Dimensionality Reduction

t-Distributed Stochastic Neighbor Embedding (t-SNE)

Where does t-SNE fit in?

- Dimensionality reduction algorithms:
  - Map high-dimensional data to a lower dimension
  - While preserving structure
- They are used for
  - Visualization
  - Performance
  - Curse of dimensionality
- A ton of algorithms exist
- t-SNE is specialized for visualization

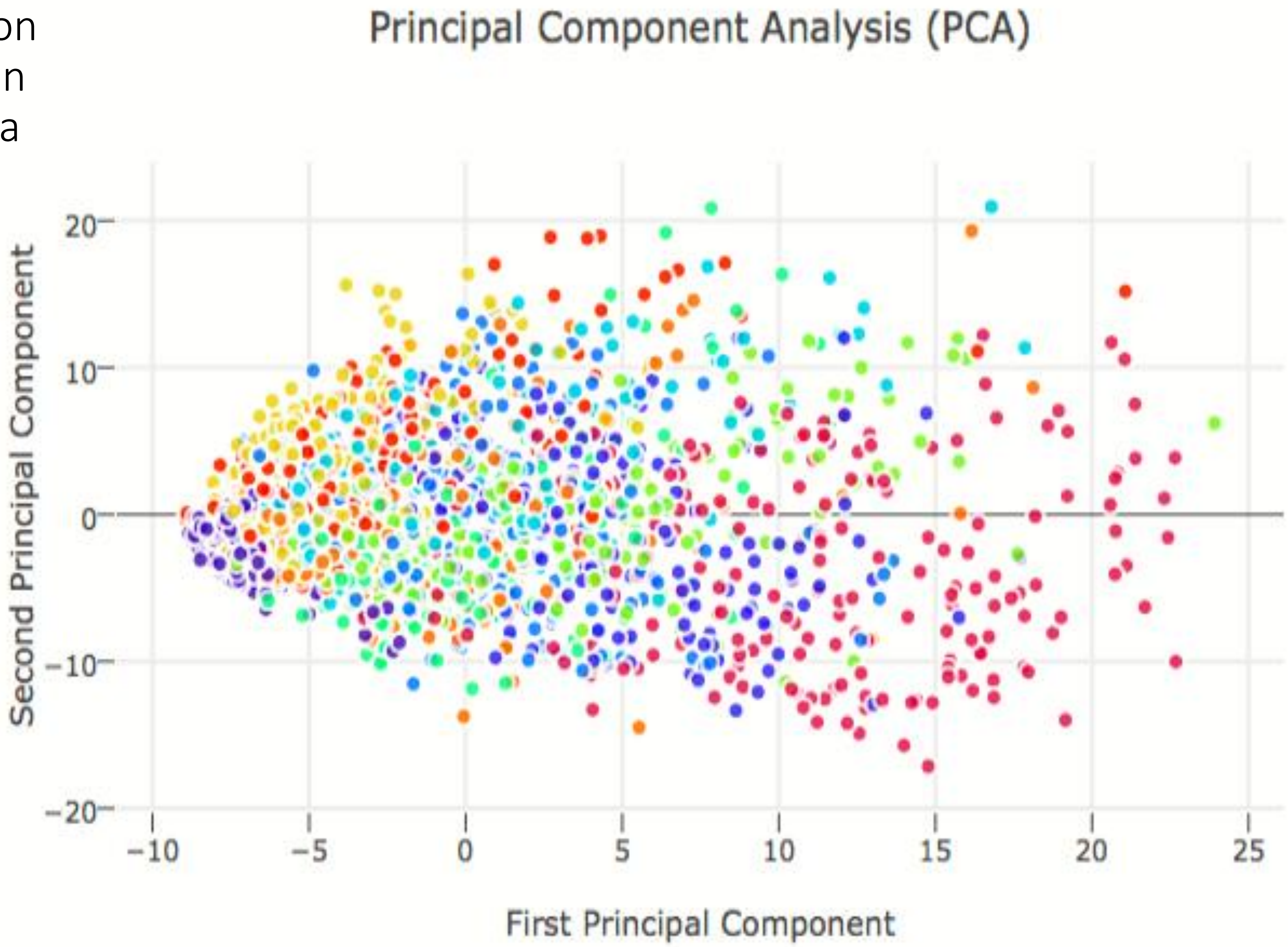
t-Stochastic neighbor embedding (t-SNE)

- t-Distributed Stochastic Neighbor Embedding is a non-linear dimensionality reduction algorithm used for exploring high-dimensional data.
- It maps multi-dimensional data to 2- or 3-dimensions for visualization.
- Better than existing techniques at creating a single map that reveals structure at many different scales.
- t-SNE is distance-based dimensionality approach, but also tends to preserve topology

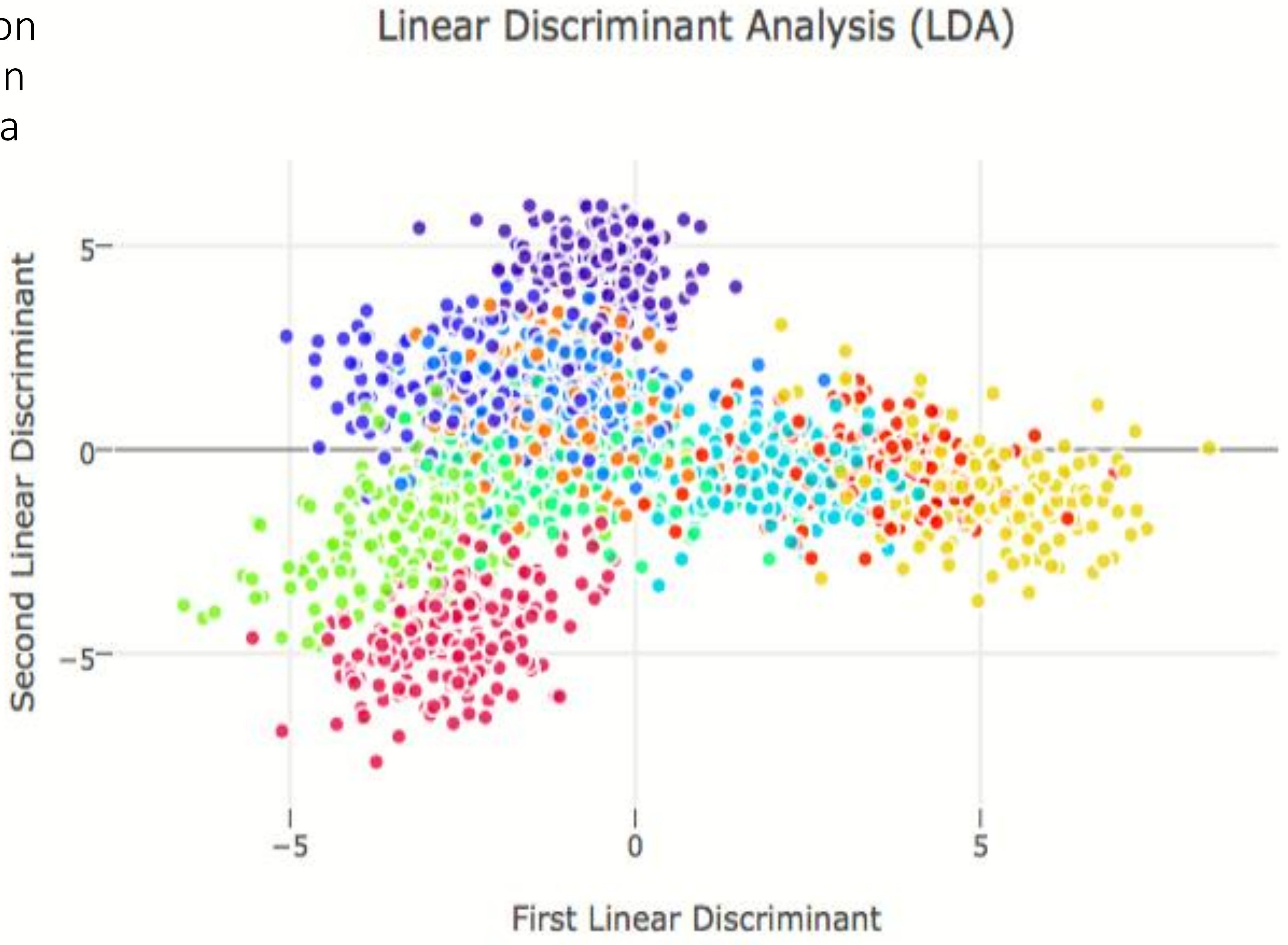
PCA, LDA, t-SNE example comparison

- PCA, LDA, and t-SNE were all applied to the MNIST handwritten digit data
  - The data is original in 783 dimensions
  - The following plots are 2 dimensions

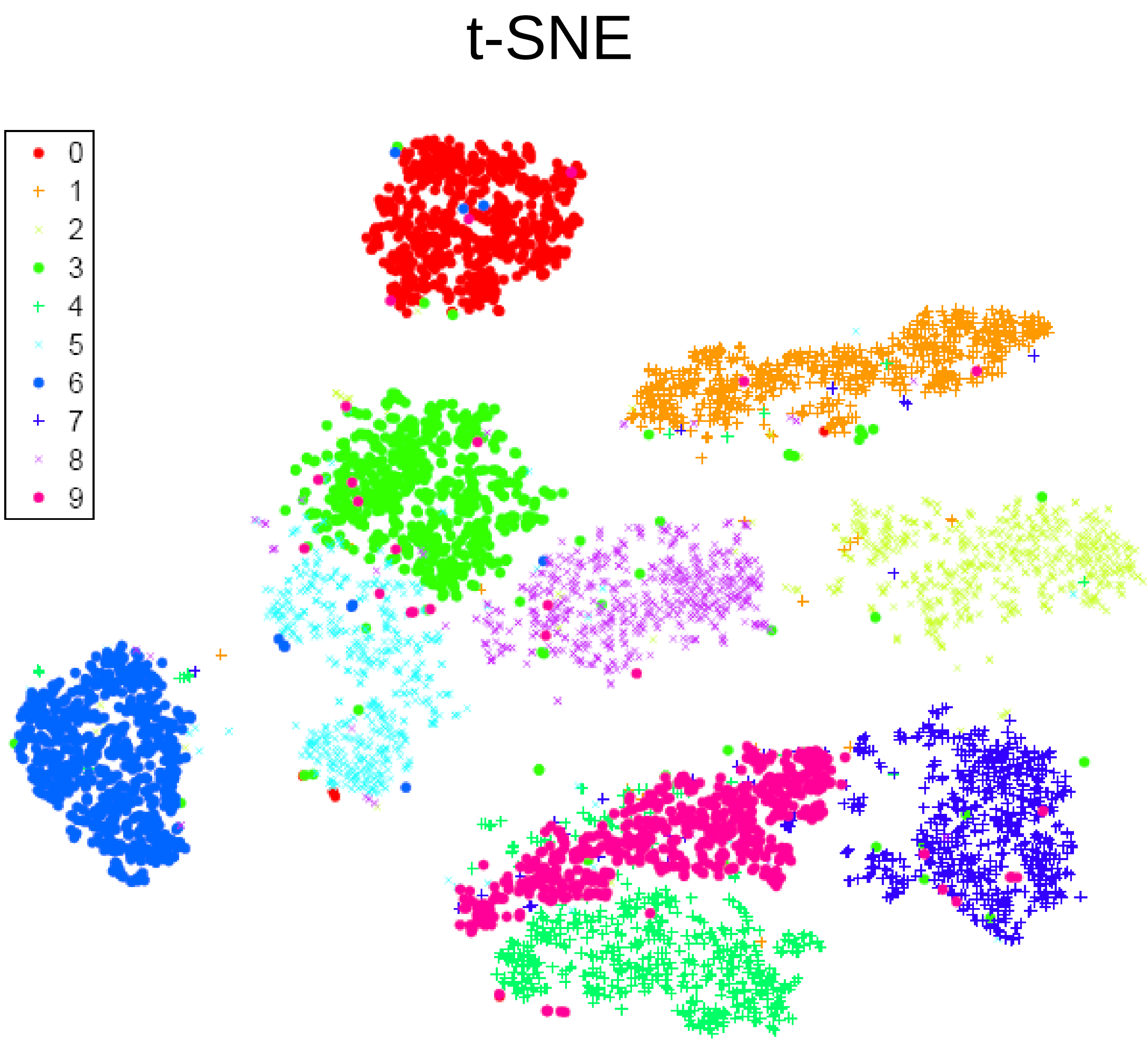
Visualization of classes in MNIST data



Visualization of classes in MNIST data



Visualization of classes in MNIST data



Stochastic Neighbor Embedding

- t-SNE is a variant of SNE – which is computationally faster
- We’ll start with SNE however:
  - it is easier to explain
  - the t-SNE improvements will not make sense unless we get the basic idea first

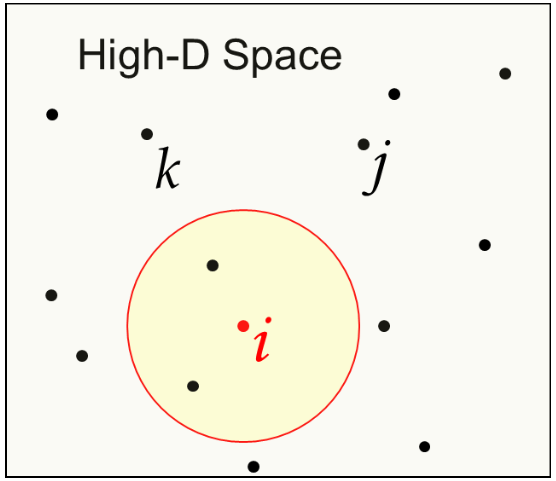
SNE summary idea

- Computes pairwise distances between points in the original high-dimensional space
  - the distance function (actually a similarity measure) is NOT Euclidean, but based on Gaussian probability assumptions
  - points close together have a value close 1; points far apart have values close to 0
- Attempts to place the points on the low-dimensional space (usually 2d or 3d space) so that they have the same pairwise distances on the lower dimensional space
- It is an iterative and stochastic technique
  - Not deterministic like PCA (every run can produce different results)
  - Not as fast as PCA
- Note: The results are not “interpretable”
  - There are no weights or loadings like PCA or LDA, only transformed points

SNE more details…

- Convert Euclidean distances between high-dimensional data points into **conditional probabilities** that represent **similarity**
- Similarity of point x_i with x_j is the conditional probability, p_{ji}, that x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at x_i.

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



Probabilistic similarity example

Euclidean Distances

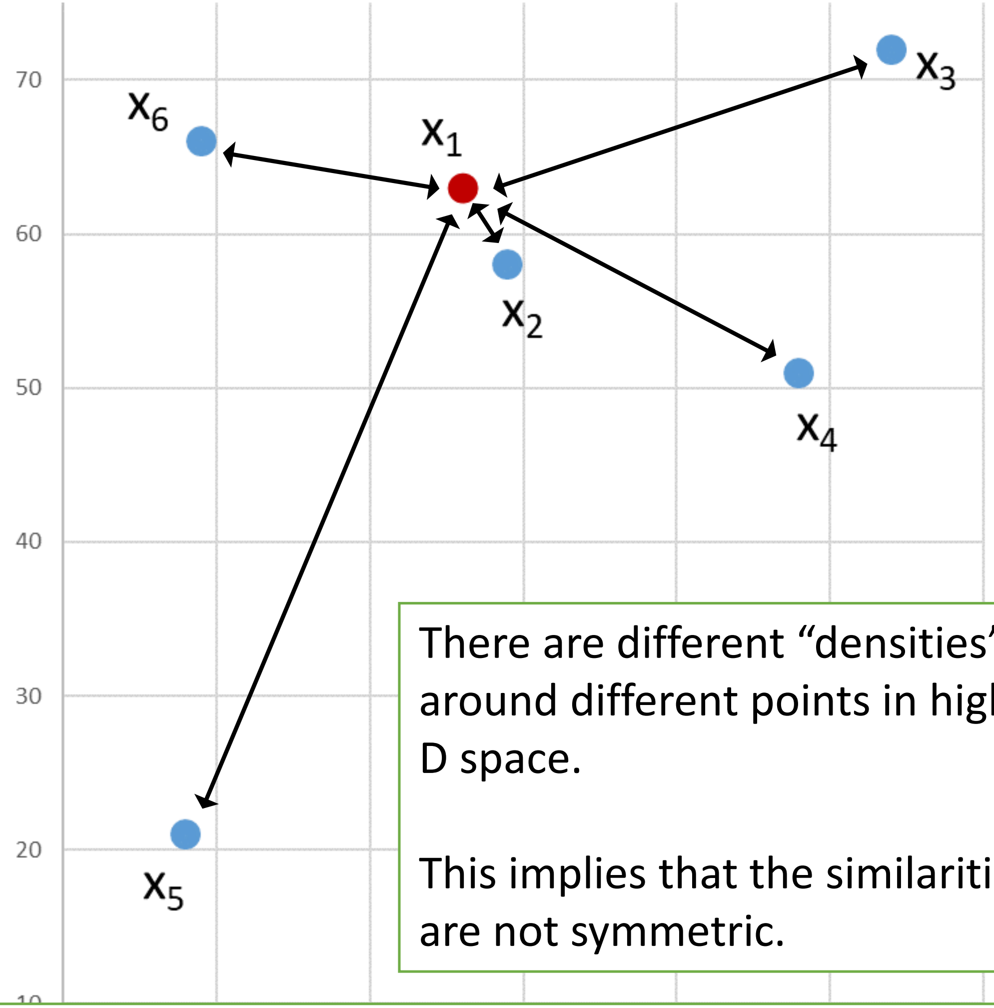
	x_2	x_3	x_4	x_5	x_6
From x_1 to:	5.8	29.4	25.1	45.7	17.3

$$\exp\left(\frac{-\|x_1 - x_k\|^2}{2\sigma_1^2}\right)$$

Assuming σ_1^2		x_2	x_3	x_4	x_5	x_6	Sum
110	From x_1 to:	0.86	0.02	0.06	0.00	0.26	1.19
40	From x_1 to:	0.65	0.00	0.00	0.00	0.02	0.68

Probabilistic similiarities

Assuming σ_1^2		x_2	x_3	x_4	x_5	x_6
110	From x_1 to:	0.72	0.02	0.05	0.00	0.22
40	From x_1 to:	0.96	0.00	0.00	0.00	0.04



There are different "densities" around different points in high-D space.

This implies that the similarities are not symmetric.

The "perplexity" parameter is used to determine the size of a neighborhood around a point; and it is used to set the σ_i^2 values.

t-SNE variation (1 of 2)

- The high-D similarities in SNE are not symmetric.
- t-SNE modifies this to make the similarities symmetric.

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

t-SNE variation (2 of 2)

- For SNE, similarities are also computed for the low dimensional space according to a Gaussian probability distribution

$$q_{ij} = \frac{\exp(-||\mathbf{y}_i - \mathbf{y}_j||^2)}{\sum_{k \neq i} \exp(-||\mathbf{y}_i - \mathbf{y}_k||^2)}$$

- To improve quality of resuls, t-SNE modifies the low-D similarity to be based on a t-Distribution

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

t-SNE continued

- The original high-D points are fixed
  - Indeed, once we have the pairwise similarity matrix; we no longer need the original points
- The representation of the original points on the lower dimensional space however is what we can change.
- The goal is to make similar high-D points also similar in low-D; and dissimilar high-D points also dissimilar in low-D
- Mathematically, t-SNE does this by arranging the low-D points in a way to minimize the Kullback-Leibler divergence function:

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

t-SNE continued

- The Kullback-Leibler divergence function: $C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$

  → large penalty if a large p_{ij} is modeled by a small q_{ij}

  →only a small penalty if a small p_{ij} is modeled by a large q_{ij}

  - This provides a focus on “local structure” of the data

- The points in low-D space are moved around to minimize C; there are multiple iterations, and the directions of movement are determined by the gradient function:

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

t-SNE continued

The movement on low-D space, influence by the gradient function, appears to be like acts like a spring action: points that should be close together are attracted; points that should be far apart are repelled.

t-SNE in R