NBA Guards dataset (data details



(a) 2D inlying subspace

(b) 3D inlying subspace

(c) 2D outlying subspace

(d) 3D outlying subspace

Fig. 1. OARank inlying/outlying subspaces for NBA player Kyle Korver (red circle)

given in Section 4.1). NBA sports commentators are interested in the features that make a player most unusual (and maybe also usual). If we take the kernel density rank as an outlyingness measure, then it can be observed that in the top 2D and 3D-inlying subspaces, Kyle has very low density ranking (198th and 168th over 220 players respectively). The attributes in which Kyle appears most usual are "Rebound (Offensive)", "Free throw (Percentage)" and "Field goal (Percentage)". On the other hand, in the top outlying subspaces, Kyle has very high density rank (2nd and 5th over 220 players respectively). Kyle is indeed very good at 3-points scoring: "3-points (Attempted)", "3-points (Made)", and "3-points (Percentage)".

Outlying aspects mining (or outlier explanation) has a close relationship with the traditional task of outlier detection, yet with subtle but critical differences. In this context, we only focus on the query object, which itself may or may not be an outlier. It is also not our main interest to verify whether the query object is an outlier or not. We simply are interested in subsets of features (i.e., subspaces) that make it most outlying. We are also not interested in other possible outliers in the data, if any. In contrast, outlier detection aims to identify all possible outlying objects in a given data set, often without explaining why such objects are considered as outliers. Outlier explanation is thus a complementary task to outlier detection, but could be used in principle to explain any object of interest. Thus, in this paper, we shall employ the term *outlying aspects mining*, which is more generic than outlier explanation.

1.1 Related work

The latest work on outlying aspects mining can be categorized into two main directions, which we refer to as feature selection approaches [4], and score-and-search approaches [5].

- In *feature selection approaches* [10, 12], the problem of outlying aspects mining is first transformed into the classical problem of feature selection for classification. More specifically, the two classes are defined as the query point (positive class) and the rest of the data (negative class). In [4], to balance the classes, the positive class is over-sampled with samples drawn from a Gaussian distribution centred at the query point, while the negative class is under-sampled, keeping k full-space neighbors of the query point and some other data points from the rest of the data. Similarly in [12], the positive class is over-sampled while keeping all other data points as the negative class. The feature subsets that result in the best classification accuracy are regarded as outlying features and selected for user inspection.

  A similar approach to feature selection is feature transformation [2], which identifies a linear transformation that best preserves the locality around the neighborhood of the query point while at the same time distinguishing the query from its neighbors. Features with high absolute weights in the linear transformation are deemed to contribute more to the outlyingness of the query.
- In *score-and-search based methods*, a measure of outlyingness degree is needed. The outlyingness degree of the query object will be compared across all possible subspaces, and the subspaces that score the best will be selected for further user inspection. In [5], the kernel density estimate was proposed as an outlyingness measure. It is well known, however, that the raw density measure tends to be smaller for subspaces of higher dimensionality, as a result of increasing sparseness. For this reason, the rank statistic was used to calibrate the raw kernel density to avoid dimensionality bias. Having defined an outlyingness measure, it is necessary to search through all possible subspaces and enumerate the ones with lowest density rank. Given a dataset of dimension d, the number of subspaces is $(2^d - 1)$. If the user specifies a parameter d_{max} as the maximum dimensionality, then the number of subspaces to search through is in the order of $O(d^{d_{max}})$.

1.2 Contribution

In this paper, we advance the state of the art in outlying aspects mining by making two important contributions. First, we show an insightful theoretical result connecting the two seemingly different approaches of *density-based score-and-search* and *feature selection* for outlying aspects mining. In particular, we show that by using a relevant measure of mutual information for feature selection, namely the quadratic mutual information, density minimization can be regarded as contributing to maximizing the mutual information criterion. Second, as exhaustive search for subspaces is expensive, our most important contribution in this paper is to propose an alternative scalable approach, named **OARank**, in which the features are ranked based on their potential to make the query point having low density. The top-ranked features are then selected either for direct user inspection, or for a more comprehensive score-and-search with the best-scored subspaces then reported to the user. The feature ranking procedure takes only quadratic time in the number of dimensions and scales linearly w.r.t the number of data points, making it much more scalable and suitable for mining large and high dimensional datasets, where a direct enumeration strategy is generally infeasible.

2 Connection between density-based score-and-search and feature selection based approaches

In the feature selection approach, the problem of explaining the query is first posed as a two-class classification problem, in which we aim to separate the query $\mathbf{q}$ (positive class c_1) from the rest of the data $\mathbf{O}$ (negative class c_0) of n objects $\{\mathbf{o}_1, \dots, \mathbf{o}_n\}$, $\mathbf{o}_i \in \mathbb{R}^d$. Let $\mathbf{D} = \{D_1, \dots, D_d\}$ be a set of d numeric features (attributes). In the feature selection approaches [4, 12], to balance the class distribution, the positive class c_1 is augmented with synthetic samples from a Gaussian distribution centred at the query. The task is then to select the top features that distinguish the two classes. These features are taken as outlying features for the query.

We now show that there exists a particular feature selection paradigm which has a close connection to density based approaches. Let us form a two-class data set

$$\mathbf{X} = \{\underbrace{\mathbf{x}_1 \equiv \mathbf{o}_1, \dots, \mathbf{x}_n \equiv \mathbf{o}_n}_{c_0}, \underbrace{\mathbf{x}_{n+1} \equiv \mathbf{q}, \dots, \mathbf{x}_{2n} \equiv \mathbf{q}}_{c_1}\}$$

Note that here we have over-sampled the positive class simply by duplicating $\mathbf{q}$ n times, so that the classification problem is balanced. Mutual information based feature selection aims to select a subset of m features such that the information shared between the data and the class variable is maximized, i.e., $\max_{\mathbf{S} \subset \mathbf{D}, |\mathbf{S}|=m} I(\mathbf{X_S}; C)$, where $\mathbf{X_S}$ is the projection of the data onto the subspace $\mathbf{S}$ and C is the class variable. We will show that by using a particular measure of entropy coupled with the Gaussian kernel for density estimation, we arrive at a formulation reminiscent of density minimization. In particular, we shall make use of the general Havrda-Charvat's α-structural entropy [6], defined as:

$$H_\alpha(\mathbf{X}) = (2^{1-\alpha} - 1)^{-1} \left[\int f(\mathbf{x})^\alpha d\mathbf{x} - 1 \right], \ \alpha > 0, \alpha \neq 1 \qquad (1)$$

Havrda-Charvat's entropy reduces to Shanon's entropy in the limit when $\alpha \to 1$, hence it can be viewed as a generalization of Shannon's entropy [7, 9].

In order to make the connection, we shall make use of a particular version of Havrda-Charvat's entropy with $\alpha = 2$, also known as quadratic Havrda-Charvat's entropy $H_2(\mathbf{X}) = 1 - \int f(\mathbf{x})^2 d\mathbf{x}$ (with the normalizing constant discarded for simplicity).

Using the Gaussian kernel $G(\mathbf{x} - \mathbf{X}_i, \sigma^2) = (2\pi\sigma^2)^{-d/2} \exp(\frac{-\|\mathbf{x}-\mathbf{X}_i\|^2}{2\sigma^2}$, the probability density of $\mathbf{X}$ is estimated as $\hat{f}(\mathbf{x}) = \frac{1}{2n} \sum_{i=1}^{2n} G(\mathbf{x} - \mathbf{X}_i, \sigma^2)$. The quadratic entropy of $\mathbf{X}$ can be estimated as:

$$H_2(\mathbf{X}) = 1 - \frac{1}{(2n)^2} \int_{\mathbf{x}} \left(\sum_{i=1}^{2n} G(\mathbf{x} - \mathbf{X}_i, \sigma^2) \right)^2 d\mathbf{x} = 1 - \frac{1}{(2n)^2} \sum_{i=1}^{2n} \sum_{j=1}^{2n} G(\mathbf{X}_i - \mathbf{X}_j, 2\sigma^2)$$

wherein we have employed a nice property of the Gaussian kernel, which is that the convolution of two Gaussian remains a Gaussian [14]:

$$\int_{\mathbf{x}} G(\mathbf{x}-\mathbf{X}_i,\sigma^2)G(\mathbf{x}-\mathbf{X}_j,\sigma^2)\,d\mathbf{x} = G(\mathbf{X}_i-\mathbf{X}_j,2\sigma^2) \tag{2}$$

The conditional quadratic Havrda-Charvat's entropy of $\mathbf{X}$ given a (discrete) variable C is defined as $H_2(\mathbf{X}|C)=\sum_{k=1}^{K}p(c_k)H_2(\mathbf{X}|C=c_k)$. We have:

$$H_2(\mathbf{X}|C=c_0) = 1-\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}G(\mathbf{X}_i-\mathbf{X}_j,2\sigma^2)$$

$$H_2(\mathbf{X}|C=c_1) = 1-\frac{1}{n^2}\sum_{i=n+1}^{2n}\sum_{j=n+1}^{2n}G(\mathbf{X}_i-\mathbf{X}_j,2\sigma^2)$$

then $H_2(\mathbf{X}|C)=\frac{1}{2}H_2(\mathbf{X}|C=c_0)+\frac{1}{2}H_2(\mathbf{X}|C=c_1)$. Finally, the quadratic mutual information between $\mathbf{X}$ and C is estimated as:

$$I_2(\mathbf{X};C)=H_2(\mathbf{X})-H_2(\mathbf{X}|C)=\frac{1}{(2n)^2}(CC+CS)$$

where

$$CC=\sum_{i,j=1}^{n}G(\mathbf{X}_i-\mathbf{X}_j,2\sigma^2)+\sum_{i,j=n+1}^{2n}G(\mathbf{X}_i-\mathbf{X}_j,2\sigma^2)$$

$$CS=-2\sum_{i=1}^{n}\sum_{j=n+1}^{2n}G(\mathbf{X}_i-\mathbf{X}_j,2\sigma^2)$$

An interesting interpretation for the quadratic mutual information is as follows: the quantity $G(\mathbf{X}_i-\mathbf{X}_j,2\sigma^2)$ can be regarded as a measure of interaction between two data points, which can be called the *information potential* [3]. The quantity $\sum_{i,j=1}^{n}G(\mathbf{X}_i-\mathbf{X}_j,2\sigma^2)$ is the total strength of intra-class interaction within the negative class and $\sum_{i,j=n+1}^{2n}G(\mathbf{X}_i-\mathbf{X}_j,2\sigma^2)$ is the total strength of intra-class interaction within the positive class (within-class total information potential), thus CC is a measure of *class compactness.* On the other hand, $\sum_{i=1}^{n}\sum_{j=n+1}^{2n}G(\mathbf{X}_i-\mathbf{X}_j,2\sigma^2)$ measures inter-class interaction (cross-class information potential), thus CS is a measure of *class separability.* For maximizing $I_2(\mathbf{X},C)$, we aim to maximize intra-class interaction while minimizing inter-class interaction.

Theorem 1. *Density minimization is equivalent to maximization of class separability in quadratic mutual information based feature selection.*

Proof. Note that since the positive class contains only $\mathbf{q}$ (duplicated n times), we have

$$CS=-2n\sum_{i=1}^{n}G(\mathbf{X}_i-\mathbf{q},2\sigma^2)=-2n^2\hat{f}(\mathbf{q}),$$

where $\hat{f}(\mathbf{q}) = \frac{1}{n}\sum_{i=1}^{n} G(\mathbf{X}_i - \mathbf{q}, 2\sigma^2)$ is nothing but the kernel density estimate of $\mathbf{q}$. Thus, it can be seen that minimizing the density of $\mathbf{q}$ is equivalent to minimizing inter-class interaction (cross-class information potential), or equivalently maximizing class separability. □

This theoretical result shows that there is an intimate connection between density-based score-and-search and feature selection based approaches for outlying aspects mining. Minimizing the density of the query will contribute to maximizing class separation, and thus maximizing the mutual information criterion. This insightful theoretical connection also points out that the mutual-information based feature selection approach is more comprehensive, in that it also aims to maximize the class compactness. The relevance of class-compactness to outlying aspects mining is yet to be explored.

3 Outlying Aspects Mining via Feature Ranking

We now present the main contribution of this paper—**OARank**—a hybrid approach for outlying aspects mining that leverages the strengths of both the feature selection and the score-and-search paradigms. This is a two-stage approach. In the first stage, we rank the features according to their potential to make the query outlying. In the second (and optional) stage, score-and-search can be performed on a smaller subset of the top-ranked $m \ll d$ features.

3.1 Stage 1: OARank–Outlying Features Ranking

We aim to choose a subset of m features $\mathbf{S} \subset \mathbf{D}$ such that the following criterion is minimized:

$$\mathcal{SS}: \min_{\substack{\mathbf{S}\subset\mathbf{D}\\ |\mathbf{S}|=m}} \left\{ C(m) \sum_{i=1}^{n} \sum_{\substack{t,j\in\mathbf{S}\\ t<j}} K(\mathbf{q}_j - \mathbf{o}_{ij}, h_j) K(\mathbf{q}_t - \mathbf{o}_{it}, h_t) \right\} \tag{3}$$

where $K(x-\mu, h) = (2\pi h^2)^{-1/2} \exp\{-(x-\mu)^2/2h^2\}$ is the one dimensional Gaussian kernel with bandwidth h and center μ, and $C(m) = \frac{2}{nm(m-1)2^{m-2}}$ is a normalization constant.

We justify this *subset selection* ($\mathcal{SS}$) objective function as follows: *the objective function in* $\mathcal{SS}$ *can be seen as a kernel density estimate at the query point* $\mathbf{q}$. To see this, we first develop a novel kernel function for density estimation, which is the sum of 2-dimensional kernels. Herein, we employ the Gaussian product kernel recommended by Scott [11], defined as:

$$K(\mathbf{q} - \mathbf{o}_i, \mathbf{h}) = \frac{1}{(2\pi)^{d/2} \prod_{j=1}^{d} h_j} \prod_{j=1}^{d} \exp\left(\frac{-(\mathbf{q}_j - \mathbf{o}_{ij})^2}{2h_j^2}\right) \tag{4}$$

where h_j's are the bandwidth parameters in each individual dimension. We note that, in the product kernel (4), a particular dimension (feature) can be 'de-emphasized', by assigning its corresponding 1D-kernel to be the 'uniform' kernel, or more precisely, a rectangular kernel with sufficiently large bandwidth σ_u, i.e.,

$$K_u(\mathbf{q}_j - \mathbf{o}_{ij}, \sigma_u) = \begin{cases} 1/(2\sigma_u) \text{ if } \frac{|\mathbf{q}_j - \mathbf{o}_{ij}|}{\sigma_u} \leq 1 \\ 0 \text{ otherwise} \end{cases} \tag{5}$$

Note that σ_u can be chosen to be arbitrarily large, so that we can assume that any query point of interest $\mathbf{q}$ will lie within σ_u-distance from any kernel center $\mathbf{o}_i$ in any dimension. In our work, we normalize the data (including the query) so that $\mathbf{o}_{ij} \in [-1, 1]$ in any dimension, thus σ_u could simply be chosen as $\sigma_u = 1$. From the product kernel (4), if we de-emphasize all dimensions, but keeping only two 'active' dimensions $\{t, j\}$, then we obtain the following d-dimensional kernel:

$$K_{tj}(\mathbf{q} - \mathbf{o}_i, \mathbf{h}) = \frac{1}{(2\sigma_u)^{d-2}} K(\mathbf{q}_j - \mathbf{o}_{ij}, h_j) K(\mathbf{q}_t - \mathbf{o}_{it}, h_t) \tag{6}$$

Averaging over all $d(d-1)/2$ pairs of dimensions yields the following kernel:

$$K(\mathbf{q} - \mathbf{o}_i, \mathbf{h}) = \frac{2}{d(d-1)(2\sigma_u)^{d-2}} \sum_{j<t}^{d} K_{tj}(\mathbf{q} - \mathbf{o}_i, \mathbf{h}) \tag{7}$$

Theorem 2. *The kernel function $K(\mathbf{q} - \mathbf{o}_i, \boldsymbol{h})$ as defined in* (7) *is a proper probability density function.*

Proof. It is straightforward to show that $K(\mathbf{q} - \mathbf{o}_i, \mathbf{h}) \geq 0$ and $\int_{\mathbb{R}^d} K(\mathbf{q} - \mathbf{o}_i, \mathbf{h}) = 1$ □

Employing this new kernel to estimate the density of the query point $\mathbf{q}$ in a subspace $\mathbf{S} \subset \mathbf{D}$, we obtain exactly the objective function in $\mathcal{SS}$, which when minimized will minimize the density at the query $\mathbf{q}$, i.e., making $\mathbf{q}$ most outlying.

3.2 Solving the Outlying-Inlying Aspects Ranking Problem

The subset selection problem $\mathcal{SS}$ can be equivalently formulated as a quadratic integer programming problem as follows:

$$\min_{\mathbf{w}} \left\{ \mathbf{w}^T \mathbf{Q}' \mathbf{w} \right\} \text{ s.t. } w_i \in \{0, 1\}, \sum w_i = m \tag{8}$$

where $\mathbf{Q}'_{tj} = \sum_{i=1}^{n} \frac{1}{h_j} K_j(\frac{\mathbf{q}_j - \mathbf{o}_{ij}}{h_j}) \times \frac{1}{h_t} K_t(\frac{\mathbf{q}_t - \mathbf{o}_{it}}{h_t}),\ t \neq j$ and $\mathbf{Q}'_{tt} = 0$. Equivalently, we can rewrite it in a maximization form as

$$\mathcal{QIP} : \max_{\mathbf{w}} \left\{ \mathbf{w}^T \mathbf{Q} \mathbf{w} \right\} \text{ s.t. } w_i \in \{0, 1\}, \sum w_i = m \tag{9}$$

where $\mathbf{Q}_{tj} = \Phi - \mathbf{Q}'_{tj}$ and $\Phi = \max_{i,j} \mathbf{Q}'_{ij}$. While (8) and (9) are equivalent, the Hessian matrix $\mathbf{Q}$ is entry-wise non-negative, a useful property that we will exploit shortly. The parameter m specifies the size of the outlying subspace we wish to find. It is noted that $\mathcal{SS}$ and $\mathcal{QIP}$ are not monotonic with respect to m, i.e., with two different m values, the resulting outlying subspaces are not necessarily subsets of one another.

As $\mathcal{QIP}$ is well known to be NP-hard [1], we relax the problem to the real domain, as follows. Note that with $w_i \in \{0,1\}, \sum w_i = m$, we also have $\|\mathbf{w}\|_2 = \sqrt{m}$. We shall now drop the integral 0-1 constraint, which in fact causes NP-hardness, while keeping the norm constraint:

$$\max_{\mathbf{w}} \left\{\mathbf{w}^T\mathbf{Q}\mathbf{w}\right\} \text{ s.t. } \|\mathbf{w}\|_2 = \sqrt{m},\ w_i \geq 0 \tag{10}$$

The additional non-negativity constraints $w_i \geq 0$ ensure that the relaxed solution can be reasonably interpreted as feature 'potential' in making $\mathbf{q}$ outlying. Also note that we can replace $\|\mathbf{w}\|_2 = \sqrt{m}$ with $\|\mathbf{w}\|_2 = 1$ without changing the optimal relative weight ordering (all the weights w_i are scaled by the same multiplicative constant $1/\sqrt{m}$). Thus, we arrive at

$$\mathcal{QP} : \max_{\mathbf{w}} \left\{\mathbf{w}^T\mathbf{Q}\mathbf{w}\right\} \text{ s.t. } \|\mathbf{w}\|_2 = 1,\ w_i \geq 0 \tag{11}$$

Observe that since $\mathbf{Q}_{ij} \geq 0$, the solution to this problem is simple and straightforward: it is the dominant eigenvector associated with the dominant eigenvalue of the Hessian matrix $\mathbf{Q}$ [13]. Note that with this relaxation scheme, the parameter m has been eliminated, thus OARank will produce a single ranking. The outcome of this quadratic program can be considered as feature potentials: features that have higher potentials contribute more to the outlyingness of the query. Features can be ranked according to their potentials. The top-ranked m features will be chosen for the next score-and-search stage.

Note that an interesting novel by-product of this ranking process is the *inlying aspects*, i.e., features with the lowest potentials. These inlying aspects are features (subspaces) in which *the query point appears to be most usual.*

3.3 Stage 2: OARank+Search

Having obtained the feature ranking in stage 1, there are two ways to proceed: (i) One can take the top k-ranked features as the single most outlying subspace of size k. A more flexible way is to (ii) perform a comprehensive score-and-search on the set of top-ranked $m \ll d$ features, and report a list of top-scored subspaces for user inspection. The search on the filtered feature set is however much cheaper than a search in the full feature space.

3.4 Complexity analysis

Ranking stage: The cost of building the Hessian matrix $\mathbf{Q}$ is $O(d^2n)$. The cost for finding the dominant eigenvector of $\mathbf{Q}$ is $O(d^2)$. Thus overall, the complexity of the ranking process is $O(d^2n)$.
Score-and-Search stage: If we employ the density rank measure, the cost for scoring (i.e., computing the density rank for the query) in a single subspace is $O(n^2)$ time. The number of subspaces to score is $O(2^m - 1)$ for exhaustive search, or $O(m^{d_{max}})$ if a maximum subspace size is imposed. Note that for practical applications, we would prefer the subspaces to be of small dimensionality for improved interpretability, thus it is practical to set, for example, $d_{max} = 5$. The overall complexity of this stage is $O(n^2m^{d_{max}})$

Overall, the complexity of both stages is $O(d^2n + n^2m^{d_{max}})$ for the proposed two-stage approach. For comparison, a direct density rank based score-and-search approach on the full space costs $O(n^2d^{d_{max}})$, which is infeasible when d is moderately large.

Techniques for further improving scalability: While the ranking phase of OARank is generally very fast, the search phase can be slow, even on the reduced feature set. To further speed up the search phase, one can further prune the search space. In particular, the search space can be explored in a stage-wise manner, expanding the feature sets gradually. In exhaustive search, every feature set of size k will be expanded to size $k + 1$ by adding 1 more feature. We can improve the efficiency of this process, sacrificing comprehensiveness, by only choosing a small subset of most promising subspaces (i.e., highly-scored) of size k to be expanded to size $k + 1$.

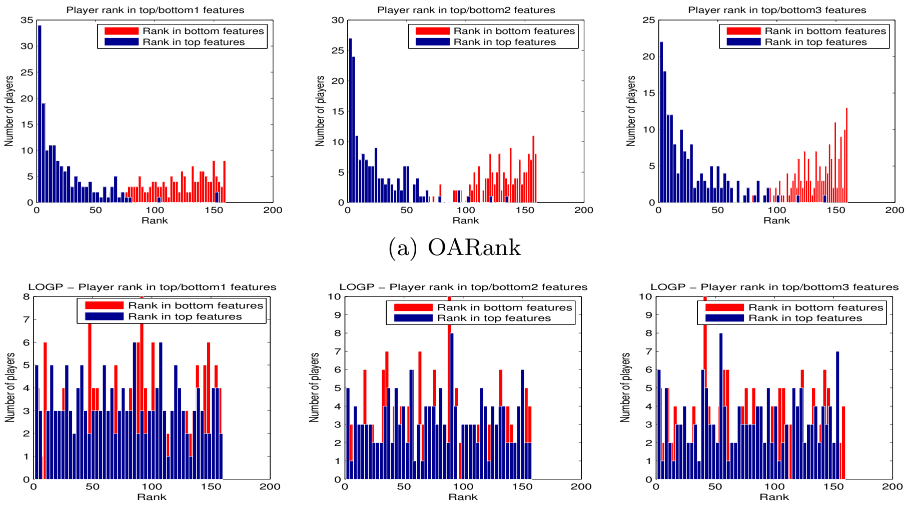
4 Experimental Evaluation

In this section, we experimentally evaluate the proposed approaches, **OARank** and **OARank+Search**. We compare our approaches with the density rank based approach in [5] and Local Outlier with Graph Projection (LOGP) [2]. LOGP is the closest method in spirit to OARank, in that it also learns a set of weights: features with higher weights contribute more to the outlyingness of the query. These weights are from a linear transformation that aims to separate the query from its full-space neighborhood. LOGP was proposed as a method for both detecting and explaining outliers. We implemented all methods in Matlab/C++ except LOGP for which the Matlab code was kindly provided by the authors. The parameters for all methods were set as recommended in the original articles [5, 2]. The bandwidth parameter for OARank was set according to [5], i.e., $h = 1.06 \min\{\sigma, \frac{R}{1.34}\}n^{-\frac{1}{5}}$ with σ being the standard sample deviation, and R being the difference between the first and third quartiles of the data distribution. In order to improve the scalability of score-and-search based methods, we apply a stage-wise strategy as discussed in Section 3.4 where only at most 100 top-scored subspaces are chosen for expansion at each dimension $k < d_{max} = 5$. All experiments were performed on an i7 quad-core PC with 16Gb of main memory. The source code for our methods will be made available via our website.

4.1 The NBA data sets

We first test the methods on the NBA data available at `http://sports.yahoo.com/nba/stats`. This data set was previously analyzed in [5], where the authors collected 20 attributes for all NBA guards, forwards and centers in the 2012-2013 season, resulting in 3 data sets. We compare the quality of the ranking returned by OARank and LOGP. More specifically, for each player, we find the top 1, 2 and 3 inlying and outlying features, and then compute the density rank for the player in his outlying-inlying subspaces.

The results of this analysis on the NBA Forwards data set are presented in Fig. 2. It can be clearly seen that OARank is able to differentiate between inlying and outlying aspects. More precisely, in the outlying subspaces (of the



(a) OARank

(b) LOGP

Fig. 2. Outlying Feature Ranking: OARank vs. LOGP on NBA Forwards data set (best viewed in color)

top-ranked 1/2/3 features), all players tend to have higher density rank than their ranks in the inlying subspaces (of the bottom-ranked 1/2/3 features). On the same data set, LOGP ranking does not seem to differentiate well between outlying and inlying features. In particular, the rank distribution appears to be uniform in both inlying and outlying subspaces. Thus, in this experiment, qualitatively we can see that OARank is more effective at identifying inlying-outlying aspects. The same conclusion applies for the NBA Guards and Centers data sets, for which we do not provide detailed results due to space restrictions. We have seen the detailed analysis for a specific player, Kyle Korver, in Figure 1. The feature weights and ranking returned by OARank for Kyle Korver can be inspected in Fig. 3(e).

4.2 Mining non-trivial outlying subspaces

For a quantitative analysis, we employ a collection of data sets proposed by Keller et al. [8] for benchmarking subspace outlier detection algorithms. This collection contains data sets of 10, 20, 30, 40, 50, 75 and 100 dimensions, each consisting of 1000 data points and 19 to 136 outliers. These outliers are challenging to detect, as they are only observed in subspaces of 2 to 5 dimensions but not in any lower dimensional projection. We note again that our task here is not outlier detection, but to explain why the annotated outliers are designated as such. For this data set, since the ground-truth (i.e., the outlying subspace for each outlier) is available as part of Keller et al.'s data, we can objectively evaluate the performance of all approaches. Let the true outlying subspace be T and the retrieved subspace be P. To evaluate the effectiveness of the algorithms, we employ the Jaccard index $Jaccard(T, P) \triangleq |T \cap P|/|T \cup P|$, and the precision, $precision \triangleq |T \cap P|/|P|$. The average Jaccard index and precision over all outliers for different approaches on all datasets are reported in Figure 3(a,b).

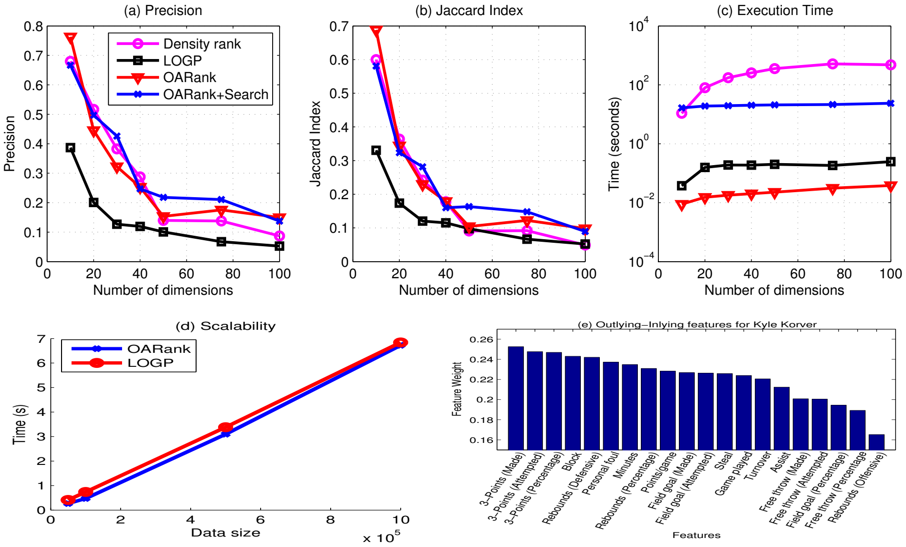


Fig. 3. (a)-(c): Performance on identifying non-trivial outlying high-dimensional subspaces; (d) Scalability; (e): OARank feature weights for Kyle Korver

We can observe that OARank and the density based score-and-search approach both outperform LOGP. OARank (without search) obtains relatively good results, slightly better than density rank score-and-search at higher dimensions. OARank (with search) did not seem to improve the results significantly on this data set. In terms of execution time (Fig. 3c), OARank is the fastest, being orders of magnitude faster than density rank score-and-search. It can also be observed that the OARank+Search approach admits a near-flat time complexity profile with regards to the number of dimensions. This agrees well with the theoretical complexity of $O(d^2n)$ for ranking and $O(n^2m^{d_{max}})$ for search. On these data sets, the ranking time was negligible compared to search time, while the search complexity of $O(n^2m^{d_{max}})$ is independent of dimensionality.

4.3 Scalability

We tested the method on several large datasets. We pick the largest of Keller's data sets of 1000 points in 100 dimensions, and introduce more synthetic examples by drawing points from a Gaussian distribution centred at each data points, resulting in several large data sets of size ranging from 50,000 to 1 million data points. The run time for OARank and LOGP is presented in Figure 3(d). It is noted that for these large datasets, the search phase using the density score is computationally prohibitive, due to quadratic complexity in data size n. Both LOGP and OARank deal well with large datasets, with linear time complexity in the number of data points. This observation matches well with OARank's theoretical complexity of $O(d^2n)$ and demonstrates that OARank is capable of handling large data sets on commodity PCs.

We shall note that another prominent feature of OARank is that it is suitable for applications on streaming data: as data come in, entries in the Hessian matrix $\mathbf{Q}$ can be updated gradually. Feature weights can also be updated on-the-fly in real time, given that there exist very efficient algorithms for computing the dominant eigenvector of symmetric matrices.

5 Conclusion

In this paper, we have made two important contributions to the outlying aspects mining problem. First, we have made an insightful connection between the density based score-and-search and the mutual information based feature selection approach for outlying aspects mining. This insight can inspire the development of further hybrid approaches, which leverage the strengths of both paradigms. Second, we proposed OARank, an efficient and affective approach for outlying aspects mining, which is inspired by the feature ranking problem in classification. We show that OARank is suitable for mining very large data sets.

Acknowledgments: This work is supported by the Australian Research Council via grant numbers FT110100112 and DP140101969.

References

1. Art W. Chaovalitwongse et al. Quadratic integer programming: complexity and equivalent forms quadratic integer programming: Complexity and equivalent forms. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *Encyclopedia of Optimization*, pages 3153–3159. 2009.
2. Xuan Hong Dang et al. Discriminative features for identifying and interpreting outliers. In *ICDE 2014*, pages 88–99, March 2014.
3. Xuan-Hong Dang and James Bailey. A hierarchical information theoretic technique for the discovery of non linear alternative clusterings. In *16th ACM SIGKDD*, pages 573–582, New York, NY, USA, 2010. ACM.
4. XuanHong Dang, Barbora Micenkova, Ira Assent, and Raymond T. Ng. Local outlier detection with interpretation. In ECML '2013, pages 304–320, 2013.
5. Lei Duan, Guanting Tang, Jian Pei, James Bailey, et al., Mining outlying aspects on numeric data. In *Data Mining and Knowledge Discovery*, 2014, in press.
6. J. Havrda and F. Charvat. Quantification method of classification processes. concept of structural α-entropy. *Kybernetika*, (3):30–35, 1967.
7. Jawaharlal Karmeshu. *Entropy Measures, Maximum Entropy Principle and Emerging Applications.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
8. Fabian Keller, Emmanuel Muller, and Klemens Bohm. HiCS: High Contrast Subspaces for Density-Based Outlier Ranking. In ICDE '12, pages 1037–1048.
9. A.M. Mathai and H.J. Haubold. On generalized entropy measures and pathways. *Physica A: Statistical Mechanics and its Applications*, 385(2):493 – 500, 2007.
10. Barbora Micenkova, Raymond T. Ng, Ira Assent, and Xuan-Hong Dang. Explaining outliers by subspace separability. In *ICDM*, 2013.
11. D.W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* Wiley, 1992.
12. Nguyen Xuan Vinh, Jeff Chan, and James Bailey. Reconsidering mutual information based feature selection: A statistical significance view. In *AAAI-14*, 2014.
13. Nguyen Xuan Vinh, Jeff Chan, Simone Romano, and James Bailey. Effective global approaches for mutual information based feature selection. In *KDD'14*, 2014.
14. Nguyen Xuan Vinh and J. Epps. mincentropy: A novel information theoretic approach for the generation of alternative clusterings. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 521–530, 2010.
15. Ji Zhang, Meng Lou, et al., Hos-miner: A system for detecting outlyting subspaces of high-dimensional data. In VLDB '04, pages 1265–1268.