Dimensionality reduction techniques: Part 2

Sanzhar Askaruly (San)

Ulsan National Institute of Science and Technology
Ph.D. Candidate in Biomedical Engineering

CodeSeoul MLA
December 24, 2022

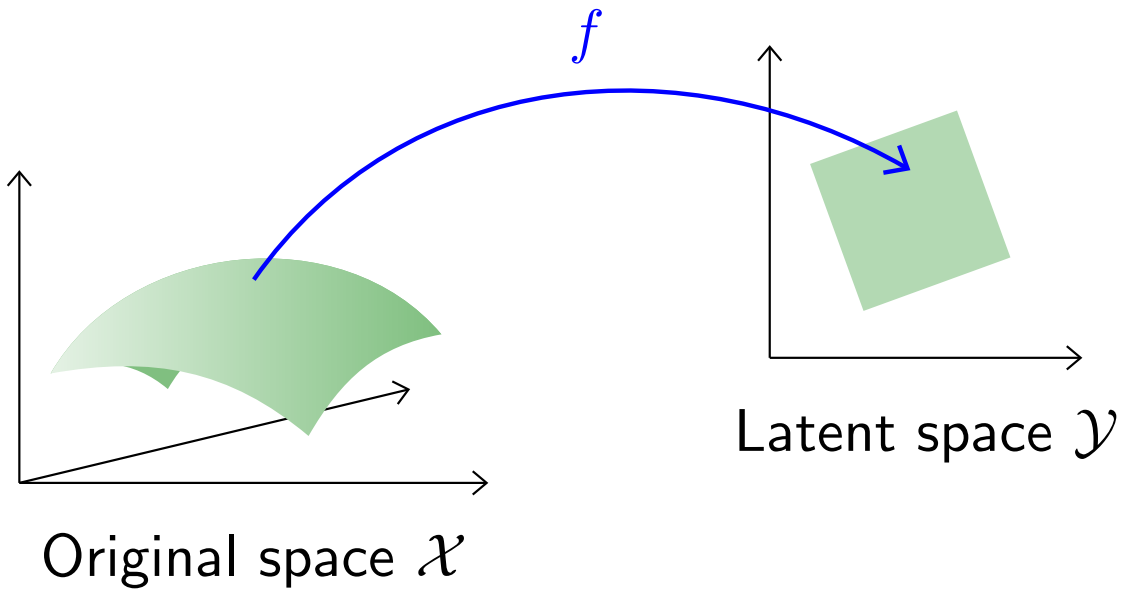
Last time...

1. Motivation
2. Background mathematics
   - Population vs sample
   - Mean, standard deviation, variance
   - Covariance, covariance matrix
   - Eigenvectors, eigenvalues
3. Dimensionality reduction techniques
   - PCA (Principal component analysis)
     - Example with IRIS dataset
     - PCA under the hood
   - t-SNE (Stochastic neighbor embedding)
4. Further
   - Lecture contents
   - References

Today

1. Problem with linear methods
2. Non-linear methods (Manifold learning)
   - SNE
   - t-SNE
   - Isomap
3. Further
   - Lecture contents
   - References

Dimensionality reduction

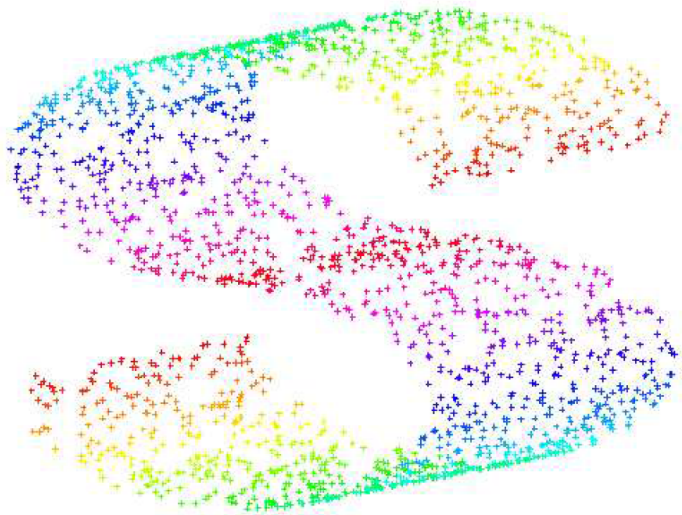


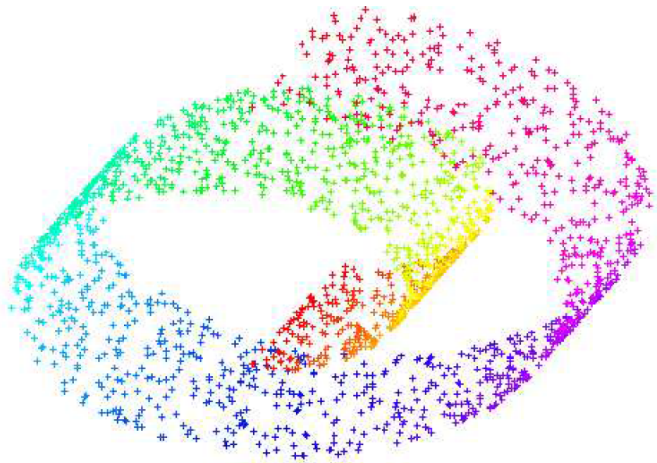
$$X = \{x_1, x_2, ..., x_n \in \mathbb{R}^{high}\} \rightarrow Y = \{y_1, y_2, ..., y_n \in \mathbb{R}^{low}\}$$

$$\min_{Y} C(X, Y)$$

Dimensionality reduction

Let's consider a **manifold**[1] embedded nonlinearly in high-dim. space:

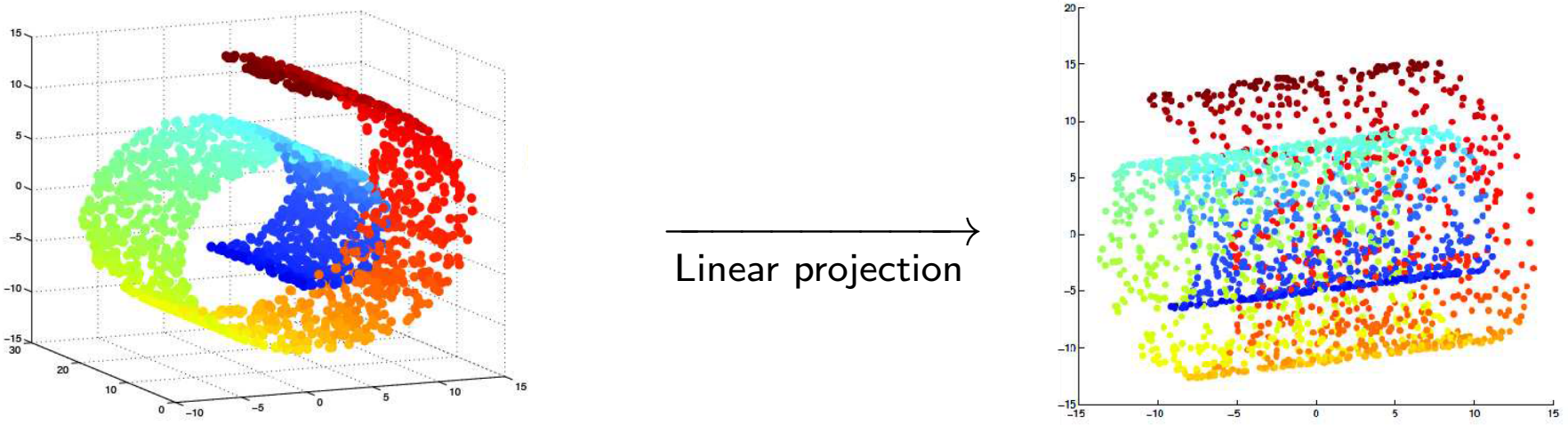




[1] a generalization of a curved surface; it is a topological space that is modeled closely on Euclidean space locally but may vary widely in global properties.

Dimensionality reduction

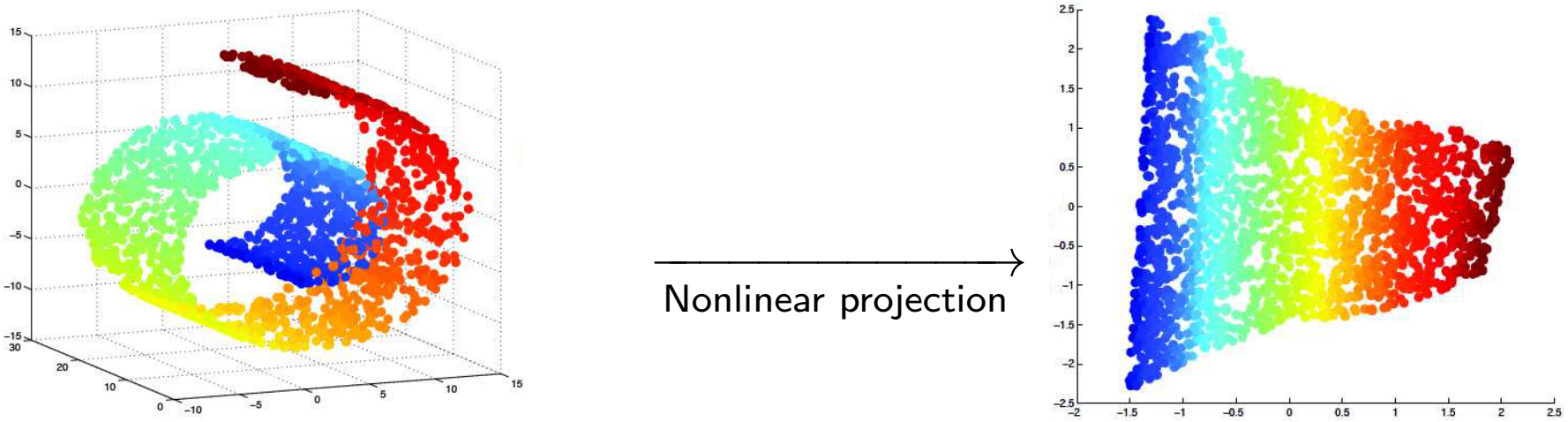
- Linear projection may not be good enough..



- Linear projection methods (*eg.* PCA) can't capture intrinsic nonlinearities

Dimensionality reduction

- Different criteria could be used for such projections



- Preserve neighborhood information
  - Locally linear structures
  - Pairwise distances

SNE: Idea

How similar is datapoint x_j to datapoint x_i in a high-dimesional space?

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

SNE uses Euclidian distance to estimate the probability (according to Gaussian) of similarity between neighbor datapoints x_i and x_j

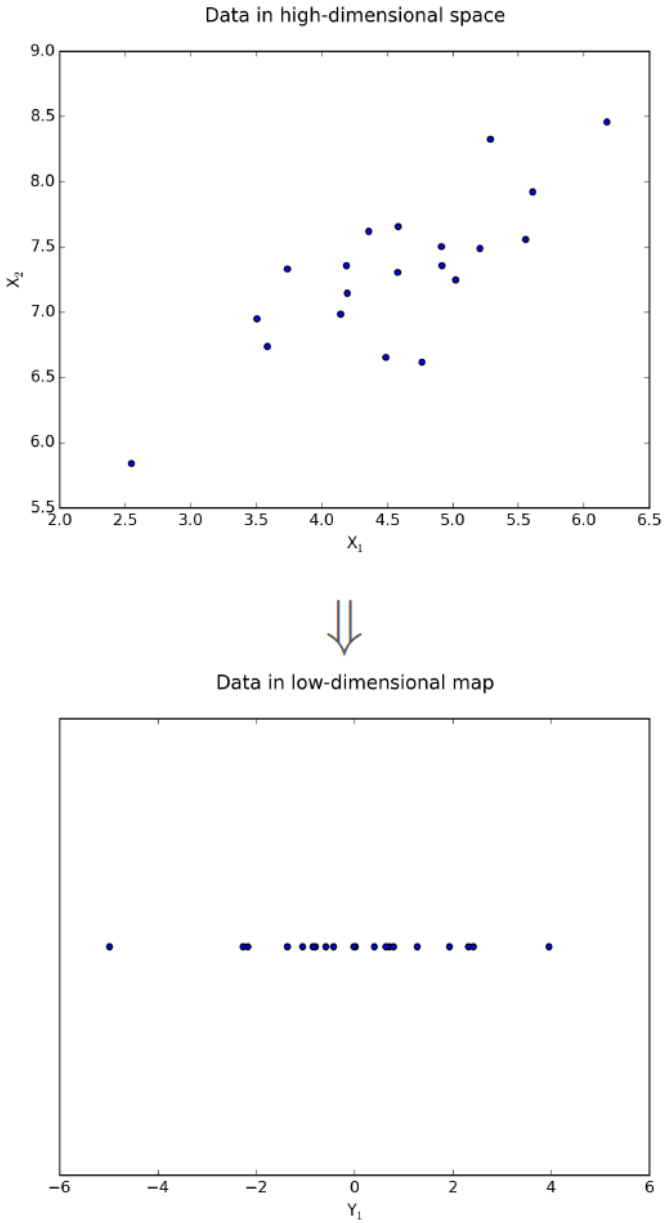
SNE: Idea

How similar is datapoint y_j to datapoint y_i in a low-dimesional space?

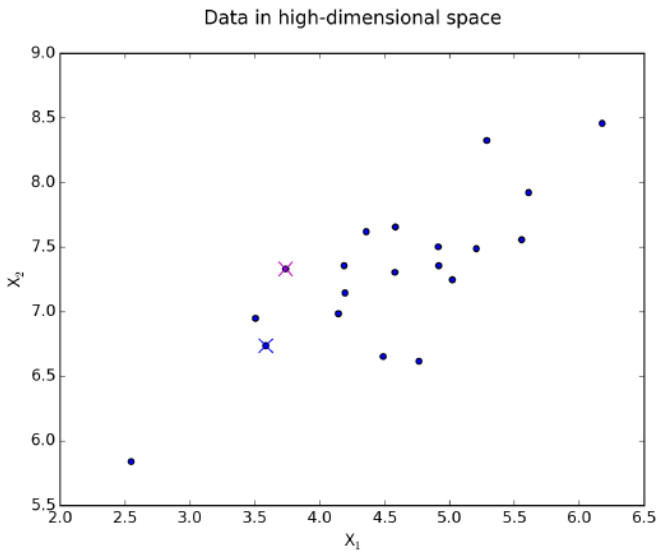
$$q_{j|i} = \frac{exp(-||y^{(i)} - y^{(j)}||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||y^{(i)} - y^{(k)}||^2/2\sigma_i^2)}$$

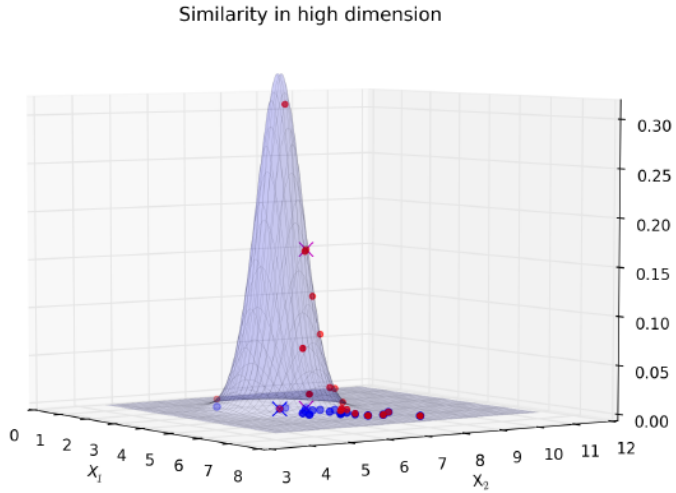
Similarly to HD, SNE uses Euclidian distance to estimate the probability of similarity between neighbor datapoints y_i and y_j

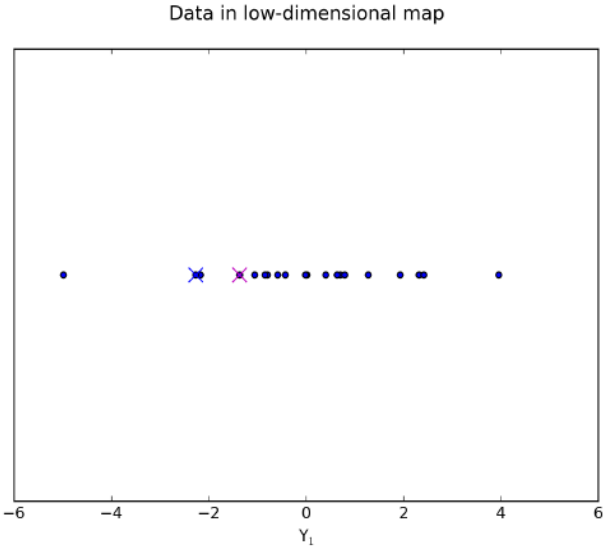
SNE: Example

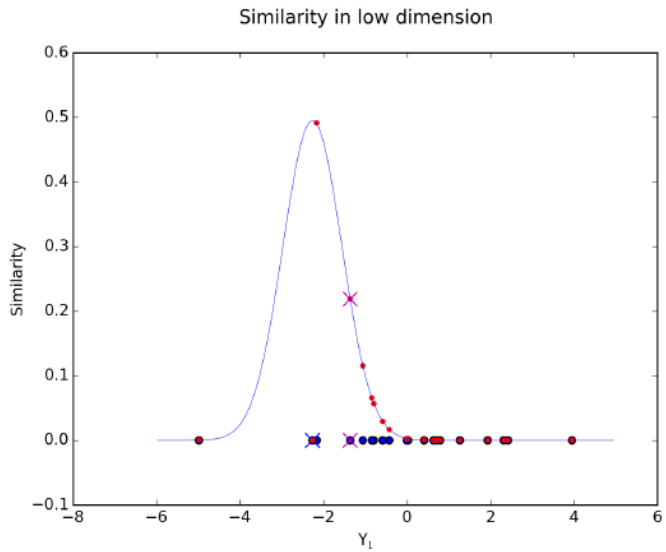

Data in high-dimensional space
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
X_2
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
X_1
Data in low-dimensional map
−6
−4
−2
0
2
4
6
Y_1

SNE: Example

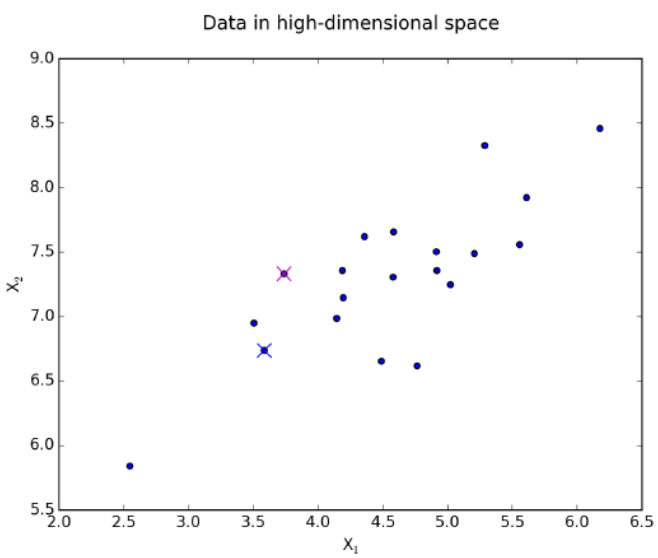

Data in high-dimensional space
X_1
X_2

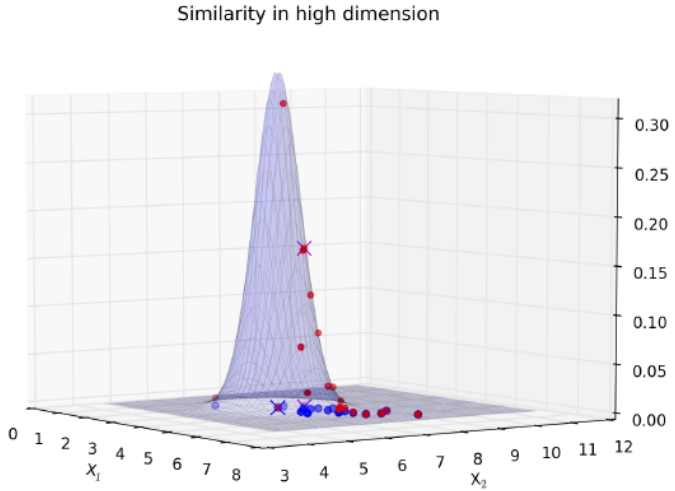

Similarity in high dimension
X_1
X_2

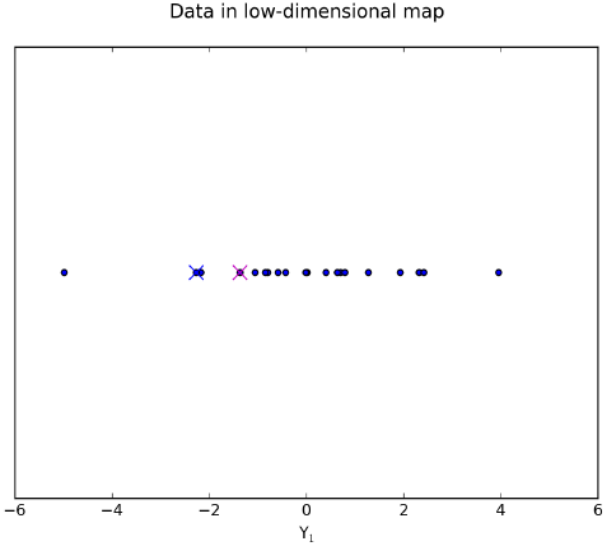

Data in low-dimensional map
Y_1

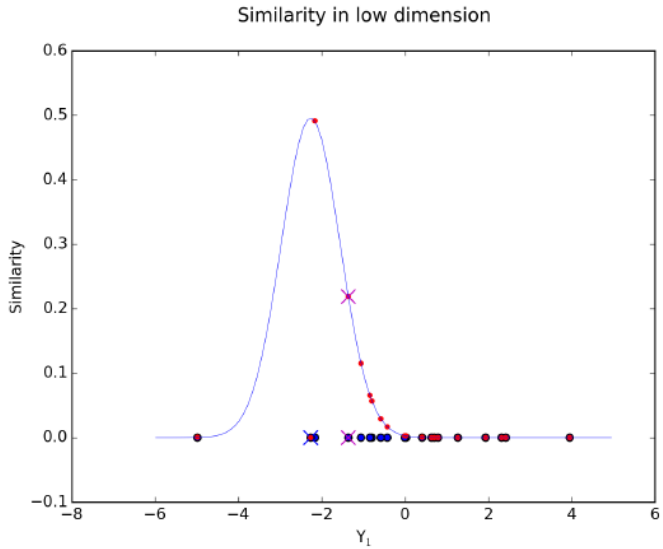

Similarity in low dimension
Similarity
Y_1

SNE: Example

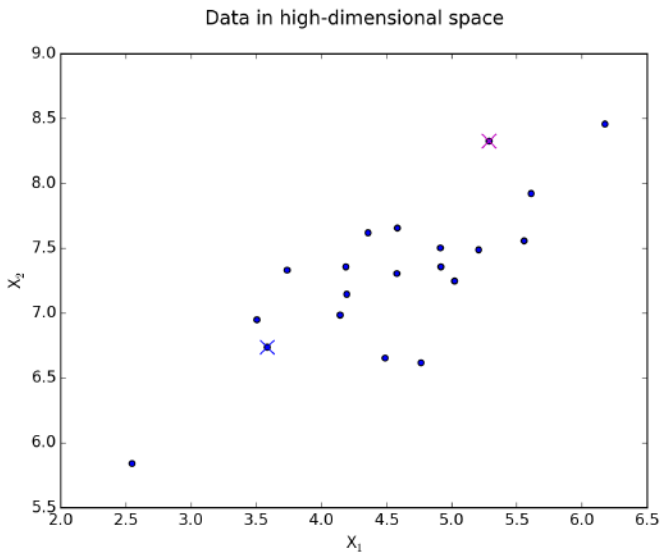

Data in high-dimensional space
X_1
X_2

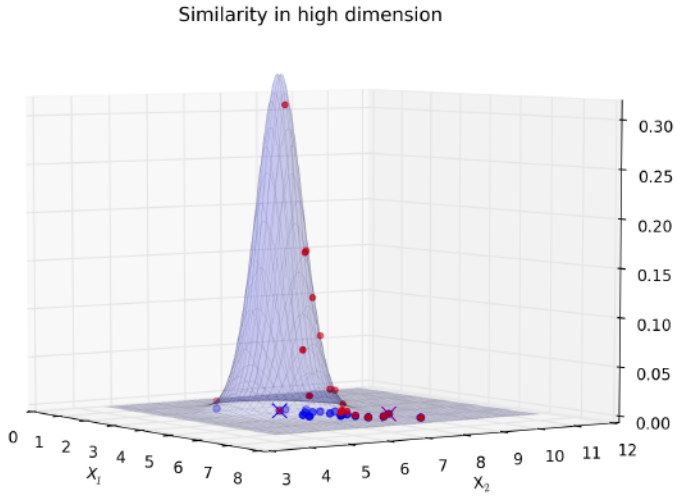

Similarity in high dimension
X_1
X_2

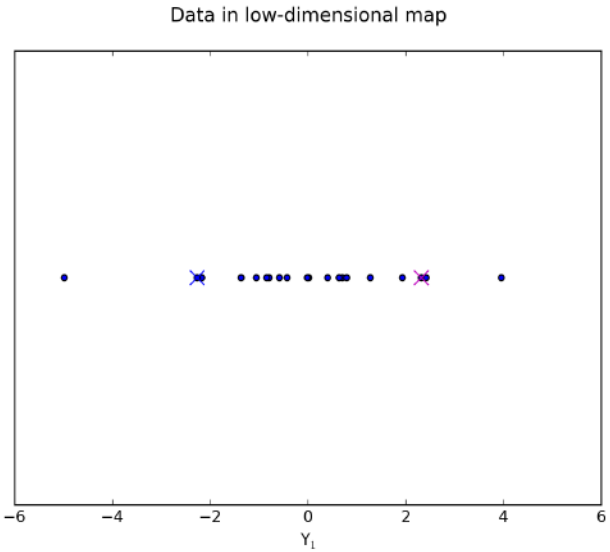

Data in low-dimensional map
Y_1

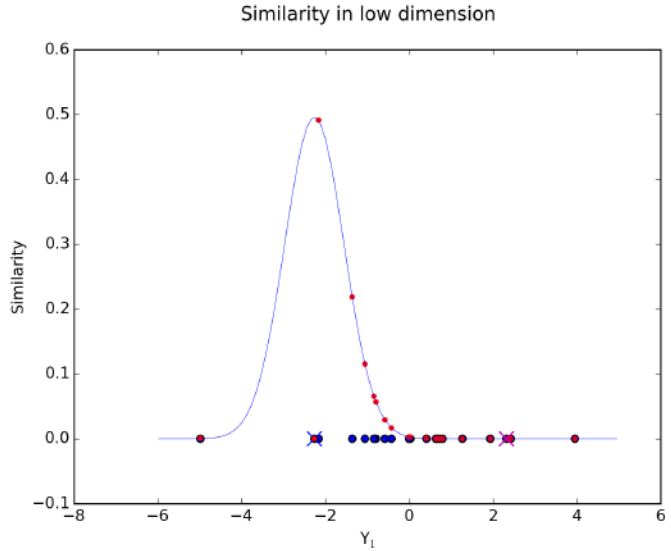

Similarity in low dimension
Similarity
Y_1

SNE: Example

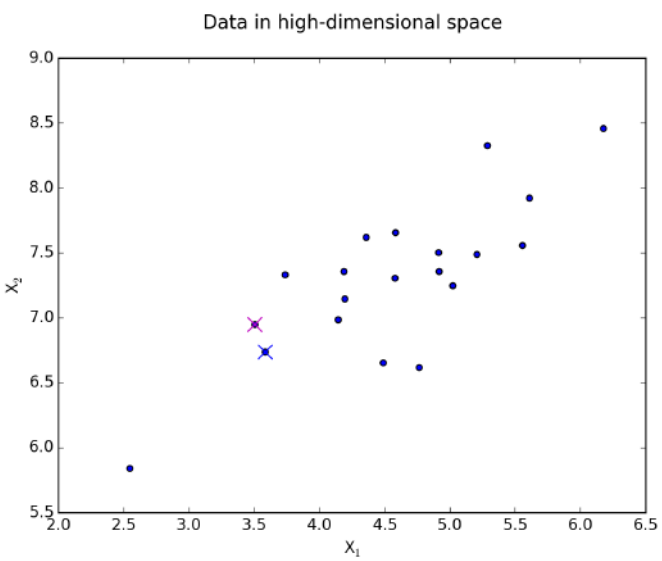

Data in high-dimensional space
X_1
X_2

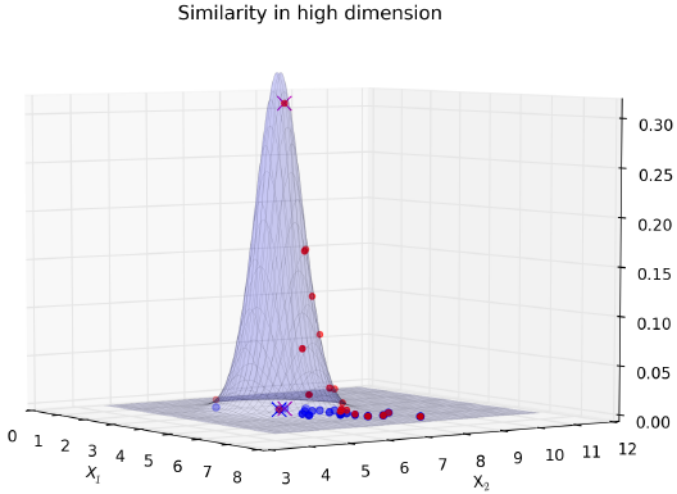

Similarity in high dimension
X_1
X_2

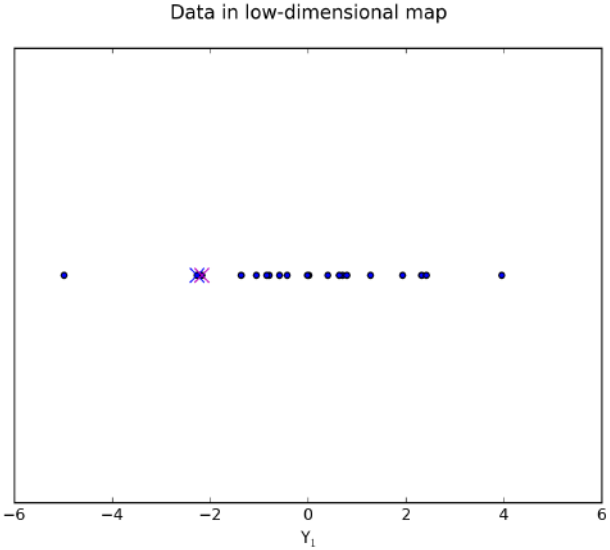

Data in low-dimensional map
Y_1

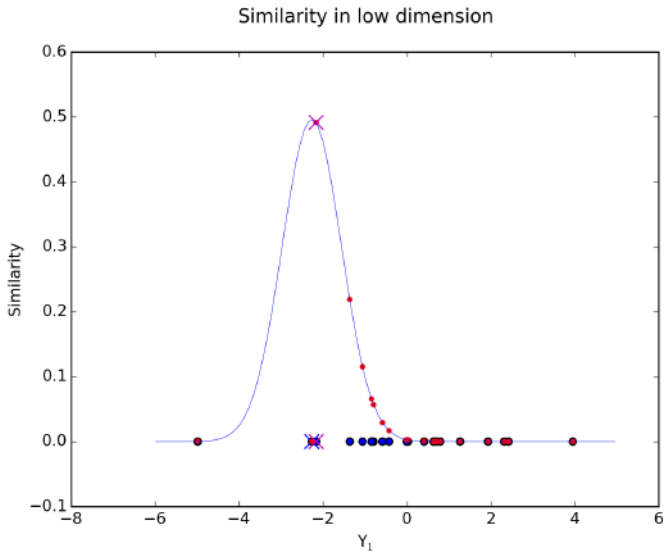

Similarity in low dimension
Similarity
Y_1

SNE: Example


Data in high-dimensional space
X_1
X_2


Similarity in high dimension
X_1
X_2


Data in low-dimensional map
Y_1


Similarity in low dimension
Similarity
Y_1

SNE: Principle

- KL divergence compares the distributions on the neighbors:

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

- Minimization problem $\min_y C(X, Y)$
- Assymetric
- Always positive

SNE: Principle

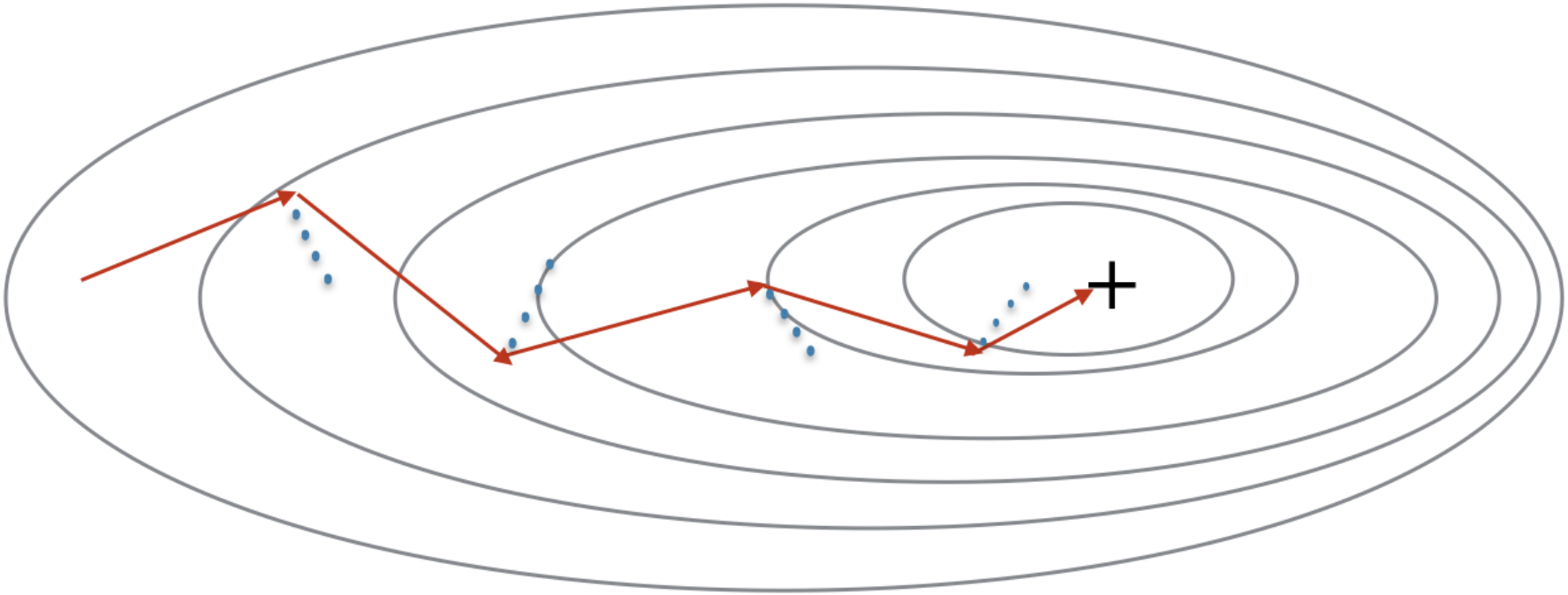
- KL divergence compares the distributions on the neighbors:

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

Minimization of this cost function is performed using gradient descent:

$$\frac{\delta C}{\delta y_i} = 2 \sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j)$$

SNE: Principle



Mathematically, the gradient update with a momentum term is given by:

$$Y^{(t)} = Y^{(t-1)} + \mu \frac{\delta C}{\delta Y} + \alpha(t) \left(Y^{(t-1)} - Y^{(t-2)} \right)$$

Crowding problem

tSNE

How similar is datapoint y_j to datapoint y_i in a low-dimesional space?

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

In low-dimension, the Gaussian distribution is replaced by student T-distribution

SNE: Principle

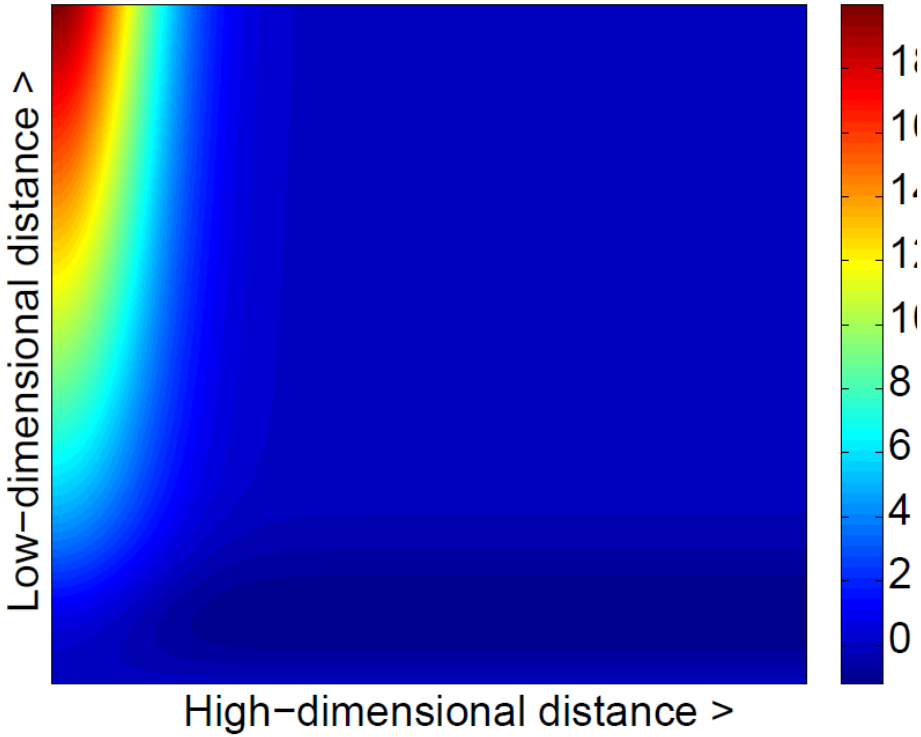
- KL divergence compares the distributions on the neighbors:

$$C = \sum_i KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ji}}$$

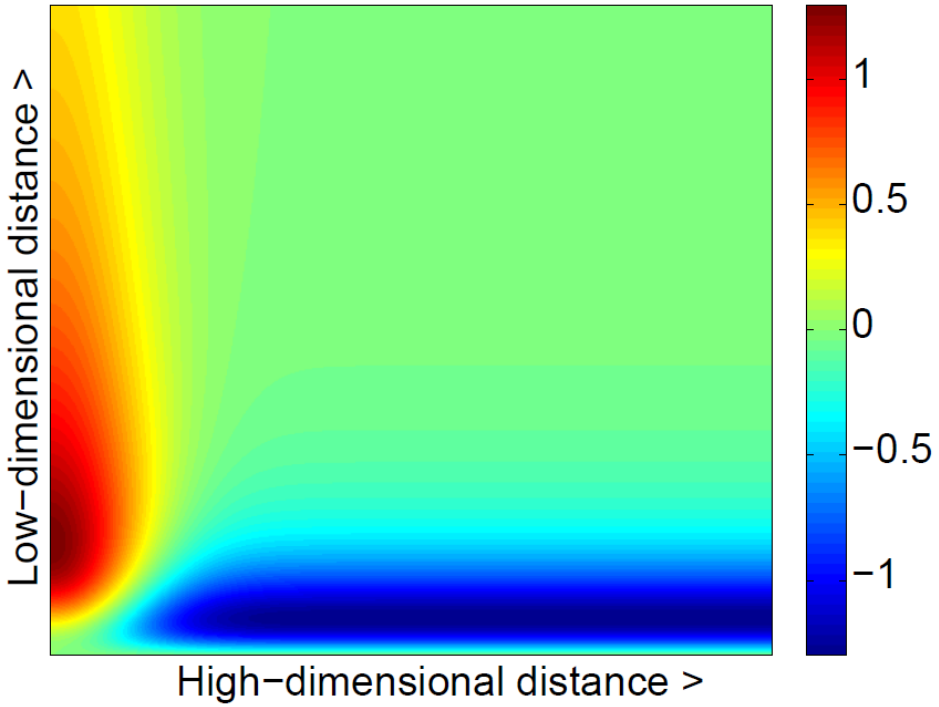
Minimization of this cost function is performed using gradient descent:

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ji})(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1}$$

tSNE



(a) Gradient of SNE.



(c) Gradient of t-SNE.

Algorithm

Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: data set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$.
begin
  compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
  set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
  sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
  for t=1 **to** T **do**
    compute low-dimensional affinities q_{ij} (using Equation 4)
    compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t)\left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
  end
end

https://www.andrew.cmu.edu/user/georgech/95-865/

Thank you for your attention!