Intrinsic t-Stochastic Neighbor Embedding for Visualization and Outlier Detection

A Remedy Against the Curse of Dimensionality?

Author manuscript – the final publication is available at Springer via
https://doi.org/10.1007/978-3-319-68474-1_13

Erich Schubert and Michael Gertz

Heidelberg University
{schubert,gertz}@informatik.uni-heidelberg.de

Abstract. Analyzing high-dimensional data poses many challenges due to the "curse of dimensionality". Not all high-dimensional data exhibit these characteristics because many data sets have correlations, which led to the notion of intrinsic dimensionality. Intrinsic dimensionality describes the local behavior of data on a low-dimensional manifold within the higher dimensional space.

We discuss this effect, and describe a surprisingly simple approach modification that allows us to reduce local intrinsic dimensionality of individual points. While this unlikely will be able to "cure" all problems associated with high dimensionality, we show the theoretical impact on idealized distributions and how to practically incorporate it into new, more robust, algorithms. To demonstrate the effect of this adjustment, we introduce the novel Intrinsic Stochastic Outlier Score (ISOS), and we propose modifications of the popular t-Stochastic Neighbor Embedding (t-SNE) visualization technique for intrinsic dimensionality, intrinsic t-Stochastic Neighbor Embedding (it-SNE).

1 Introduction

Analyzing high-dimensional data is a major challenge. Many of our intuitions from low-dimensional space such as distance and density no longer apply in high-dimensional data the same way they do in 2- or 3-dimensional space. For example, the center of a high-dimensional ball contains only very little mass, whereas the majority of the mass of a high-dimensional ball is in its shell. Grid-based approaches do not work well to partition high-dimensional data, because the number of grid cells grows exponentially with the dimensionality, so almost all cells will be empty. We are particularly interested in anomaly detection approaches for high-dimensional data, where many distance-based algorithms are known to suffer from the "curse of dimensionality" [43].

To understand the performance of algorithms, it is advisable to visualize the results, but visualization of high-dimensional data has similar problems because of the sheer number and correlations of attributes to visualize [1]. A promising recent visualization method is t-SNE [35], which embeds data in a way that preserves neighborhoods, but not distances and densities, as seen in Figure 1, where the density information of the Gaussian distribution is largely lost, but neighborhoods are to a large extend preserved.

In this article, we improve the concept of "stochastic neighbors" which forms the base for SNE [16], t-SNE [35], and the outlier detection method SOS [24]. We study the distance concentration effect and construct a way to avoid the loss of discrimination (although not a universal "cure" for the curse of dimensionality), which we integrate into stochastic neighbors, to construct the improved ISOS outlier detection and it-SNE projection technique for visualizing anomalies in high intrinsic dimensionality.

2 Related Work

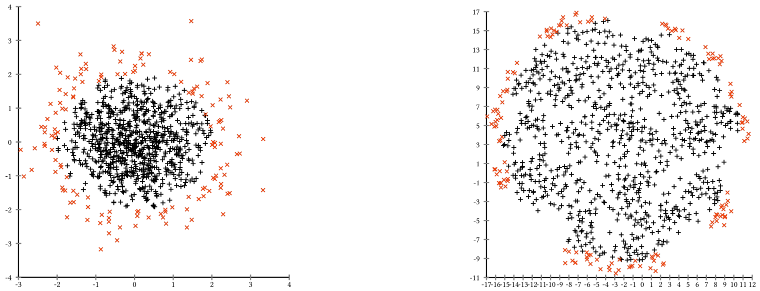
2.1 The Curse of Dimensionality

The "curse of dimensionality" was initially coined in combinatorial optimization [4], but now refers to a whole set of phenomena associated with high dimensionality [20,17]. We focus here on the loss of "discrimination" of distances as described by [6]. Intuitively, this curse means that the distances to the closest neighbor and the farthest neighbor become relatively similar, up to the point where they become "indiscernible". This can be formalized as:

$$\lim_{\dim\to\infty} E\left[\frac{\max_{y\neq x} d(x,y)-\min_{y\neq x} d(x,y)}{\min_{y\neq x} d(x,y)}\right] \to 0. \tag{1}$$

This can be proven for idealized distributions, but the effect can be observed in real data, and affects the ability of many distance-based methods, e.g., in outlier detection [43].

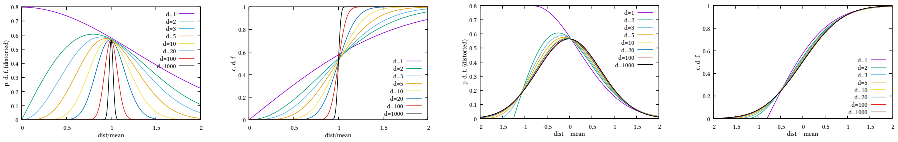
Figure 2a visualizes the distribution of distances from the origin of a multivariate standard normal distribution, i.e. $X = (\sum_d Y_i^2)^{1/2}$ with $Y_i \sim \mathcal{N}(0;1)$. The resulting distance distribution is a Chi distribution with d degrees of freedom. To visualize the concentration of relative distances, we normalize the x-axis by the mean distance. We can see the p.d.f. concentrate around the mean, and the c.d.f. change abruptly at the



(a) Gaussian distribution

(b) t-SNE projection

Fig. 1: t-SNE projections do not preserve distances or density, but try to preserve neighbors (red x markers indicate points more than 2 standard deviations from the center)



(a) Relative deviation

(b) Absolute deviation

Fig. 2: Deviation from the expected value of a multivariate standard normal distribution.

mean, as expected by Equation 1. However, if we look at absolute deviations from the mean in Figure 2b (by centering the distributions on the mean rather than scaling them), we can no longer see any distance concentration. In terms of deviation from the mean, the distributions appear very similar (for $d > 2$)—so there should be some leeway here against the curse of dimensionality. (But unfortunately, this transformation yields negative values, so it cannot be used as a distance normalization in most applications.)

[43] have shown that the distance concentration effect itself is not the main problem, and outliers can still be easy to detect if this effect occurs. [5] have shown that we can discern well-separated clusters in high dimensionality, because we can still distinguish near from far neighbors. [20] show that by ignoring the absolute distance values, but instead counting the overlap of neighborhoods ("shared nearest neighbors"), we can still cluster high-dimensional data, reflecting the observation that the ranking of near points remains meaningful, even when the relative distances do not provide contrast.

There are many other aspects of the curse of dimensionality [43,17], such as hubness [37], which we will not focus on here (and hubness has also been observed in lower dimensional data [33]). Some issues with high dimensionality are very practical in nature: preprocessing, scaling, and weighting of features is often very important for data analysis, but becomes difficult to do with a large number of features of very different nature, such as when combining continuous, discrete, ordinal and categoricial features. Such problems are also beyond the scope of this article.

2.2 Intrinsic Dimensionality

Data on a line in a 10-dimensional space will essentially behave as if it were in a 1-dimensional space. This led to the notion of intrinsic dimensionality, and this intuition has been formally captured for example by the expansion dimension [26].

Text data is often represented in a very high-dimensional data space, where every different word in the corpus corresponds to a dimension. Based on a naive interpretation of the curse of dimensionality, one would assume such a representation to be problematic; yet text search works very well. In the vector space model, text data usually is sparse, i.e., most attributes are zero. Adding additional attributes that are constant, or copies of existing attributes, usually do not increase the difficulty of a data set much.

Therefore, it is good to distinguish between the representation dimensionality—the number of attributes used for encoding the data—and the effective dimensionality for data analysis. [17] establishes the theoretical connection between dimensionality, discriminability, density, and distance distributions; as well as the connection to extreme value theory [18]. Intrinsic dimensionality is often estimated using tail estimators, in particular using the Hill [15] estimator, or a weighted average thereof [21]. More recent approaches involve the expansion dimension [26] and the Generalized Expansion Dimension (GED) [19]. [2] survey and compare several estimation techniques for intrinsic dimensionality. Implementations of several estimators for intrinsic dimensionality can be found in the ELKI data mining toolkit [39]. The Hill maximum-likelihood estimator uses the sorted distances of x to its k-nearest neighbors $y_1 \dots y_k$ for estimation [2]:

$$\widehat{\mathrm{ID}}_{\text{Hill}}(x) := -\left(\tfrac{1}{k-1} \sum\nolimits_{i=1}^{k-1} \log \tfrac{d(x,y_i)}{d(x,y_k)}\right)^{-1} \tag{2}$$

2.3 Outlier Detection

Distance-based outlier detection is focused around the idea that outliers are in less dense areas of the data space [28], and that distances can be used to quantify density. Since then, many outlier detection methods have been proposed. We focus our comparison on methods that use the full-dimensional k-nearest neighbors, although many other methods exist [43]. [38] use the distance to the k-nearest neighbor, which can be seen as a "curried" version of the original DB-outlier approach by [28]. [3] use the average distance to all k-nearest neighbors instead. LOF [7] introduced the idea of comparing the density of a point to the densities of its neighbors. LoOP [29] attempts to estimate a local outlier probability, while INFLO [25] also takes reverse nearest neighbor relationships into account, while KDEOS [40] uses kernel density estimation instead of the simpler estimate of aforementioned methods. ODIN [14] simply counts how often a point occurs in the nearest-neighbors of others, while SOS [24] (c.f. Section 3.3) uses the probability of a point not occurring in stochastic neighborhoods as outlier score. Many more variations of these ideas exist [43,8], and a fair evaluation of such methods is extremely difficult, due to the sensitivity of the methods to data sets, preprocessing, and parameterization [8]. There exist many methods that focus on identifying outliers in feature subspaces [30,36,27,10] or with respect to correlations in the data [32].

2.4 Stochastic Neighbor Embedding

Stochastic neighbor embedding (SNE) [16] and t-distributed stochastic neighbor embedding (t-SNE) [35] are visualization techniques designed for visualizing high-dimensional data in a low-dimensional space (typically 2 or 3 dimensions). These methods originate from computer vision and deep learning research where they are used to visualize large image collections. In contrast to techniques such as principal component analysis (PCA) and multidimensional scaling (MDS), which try to maximize the spread of dissimilar objects, SNE focuses on placing similar objects close to each other, i.e., it preserves locality rather than large distances. But while these methods were developed (and used with great success) on data sets with a high representational dimensionality, [35] noted that the "relatively local nature of t-SNE makes it sensitive to the curse of the intrinsic dimensionality of the data" and that "t-SNE might be less successful if it is applied on datasets with a very high intrinsic dimensionality" [35].

The key idea of these methods is to model the high-dimensional input data with an affinity probability distribution, and use gradient descent to optimize the low-dimensional projection to exhibit similar affinities. By using an affinity which has more weight on nearby points rather than Euclidean distance, one obtains a non-linear projection that preserves local neighborhoods, while away points are mostly independent of each other. In SNE, Gaussian kernels are used in the projected space, whereas t-SNE uses a Student-t distribution. This distribution is well suited for the optimization procedure because it is computationally inexpensive, heavier-tailed, and has a well-formed gradient. The heavier tail of t-SNE is beneficial for visualization, because it increases the tendency of the projection to separate unrelated points in the projected space. But as seen in Figure 1, t-SNE does not preserve distances or densities well, so we should rather not use the projected coordinates for clustering or outlier detection.

In the input domain, (t-)SNE uses a Gaussian kernel for the input distribution. Given a point i, the conditional probability density $p_{j|i}$ of any neighbor point j is computed as

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)} \tag{3}$$

where $\|x_i - x_j\|$ is the Euclidean distance, and the kernel bandwidth σ_i is optimized for every point to have the desired perplexity h (an input parameter roughly corresponding to the number of neighbors to preserve). The symmetric affinity probability p_{ij} is then obtained as the average of the conditional probabilities $p_{ij} = \frac{1}{2}(p_{i|j} + p_{j|i})$ and is subsequently normalized such that the total sum is $\sum_{i \neq j} p_{ij} = 1$.

SNE uses a Gaussian distribution (similar to Equation 3, but with constant σ) in the projected space, and t-SNE improved this by using the Student-t distribution instead:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}} \tag{4}$$

The denominator normalizes the sum to a total of $\sum_{i \neq j} q_{ij} = 1$. The mismatch between the two distributions P and Q (given by p_{ij} and q_{ij}) can now be measured using the Kullback-Leibler divergence [16]:

$$\mathrm{KL}(P \,||\, Q) := \sum\nolimits_i \sum\nolimits_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{5}$$

By also using a small constant minimum p_{ij} and q_{ij}, we can prevent unrelated points from being placed too close. To minimize the mismatch of the two distributions, we can use the vector gradient $\frac{\delta C}{\delta y_i}$ (for Student-t / t-SNE, as derived by [35]):

$$\frac{\delta C}{\delta y_i} := 4 \sum\nolimits_j (p_{ij} - q_{ij}) \, q_{ij} \, Z \, (y_i - y_j) \tag{6}$$

where $Z = \sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}$ (c.f. [34]).

Starting with an initial random solution $Y_0 = \{y_i\}$, the solution is then iteratively optimized using gradient descent with learning rate η and momentum α as used by [35]:

$$Y_{t+1} \leftarrow Y_t - \eta \frac{\delta C}{\delta Y} + \alpha \, (Y_t - Y_{t-1}) \tag{7}$$

The resulting projection y is usually good for visualization, because it preserves neighborhood rather well, but also does not place objects too close to each other. The t-distributed variant t-SNE is often subjectively nicer, because the heavier tail of the student-t distribution leads to a more even tendency to separate points, and thus to more evenly fill the available space. The resulting projections in general tend to be circular.

3 Intrinsic Stochastic Neighbors

3.1 Distance Power Transform for the Curse of Intrinsic Dimensionality

The Stochastic Neighbor Embedding approaches are susceptible to the curse, because they use the distance to the neighbors to compute neighbor weights, which will become too similar to be useful at discriminating neighbors. When we lose distance discrimination, it follows from Equation 3 that for a data set of size N: $\lim_{d \to \infty} p_{j|i} \to 1/(N-1)$, $\lim_{d \to \infty} p_{ij} \to 1/(N-1)^2$, and that therefore SNE does no longer work well.

Recent advances in understanding intrinsic dimensionality [17] connect intrinsic dimensionality to modeling the near-neighbor tail of the distance distribution with extreme value theory [18]. An interesting property of intrinsic dimensionality is that it changes with certain transformations [18, Table 1], such as the power transform. Let X be a random variable as in [18], and $g(x) := c \cdot x^m$ with c and m constants. Let F_X be the cumulative distribution of X, $Y = g(X)$ and F_Y the resulting cumulative distribution. Then the intrinsic dimensionality changes by $\mathrm{ID}_{F_X}(x) = m \cdot \mathrm{ID}_{F_Y}(c \cdot x^m)$ [18, Table 1]. By choosing $m = \mathrm{ID}_{F_X}(x)/t$ for any $t > 0$, we therefore obtain:

$$\mathrm{ID}_{F_Y}(c \cdot x^m) = \mathrm{ID}_{F_X}(x)/m = t \tag{8}$$

where we can choose $c > 0$ as desired, e.g., for numerical reasons. This variable X serves a theoretical model for the distance distribution on the "short tail" (the nearest neighbors), and ID_{F_X} is the intrinsic dimensionality. This observation means that we can transform our distance distribution of *any* desired dimensionality t.

In Figure 3, we revisit the theoretical model of a multivariate normal distribution that we used in Section 2.1, but this time we transform the x-axis with a power transform using $m = \sqrt{d}$ and c such that the mean is 1. The power transform yields a transformation that retains the 0 (which the deviation from the mean in Figure 2b did not), but which allows the numerical discrimination of distances. One may have assumed that $m = d$ would be the best choice in this scenario of a d-dimensional hyperball. This holds true in the limit at the center of the ball, but the decreasing density of the Gaussian yields a smaller expansion rate and therefore a decreasing intrinsic dimensionality as we move outward [19]. Beware that this is a very much idealized model, and that in practical applications, we will simply estimate m from the data.

To improve stochastic neighbor approaches, we propose the following remedy to the distance concentration effect: Based on the k-nearest neighbors of the point of interest x, first estimate the local intrinsic dimensionality $\mathrm{ID}(x)$. Then use $d'(x, y_i) := c \cdot d(x, y_i)^m$ (c.f. Equation 8) with $m = \mathrm{ID}(x)/2$ to transform them into squared distances, and $c = 1/\max_y d(x, y)^m$ such that the farthest neighbor always has distance 1. The distances $d(x, y)$ to the neighbors are transformed using

$$d'^2(x, y) = d(x, y)^m / \max_z d(x, z)^m = \left(\tfrac{d(x,y)}{\max_z d(x,z)}\right)^m \tag{9}$$

We then use this locally modified distance instead of the squared Euclidean distance to compute $p_{i|j}$ using Equation 3 (to simplify, we also substitute $\beta_i := -1/2\sigma_i^2$):

$$p'_{j|i} = \frac{\exp(\beta_i d_i'^2(x_i, x_j))}{\sum_{k \neq i} \exp(\beta_i d_i'^2(x_i - x_k))} \tag{10}$$

We can then continue to optimize β_i by binary search to obtain the desired perplexity h as done for regular SNE and t-SNE.

$$\log_2 \text{Perplexity} = -\sum\nolimits_{j \neq i} p_{j|i} \log_2 p_{j|i} \tag{11}$$

On data that was not normalized, regular t-SNE may fail to find a suitable β_i with binary search.[1] This happens when the binary search begins with $\beta_i = -1$ (or $\sigma_i = 1$),

[1] The author of t-SNE writes: "Presumably, your data contains some very large numbers, causing the binary search for the correct perplexity to fail. [...] Just divide your data or distances by a big number, and try again." `https://lvdmaaten.github.io/tsne/#faq`

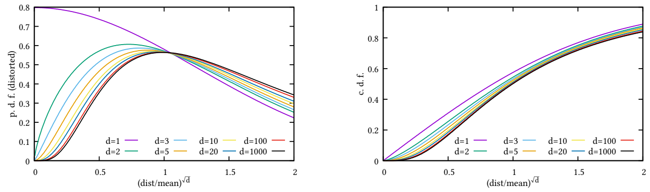


Fig. 3: Deviation from a multivariate standard normal distribution, after power transform.

but $\exp(\beta_i d_i(x_i,x_j))=0$ for all j if the initial distances are too large. With our choice of c this will not happen anymore (as the maximum d' is 1), but the ELKI [39] implementation that we use initializes search with the heuristic estimate $\hat{\beta}_i=-\frac{1}{2}h/\operatorname{mean}_j d(x_i,x_j)^2$ (motivated by $\hat{\sigma}^2 \sim \operatorname{mean}_j d(x_i,x_j)^2$) which usually converges with fewer iterations. Since we only rely on the nearest neighbors, our new approach is compatible with the fast Barnes-Hut approximation [34].

3.2 Consensus Affinity Combination

SNE and t-SNE produce a symmetric affinity by averaging the two asymmetric affinities: $p_{ij}=\frac{1}{2}(p_{i|j}+p_{j|i})$. While this has the desirable property of retaining the total sum, it also tends to pull outliers too close to their neighbors. From a probabilistic point of view, we can interpret this as point x_i and x_j being connected if *either* of them chooses to link. Instead, we may desire them to link only if there is “consensus”, by using

$$p'_{ij} := \sqrt{p'_{i|j} \cdot p'_{j|i}}. \tag{12}$$

The resulting affinity matrix will be more sparse, and therefore it is desirable to use a larger perplexity and neighborhood size than for t-SNE. But since the estimation of intrinsic dimensionality suggests to use at least 100 neighbors, whereas t-SNE is often used with a perplexity of about 40, this is not an additional restriction.

Next, the resulting affinities are normalized to have a total sum of 1 (as in regular t-SNE), to balance attractive and repulsive forces during the t-SNE optimization process. We then simply replace p_{ij} in the gradient (Equation 6) with the new p'_{ij} (Equation 12).

3.3 Intrinsic Stochastic Outlier Selection

The new outlier detection method Intrinsic Stochastic Outlier Selection (ISOS) is—as the name indicates—a modification of the earlier but rather unknown SOS method published in a technical report [23], a PhD thesis [22], and in a maritime application [24]. The key idea of this approach is that every data point “nominates” its neighbors, and can be seen as a smooth version of ODIN [14].

The original proposal of SOS involved generating random graphs based on an affinity distribution in order to identify frequently unlinked objects as outliers. But the expensive graph sampling process can be avoided, and the probability of a node being disconnected can be computed in closed-form using the simple equation [23]:

$$SOS(x_i) := \prod\nolimits_{j \neq i} 1 - p_{i|j} \tag{13}$$

Algorithm 1: Pseudocode for ISOS

```
Input: $DB$: Database
Input: $k$: Number of neighbors to use
Data: $logscore$: Outlier scores, initially 1
Build a neighbor search index on database $DB$ (if not present)
foreach point $x_i$ in database $DB$ do
    $kNN(x_i) \leftarrow$ Find $k$-nearest neighbors (with distances)
    $ID(x_i) \leftarrow$ Estimate intrinsic dimensionality of $kNN(x_i)$
    $d'(x_i) \leftarrow$ Adjust squared distances (Equation 9)
    Choose $\beta_i$ such that perplexity $\approx k/3$
    $p_{j|i} \leftarrow$ Compute normalized affinities (Equation 10)
    foreach neighbor $x_j$ in $kNN(x)$ do
        $logscore(x_j) \leftarrow logscore(x_j) + \log(1 - p_{j|i})$
return $1/\left(1 + e^{-x \cdot \log h} \cdot (1-\varphi)/\varphi\right)$ for each score in $logscore$
```

The original algorithm, similar to SNE and t-SNE, has quadratic runtime complexity, making it expensive to apply to large data. But because of the exponential function, affinities will quickly drop to a negligible value. Van der Maaten [34] uses the $k = \lceil 3h \rceil$ nearest neighbors to approximate the $p_{i|j}$. We incorporate this idea into SOS for two reasons: (i) to improve scalability, and (ii) to make it more comparable to k-nearest neighbor based outlier detection algorithms. Instead of the perplexity parameter h, this variant—which we denote as KNNSOS—has the neighborhood size parameter k common to k-nearest neighbor approaches, and uses a derived perplexity of $h = k/3$. Our ISOS method in turn is an extension of this KNNSOS approach, which uses the k-nearest neighbors first to estimate the local intrinsic dimensionality of each point, then uses Equation 13 with our adjusted affinity $p'_{i|j}$. For $p_{i|j} \ll 1$, Equation 13 does not give a high numerical precision. We therefore suggest to compute the scores in logspace,

$$\log SOS(x_i) := \sum\nolimits_{j \neq i} \log(1 - p_{i|j}) \tag{14}$$

and use the `log1p(-p_i|j)` function if available for increased numerical precision. While SOS yields an outlier probability (which makes the score more interpretable by users [31]), it is not as well-behaved as indicated by its authors [23], because the expected value even for a clear inlier is not 0, since we normalized the $p_{i|j}$ to sum up to 1. Intuitively, every point is supposed to distribute its weight to approximately h (the perplexity) neighbors. Assuming a clique of $h + 1$ objects, each at the same distance such that $p_{i|j} = 1/h$, every point will have the probability

$$E[SOS] := \prod\nolimits_1^h (1 - 1/h) = \left(\tfrac{h-1}{h}\right)^h \approx_{h \to \infty} 1/e \tag{15}$$

Alternatively, we can assume the null model that every point is equidistant, and thus every neighbor is chosen with $p_{i|j} = 1/N$, which yields the same limit. Note that $\log 1/e = -1$, if we perform the same computations in logscale. Therefore, we further propose to normalize to the resulting outlier probabilities, by comparing them to the expected value. The likelihood ratio $SOS(x_i)/E[SOS]$ in logspace yields simply the addition of 1 to the log scores. After this transformation, the average score will be

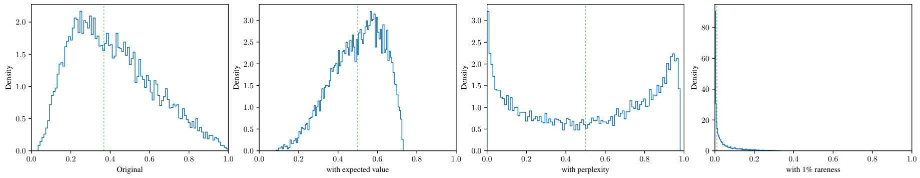


Fig. 4: Histogram of scores on MNIST data. Dotted lines indicate the expected average value.

about 0.5, but centeral values will be too frequent. This is caused by the aggregation over effectively $\approx h$ values, and we can reduce this by multiplication with $\log h$ (in log space). Last but not least, we need to add a prior to reflect that anomalies are rare and not half of the data, but rather the majority of points should have a very low score. We use a desired outlier rate of $\varphi = 1\%$, which yields a prior odds ratio of $(1-\varphi)/\varphi$ [32,40].

To convert this back to a probability, we can use the logistic function:

$$l = -(\log SOS'(x_i) + 1) \cdot \log h \tag{16}$$

$$ISOS(x_i) = 1/\left(1 + \exp(l) \cdot (1-\varphi)/\varphi\right) \tag{17}$$

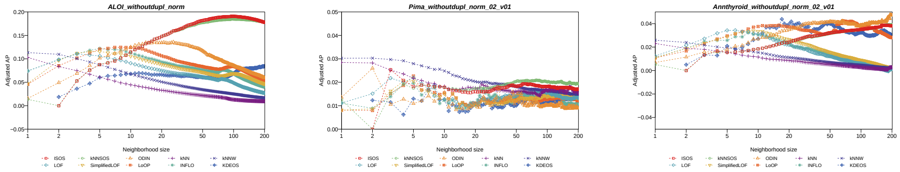
Figure 4 shows (i) the original score before the adjustments on the MNIST test data set, (ii) after adjusting for the expected value (and logistic transformation), (iii) after also taking the perplexity into account, and (iv) with the prior assumption of outliers being 1% rare. The last histogram is the least "informative", but naturally we must expect the majority of outlier scores to be close to zero, so in fact only the exponential-like curve in the final histogram indicates a score that can satisfy the intuition of an "outlier probability". We show the top 50 outliers in Figure 8b.

Algorithm 1 gives the pseudocode for ISOS. Rather than directly computing the score for every point, we initialize all scores with 1 $(= -\log 1/e)$, then iterate over each point x_i and adjust the scores of each neighbor x_j by adding $\log(1 - p_{j|i})$. This reduces the memory requirements from $O(n^2)$ to $O(n)$, and makes the algorithm trivial to distribute except for the nearest neighbor search. For distributed and parallel processing, approximative nearest neighbor search is preferable, and has shown to be surprisingly effective for outlier detection, because errors may be larger for outliers than for inliers [42]. Note that in line 8 we can stop when $p_{j|i}$ is zero, as further away points will no longer change the scores of neighbors. For KNNSOS, do not estimate intrinsic dimensionality in line 4, and use the unmodified distances in line 5 of Algorithm 1.

This is a second-order local outlier detection method (c.f. [41]), where the kNN are used to estimate affinity, and the score depends on the reverse kNN. But because of the efficient message-based algorithm above, we do not need to compute the reverse nearest neighbors (which would require complex indexes for acceleration [11,9]).

4 Experiments

We implemented our approach in Java as part of the ELKI [39] data mining framework, extending the existing Barnes-Hut approximation [34] t-SNE variant, and using the aggregated Hill estimator [21] for intrinsic dimensionality as default.



(a) ALOI data set, AdjAP (normalized, no duplicates, k logscale)

(b) Pima data set, AdjAP (normalized, no duplicates, k logscale)

(c) ANNThyroid data, AdjAP (normalized, no duplicates, k logscale)

Fig. 5: Performance of ISOS and related algorithms on selected data sets.

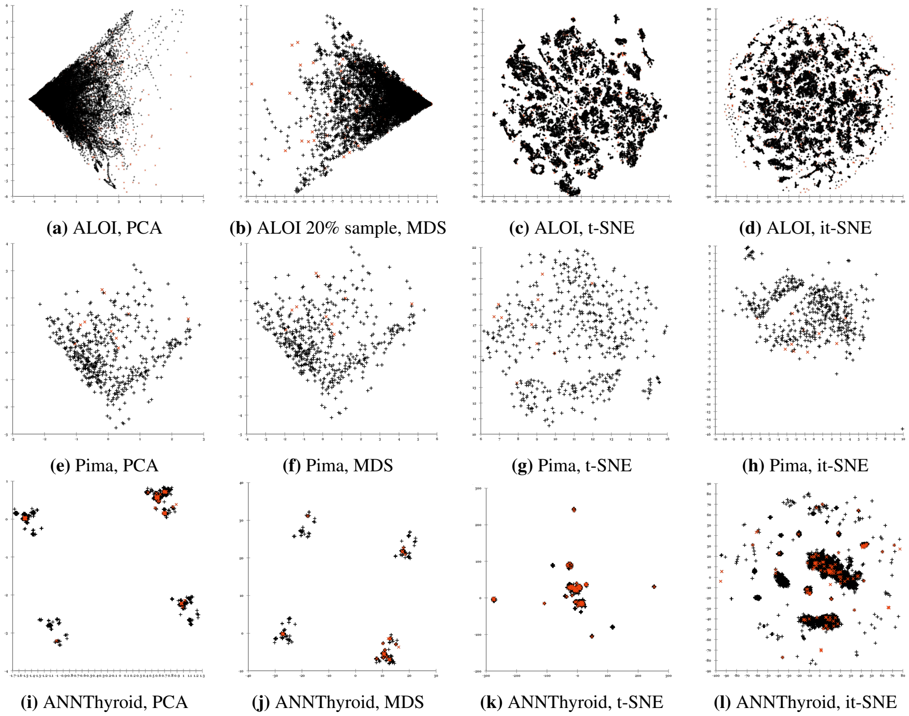
4.1 ISOS Outlier Detection

As common when performing a throrough evaluation of outlier detection, the results here remain unconclusive when performed on a large scale of methods and parameters [8,13]. For any method, we can find parameters and data where it performs best, or worst. KNNSOS and ISOS are, not surprisingly, no exception to this rule. Results claiming superior performance on a task as unspecific as anomaly detection are unfortunately often based on too narrow experiments, and unfair parameterization. We will contribute this method to the ELKI data mining toolkit, and submit the entire results for integration into the benchmark repository of [8].

Figure 5a shows anecdotal evidence of the capabilities of ISOS on the popular ALOI data set (color histograms from images of small objects [12], prepared as in [31]) with respect to adjusted average precision. We show the results for the normalized variant with duplicates removed [8], but the results on the other variants and with other evaluation measures are similar. On this data set, ISOS outperforms all other methods by a considerable margin (except for KNNSOS, which it only outperforms a little bit). Furthermore, the proposed method is fairly stable with respect to the choice of k, as long as the values are not chosen too small (for a reliable estimation of intrinsic dimensionality $k \geq 100$ is suggested). This makes it rather easy to choose the parameters. On other data sets such as Pima (Figure 5b), the simple kNN distance methods work better—although none of the methods really worked well at all. This data set is also likely too small for methods based on intrinsic dimensionality. On ANNThyroid data, KDEOS, LoOP and ODIN compete for the lead, but both KNNSOS and ISOS work reasonably well, too. But again, the results are so low, that the data set must be considered questionable for distance based outlier detection. In Figure 6, we visualize the data sets with PCA, MDS, t-SNE and it-SNE. In none of these projections, the labeled outlier correspond well to the human intuition of outlierness, and we cannot expect any unsupervised algorithm to perform well. For ANNThyroid, we can see artifacts caused by binary attributes in this data set in each projection. In conclusion of the outlier experiments—and in line with prior research [31,8,13]—there is no clear winner, and ensemble approaches that combine kNN outlier, LOF, but also ISOS, remain the most promising research direction.

4.2 it-SNE Visualization

In Figure 7 we apply t-SNE on the popular MNIST data set, using the smaller "test" data set only. Colors indicate different digits. All runs used the same random seed for



(a) ALOI, PCA **(b)** ALOI 20% sample, MDS **(c)** ALOI, t-SNE **(d)** ALOI, it-SNE

(e) Pima, PCA **(f)** Pima, MDS **(g)** Pima, t-SNE **(h)** Pima, it-SNE

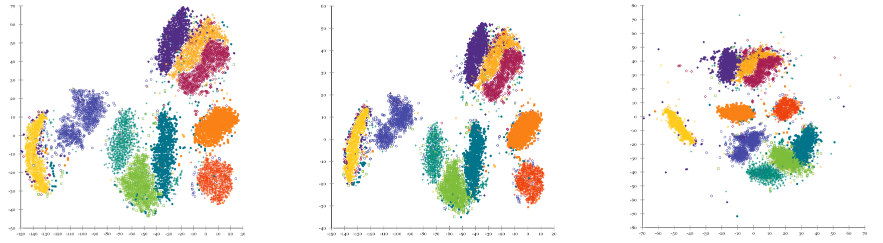
(i) ANNThyroid, PCA **(j)** ANNThyroid, MDS **(k)** ANNThyroid, t-SNE **(l)** ANNThyroid, it-SNE

Fig. 6: Projections of outlier detection data sets. Red x indicate the labeled outliers.

comparability. The difference between regular t-SNE (Figure 7a) and t-SNE with the distances adjusted according to intrinsic dimensionality (Equation 9, Figure 7b) is not very big (classes are slightly more compact in the new projection). This can easily be explained with this data set having nominally 784 dimensions (28 × 28 pixel), but the intrinsic dimensionality is on average just 6.1. Therefore, from an intrinsic dimensionality point of view, it is not a very high-dimensional data set.

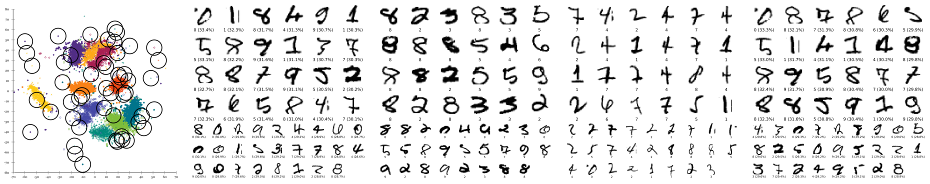
Using the consensus affinity (Equation 12), yields a better result in Figure 7c. Outliers are more pronounced in this visualization, as they are pushed away from all other points rather than attaching themselves to the border of a nearby class (we can also see the same effect in the outlier detection data sets, Figure 6). Because of the overall greater extend, the classes appear more compact. The difference is most pronounced with the yellow class (containing the digit 1), which had many outlier foreign-class attached to it, that are now separate. Why these objects apparently prefer attaching to digit 1 is not clear, but may related to the fact that this class has on average the fewest pixels, the least variation within the class, and the lowest intrinsic dimensionality.

In Figure 8 we visualize the top 50 outliers detected by ISOS, in the it-SNE projection (Figure 8a) as well as the images (Figure 8b), as well as the images for KNN, LOF, and KNNSOS. Most of these outliers were separated from the data classes well by the projection, but we need to keep in mind that the outlier algorithm did not use the projection, and that the projection does not guarantee to separate everything as desired.



(a) t-SNE **(b)** it-SNE with $p_{ij} = p_{i|j} + p_{j|i}$ **(c)** it-SNE with $p_{ij} = \sqrt{p_{i|j} \cdot p_{j|i}}$

Fig. 7: Comparison of MNIST test data (using Barnes-Hut approximations).



(a) ISOS, it-SNE **(b)** ISOS, $k = 150$ **(c)** kNN, $k = 1$ **(d)** LOF, $k = 20$ **(e)** KNNSOS, $k = 150$

Fig. 8: Top 50 outliers in MNIST test data

The rather low scores of $\approx 30\%$ reflect the fact that these outliers are still recognizable digits. Note that these outliers were found based on the raw pixel information. Better results can be expected by using deep learning and class information.

5 Conclusions

This paper contributes important insights into the distance-based aspects of the curse of dimensionality, contributes a much improved outlier detection method, and modifies the popular t-SNE method for intrinsic dimensionality and use in anomaly detection.

- We have shown that the distance concentration effect of the "curse of dimensionality" sometimes can be avoided with a simple power transform.
- The proposed adjustment for intrinsic dimensionality provides more discriminative affinities when using stochastic neighbor approaches on high-dimensional data.
- The "consensus" affinity separates outliers from nearby clusters better, and thus provides substantially better visualization when used for outlier detection, as regular t-SNE would attach outliers to nearby clusters.
- The SOS outlier detection method was accelerated using the k-nearest neighbors (KNNSOS), a correction for intrinsic dimensionality was added (ISOS), and the resulting outlier scores are normalized such that they can be interpreted as a probability how likely an object belongs to a rare "outlier" class.

The use of the power transform is a promising direction to avoid the distance concentration effect in the later stages of data mining, but it is an open research question how a similar improvement could be achieved to improve for example nearest neighbor search. Thus, it is not a universal "cure" to the curse of dimensionality, yet.

References

1. Achtert, E., Kriegel, H., Schubert, E., Zimek, A.: Interactive data mining with 3d-parallel-coordinate-trees. In: ACM SIGMOD (2013)
2. Amsaleg, L., Chelly, O., Furon, T., Girard, S., Houle, M.E., Kawarabayashi, K., Nett, M.: Estimating local intrinsic dimensionality. In: ACM SIGKDD (2015)
3. Angiulli, F., Pizzuti, C.: Fast outlier detection in high dimensional spaces. In: Principles of Data Mining and Knowledge Discovery (2002)
4. Bellman, R.: Adaptive Control Processes. A Guided Tour. Princeton University Press, Princeton (1961)
5. Bennett, K.P., Fayyad, U.M., Geiger, D.: Density-based indexing for approximate nearest-neighbor queries. In: ACM SIGKDD (1999)
6. Beyer, K.S., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is ”nearest neighbor” meaningful? In: Int. Conf. Database Theory ICDT (1999)
7. Breunig, M.M., Kriegel, H., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: ACM SIGMOD (2000)
8. 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 Min. Knowl. Discov. 30(4) (2016)
9. Casanova, G., Englmeier, E., Houle, M.E., Kröger, P., Nett, M., Schubert, E., Zimek, A.: Dimensional testing for reverse k-nearest neighbor search. VLDB Endowment 10 (2017), to appear
10. Dang, X.H., Assent, I., Ng, R.T., Zimek, A., Schubert, E.: Discriminative features for identifying and interpreting outliers. In: IEEE Int. Conf. Data Engineering, ICDE (2014)
11. Emrich, T., Kriegel, H., Kröger, P., Niedermayer, J., Renz, M., Züfle, A.: On reverse-k-nearest-neighbor joins. GeoInformatica 19(2) (2015)
12. Geusebroek, J., Burghouts, G.J., Smeulders, A.W.M.: The amsterdam library of object images. Int. J. of Computer Vision 61(1) (2005)
13. Goldstein, M., Uchida, S.: A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PLOS ONE 11(4) (2016)
14. Hautamäki, V., Kärkkäinen, I., Fränti, P.: Outlier detection using k-nearest neighbour graph. In: Int. Conf. Pattern Recognition, ICPR (2004)
15. Hill, B.M.: A simple general approach to inference about the tail of a distribution. The Annals of Statistics 3(5) (1975)
16. Hinton, G.E., Roweis, S.T.: Stochastic neighbor embedding. In: Adv. in Neural Information Processing Systems 15, NIPS (2002)
17. Houle, M.E.: Dimensionality, discriminability, density and distance distributions. In: ICDM Workshops (2013)
18. Houle, M.E.: Inlierness, outlierness, hubness and discriminability: an extreme-value-theoretic foundation. Tech. Rep. NII-2015-002E, National Institute of Informatics, Tokyo, Japan (2015)
19. Houle, M.E., Kashima, H., Nett, M.: Generalized expansion dimension. In: ICDM Workshops (2012)
20. Houle, M.E., Kriegel, H., Kröger, P., Schubert, E., Zimek, A.: Can shared-neighbor distances defeat the curse of dimensionality? In: Scientific and Statistical Database Management, SSDBM (2010)
21. Huisman, R., Koedijk, K.G., Kool, C.J.M., Palm, F.: Tail-index estimates in small samples. Business & Economic Statistics 19(2) (2001)
22. Janssens, J.H.M.: Outlier selection and one-class classification. Ph.D. thesis, Tilburg University (2013)

23. Janssens, J.H.M., Huszár, F., Postma, E.O., van den Herik, H.J.: Stochastic outlier selection. Tech. Rep. TiCC TR 2012–001, Tilburg Center for Cognition and Communication (2012)
24. Janssens, J.H.M., Postma, E.O., van den Herik, H.J.: Density-based anomaly detection in the maritime domain. In: Situation Awareness with Systems of Systems (2013)
25. Jin, W., Tung, A.K.H., Han, J., Wang, W.: Ranking outliers using symmetric neighborhood relationship. In: Pacific-Asia Conference on Adv. in Knowledge Discovery and Data Mining, PAKDD (2006)
26. Karger, D.R., Ruhl, M.: Finding nearest neighbors in growth-restricted metrics. In: ACM Symp. on Theory of Computing, STOC (2002)
27. Keller, F., Müller, E., Böhm, K.: Hics: High contrast subspaces for density-based outlier ranking. In: IEEE Int. Conf. Data Engineering (2012)
28. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: Very Large Data Bases, VLDB (1998)
29. Kriegel, H., Kröger, P., Schubert, E., Zimek, A.: Loop: local outlier probabilities. In: ACM Conf. Information and Knowledge Management (2009)
30. Kriegel, H., Kröger, P., Schubert, E., Zimek, A.: Outlier detection in axis-parallel subspaces of high dimensional data. In: Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining, PAKDD (2009)
31. Kriegel, H., Kröger, P., Schubert, E., Zimek, A.: Interpreting and unifying outlier scores. In: SIAM Data Mining, SDM (2011)
32. Kriegel, H., Kröger, P., Schubert, E., Zimek, A.: Outlier detection in arbitrarily oriented subspaces. In: IEEE Int. Conf. Data Mining (2012)
33. Low, T., Borgelt, C., Stober, S., Nürnberger, A.: The hubness phenomenon: Fact or artifact? In: Towards Advanced Data Analysis by Combining Soft Computing and Statistics. Studies in Fuzziness and Soft Computing, Springer (2013)
34. van der Maaten, L.: Accelerating t-SNE using tree-based algorithms. J. Machine Learning Research 15(1) (2014)
35. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Machine Learning Research 9(11) (2008)
36. Nguyen, H.V., Gopalkrishnan, V., Assent, I.: An unbiased distance-based outlier detection approach for high-dimensional data. In: Database Systems for Advanced Applications, DASFAA (2011)
37. Radovanovic, M., Nanopoulos, A., Ivanovic, M.: Hubs in space: Popular nearest neighbors in high-dimensional data. J. Machine Learning Research 11 (2010)
38. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: ACM SIGMOD (2000)
39. Schubert, E., Koos, A., Emrich, T., Züfle, A., Schmid, K.A., Zimek, A.: A framework for clustering uncertain data. VLDB Endowment 8(12) (2015), `https://elki-project.github.io/`
40. Schubert, E., Zimek, A., Kriegel, H.: Generalized outlier detection with flexible kernel density estimates. In: SIAM Data Mining (2014)
41. Schubert, E., Zimek, A., Kriegel, H.: Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. Data Min. Knowl. Discov. 28(1) (2014)
42. Schubert, E., Zimek, A., Kriegel, H.: Fast and scalable outlier detection with approximate nearest neighbor ensembles. In: Database Systems for Advanced Applications, DASFAA (2015)
43. Zimek, A., Schubert, E., Kriegel, H.: A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining 5(5) (2012)