An Empirical Analysis of Hubness in Unsupervised Distance-Based Outlier Detection

Arthur Flexer
The Austrian Research Institute for Artificial Intelligence
Freyung 6/6/7, 1010 Vienna, Austria
Email: arthur.flexer@ofai.at

Abstract—**Outlier detection is the task of automatic identification of unknown data not covered by training data (e.g. a previously unknown class in classification). We explore outlier detection in the presence of hubs and anti-hubs, i.e. data objects which appear to be either very close or very far from most other data due to a problem of measuring distances in high dimensions. We compare a classic distance based method to two new approaches, which have been designed to counter the negative effects of hubness, on six high-dimensional data sets. We show that mainly anti-hubs pose a problem for outlier detection and that this can be improved by using a hubness-aware approach based on re-scaling the distance space.**

I. Introduction

Outlier detection[1] is the identification of new or unknown data that a machine learning system is not aware of during training (see [20] for a recent review and [30] for a survey on high-dimensional outlier detection). It is a fundamental requirement for every machine learning system to automatically identify data from regions not covered by the training data since in this case no reasonable decision can be made.

Hubness is a general problem of learning in high-dimensional spaces and has been recognized as a new aspect of the curse of dimensionality in machine learning literature [22], [25]. Hub objects appear very close to many other data objects and anti-hubs very far from most other data objects. It has been argued and demonstrated that anti-hubs might act as 'artificial' outliers since they are far away from many other data points [22]. A recent review on outlier detection in high dimensional data concluded that the "relation of hubness and outlier degree appears to be remaining an open issue" [30]. First studies on outlier detection in the field of music information retrieval exist, where the authors are able to show that it is possible to improve outlier detection by using a hubness reduction method as a preprocessing step [9], [6].

What is still missing is a comprehensive and detailed exploration of the role of hubs and anti-hubs in outlier detection and how this can be changed by applying a hubness reduction method. This is done in this paper by analyzing the performance of hubs and anti-hubs in a classic distance based method and in two hubness-aware approaches, all applied in a high-dimensional classification setting.

II. Related work

Outlier detection, also known as novelty detection, is the task of automatically recognizing data that differ in some respect from the data seen during training by a machine learning system. In case new data differs substantially from training data, no sensible decision can be made by a machine learning system. This should of course be an integral part of any data analysis system and therefore a vast literature concerning the topic exists. For this paper, we follow the systematic of a very recent and comprehensive review [20], which also contains a representative list of references concerning the topic. According to this review, outlier detection can be distinguished into probabilistic, distance-based, reconstruction-based, domain-based and information theoretic approaches. The methods we will present in Section IV are all distance-based, more specifically based on nearest neighbor information. They are also all unsupervised, i.e. class labels are not needed for detection of outliers. Of greater importance is a recent review that deals specifically with outlier detection in high-dimensional data [30]. Although the authors show how some classic outlier detection methods are affected by the concentration of distances (see also next paragraph), hubness is only reviewed as a remaining open issue.

Hubness itself has been discovered in the field of music information retrieval [2], but then gained attention in a machine learning context where it has been discussed as a new aspect of the curse of dimensionality and a general problem of learning in high-dimensional spaces [22], [25]. Hubness is related to the phenomenon of concentration of distances, which is the fact that all points are at almost the same distance to each other for dimensionality approaching infinity [13]. Radovanović et al. [22] presented the argument that for any finite dimensionality, some points are expected to be closer to the center of all data than other points and are at the same time closer, on average, to all other points. Such points closer to the center have a high probability of being hubs, i.e. of appearing in nearest neighbor lists of many other points. Points which are further away from the center have a high probability of being anti-hubs, i.e. points that never appear in any nearest neighbor list. Hubness has been shown to have a negative impact on many tasks including classification [22], nearest

Acknowledgement: This research is supported by the Austrian Science Fund (FWF): P27082, P27703.

[1] Please note that the terms outlier and novelty detection are closely related although not fully synonymous. We will use the term outlier throughout the paper without further distinction for reasons of convenience.

neighbor based recommendation [12] and retrieval [26], clustering [28], [24], visualization [7] and graph construction [11]. It also affects data from diverse domains including multimedia (text, music, images, speech), biology and general machine learning (see [22], [25], [5] for large scale empirical studies). It is also important to note that the degree of concentration and hubness is linked to the intrinsic rather than extrinsic dimension of the data space. Whereas the extrinsic dimension is the actual number of dimensions of a data space the intrinsic dimension is the, often much smaller, number of degrees of freedom of the submanifold in which the data space can be represented [13].

In order to reduce hubness and its negative effects, basically three different approaches have been proposed: re-scaling [25], [27] of the distance space, centering of the data [15], using l^p norms different than Euclidean l^2 norm [8]. One of the re-scaling methods, mutual proximity (MP) [25], has already been applied for outlier detection in a music genre recognition context [9], [6]. MP aims at repairing asymmetric nearest neighbor relations. The asymmetric relations are a direct consequence of the presence of hubs. A hub y is the nearest neighbor of x, but the nearest neighbor of the hub y is most likely another point a ($a \neq x$). This is because hubs are by definition nearest neighbors to very many data points but only one data point can be the nearest neighbor to a hub. The principle of the scaling algorithms is to re-scale distances to enhance symmetry of nearest neighbors. A small distance between two objects should be returned only if their nearest neighbors concur. Application of MP resulted in a decrease of hubness and an accuracy increase in k-nearest neighbor classification on thirty real world datasets including text, image and music data [25]. The somewhat related classic shared nearest neighbors method (SNN) has been proposed to reduce concentration of distances [17] and its impact on hubness has been studied for nearest neighbor classification [27] and compared to MP [10].

Concerning outlier detection, it has been demonstrated [9] that outlier detection based on MP improves the ability to reject outlier data when compared to a classic distance based method. It has also been shown [6] that especially anti-hubs are problematic for distance-based methods. Both studies use music data bases with Kullback-Leibler divergence between signal representations of songs, which is not a full metric. A necessary next step is a more general investigation concerning the improvement achieved and whether this is on account of the changed role of hubs and anti-hubs due to the application of mutual proximity. It seems clear that anti-hubs, being far away from most points, will probably always be rejected as outliers and that hub objects, being close to many points, should be harder to reject. It is therefore our hypothesis that in high-dimensional data, hub and anti-hub points are responsible for many errors being made when rejecting data.

An analysis of the role of anti-hubs in outlier detection has recently been presented [21]. More specifically, the authors have analysed two variants of the ODIN method [16], k-NN outlier scoring [23] and three other methods concerning their relation to anti-hubs. The two variants of the ODIN method use reverse nearest neighbor counts, i.e. counts of how often every data object appears among the k nearest neighbors of every other data object. Per definition anti-hubs have very small or even zero reverse nearest neighbor counts. The authors show that outlier scores based on these counts are correlated to scores from other detection methods but do provide some extra information. Outlier detection results of the ODIN-based methods are somewhat mixed when compared to other methods applied to twelve real world data sets. It also has to be said that these data sets are not really very high dimensional (at most 100) and therefore very likely are not affected by hubness at all.

III. Data

For our analysis, we use six standard machine learning classification data sets following the hypothesis that data objects within a certain class are more similar to each other than objects from different classes. During evaluation in Section V, we will always reserve all data objects belonging to one of the classes as outlier objects to be detected against the rest of the data from all other classes. Data from an unknown class therefore act as outliers, while all other data from remaining classes are treated as inliers.

The data used are GISETTE, SPLICE, DOROTHEA and PROTEIN from the UCI machine learning archive [14], SPLICE from the LibSVM archive [4], and two standard image classification data sets (LEEDS Butterfly [29], 17FLOWERS [19]). The size of data set N, dimensionality d, number of classes G are given in Table I. Data sets are used as they are available on their respective websites without any additional normalization. We always use Euclidean distance to compute distance matrices D.

IV. Methods

We now describe all three methods we will use for outlier detection. All of them compute an outlier score (S^{kNN}, S^{AH} and S^{MP}) which is bounded between 0 and 1 and is compared to a threshold p to decide whether a data object is an outlier or not. A data object is rejected if:

$$S > p \tag{1}$$

A. *kNN-reject*

The first method is a standard distance-based approach known as k-NN outlier scoring [23], or rather *kNN-weight* [1] due to the averaging step. The outlier score is the average distance to the k nearest neighbors:

$$S^{kNN}(x) = \frac{1}{k} \sum_{i=1}^{k} D_{x,NN_i(x)} \tag{2}$$

with $NN_i(x)$ being the ith nearest neighbor of x. A distance matrix D (see Section III) is normalized to the interval 0 to 1 by subtracting the minimum distance and dividing through the maximum distance. This also bounds the outlier score S^{kNN} between 0 and 1.

TABLE I
HUBNESS ANALYSIS OF ALL DATA SETS

data set	N	d	G	$\#hub$	$\#anti$	$\#normal$	H^n
GISETTE	6000	5000	2	49	635	5316	4.48
SPLICE	1000	60	2	28	289	683	4.55
DOROTHEA	800	100000	2	19	729	52	12.96
PROTEIN	6621	357	3	228	4508	1885	36.12
17FLOWERS	1360	36000	17	43	369	948	3.91
LEEDS	832	36000	10	29	259	544	3.54

B. AH-reject

The next method is based on previous work of using reverse nearest neighbor counts for outlier detection [21]. In hubness research, the reverse nearest neighbor count of a point x is usually called n-occurrence $O^n(x)$ [22]. It is the number of times x occurs in the first n nearest neighbors of all other objects in the collection. The proposed method simply uses $O^n(x)$ as the outlier score. It has been termed "Antihub" because anti-hubs have very small or even zero n-occurrence and are therefore very likely to be rejected as outliers. The same authors proposed a variant called "Antihub2", which also includes information from the k nearest neighbors of x:

$$S^{AH}(x) = (1-\alpha)\frac{1}{O^n(x)+1} + \alpha\sum_{i=1}^{k}\frac{1}{O^n(NN_i(x))+1} \quad (3)$$

We set $\alpha = k/(k+1)$, which basically gives the average across the n-occurrences of x itself and its k nearest neighbors $NN_i(x)$. The original formulation of "Antihub2" [21] contains an optimization step searching for optimal values for weight parameters α. Since this yielded rather mixed results compared to simply using $O^n(x)$ as outlier score, we used averaging instead. Therefore our method AH is a simplified version of "Antihub2" [21]. The outlier score S^{AH} is also bounded between 0 and 1, with $S^{AH} = 1$ in case all involved n-occurrences O^n are equal zero, and $S^{AH} = 1/(N+1)$ in case all involved $O^n = N$. An n-occurrence is equal N in case a data point appears in all neighborhood lists of all data points.

C. MP-reject

Mutual Proximity (MP) [25] rescales the original distance space so that two objects sharing similar nearest neighbors are more closely tied to each other, while two objects with dissimilar neighborhoods are repelled from each other. MP has been devised to counter the negative effects of hubness in high dimensional data spaces. For MP-reject we exploit the fact that MP rescales distances to probabilities which enables comparability and simple thresholding. MP reinterprets the distance of two objects as a mutual proximity in terms of their distribution of distances. To compute MP, we assume that the distances $D_{x,i=1..N}$ from an object x to all other objects in our data set follow a certain probability distribution, thus any distance $D_{x,y}$ can be reinterpreted as the probability of y being the nearest neighbor of x, given their distance $D_{x,y}$ and the probability distribution $P(X)$:

$$P(X > D_{x,y}) = 1 - P(X \leq D_{x,y}) = 1 - \mathcal{F}_x(D_{x,y}) \quad (4)$$

with $\mathcal{F}$ denoting the cumulative distribution function (cdf). MP is then defined as the probability that y is the nearest neighbor of x given $P(X)$ and x is the nearest neighbor of y given $P(Y)$:

$$MP(D_{x,y}) = P(X > D_{x,y} \cap Y > D_{y,x}). \quad (5)$$

To compute MP in our experiments we assume that the distances $D_{x,i=1..N}$ follow a Gaussian distribution. Previous work [25] has shown that the Gaussian assumption yields results comparable to using the full empirical distribution. We define the outlier score as the average of the MP-distances to the k nearest neighbors of x:

$$S^{MP}(x) = \frac{1}{k}\sum_{i=1}^{k}(1 - MP(x, NN_i(x))) \quad (6)$$

with $NN_i(x)$ being the ith nearest neighbor of x. Please note that we use the term $(1 - MP)$ because mutual proximity computes similarities and we need distances for the rejection rule. Outlier score S^{MP} is bounded between 0 and 1 since it is based on MP which computes a probability.

V. RESULTS

Before actually evaluating the outlier detection methods, we present an analysis of the hubness of all data sets in Table I. The table gives the number of hubs ($\#hub$), anti-hubs ($\#anti$) and normal ($\#normal$) data objects. Anti-hubs are defined as data objects with an n-occurrence O^n (see Sec. IV-B) equal 0, hubs with $O^n > 5n$, all based on numbers of nearest neighbors of $n = 5$. Normal data objects are all non-hub and non-anti-hub objects, i.e. $0 < O^n \leq 5n$. Please note that the mean n-occurrence across all objects in a data base is equal to n. Any n-occurrence significantly bigger than n therefore indicates existence of a hub. As has been done before [25], we chose objects appearing more than five times the expected value ($5n = 25$) as hub objects.

The last column of Table I gives the hubness H^n. It is the skewness of the distribution of n-occurrences, i.e. the third moment of the distribution. A data set having high hubness produces few hub objects with very high n-occurrence and many anti-hubs with n-occurrence of zero. This makes the distribution of n-occurrences skewed with positive skewness indicating high hubness. The hubness values H^n range from

3.54 for LEEDS to 36.12 for PROTEIN, indicating that there is a clear hubness effect in these data sets. Previous work [25] has shown that values above 1.4 are already problematic. As can be seen, consistent with theory, in all data sets there are small numbers of hubs and large numbers of anti-hubs. The data set with smallest hubness is LEEDS, which contains 29 hubs and 259 anti-hubs in 832 data objects. The data set with largest hubness is PROTEIN, which contains 228 hubs and 4508 anti-hubs in 6621 data objects.

To evaluate the three outlier detection methods described in Sec. IV we use the following approach shown as pseudo-code in Table II. First we set aside all data objects belonging to a class `g` as new data objects (`[new,data]=separate(alldata,g)`) which yields data sets `new` and `data` (all data objects not belonging to class g). Then we do a $C = 10$-fold crossvalidation using `data` and `new`: we randomly split `data` into `train` and `test` fold (`[train,test] = split(data,c)`) with `train` always consisting of 90% and `test` of 10% of `data`. We compute the percentage of `new` data objects which are rejected as being outliers (`outlier_reject(g,c) = outlier(new)`) and do the same for the `test` data objects (`test_reject(g,c) = outlier(test)`). The evaluation procedure gives $G \times C$ (e.g. 17×10 for 17FLOWERS) matrices of `outlier_reject` and `test_reject` for each parameterization of the outlier detection approaches, i.e. for different values of neighborhood size k (see Sec. IV). In what follows we always report average numbers across these $G \times C$ sized matrices of results, i.e. averages across crossvalidation folds and classes.

TABLE II
OUTLINE OF EVALUATION PROCEDURE

```
for g = 1 : G
  [new,data] = separate(alldata,g)
  for c = 1 : C
    [train,test] = split(data,c)
    outlier_reject(g,c) = outlier(new)
    test_reject(g,c) = outlier(test)
  end
end
```

The results for outlier detection are given in Figs. 1 and 2 as Receiver Operating Characteristic (ROC) curves for two of our data sets (see also last paragraph of this section). To obtain an ROC curve, the fraction of false positives (object is not an outlier but it is rejected, in our case `test_reject`) is plotted versus the fraction of true positives (object is an outlier and correctly rejected, in our case `outlier_reject`) for varying threshold values p. We vary the threshold values from $p = 0$ to $p = 1$ in steps of .02. An ROC curve shows the trade off between how sensitive and how specific a method is. Any increase in sensitivity will be accompanied by a decrease in specificity. If a method becomes more sensitive towards outlier objects it will reject more of them but at the same it will also become less specific and also falsely reject more non-outlier objects. Consequently, the closer a curve follows the left-hand border and then the top border of the ROC space, the better the performance of the method is.

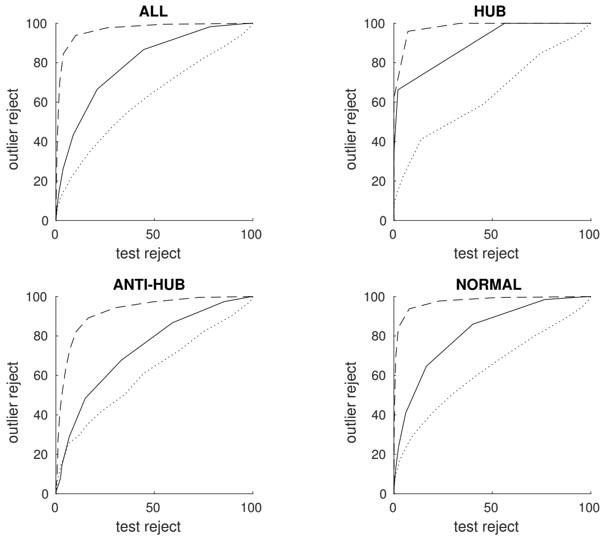

Fig. 1. ROC plots for data set GISETTE, solid line for kNN, dotted for AH, dashed for MP.

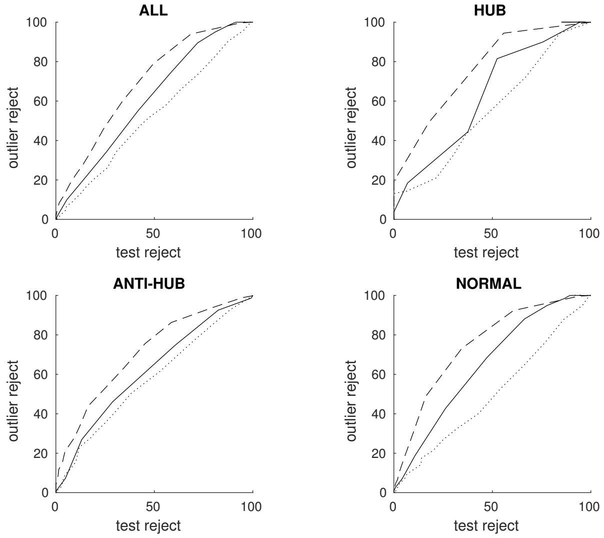

Fig. 2. ROC plots for data set SPLICE, solid line for kNN, dotted for AH, dashed for MP.

To summarize the information contained in ROC curves, we also compute the Area Under the Curve (AUC), which gives the percentage of the whole ROC space that lies underneath an ROC curve. An AUC of 1 indicates perfect performance, while an AUC of .5 indicates performance at chance level.

As a first analysis step, we tried to find optimal parameters k (neighborhood size for algorithms kNN, AH and MP) by comparing AUC results based on values of $k = 1, 2, 3, 5, 10, 20, 30, 40, 50$. Results for all data sets can be found in Figures 3 to 8. Our first observation is that AUC values for method MP are consistently outperforming kNN

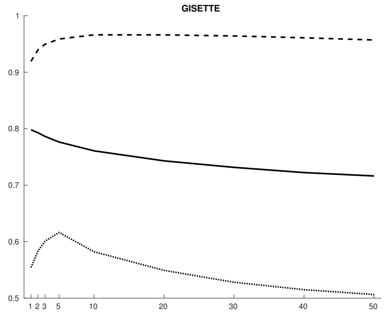

Fig. 3. AUC (y-axis) analysis for data set GISETTE and parameter k ranging from 1,2,3,5,10,20,30,40 to 50 (x-axis), solid line for kNN, dotted for AH, dashed for MP.

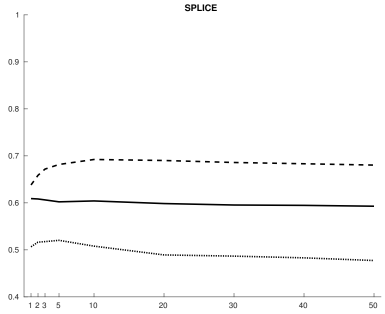

Fig. 4. AUC (y-axis) analysis for data set SPLICE and parameter k ranging from 1,2,3,5,10,20,30,40 to 50 (x-axis), solid line for kNN, dotted for AH, dashed for MP.

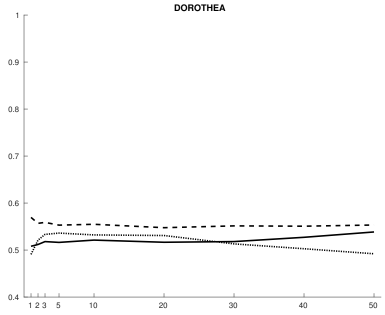

Fig. 5. AUC (y-axis) analysis for data set DOROTHEA and parameter k ranging from 1,2,3,5,10,20,30,40 to 50 (x-axis), solid line for kNN, dotted for AH, dashed for MP.

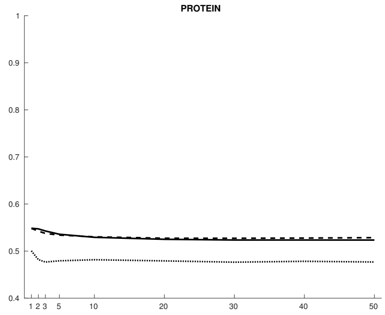

Fig. 6. AUC (y-axis) analysis for data set PROTEIN and parameter k ranging from 1,2,3,5,10,20,30,40 to 50 (x-axis), solid line for kNN, dotted for AH, dashed for MP.

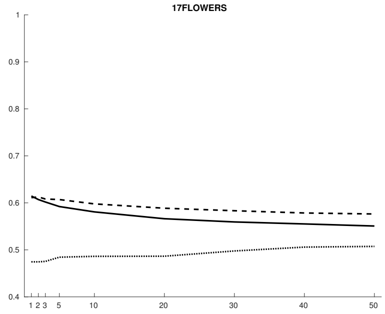

Fig. 7. AUC (y-axis) analysis for data set 17FLOWERS and parameter k ranging from 1,2,3,5,10,20,30,40 to 50 (x-axis), solid line for kNN, dotted for AH, dashed for MP.

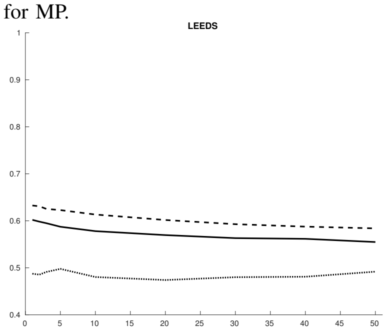

Fig. 8. AUC (y-axis) analysis for data set LEEDS and parameter k ranging from 1,2,3,5,10,20,30,40 to 50 (x-axis), solid line for kNN, dotted for AH, dashed for MP.

and AH for four out of six data sets (GISETTE, SPLICE, DOROTHEA, LEEDS) across the full range of k values. For data set PROTEIN the performance of MP and kNN is almost identical for all values of k, for data set 17FLOWERS the highest AUC value of .61 is reached for $k = 1$ for both MP and kNN. Our second observation is that method AH consistently performs worse than both MP and kNN across the whole range of k values, with the only exception being data set DOROTHEA where it performs at a comparable level to kNN until $k = 20$, but still falls behind the peak performances of its competitors for higher k. In general, peak AUC performances are reached for $k = 1, 5$ or 10, except kNN for DOROTHEA and AH for 17FLOWERS which peak at $k = 50$. Values of k higher than 50 have not been evaluated but might be able to change results in these two instances. Even more so, since it has already been noted [21] that reverse nearest neighbor information in high dimensional outlier detection is most useful with high values of k. We conclude that MP is able to improve outlier detection in 4 out of 6 cases compared to the baseline kNN approach, and that AH fails to show any improvements at all. The largest differences between methods can be observed for data set GISETTE with AUC values of .97 for MP ($k = 10$), .80 for kNN ($k = 1$) and only .62 ($k = 5$) for AH.

We now make a detailed AUC analysis to explore the influence of hubs, anti-hubs and normal data as defined at the beginning of this section. Please note that all following results are based on optimal values of k indicated by peak AUC values in Figures 3 to 8: GISETTE – kNN 1, MP 10, AH 5; SPLICE – kNN 1, MP 10, AH 5; DOROTHEA – kNN 50, MP 1, AH 5; PROTEIN – kNN 1, MP 1, AH 1; 17FLOWERS – kNN 1, MP 2, AH 50; LEEDS – kNN 1, MP 1, AH 5. In

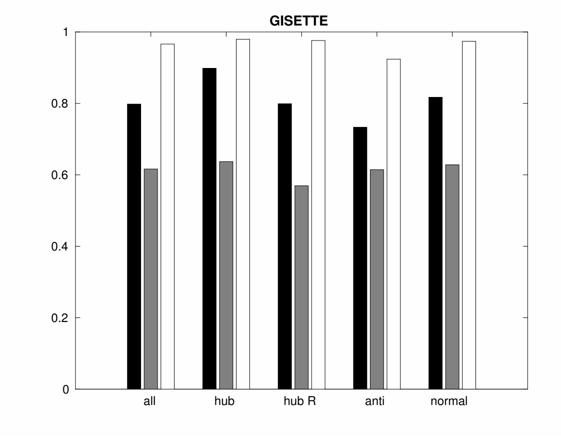

Fig. 9. AUC plot for data set GISETTE, black bars for kNN, grey for AH, white for MP.

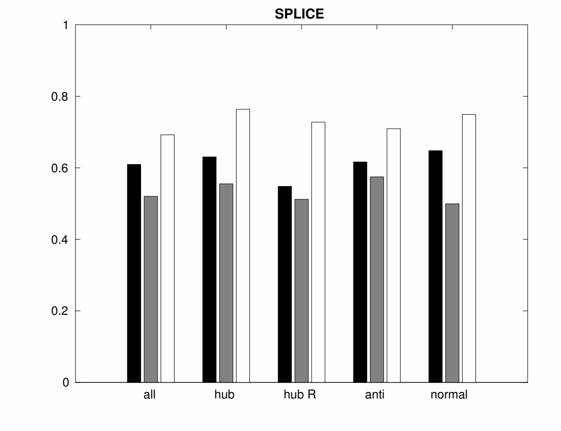

Fig. 10. AUC plot for data set SPLICE, black bars for kNN, grey for AH, white for MP.

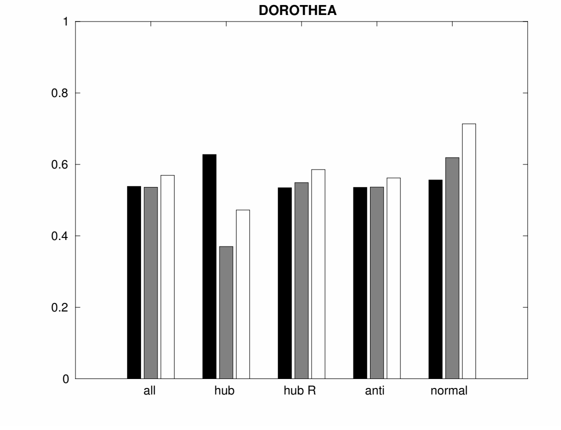

Fig. 11. AUC plot for data set DOROTHEA, black bars for kNN, grey for AH, white for MP.

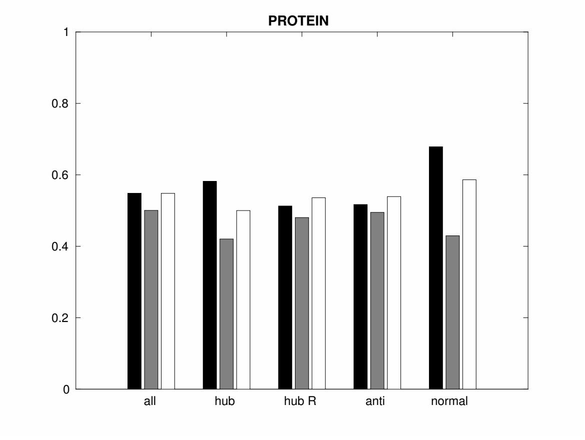

Fig. 12. AUC plot for data set PROTEIN, black bars for kNN, grey for AH, white for MP.

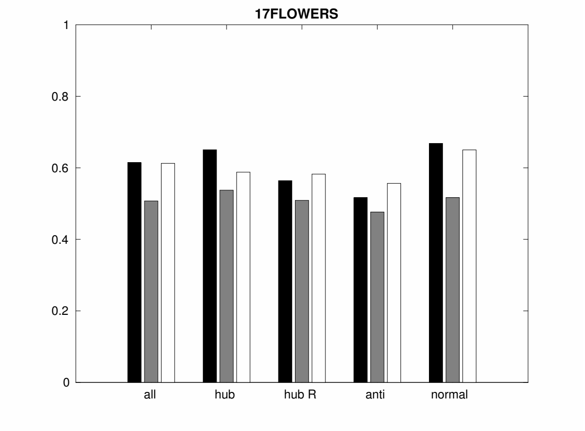

Fig. 13. AUC plot for data set 17FLOWERS, black bars for kNN, grey for AH, white for MP.

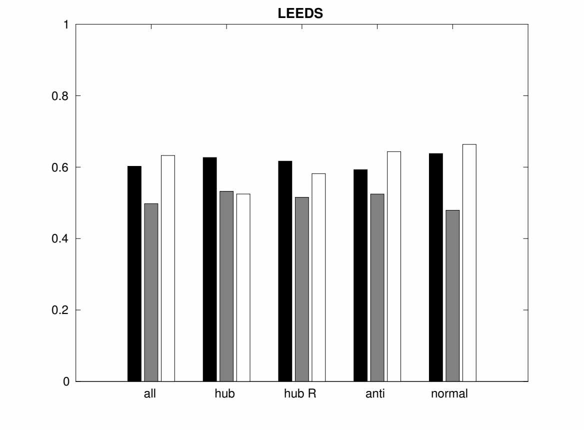

Fig. 14. AUC plot for data set LEEDS, black bars for kNN, grey for AH, white for MP.

Figures 9 to 14 we show AUC results separate for all data, hubs, anti-hubs and normal data ('all', 'hub', 'anti', 'normal') when presented as 'new' or 'test' data (see Table II) to the three outlier detection methods, indicated via black (kNN), grey (AH) and white bars (MP). We also add another group of bars to the figures, showing AUC results for the case where at least one of the k nearest neighbors used to compute the kNN, MP and AH outlier scores is a hub object. Since hubs are per definition members of many nearest neighbor lists, and since they have a small distance to most data objects, it is expected that they have a negative impact on the outlier scores. These additional bars are termed 'hub R' (R for reverse) in the figures.

Looking at the groups of bars termed 'all', we can see again that MP (white bars) improves AUC results relative to kNN (black bars) for all data sets except PROTEIN (Fig. 12) and 17FLOWERS (Fig. 13). We also see that AH degrades AUC performance for all data sets except DOROTHEA (Fig. 11). Comparing kNN, MP and AH for 'hub', 'hub R', 'anti' and 'normal' data objects, we observe that if there is an improvement for 'all' data (GISETTE, DOROTHEA, SPLICE, LEEDS), MP improves over kNN for almost all types of data and AH worsens relative to kNN for almost all types of data. The only exceptions are when 'hub' data degrade performance for MP on DOROTHEA and LEEDS, and 'hub R' data do so for LEEDS. Also AH improves performance only for 'hub R' and 'normal' DOROTHEA data. We can conclude that MP improves AUC results relative to kNN for 4 out of 6 data sets, and that this improvement is visible for all types of data, no matter whether they are hubs, anti-hubs or normal data objects. It is also evident that AH degrades AUC results for almost all data sets and all types of data.

Comparing AUC results for 'normal' and 'anti' data objects for method kNN, we see that anti-hubs yield worse AUC results compared to normal data for all data sets. Sometimes this difference is quite small (DOROTHEA: .54 'anti', .56 'normal'), but sometimes also quite large (PROTEIN: .52 'anti', .68 'normal'). Comparing AUC results for 'normal' and 'hub' data objects for method kNN, we see quite mixed results. AUC results for 'hub' data are higher for data sets GISETTE and DOROTHEA, lower for PROTEIN and almost at the same level as 'normal' AUC results for the three remaining data sets. Comparing AUC results for 'normal' and 'hub R' data, results for 'hub R' data are worse for data sets SPLICE, PROTEIN and 17FLOWERS, and reach about the same AUC level as 'normal' data for the other three data sets. Our main conclusion from this analysis is that especially anti-hubs present a problem for outlier detection and that the situation concerning hub objects is still not fully clear and warrants further investigation.

We now present the ROC plots that the above AUC results are based on, but due to space limitations only for data sets GISETTE and SPLICE, where improvements due to MP are most pronounced. The ROC plots are given for all data as well as for hubs, anti-hubs and normal data separately. Looking at the results for GISETTE in Figure 1, we can see that the ROC curve for method MP (dashed line) is above those for kNN (solid line) and AH (dotted line) for almost the whole ROC space for all data together as well as for hubs, anti-hubs and normal objects (sub-plots titled 'ALL', 'ANTI-HUB' and 'NORMAL'). It is also interesting to see, that the ROC curve for method AH and anti-hub objects (sub-plot titled 'ANTI-HUB', dotted line) is much closer to the main diagonal indicating performance at chance level. This explains the low AUC of .61 for this curve, with .5 indicating chance level. This is due to the fact, that method AH is based on n-occurrence counts (see Section IV-B), basically detecting everything with a low n-occurrence as outliers. It therefore rejects all anti-hubs as outliers, no matter whether they are true outliers or test data that should not be rejected. Figure 2 gives the ROC plots for SPLICE, repeating the same patterns of behavior we just described, albeit less clearly and at a lower performance level. ROC curves for method MP more or less dominate those for kNN and AH for all data together as well as hubs, anti-hubs and normal data.

VI. Discussion

In discussing our results obtained in Section V, we like to recapitulate our main findings.

Our first result is that classic distance-based outlier detection methods are negatively affected by hubness. As can be seen by looking at the results for distance-based method kNN separately for hubs, anti-hubs and normal data, especially anti-hubs present problems for outlier detection as evident from their lower AUC values. Since anti-hubs per definition are far away from most other data points in a data base, it seems logical that they are being detected as outliers even when they not really are. As for hub objects, we would have expected that they are also responsible for more detection errors than normal data, which is not really the case. This is true for hub objects as candidates for outlier detection but also for hub objects as part of nearest neighbor lists (termed 'hub R' in Section V) influencing computation of outlier scores for non-hub objects. When analyzing these 'hub R' data objects, a single hub object in a nearest neighbor list was sufficient to define a 'hub R' data object. But since the nearest neighbor lists were of varying size (parameter k ranged from 1 to 50, see Section V), a deeper analysis of the actual count of hub objects in nearest neighbor lists might yield further insights.

Our second result is that reverse nearest neighbor information, as being used in our method AH, cannot improve distance-based outlier detection that is affected by hubness. On the contrary, our results for method AH show that AUC values compared to kNN even deteriorate. Especially AUC results for anti-hubs are sometimes close to chance level. Given the fact that AH basically detects everything with a low reverse nearest neighbor count as an outlier, this is not surprising. After all this means that anti-hubs, which already are a problem for distance-based methods, are detected as outliers no matter whether they really are or not. It should be noted that AH is a simplified version of "Antihub2" [21], lacking an additional optimization step. Furthermore, as with "Antihub2" [21], including reverse nearest neighbor information of many more than $k = 50$ nearest neighbors might improve results for AH also.

Our third result is that our hubness-aware algorithm MP is able to improve outlier detection and that this improvement is also due to the changed role of anti-hubs. Looking at our results for using MP, we can see that we gain overall improvements in AUC which are visible for hub, anti-hub and normal data. Mutual proximity has been shown to decisively reduce the negative impact of hubs and anti-hubs and produce distance spaces with much more normal behavior [25], which might explain its successful application to outlier detection.

Given these three main results, it would now of course be very interesting to analyse the behavior of other outlier detection methods when confronted with data affected by hubness. Of special interest are methods that have already been designed for high-dimensional outlier detection like e.g. "angle-based outlier detection" (ABOD) [18].

Concerning our overall evaluation setting, it has to be said that we always used all members of one class as outliers versus all remaining data of all other classes as inliers. Given that three of our six data sets consist of only two classes, there are sometimes more outliers than inliers. Future work should consider down sampling, i.e. keeping only a small part of one class as outliers to prevent this (see e.g [3]). The reason for resorting to this procedure in the first place was the lack of semantically meaningful outlier data sets, i.e. data sets with both rare and deviating instances, that are truly high dimensional. A very recent comprehensive paper on evaluation of outlier detection and available data sets might help in this respect [3].

VII. Conclusions

We have presented a detailed empirical study of the role of hubness in outlier detection, which builds on previous work on the same topic [9], [21], [6]. We found that classic distance-based methods for outlier rejection are negatively impacted by hubness, where especially anti-hubs pose a problem. We also showed that a simple outlier detection method based on reverse nearest neighbor counts is not able to help in this respect. But a hubness-aware method based on hubness reduction via computation of mutual proximity is able to improve outlier detection results. Improvements concerning the problematic role of anti-hubs are part of the success, but performance gains are also visible for hubs and normal data. Our empirical evaluation analysed six high dimensional data sets from diverse domains. Future work should try to corroborate our results on even more data as well as analyse the impact of hubness on other outlier detection methods.

References

[1] Angiulli F., Pizzuti C.: Outlier Mining in Large High-Dimensional Data Sets, IEEE Transactions on Knowledge and Data Engineering, 17(2):203-215, 2005.

[2] Aucouturier, J.-J., Pachet F.: Improving Timbre Similarity: How high is the sky?, Journal of Negative Results in Speech and Audio Sciences, 1(1), 2004.

[3] Campos G.O., Zimek A., Sander J., Campello R. J. G. B., Micenková B., Schubert E., Assent I., Houle M.E.: On the Evaluation of Unsupervised Outlier Detection: Measures, Datasets, and an Empirical Study, Data Mining and Knowledge Discovery 30(4): 891-927, 2016.

[4] Chang C.-C., Lin C.-J.: LIBSVM: a library for support vector machines, 2001, available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[5] Feldbauer R., Flexer A.: Centering versus Scaling for Hubness Reduction, Proceedings of the 25th International Conference on Artificial Neural Networks, Part I, pp. 175-183, Springer International Publishing, 2016.

[6] Flexer A.: Hubness-Aware Outlier Detection for Music Genre Recognition, Proceedings of the 19th International Conference on Digital Audio Effects, 2016.

[7] Flexer A.: Improving visualization of high-dimensional music similarity spaces, Proceedings of the 16th International Society for Music Information Retrieval Conference, Malaga, Spain, 2015.

[8] Flexer A., Schnitzer D.: Choosing l^p norms in high-dimensional spaces based on hub analysis, Neurocomputing, Volume 169, pp. 281-287, 2015.

[9] Flexer A., Schnitzer D.: Using mutual proximity for novelty detection in audio music similarity, Proceedings of the 6th International Workshop on Machine Learning and Music, Prague, Czech Republic, 2013.

[10] Flexer A., Schnitzer D.: Can Shared Nearest Neighbors Reduce Hubness in High-Dimensional Spaces?, Proceedings of 1st International Workshop on High Dimensional Data Mining, in conjunction with the IEEE International Conference on Data Mining, 2013.

[11] Flexer A., Stevens J.: Mutual proximity graphs for music recommendation, Proceedings of the 9th International Workshop on Machine Learning and Music, 2016.

[12] Flexer A., Schnitzer D., Schlüter J.: A MIREX meta-analysis of hubness in audio music similarity, Proceedings of the 13th International Society for Music Information Retrieval Conference, 2012.

[13] Francois D., Wertz V., Verleysen M.: The concentration of fractional distances, IEEE Transactions on Knowledge and Data Engineering, 19:873-886, 2007.

[14] Frank A., Asuncion A.: UCI machine learning repository, 2010, available at: http://archive.ics.uci.edu/ml

[15] Hara K., Suzuki I., Shimbo M., Kobayashi K., Fukumizu K., Radovanović M.: Localized centering: Reducing hubness in large-sample data hubness in high-dimensional data, Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 2645-2651, 2015.

[16] Hautamäki V., Kärkkäinen I., Fränti P.: Outlier Detection Using k-Nearest Neighbour Graph, Proceedings of the 17th International Conference on Pattern Recognition, pp. 430-433, 2004.

[17] Houle M.E., Kriegel H.-P., Kröger P., Schubert E., Zimek A.: Can Shared-Neighbor Distances Defeat the Curse of Dimensionality?, in Scientific and Statistical Database Management, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6187, ch. 34, pp. 482-500.

[18] Kriegel H. P., Schubert M., Zimek A.: Angle-based outlier detection in high-dimensional data, Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 444-452, 2008.

[19] Nilsback M.-E., Zisserman A.: Automated flower classification over a large number of classes, Proceedings of the 6th Indian Conference on Computer Vision, Graphics & Image Processing, IEEE, pp. 722-729, 2008.

[20] Pimentel M.A.F., Clifton D.A., Clifton L., Tarassenko L.: A review of novelty detection, Signal Processing, Vol. 99, pp. 215-249, 2014.

[21] Radovanović M., Nanopoulos A., Ivanović M.: Reverse nearest neighbors in unsupervised distance-based outlier detection, IEEE Transactions on Knowledge and Data Engineering, 27(5), 1369-1382, 2015.

[22] Radovanović M., Nanopoulos A., Ivanović M.: Hubs in space: Popular nearest neighbors in high-dimensional data, Journal of Machine Learning Research, 11:2487-2531, 2010.

[23] Ramaswamy S., Rastogi R., Shim K.: Efficient algorithms for mining outliers from large data sets, Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pp. 427-438, 2000.

[24] Schnitzer D., Flexer A.: The Unbalancing Effect of Hubs on K-medoids Clustering in High-Dimensional Spaces, Proceedings of the International Joint Conference on Neural Networks, 2015.

[25] Schnitzer D., Flexer A., Schedl M., Widmer G.: Local and Global Scaling Reduce Hubs in Space, Journal of Machine Learning Research, 13(Oct):2871-2902, 2012.

[26] Schnitzer D., Flexer A., Tomašev N.: A Case for Hubness Removal in High-Dimensional Multimedia Retrieval, Proceedings of the 36th European Conference on Information Retrieval, 2014.

[27] Tomašev N., Mladenić D.: Hubness-aware shared neighbor distances for high-dimensional k-nearest neighbor classification, in Hybrid Artificial Intelligent Systems, pp. 116-127, Springer, 2012.

[28] Tomašev N., Radovanović M., Mladenić D., Ivanović M.: The Role of Hubness in Clustering High-dimensional Data, IEEE Transactions on Knowledge and Data Engineering, Volume 26, Issue 3, 2013.

[29] Wang J., Markert K., Everingham M.: Learning models for object recognition from natural language descriptions, Proceedings of the British Machine Vision Conference, 2009.

[30] Zimek A., Schubert E., Kriegel H.-P.: A survey on unsupervised outlier detection in high-dimensional numerical data, Statistical Analysis and Data Mining, 5: 363-387, 2012.