Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm

Markus Goldstein and Andreas Dengel

German Research Center for Artificial Intelligence (DFKI),
Trippstadter Str. 122, 67663 Kaiserslautern, Germany
{markus.goldstein,andreas.dengel}@dfki.de

Abstract. Unsupervised anomaly detection is the process of finding outliers in data sets without prior training. In this paper, a histogram-based outlier detection (HBOS) algorithm is presented, which scores records in linear time. It assumes independence of the features making it much faster than multivariate approaches at the cost of less precision. A comparative evaluation on three UCI data sets and 10 standard algorithms show, that it can detect global outliers as reliable as state-of-the-art algorithms, but it performs poor on local outlier problems. HBOS is in our experiments up to 5 times faster than clustering based algorithms and up to 7 times faster than nearest-neighbor based methods.

1 Introduction

Anomaly detection is the process of finding instances in a data set which are different from the majority of the data. It is used in a variety of application domains. In the network security domain it is referred to as intrusion detection, the process of finding outlying instances in network traffic or in system calls of computers indicating compromised systems. In the forensics domain, anomaly detection is also heavily used and known as outlier detection, fraud detection, misuse detection or behavioral analysis. Applications include the detection of payment fraud analyzing credit card transactions, the detection of business crime analyzing financial transactional data or the detection of data leaks from company servers in data leakage prevention (DLP) systems. Furthermore, anomaly detection has been applied in the medical domain as well by monitoring vital functions of patients and it is used for detecting failures in complex systems, for example during space shuttle launches.

However, all of these application domains have in common, that normal behavior needs to be identified and outlying instances should be detected. This leads to two basic assumptions for anomaly detection:

- anomalies only occur very rarely in the data and
- their features do differ from the normal instances significantly.

From a machine learning perspective, three different scenarios exists with respect to the availability of labels [4]: (1) Supervised anomaly detection has a labeled training and test set such that standard machine learning approaches can be applied. (2) Semi-supervised anomaly detection uses a anomaly-free training set consisting of the normal class only. A test set then comprises of normal records and anomalies, which need to be separated. The most difficult scenario is (3) unsupervised anomaly detection, where only a single data set without labels is given and the appropriate algorithm should be able to identify outliers based on their feature values only. In this paper, we introduce an unsupervised anomaly detection algorithm, which estimates densities using histograms.

2 Related Work

Unsupervised Anomaly Detection: Many algorithms for unsupervised anomaly detection have been proposed, which can be grouped into three main categories [4]. In practical applications, *nearest-neighbor based* algorithms seem to be the most used and best performing methods today [1, 2]. In this context, outliers are determined by their distances to their nearest neighbors, whereas global [11] and local methods exist. A very well known local algorithm is the Local Outlier Factor (LOF) [3] on which many other algorithms are based on. Although some algorithms suggest speed up enhancements [5, 10], the basic run time for the nearest-neighbor search is $O(n^2)$. The second category, *clustering based* algorithms can be much faster. Here, a clustering algorithm usually computes centroids and outliers are detected by having a large distance to the dense areas. CBLOF [6] or LDCOF [1] are using k-means as a clustering algorithm leading to a faster computation [2]. The third category comprises of statistical methods, both using parametric and non-parametric models for anomaly detection. Parametric models, for example computing Gaussian Mixture Models (GMM), are usually also very compute intense, depending on the used parameter estimation method. Non-parametric models, such as histograms or kernel-density estimators (KDE) can be used for anomaly detection, especially if a very fast computation is essential.

Histograms in Network Security: In the network security domain it is required that results of outlier detection algorithms are available immediately. Furthermore, the data sets to be processed are very large. This is the reason why histograms are often used as a density estimator for semi-supervised anomaly detection [8]. If multivariate data has to be processed, a histogram for each single feature can be computed, scored individually and combined at the end [7]. In most of the proposed methods, a fixed bin width of the histogram is given or the bin widths are even defined manually.

In this work we are using this basic idea and introduce an *unsupervised* anomaly detection algorithm based on histograms. Furthermore, we propose a dynamic bin-width approach to cover also very unbalanced long-tail distributions.

3 Histogram-based Outlier Score (HBOS)

Besides network security, histogram-based outlier scoring might also be of interest for several other anomaly detection scenarios. Although it is only a combination of univariate methods not being able to model dependencies between features, its fast computation is charming for large data sets. The presented HBOS algorithm allows applying histogram-based anomaly detection in a general way and is also available as open source as part of the anomaly detection extension[1] of RapidMiner [9].

For each single feature (dimension), an univariate histogram is constructed first. If the feature comprises of categorical data, simple counting of the values of each category is performed and the relative frequency (height of the histogram) is computed. For numerical features, two different methods can be used: (1) Static bin-width histograms or (2) dynamic bin-width histograms. The first is the standard histogram building technique using k equal width bins over the value range. The frequency (relative amount) of samples falling into each bin is used as an estimate of the density (height of the bins). The dynamic bin-width is determined as follows: values are sorted first and then a fixed amount of $\frac{N}{k}$ successive values are grouped into a single bin where N is the number of total instances and k the number of bins. Since the area of a bin in a histogram represents the number of observations, it is the same for all bins in our case. Because the width of the bin is defined by the first and the last value and the area is the same for all bins, the height of each individual bin can be computed. This means that bins covering a larger interval of the value range have less height and represent that way a lower density. However, there is one exception: Under certain circumstances, more than k data instances might have exactly the same value, for example if the feature is an integer and a long-tail distribution has to be estimated. In this case, our algorithm must allow to have more than $\frac{N}{k}$ values in the same bin. Of course, the area of these larger bins will grow appropriately.

The reason why both methods are offered in HBOS is due to the fact of having very different distributions of the feature values in real world data. Especially when value ranges have large gaps (intervals without data instances), the fixed bin width approach estimates the density poorly (a few bins may contain most of the data). Since anomaly detection tasks usually involve such gaps in the value ranges due to the fact that outliers are far away from normal data, we recommend using the dynamic width mode, especially if distributions are unknown or long tailed. Besides, also the number of bins k needs to be set. An often used rule of thumb is setting k to the square root of the number of instances N.

Now, for each dimension d, an individual histogram has been computed (regardless if categorical, fixed-width or dynamic-width), where the height of each single bin represents a density estimation. The histograms are then normalized such that the maximum height is 1.0. This ensures an equal weight of each feature to the outlier score. Finally, the HBOS of every instance p is calculated using the corresponding height of the bins where the instance is located:

$$HBOS(p) = \sum_{i=0}^{d} log(\frac{1}{hist_i(p)}) \tag{1}$$

The score is a multiplication of the inverse of the estimated densities assuming independence of the features similar to [7]. This could also be seen as (the inverse

[1] For source code and binaries see `http://madm.dfki.de/rapidminer/anomalydetection`.

of) a discrete Naive Bayes probability model. Instead of multiplication, we take the sum of the logarithms which is basically the same ($log(a \cdot b) = log(a) + log(b)$) and applying a $log(\cdot)$ does not change the order of the scores. The reason why we decided to apply this trick is that it is less sensitive to errors due to floating point precision in extremely unbalanced distributions causing very high scores.

4 Evaluation

For a quantitative evaluation of HBOS on real world data, we evaluated the proposed method on three UCI machine learning data sets commonly used in the anomaly detection community. These data sets, the breast cancer data set and the pen-based (global and local) data set, have been preprocessed as in [1]. The receiver operator characteristic (ROC) is generated by varying the outlier threshold and the area under the curve (AUC) is used for comparison afterwards. Table 4 shows the AUC results for 11 different outlier detection algorithms. It can be seen, that HBOS performs quite well compared to other algorithms on the breast-cancer and pen-global data set. On the local anomaly detection problem, it fails, which is due to the fact that histograms cannot model local outliers with their density estimation.

Algorithm	Breast-cancer	Pen-global	Pen-local
HBOS	**0.9910**	**0.9214**	**0.7651**
k-NN	0.9826	0.9892	0.9852
LOF	0.9916	0.8864	0.9878
Fast-LOF	0.9882	0.9050	0.9937
COF	0.9888	0.9586	0.9688
INFLO	0.9922	0.8213	0.9875
LoOP	0.9882	0.8492	0.9864
LOCI	0.9678	0.8868	–[3]
CBLOF	0.8389	0.6808	0.7007
u-CBLOF	0.9743	0.9923	0.9767
LDCOF	0.9804	0.9897	0.9617

Table 1. Comparing HBOS performance (AUC) with various algorithms using optimal parameter settings.

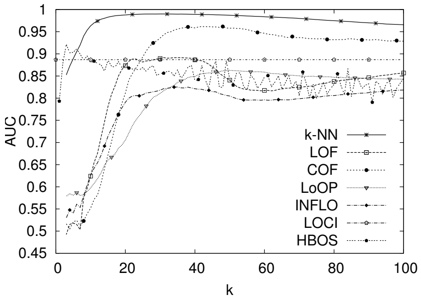

Fig. 1. Comparing AUCs of nearest-neighbor based algorithms with HBOS. k is the number of nearest-neighbors and in HBOS the number of bins.

Besides comparing the outlier detection performance, also the run time of the algorithms was compared. Since the used standard data sets for evaluation are very small (e.g. only 809 instances in the pen-global data set), the experiment was repeated 10,000 times and the mean execution time was taken using an AMD Phenom II X6 1100T CPU with one thread only. The global k-NN method took 28.5ms on average and LOF took 28.0ms to process the pen-global data set. In general, all nearest-neighbor methods perform very similar since the highest effort in this algorithms is the nearest-neighbor search ($O(n^2)$). As a

[3] Not computable due to too high memory requirements for this dataset using LOCI.

clustering based algorithm, LDCOF with k-means was used. The algorithm was started once with 30 random centroids. Using 10 optimization steps, an average run time of 20.0ms was achieved, with 100 optimization steps, which was our default setting for the performance comparison, the algorithm took 30.0ms. We expect clustering based methods to be much faster than nearest-neighbor based algorithms on larger data sets. However, HBOS was significantly faster than both: It took 3.8ms with dynamic bin widths and 4.1ms using a fixed bin width. Thus, in our experiments HBOS was 7 times faster than nearest-neighbor based methods and 5 times faster than the k-means based LDCOF. On larger data sets the speed-up can be much higher: On a not publicly available data set comprising of 1,000,000 instances with 15 dimensions, LOF took 23 hours and 46 minutes whereas HBOS took 38 seconds only (dynamic bin-width: 46 seconds).

5 Conclusion

In this paper we present an unsupervised histogram-based outlier detection algorithm (HBOS), which models univariate feature densities using histograms with a fixed or a dynamic bin width. Afterwards, all histograms are used to compute an anomaly score for each data instance. Compared to other algorithms, HBOS works in linear time $O(n)$ in case of fixed bin width or in $O(n \cdot log(n))$ using dynamic bin widths. The evaluation shows that HBOS performs well on global anomaly detection problems but cannot detect local outliers. A comparison of run times also show that HBOS is much faster than standard algorithms, especially on large data sets.

References

1. Amer, M.: Comparison of unsupervised anomaly detection techniques. Bachelor's Thesis 2011, http://www.madm.eu/_media/theses/thesis-amer.pdf
2. Amer, M., Goldstein, M.: Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer. In: Proc. of the 3rd RCOMM 2012
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. SIGMOD Rec. 29(2), 93–104 (2000)
4. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. 41(3), 1–58 (2009)
5. Goldstein, M.: FastLOF: An expectation-maximization based local outlier detection algorithm. In: Proc. of the Int. Conf. on Pattern Recognition (2012)
6. He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers. Pattern Recognition Letters 24(9-10), 1641 – 1650 (2003)
7. Kim, Y., Lau, W.C., et al: Packetscore: statistics-based overload control against distributed denial-of-service attacks. In: INFOCOM 2004. vol. 4, pp. 2594 – 2604
8. Kind, A., Stoecklin, M., Dimitropoulos, X.: Histogram-based traffic anomaly detection. Network and Service Management, IEEE Transactions on 6(2), 110 –121
9. Mierswa, I., Wurst, M., et al: Yale (now: Rapidminer): Rapid prototyping for complex data mining tasks. In: Proc. of the ACM SIGKDD 2006
10. Papadimitriou, S., Kitagawa, H., et al: Loci: Fast outlier detection using the local correlation integral. Int. Conf. on Data Engineering p. 315 (2003)
11. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. pp. "427–438". SIGMOD '00