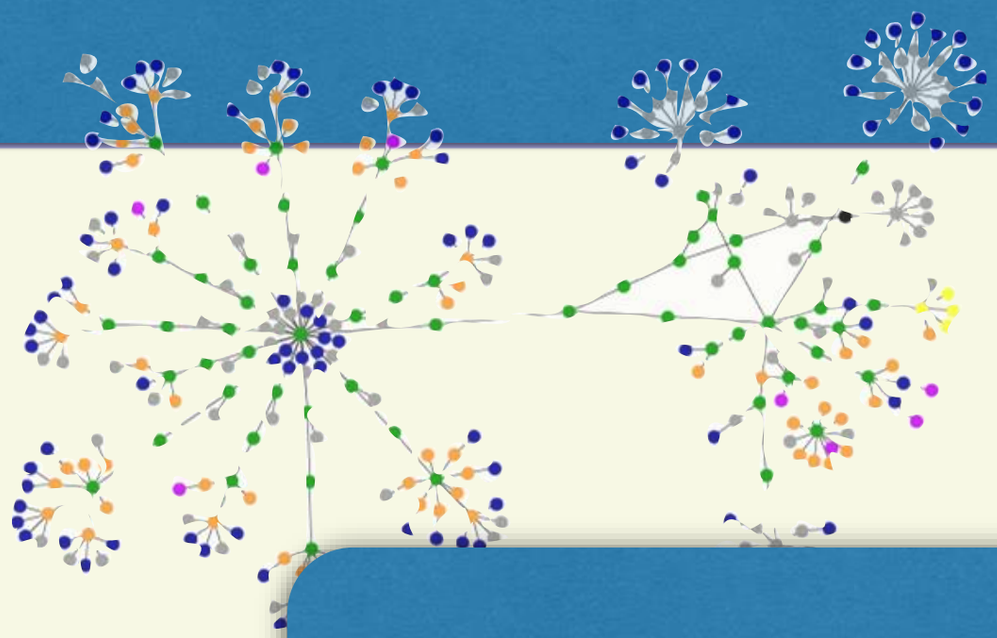
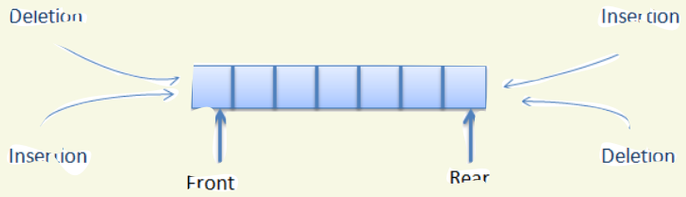
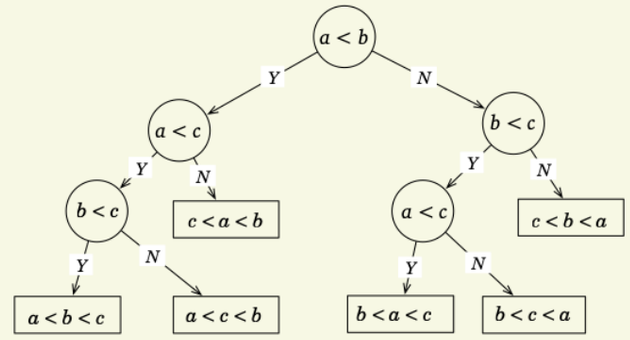






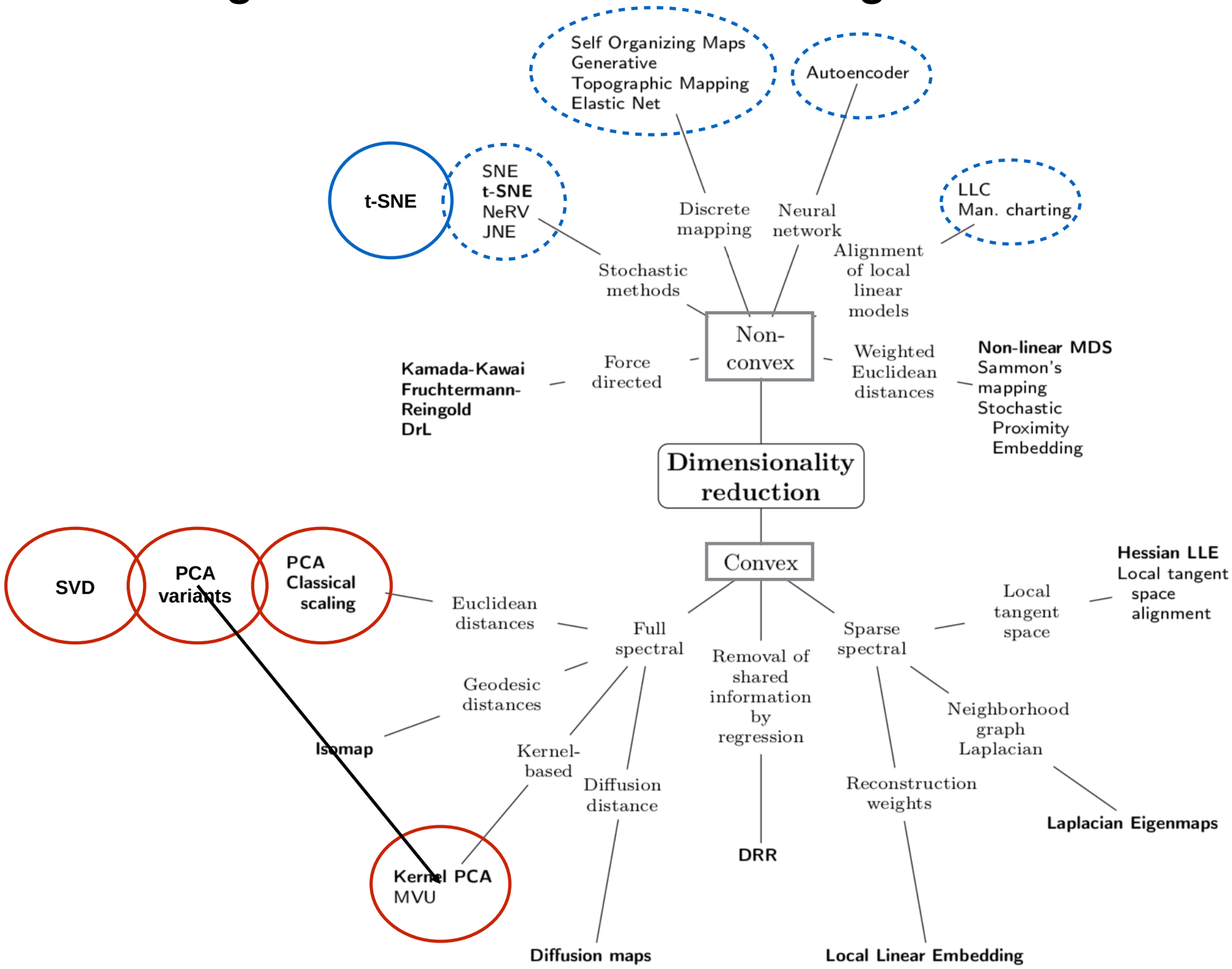
COURSE
“Técnicas Matemáticas para Big Data”
University of Aveiro

Algorithm 2.1: INSERTION-SORT(A)

```
1  for j ← 2 to A.size do
2      key ← A[j]
       // Insert A[j] into the sorted sequence A[1 . . j − 1
3      i ← j − 1
4      while i > 0 and A[i] > key do
5          A[i + 1] ← A[i]
6          i−−
7      A[i + 1] ← key
```

- Dimension Reduction:
  - t-Distributed Stochastic Neighbour Embedding
  - t-SNE vs PCA
  - Practical notebook

Brief Catalog of Dimension Reduction Algorithms



Brief Catalog of Dimension Reduction Algorithms

1. Methods based on Statistics and Information Theory

- 1.1. Vector quantisation and mixture models
- 1.2. Principal component analysis (PCA)
- 1.3. Singular value decomposition (SVD)
- 1.4. Factor Analysis
- 1.5. Principal curves, surfaces and manifolds
- 1.6. Generative topographic mapping
- 1.7. Self-organising maps

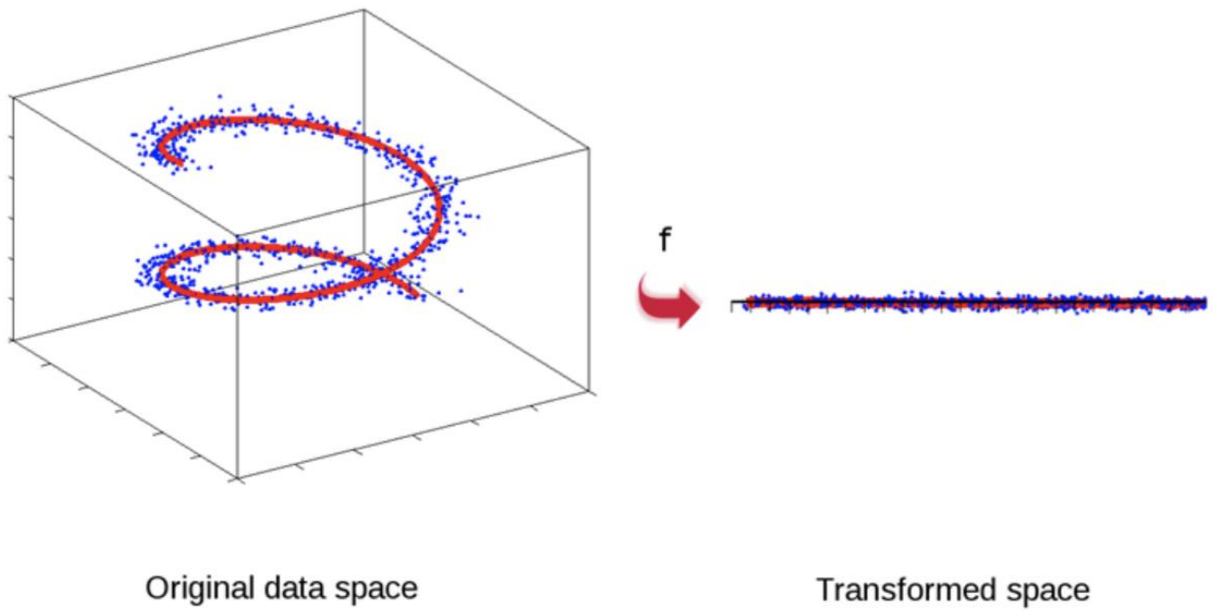
→

- 1.2a Incremental, stream or online PCA
- 1.2b Nonlinear PCA
- 1.2c PCA rotations and Sparse PCA
- 1.2d Localised PCA and subspace segmentation
- 1.2e Kernel PCA and multidimensional scaling
- 1.2f Robust PCA
- . . .

- 1.8. Elastic maps, nets, principal graphs and principal trees
- 1.9. Kernel entropy component analysis

Brief Idea (Principal curve)

If data do not lie in a linear manifold, approximating the curve by a straight line will not perform a good approximation of the original data



Brief Catalog of Dimension Reduction Algorithms

1. Methods based on Statistics and Information Theory

- **1.1. Vector quantisation and mixture models**
- **1.2. Principal component analysis (PCA)**
- **1.3. Singular value decomposition (SVD)**
- **1.4. Factor Analysis**
- **1.5. Principal curves, surfaces and manifolds**
- **1.6. Generative topographic mapping**
- **1.7. Self-organising maps**
- **1.8. Elastic maps, nets, principal graphs and principal trees**
- **1.9. Kernel entropy component analysis**

- **1.2a Incremental, stream or online PCA**
- **1.2b Nonlinear PCA**
- **1.2c PCA rotations and Sparse PCA**
- **1.2d Localised PCA and subspace segmentation**
- **1.2e Kernel PCA and multidimensional scaling**
- **1.2f Robust PCA**
- . . .

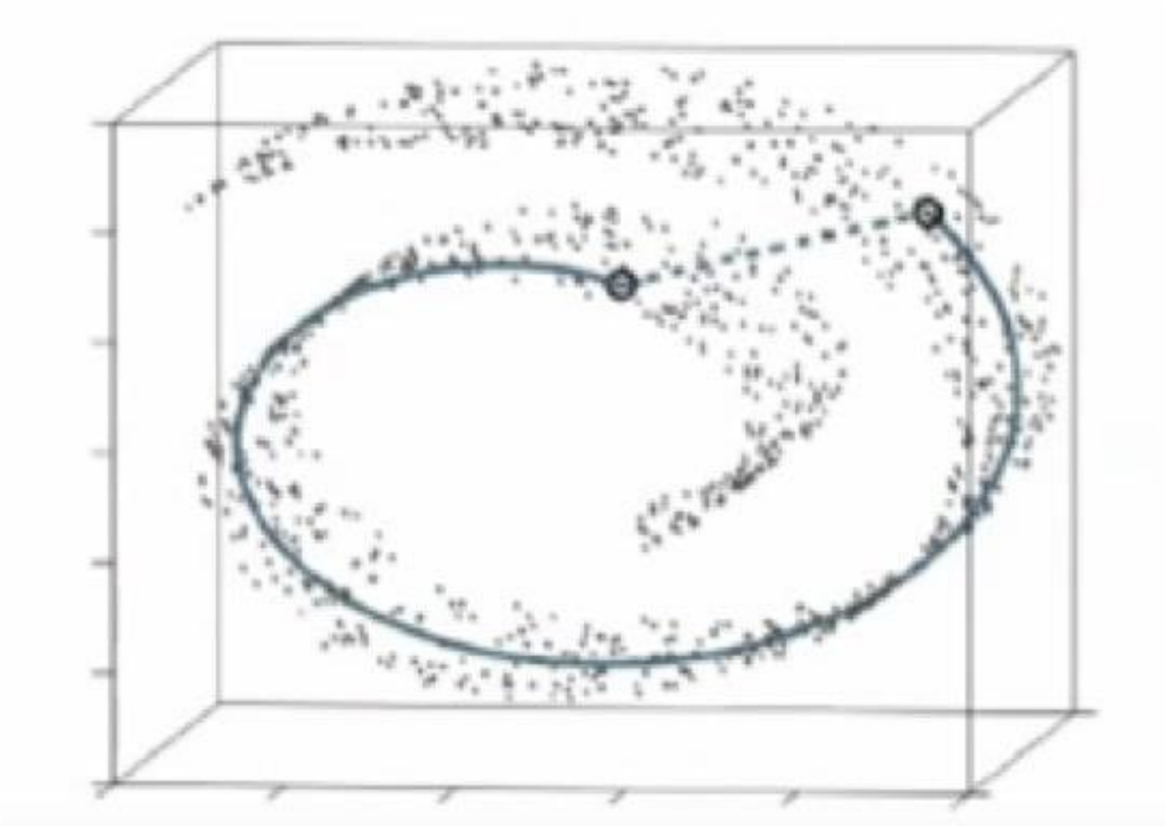
Similar idea: **t-SNE**

t-Distributed Stochastic Neighbor Embedding (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualisation of high-dimensional data. In simpler terms, t-SNE gives you a feel or intuition of how the data is arranged in a high-dimensional space. It was developed by Laurens van der Maatens and Geoffrey Hinton in 2008.

t-SNE vs PCA

PCA is a linear dimension reduction technique that seeks to maximize variance and preserves large pairwise distances. In other words, things that are different end up far apart. This can lead to poor visualisation especially when dealing with non-linear manifold structures. Think of a manifold structure as any geometric shape like: cylinder, ball, curve, etc.

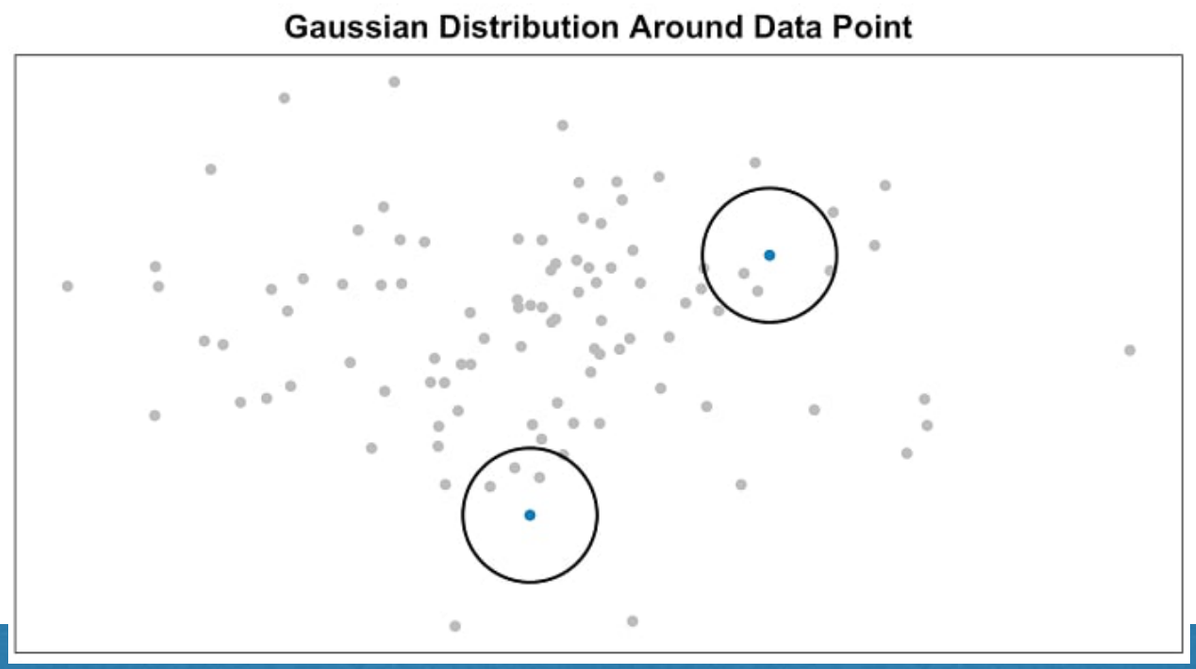
t-SNE differs from PCA by preserving only small pairwise distances or local similarities whereas PCA is concerned with preserving large pairwise distances to maximize variance. Laurens illustrates the PCA and t-SNE approach pretty well using the Swiss Roll dataset in Figure 1. You can see that due to the non-linearity of this toy dataset (manifold) and preserving large distances that PCA would incorrectly preserve the structure of the data.



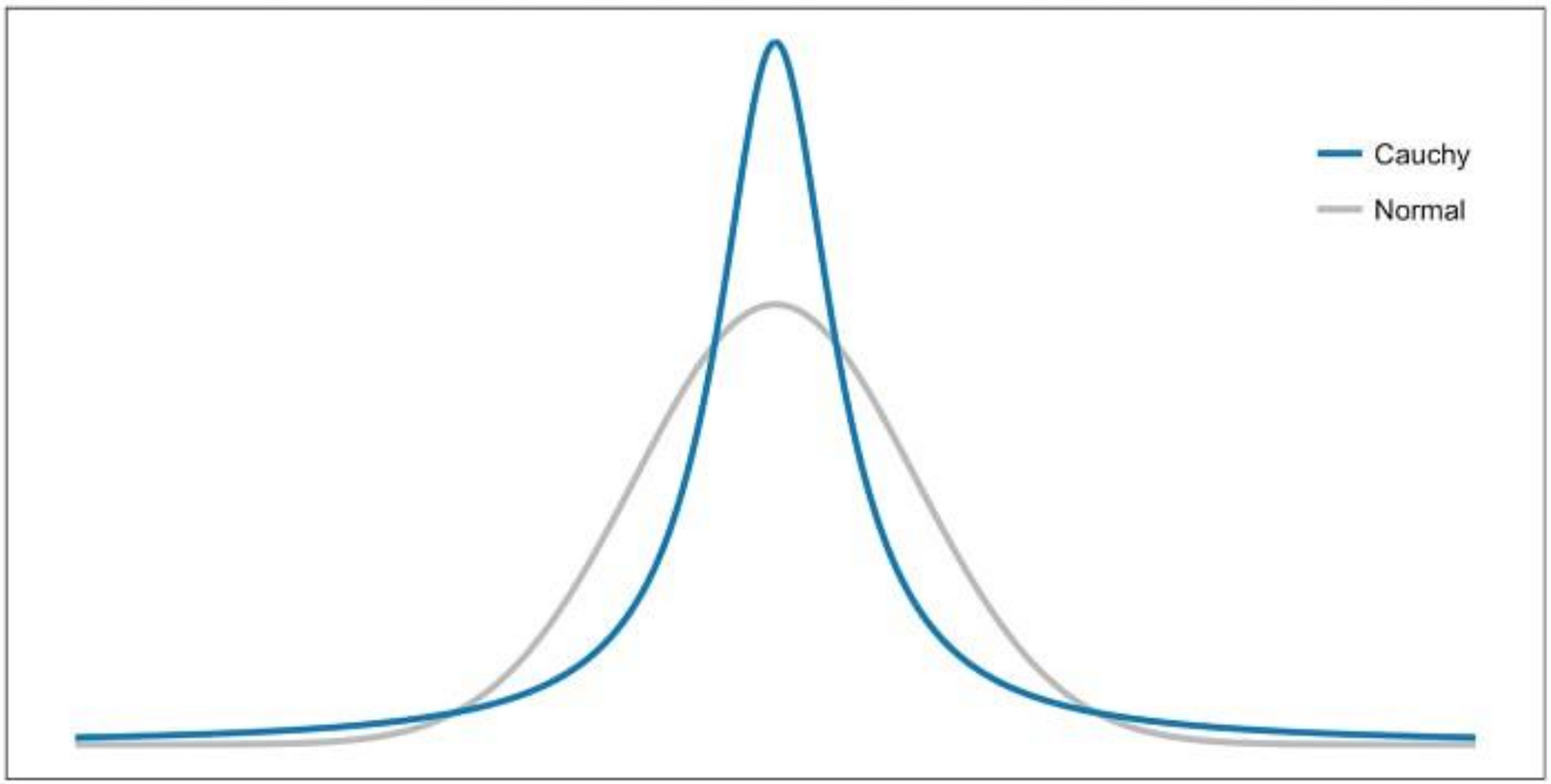
How t-SNE works

The t-SNE algorithm calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. It then tries to optimize these two similarity measures using a cost function. Let's break that down into 3 basic steps.

Step 1: Measure similarities between points in the high dimensional space. For each data point (x_i), center a Gaussian distribution over that point, then measure the density of all points (x_j) under that Gaussian distribution. Then renormalise for all points. This gives us a set of probabilities (P_{ij}) for all points. Those probabilities are proportional to the similarities. All that means is, if data points x1 and x2 have equal values under this Gaussian circle then their proportions and similarities are equal and hence you have local similarities in the structure of this high-dimensional space. The Gaussian distribution or circle can be manipulated using what's called perplexity, which influences the variance of the distribution (circle size) and essentially the number of nearest neighbours. Usual range for perplexity is between 5 and 50.



Step 2: Similar to step 1, but instead of using a Gaussian distribution you use a Student t-distribution with one degree of freedom, which is also known as the Cauchy distribution (see figure). This gives us a second set of probabilities (Q_{ij}) in the low dimensional space. As you can see the Student t-distribution has heavier tails than the normal distribution. The heavy tails allow for better modelling of far apart distances.



Step 3: It is required that the set of probabilities from the low-dimensional space (Q_{ij}) to reflect those of the high dimensional space (P_{ij}) as best as possible, i.e. the two map structures to be similar. Measure the difference between the probability distributions of the two-dimensional spaces using Kullback-Liebler divergence (KL); which is an asymmetrical approach that efficiently compares large P_{ij} and Q_{ij} values.

Finally, use gradient descent to minimize the KL cost function since KL Divergence measure how much information we lose when we choose a distribution vs other.

(see original paper in note N03 or in the url)

Visualizing Data using t-SNE

Laurens van der Maaten L.VANDERMAATEN@MICC.UNIMAAS.NL
MICC-IKAT
Maastricht University
P.O. Box 616, 6200 MD Maastricht, The Netherlands

Geoffrey Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
6 King's College Road, M5S 3G4 Toronto, ON, Canada

Work on the notebooks:

- C04-t-SNEtemplate and try to apply (1) PCA to the dataset and (2) t-SNE.