Machine Learning Course

Lecture 14: Unsupervised learning

Harbour.Space University
March 2020

Radoslav Neychev
Vladislav Goncharenko

Outline

1. Dimensionality reduction
   a. Multidimensional Scaling (MDS)
   b. Isomap
   c. Locally linear embedding (LLE)
   d. t-SNE
   e. LargeVis
2. Clustering
   a. K-means
   b. DBSCAN

Dimensionality reduction

Multidimensional Scaling (MDS)

Decrease the dimensionality using linear methods

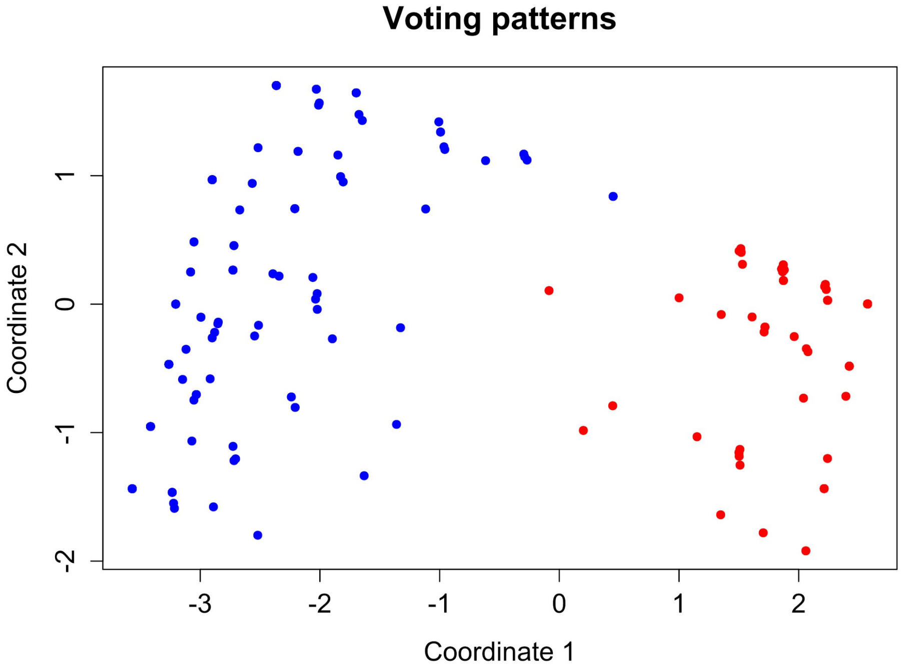
How to do that?

PCA

$$L = ||D_x - D_y||_2 \to \min_{y=Ax}$$

$$y = \Lambda^{1/2} V^T$$

Parameters: p - target dimensionality



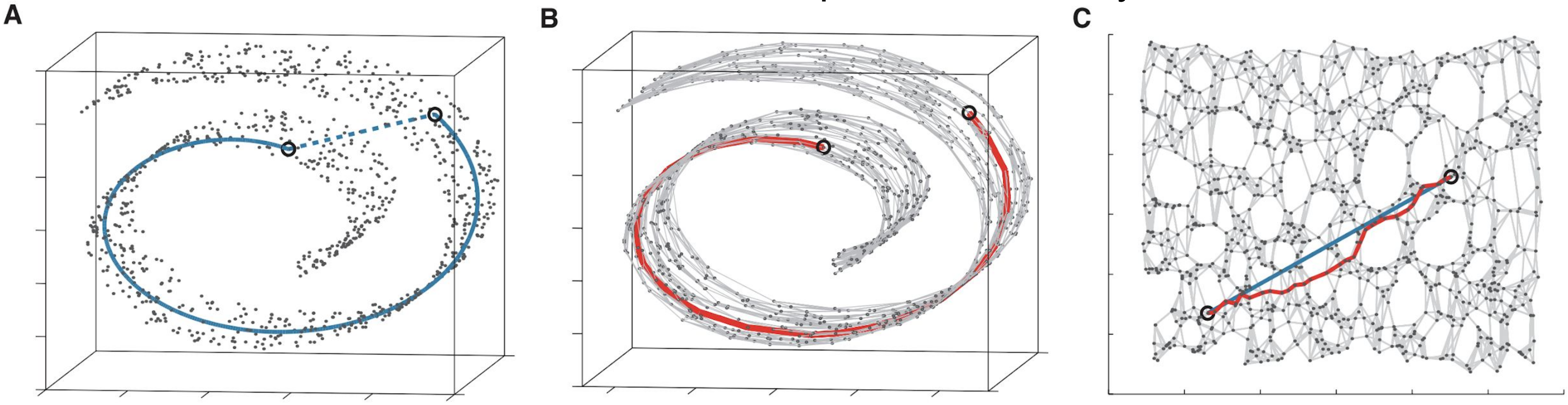
Wikipedia page: [link]

Isomap

Idea: Make distancies geodesic!
Measure distances on the produced graph.

Parameters:

- n - number of neighbours to connect
- p - dimensionality of manifold



Source: [A Global Geometric Framework for Nonlinear Dimensionality Reduction]

Isomap

Step		
1	Construct neighborhood graph	Define the graph G over all data points by connecting points i and j if [as measured by $d_X(i,j)$] they are closer than ϵ (ϵ-Isomap), or if i is one of the K nearest neighbors of j (K-Isomap). Set edge lengths equal to $d_X(i,j)$.
2	Compute shortest paths	Initialize $d_G(i,j) = d_X(i,j)$ if i,j are linked by an edge; $d_G(i,j) = \infty$ otherwise. Then for each value of $k = 1, 2, \ldots, N$ in turn, replace all entries $d_G(i,j)$ by $\min\{d_G(i,j), d_G(i,k) + d_G(k,j)\}$. The matrix of final values $D_G = \{d_G(i,j)\}$ will contain the shortest path distances between all pairs of points in G (*16*, *19*).
3	Construct d-dimensional embedding	Let λ_p be the p-th eigenvalue (in decreasing order) of the matrix $\tau(D_G)$ (*17*), and v_p^i be the i-th component of the p-th eigenvector. Then set the p-th component of the d-dimensional coordinate vector $\mathbf{y}_i$ equal to $\sqrt{\lambda_p} v_p^i$.

Floyd–Warshall algorithm

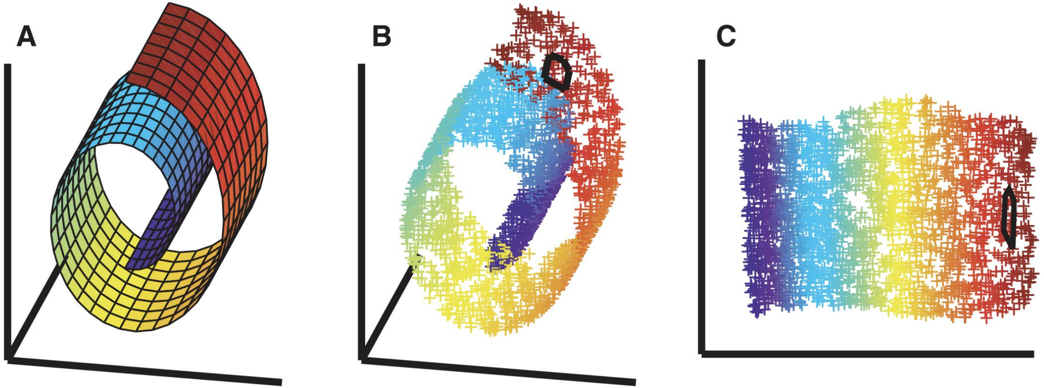
17. The operator τ is defined by $\tau(D) = -HSH/2$, where S is the matrix of squared distances $\{S_{ij} = D_{ij}^2\}$, and H is the "centering matrix" $\{H_{ij} = \delta_{ij} - 1/N\}$ (*13*).

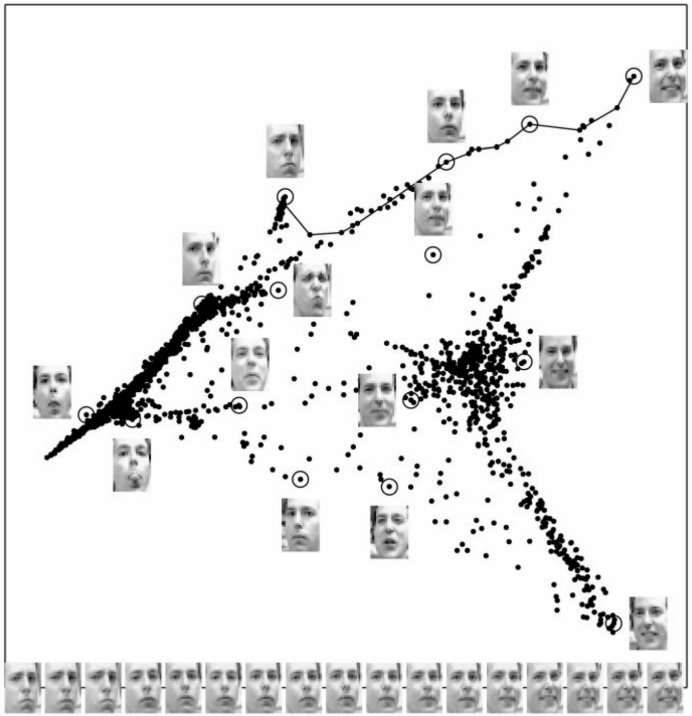
Locally linear embedding (LLE)

Idea:

Smooth manifold can be locally approximated linearly.

Linear pieces can be flattened





Source: [Nonlinear Dimensionality Reduction by Locally Linear Embedding]()

Locally linear embedding (LLE)

Two steps of embedding and two objective functions:

1. estimate point by its K neighbours

$$\varepsilon(W) = \sum_{i=1}^{n} ||x_i - \sum_{j=1}^{K} W_{ij} x_j||^2$$

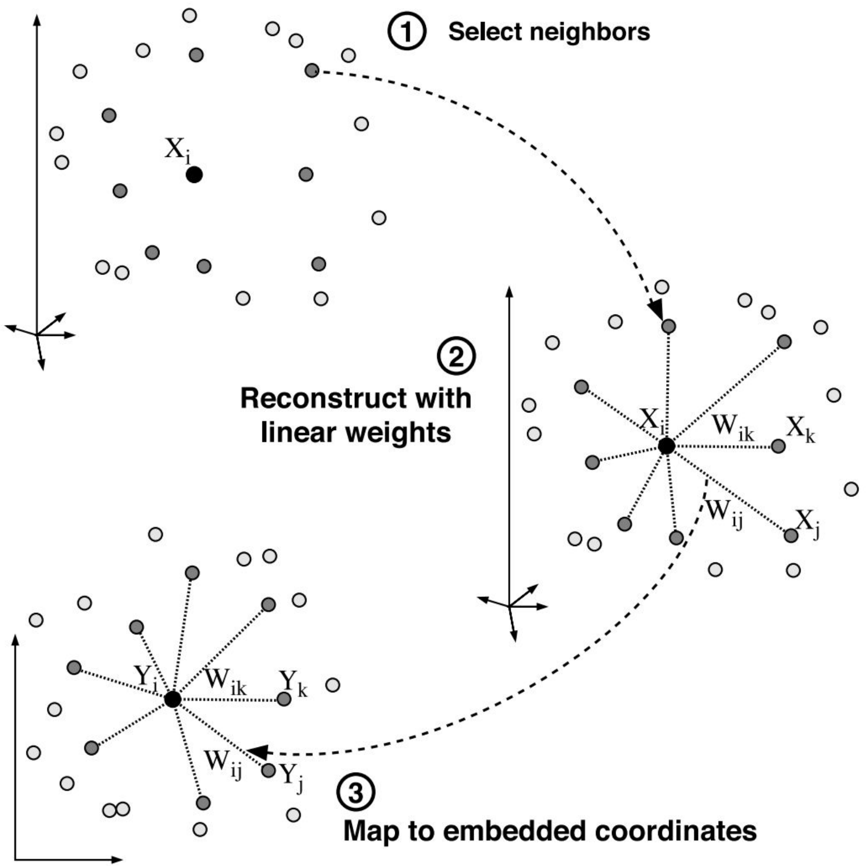
2. Estimate new points based on known relations

$$\Phi(Y) = \sum_{i=1}^{n} ||y_i - \sum_{j=1}^{n} W_{ij} y_j||^2$$

Parameters:

n - number of neighbours to connect

p - dimensionality of manifold

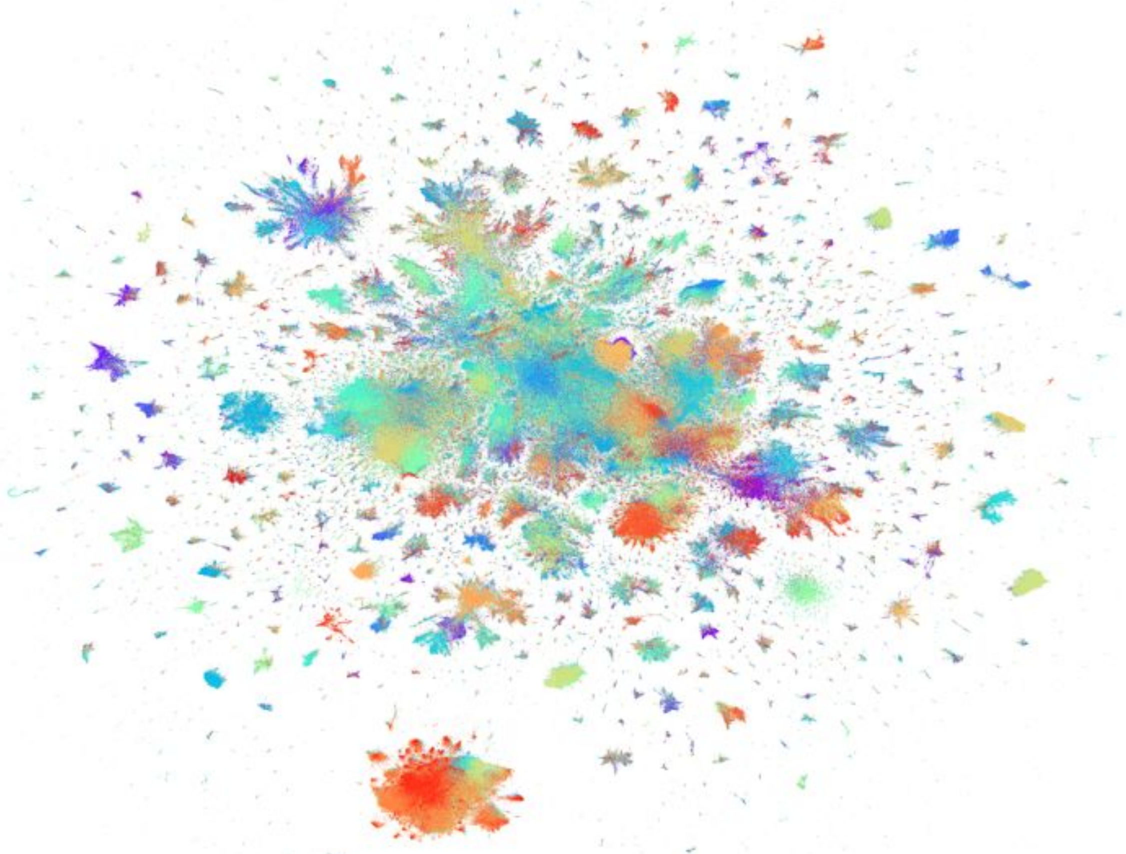


Many more...

- Hessian Eigenmapping
- Spectral Embedding
- Local Tangent Space Alignment
- Riemannian Geometry
- ……

t-distributed Stochastic Neighbor Embedding

t-SNE makes every slide 42% better (c)



Source: [Habrahabr post on t-SNE]

Stochastic Neighbor Embedding

Idea:

Convert pairwise distances to probabilities, preserve probabilities through the spaces

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

asymmetric probability of object i chooses j as its neighbour

$$q_{j|i} = \frac{\exp(-||y_i - y_j||^2)}{\sum_{k\neq i}\exp(-||y_i - y_k||^2)}$$

the same in target space

Idea: construct embedding s.t. this distributions are close.
What are close distributions?

Kullback–Leibler divergence

$$D_{KL}(P \parallel Q) = \sum_{i,j} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

Seems similar to cross-entropy, doesn’t it?

Stochastic Neighbor Embedding

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

$$q_{j|i} = \frac{\exp(-||y_i - y_j||^2)}{\sum_{k \neq i} \exp(-||y_i - y_k||^2)}$$

$$D_{KL}(P \,||\, Q) \to \min_{Y}$$

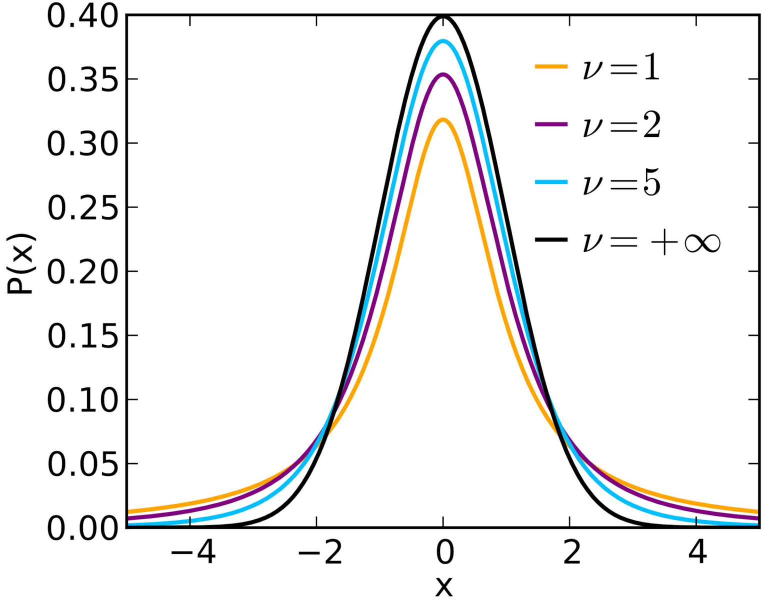
t-distributed Stochastic Neighbor Embedding

How to improve it:

1. make distribution symmetric
2. make it decrease faster than Gaussian
   (use Student's t-distribution)

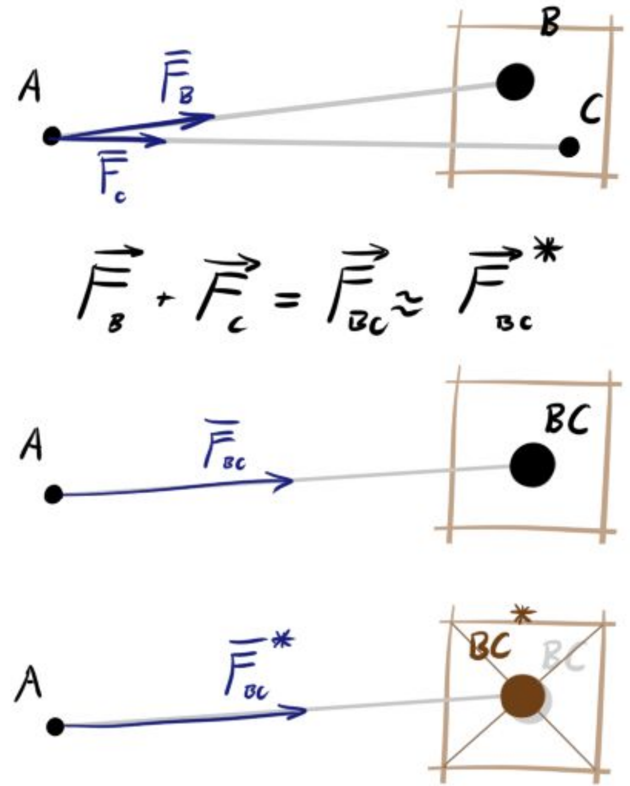
$$p_{ij} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)}{\sum_{k \neq l} \exp\left(-\|x_k - x_l\|^2 / 2\sigma^2\right)}$$

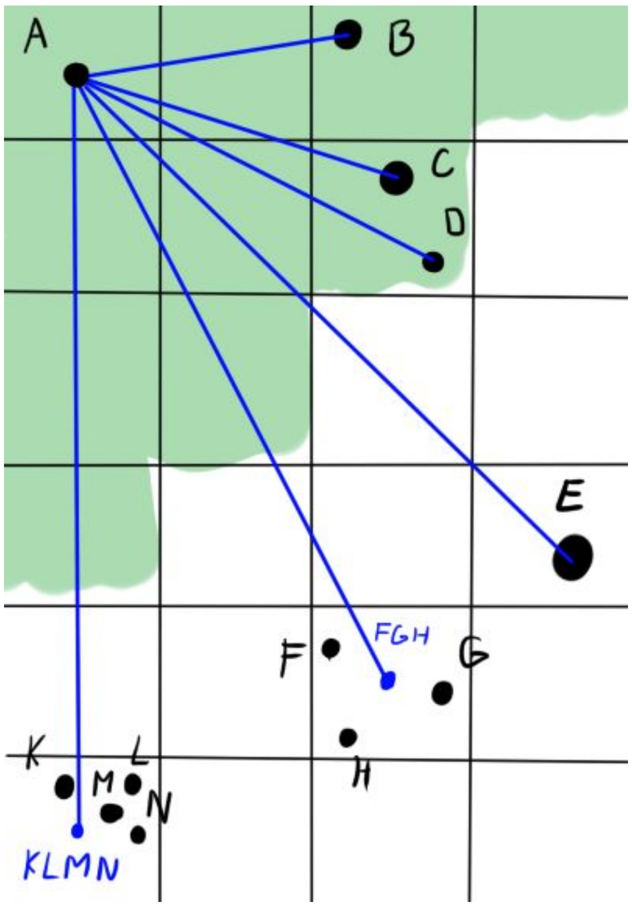
$$q_{ij} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq l} \exp\left(-\|y_k - y_l\|^2\right)}$$



Link to original paper

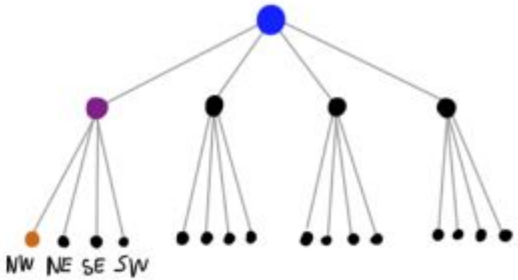
Make it faster: Barnes-Hut procedure

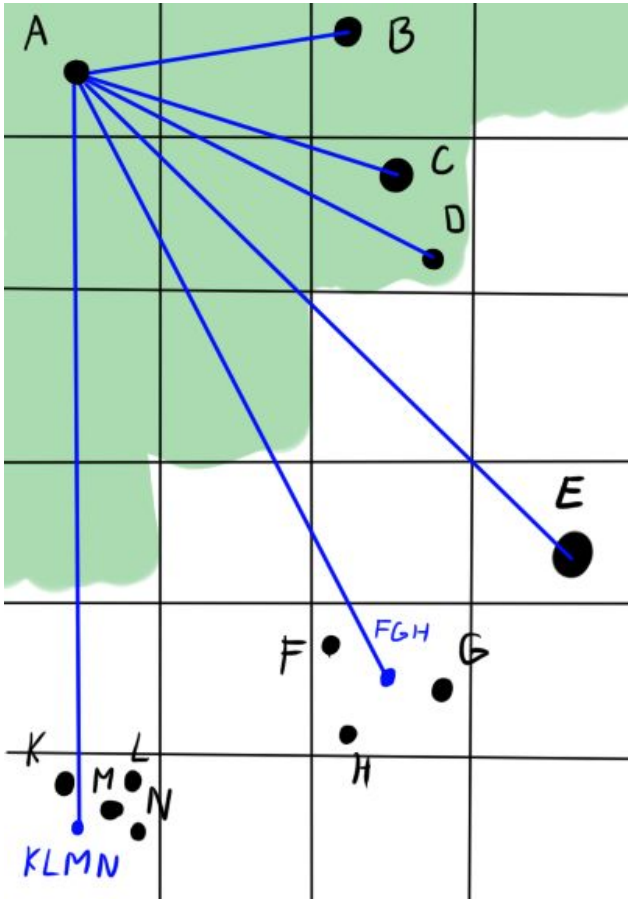




Source: Habrahabr post on t-SNE

Make it faster: Barnes-Hut procedure

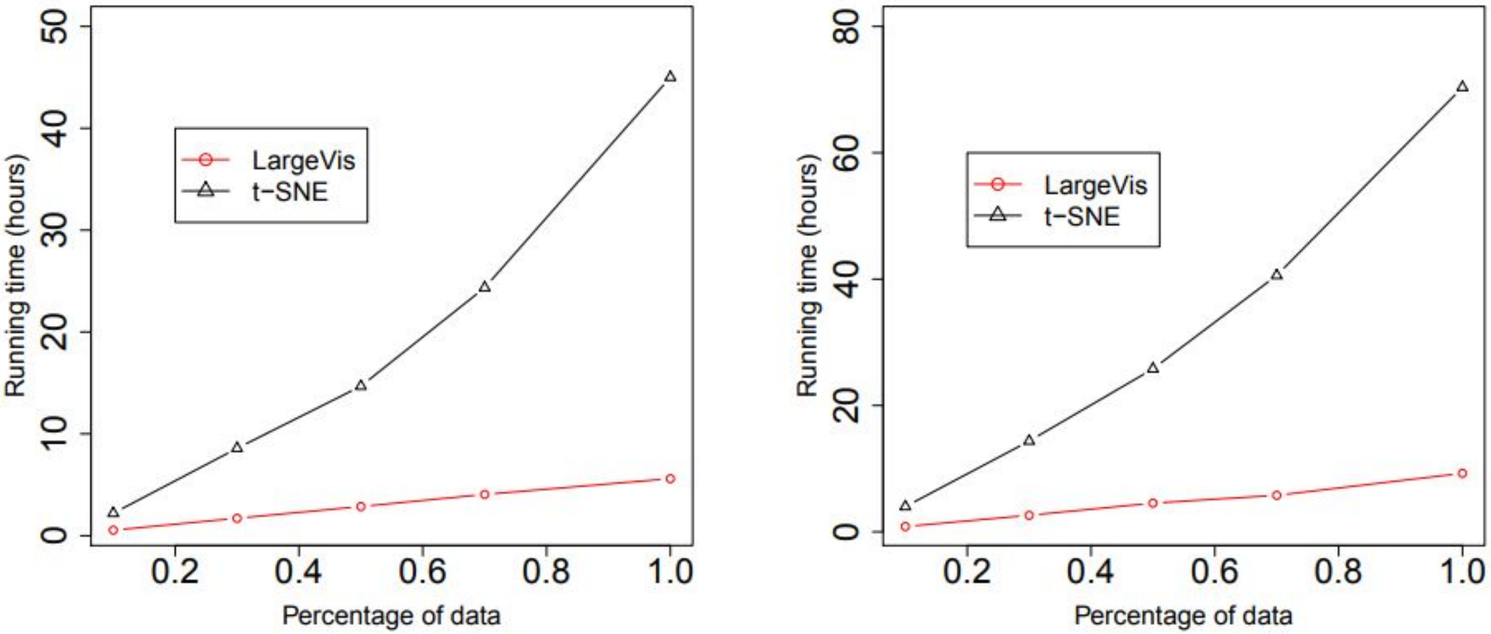




Source: [Habrahabr post on t-SNE]()

We need more speed! LargeVis

Idea: Use stochastic trees

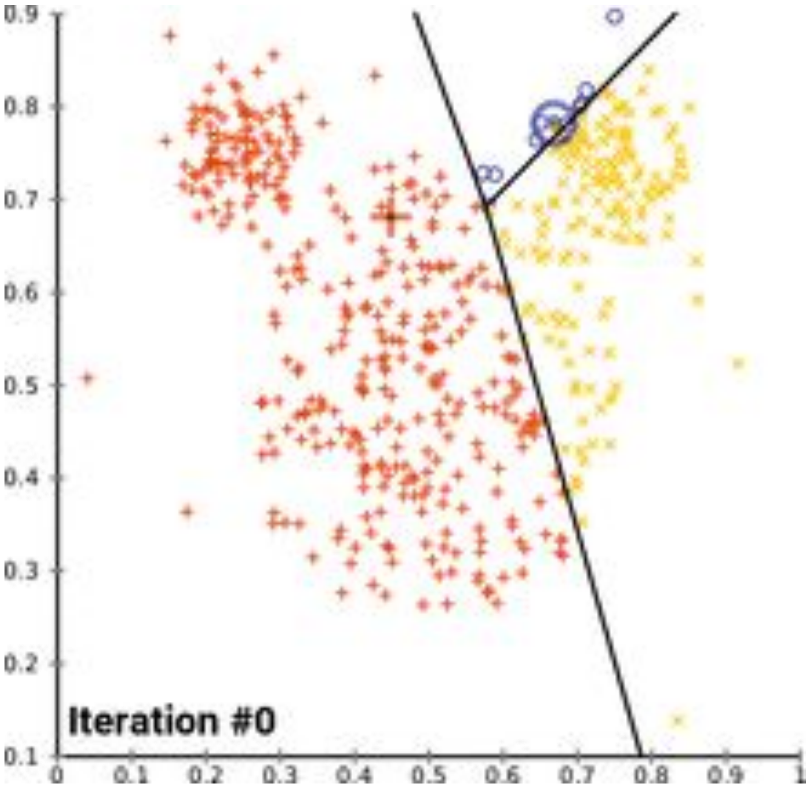


(c) Time (WikiDoc) (d) Time (LiveJournal)

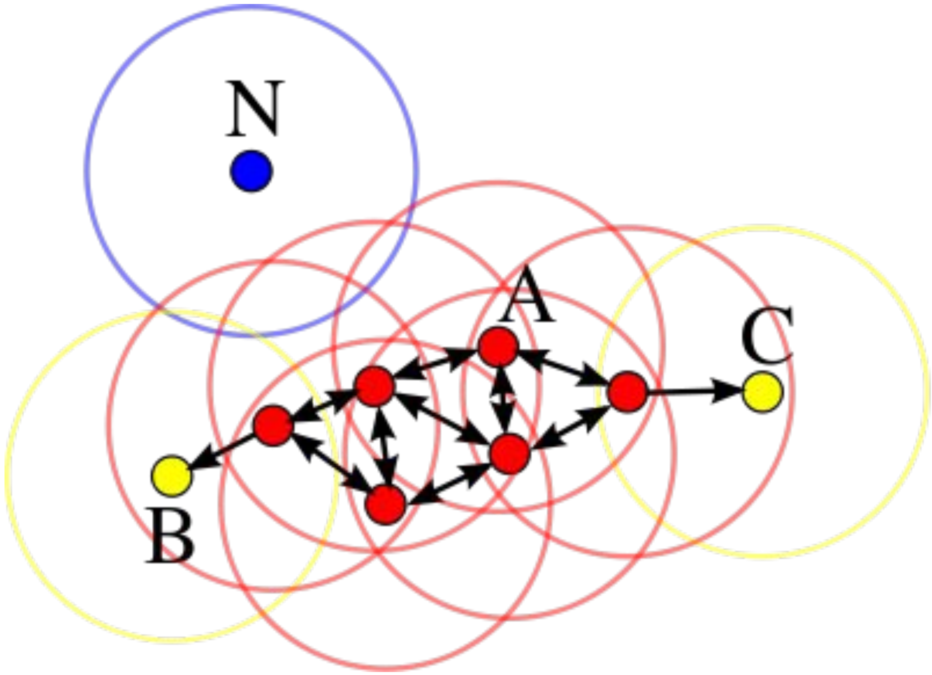
Source: [Habrahabr post on t-SNE]()

Clustering

k-means



DBSCAN



Q & A

Further readings:

1. Good lecture on MDS, Isomap, LLE
2. Lecture on t-SNE
3. Slides about clusterization
4. Metrics in clusterization
5. Slides about ICA