Intrinsic t-Stochastic Neighbor Embedding for Visualization and Outlier Detection

A Remedy Against the Curse of Dimensionality?

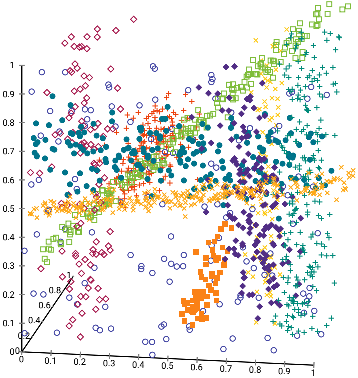
Erich Schubert, Michael Gertz

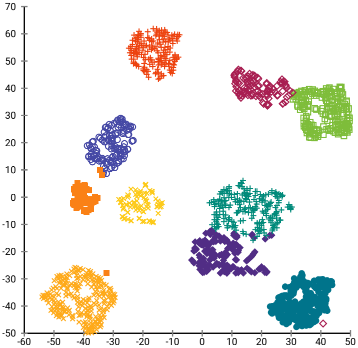
October 4, 2017, Munich, Germany

Heidelberg University

t-Stochastic Neighbor Embedding

t-SNE [MH08], based on SNE [HR02] is a popular “neural network” visualization technique using stochastic gradient descent (SGD)



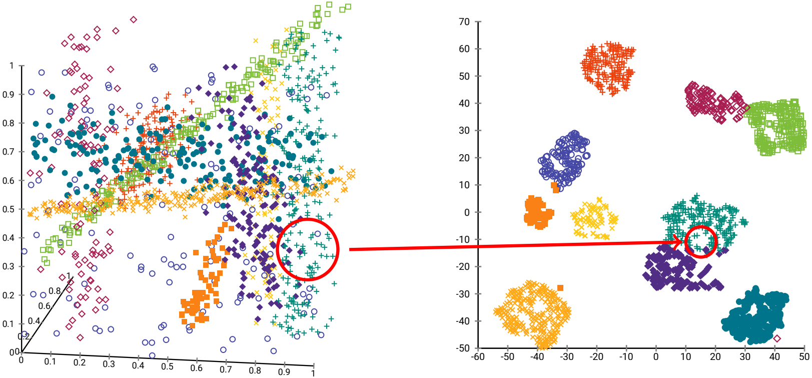


10 dimensional space ⟶ 2 dimensional space

Tries to preserve the neighbors – but not the distances.

t-Stochastic Neighbor Embedding

t-SNE [MH08], based on SNE [HR02] is a popular “neural network” visualization technique using stochastic gradient descent (SGD)

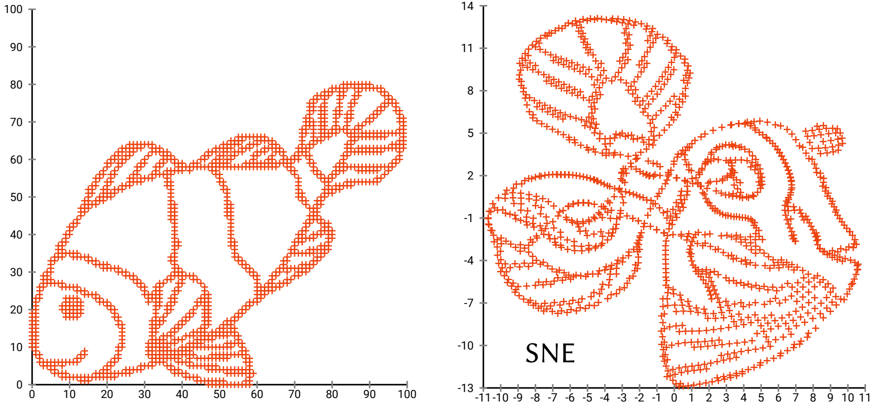


10 dimensional space ⟶ 2 dimensional space

Tries to preserve the neighbors – but not the distances.

t-Stochastic Neighbor Embedding

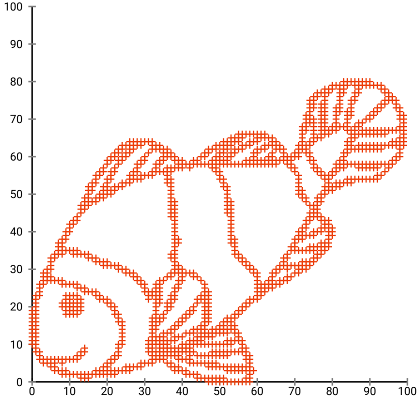
SNE [HR02] and t-SNE [MH08] are popular "neural network" visualization techniques using stochastic gradient descent (SGD)

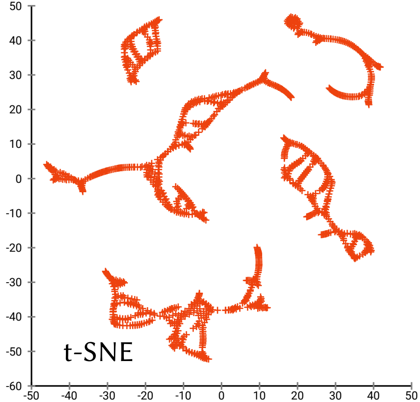


SNE/t-SNE do not preserve density / distances.

t-Stochastic Neighbor Embedding

SNE [HR02] and t-SNE [MH08] are popular “neural network” visualization techniques using stochastic gradient descent (SGD)





SNE/t-SNE do not preserve density / distances.

SNE and t-SNE use a Gaussian kernel in the input domain:

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

where each σ_i^2 is optimized to have the desired perplexity

(Perplexity $\approx$ number of neighbors to preserve)

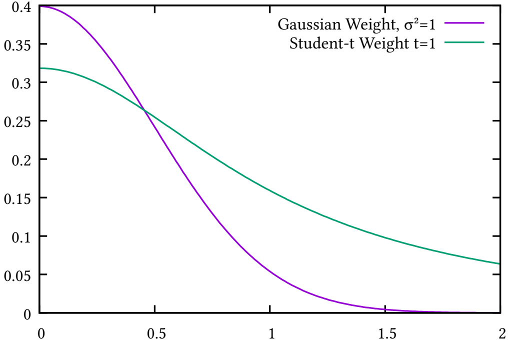
Asymmetric, so they simply use: $p_{ij} := (p_{i|j} + p_{j|i})/2$
(We suggest to prefer $p_{ij} = \sqrt{p_{i|j} \cdot p_{j|i}}$ for outlier detection)

In the output domain, as q_{ij}, SNE uses a Gaussian (with constant σ), t-SNE uses a Student-t-Distribution.

➨ Kullback-Leibler divergence can be minimized using stochastic gradient descent to make input and output affinities similar.

SNE vs. t-SNE

Gaussian weights in the output domain as used by SNE vs. t-SNE:



t-SNE has more emphasis on separating points.

- even neighbors will be “fanned out” a bit
- “better” separation of far points (SNE has 0 weight on far points)

Loss of "discrimination" of distances [Bey+99]:

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

➡ Distances to near points and to far points become similar.

The Curse of Dimensionality

Loss of "discrimination" of distances [Bey+99]:

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

- Distances to near points and to far points become similar.

The Gaussian kernel uses relative distances:

$$\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)$$

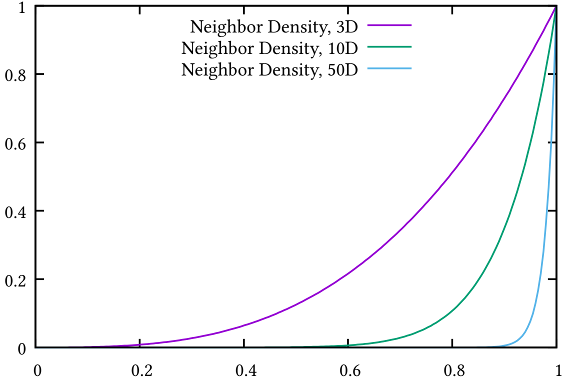
Distance Expected Distance

With high-dimensional data, all p_{ij} become similar!

- We cannot find a "good" σ_i anymore.

Distribution of Distances

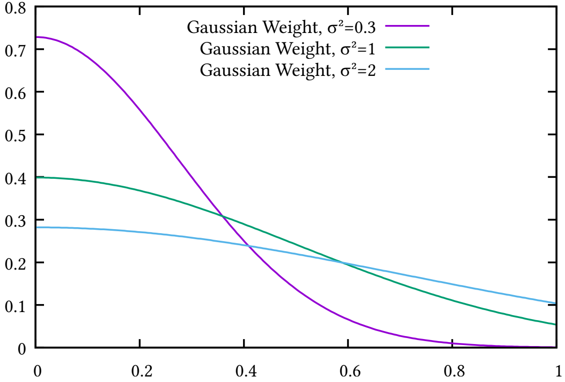
On the short tail distance distributions often look like this:



In high-dimensional data, almost all nearest neighbors concentrate on the right hand side of this plot.

Distribution of Distances

Gaussian weights as used by SNE / t-SNE:

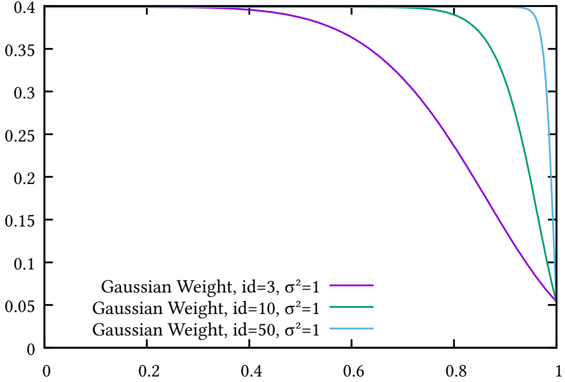


For low-dimensional data, Gaussian weights work good.

For high-dimensional data: almost the same weight for all points.

Distribution of Distances

Gaussian kernels adjusted for intrinsic dimensionality:



In theory, they behave like Gaussian kernels in low dimensionality.

Distance Power Transform

Let X be a random variable ("of distances") as in [Hou15],

For constants c and m, use the transformation

$$Y = g(X) \qquad \text{with } g(x) := c \cdot x^m$$

Let F_X, F_Y be the cumulative distribution of X, Y.

Then $\quad \mathrm{ID}_{F_X}(x) = m \cdot \mathrm{ID}_{F_Y}(c \cdot x^m) \quad$ [Hou15, Table 1].

By choosing $m = \mathrm{ID}_{F_X}(x)/t$ for any $t > 0$, one therefore obtains:

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

where one can choose $c > 0$ as desired, e.g., for numerical reasons.

➡ We can transform distances to any desired $\mathrm{ID} = t$!

Distance Power Transform

For each point p:

1. Find k' nearest neighbors of p (should be $k' > 100$, $k' > k$)
2. Estimate ID at p
3. Choose $m = \mathrm{ID}_{F_X}(x)/t$, $t = 2$, $c = k$-distance
4. Transform distances:
$$d'(p,q) := c \cdot d(p,q)^m$$
5. Use Gaussian kernel, perplexity, t-SNE, …

Can we defeat the curse this easily?

Distance Power Transform

For each point p:

1. Find k' nearest neighbors of p (should be $k' > 100$, $k' > k$)
2. Estimate ID at p
3. Choose $m = \mathrm{ID}_{F_X}(x)/t$, $t = 2$, $c = k$-distance
4. Transform distances:
$$d'(p,q) := c \cdot d(p,q)^m$$
5. Use Gaussian kernel, perplexity, t-SNE, …

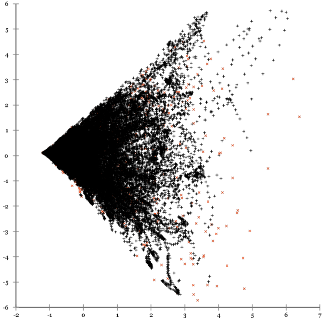
Can we defeat the curse this easily?

Probably not: this is a hack to cure one symptom.

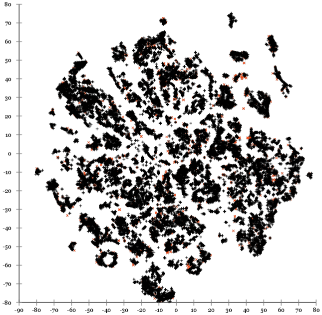
Question: is our definition of ID too permissive?

Experimental Results: it-SNE

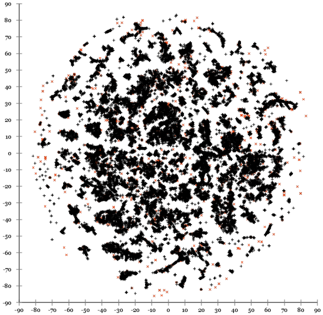
Projections of the ALOI outlier data set (as available at [Cam+16]):



PCA



t-SNE



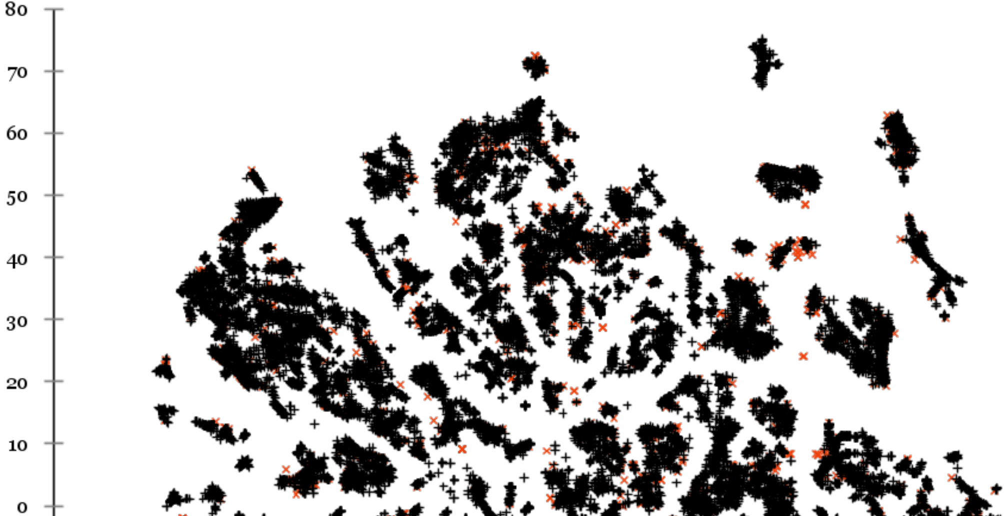
it-SNE

Data set: Color histograms of 50.000 photos of 1000 objects

Each class: same object, different angles & different light

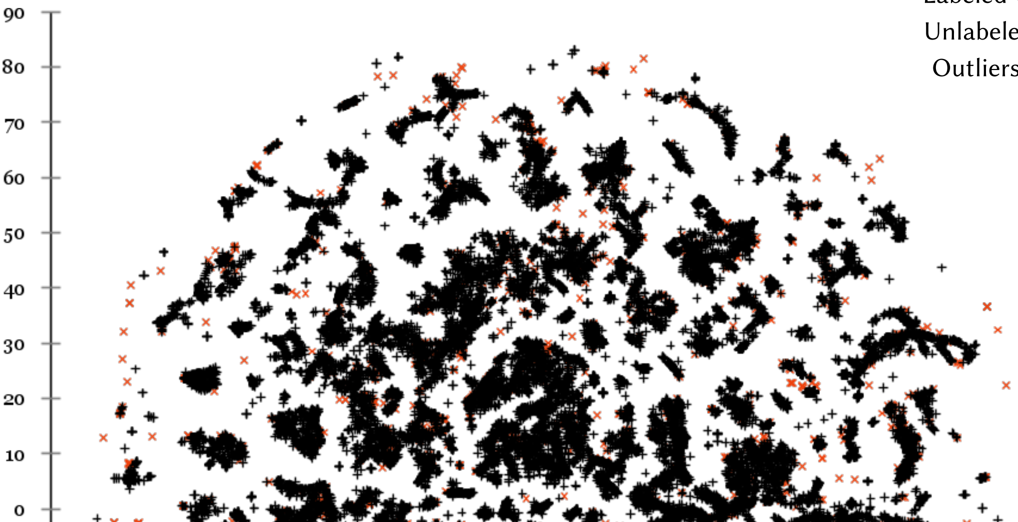
Labeled outliers: classes reduced to 1-3 objects — May contain other “true” outliers!

Projection of the ALOI outlier data set with t-SNE:



Experimental Results: it-SNE

Projection of the ALOI outlier data set with it-SNE:



Labeled & Unlabeled Outliers!

Experimental Results: it-SNE

On the well-known MNIST data set t-SNE:

On the well-known MNIST data set it-SNE:

Outliers!

Outlier Detection: ODIN

ODIN (Outlier Detection using Indegree Number) [HKF04]:

1. Find the k nearest neighbors of each object.
2. Count how often each object was returned.
   = in-degree of the k nearest neighbor graph
3. Objects with no (or fewest) occurrences are outliers.

Works, but many objects will have the exact same score.

Which k to use? Can change abruptly with k.

Can we make a continuous ("smooth") version of this idea?

SOS (Stochastic Outlier Selection) [JPH13]

Idea: assume every object can link to one neighbor randomly.

Inliers: likely to be linked to, outliers: likely to be not linked to.

1. Compute $p_{j|i}$ of SNE / t-SNE for all i, j:
$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$$
use Gaussian weights to prefer near neighbors.
2. The SOS outlier score is then:
$$\mathrm{SOS}(x_j) := \prod\nolimits_{i \neq j} 1 - p_{j|i}$$
= probability that no neighbor links to object j.

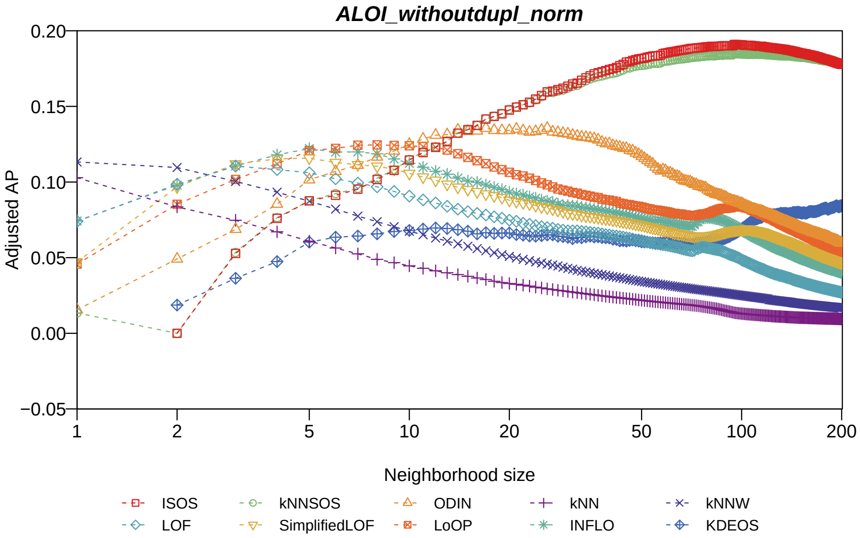
KNNSOS and ISOS Outlier Detection

We propose two variants of this idea:

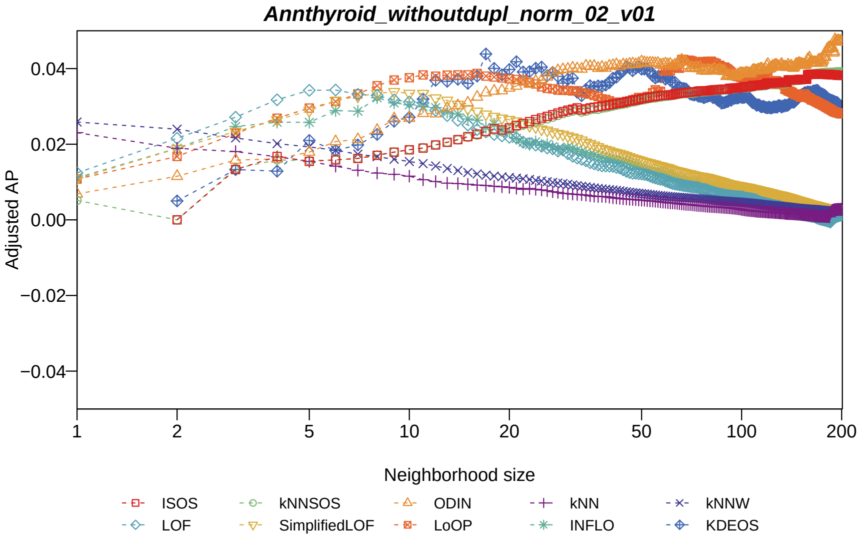
1. Since most $p_{j|i}$ will be zero, use only the k nearest neighbors.
   Reduces runtime from $O(n^2)$ to possibly $O(n \log n)$, $O(n^{4/3})$.
   $$\mathrm{KNNSOS}(x_j) := \prod\nolimits_{i \in k\mathsf{NN}(x_j)} 1 - p_{j|i}$$
2. Estimate $\mathrm{ID}(x_i)$, and use transformed distances for $p_{j|i}$.
   ISOS: Intrinsic-dimensionality Stochastic Outlier Selection

Note: The t-SNE author, van der Maaten, already proposed an approximate and index-based variant of t-SNE:
Barnes-Hut t-SNE, which also uses the kNN only [Maa14].

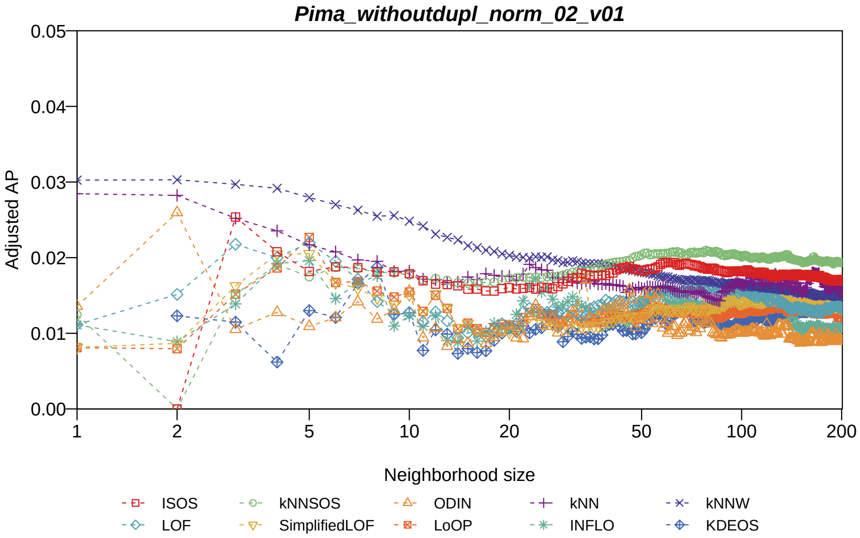
Experimental Results: Outlier Detection



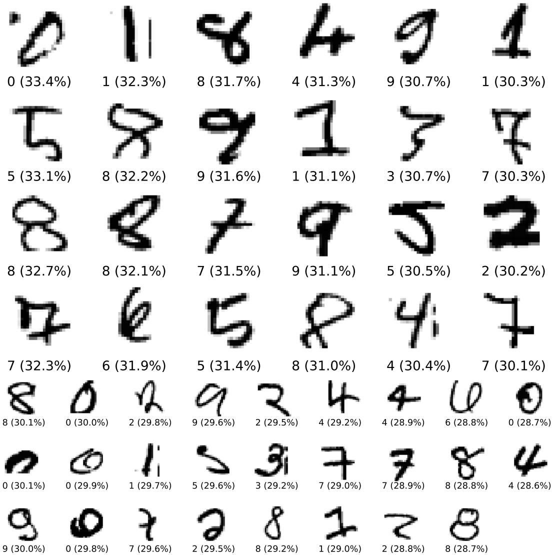
Experimental Results: Outlier Detection



Experimental Results: Outlier Detection



Experimental Results: Outliers in MNIST



Conclusions

- We can "reduce" intrinsic dimensionality to $\mathrm{ID} = t$ using:

  $$m = \mathrm{ID}_{F_X}(x)/t$$

  But is this more than a cure for a symptom (for our estimate)?
- t-SNE benefits from this adjustment:
  We get more difference in neighbor weights.
  (We can also use SNE, but we did not experiment with this.)
- t-SNE tends to hide outliers, unless we use

  $$p_{ij} = \sqrt{p_{i|j} \cdot p_{j|i}} \quad \text{instead of} \quad p_{ij} = \tfrac{1}{2}(p_{i|j} + p_{j|i})$$

- We can make SOS outlier faster using the KNN only
- ISOS improves SOS by adjusting for ID.

Thank You!

Questions?

Thank You!

Questions?

How do we fix ID?

References i

[Bey+99] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. "When Is "Nearest Neighbor" Meaningful?" In: Int. Conf. Database Theory ICDT. 1999.

[Cam+16] G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle. "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study". In: Data Min. Knowl. Discov. 30.4 (2016).

[HKF04] V. Hautamäki, I. Kärkkäinen, and P. Fränti. "Outlier Detection Using k-Nearest Neighbour Graph". In: Int. Conf. Pattern Recognition, ICPR. 2004.

[Hou15] M. E. Houle. Inlierness, outlierness, hubness and discriminability: an extreme-value-theoretic foundation. Tech. rep. NII-2015-002E. National Institute of Informatics, Tokyo, Japan, 2015.

[HR02] G. E. Hinton and S. T. Roweis. "Stochastic Neighbor Embedding". In: Adv. in Neural Information Processing Systems 15, NIPS. 2002.

[JPH13] J. H. M. Janssens, E. O. Postma, and H. J. van den Herik. "Density-Based Anomaly Detection in the Maritime Domain". In: Situation Awareness with Systems of Systems. 2013.

References ii

[Maa14] L. van der Maaten. "Accelerating t-SNE using tree-based algorithms". In: J. Machine Learning Research 15.1 (2014).

[MH08] L. van der Maaten and G. Hinton. "Visualizing Data using t-SNE". In: J. Machine Learning Research 9.11 (2008).