Unsupervised Learning

& naïve bayes

Vladislav Goncharenko
MIPT, 2019

Outline

I. Naïve Bayes
II. Unsupervised learning
    A. Maniflod assumption
    B. Dimensionality reduction
        1. Multidimensional Scaling (MDS)
        2. Isomap
        3. Locally linear embedding (LLE)
        4. t-SNE
    C. Clustering
        1. k-means
        2. DBSCAN

Naïve Bayes

Naïve Bayes

Naive assumption
of features independence
leads to simple
and easy to calculate result

$$P(y|x_1,\dots,x_n) = P(y)\cdot\frac{P(x_1,\dots,x_n|y)}{P(x_1,\dots,x_n)}$$

$$\boxed{P(x_i|y,x_1,\dots,x_{i-1},x_{i+1},\dots,x_n) = P(x_i|y)}$$

$$P(y|x_1,\dots,x_n) = P(y)\cdot\frac{\prod_i P(x_i|y)}{P(x_1,\dots,x_n)}$$

$$\boxed{P(x_1,\dots,x_n) \equiv const}$$

$$\hat{y} = \arg\max_y P(y)\cdot\prod_i P(x_i|y)$$

What $P(x_i|y)$ really is?

Typical likelihood of the features

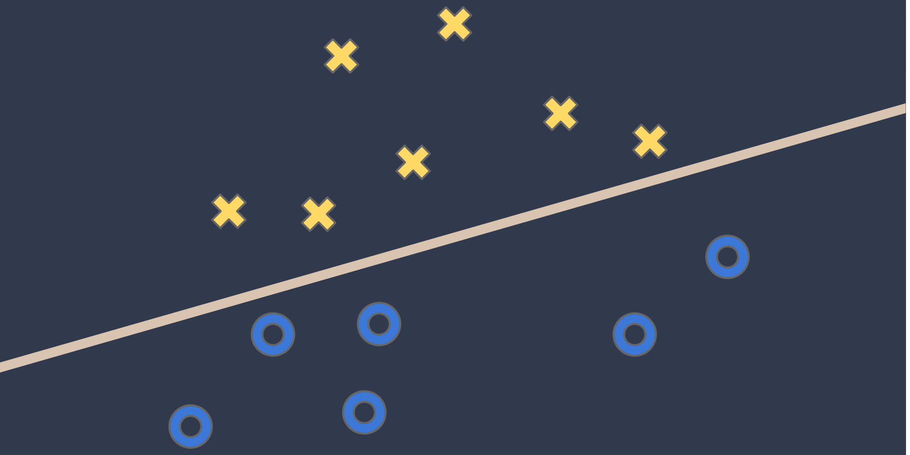
1. Gaussian
2. Multinomial
3. Bernoulli

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

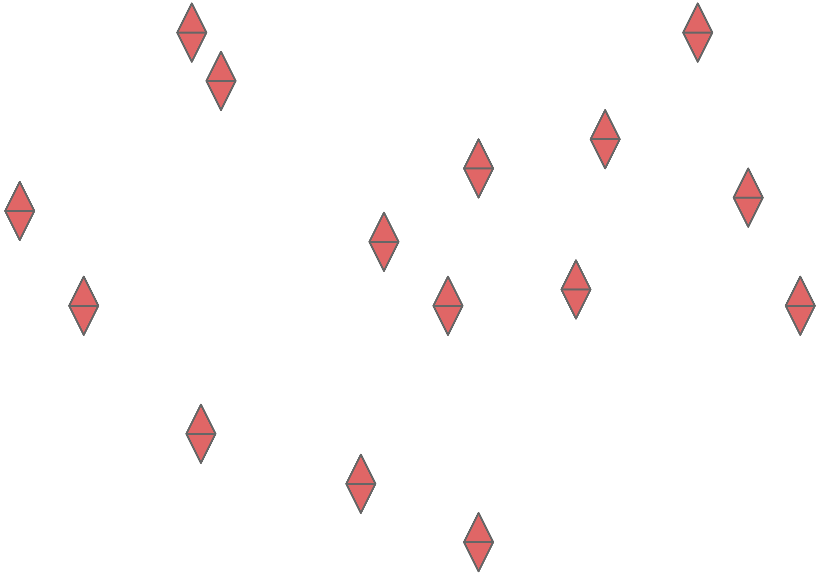
$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

Unsupervised learning

Supervised learning



Unsupervised learning

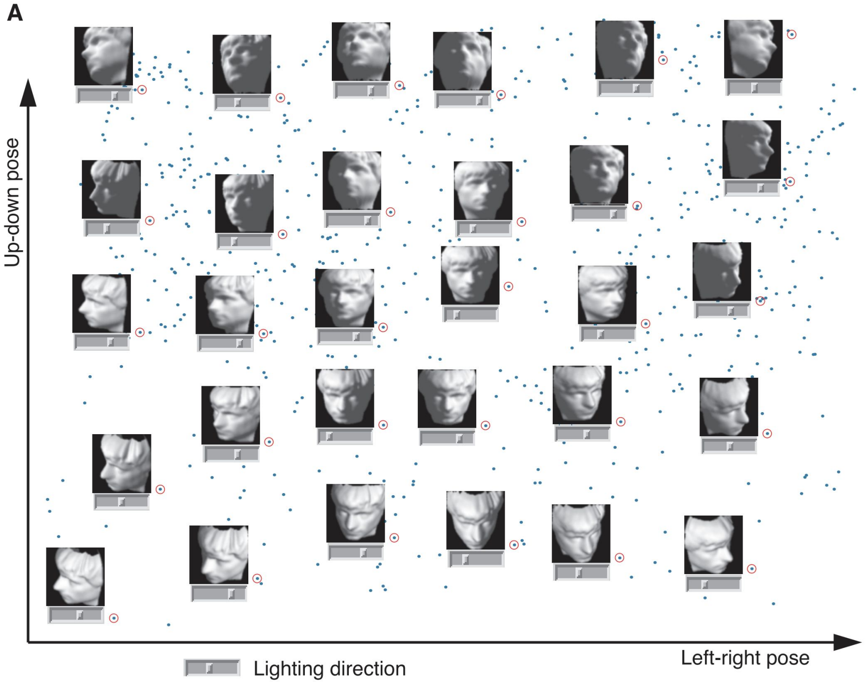


Manifold assumption

The data lie approximately
on a manifold of much lower dimension
than the input space

So problem dimensionality could be
(non-)linearly reduced

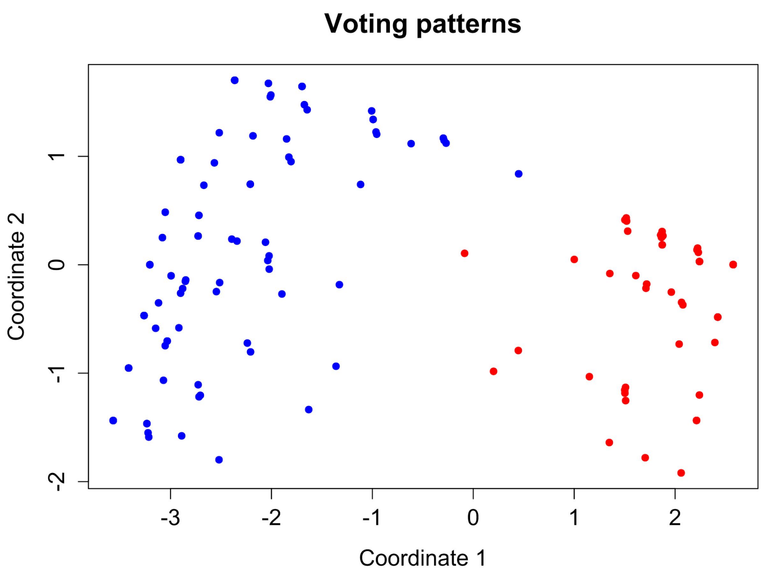
Approach doesn't require any labels



[Tenenbaum, de Silva, Langford](#)
[A Global Geometric Framework for Nonlinear Dimensionality Reduction](#)

Dimensionality reduction

Multidimensional Scaling (MDS)



Voting patterns in the United States House of Representatives

Goal:

Linearly embed to given lower space

Solution:

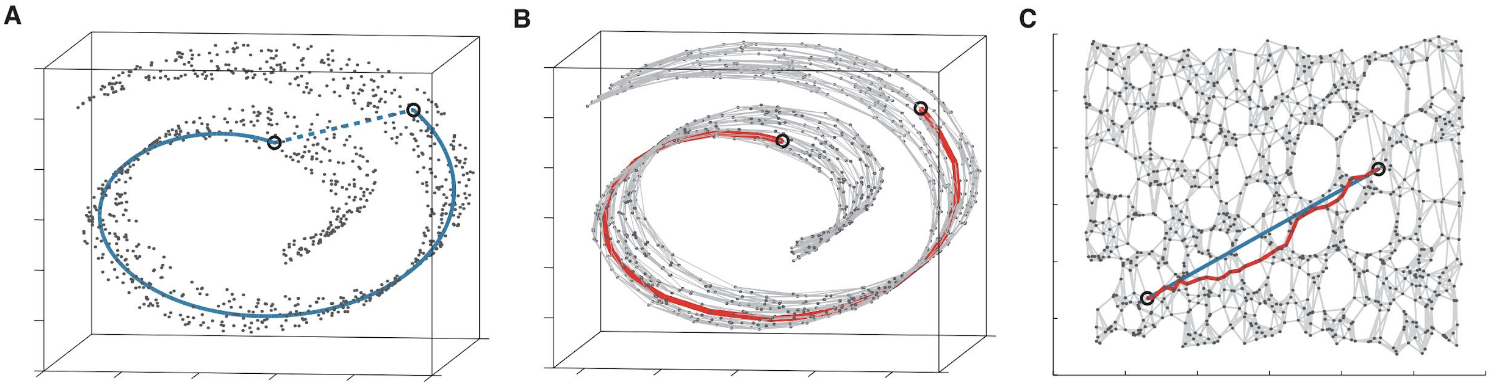
PCA

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

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

Params: p - target dimensionality

Isomap



Now make distancies geodesic!
And measure distances on the produced graph

Params:

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

[Original article]

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

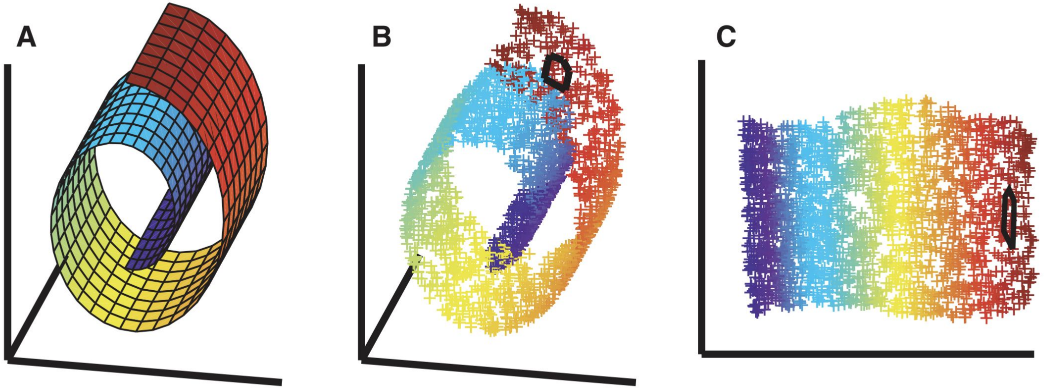
17. The operator τ is defined by $\tau(D) = -HSH/2$, where S is the matrix of squared distances $\{S_{ij} = D_{ij}^2\}$, and H is the "centering matrix" $\{H_{ij} = \delta_{ij} - 1/N\}$ (*13*).

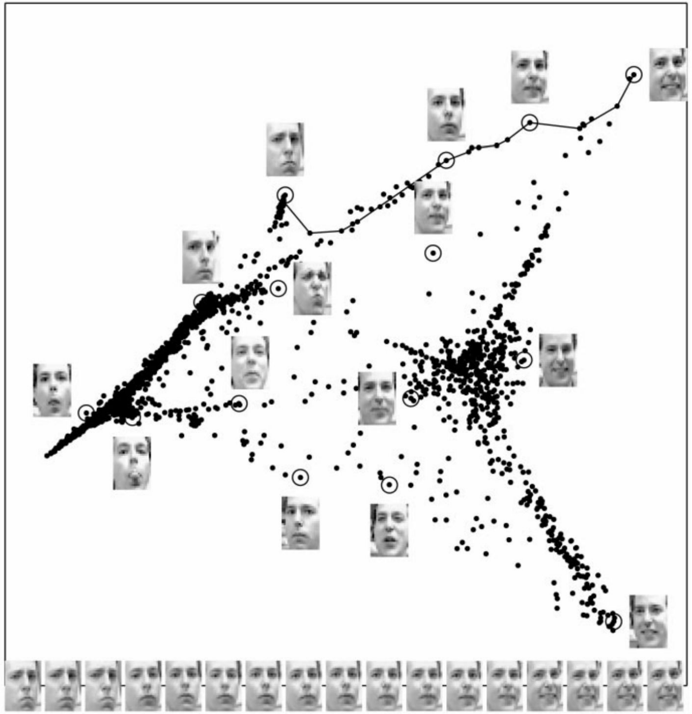
Locally linear embedding (LLE)

Idea:

Smooth manifold can be locally approximated linearly.
Linear peases can be flattened

Original article





Locally linear embedding (LLE)

Two steps of embedding and two objective functions:

1. estimate point by its K neighbours

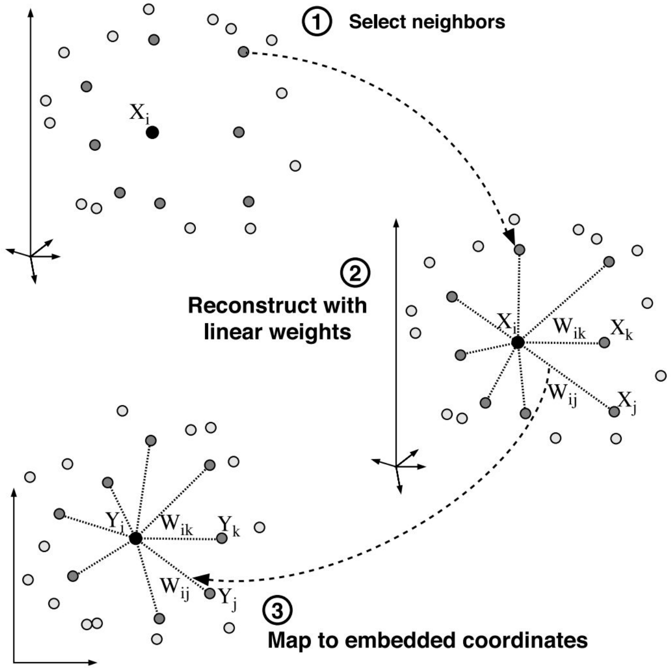
$$\varepsilon(W) = \sum_{i=1}^{n} ||x_i - \sum_{j=1}^{K} W_{ij} x_j||^2$$

2. Estimate new points based on known relations

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

Params:

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



Many more...

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

t-SNE

t-distributed Stochastic Neighbor Embedding

$\underbrace{\text{Stochastic Neighbor Embedding}}_{\text{SNE}}$

SNE

original article

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

Let’s construct embedding s.t. this distributions are close.
What are close distributions?

Kullback–Leibler divergence

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

Suspiciously similar to Shannon entropy

Learn more

Stochastic Neighbor Embedding

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

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

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

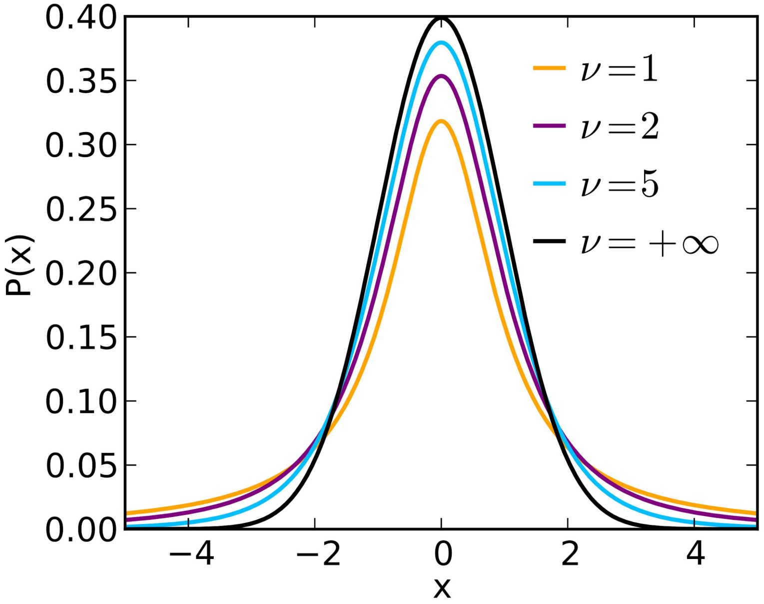
t-distributed Stochastic Neighbor Embedding

Patches over SNE:

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

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

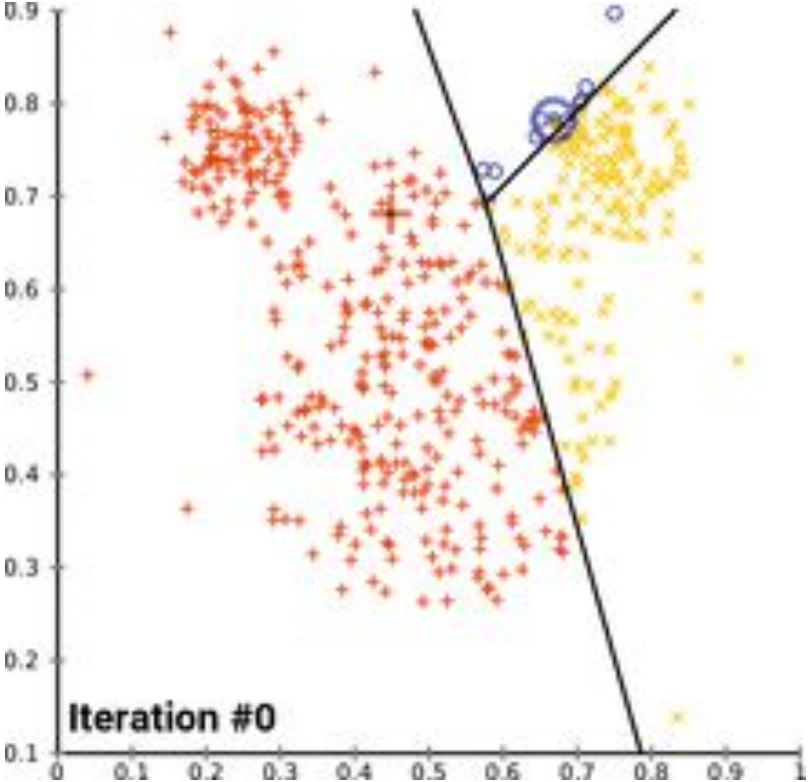
$$q_{ij} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq l} \exp\left(-\|y_k - y_l\|^2\right)}$$



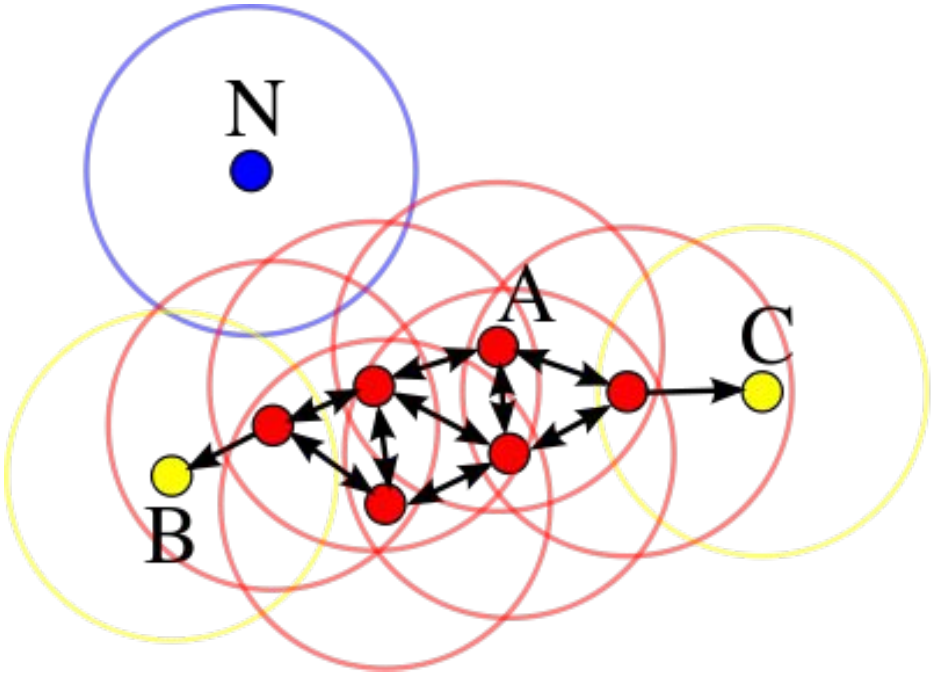
Original article

Clustering

k-means



DBSCAN



Links

1. Good lecture on MDS, Isomap, LLE
2. Lecture on t-SNE (this one is good too)
3. Slides about clusterization
4. Metrics in clusterization
5. Slides about ICA
6. More clustering methods (in russian)