CVME Seminar

Outlier Detection

Dominik Schörkhuber

Vienna University of Technology, May 2015

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Introduction

Outlier Detection is the identification of data which is not conform to an expected pattern. Unlike in clustering, where items of common behavior are grouped together, and outliers may be seen as unwanted noise, we are actively searching with outlier detection methods for anomalies rather than common patterns in data. Outlier detection is a heavily researched topic and has in its generality a wide range of applications. Some examples are fault detection in safety critical systems, supervision of banking systems, video surveillance or intrusion detection in cyber security. In all those cases we want to automatically distinguish usual behavior from abnormal behavior, which may indicate a system breakdown, fraud, etc. Outliers may also be of use in a more positive way. For example human resource management may use outlier detection techniques to find people capable of a very diverse set of skills. From those examples we can conclude that in many cases abnormal data may be of great interest instead of data following a common pattern. Until now the very generic term data was used, we now want to closer specify how input data for outlier detection algorithms may look like. A single data instance consists of multiple attributes, similar to descriptors in image analysis. Data attributes can be classified in different types of data like binary, categorial, discrete or continous data. Each data instance may contain one (univariate) or multiple (multivariate) attributes. In the multivariate case the attribute types may be coherent or different among attributes. Consider Fig. 1 as an abstract example with input data in the attribute space $\mathbb{R}^2$. Each data point consists of two attributes $p = (x, y)$ with $x, y \in \mathbb{R}$. The data is split into several regions. N_1 and N_2 refer to normal regions, those points describe normal behavior. O_1 and O_2 are single outliers, O_3 is an outlying region.

1.1 Types of Supervision

Outlier detection methods can be classified by their type of supervision. Besides the actual data instances an outlier detection algorithm may also use additional information for distinction of outliers from inliers. Such information can be class labels as they are used in pattern matching algorithms. The data instances of a training dataset is augumented with class information. This information enables us to generate a predictive model which may classify data instances of test data. Depending on how much a method utilizes this additional information we distinguish three classes of supervision.
Supervised methods make heavy use of labeling information. These techniques require class (labeling) information for each normal and outlier point in the training dataset. Typically a predicitive model is built which enables the algorithm to classify further data

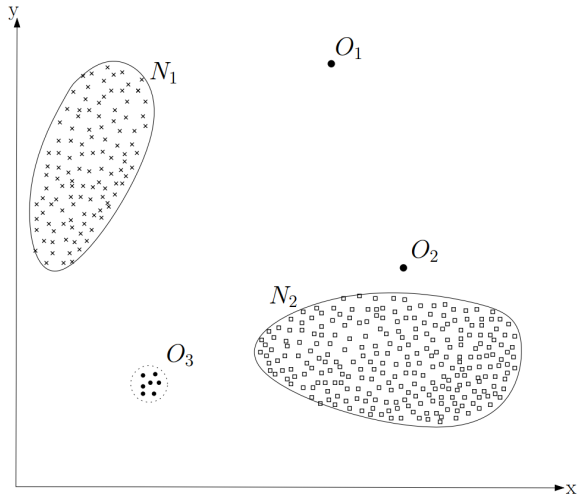

Abbildung 1: Example for Outlier Detection

points into normals or outliers. The main drawback of supervised methods is it may be very expensive to acquire correctly labeled training datasets. Depending on the field of use a human expert may be required to do a correct labeling or atleast correct a given automatic labeling of training data.
Semi-supervised methods on the other hand only require labeling information for one class of points. Either only normal or abnormal behavior is captured by the training set. It is often very difficult to provide correct labels for both classes of points. For example in a network intrusion detection system it is usually impossible to model all possible attack vectors as outliers in a training dataset. Whereas information of normal use of the network may be automatically acquired. For this reason unsupervised methods with known sets of outliers are not very popular. Since normal behavior is easier to model techniques with known normal datasets are usually used.
Unsupervised methods as a third method does not rely on any additional information. Therefore these methods are widely applicable since no additional information must be provided. Still we need some way of categorising input data points. For example statistic methods can be used to adapt a parametric distribution to normal and/or abnormal data. Based on a statistical test we may distinguish inliers from outliers. Also many techniques rely on the fact that normal data instances are occuring more frequently than abnormal data instances. Therefore frequently occuring patterns are classified as normal, whereas rare patterns are assumed to be abnormal.

1.2 Results of Outlier Techniques

Another important aspect of an outlier detection technique is the manner in which an outlier is reported. In section 1 we refered to training data as labeled input data. Labeled either as normal data instance or as an outlier. For outputs of outlier detection techniques we distinguish two classes. The first class are labeling outlier techniques. Those techniques put binary labels on each point identifying them as outliers or inliers, just as we did before for training data. Often these techniques are also called hard classifiers [4]. On the other hand there are scoring outlier detection techniques, which are also called soft classification techniques. Scoring techniques determine an outlier score which indicates the outlierness of a point. Which is a continuous value ranging e.g. from being an inlier (0) to being an outlier(1). Scoring techniques enable us to create a ranking among data instances. Through this ranking we can find the most outlying point or based on a threshold we can also compute binary labels for results of a scoring approach.

global vs local TypeI TypeII outliers

eventually more basic terms to describe if needed

2 Density-Based Local Outliers

For a first method we will look into LOF [5]. This method uses the local densities of data instances to describe its outlierness. Therefore we can classify LOF as a scoring method. This score is called the *local outlier factor*. LOF is local in the sense that it uses the degree of isolation of objects depend on the distances to the local neighbours of a data instance. This local procedure has an import benefit in contrast to other global methods.

Consider Fig 2 as an example showing be difference between local and global methods. For a simple global method we first compute all distances $d(p, q)$ between data instances. From a fixed point p we check distances to all other data instances. We count the number of distances being greater than *dmin* a fixed (global) threshold. If this number is greater than *pct* a certain percentage of points we define this point as an outlier. Applied to the dataset in Fig 2 we need *dmin* to be big enough such that data instances in $C1$ are not recognised as outliers. Since the distances between points in $C2$ are much smaller all points are correctly identified to be inliers. Also $O1$ is correctly identified as an outlier, but $O1$ is not. This is because the distance value *dmin* was chosen globally to fit data points in $C1$. See [5] for further details on this example. Using density based local outliers we are able to surpass these shortcomings.

The outlier definition for LOF is based on the k-distance for each data instance. The k-distance is defined to be the distance to the kth-nearest-neighbour. Further LOF defines

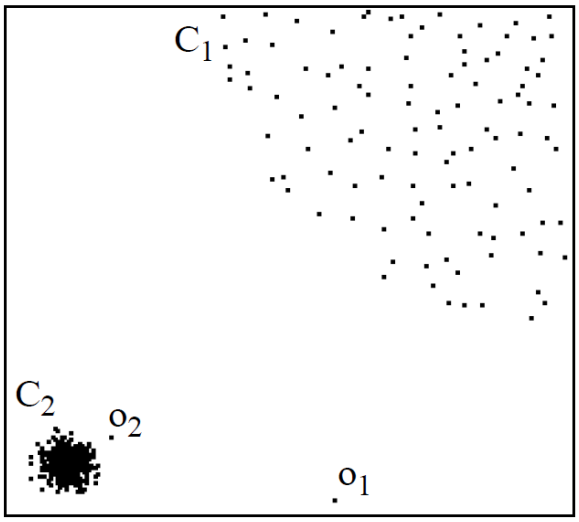

Abbildung 2: Global vs. Local Outlier Detection

the reachability distance from p to q to be the maximum of the distance from p to q and the k-distance of p. For each data instance the local reachability density (lrd) is computed by summing up the reachability distances from its k-nearest-neighbours and normalizing it. To compute the final LOF factor for a point p the local reachability densities from the k-neighbourhood are summed and normalized by the lrd of p.

$$lrd(p) = \left(\frac{\sum_{o \in N_k(p)} reachdist(p, o)}{|N_k(p)|}\right)^{-1} \tag{1}$$

where $N_k(p)$ is the set of k-nearest-neighbors

$$LOF(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd(o)}{lrd(p)}}{|N_k(p)|} \tag{2}$$

See [5] for equations and further details. The defined LOF equation computes the factor of outlierness.

3 Outlier Detection in High Dimensional Data[2]

Outlier detection has applications in numerous fields, including fraud detection, network

add info from [10]

analysis and pattern recognition. Many of these fields usually work with high dimensional data. But with high dimensional data also problems arise. Many outlier detections rely on distance metrics (L1-,L2-norm) which become meaningless in higher dimensions. In the last section we looked at LOF, which also suffers from these kind of problems. Since high dimensional data is often used it is meaningful to further investigate those problems. In this section we will describe the upcoming problems for high dimensional datasets.

3.1 Distance Metrics in High Dimensional Spaces

An important problem not only in outlier detection but also in other fields like machine learning, statistics and optimization is the meaningfulness of distance metrics in higher dimensions. As a prominent example remember the LOF algorithm which has been previously investigated. According to Aggarwal et. al. [1] the expressiveness of the L_k norm vanishes with a high number of dimensions. More specifically it is shown that if $\lim_{d\to\infty} var(\frac{\|X_d\|_k}{E[\|X_d\|_k]}) = 0$ then $\frac{Dmax_d^k - Dmin_d^k}{Dmin_d^k} \to 0$. Which means that the expressiveness gets lower with an increasing number of dimensions. In other words the variance approaches zero, and so does the the distance measure between a maximum and a minimum point in a dataset. Further [1] shows that in a high dimensional space the contrast between $Dmax_d^k - Dmin_d^k$ increases with $d^{1/k-1/2}$. For the manhattan distance this approaches zero, and for the euclidean distance it's constantly 1. For any $k > 0$ we approach infinity. For the selection of a metric this means that in higher dimensions L_k norms with lower k are more expressive.

3.2 Meaningful Dimensions

describe more problems from [10]

4 Angle Based Outlier Detection

We previously investigated the problems arising with distance metrics in high dimensional spaces. Since these metrics become very inaccurate with an increasing number of dimensions we would like to avoid them. Angle based outlier detection is one method to successfully avoid flawed distance metrics. Instead of distances we look at angles of vectors from a viewed point A to any other pair of points B, C. Angles between the computed vectors are accumulated to a spectrum of angles. While a broad spectrum respectively a high variance in the angle values denotes a point inside a cluster, a point with a small spectrum of angles is identified to be an outlier. This is an intuitive method because inside a cluster of points each point is surrounded by many neighbouring points

in all directions. Approaching the boarder of a cluster the spectrum of angles becomes smaller, and leaving the cluster continuously narrows the spectrum. Since angles between closer points B, C to A should have more impact on its outlierness, also a weighting factor is introduced. Which weights the importance of angles according to the distance of the currently computed points p, l. Finally the ABOD outlier factor is computed as the variance of angles between any two vector pairs divided by the product of the squared vector lengths [7].

$$ABOD(\vec{A}) = VAR_{\vec{B},\vec{C}}\left(\frac{\langle\overline{AB}, \overline{AC}\rangle}{\|\overline{AB}\|^2 \cdot \|\overline{AC}\|^2}\right)$$

Although Aggarwal et. al. in [1] showed the poor performance of L-norms in high dimensional spaces these are used in ABOD for weighting. But since ABOD only uses the distance metrics as a secondary criterion besides the variance of angles, ABOD still performs well in high dimensional spaces. Another important aspect of ABOD is that it is completely parameter free, while many other algorithms need to be tweaked by its parameters to generate reasonable outcome. Still a main drawback is the time complexity which is $O(n^3)$, while LOF only has a time complexity of $O(k \cdot n^2)$.

4.1 Fast ABOD

Because the time complexity of the original ABOD algorithm is exceptionally bad Kriegl et. al. [7] improved ABOD to FastABOD. As mentioned previously ABOD relies on the computation of angles between vectors on a specific point weighted by distances. Since the angles are weighted quadratically by distance near points have a much higher impact on the final result, while further points have only small effects. Therefore FastABOD only uses points of high importance to generate an approximative approach. It is easily shown that the points with highest importance are equal to the set of k-nearest-neighbours similarly used in LOF. This approach lowers the time complexity from $O(n^3)$ to $O(n \cdot k + n^2)$, and also introduces the parameter k which controls the quality of approximation. Although distance metrics are only used as a secondary criterion in FastABOD results in high dimensional spaces are not satisfying. It shows that also the quality of selection of k-nearest-neighbours depends on the number of dimensions, and worsens with increasing number of dimensions [7]. Experiments show good performance up to a hundred dimensions.

4.2 FastVOA

Based on the work on ABOD Phag et. al. further explored the use of angle based outlier detection in high dimensional spaces. Their target was to further improve the cubic/quadratic time complexity and improve behaviour in high dimensional spaces. Experiments from Kriegl et. al. show that the combined use of angles and distances reduces the curse of dimensionality, but the problem is still present. As for ABOD the basis for this algorithm is the variance of angles (VOA). Through random hyperplane projections and AMS Sketches unbiased estimators for the first and second moments of VOA are computed. See [9] for further details. The runtime complexity of FastVOA is denoted by $O(s_1 s_2 nlogn)$ which is nearly linear instead of quadratic complexity in FastABOD. $s_1 s_2$ is the number of AMS sketches. Also FastVOA is easily parallelizable.

5 Evolutionary Outlier Detection [2]

6 Outlier Detection with Ensembles [8]

ensembles

image processing aggarwal p19

[2] [3] [4] [5] [6] [10] [1] [8]

Literatur

[1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. *On the surprising behavior of distance metrics in high dimensional space*. Springer, 2001.

[2] Charu C Aggarwal and Philip S Yu. Outlier detection for high dimensional data. In *ACM Sigmod Record*, volume 30, pages 37–46. ACM, 2001.

[3] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *Principles of data mining and knowledge discovery*, pages 15–27. Springer, 2002.

[4] Irad Ben-Gal. Outlier detection. In *Data Mining and Knowledge Discovery Handbook*, pages 131–146. Springer, 2005.

[5] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Outlier detection: A survey. *ACM Computing Surveys*, 2007.

[7] Hans-Peter Kriegel, Arthur Zimek, et al. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452. ACM, 2008.

[8] Hoang Vu Nguyen, Hock Hee Ang, and Vivekanand Gopalkrishnan. Mining outliers with ensemble of heterogeneous detectors on random subspaces. In *Database Systems for Advanced Applications*, pages 368–383. Springer, 2010.

[9] Ninh Pham and Rasmus Pagh. A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 877–885. ACM, 2012.

[10] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012.