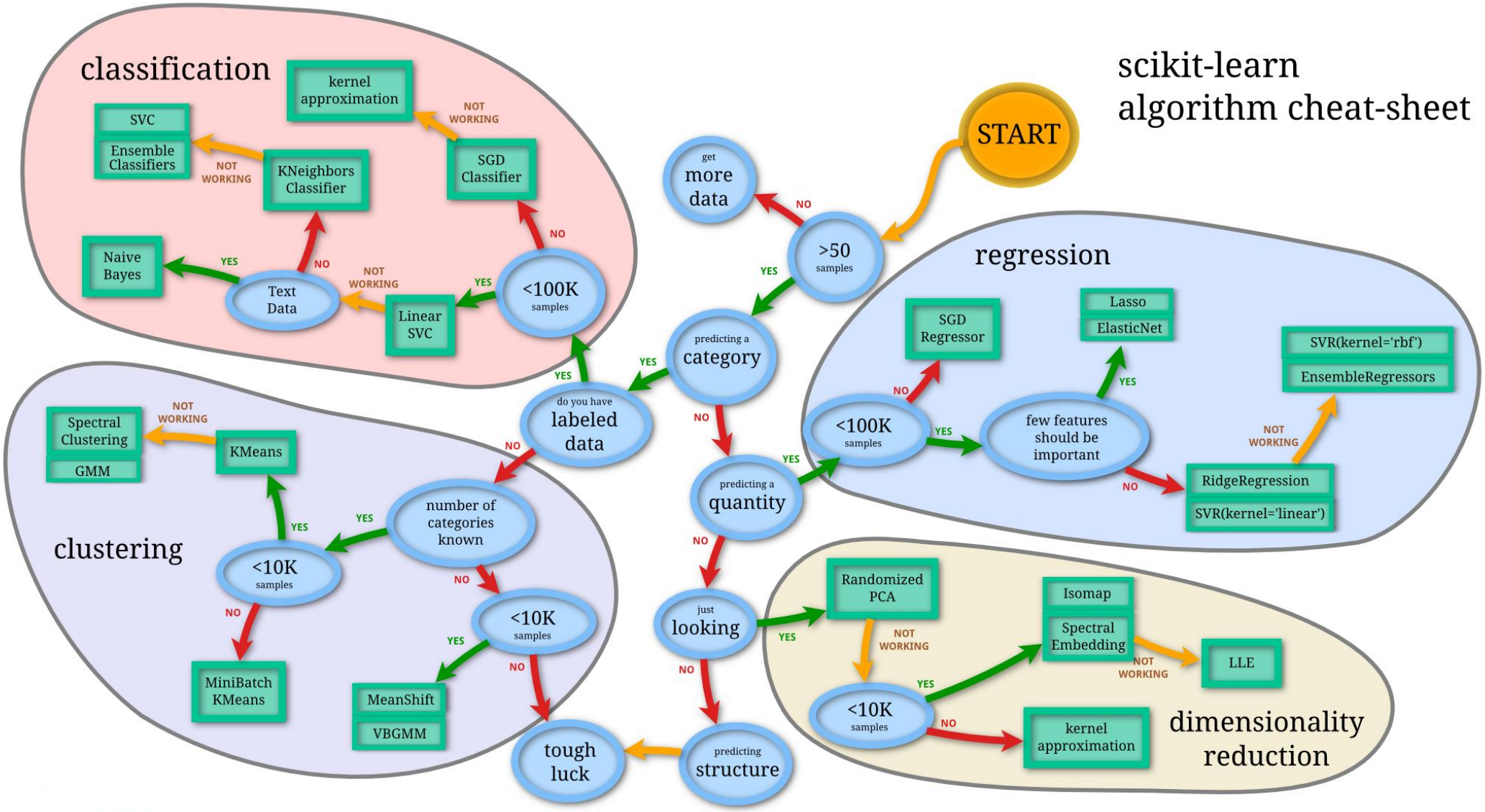
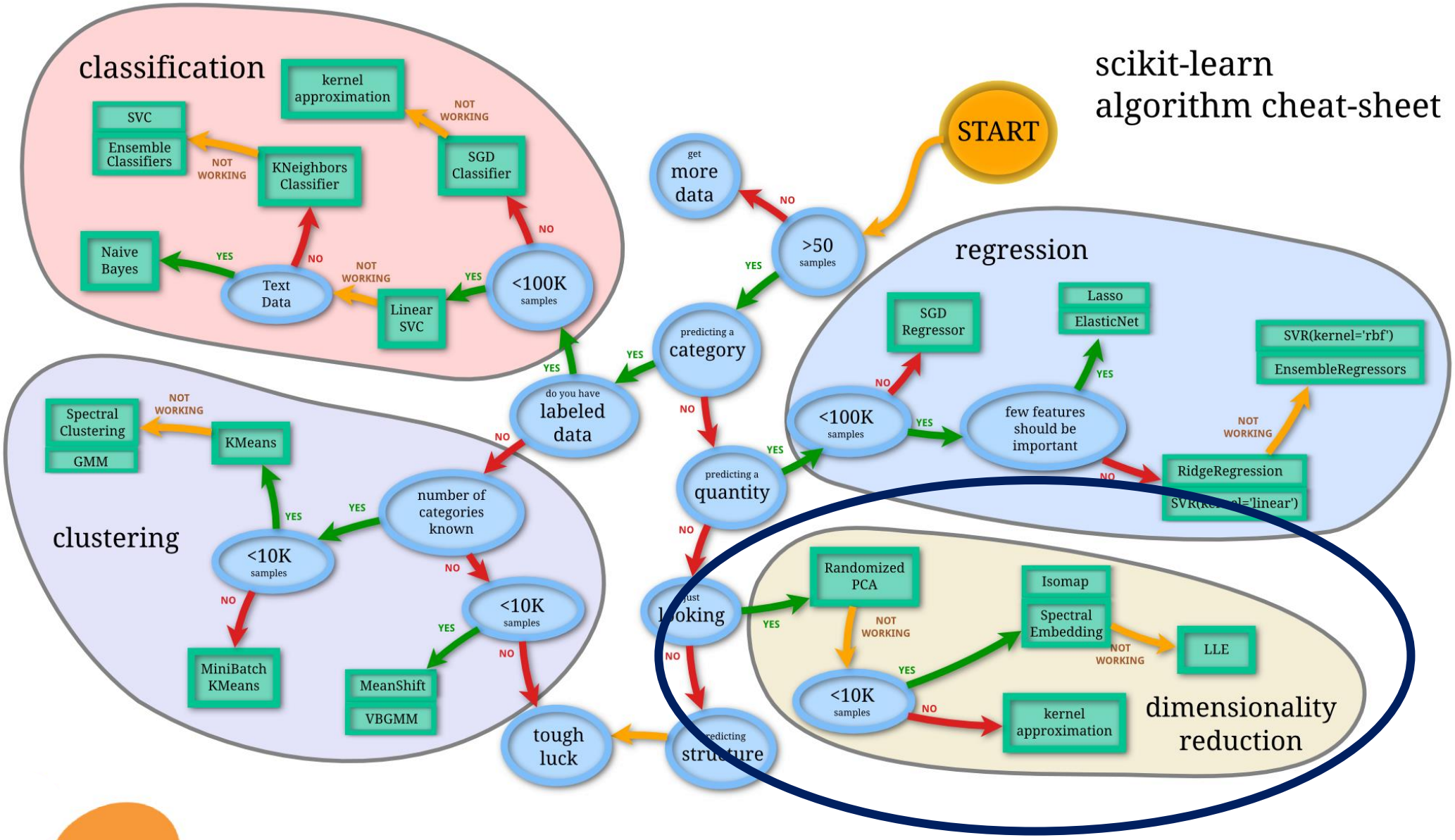

scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
NO
Text Data
NOT WORKING
YES
Linear SVC
<100K samples
get more data
NO
>50 samples
YES
predicting a category
YES
do you have labeled data
YES
NO
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
NO
RidgeRegression
SVR(kernel='linear')
NO
predicting a quantity
YES
NO
Spectral Clustering
GMM
NOT WORKING
KMeans
clustering
YES
YES
number of categories known
<10K samples
NO
NO
<10K samples
YES
NO
MiniBatch KMeans
MeanShift
VBGMM
tough luck
just looking
NO
predicting structure
Randomized PCA
YES
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
NO
Text Data
NOT WORKING
YES
Linear SVC
<100K samples
get more data
NO
>50 samples
YES
predicting a category
YES
do you have labeled data
YES
NO
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
RidgeRegression
NO
SVR(kernel='linear')
predicting a quantity
YES
NO
clustering
Spectral Clustering
GMM
NOT WORKING
KMeans
NO
number of categories known
YES
YES
<10K samples
NO
MiniBatch KMeans
<10K samples
YES
NO
MeanShift
VBGMM
tough luck
just looking
NO
predicting structure
Randomized PCA
YES
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
<10K samples
YES
NO
kernel approximation
dimensionality reduction

Dimensionality Reduction

Feature Selection ⟷ Feature Extraction

Dimensionality Reduction

	gene0	gene1	gene2	gene3	gene4	label
sample0	-0.325058	0.257257	-3.167128	-0.994030	2.259699	0
sample1	-1.813613	1.364216	-1.376160	-2.160071	2.232989	0
sample2	-1.228174	-1.049257	0.067691	-1.415496	0.920494	0
sample3	-1.627611	-1.805168	1.959821	-2.715868	0.372130	1
sample4	-1.659876	-0.974302	1.474967	-1.670365	0.375689	1
...	...	...	...	...	...	...
sample95	-1.748034	0.115954	1.640853	-2.032271	0.421240	1
sample96	-1.914813	-0.150770	-1.903919	-1.106743	2.290877	0
sample97	-1.774981	-0.353924	1.931081	-2.177989	0.306350	1
sample98	-0.310395	-0.815936	-0.555645	-2.155341	1.065890	0
sample99	-1.441635	0.140666	1.295502	-3.213609	0.810031	1

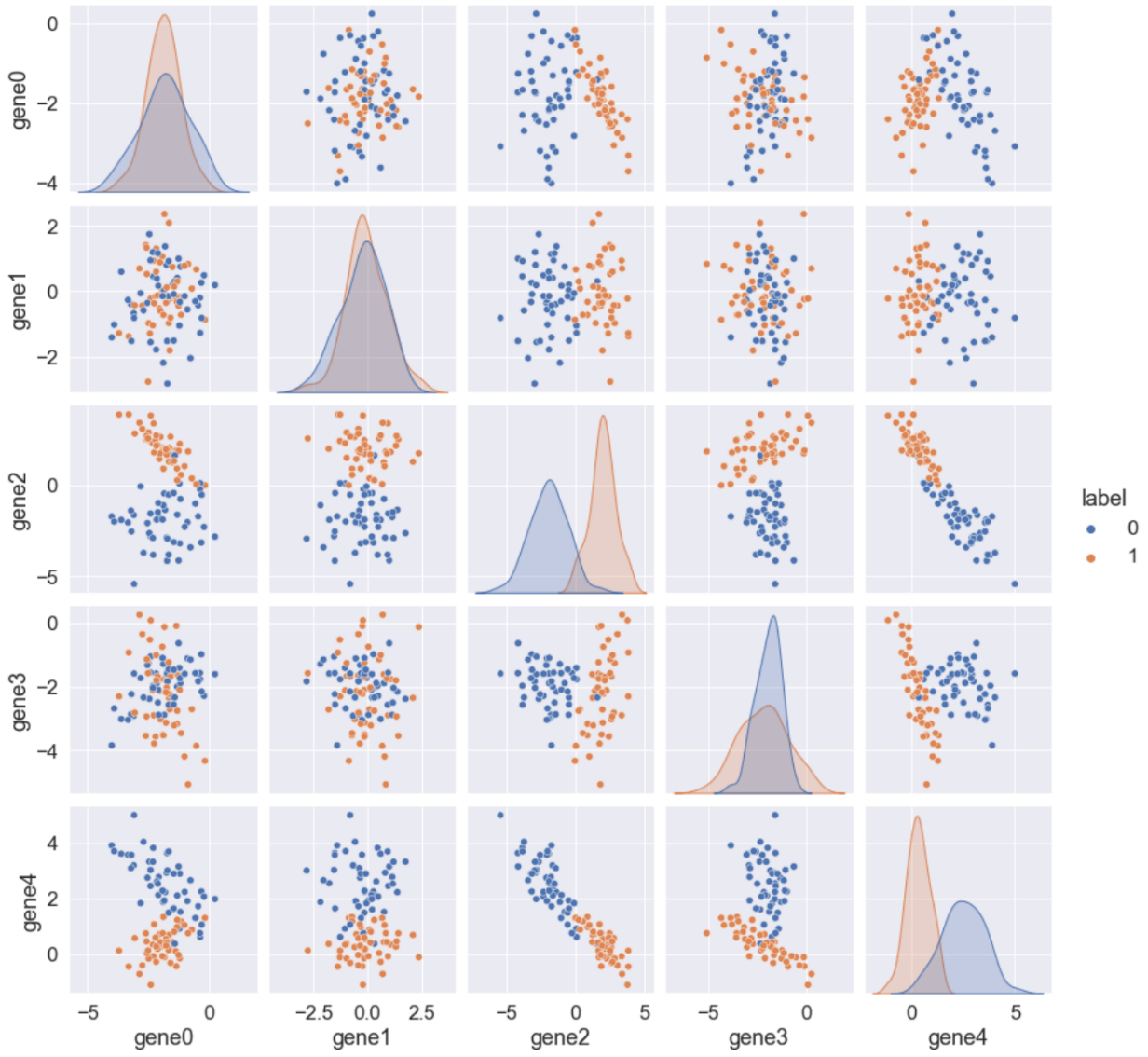
100 rows × 6 columns

100 samples: patients

5 dimensions: gene expression levels → features

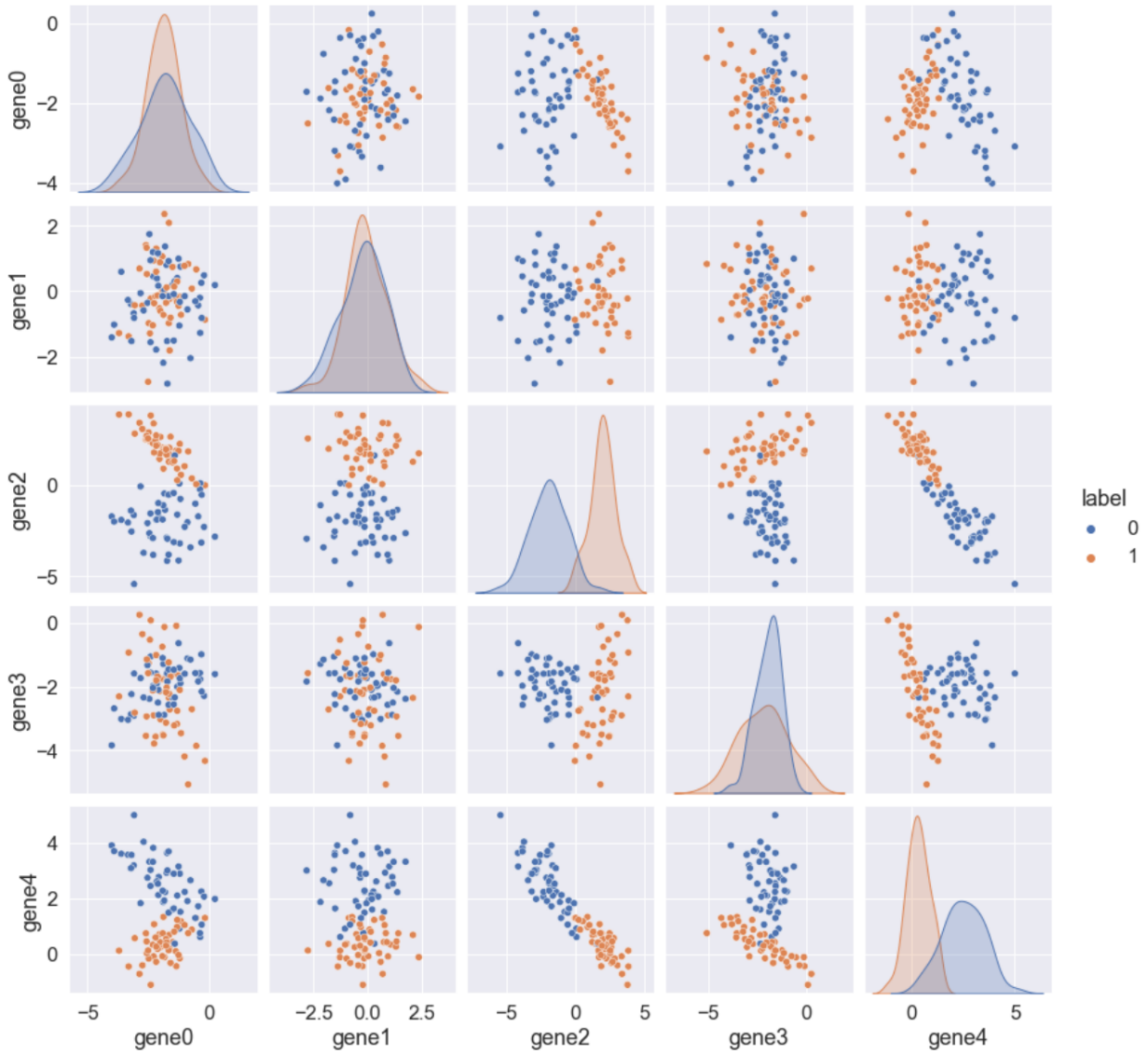
2 classes: 0 or 1 (‘label‘ column)

Dimensionality Reduction: Feature Selection



What are the two most informative genes to classify the samples into the two classes?

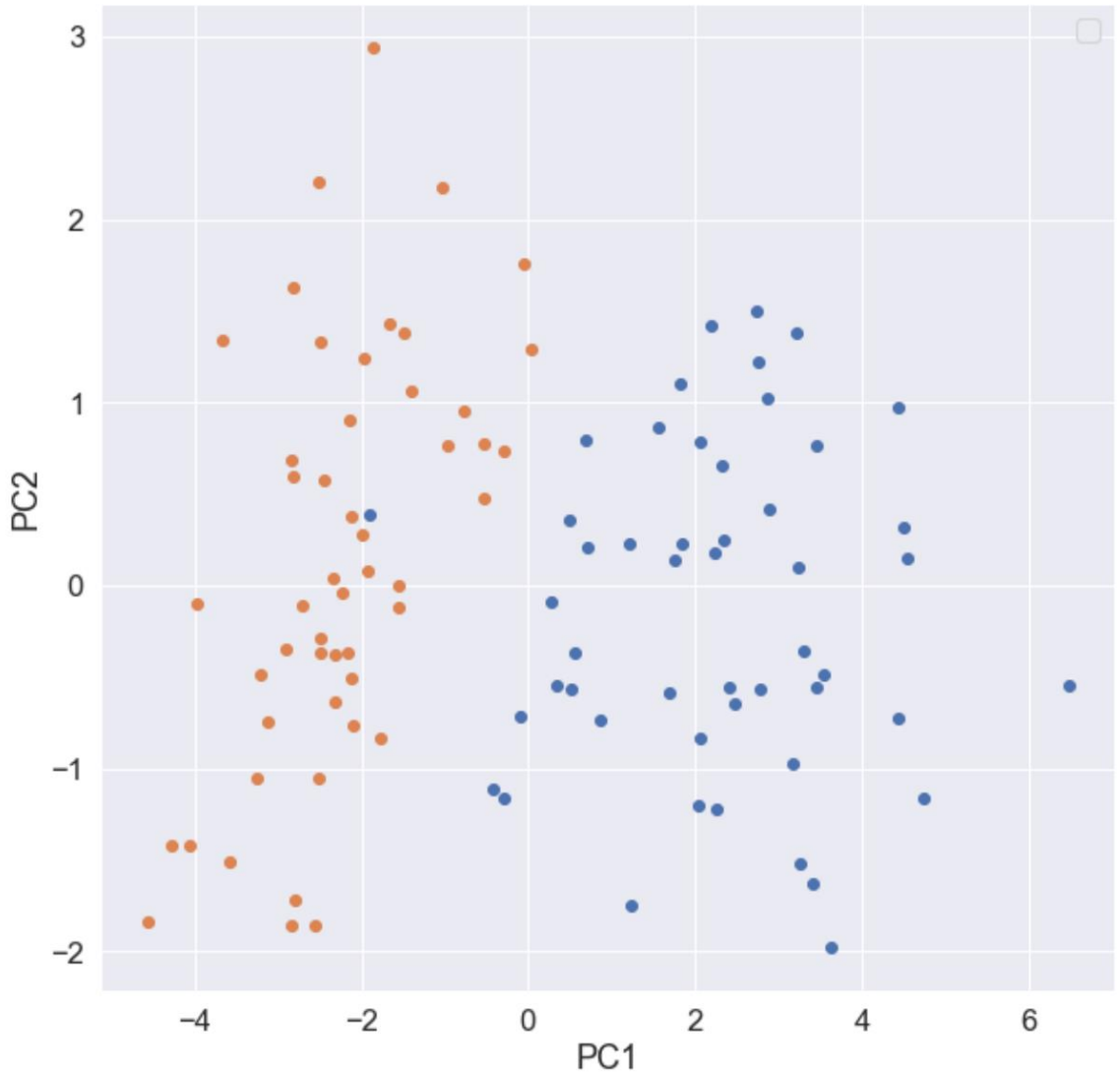
Dimensionality Reduction: Feature Selection



What are the two most informative genes to classify the samples into the two classes?

Becomes complicated when gene interactions can be informative as well.

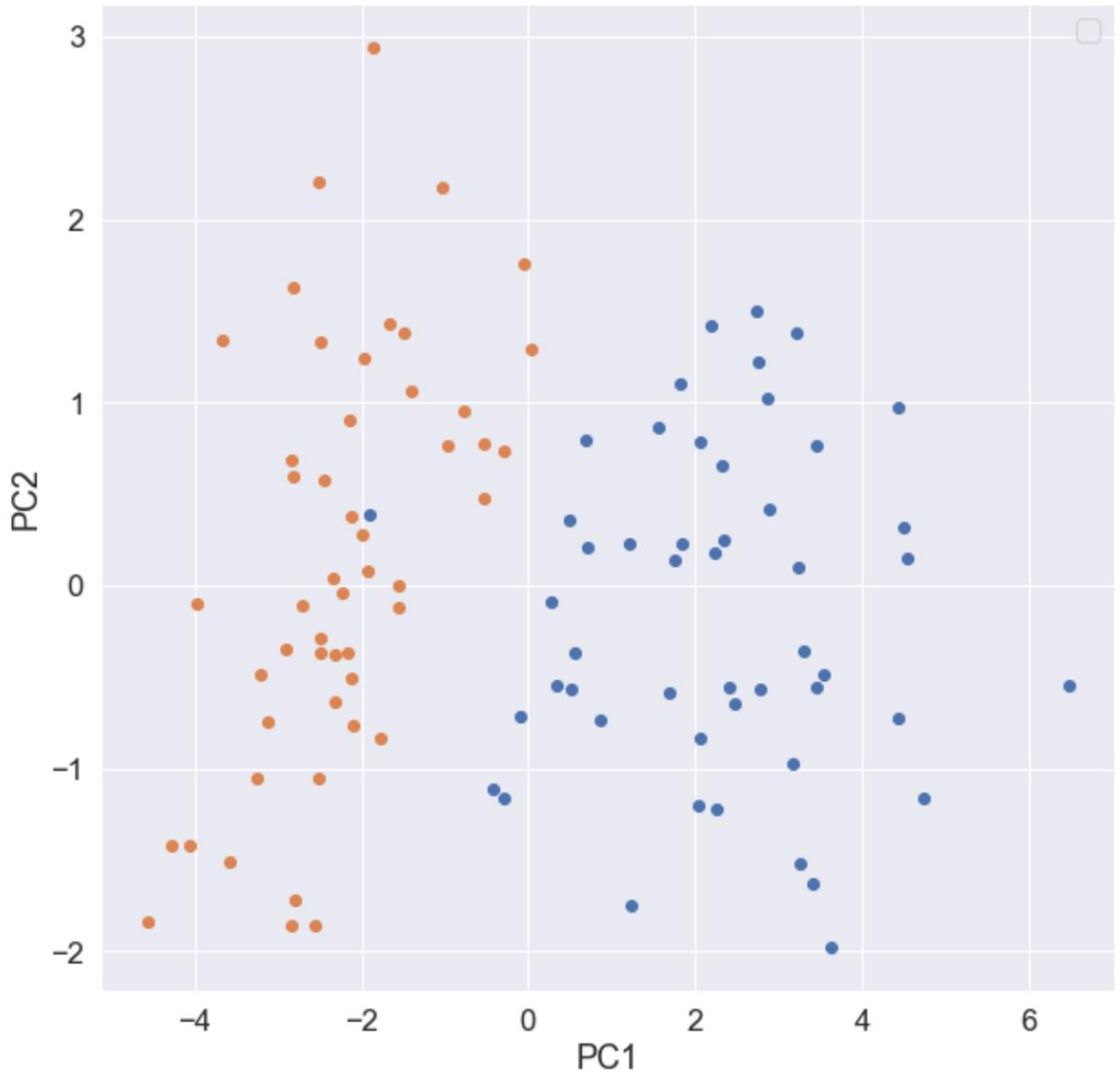
Dimensionality Reduction: Feature Extraction



Computes derived features intended to be informative and non-redundant.

E.g. Principal Component Analysis, and Multi-dimensional Scaling.

Dimensionality Reduction: Feature Extraction

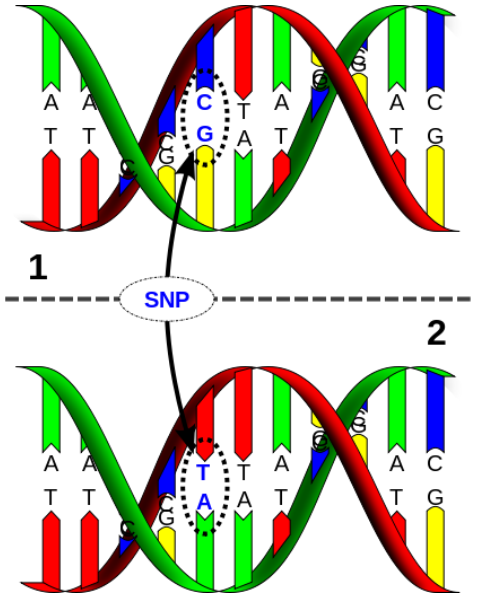


Principal Component Analysis (PCA) reduced 5 dimensions to 2 dimensions: the first two principal components.

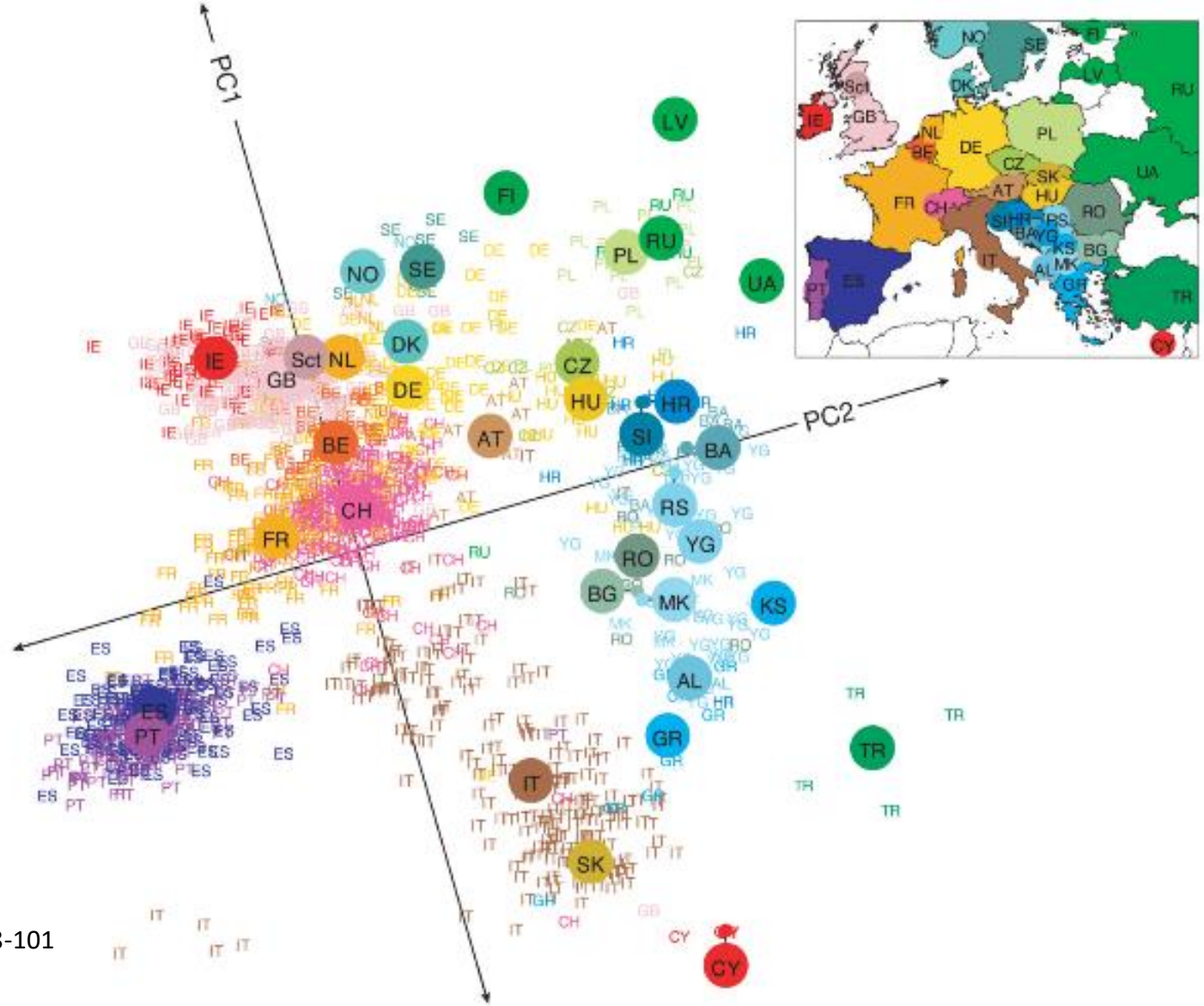
PCA is a linear dimensionality reduction technique.

PCA tries to preserve the global structure of the data.

Dimensionality Reduction: Feature Extraction

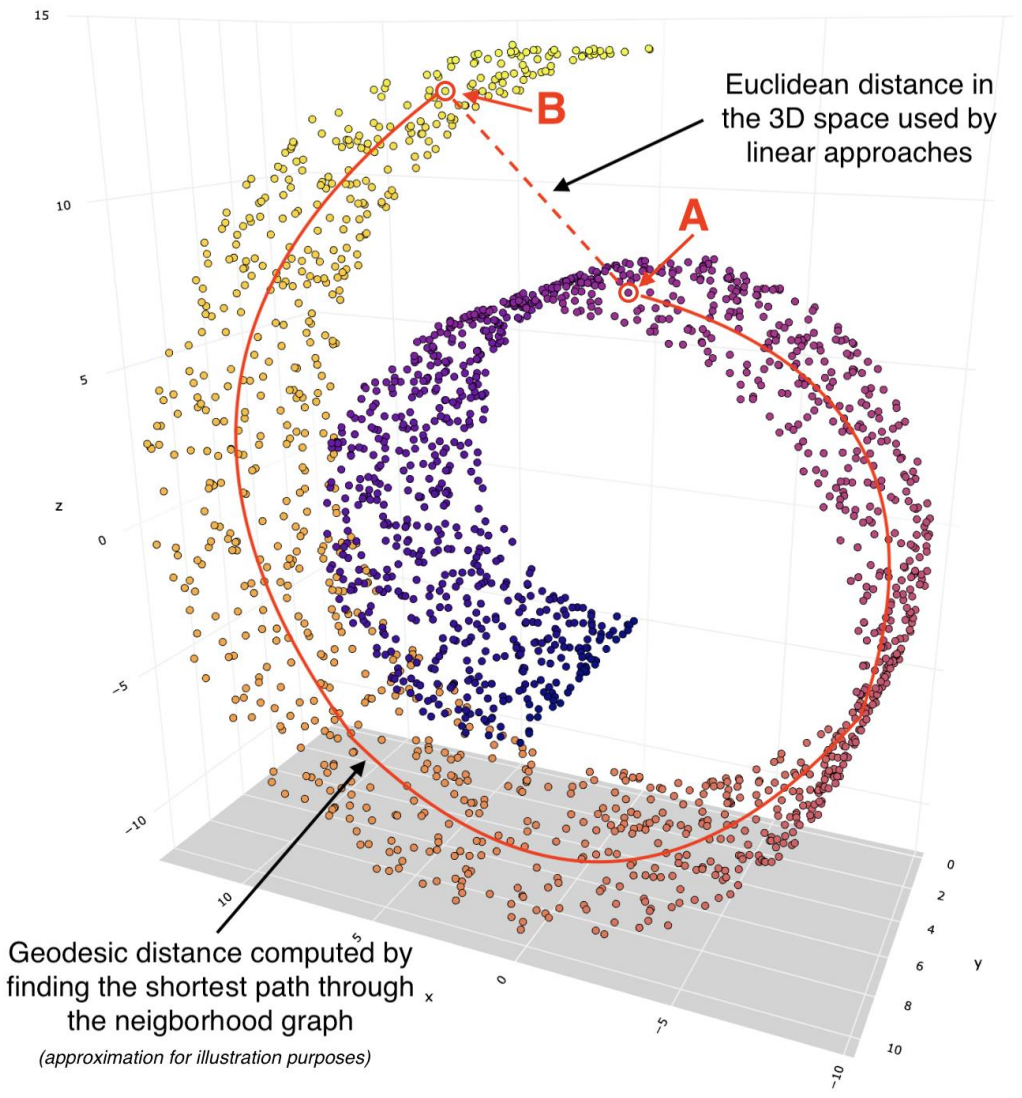


"The data set contains 3,192 individuals who were genotyped at 500,568 SNPs using the Affymetrix 500K SNP chip."

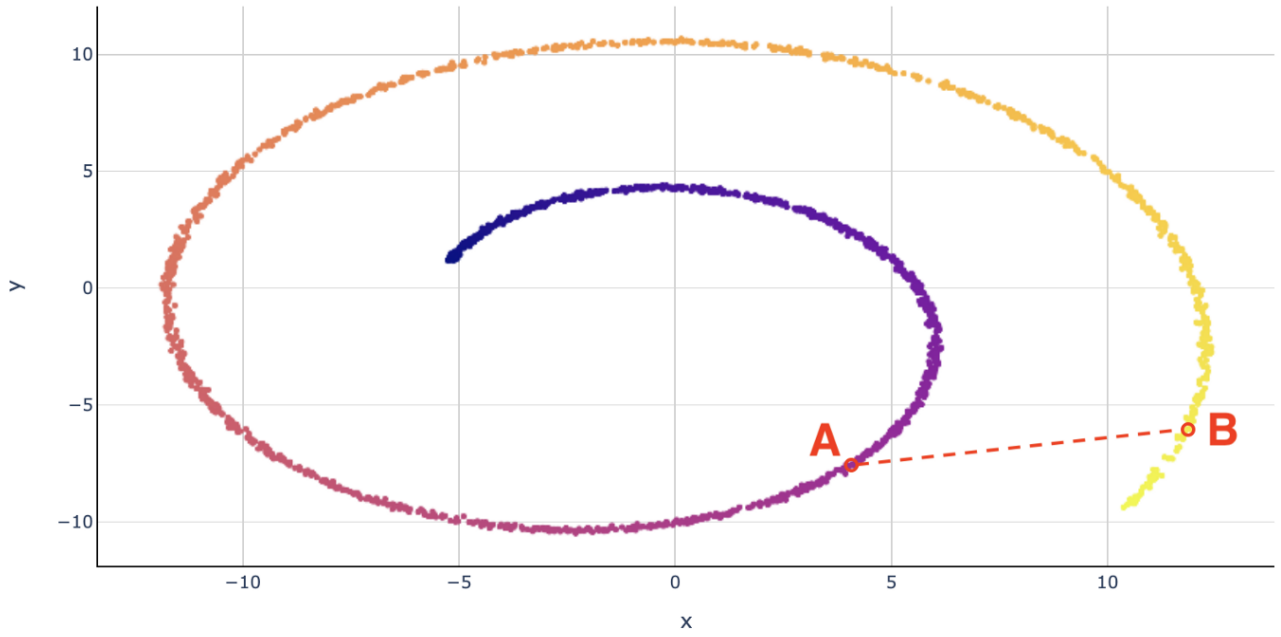


Novembre *et al.* (2008) "Genes mirror geography within Europe" Nature 456, 98-101

Manifolds

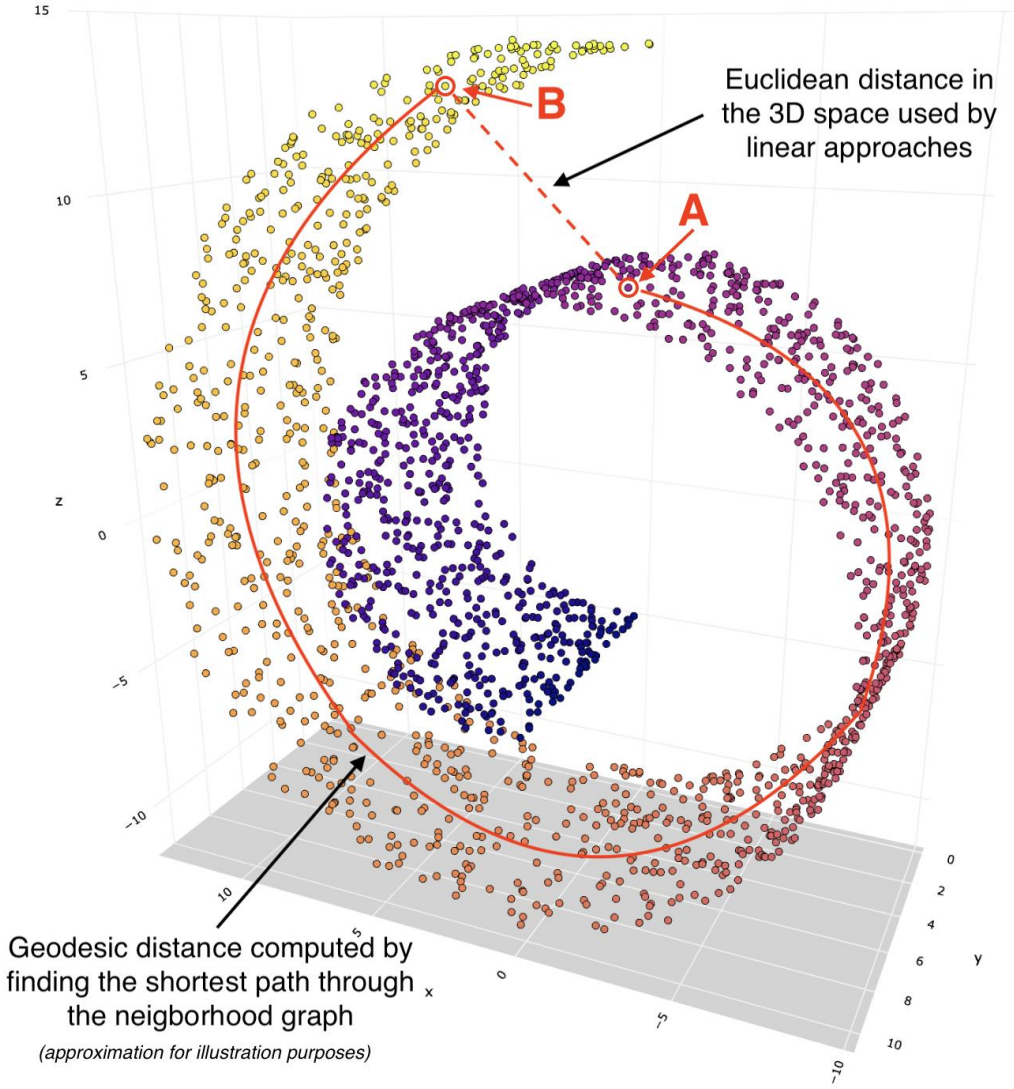


PCA

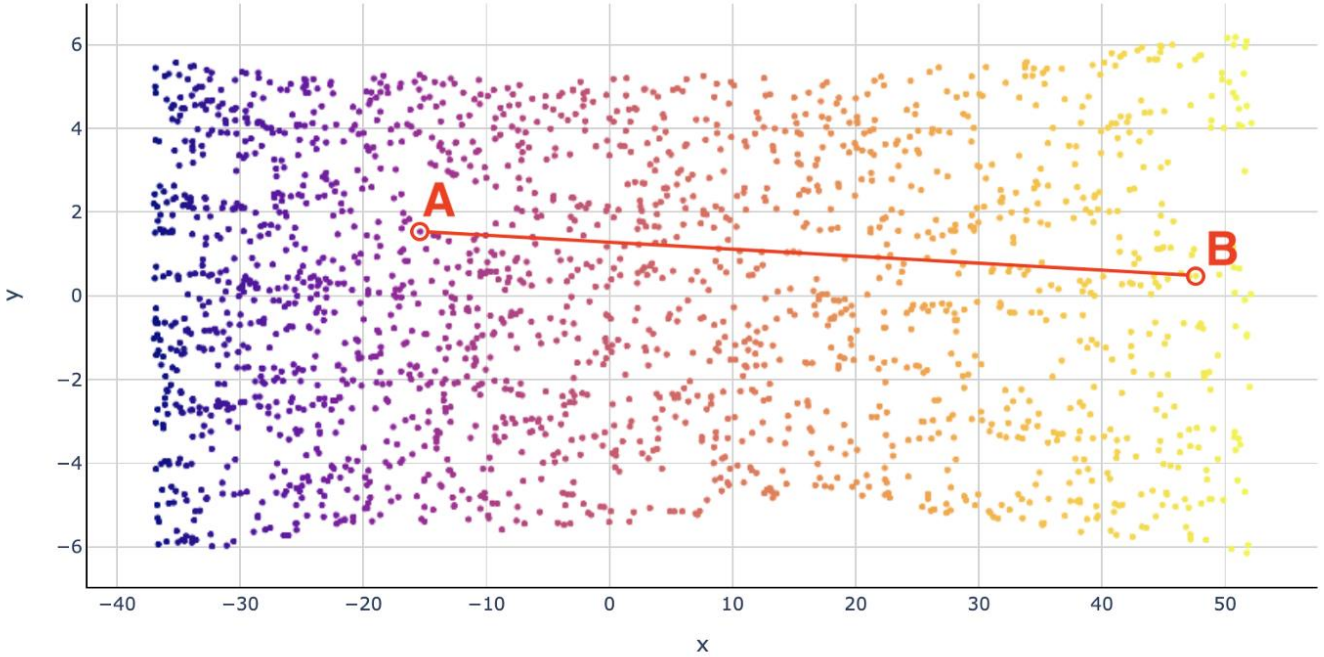


source: solclover.com

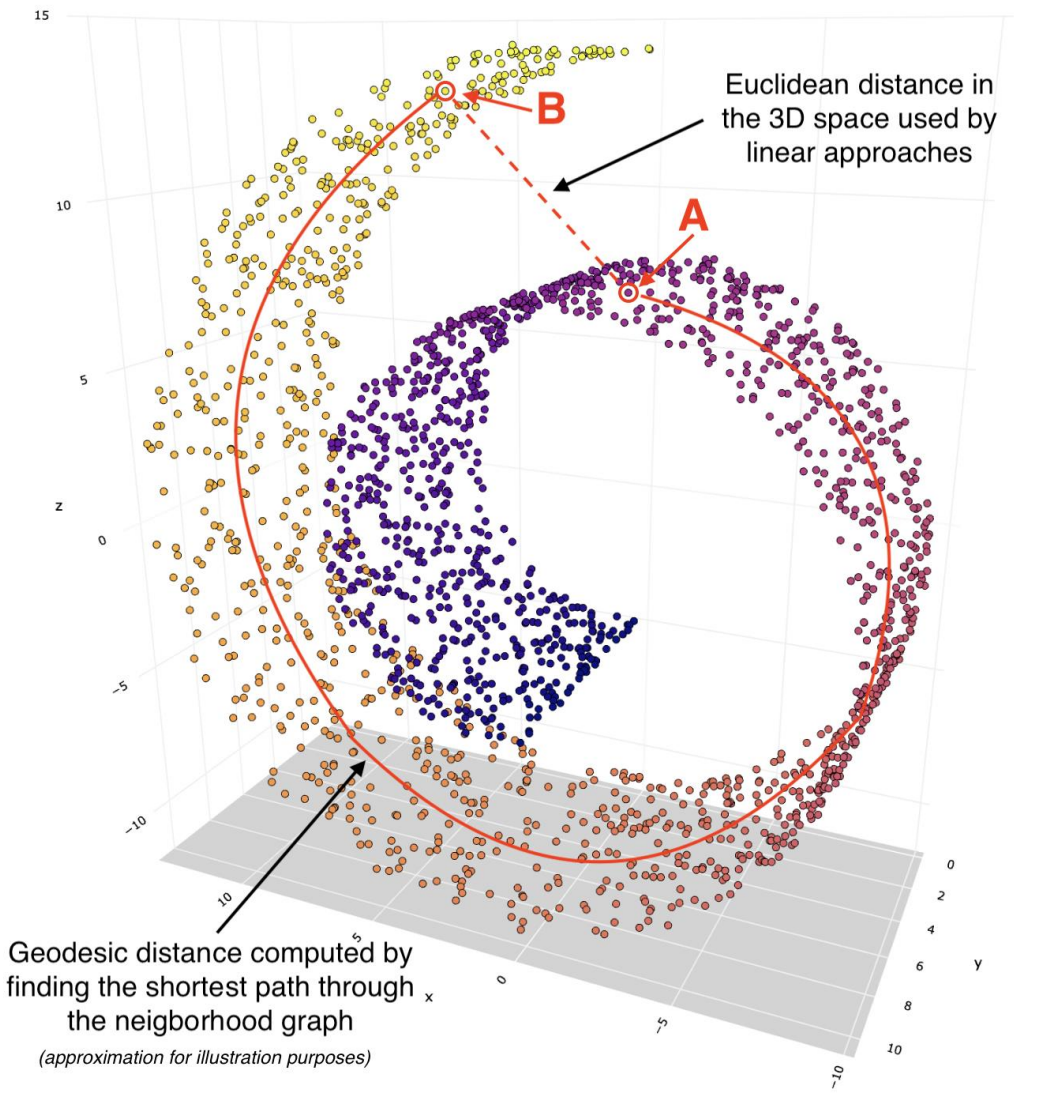
Non-linear Manifolds



t-distributed stochastic neighbor embedding (t-SNE)



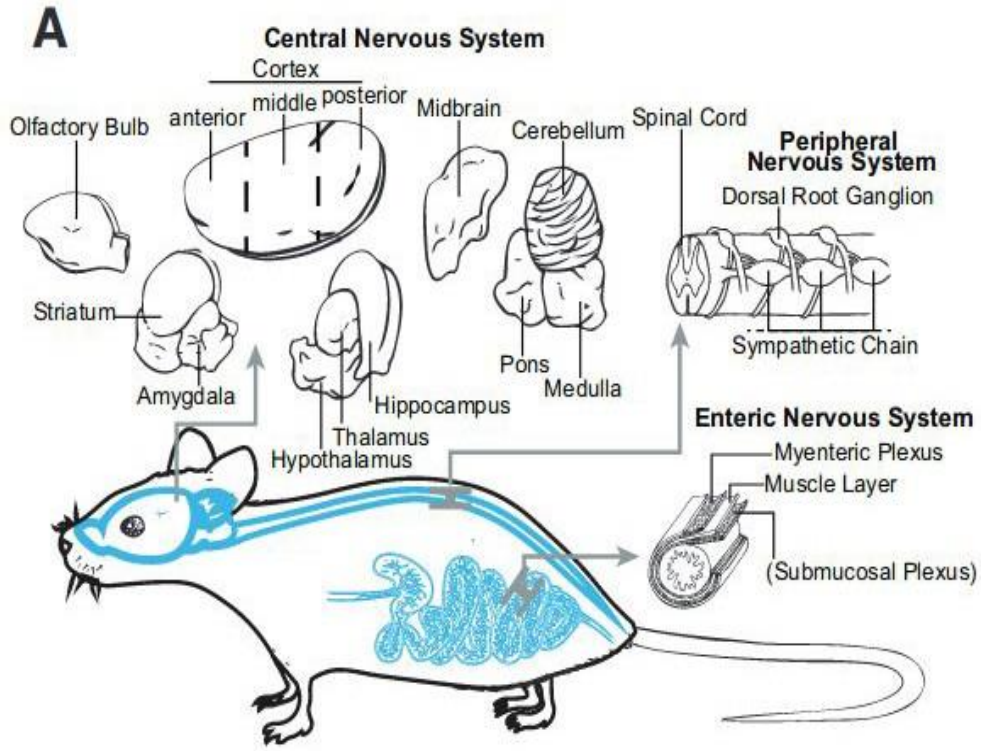
Non-linear Manifolds

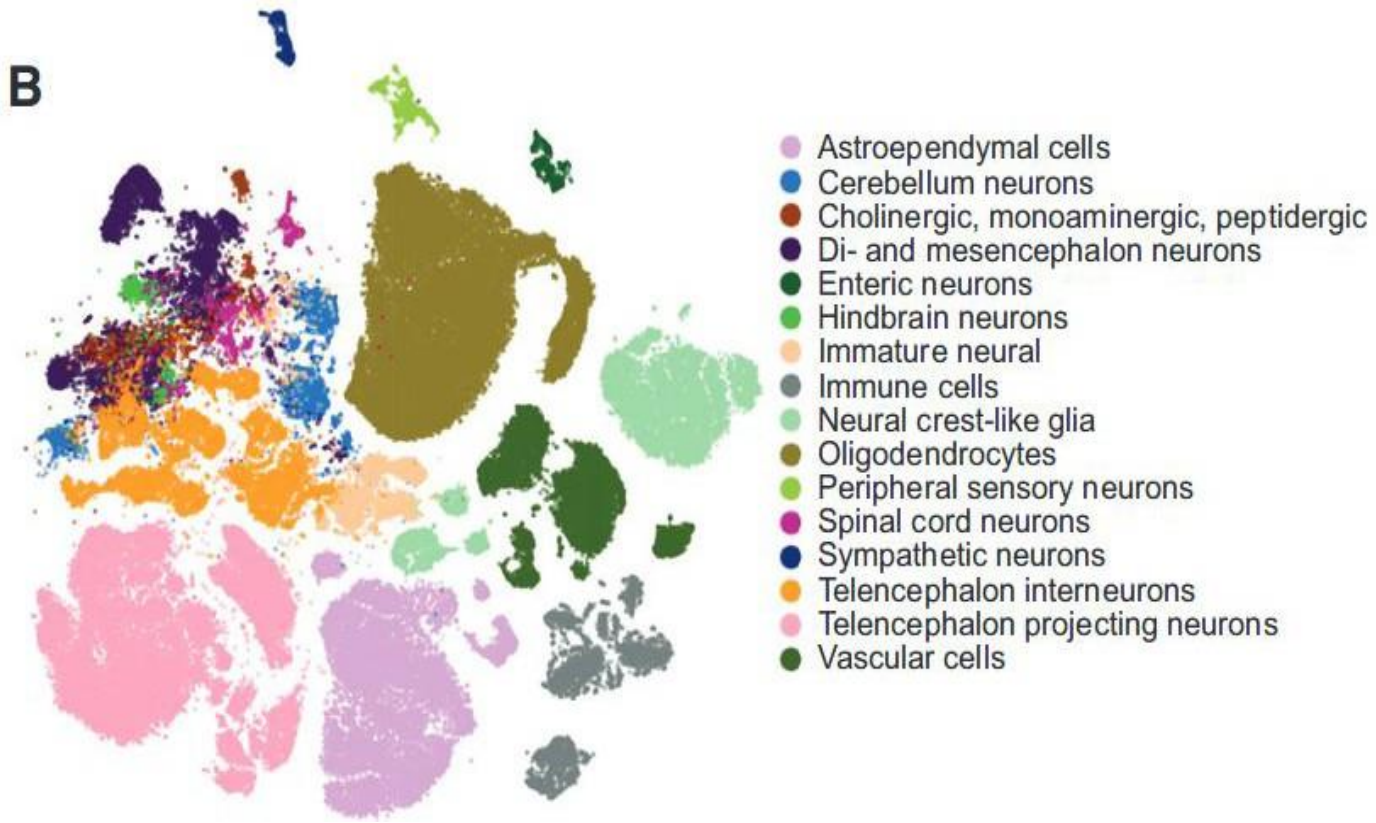


PCA tries to preserve the global structure of the data.

For manifolds we want to preserve the local structure of the data.

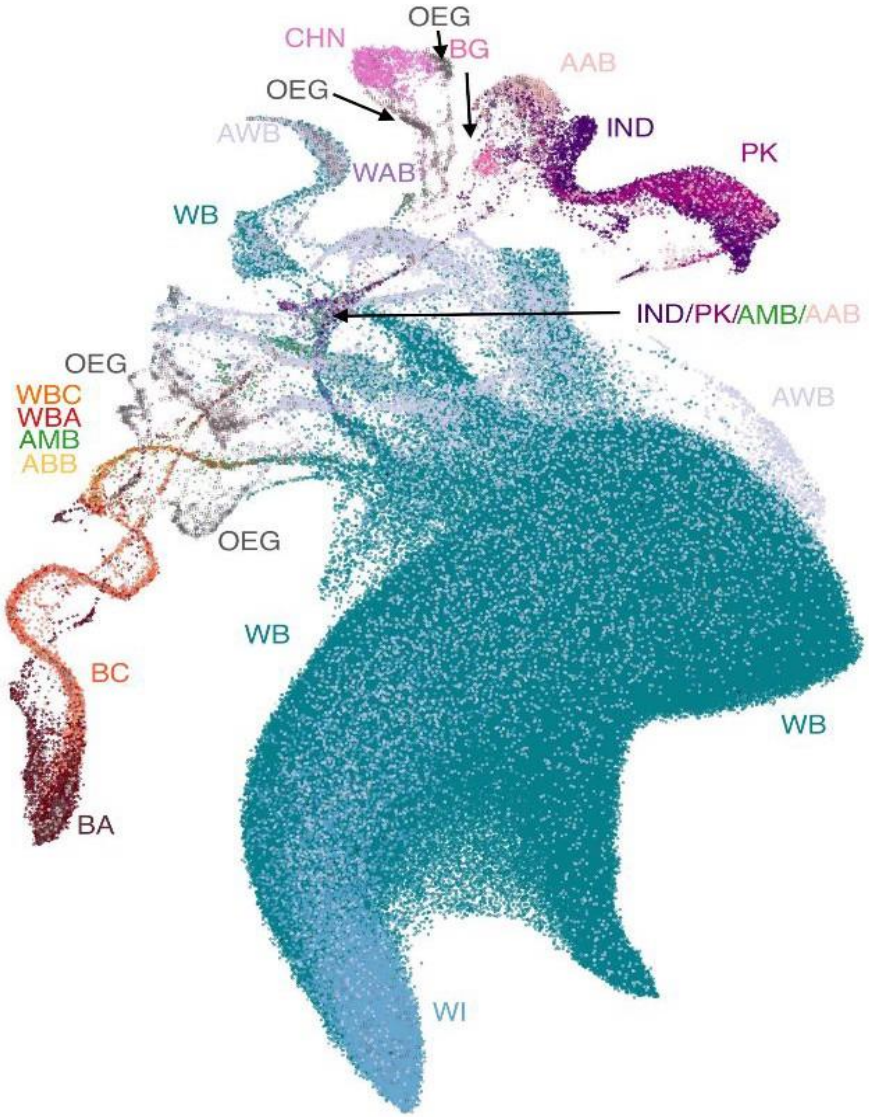
Non-linear Manifolds





Zeisel *et al.* (2018)

Non-linear Manifolds

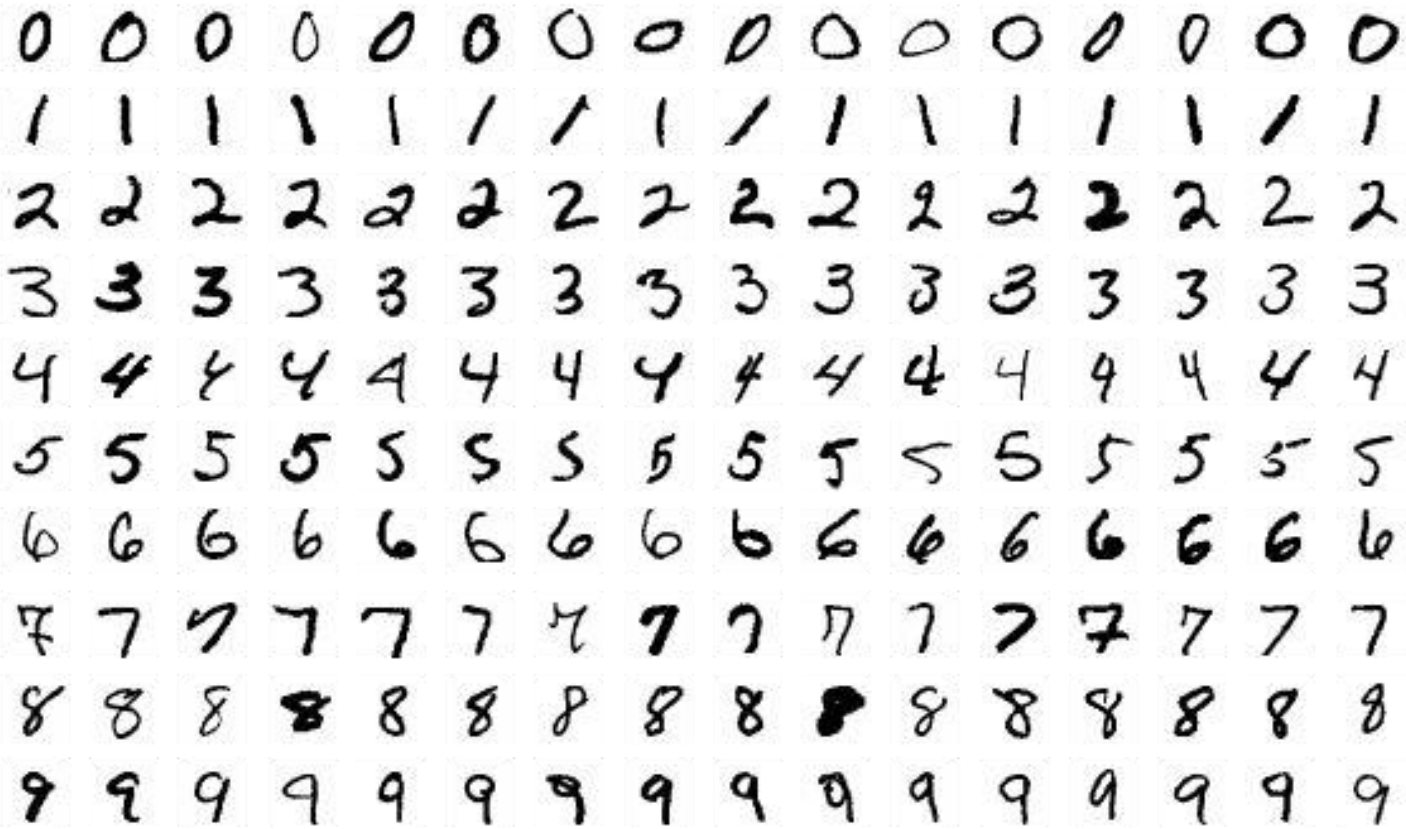


The UK BioBank provides genotype data on 488,377 individuals along with self-identified ethnic backgrounds.

The dataset is majority White (88.3% British, 2.6% Irish, 3.4% other), with large populations identifying as Black (1.6% either African, Caribbean, or other), Asian (1.9% either Indian, Pakistani, Bangladeshi, or other), Chinese (0.3%), another ethnic group (0.8%), mixed ethnicity (0.6%), or an unavailable response (0.5%)

Diaz-Papkovich *et al*. (2019)

MNIST (Modified National Institute of Standards and Technology database)



n = 70,000

28×28 images = 784 pixels

Yann LeCun *et al*. (1998)

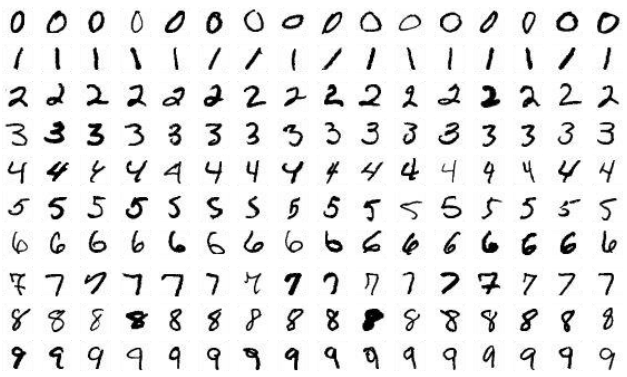
MNIST (Modified National Institute of Standards and Technology database)

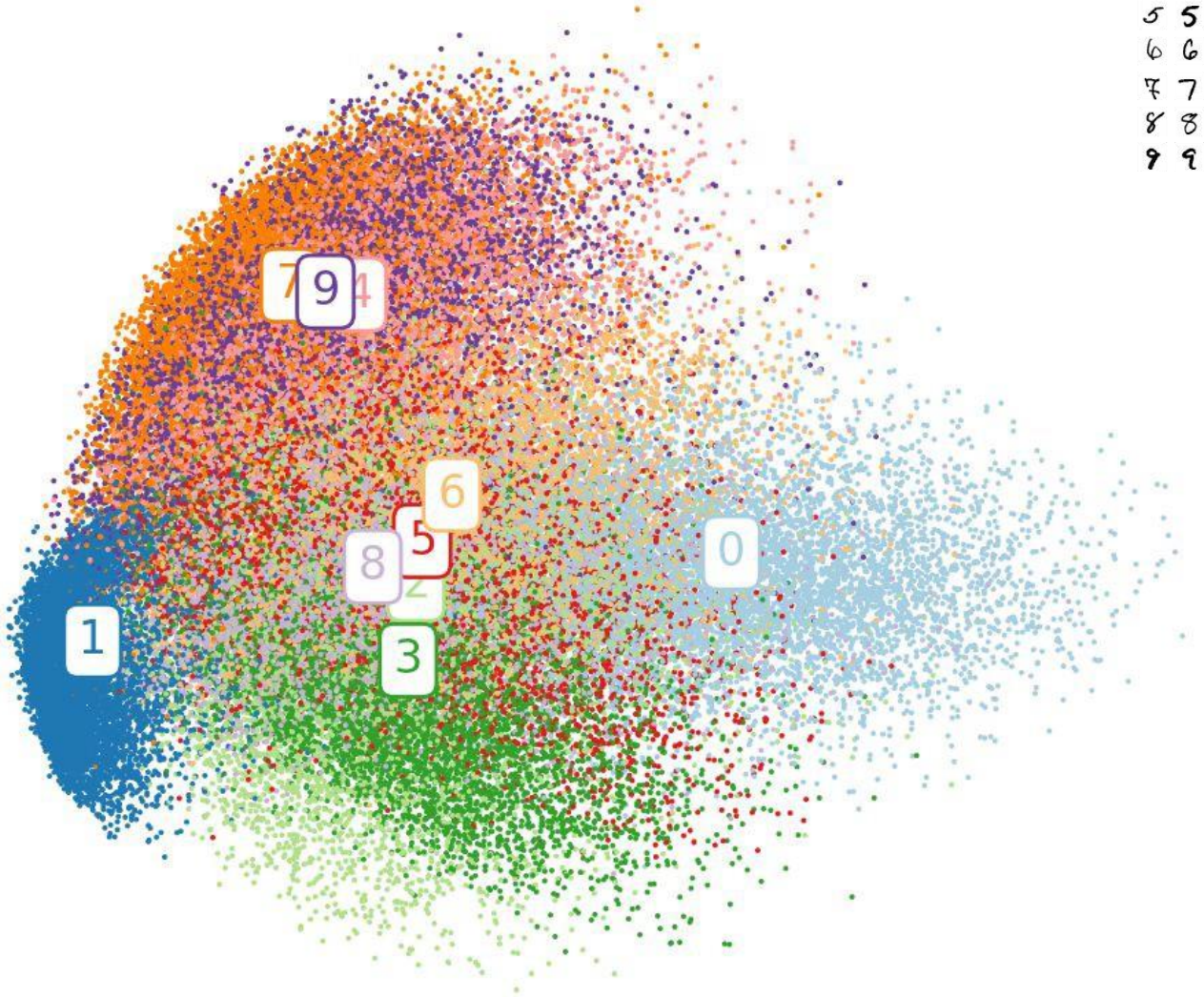
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	13	18	18	18	134	150	255	169	37	0	0	0	0	0	0
7	0	0	0	0	0	5	92	36	36	140	154	154	154	224	253	253	253	253	253	253	253	170	0	0	0	0	0	0
8	0	0	0	0	0	36	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253	145	0	0	0	0	0	0
9	0	0	0	0	0	9	144	202	253	223	182	182	182	182	182	138	65	161	253	253	235	38	0	0	0	0	0	0
10	0	0	0	0	0	0	0	13	47	27	0	0	0	0	0	0	0	130	253	253	147	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	235	253	253	70	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	253	253	230	36	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54	243	253	251	131	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	176	253	253	224	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78	253	253	253	141	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	168	253	253	208	25	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	95	244	253	253	124	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	177	253	253	203	14	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	2	107	252	253	253	81	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	38	253	253	253	253	92	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	24	163	253	253	253	236	60	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	125	253	253	253	233	63	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	218	253	253	253	137	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	243	253	253	101	8	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	187	253	234	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

n = 70,000

28×28 images = 784 pixels

MNIST: PCA



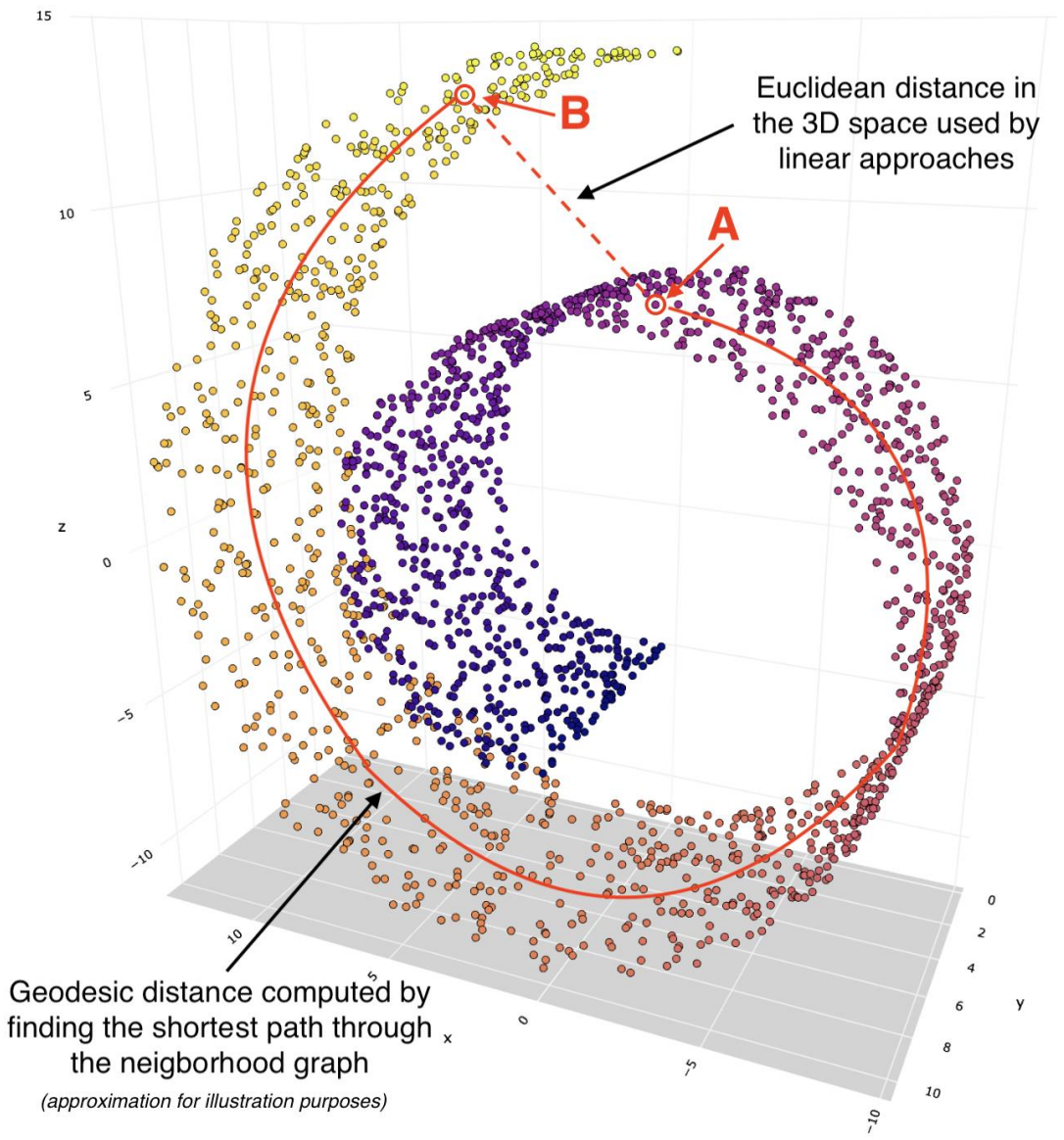


MNIST: Multi-dimensional scaling (MDS)

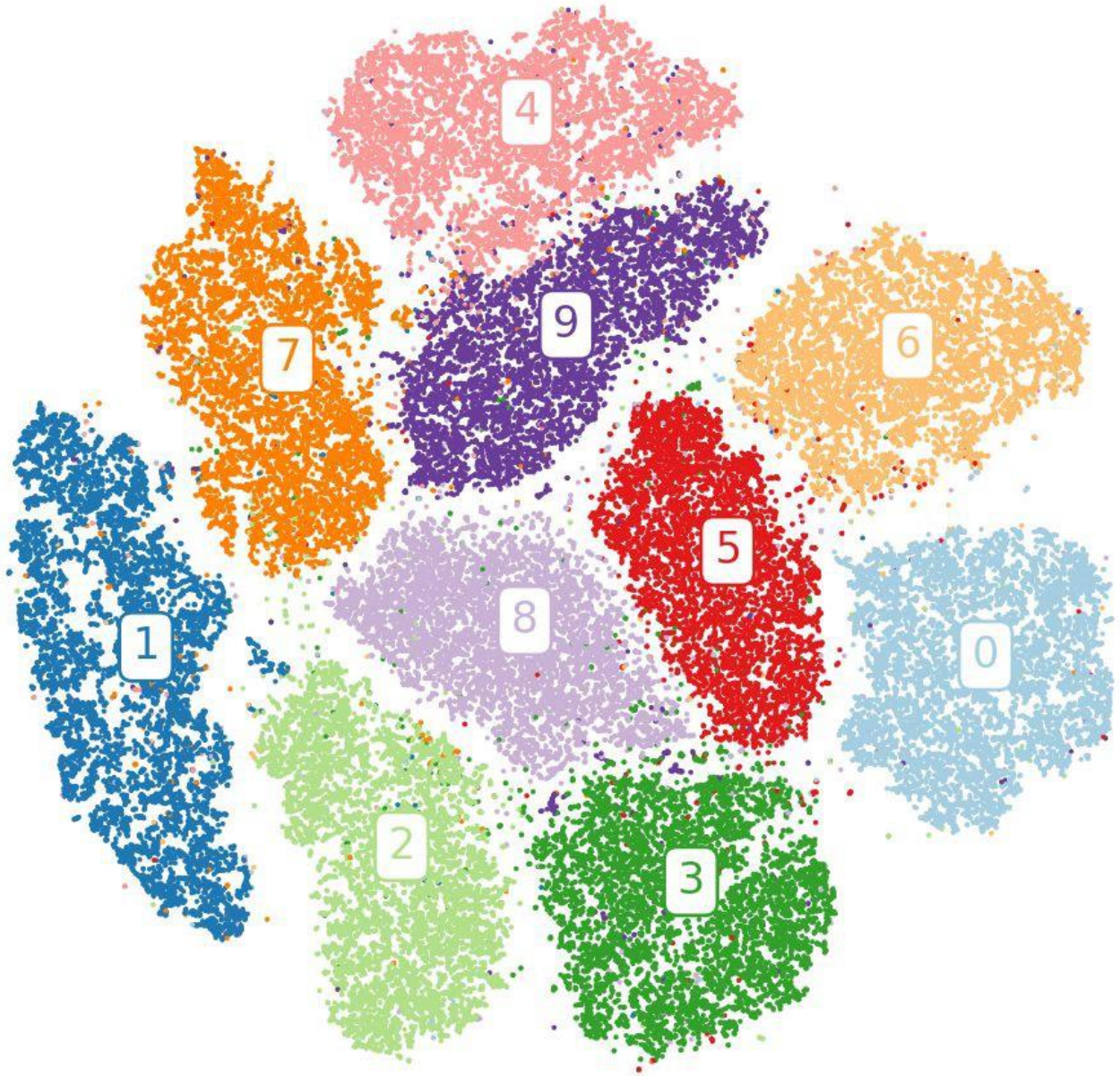
$$\mathcal{L} = \sum_{i<j} (d_{ij} - \|\mathbf{y}_i - \mathbf{y}_j\|)^2$$

t-SNE

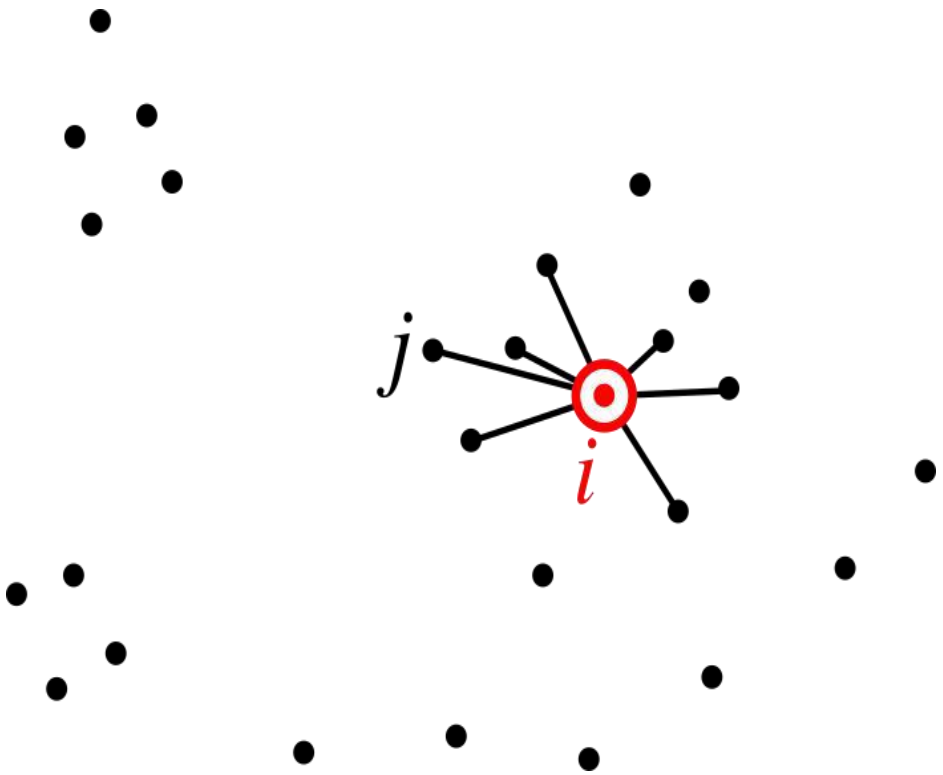
Preserve **nearest neighbours** instead of preserving distances.



MNIST: t-SNE



t-SNE

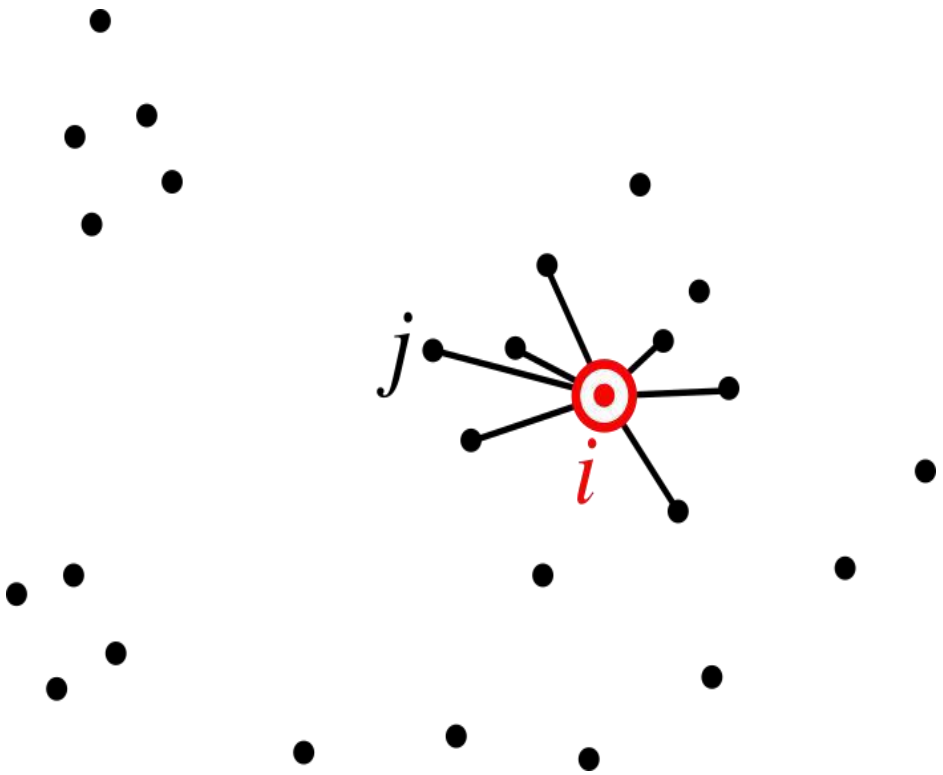


Loss function: **Kullback-Leibler divergence** between pairwise similarities (affinities) in the high-dimensional and in the low-dimensional spaces.

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

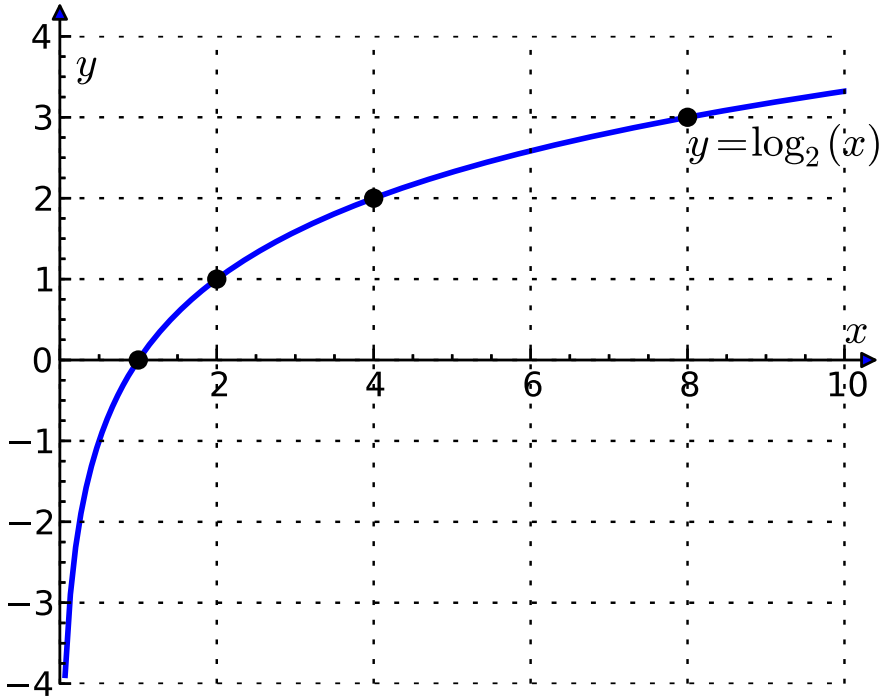
So, we want to find the q_{ij} that minimize $\mathcal{L}$.

t-SNE

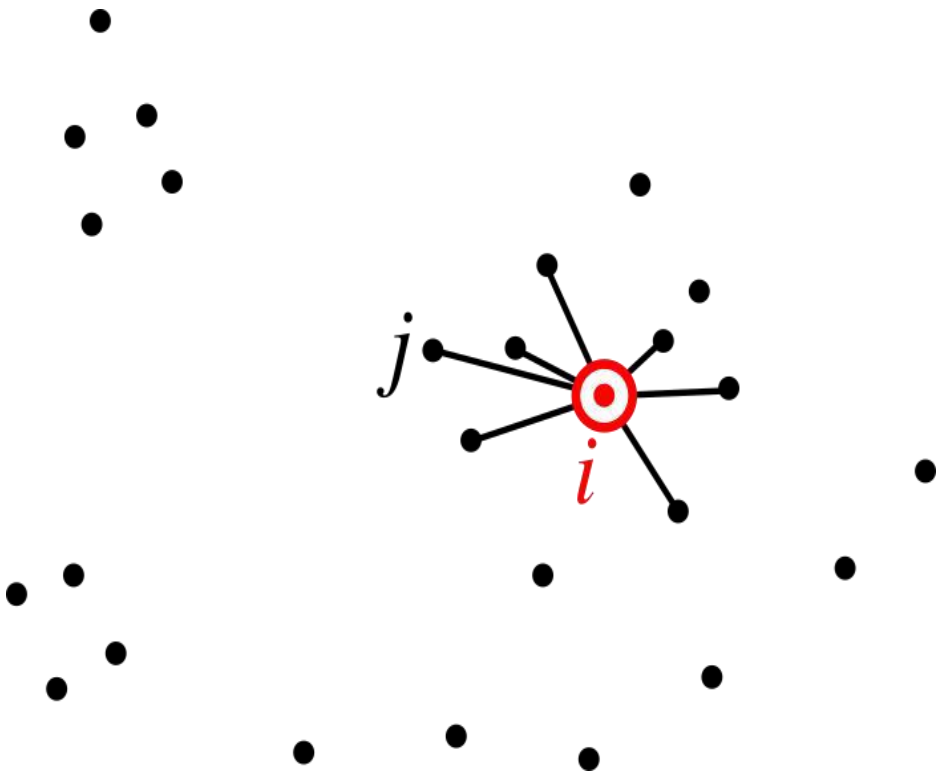


So, we want to find the q_{ij} that minimize $\mathcal{L}$.

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$



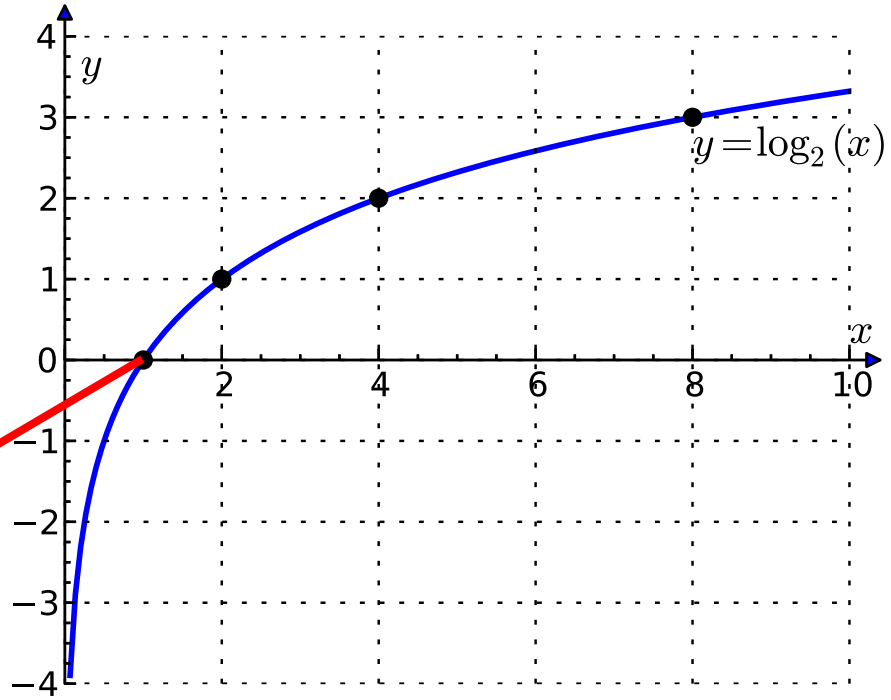
t-SNE



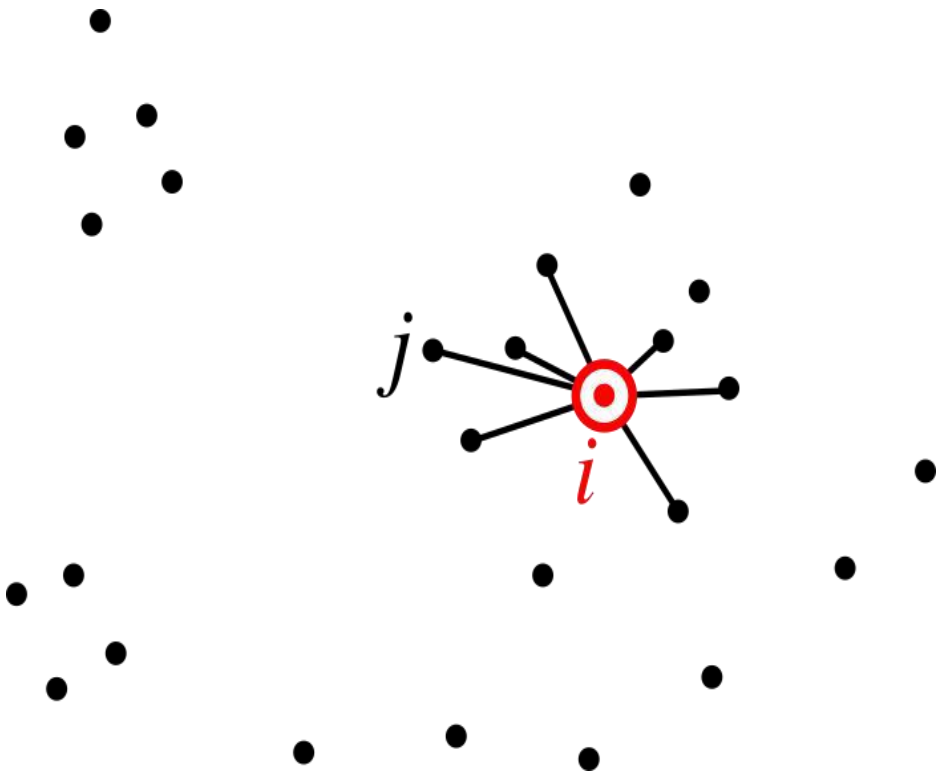
So, we want to find the q_{ij} that minimize $\mathcal{L}$.

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$$p_{ij} = q_{ij}$$

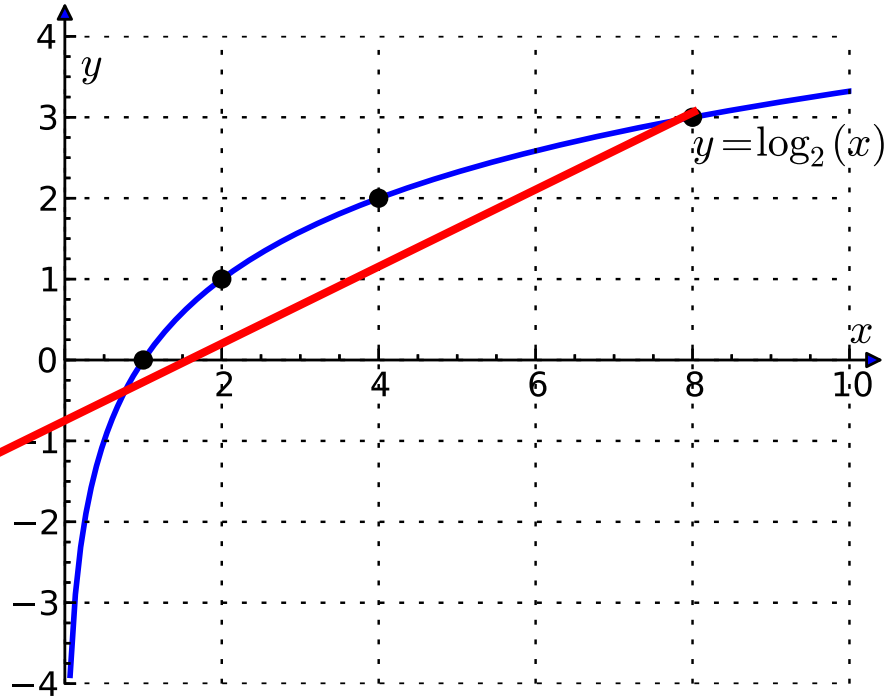


t-SNE



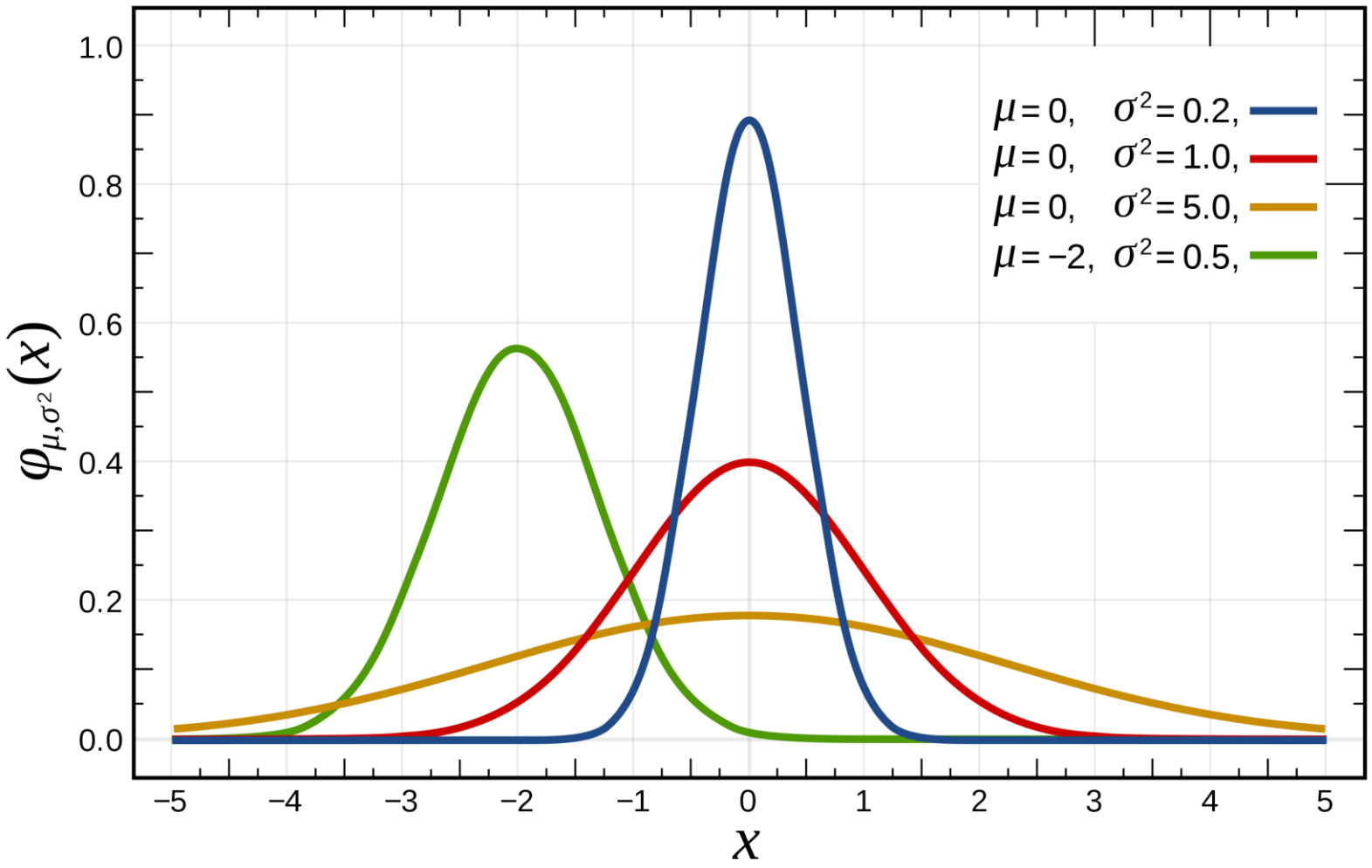
So, we want to find the q_{ij} that minimize $\mathcal{L}$.

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$



p_{ij} >> q_{ij}

Intermezzo: Gaussian Distribution



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

t-SNE

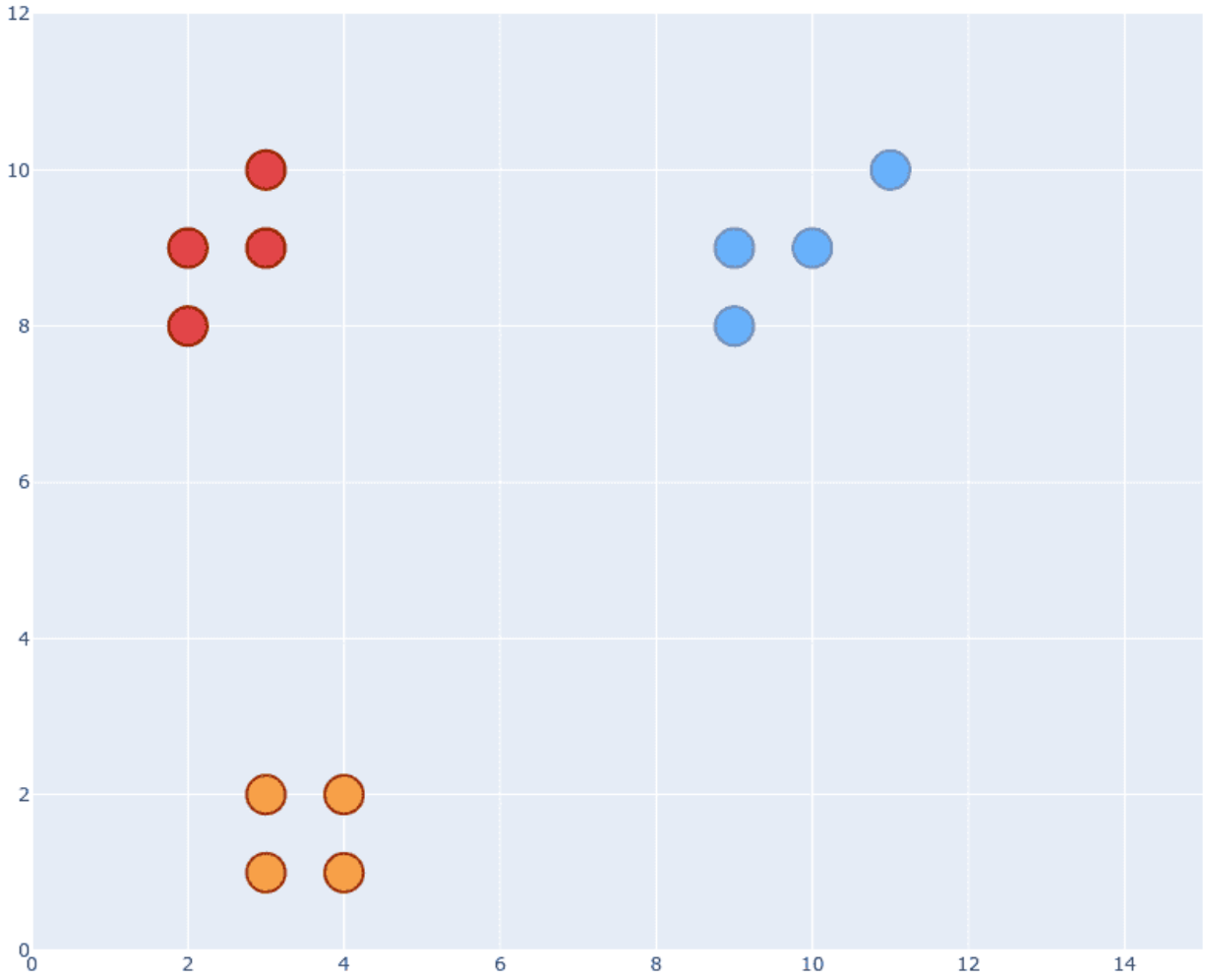
$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

t-SNE

Create a **probability distribution** that represents similarities between neighbours.

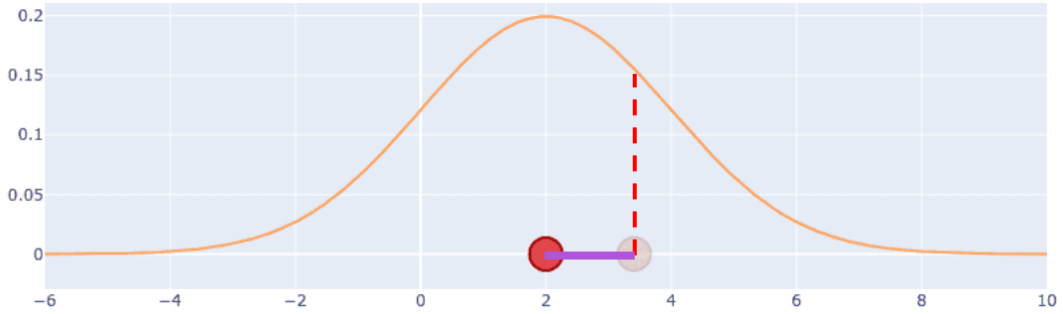
"similarity of datapoint $\mathbf{x}_j$
to datapoint $\mathbf{x}_i$
is the conditional probability $p_{j|i}$
that $\mathbf{x}_i$ would pick $\mathbf{x}_j$ as its neighbour"

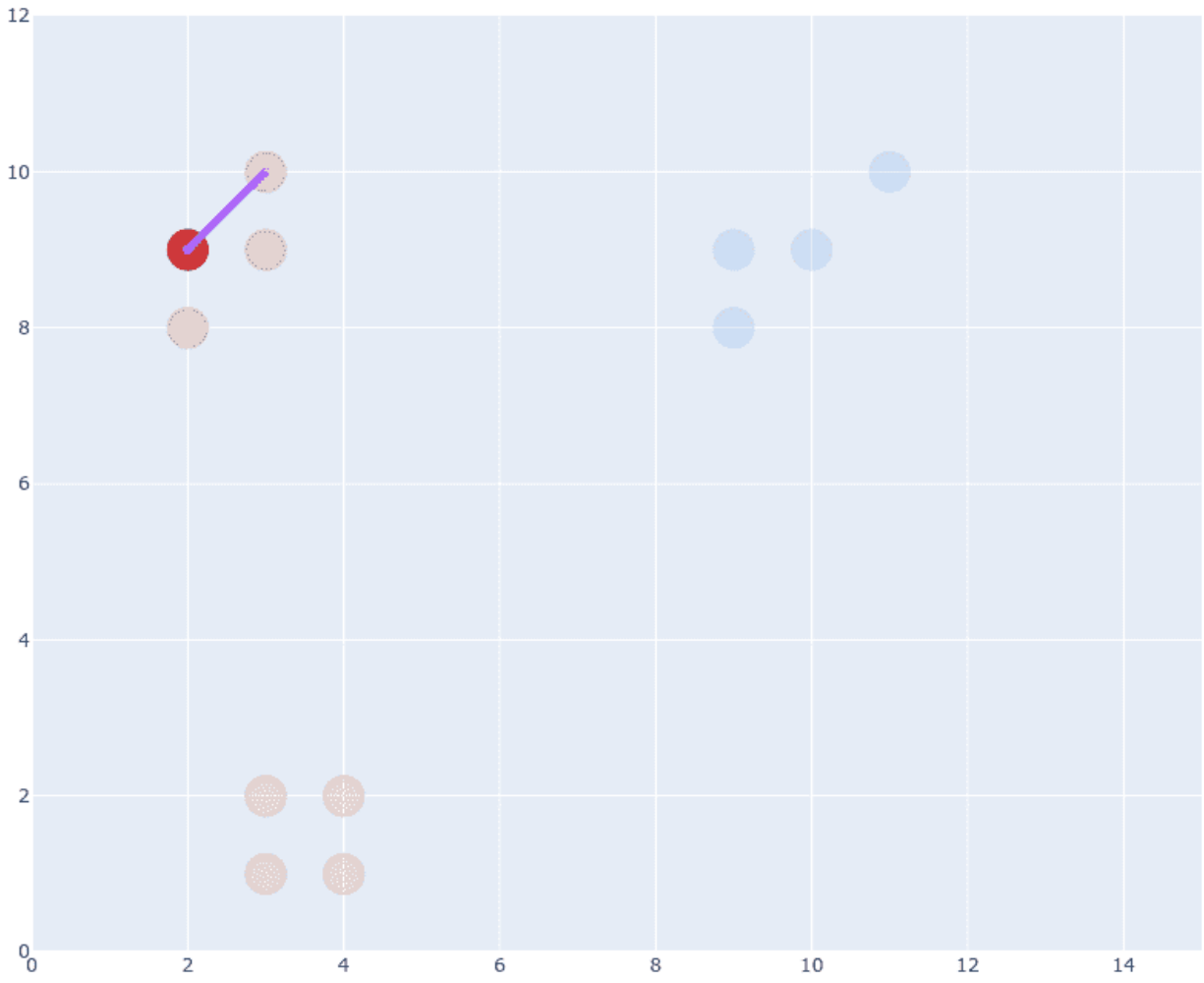


t-SNE

Create a **probability distribution** that represents similarities between neighbours.

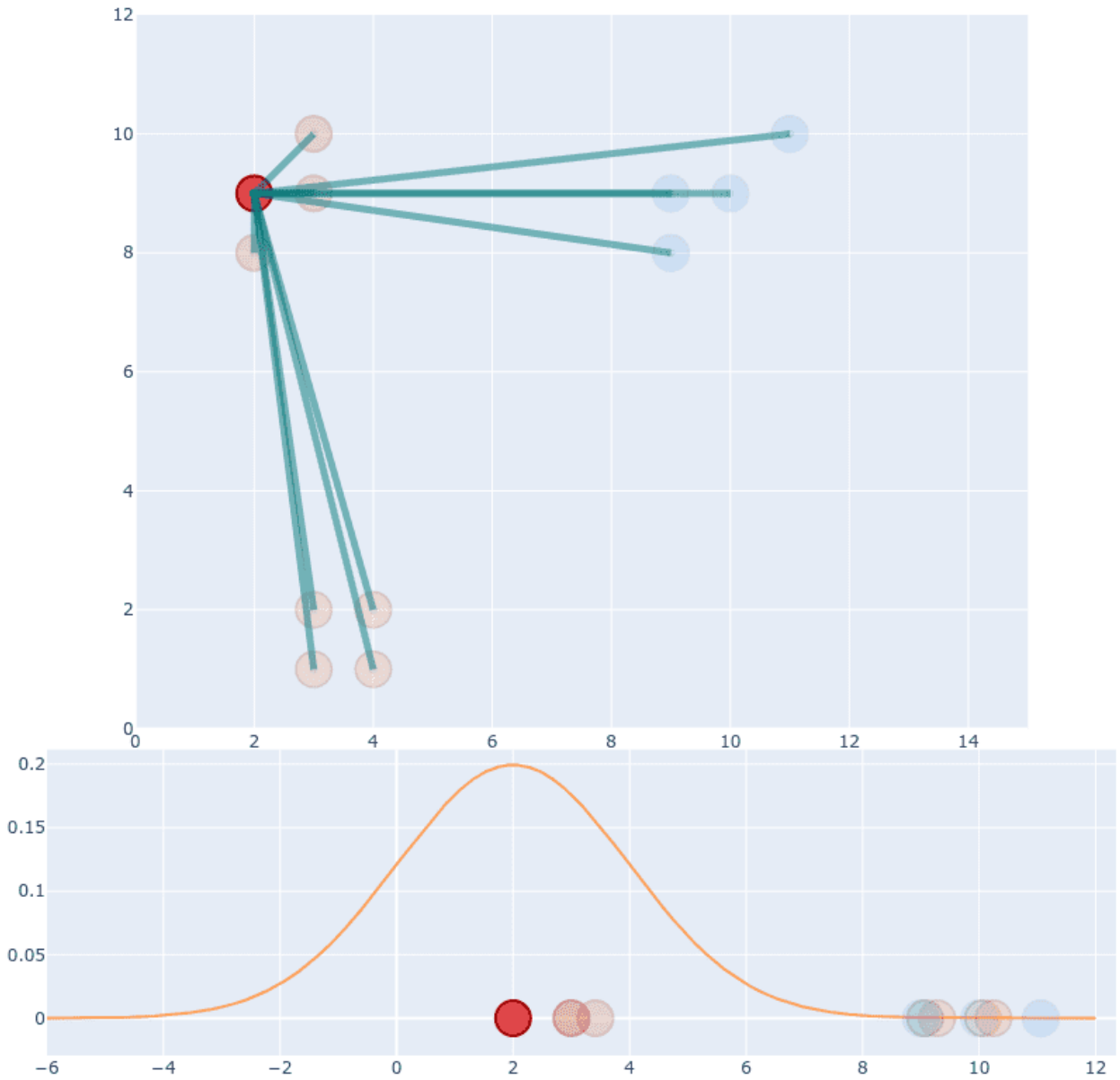
Generate a Gaussian distribution with mean at $\mathbf{x}_i$, and place the distances on the x-axis.





t-SNE

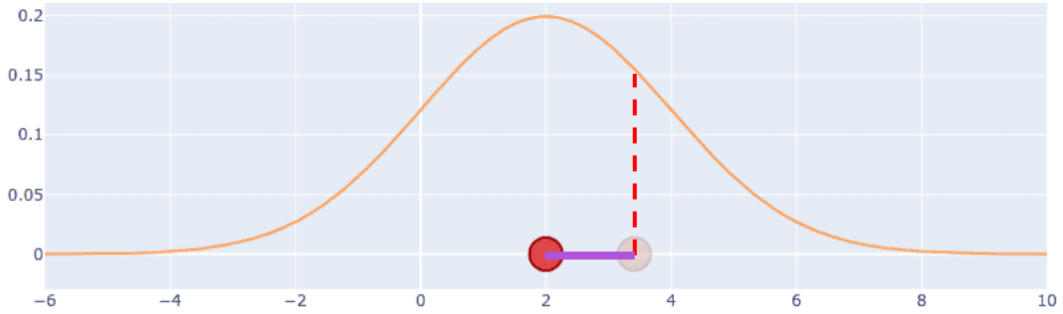
Probabilities become zero for datapoints far away.

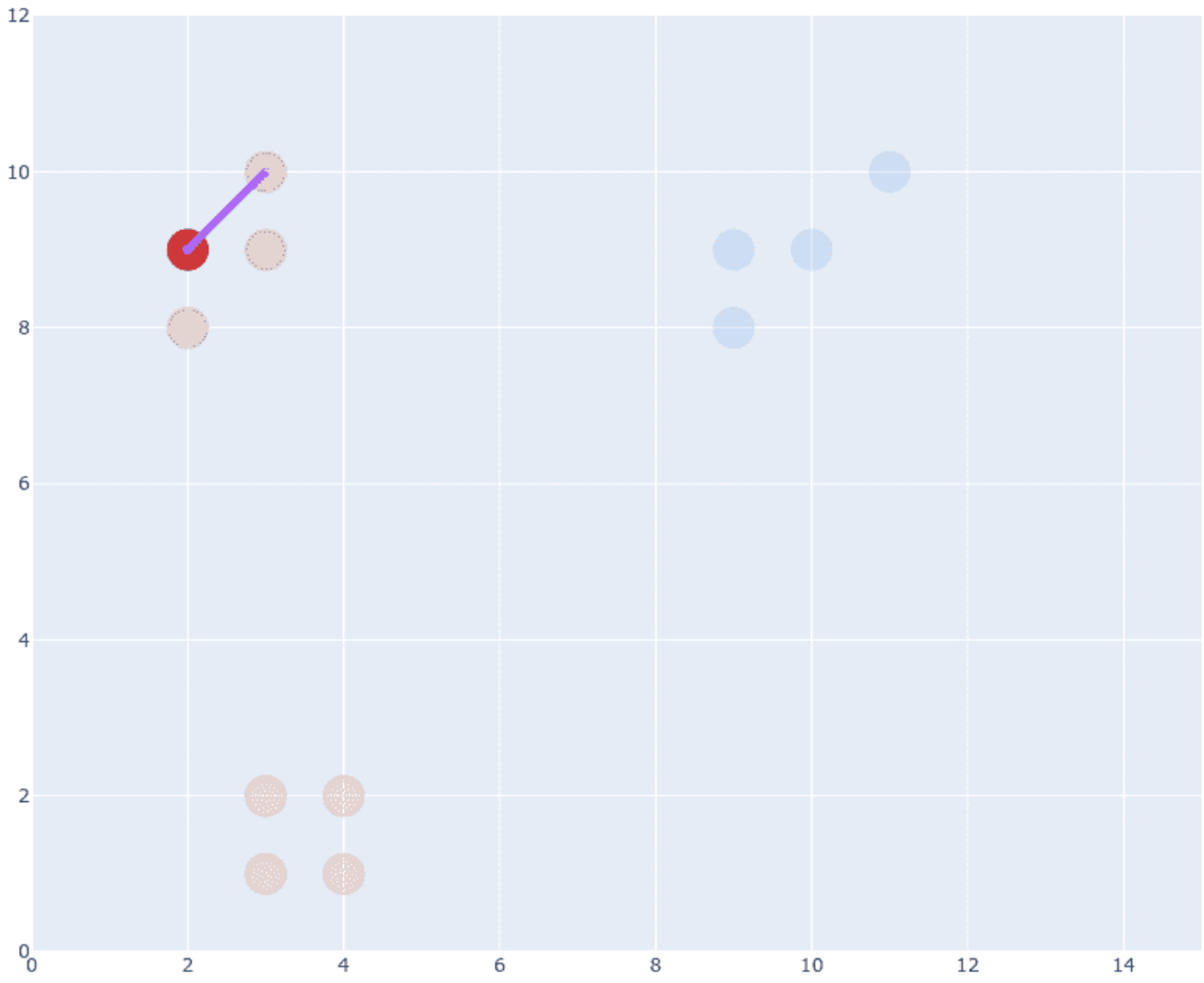


t-SNE

Create a **probability distribution** that represents similarities between neighbours.

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

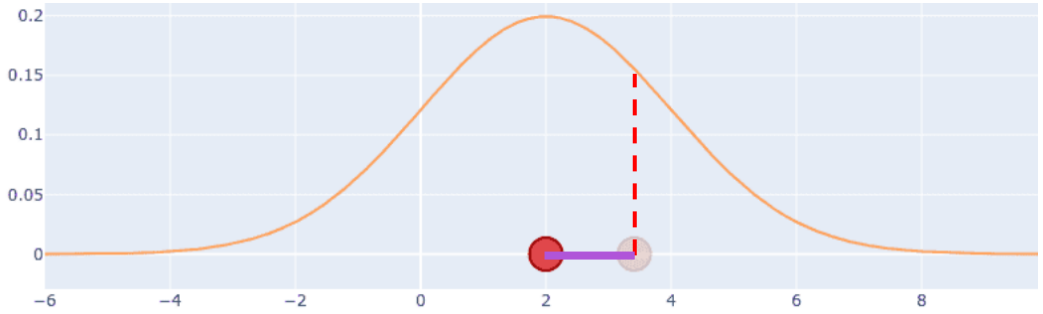


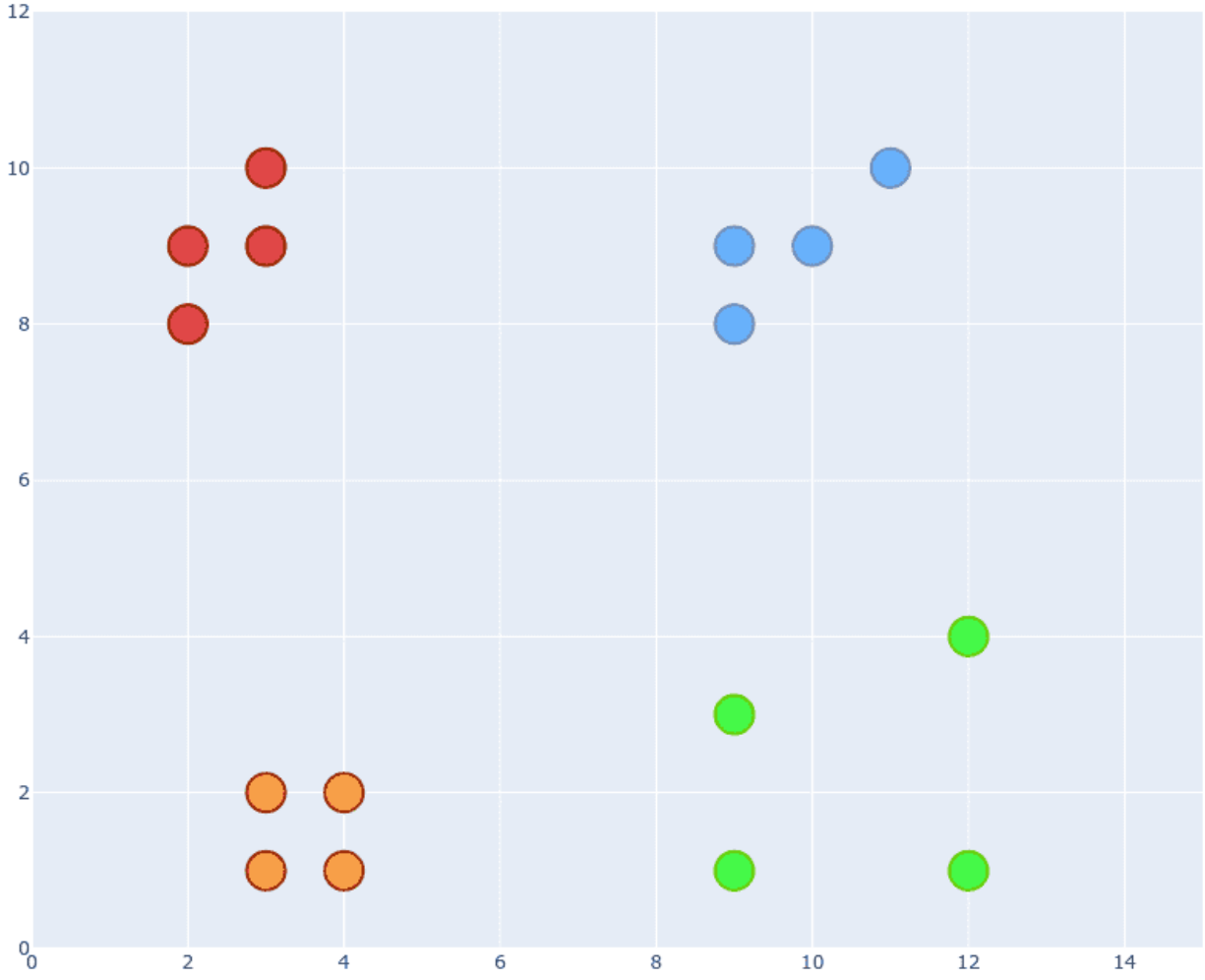


t-SNE

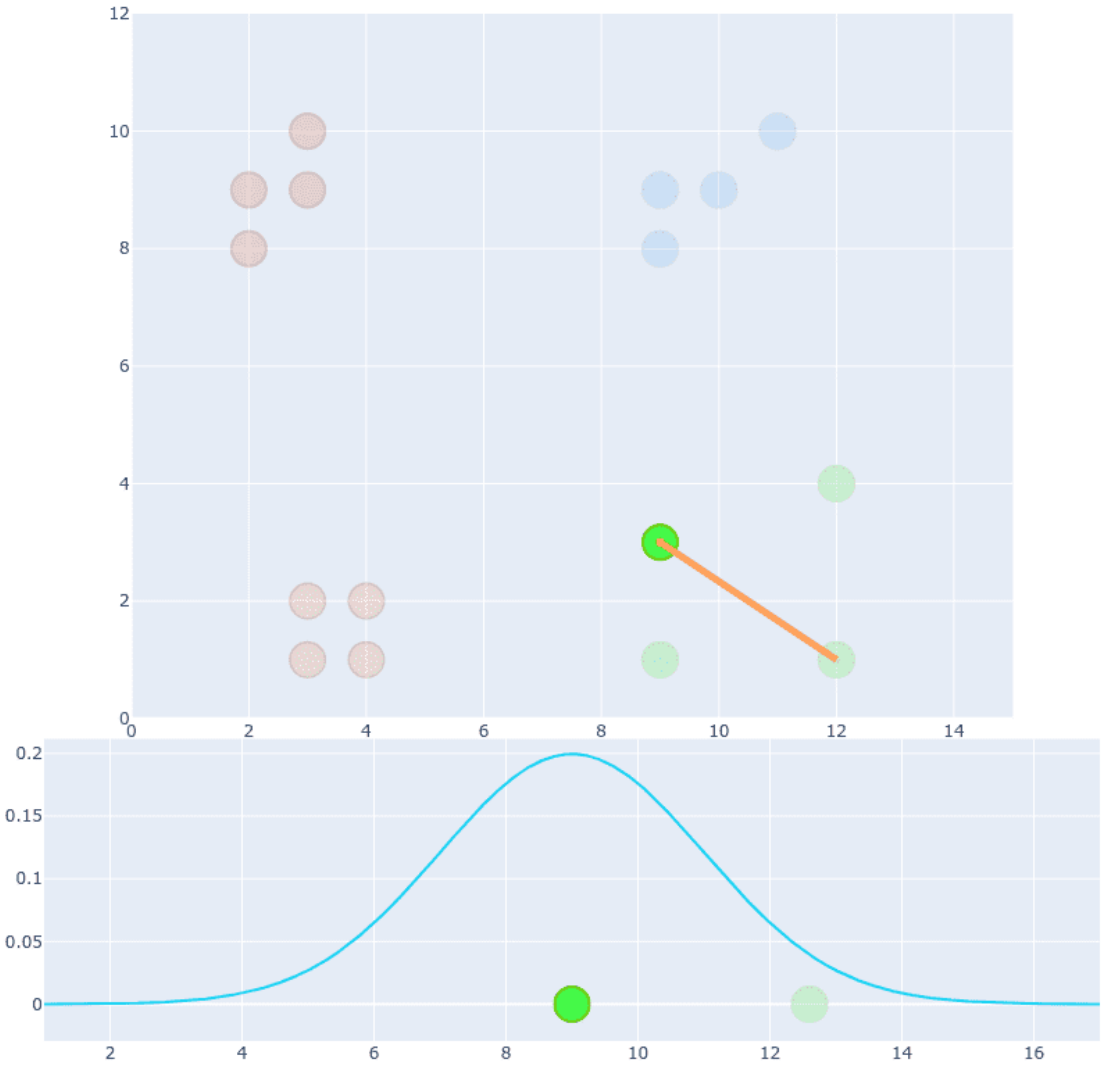
Create a **probability distribution** that represents similarities between neighbours.

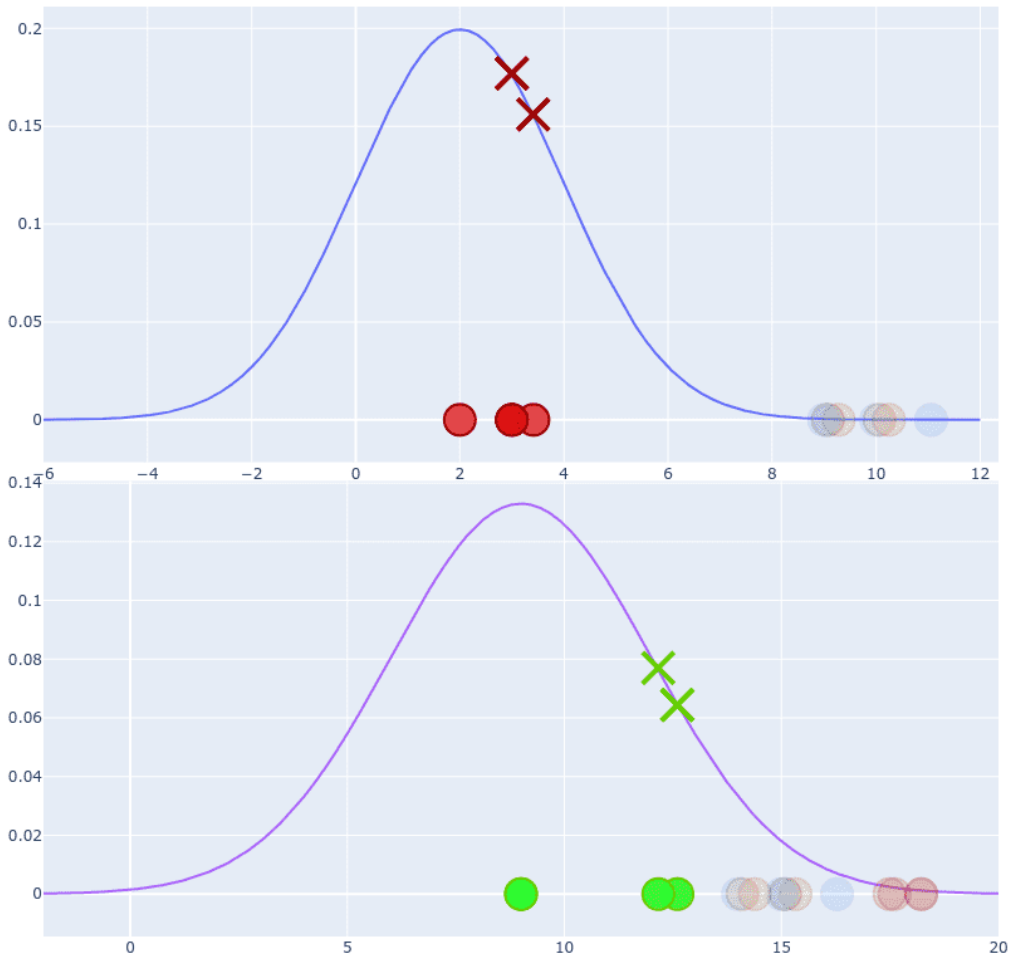
$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$





t-SNE

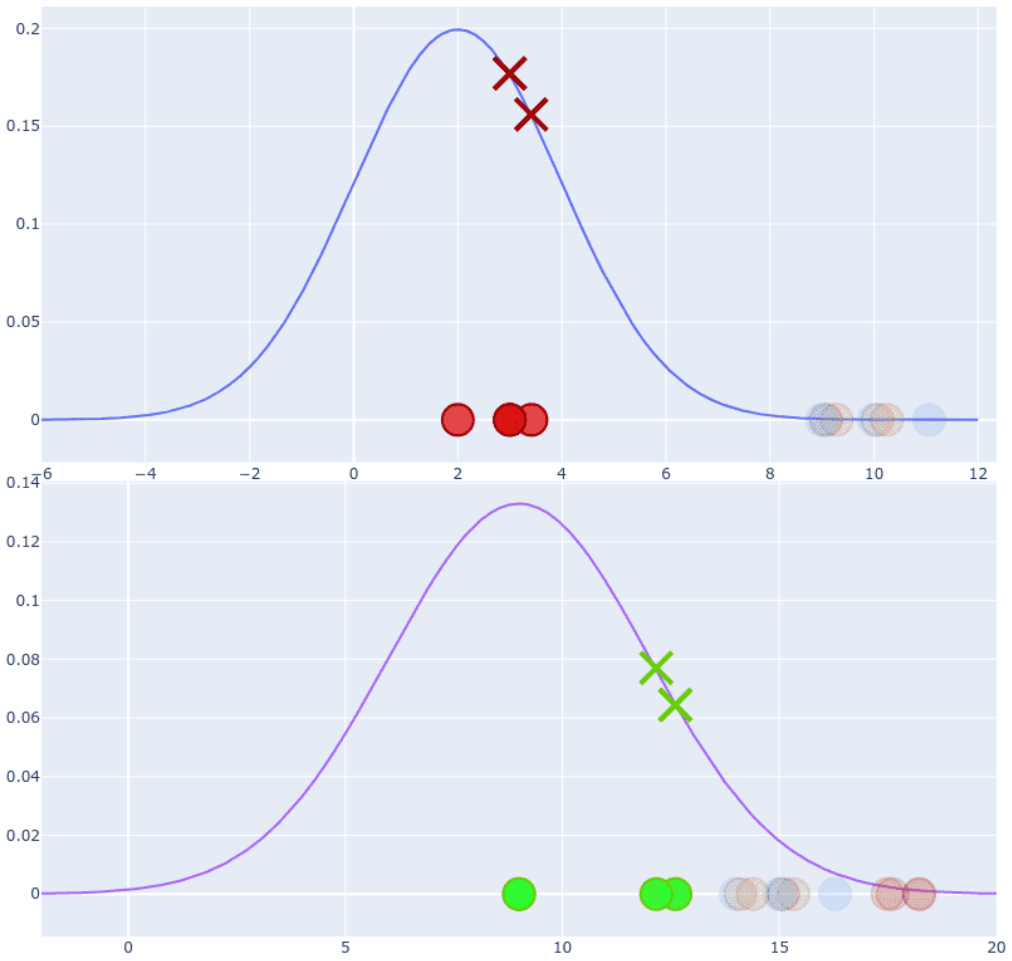

12
10
8
6
4
2
0
0
2
4
6
8
10
12
14
0.2
0.15
0.1
0.05
0
2
4
6
8
10
12
14
16


0.2
0.15
0.1
0.05
0
−6
−4
−2
0
2
4
6
8
10
12
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
5
10
15
20

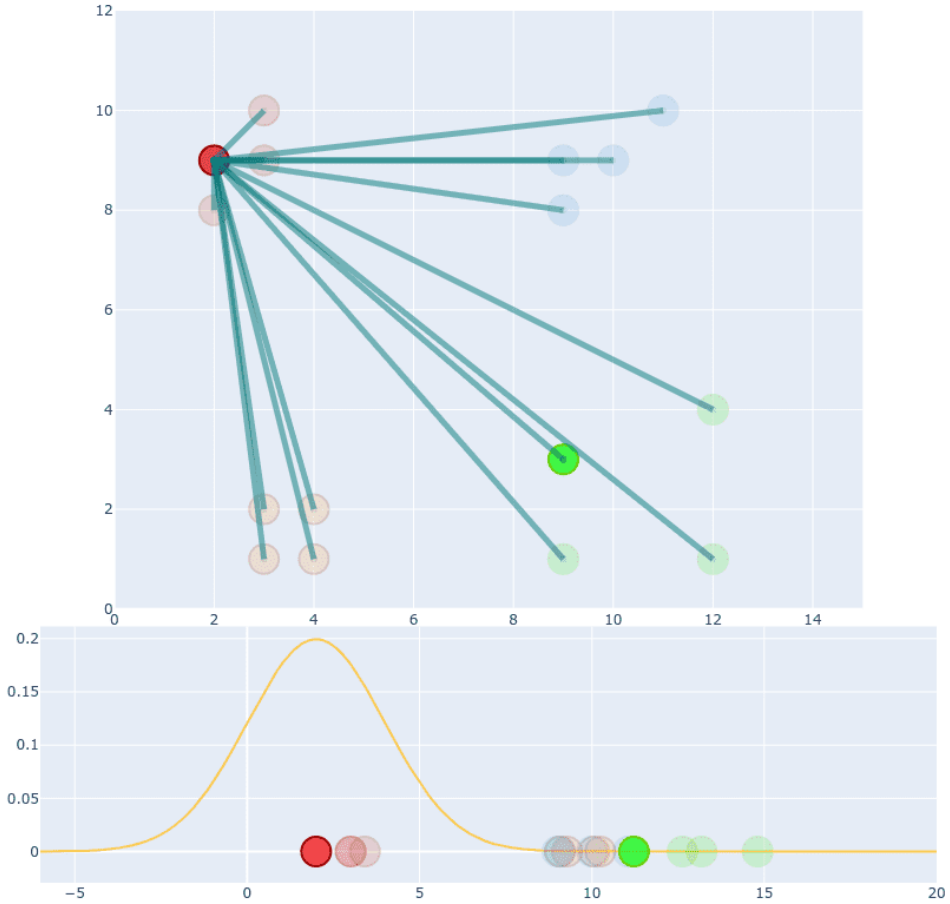
t-SNE

A **perplexity** is more or less a target number of neighbours for our central point.

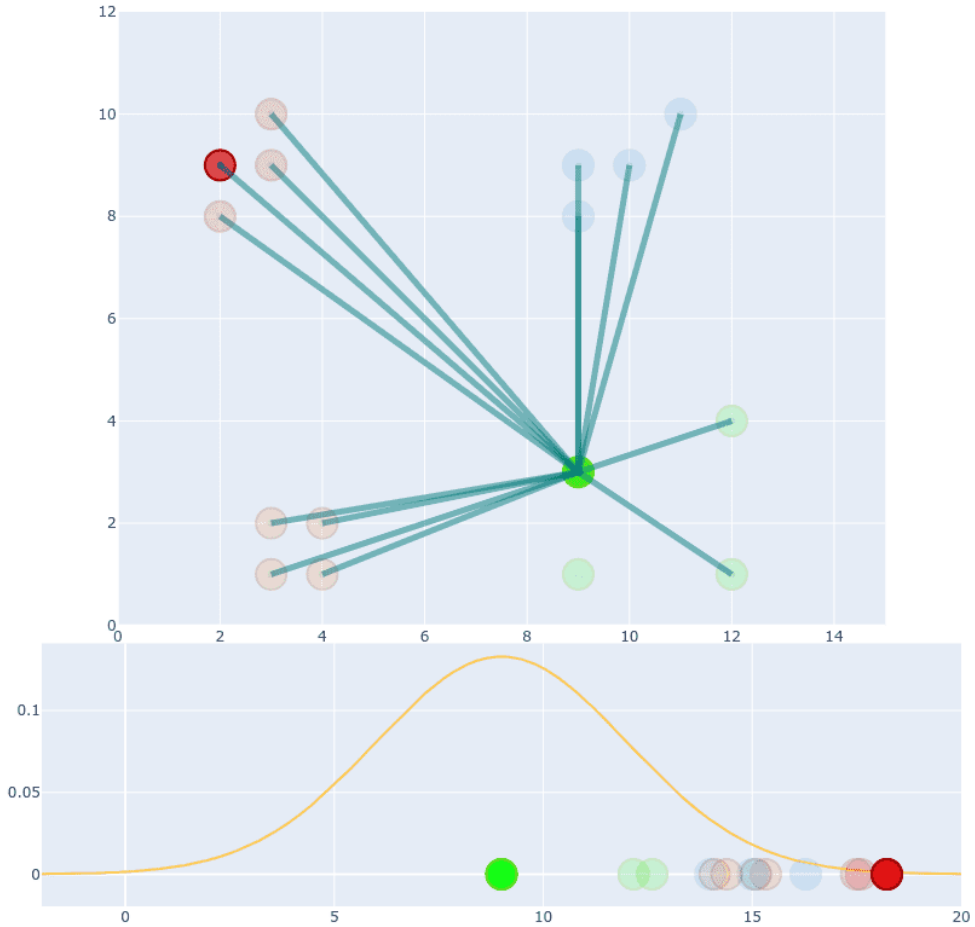
The higher the perplexity is, the higher value the variance of the Gaussian has.



t-SNE



$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$$



t-SNE

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$$

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

SNE

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$$

$$q_{ij} = \frac{\exp(-\|y_i - y_j\|^2/2\sigma_i^2)}{\sum_{k\neq l}\exp(-\|y_k - y_l\|^2/2\sigma_i^2)}$$

$$\mathcal{L} = \sum_{i,j} p_{ij}\log\frac{p_{ij}}{q_{ij}}$$

t-SNE

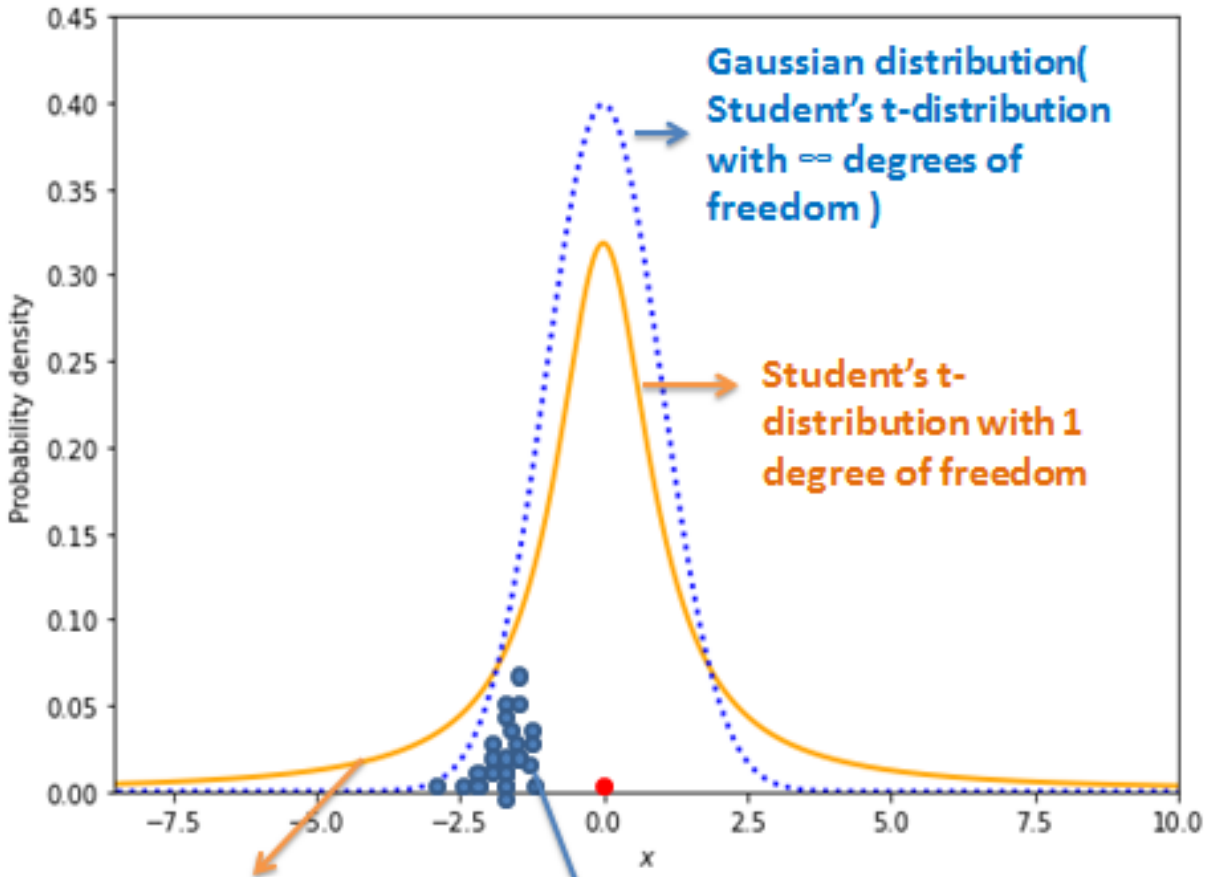
$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$$

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

$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

t-SNE



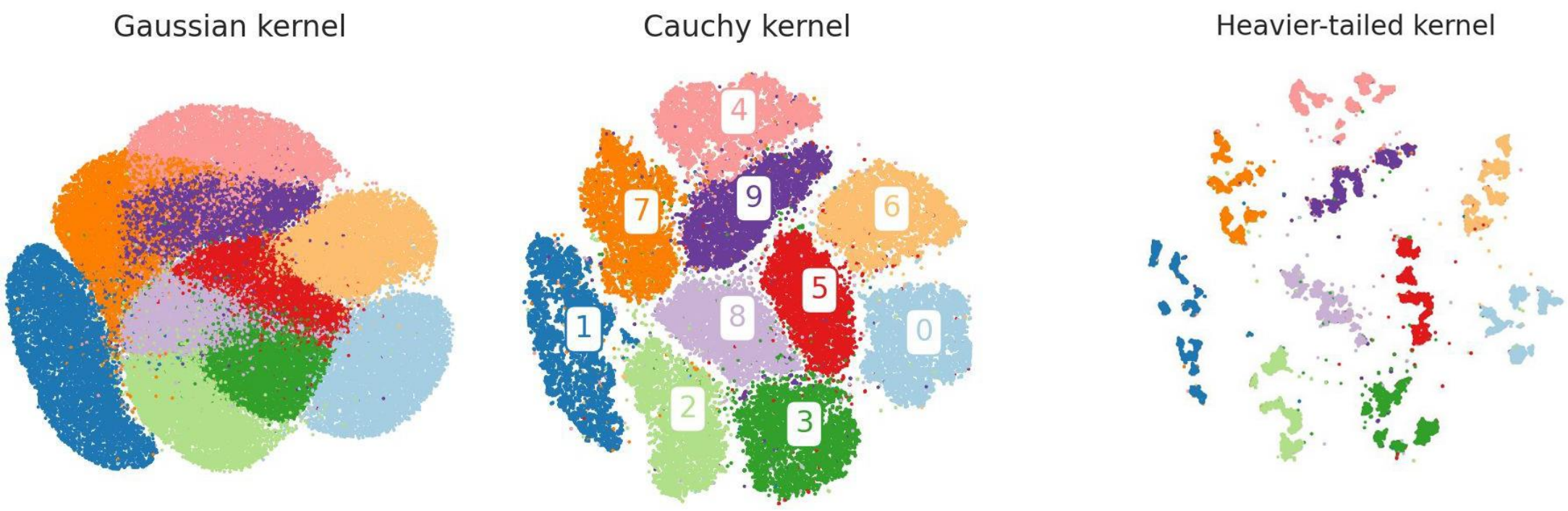
$$q_{ij} = \frac{\exp(-\|y_i - y_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq l} \exp(-\|y_k - y_l\|^2 / 2\sigma_i^2)}$$

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

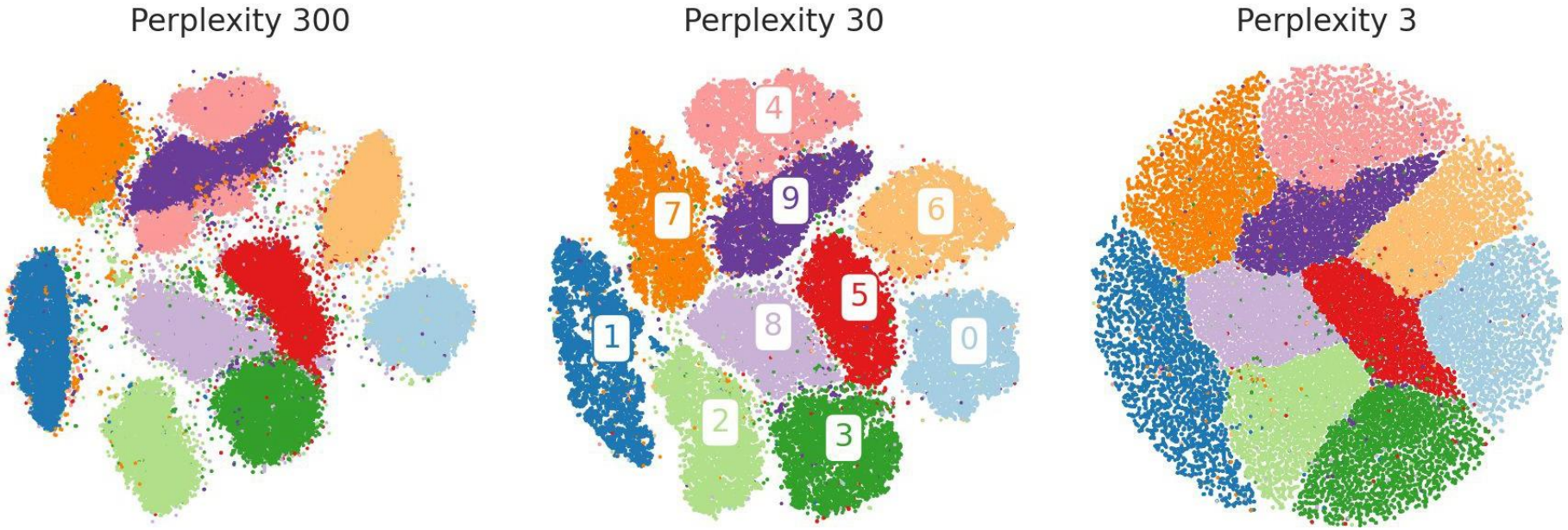
This distribution has a heavy tail that allows the moderately distant points to better preserve their neighborhood.

Less space in a Gaussian distribution for the data points that are at moderate distances from x_i

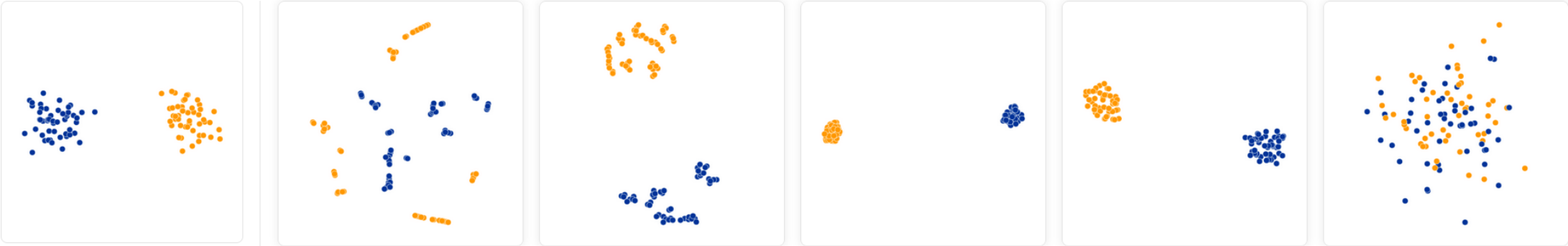
t-SNE


Gaussian kernel
Cauchy kernel
4
7
9
6
5
1
8
0
2
3
Heavier-tailed kernel

t-SNE: Perplexity



t-SNE: Perplexity



Original

Perplexity: 2
Step: 5,000

Perplexity: 5
Step: 5,000

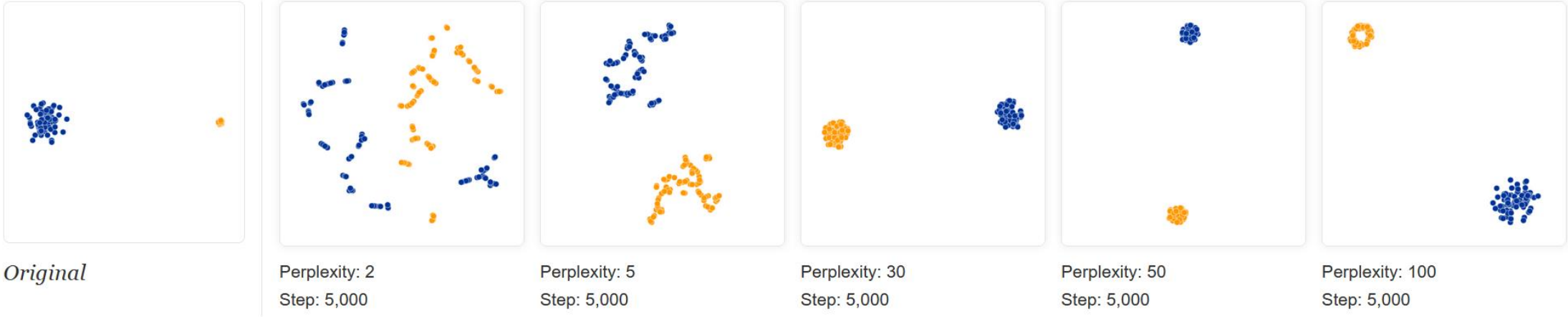
Perplexity: 30
Step: 5,000

Perplexity: 50
Step: 5,000

Perplexity: 100
Step: 5,000

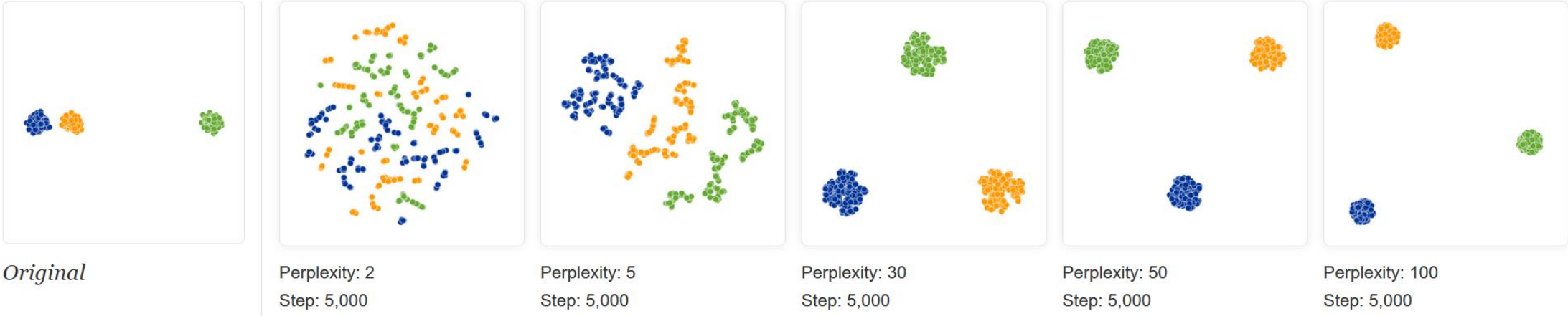
Source: https://distill.pub/2016/misread-tsne/

t-SNE: Cluster sizes in a t-SNE plot mean nothing



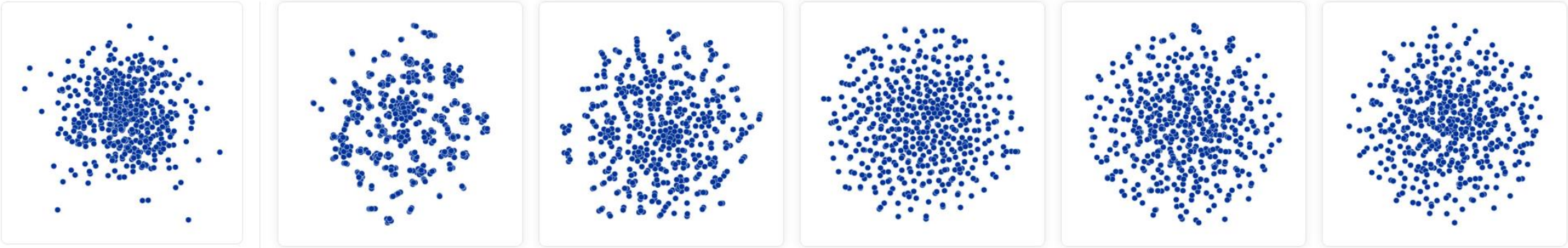
Source: https://distill.pub/2016/misread-tsne/

t-SNE: Distances between clusters might not mean anything



Source: https://distill.pub/2016/misread-tsne/

t-SNE: Random noise doesn’t always look random



Original

Perplexity: 2
Step: 5,000

Perplexity: 5
Step: 5,000

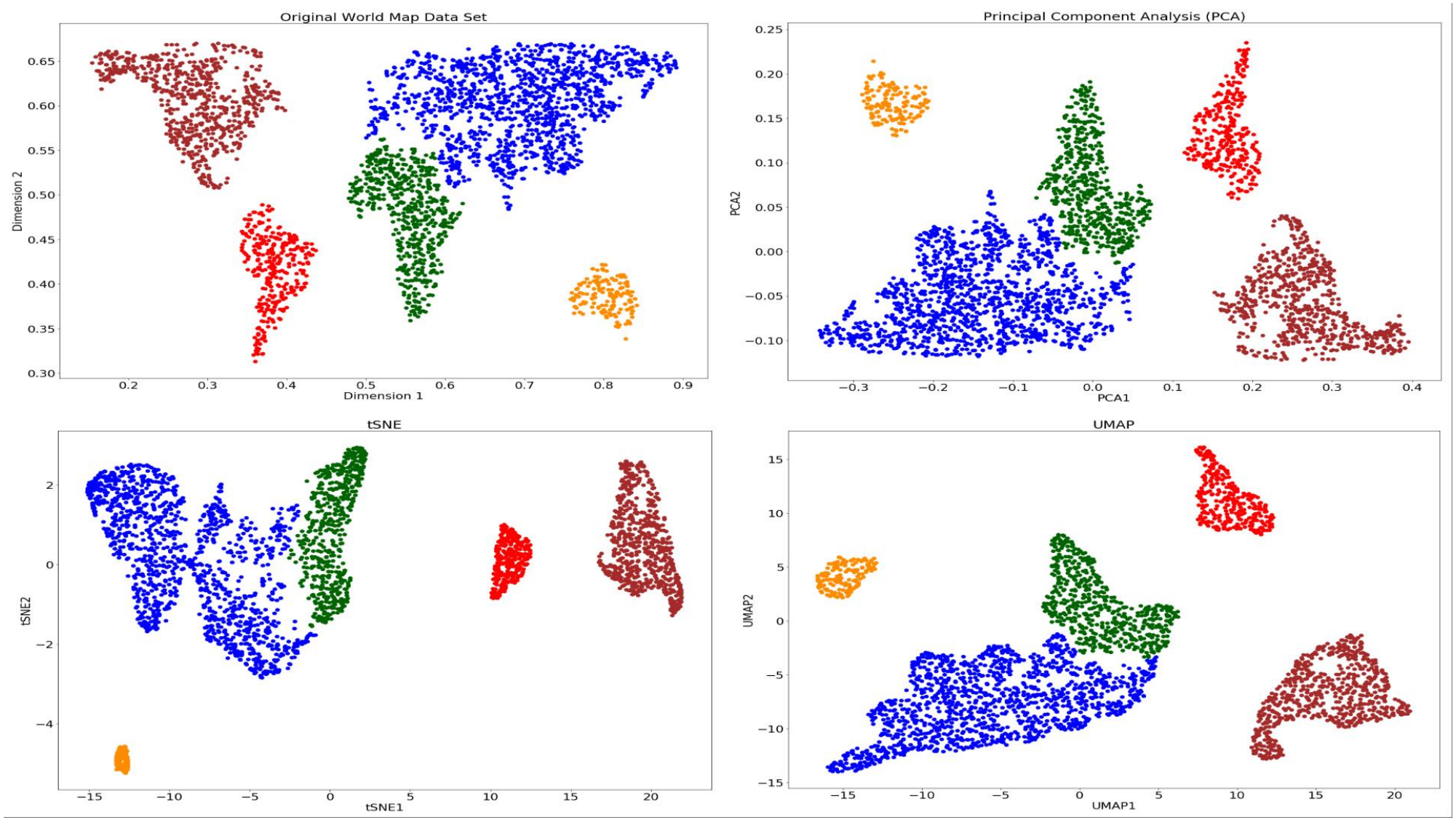
Perplexity: 30
Step: 5,000

Perplexity: 50
Step: 5,000

Perplexity: 100
Step: 5,000

Source: https://distill.pub/2016/misread-tsne/

t-SNE: Global structure is not necessarily preserved

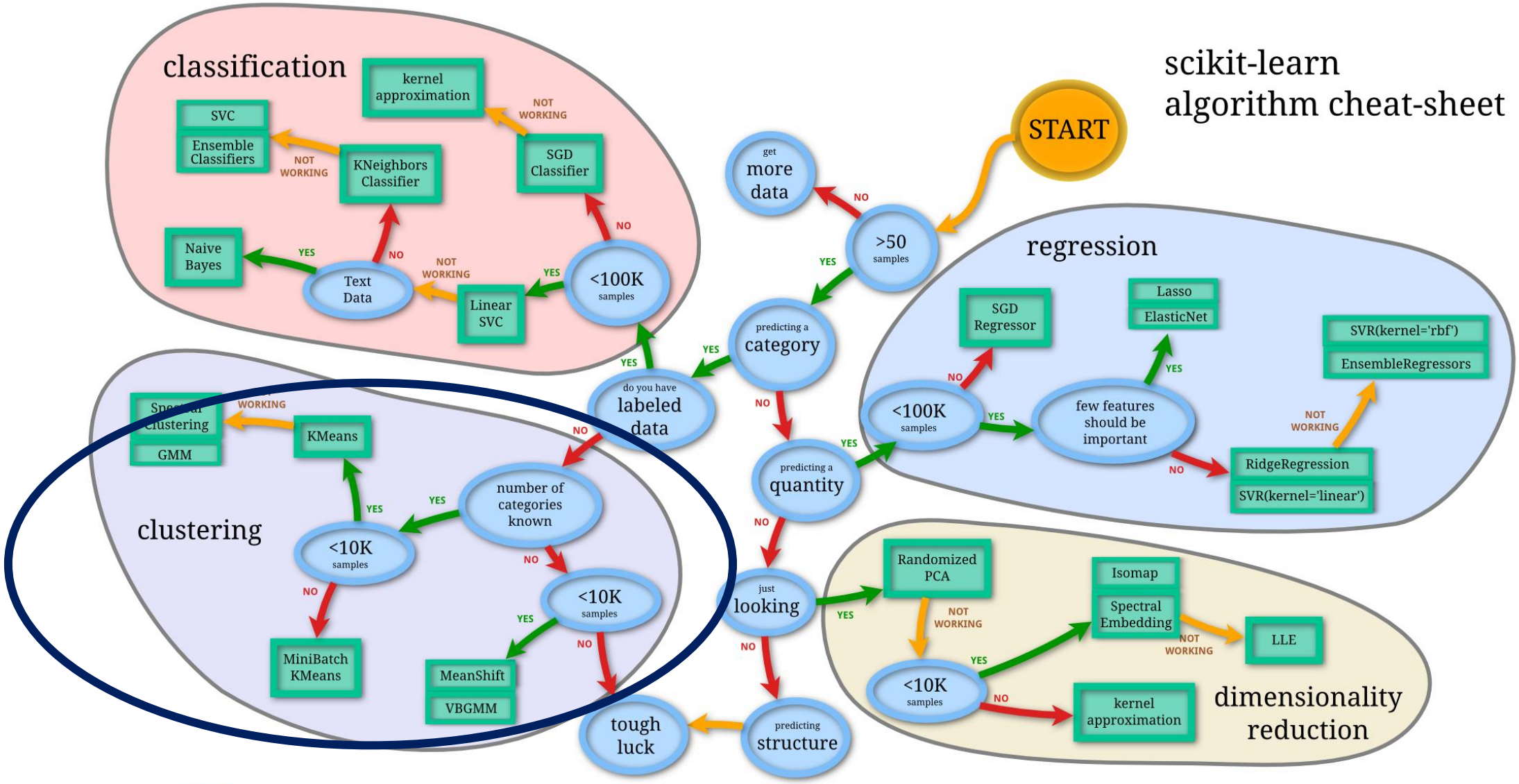


UMAP: Uniform Manifold Approximation and Projection

t-SNE is controlled by the perplexity parameter, UMAP is controlled by a parameter that sets the number of neighbors

UMAP can be much faster to compute. This is interesting for large datasets such as scRNA-seq datasets.

UMAP gives a better balance between local versus global structure, thus overall gives a more accurate presentation of the global structure.

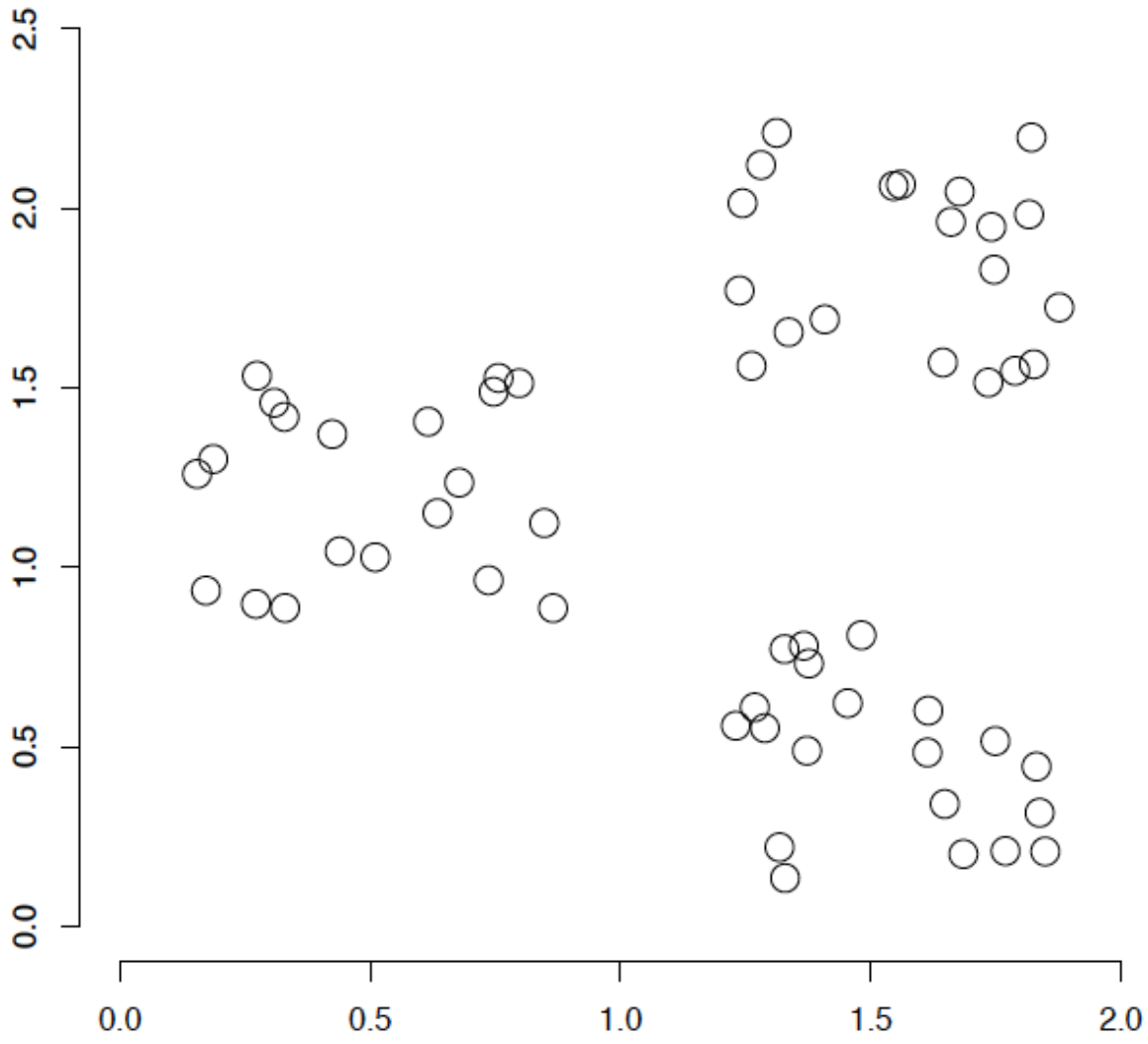

scikit-learn
algorithm cheat-sheet
START
>50
samples
get
more
data
NO
YES
predicting a
category
do you have
labeled
data
predicting a
quantity
just
looking
predicting
structure
tough
luck
classification
kernel
approximation
NOT
WORKING
SVC
Ensemble
Classifiers
KNeighbors
Classifier
SGD
Classifier
Naive
Bayes
Text
Data
Linear
SVC
<100K
samples
regression
SGD
Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
<100K
samples
few features
should be
important
RidgeRegression
SVR(kernel='linear')
clustering
Spectral
Clustering
GMM
KMeans
number of
categories
known
<10K
samples
MiniBatch
KMeans
MeanShift
VBGMM
dimensionality
reduction
Randomized
PCA
Isomap
Spectral
Embedding
LLE
kernel
approximation

Clustering

The process of grouping a set of objects into classes of similar objects:

- datapoints within a cluster should be **similar**
- datapoints from different clusters should be **dissimilar**

Clustering



How many clusters?

K-means clustering

- supervised learning
- unsupervised learning
- hidden structure
- data groups
- feature relevance/overfitting

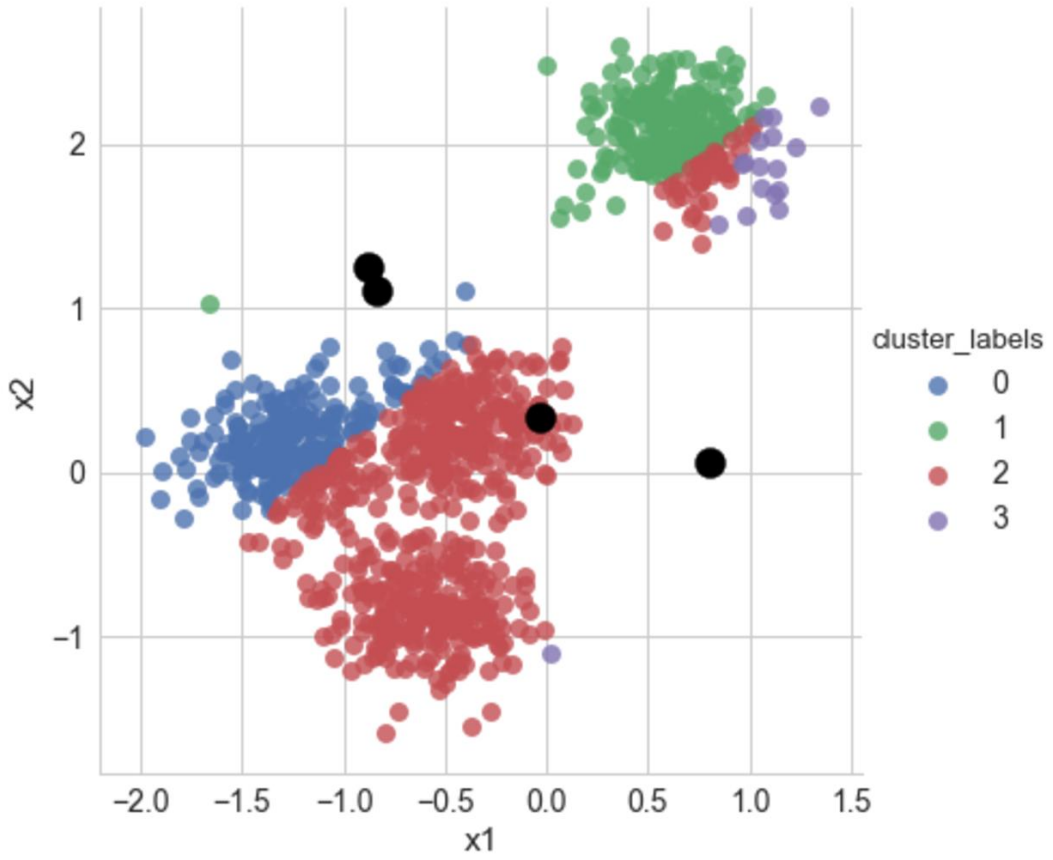
K-means clustering

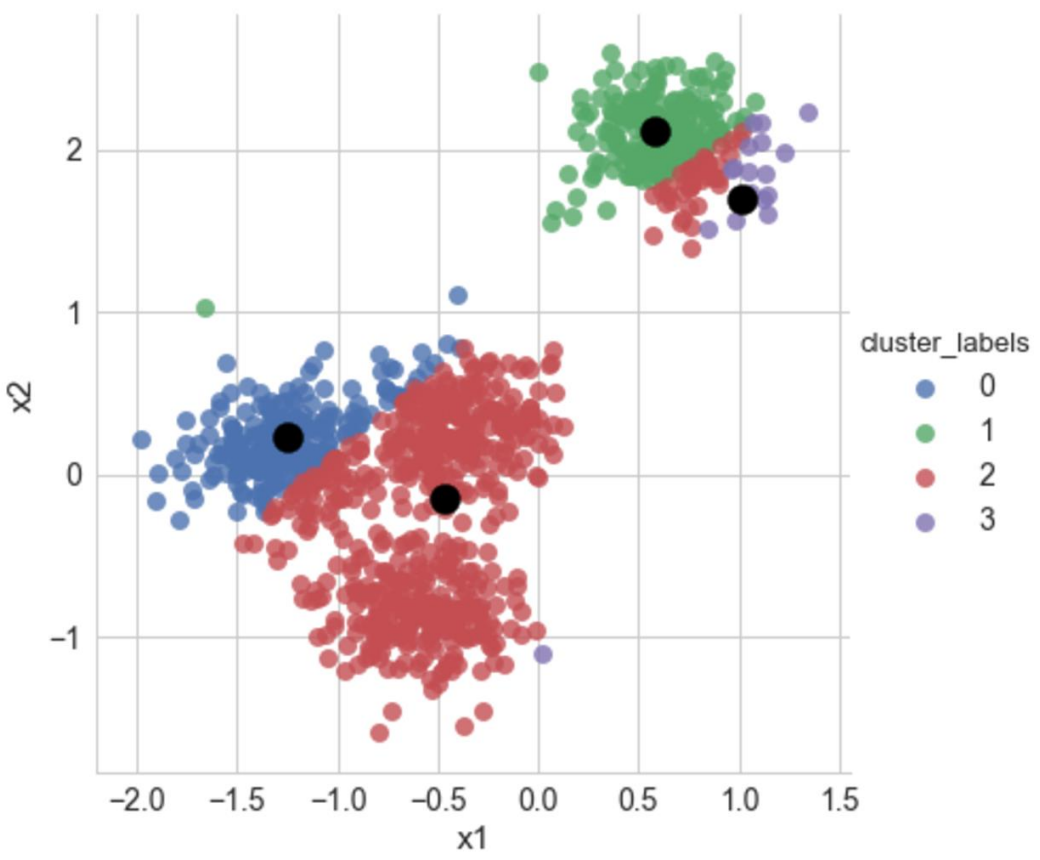
$$\mu_1^{(0)}, \ldots, \mu_k^{(0)} \in \mathbb{R}^n$$

$$c_i = \underset{j}{\text{argmin}} \; \|x^{(i)} - \mu_j^{(s)}\|$$

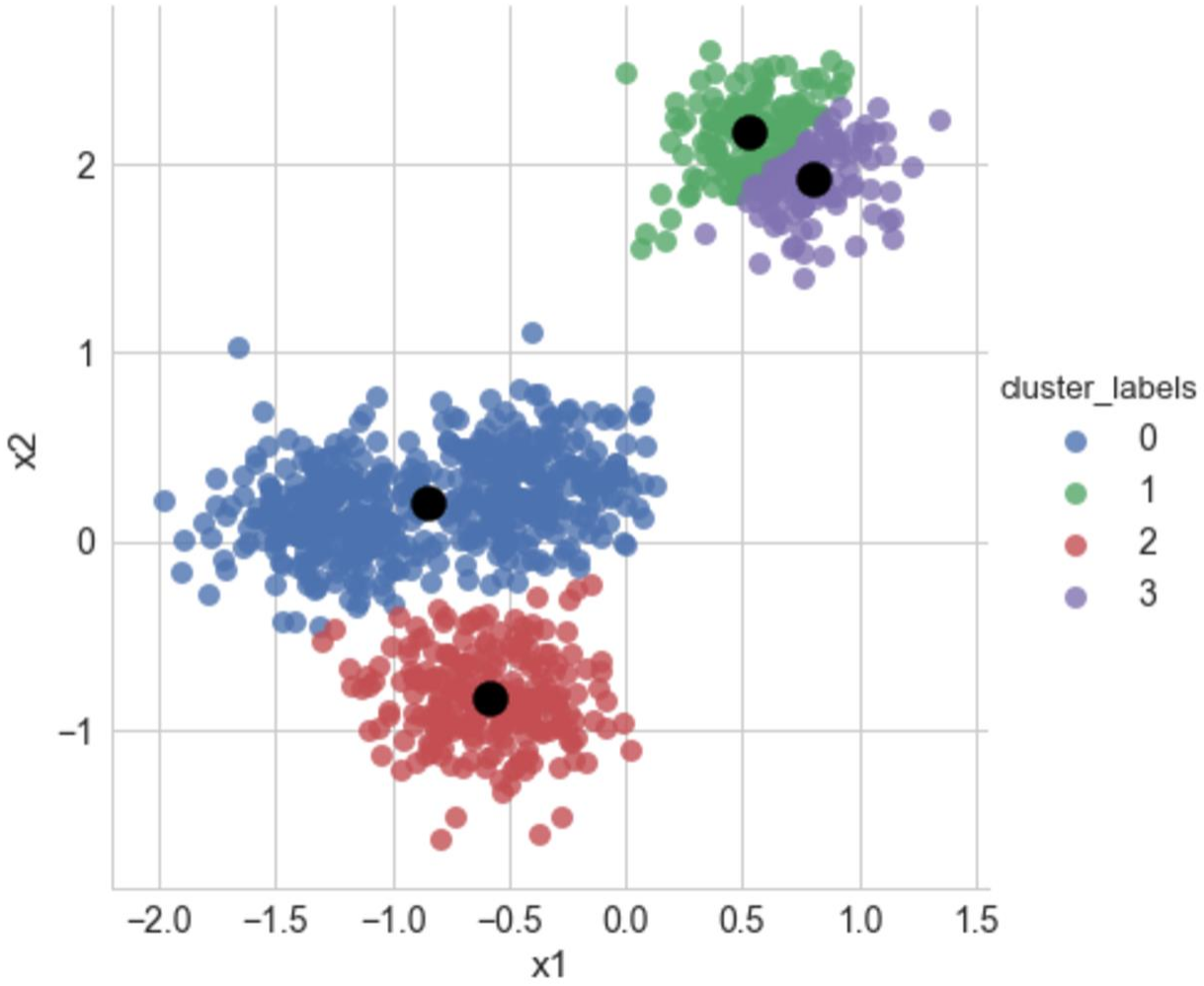
$$\mu_j^{(s+1)} = \frac{\sum_{i=1}^n I[c_i = j] x^{(i)}}{\sum_{i=1}^n I[c_i = j]}$$

K-means clustering

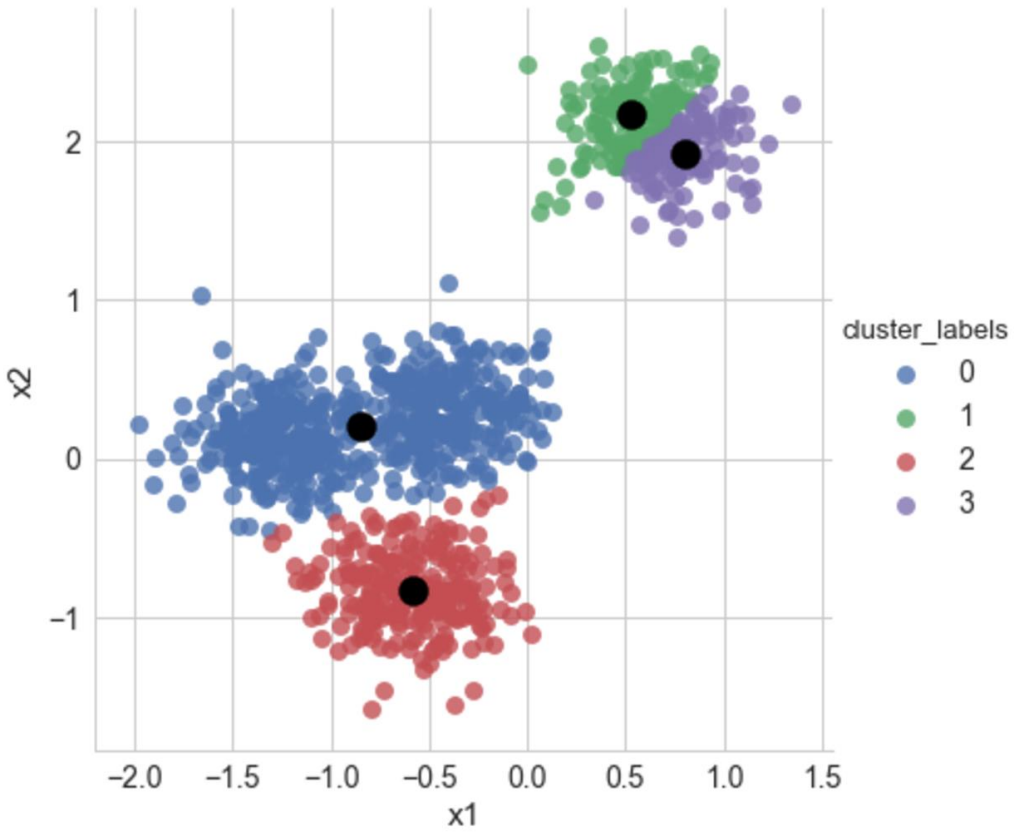




K-means clustering

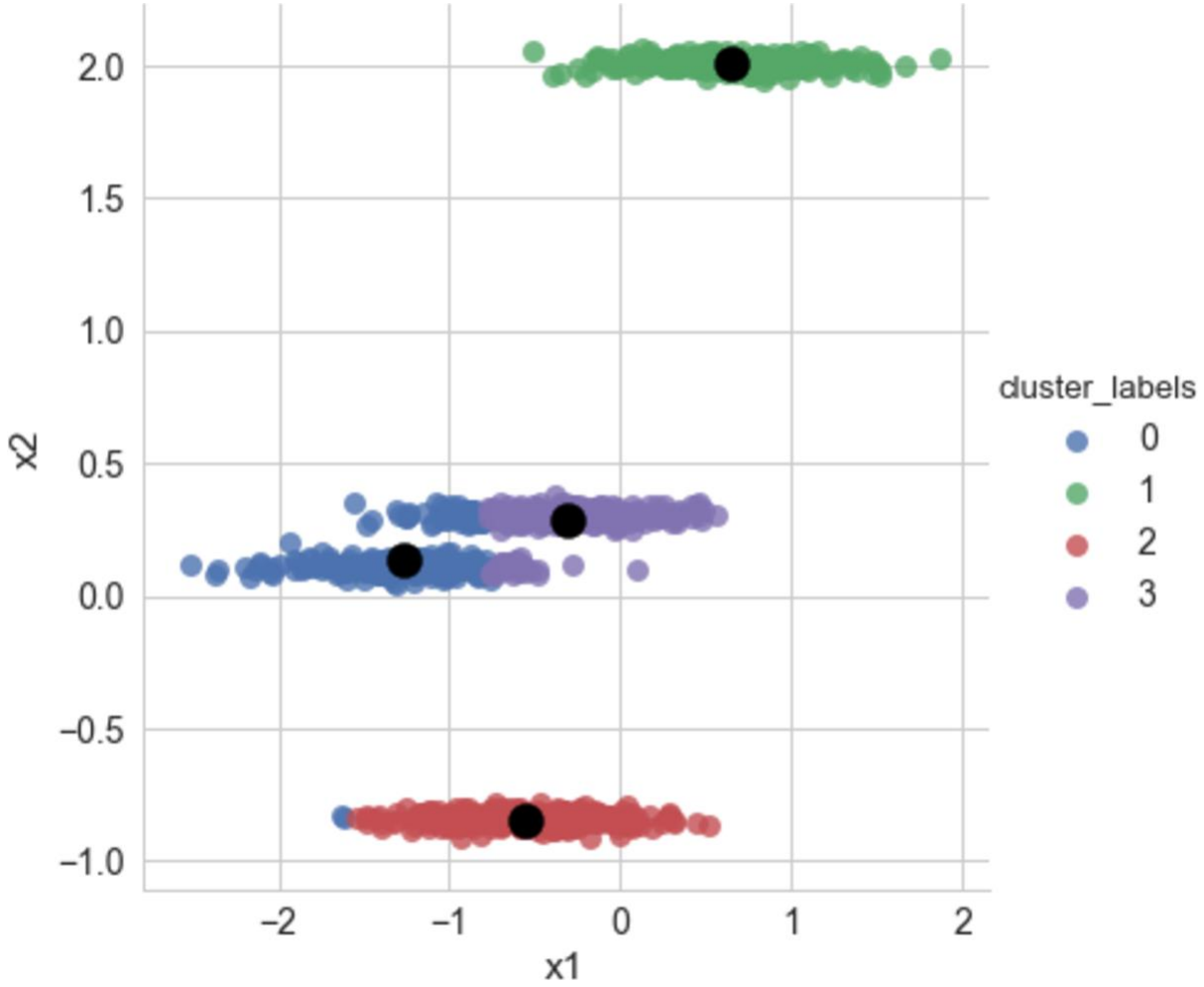


K-means++ clustering



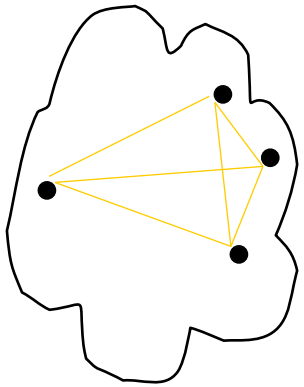
- the first centroid is chosen uniformly at random from the data points that are being clustered
- each subsequent centroid is chosen from the remaining data points with probability proportional to its squared distance from the point's closest existing centroid
- better spread of the initial centroids

k-means++ clustering

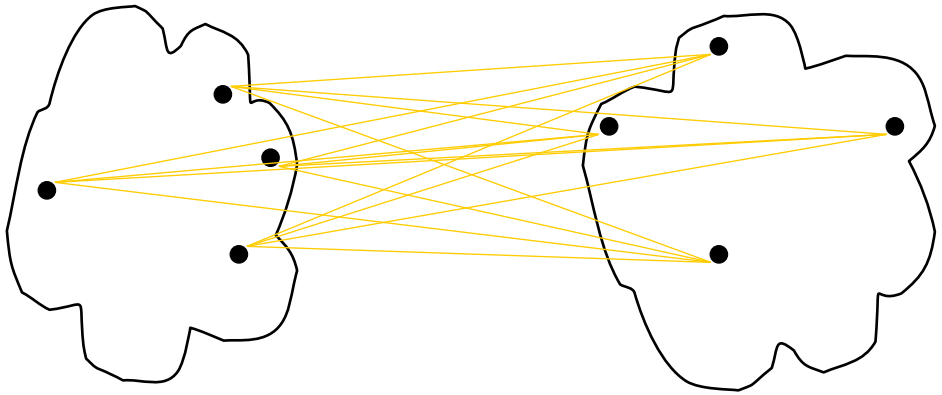


Clustering evaluation

- **Cohesion**: measures how closely related are objects in a cluster
- **Separation**: measure how well-separated a cluster is from other clusters



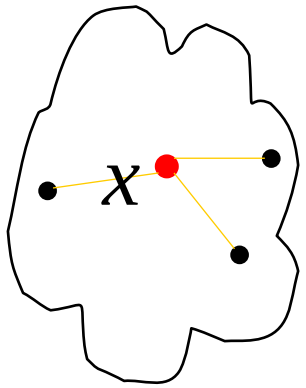
cohesion



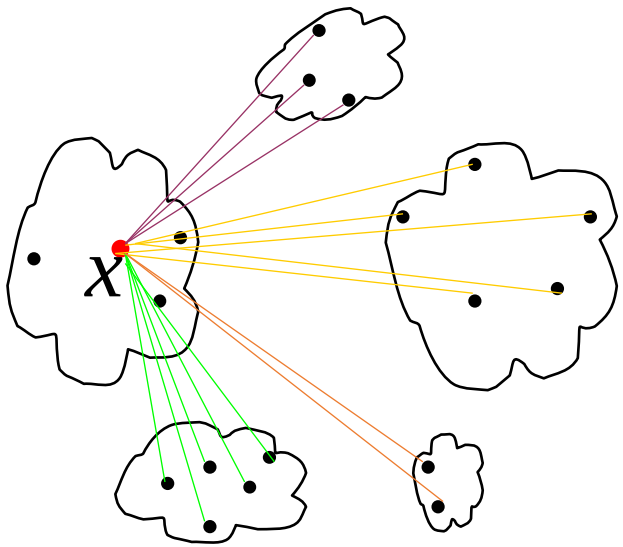
separation

Clustering evaluation

- **Cohesion a(x)**: the mean distance between the data point and all other points in the same cluster
- **Separation b(x)**: the mean distance between the data point and all other points in the next nearest cluster



cohesion



separation

Clustering evaluation

- **Silhouette score s(x)**:

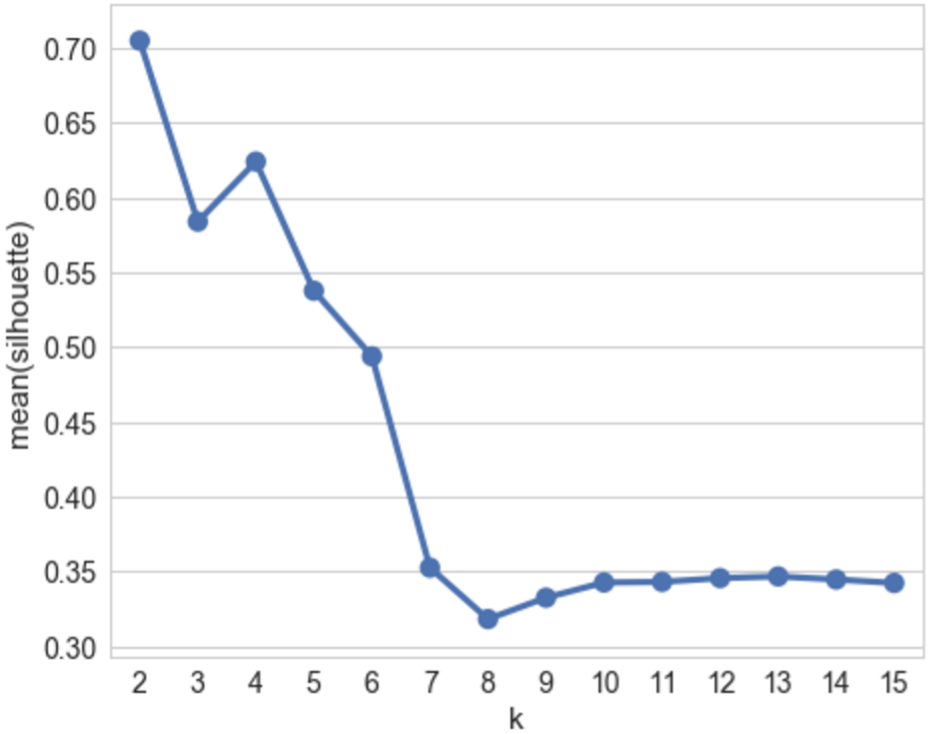
$$s(x) = \frac{b(x) - a(x)}{max(a(x), b(x))}$$

- **Silhouette coefficient *SC***:

$$SC = \frac{1}{n} \sum_{i=1}^{n} s(x)$$

- **Inertia**: the sum of squared distance for each point to it's closest cluster centroid, i.e. its assigned cluster

Clustering evaluation



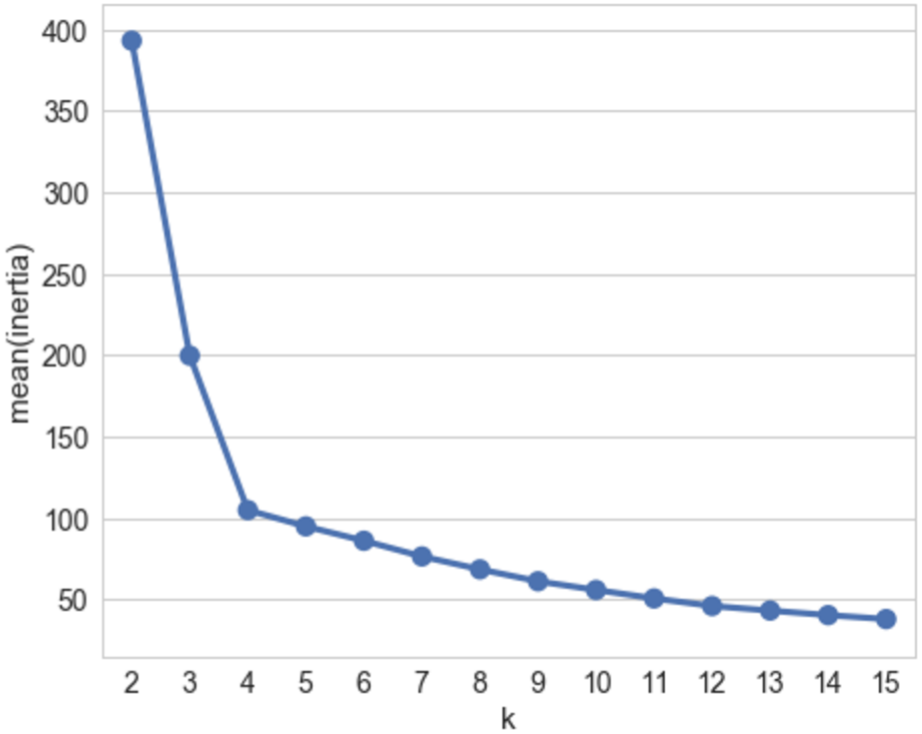
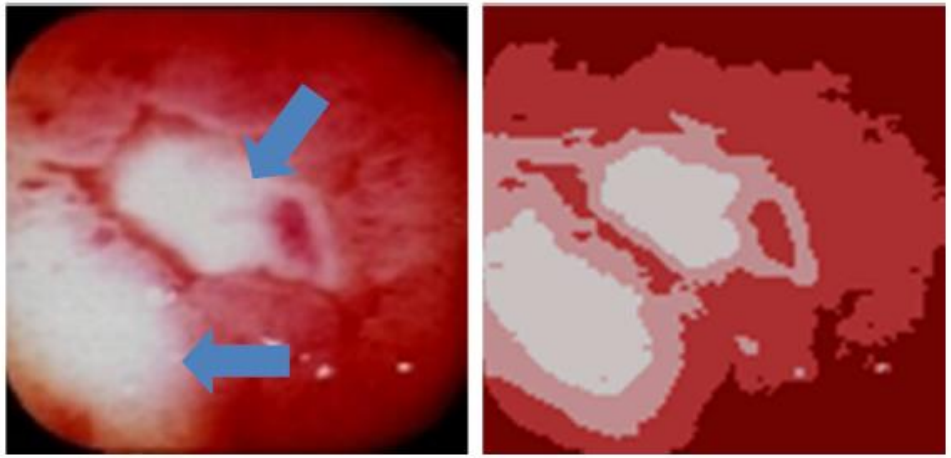


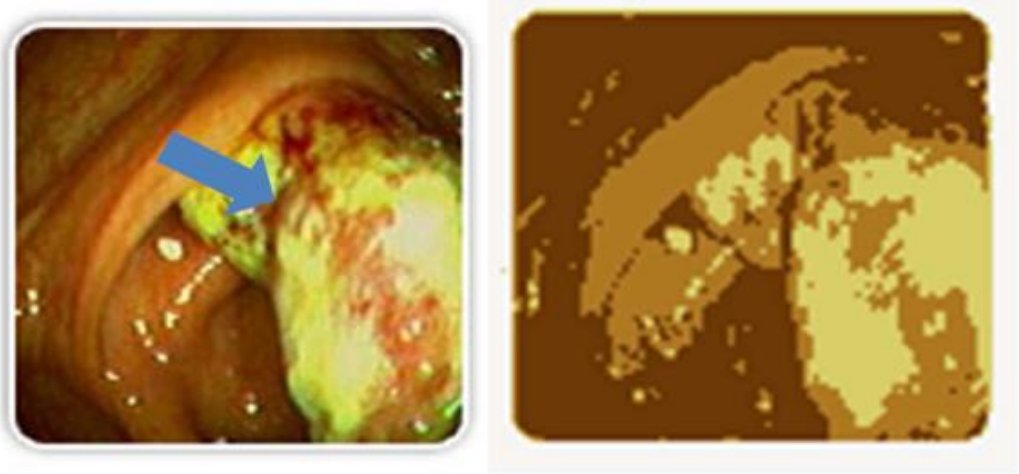
Image segmentation:

- Each image is represented in the *RGB* color space.
- An image pixel is represented as a 3D color vector
- Pixels are clustered to find the segments.



(a) (b)

FIGURE 1. Results of segmentation on an image diagnosed with duodenal ulcer: (a) Original image; (b) Image segmented with 4 regions.



(a) (b)

FIGURE 3. Results of segmentation on an image diagnosed with colon cancer: (a) Original image; (b) Image segmented with 4 regions.