Advanced Data Visualization

CS 6965

Spring 2018

Prof. Bei Wang Phillips

University of Utah

Lecture 03

Dim Reduction & Vis
t-SNE

HD

Announcement

- Project 1 has been posted on the schedule webpage:
  - http://www.sci.utah.edu/~beiwang/teaching/cs6965-spring-2018/schedule.html
  - project1_posted.zip
  - Please start early
- Project 1 is due on Feb. 1st, Thursday, before the start of the class

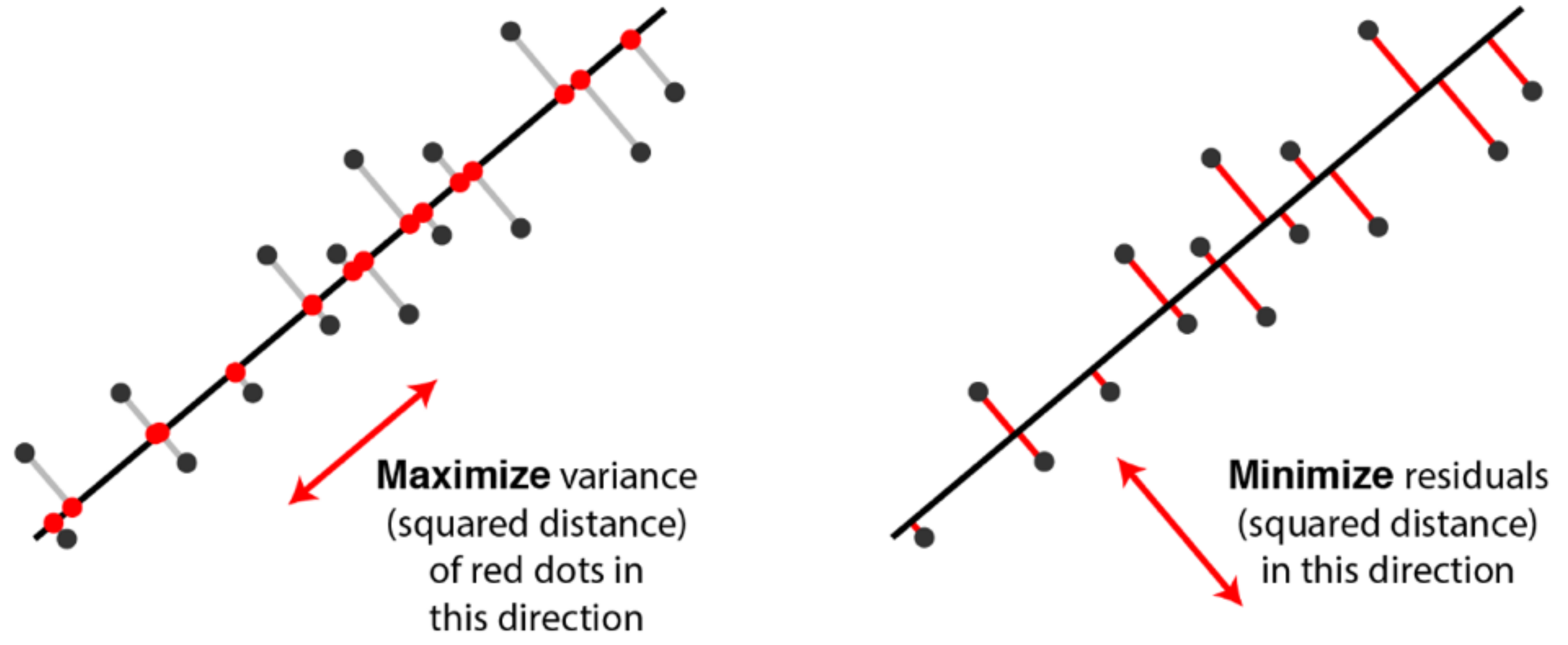
Vis + DR: PCA

Revisited

Two interpretation of PCA

PCA can be interpreted in two different ways:

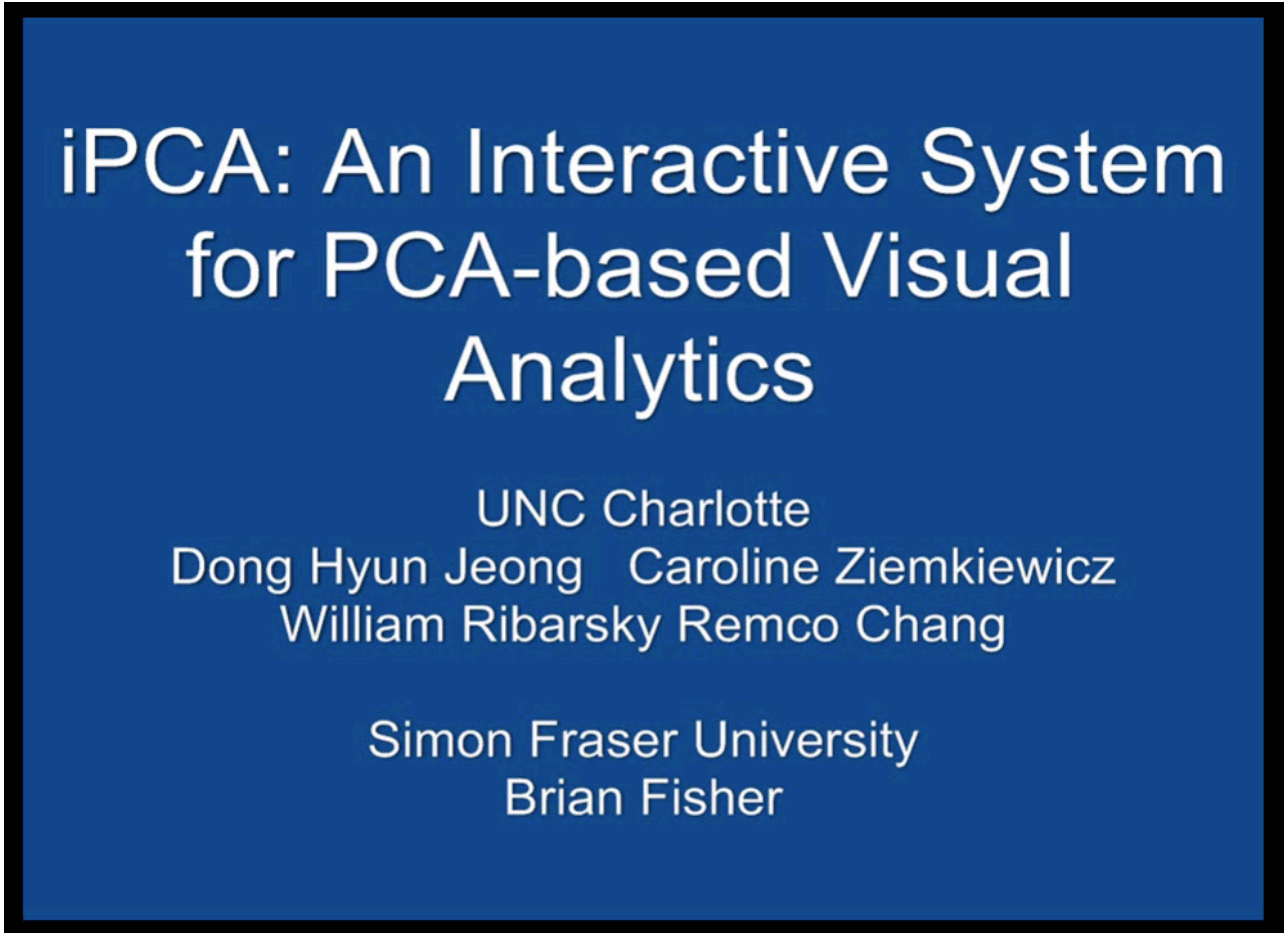
- Maximize the variance of projection along each component (dimension).
- Minimize the reconstruction error, that is, the squared distance between the original data and its projected coordinates.



Two equivalent views of principal component analysis.

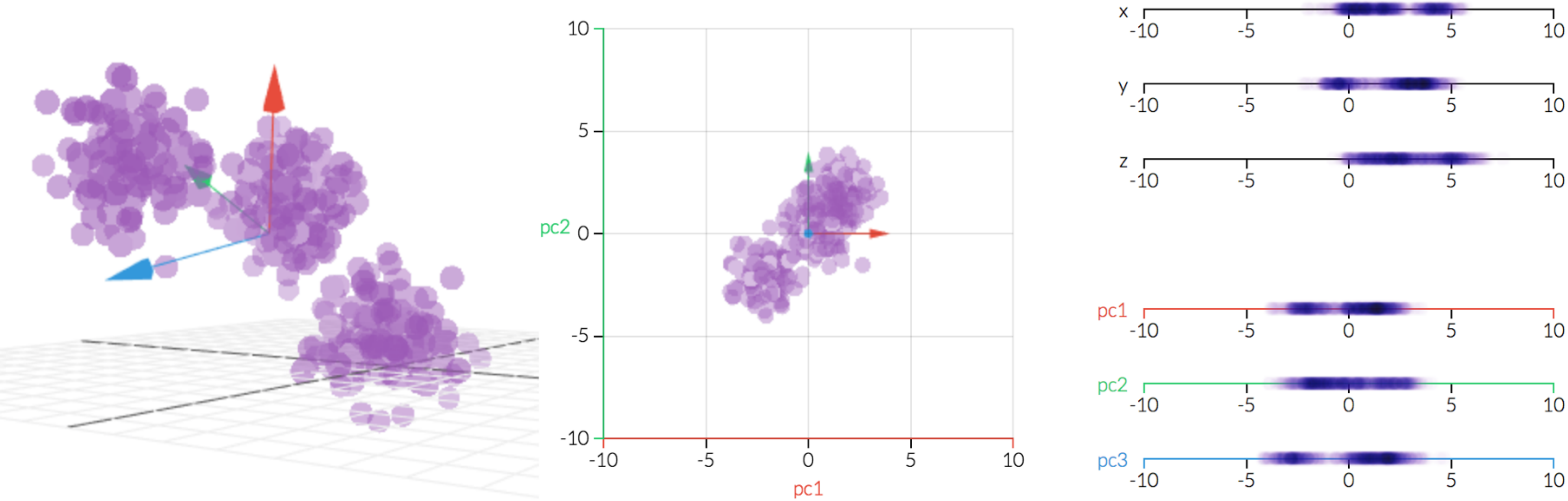
http://alexhwilliams.info/itsneuronalblog/2016/03/27/pca/#some-things-you-maybe-didnt-know-about-pca

iPCA: interactive PCA



Source: http://www.knowledgeviz.com/iPCA/ [JeongZiemkiewiczFisher2009]
Video also available at: http://www.cs.tufts.edu/~remco/publication.html

Visually explaining PCA



Source: http://setosa.io/ev/principal-component-analysis/

Additional thoughts on Vis+PCA

- Use visualization to explain the inner-working of PCA algorithms (or any other DR algorithms)
- Manipulate algorithm input and output and observe its behavior, e.g. add/delete/move data points, rescaling, etc.
- Observe the algorithmic process, e.g. eigenvectors, etc.

Vis + DR: t-SNE

A case study with a nonlinear DR method

[vanderMaatenHinton2008]
The material from this section is heavily drawn from Jaakko Peltonen
http://www.uta.fi/sis/mtt/mtts1-dimensionality_reduction/drv_lecture10.pdf

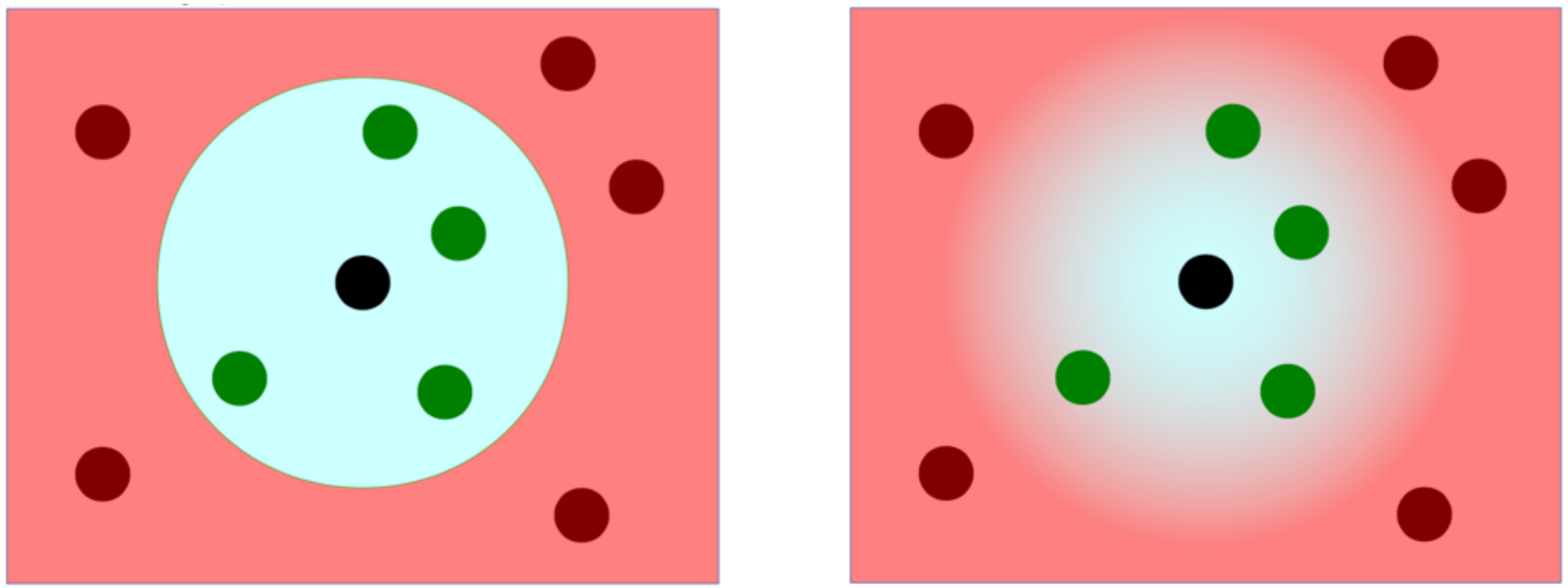
DR: preserving distances

$$C = \frac{1}{a}\sum_{ij} w_{ij}(d_X(x_i, x_j) - d_Y(y_i, y_j))^2$$

- Many DR methods focus on preserving distances, e.g. the above is the cost function for a particular DR method called metric MDS
- An alternative idea is preserving neighborhoods.

DR: preserving neighborhoods

- Neighbors are an important notion in data analysis, e.g.social networks, friends, twitter followers…
- Object nearby (in a metric space) are considered neighbors
- Consider hard neighborhood and soft neighborhood
- Hard: each point is a neighbor (green) or a non-neighbor (red)
- Soft: each point is a neighbor (green) or a non-neighbor (red) with some weight

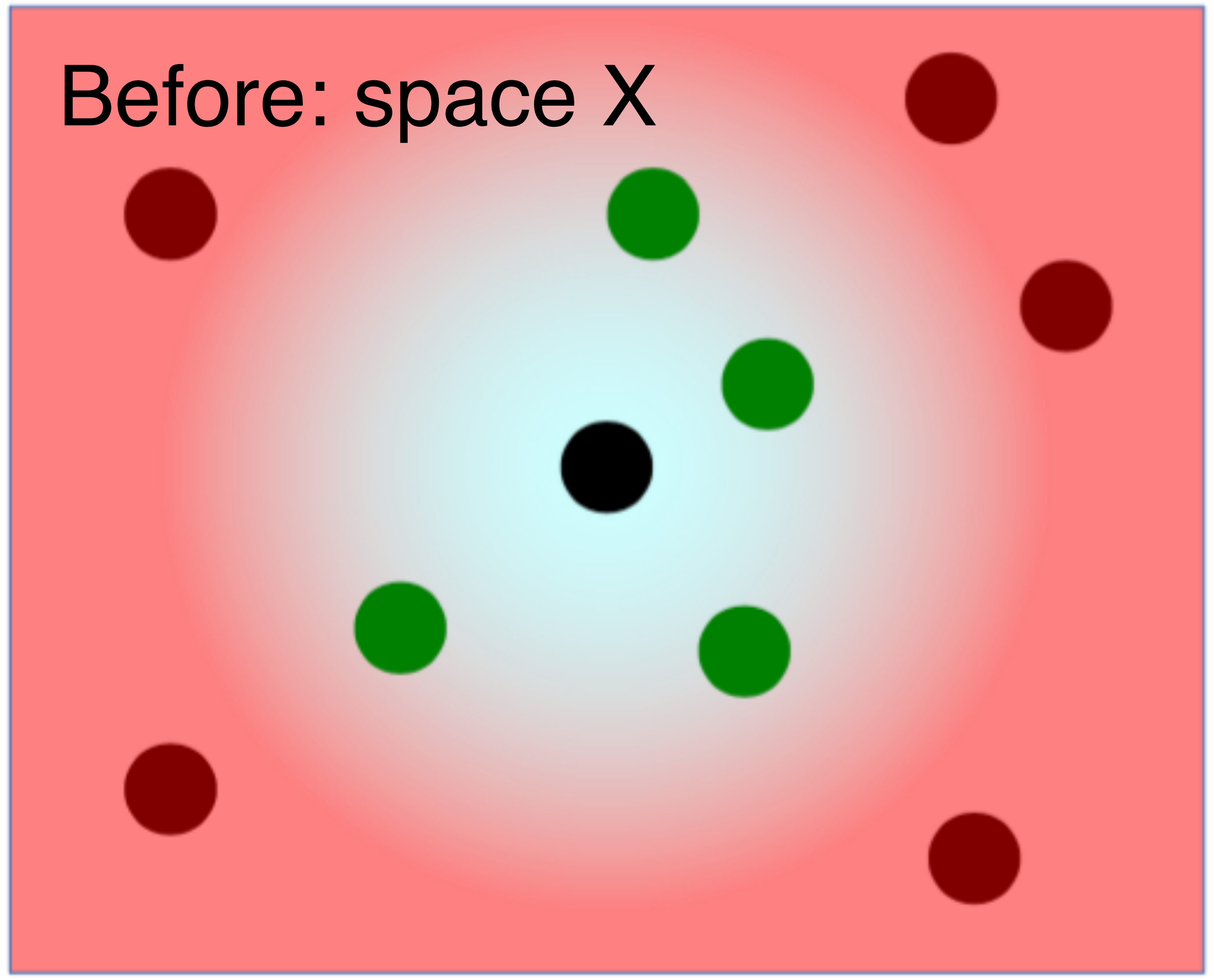


Probabilistic neighborhood

- Derive a probability of point j to be picked as a neighbor of i in the input space

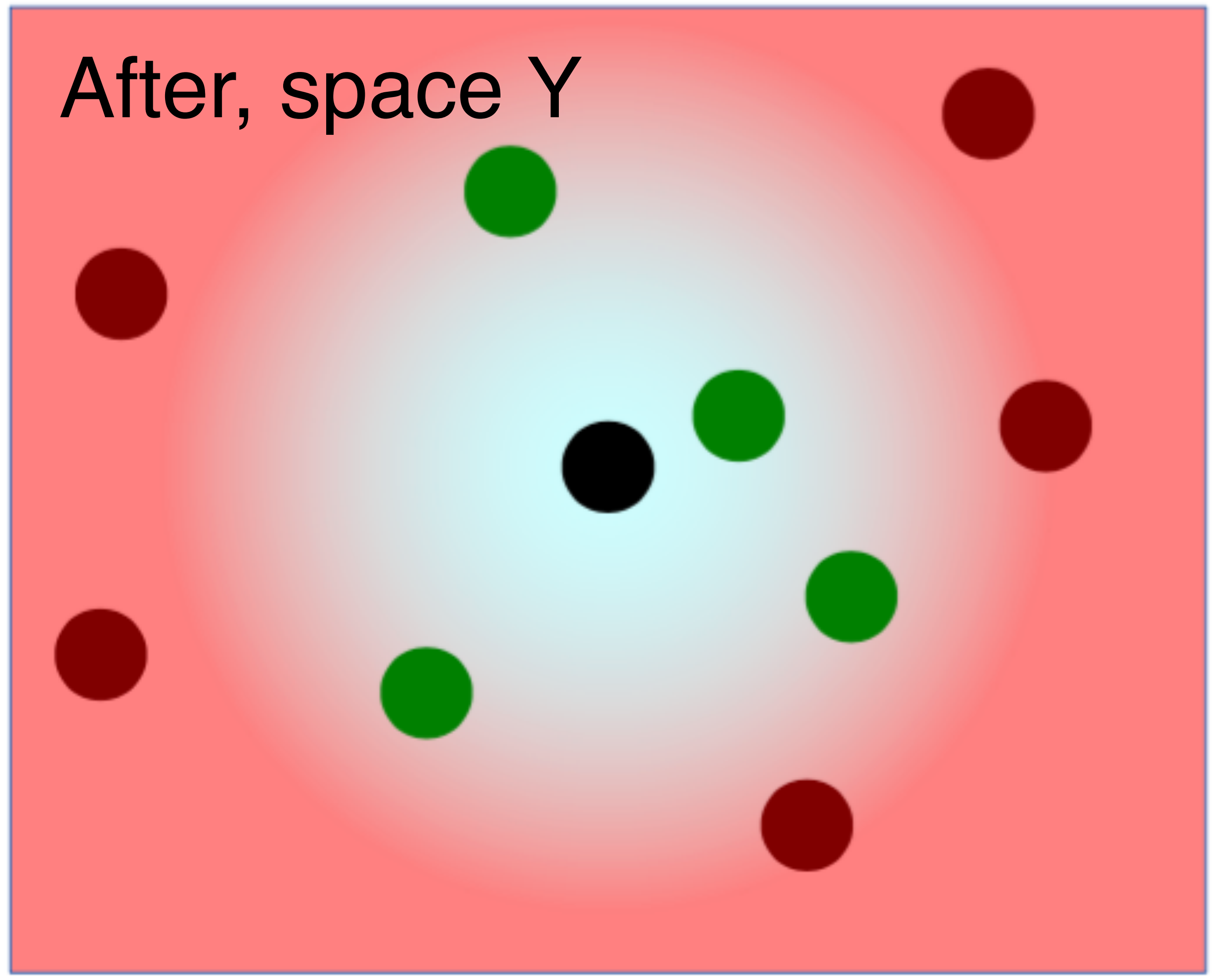
$$p_{ij} = \frac{exp(-d_{ij}^2)}{\sum_{k \neq i} exp(-d_{ik}^2)}$$

Preserving nbhds before & after DR



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

Probabilistic input neighborhood:
Probability to be picked as a neighbor in space X (input coordinates)



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

Probabilistic output neighborhood:
Probability to be picked as a neighbor in space Y (display coordinates)

Stochastic neighbor embedding

- Compare neighborhoods between the input and output!
- Using Kullback-Leibler (KL) divergence
- KL divergence: relative entropy (amount of surprise when encounter items from 1st distribution when they are expected to come from the 2nd)
- KL divergence is nonnegative and 0 iff the distributions are equal
- SNE: minimizes the KL divergence using gradient descent

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

SNE: choose the size of a nbhd

- How to set the size of a neighborhood? Using a scale parameter: σ_i

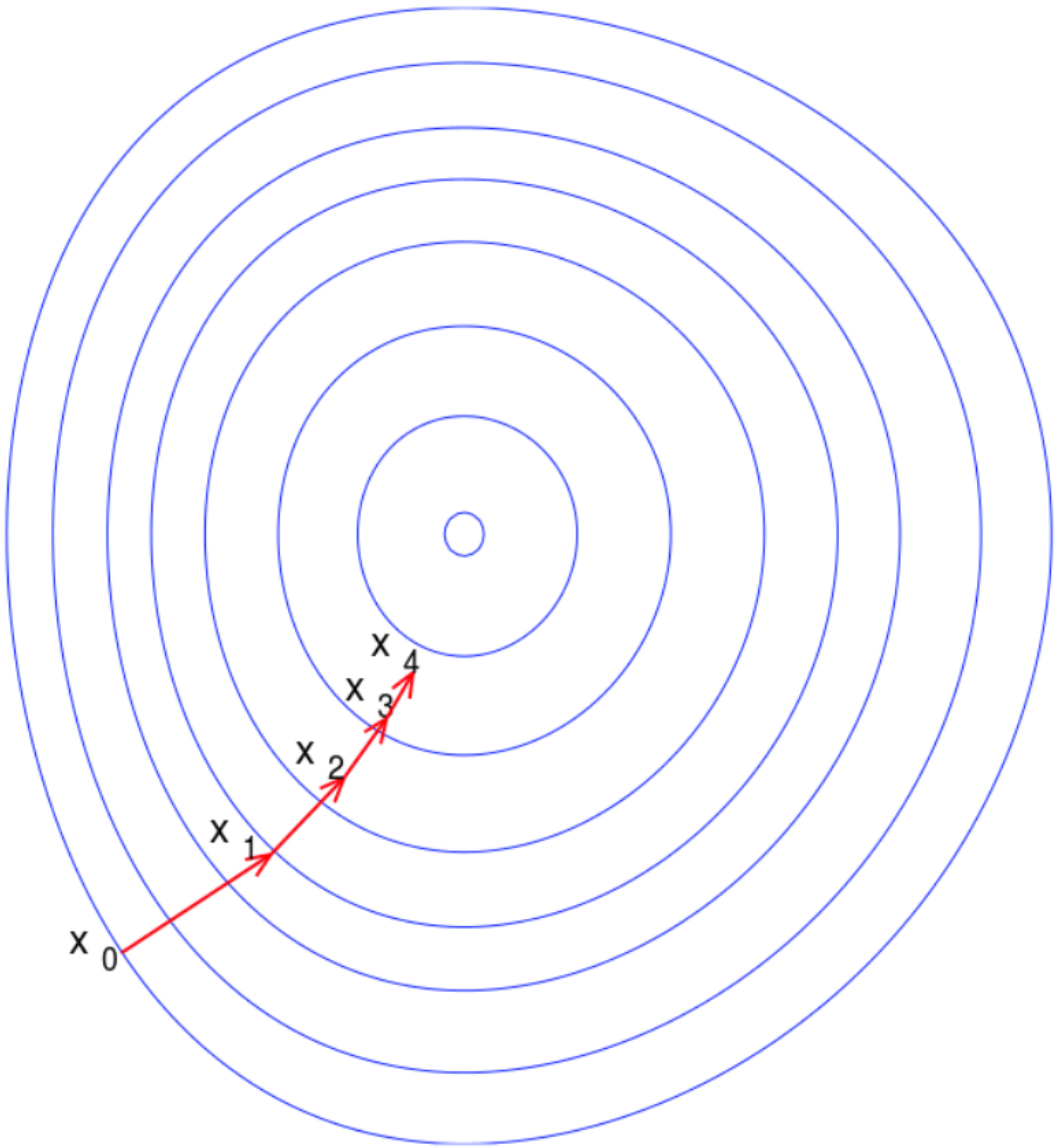
$$d_{ij}^2 = \frac{||x_i - x_j||^2}{2\sigma_i^2}$$

- The scale parameter can be chosen without knowing much about the data, but…
- It is better to choose the parameter based on local neighborhood properties, and for each point
- E.g., in sparse region, distance drops more gradually

SNE: choose a scale parameter

Choose an effective number of neighbors:

- In a uniform distribution over k neighbors, the entropy is log(k)
- Find the scale parameter using binary search so that the entropy of p_{ij} becomes log(k) for a desired value of k.



Gradient descent

https://en.wikipedia.org/wiki/Gradient_descent

SNE: gradient descent

- Adjusting the output coordinates using gradient descent
- Gradient descent: iterative process to find the minimal of a function

- Start from a random initial output configuration, then iteratively take steps along the gradient
- Intuition: using forces to pull and push pairs of points to make input and output probabilities more similar

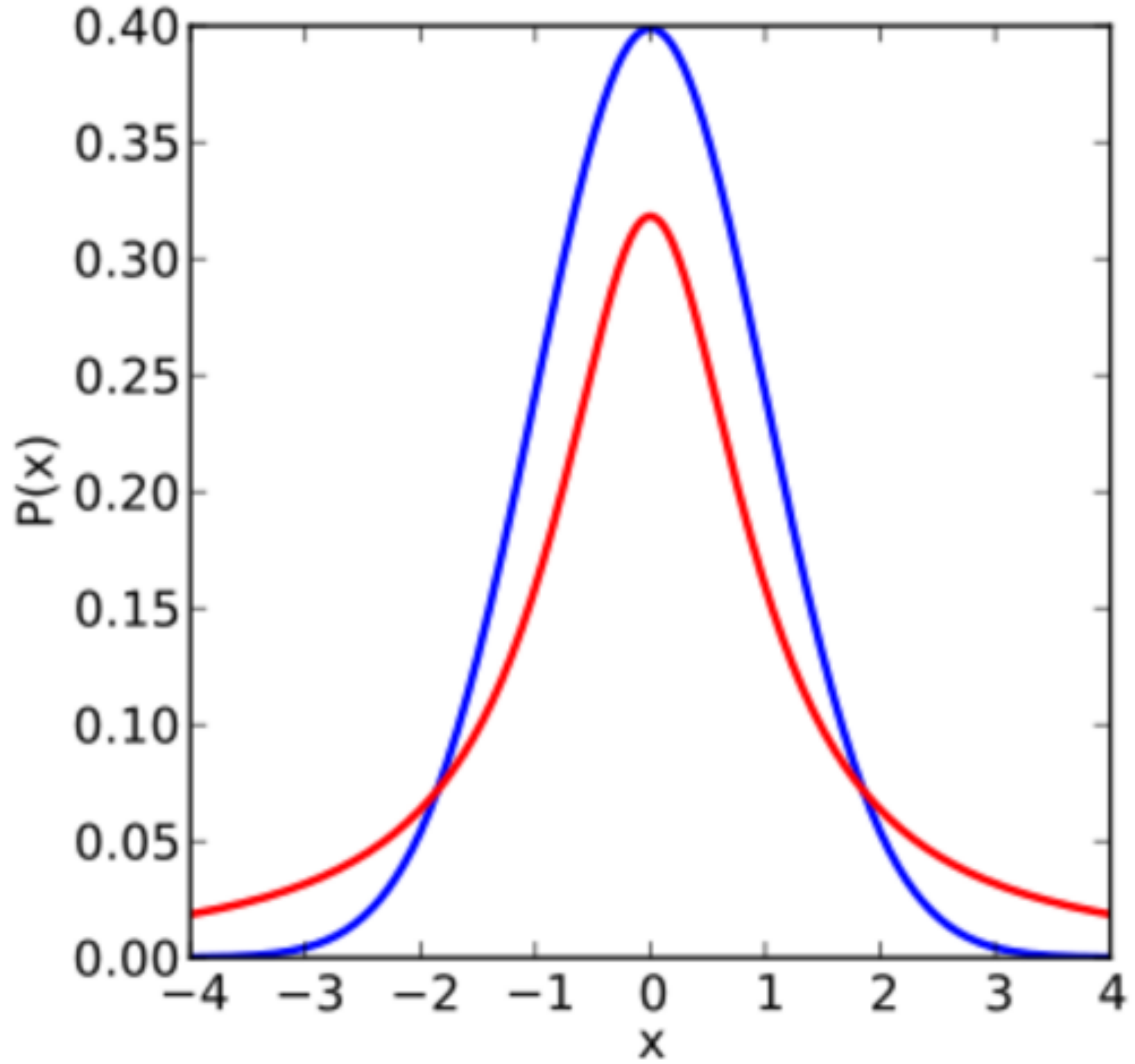
$$\frac{\partial C}{\partial y_i} = 2 \sum_j (y_i - y_j)(p_{ij} - q_{ij} + p_{ji} - q_{ji})$$

SNE: the crowding problem

- When embedding neighbors from a high-dim space into a low- dim space, there is too little space near a point for all of its close-by neighbors.
- Some points end up too far-away from each other
- Some points that are neighbors of many far-away points end up crowded near the center of the display.
- In other words, these points end up crowded in the center to stay close to all of the far-away points.
- t-SNE: using heavy-tailed distributions (i.e., t-distributions) to define neighbors on the display, to resolve the crowding problem

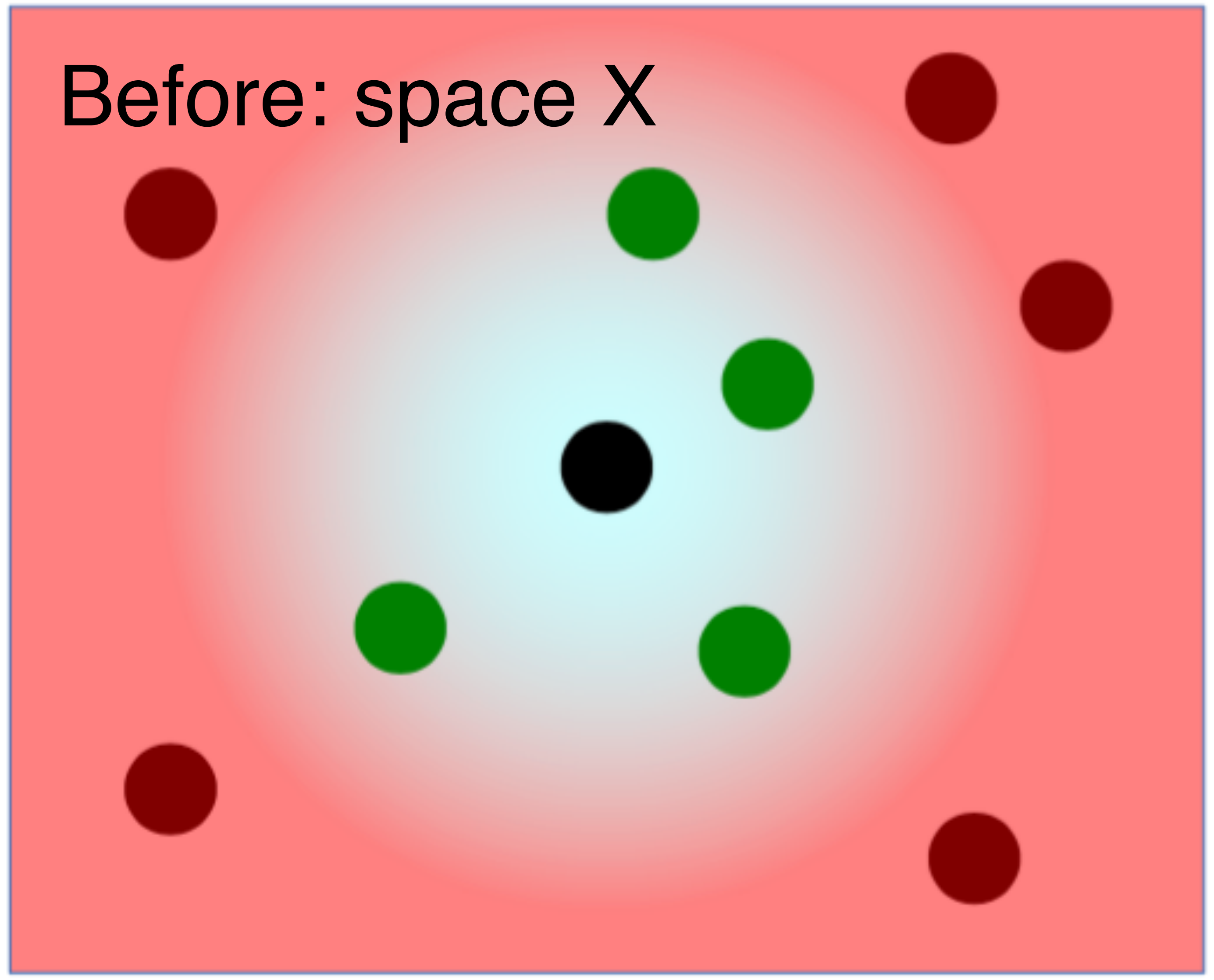
t-distributed SNE

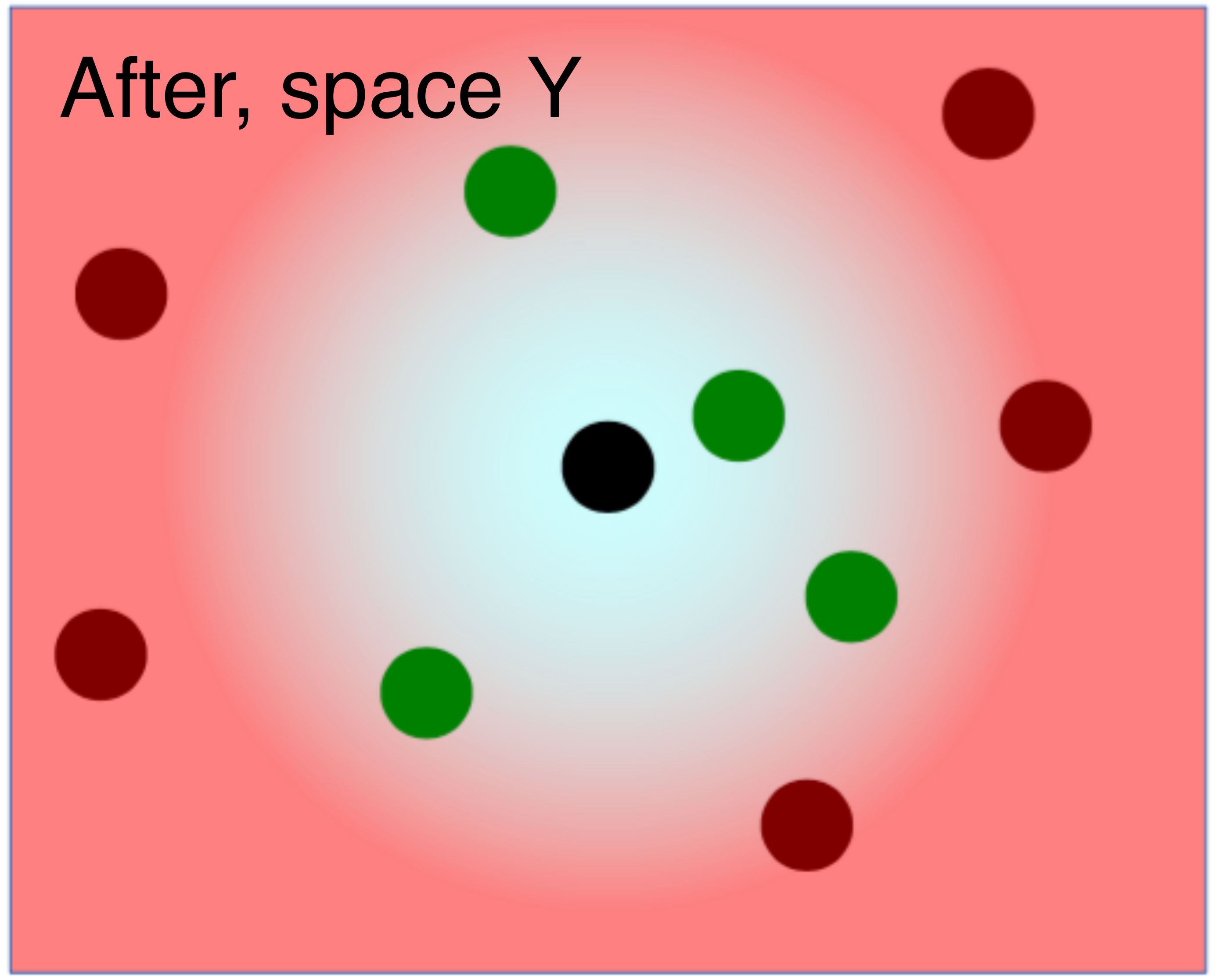
- Avoids crowding problem by using a more heavy-tailed neighborhood distribution in the low-dim output space than in the input space.
- Neighborhood probability falls off less rapidly; less need to push some points far off and crowd remaining points close together in the center.
- Use student-t distribution with 1 degree of freedom in the output space
- t-SNE (joint prob.); SNE (conditional prob.)



Blue: normal dist.
Red: student-t dist. with 1 deg. of freedom

t-SNE: preserving nbhds





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

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

Probabilistic input neighborhood:
Probability to be picked as a neighbor in space X (input coordinates)

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

Probabilistic output neighborhood:
Probability to be picked as a neighbor in space Y (display coordinates)

t-SNE minimization

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

Minimize divergence between symmetric probabilities

[vanderMaatenHinton2008]

Various Components of t-SNE alg.

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}, \tag{1}$$

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}. \tag{4}$$

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

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j) \left(1 + \|y_i - y_j\|^2\right)^{-1}. \tag{5}$$

[vanderMaatenHinton2008]

Implementing t-SNE

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

Data: data set $\mathcal{X} = \{x_1, x_2, ..., x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, ..., y_n\}$.
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 $\mathcal{Y}^{(0)} = \{y_1, y_2, ..., y_n\}$ 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

What is Perplexity?

$$Perp(P_i) = 2^{H(P_i)},$$

where $H(P_i)$ is the Shannon entropy of P_i measured in bits

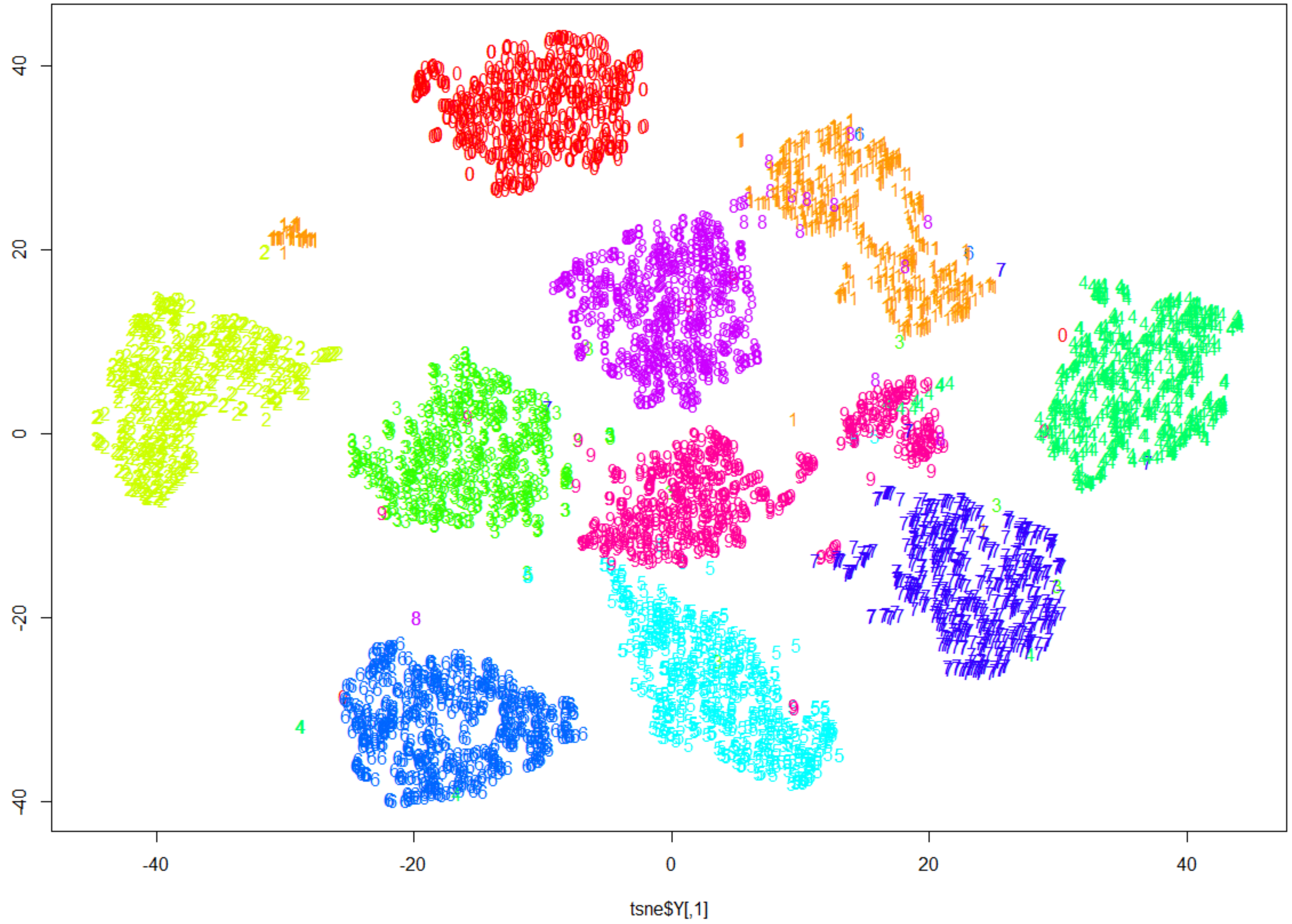
$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}.$$

- Perform a binary search for the value of σi that produces a Pi with a fixed perplexity that is specified by the user
- Perplexity increases monotonically with the variance σi.
- t-SNE determines the local neighborhood size for each datapoint separately based on the local density of the data (by forcing each conditional probability distribution Pi to have the same perplexity).

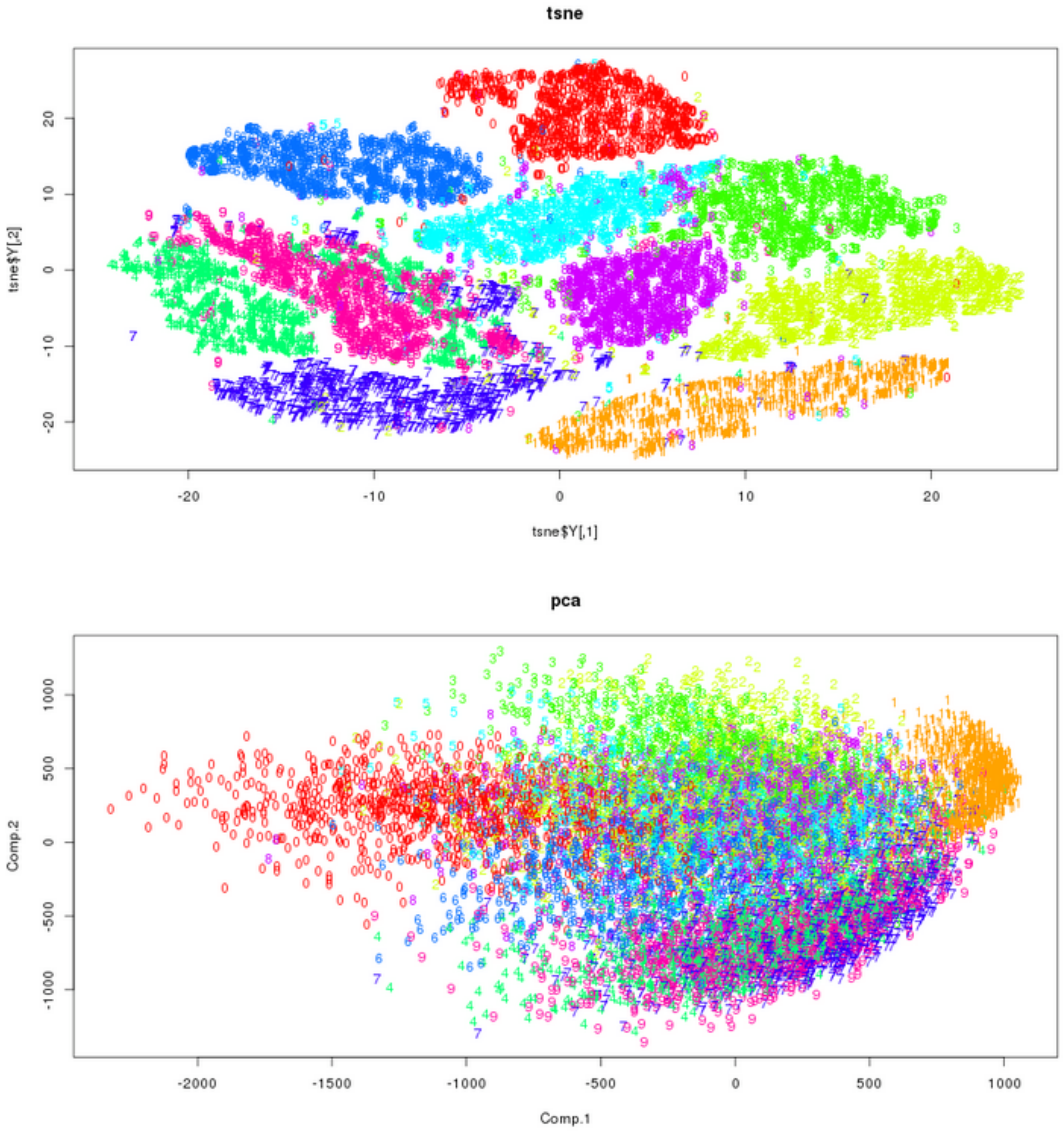
What is Perplexity?

The perplexity can be interpreted as a smooth measure of the effective number of neighbors. The performance of SNE is fairly robust to changes in the perplexity, and typical values are between 5 and 50.

Classic t-SNE result



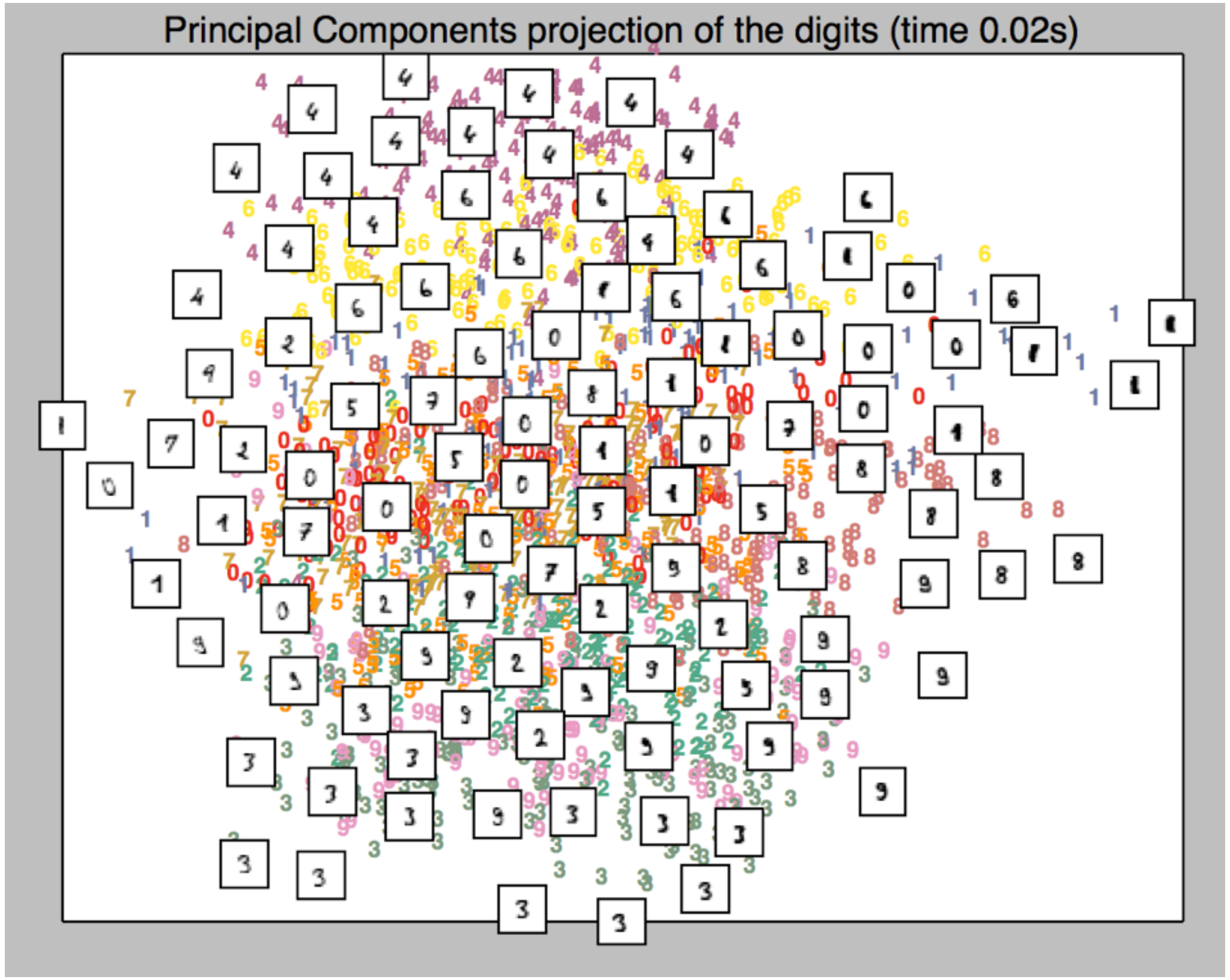
t-SNE vs PCA

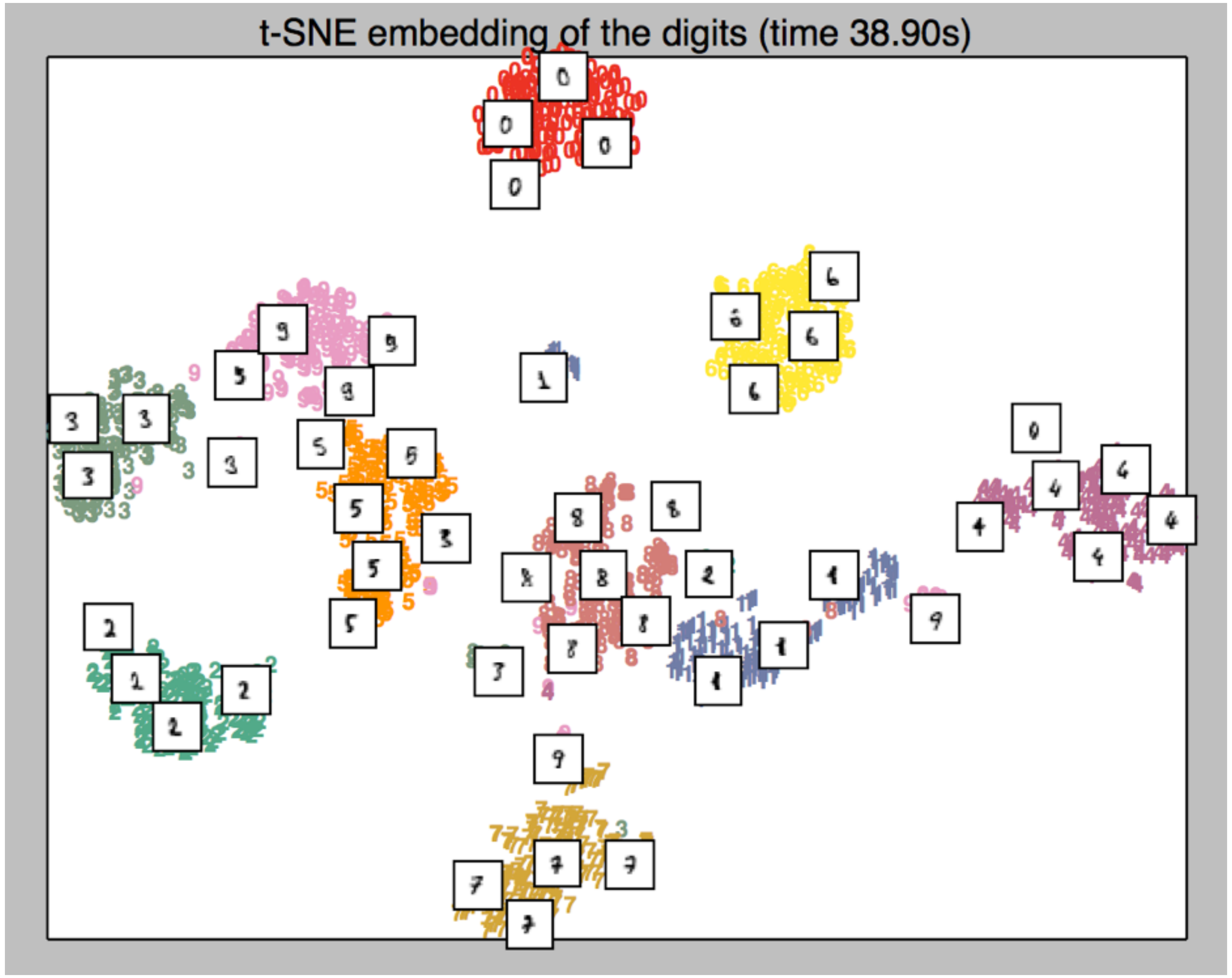


t-SNE with scikit-learn: demo

http://scikit-learn.org/stable/auto_examples/manifold/plot_lle_digits.html

t-SNE with scikit-learn: demo 2





http://scikit-learn.org/stable/auto_examples/manifold/plot_lle_digits.html

t-SNE in a nutshell

- t-SNE: minimize KL divergence.
- Nonlinear DR.
- Perform diff. transformation on diff. regions: main source of confusing.
- Parameter: perplexity, how to balance attention between local and global aspects of your data; guess the # of close neighbor each point has.
- “The performance of t-SNE is fairly robust under different settings of the perplexity. The most appropriate value depends on the density of your data. Loosely speaking, one could say that a larger / denser dataset requires a larger perplexity. Typical values for the perplexity range between 5 and 50.” (Laurens van der Maaten)

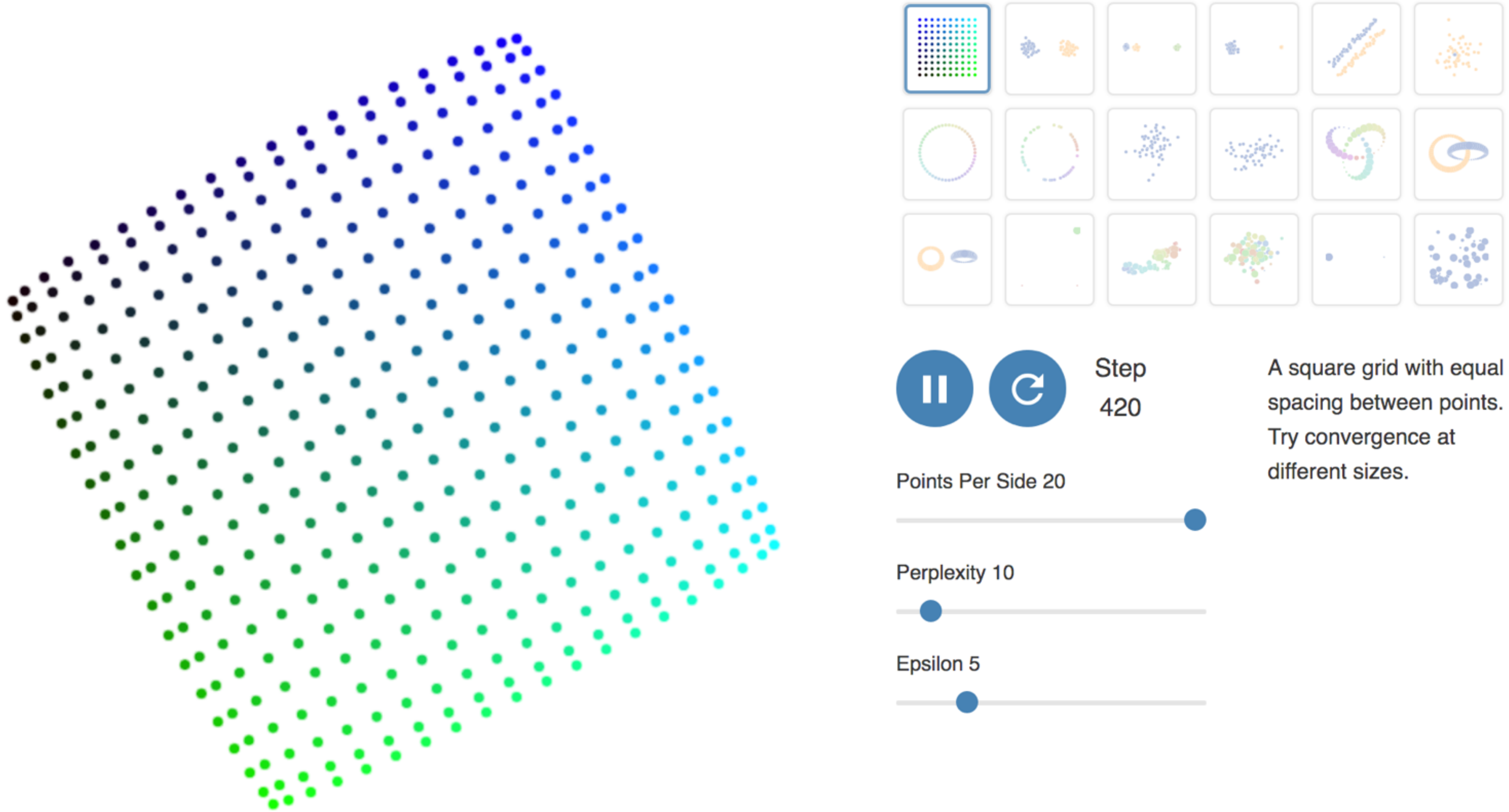
Source: https://distill.pub/2016/misread-tsne/

What is perplexity anyway?

- “Perplexity is a measure for information that is defined as 2 to the power of the Shannon entropy. The perplexity of a fair die with k sides is equal to k. In t-SNE, the perplexity may be viewed as a knob that sets the number of effective nearest neighbors. It is comparable with the number of nearest neighbors k that is employed in many manifold learners.”

Source: https://lvdmaaten.github.io/tsne/

How not to misread t-SNE



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

Playing with t-SNE further

- http://scikit-learn.org/stable/auto_examples/manifold/plot_t_sne_perplexity.html
- https://lvdmaaten.github.io/tsne/

Weakness of t-SNE

- Not clear how it performs on general DR tasks
- Local nature of t-SNE makes it sensitive to intrinsic dim of the data
- Not guaranteed to converge to global minimum

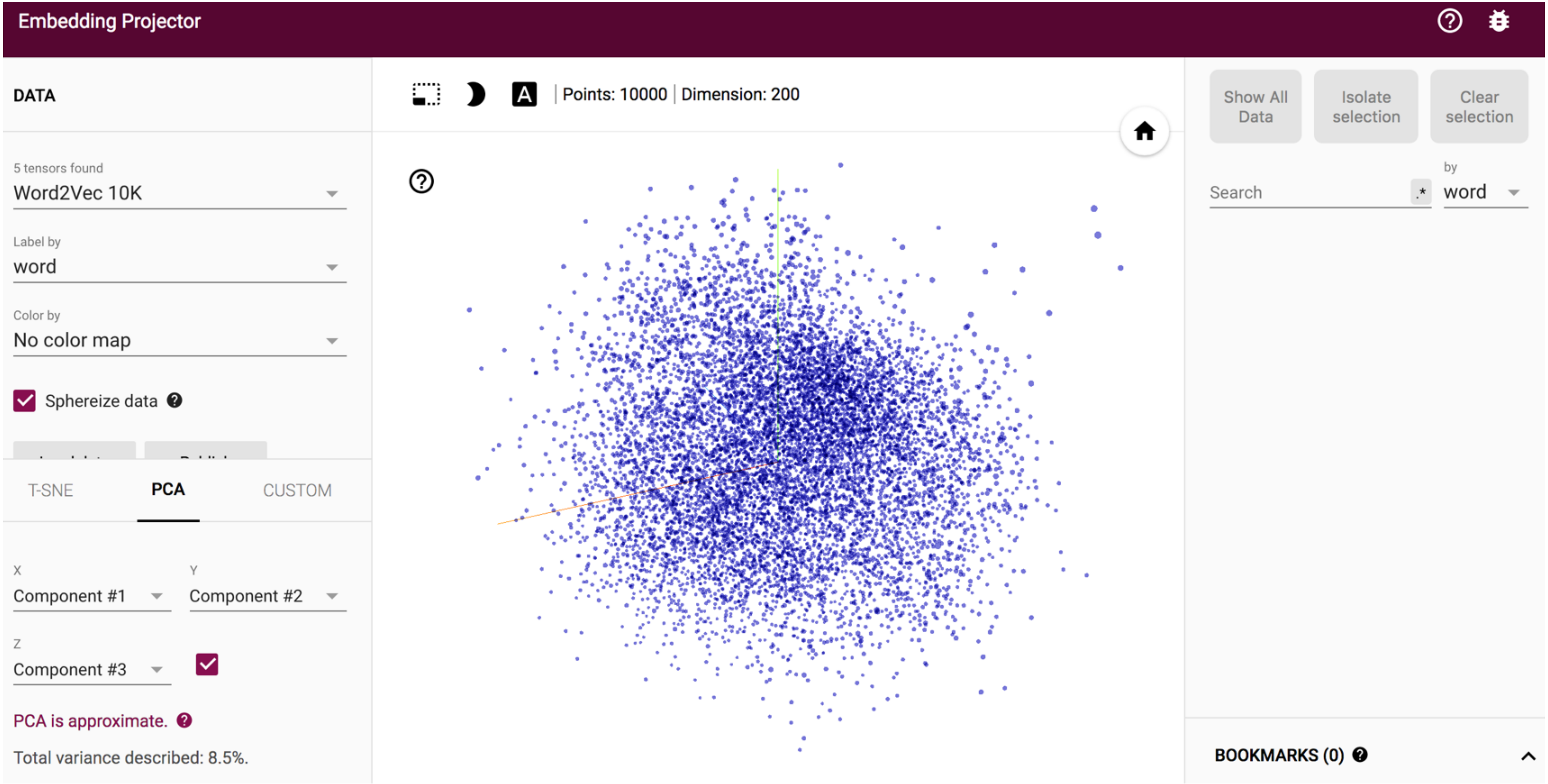
Take home message

- Even a simple DR method like PCA can have interesting visualization aspects to it
- Using visualization to manipulate the input to the ML algorithm, and at the same time understanding the interworking of the algorithm
- Cooperative analysis, mobile devices, virtue reality?

- t-SNE is useful, but only when you know how to interpret it
- Those hyper-parameters, such as perplexity, really matter
- Use visualization to interpret the ML algorithm
- Educational purposes to distill algorithms as glass boxes

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

Embedding Projector



http://projector.tensorflow.org/

Potential Final Projects

- Inspired by:
  - http://setosa.io/ev/principal-component-analysis/
  - https://distill.pub/2016/misread-tsne/
- Extending Embedding Projector: Interactive Visualization and Interpretation of Embeddings
  - https://opensource.googleblog.com/2016/12/open-sourcing-embedding-projector-tool.html
  - http://projector.tensorflow.org/
  - https://www.tensorflow.org/versions/r1.2/get_started/embedding_viz

Can you create a web-based tools that give good visual interpretation of two linear DR and two nonlinear DR techniques?

Getting ready for Project 1

- Scikit-learn tutorial:
  - http://scikit-learn.org/stable/tutorial/basic/tutorial.html
- Install and read the documentation of kepler-mapper:
  - https://github.com/MLWave/kepler-mapper
- Play with examples provided by kepler-mapper

Thanks!

Any questions?

You can find me at: **beiwang@sci.utah.edu**

CREDITS

Special thanks to all people who made and share these awesome resources for free:

- Presentation template designed by Slidesmash
- Photographs by unsplash.com and pexels.com
- Vector Icons by Matthew Skiles

Presentation Design

This presentation uses the following typographies and colors:

Free Fonts used:

http://www.1001fonts.com/oswald-font.html

https://www.fontsquirrel.com/fonts/open-sans

Colors used