Advanced Data Visualization

CS 6965

Fall 2019

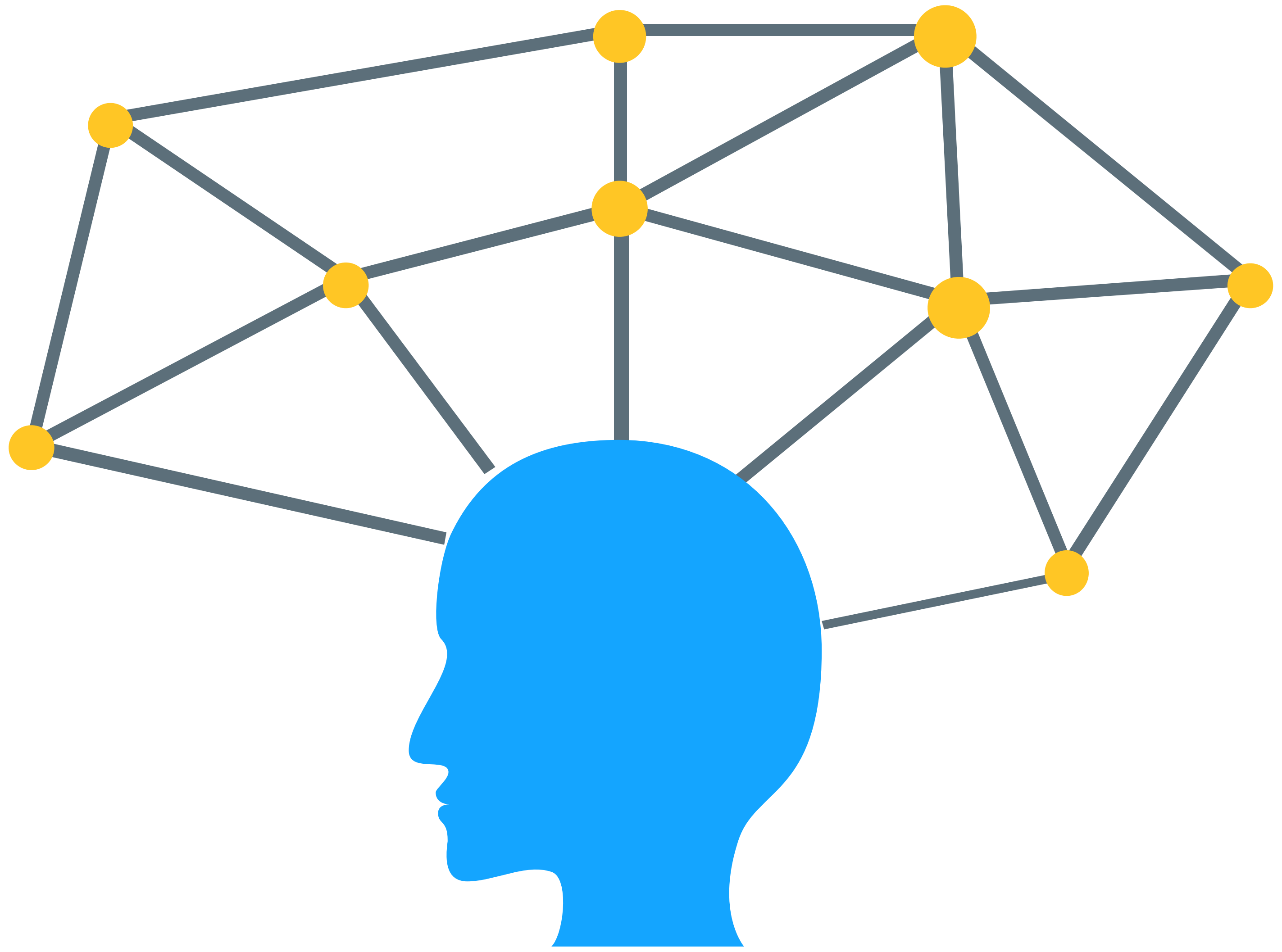
Prof. Bei Wang Phillips

University of Utah

Lecture 02

Dim Reduction & Vis

HD



Visualization
is the secret weapon for
Machine learning

Roles of ML in HD data visualization

From Black Box to Glass Box:

- ML as part of data transformation in the visualization pipeline
- Visualization increase the interpretability of the algorithmic results (visualizing algorithm output)
- Visualization increases the interpretability of ML algorithms (visualizing algorithmic processes)
- (Interactive) visualization becomes part of the ML algorithm

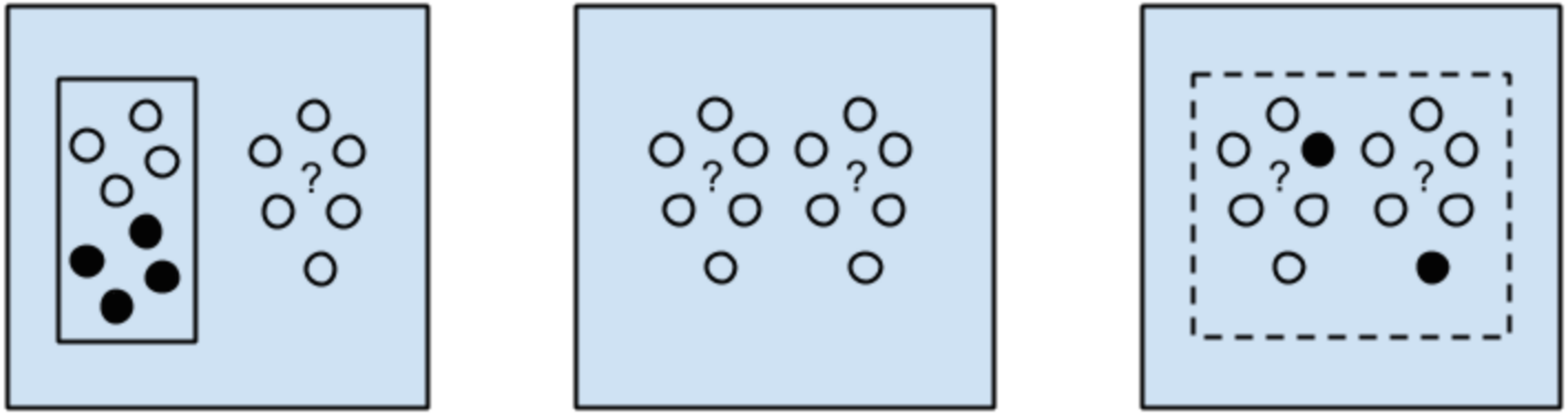
ML algorithms in a nutshell

Not a full-blown ML class, but

How to best incorporate vis into ML algorithms?

- A simple approach is to treat the ML algorithm as a black box, and build vis surrounding its input/output
- Not knowing the interworking of the algorithm (e.g. a glass box) may lead to misinterpretation of the algorithm output
- We need to have a good understanding of the core of some ML algorithms
- We will review some ML algorithms with a focus on their inner-workings so as to think about how visualization can be incorporated
- You are encouraged to read about ML in general (see recommended reading, and talk to the instructor)
- Keep in mind, our focus is ML+Vis

ML algorithm by learning styles

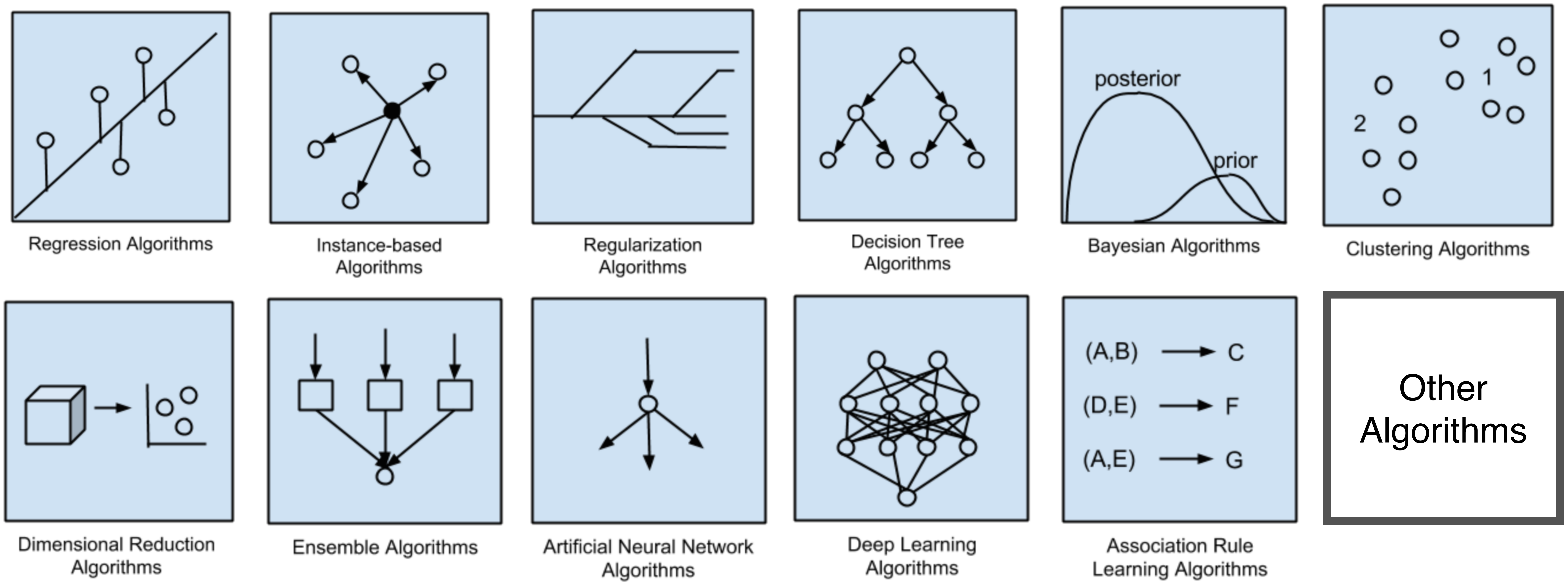


Supervised Learning	Unsupervised Learning	Semi-supervised Learning
Problems: Classification Regression	Problems: Clustering Dimensionality Reduction	Problems: Classification Regression

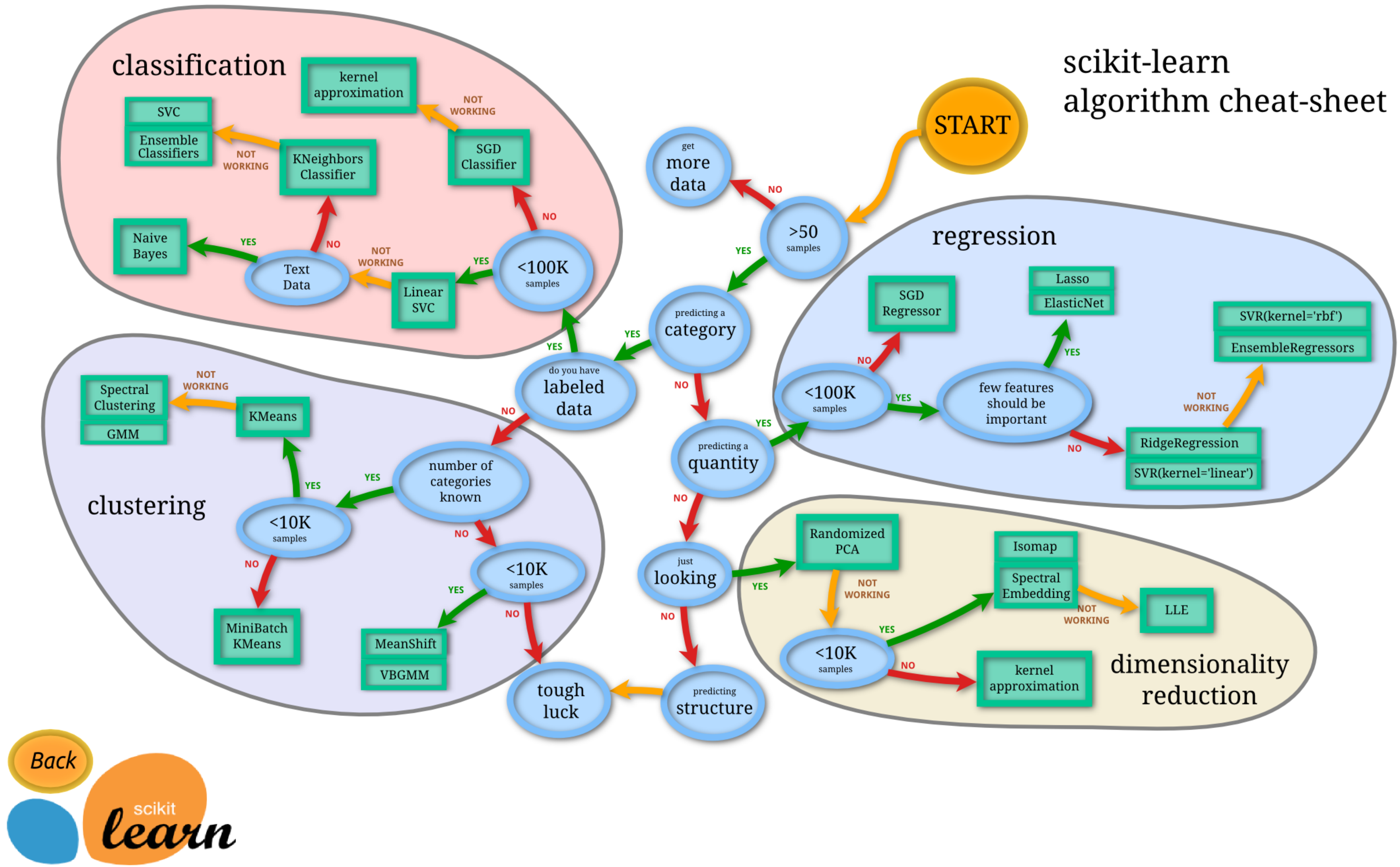
Source: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

MACHINE LEARNING MASTERY

ML algorithm by similarity (how they work)

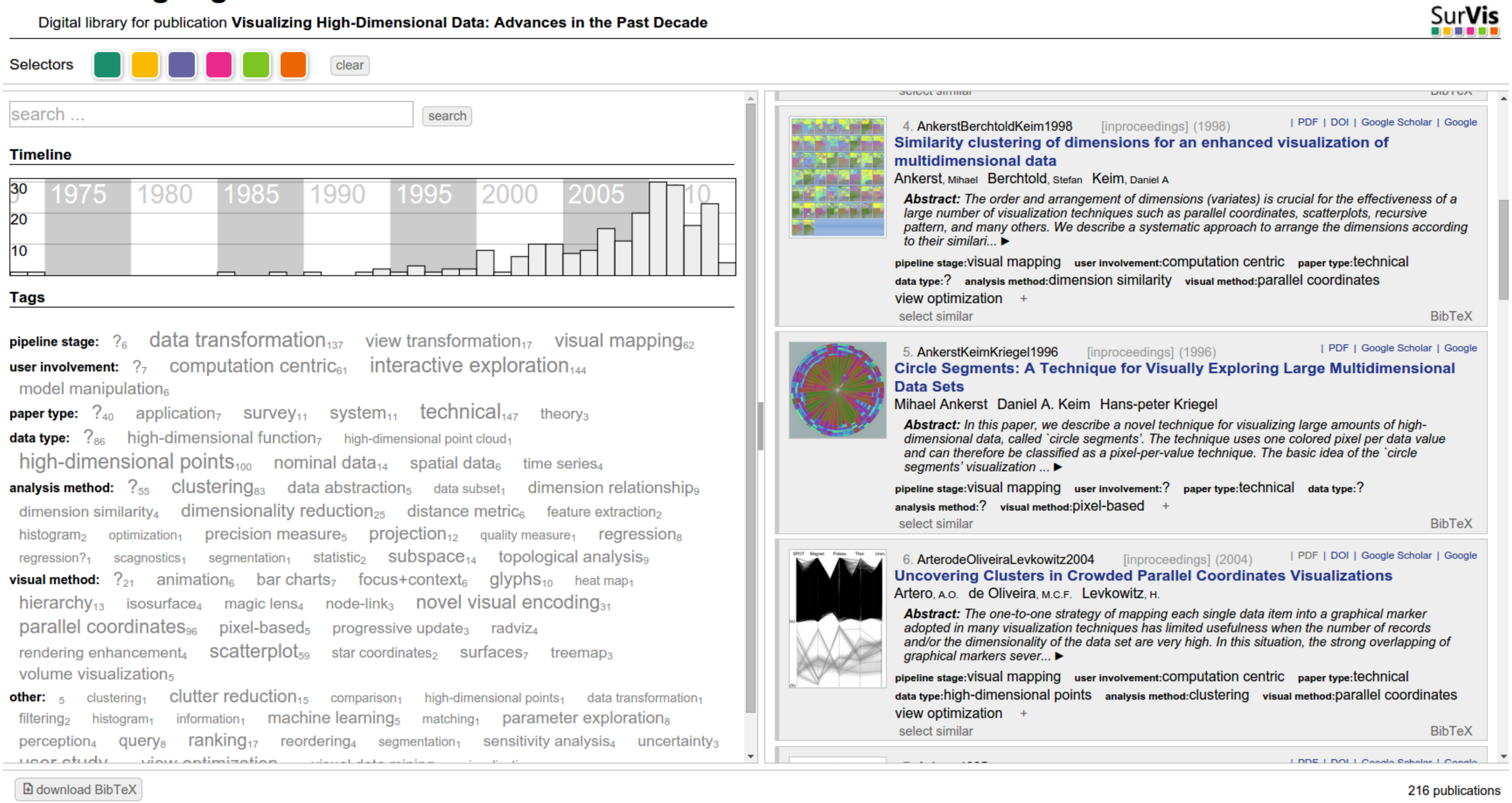


Source: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

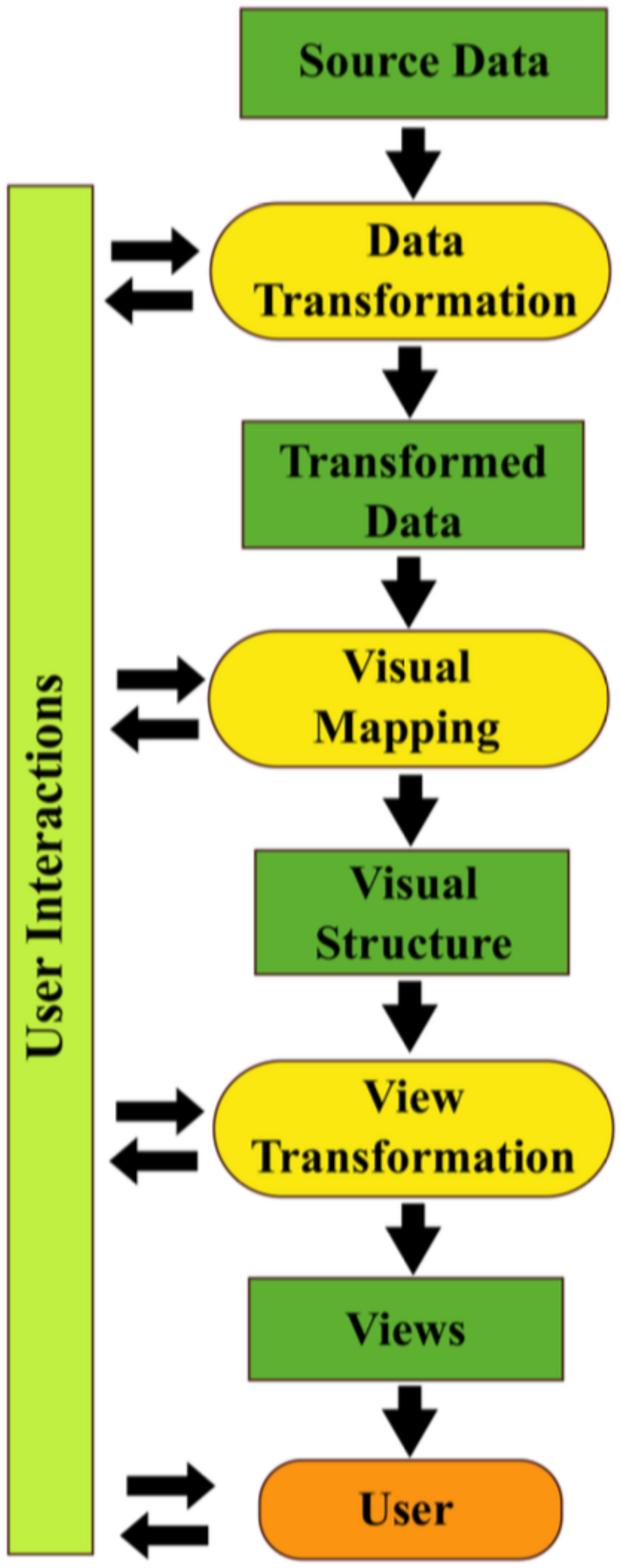

scikit-learn
algorithm cheat-sheet
START
>50 samples
get more data
predicting a category
do you have labeled data
<100K samples
Linear SVC
Text Data
Naive Bayes
KNeighbors Classifier
SVC
Ensemble Classifiers
SGD Classifier
kernel approximation
classification
clustering
number of categories known
<10K samples
KMeans
Spectral Clustering
GMM
MiniBatch KMeans
MeanShift
VBGMM
tough luck
predicting a quantity
regression
SGD Regressor
few features should be important
Lasso
ElasticNet
RidgeRegression
SVR(kernel='linear')
SVR(kernel='rbf')
EnsembleRegressors
just looking
predicting structure
Randomized PCA
Isomap
Spectral Embedding
LLE
kernel approximation
dimensionality reduction
YES
NO
NOT WORKING
Back
scikit learn

Advances in HD Vis

Visualizing High-Dimensional Data: Advances in the Past Decade



Bug Report Welcome! [LiuMaljovecWang2017] http://www.sci.utah.edu/~shusenl/highDimSurvey/website/

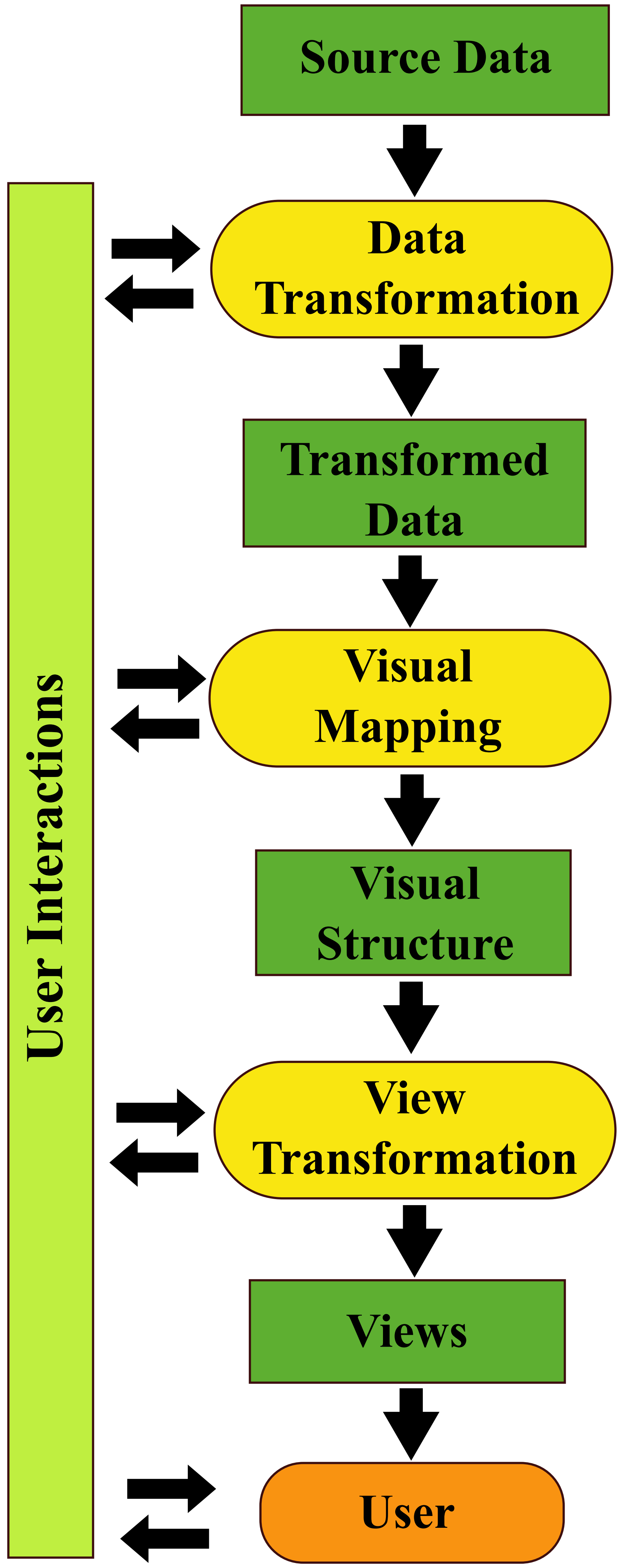


Visualization pipeline for high-dim data

[LiuMaljovecWang2017]

Visualization pipeline for HD data

[LiuMaljovecWang2017]



Data Transformation

Dimension Reduction
Linear Projection [23], [25],
Nonlinear DR [26], [30],
Control Points Projection [34], [37]
Distance Metric [38, 39],
Precision Measures [42], [44]

Subspace Clustering
Dimension Space Exploration
[47], [48], [49],
Subset of Dimension [51], [53],
Non-Axis-Parallel Subspace
[56], [57], [58]

Regression Analysis
Optimization &
Design Steering
[61], [62], [63],
Structural Summaries
[67], [68]

Topological Data Analysis
Morse-Smale Complex
[166], [168], [169], [170],
Reeb Graph [174], [175], [181]
Contour Tree [179, 180],
Topological Features [191], [192]

Visual Mapping

Axis-Based
Scatterplot Matrix [70],
Parallel Coordinate [77],
Radial Layout [89], [90],
Hybrid Construction
[93], [94], [95], [96]

Glyphs
Per-Element Glyphs
[99], [100],
[101], [102],
Multi-Object Glyphs
[103], [104], [105]

Pixel-Oriented
Jigsaw Map [109],
Pixel Bar Charts [108],
Circle Segment [107]
Value & Relation
Dispaly [110]

Hierarchy-Based
Dimension
Hierarchy [113],
Topology-Based
Hierarchy [197], [198],
Others [115], [117]

Animation
GGob i[119],
TripAdvisor[ND]
[52],
Rolling the
Dice [120]

Evaluation
Scatterplot Guideline
[122], [123]
Parallel Coordinates
Effectiveness [124],
Animation [127]

View Transformation

Illustrative Rendering
Illustrative PCP [128],
Illuminated 3D Scatterplot [129],
PCP Transfer Function [130],
Magic Lens [132, 133]

Continuous Visual Representation
Continuous Scatterplot [134], [135]
Continuous Parallel Coordinates [136],
Splatterplots [138],
Splatting in Parallel Coordinates [136]

Accurate Color Blending
Hue-Preserving
Blending [140],
Weaving vs. Blending
[141]

Image Space Metrics
Clutter Reduction
[142], [143],
Pargnostics [144],
Pixnostics [145]

Visualization pipeline for HD data

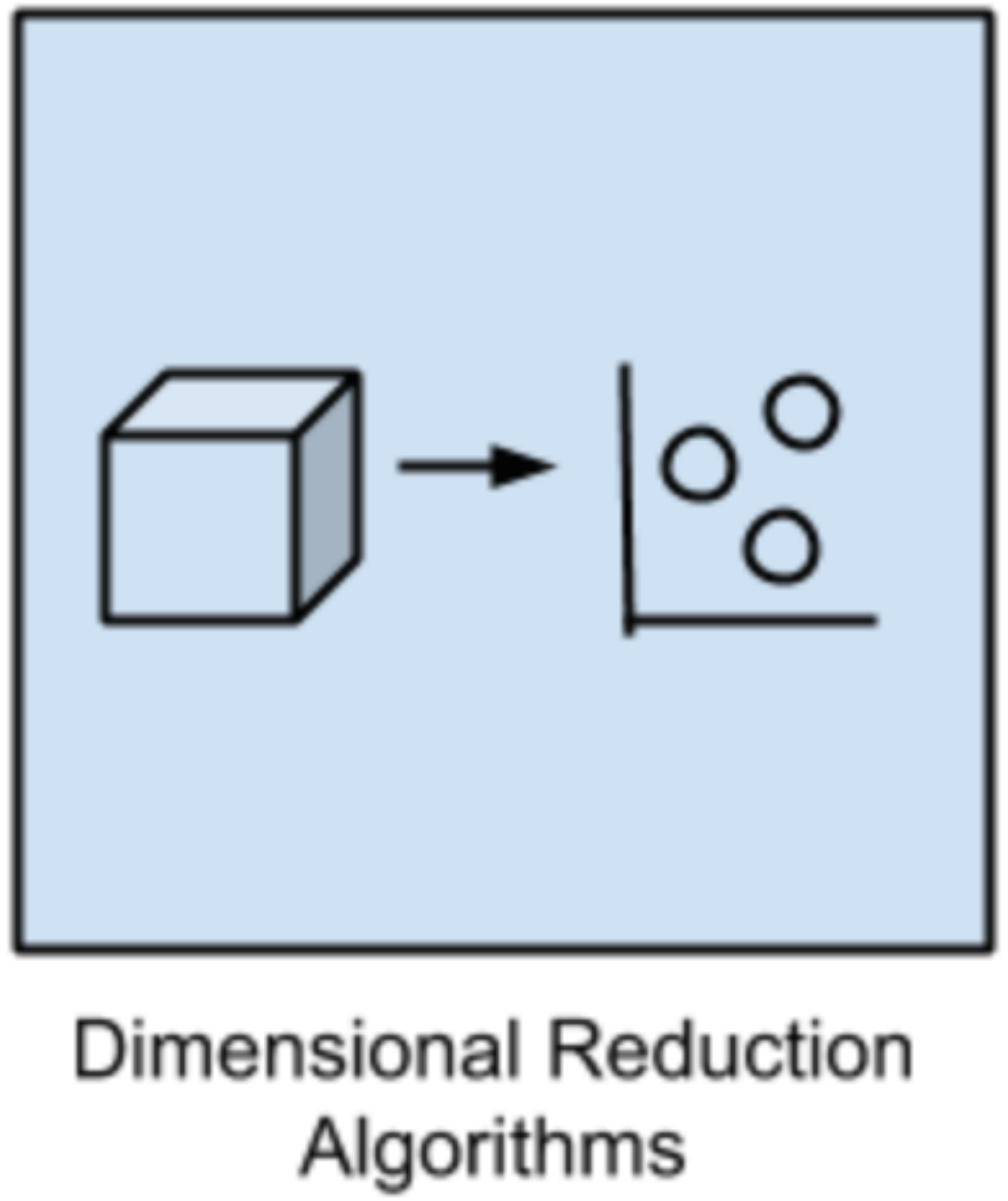
[LiuMaljovecWang2017]

ML in data transformation

Dimension Reduction	Subspace Clustering	Regression Analysis	Topological Data Analysis
Linear Projection [23], [25],	Dimension Space Exploration	Optimization &	Morse-Smale Complex
Nonlinear DR [26], [30],	[47], [48], [49],	Design Steering	[166], [168], [169], [170],
Control Points Projection [34], [37]	Subset of Dimension [51], [53],	[61], [62], [63],	Reeb Graph [174], [175], [181]
Distance Metric [38, 39],	Non-Axis-Parallel Subspace	Structural Summaries	Contour Tree [179, 180],
Precision Measures [42], [44]	[56], [57], [58]	[67], [68]	Topological Features [191], [192]

Dimensionality Reduction (DR)

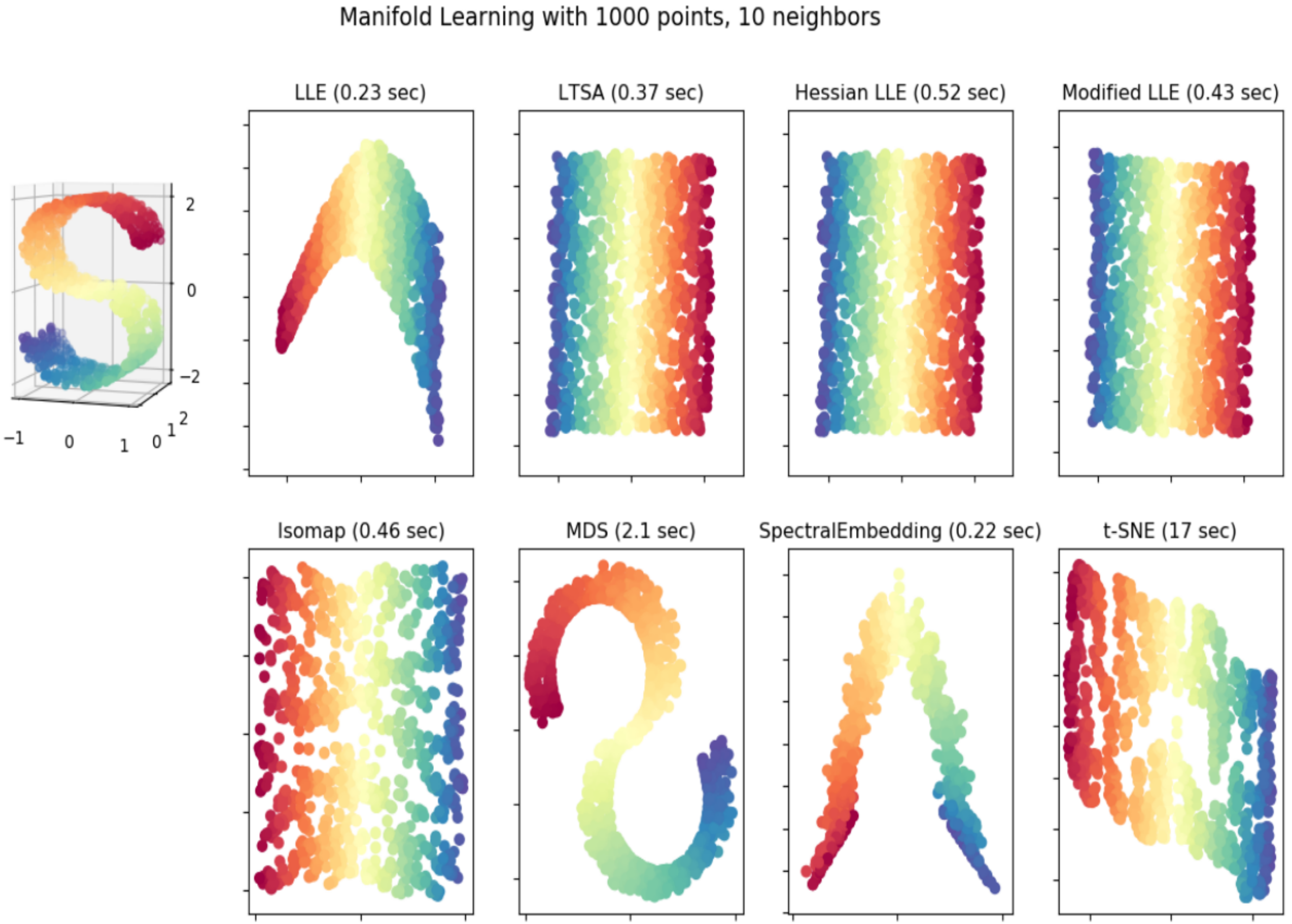
Vis+DR can be a semester worth of material…



- Seek and explore the inherent structure in data
- Unsupervised
- Data compression, summarization
- Pre-processing for vis and supervised learning
- Can be adapted for classification and regression
- Well-known DR algorithms:
  - Principal Component Analysis (PCA)
  - Principal Component Regression (PCR)
  - Partial Least Squares Regression (PLSR)
  - Multidimensional Scaling (MDS)
  - Projection Pursuit
  - Linear Discriminant Analysis (LDA)
  - Mixture Discriminant Analysis (MDA)
  - …

Linear vs nonlinear DR

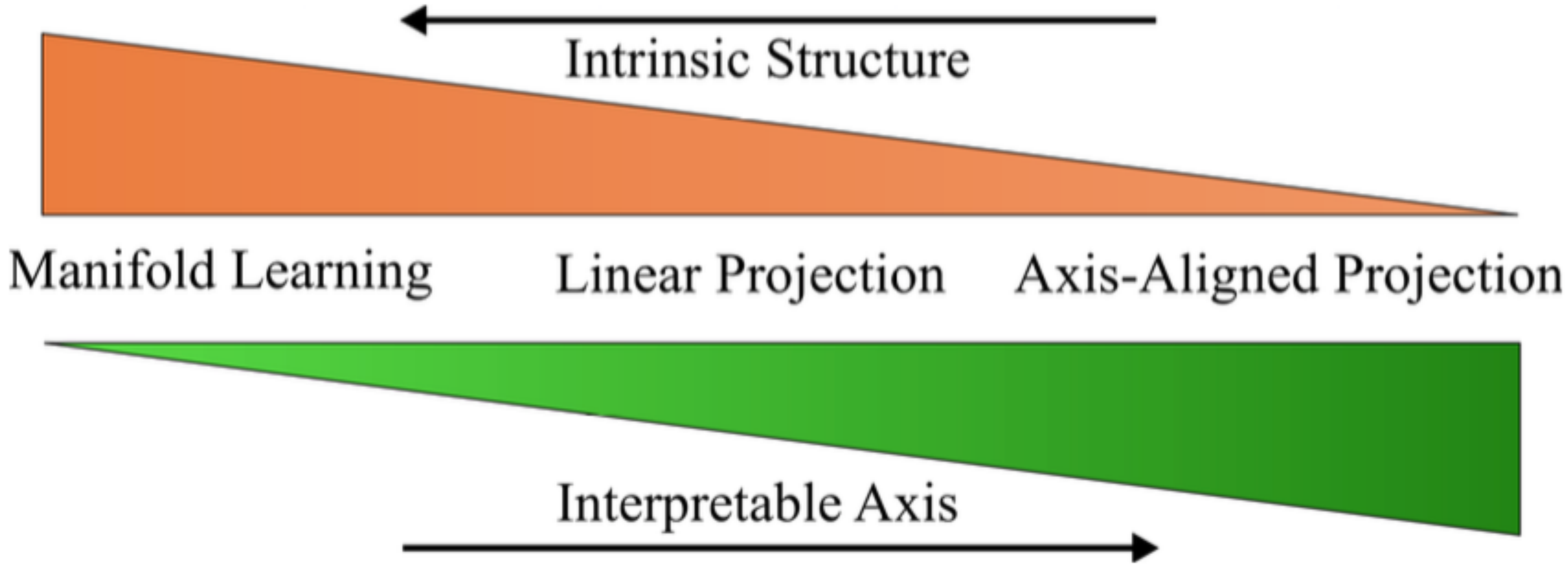
- Linear: Principal Component Analysis (PCA)
- Nonlinear DR, Manifold learning:
  - Isomap
  - Locally Linear Embedding (LLE)
  - Hessian Eigenmapping
  - Spectral Embedding
  - Multi-dimensional Scaling (MDS)
  - t-distributed Stochastic Neighbor Embedding (t-SNE)



Manifold learning

Source: http://scikit-learn.org/stable/modules/manifold.html

Interpretability trade off



[LiuMaljovecWang2017]

DR and Vis Overview

How do we proceed from here

- Give two case studies involving DR + Vis
  - Case 1: PCA + Vis (simple)
  - Case 2: SNE and t-SNE + Vis (more involved)
- We do not go through all (but some of) the mathematical details of these algorithms, but instead give a high-level overview of what the algorithm is trying to do
- You are encouraged to follow references and recommended readings to obtain in-depth understanding of these algorithms
- You can use these case studies to think about what might be a good final project

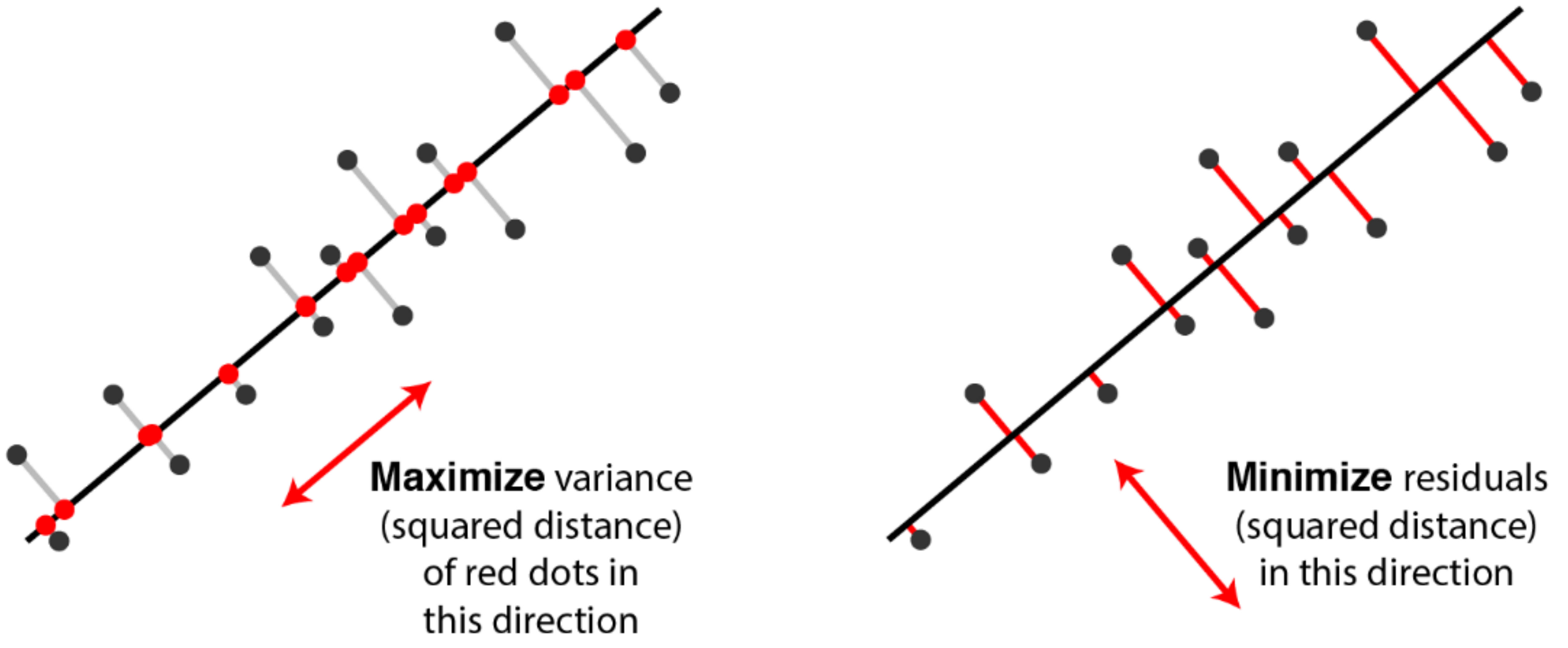
Vis + DR: PCA

A case study with a linear DR method

Three interpretation of PCA

PCA can be interpreted in 2 different ways:

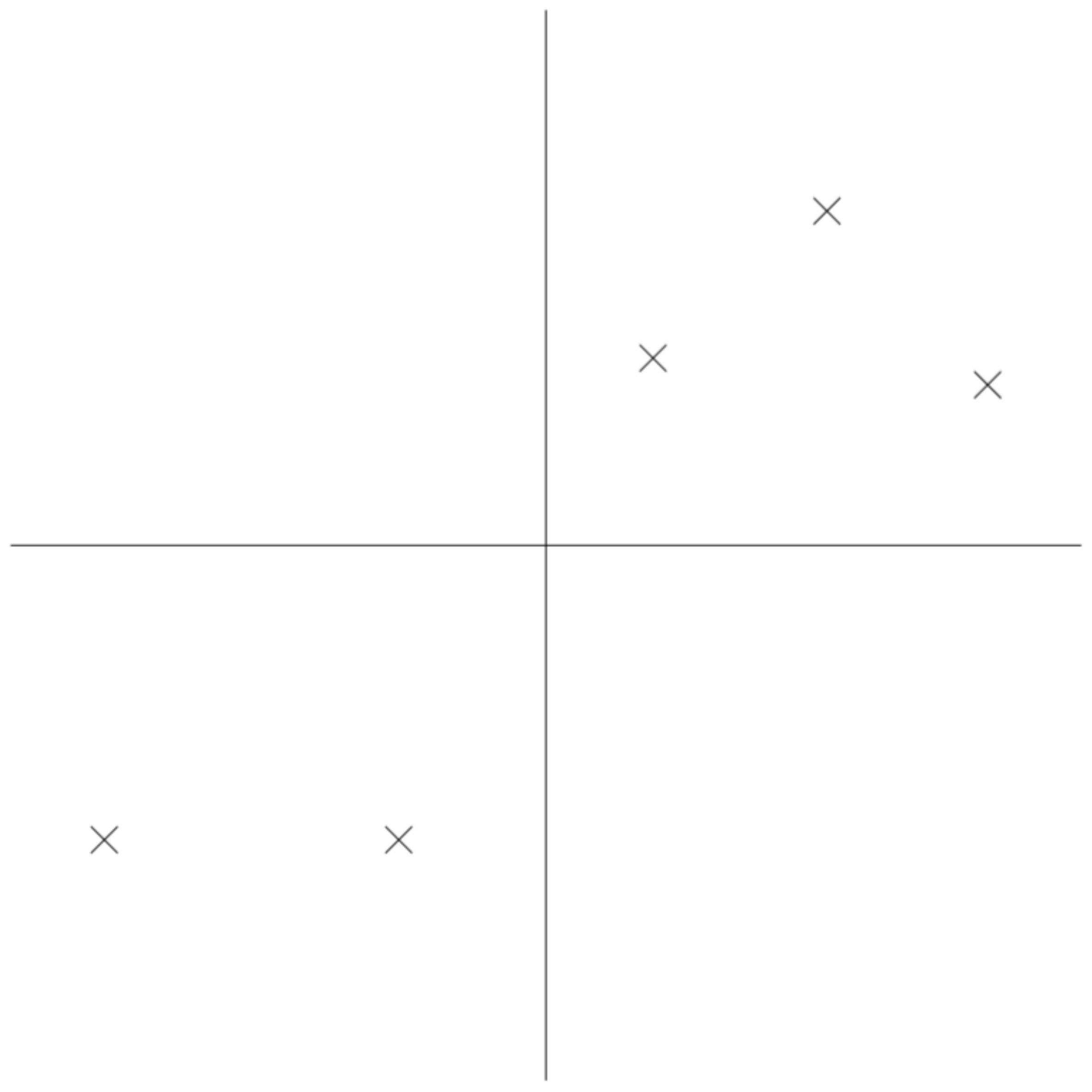
- Maximize the variance of projection along each component (dimension).
- Minimize the reconstruction error, that is, the squared distance between the original data and its projected coordinates.



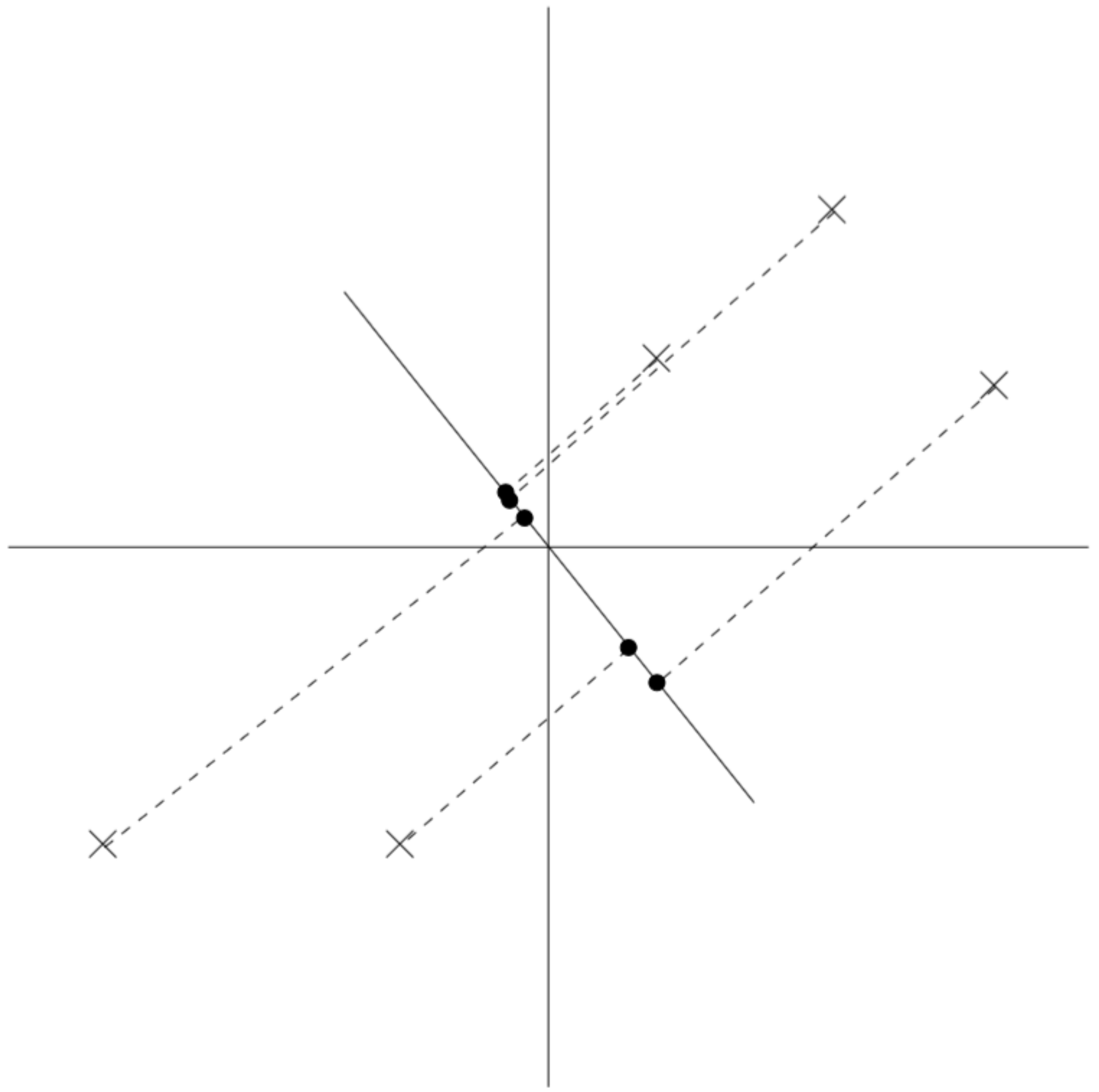
Two equivalent views of principal component analysis.

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

PCA at a glance



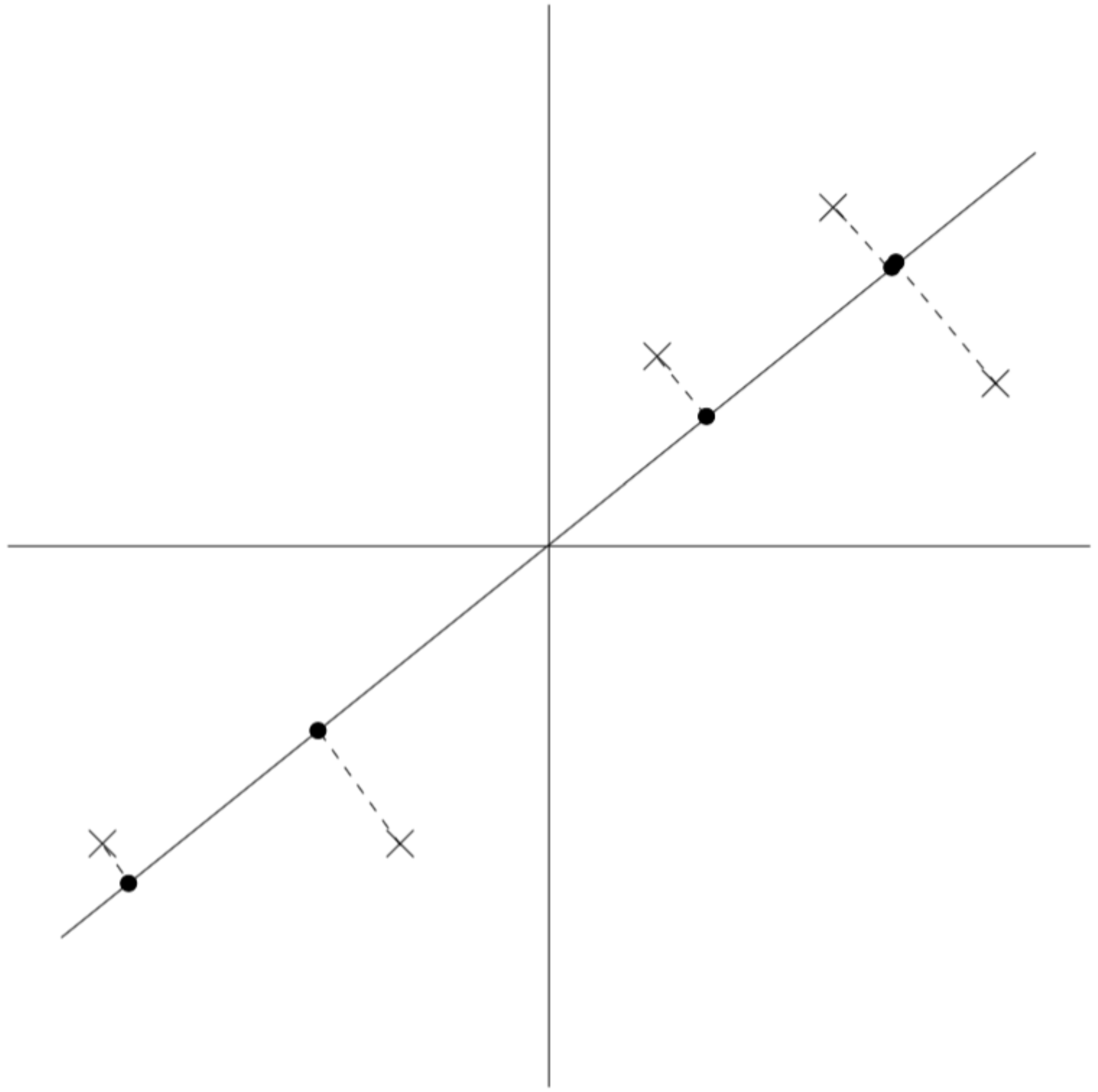
Data after normalization



A projection with small variance

Source: http://cs229.stanford.edu/notes/cs229-notes10.pdf

PCA at a glance



A projection with large variance

- PCA automatically choose project direction that maximizes the variance
- The direction of maximum variance in the input space happens to be the same as the principal eigenvector of the covariance matrix of the data
- PCA algorithm: finding the eigenvalues and eigenvectors of the covariance matrix.
- The eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset; this is the principle component.

Source: http://cs229.stanford.edu/notes/cs229-notes10.pdf

Eigenvalues and eigenvectors

For a given matrix $\mathbf{A}$, what are the vectors $\mathbf{x}$ for which the product $\mathbf{Ax}$ is a scalar multiple of $\mathbf{x}$? That is, what vectors $\mathbf{x}$ satisfy the equation

$$\mathbf{Ax} = \lambda\mathbf{x}$$

for some scalar λ?

Source: https://www.calvin.edu/~scofield/courses/m256/materials/eigenstuff.pdf

Eigen decomposition theorem

Let P be a matrix of eigenvectors of a given square matrix A and D be a diagonal matrix with the corresponding eigenvalues on the diagonal. Then, as long as P is a square matrix, A can be written as an eigen decomposition

$$A = P D P^{-1},$$

where D is a diagonal matrix. Furthermore, if A is symmetric, then the columns of P are orthogonal vectors.

http://mathworld.wolfram.com/EigenDecompositionTheorem.html

Covariance matrix

$$Q = XX^T = \begin{bmatrix} \mathbf{x}_1 - \bar{\mathbf{x}} & \mathbf{x}_2 - \bar{\mathbf{x}} & \cdots & \mathbf{x}_n - \bar{\mathbf{x}} \end{bmatrix} \begin{bmatrix} (\mathbf{x}_1 - \bar{\mathbf{x}})^T \\ (\mathbf{x}_2 - \bar{\mathbf{x}})^T \\ \vdots \\ (\mathbf{x}_n - \bar{\mathbf{x}})^T \end{bmatrix}$$

X: data; each col is a data point; each row is a dim.
Don't want to explicitly compute Q: can be huge!
Instead, using SVD, singular value decomposition.

Singular value decomposition (SVD)

Any m x n matrix X can be decomposed into three matrices:

$$X = U\Sigma V^T$$

- U is m x m and its columns are orthonormal vectors (i.e. perpendicular)
- Σ is n x n and its columns are orthonormal vectors
- D is m x n diagonal and its diagonal elements are called the singular values of X

Relation between PCA and SVD

Simply put, the PCA viewpoint requires that one compute the eigenvalues and eigenvectors of the covariance matrix, which is the product $\mathbf{X}\mathbf{X}^\top$, where $\mathbf{X}$ is the data matrix. Since the covariance matrix is symmetric, the matrix is diagonalizable, and the eigenvectors can be normalized such that they are orthonormal:

$$\mathbf{X}\mathbf{X}^\top = \mathbf{W}\mathbf{D}\mathbf{W}^\top$$

On the other hand, applying SVD to the data matrix $\mathbf{X}$ as follows:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

and attempting to construct the covariance matrix from this decomposition gives

$$\mathbf{X}\mathbf{X}^\top = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top)(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top)^\top$$
$$\mathbf{X}\mathbf{X}^\top = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top)(\mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top)$$

and since $\mathbf{V}$ is an orthogonal matrix ($\mathbf{V}^\top\mathbf{V} = \mathbf{I}$),

$$\mathbf{X}\mathbf{X}^\top = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^\top$$

and the correspondence is easily seen (the square roots of the eigenvalues of $\mathbf{X}\mathbf{X}^\top$ are the singular values of $\mathbf{X}$, etc.)

https://math.stackexchange.com/questions/3869/what-is-the-intuitive-relationship-between-svd-and-pca

Performing SVD on data matrix

X is the (normalized) data matrix, perform SVD on X:

$$X = UDV^T$$

- The columns of U are the eigenvectors of covariance matrix: XX^T
- The columns of V are the eigenvectors of X^T X
- The squares of the diagonal elements of D are the eigenvalues of XX^T and X^T X

PCA related readings

- Many PCA lectures are available on the web
- Reading materials
  - http://www.cse.psu.edu/~rtc12/CSE586Spring2010/lectures/pcaLectureShort.pdf
  - http://cs229.stanford.edu/notes/cs229-notes10.pdf
- Things you should pay attention when using PCA
  - Make sure the data is centered: normalize mean and variance

Using PCA with scikit-learn

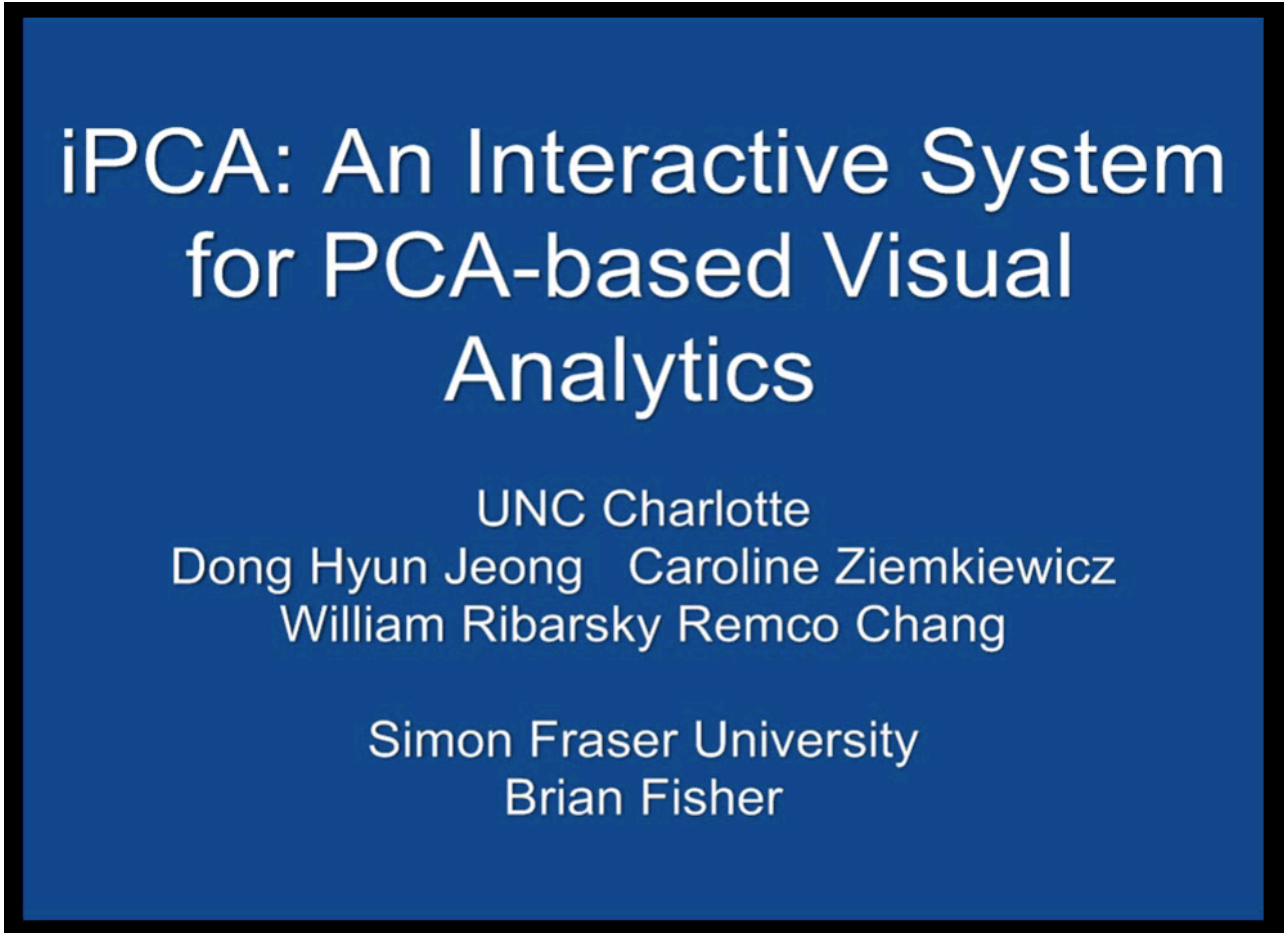
```
import numpy as np
from sklearn.decomposition import PCA
X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
pca = PCA(n_components=2)
pca.fit(X)


print(pca.explained_variance_ratio_)

print(pca.singular_values_)
```

http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

iPCA: interactive PCA



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

iPCA extension: collaborative sys

Button	Meaning	Button	Meaning
	Go back to the initial state		Delete the selected item(s)
	Individual item selection		Partition the selected item(s) into a new workspace
	Range item(s) selection		Close the application
	Manipulation		Create a new application
	Trail enable – on/ off		Rotate the application
	Cancel the selected item(s)		Make the sliderbar panel appear / disappear

[JeongRibarskyChang2009]: Designing a PCA-based Collaborative Visual Analytics System

Vis + DR: t-SNE

A case study with a nonlinear DR method

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

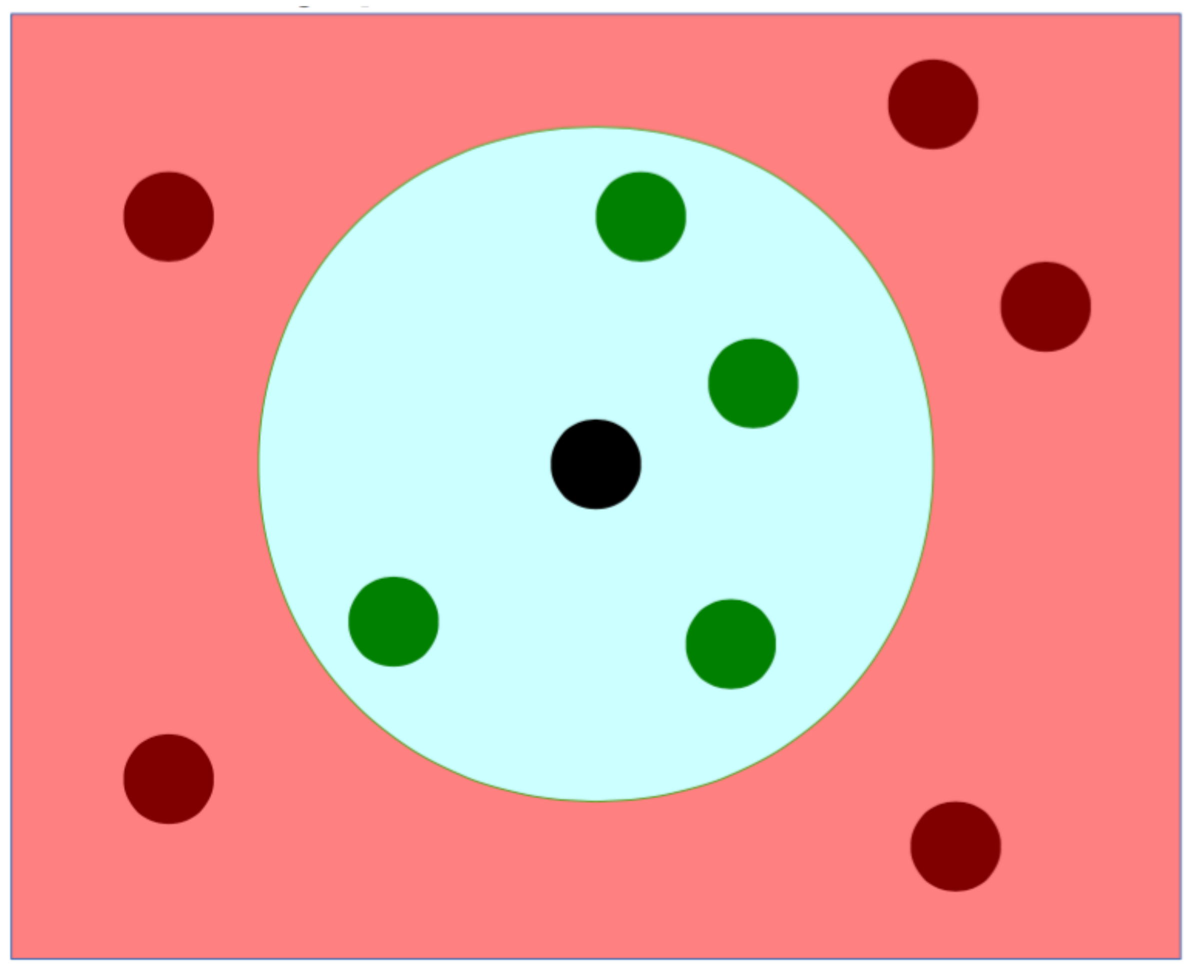
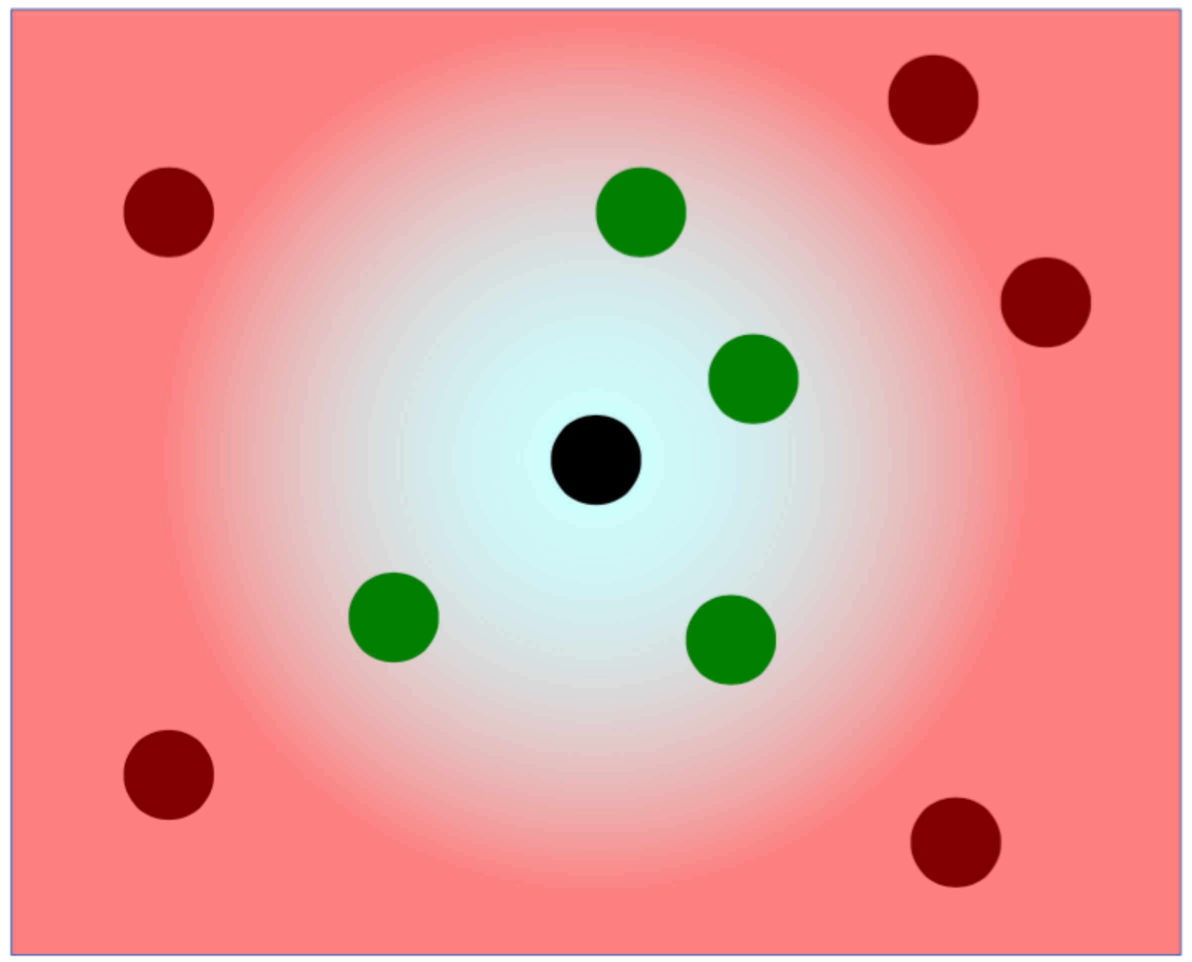
DR: preserving distances

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

- 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

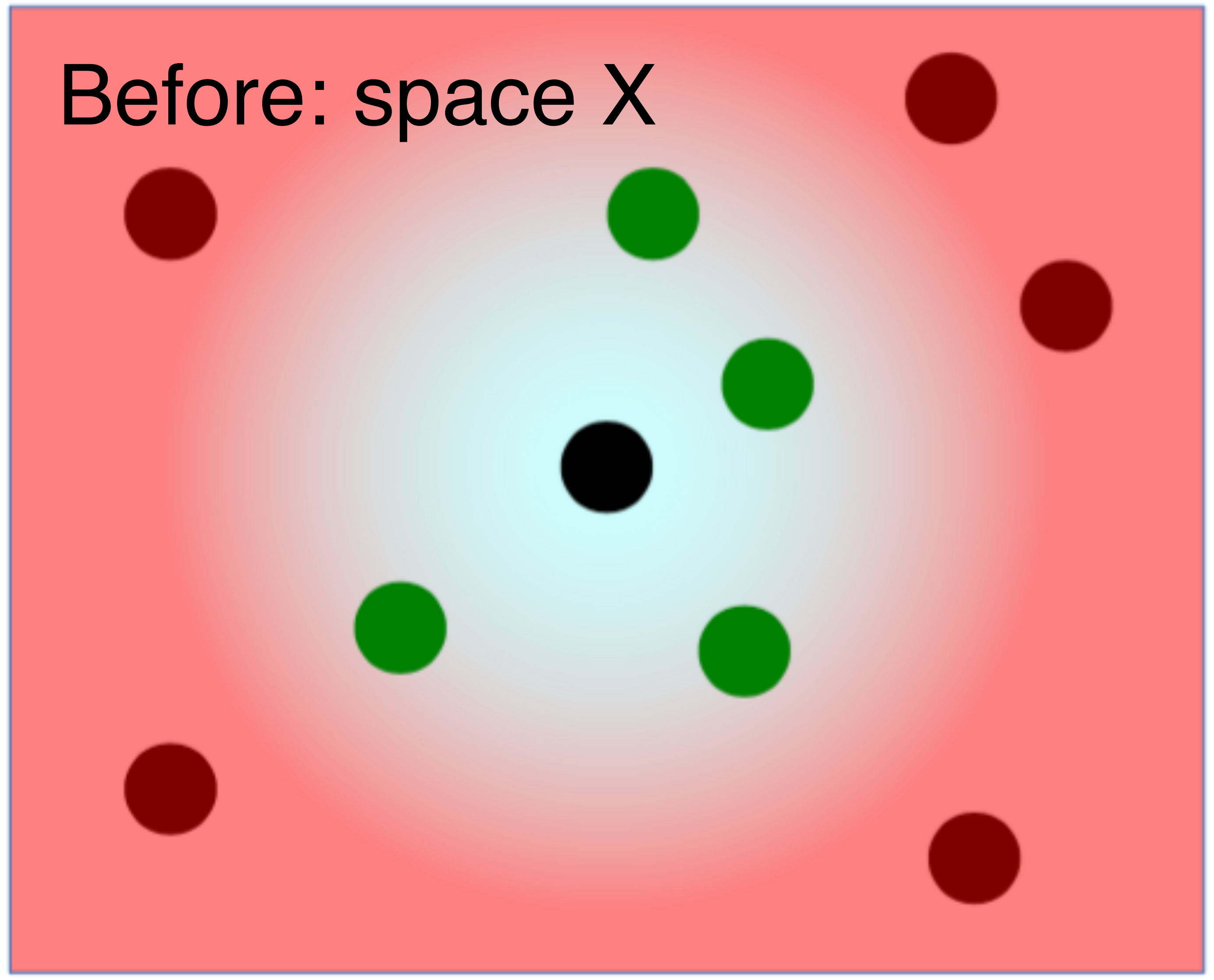



Probabilistic neighborhood

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

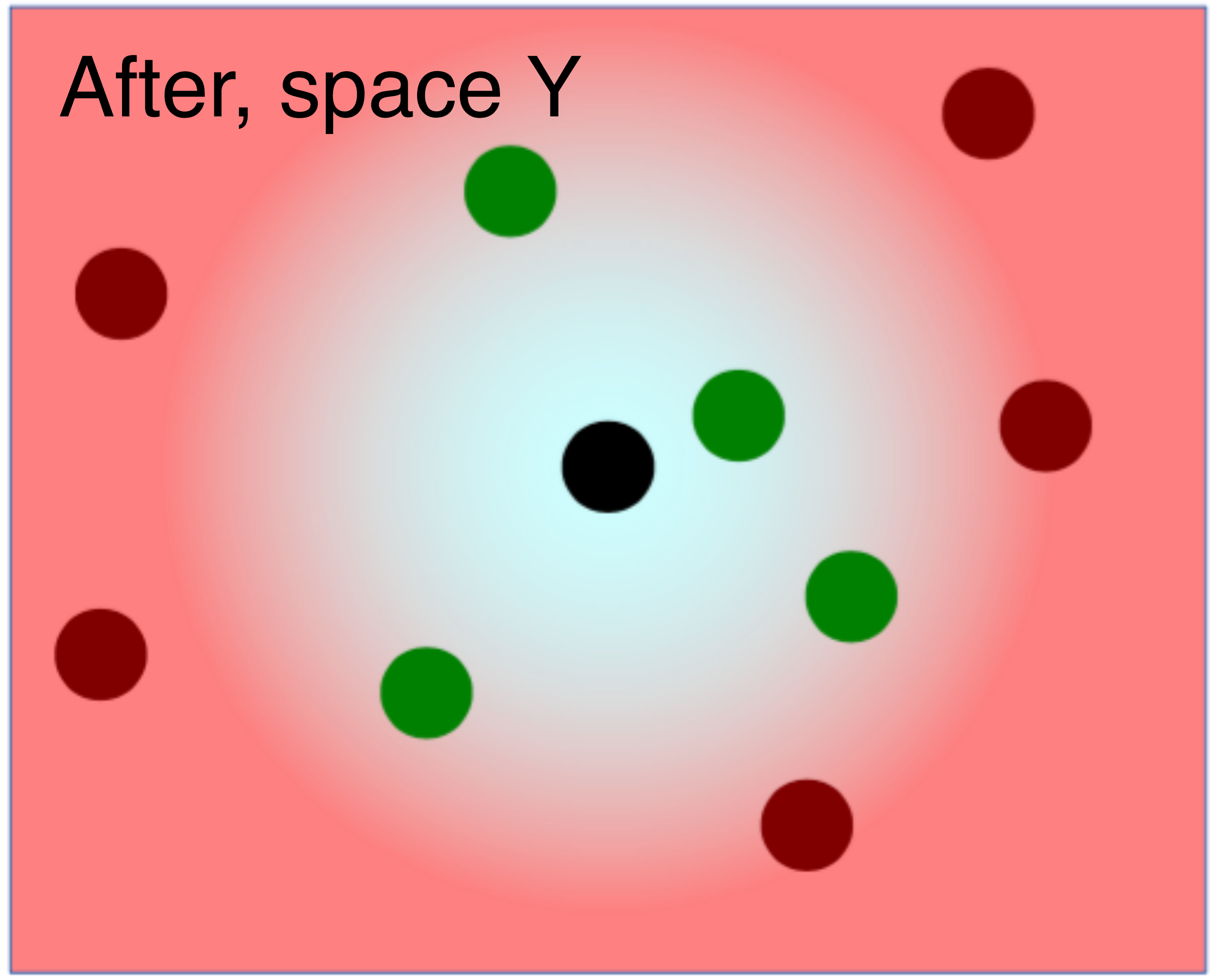
$$p_{ij} = \frac{exp(-d^2_{ij})}{\sum_{k \neq i} exp(-d^2_{ik})}$$

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

$$C = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}$$

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.

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

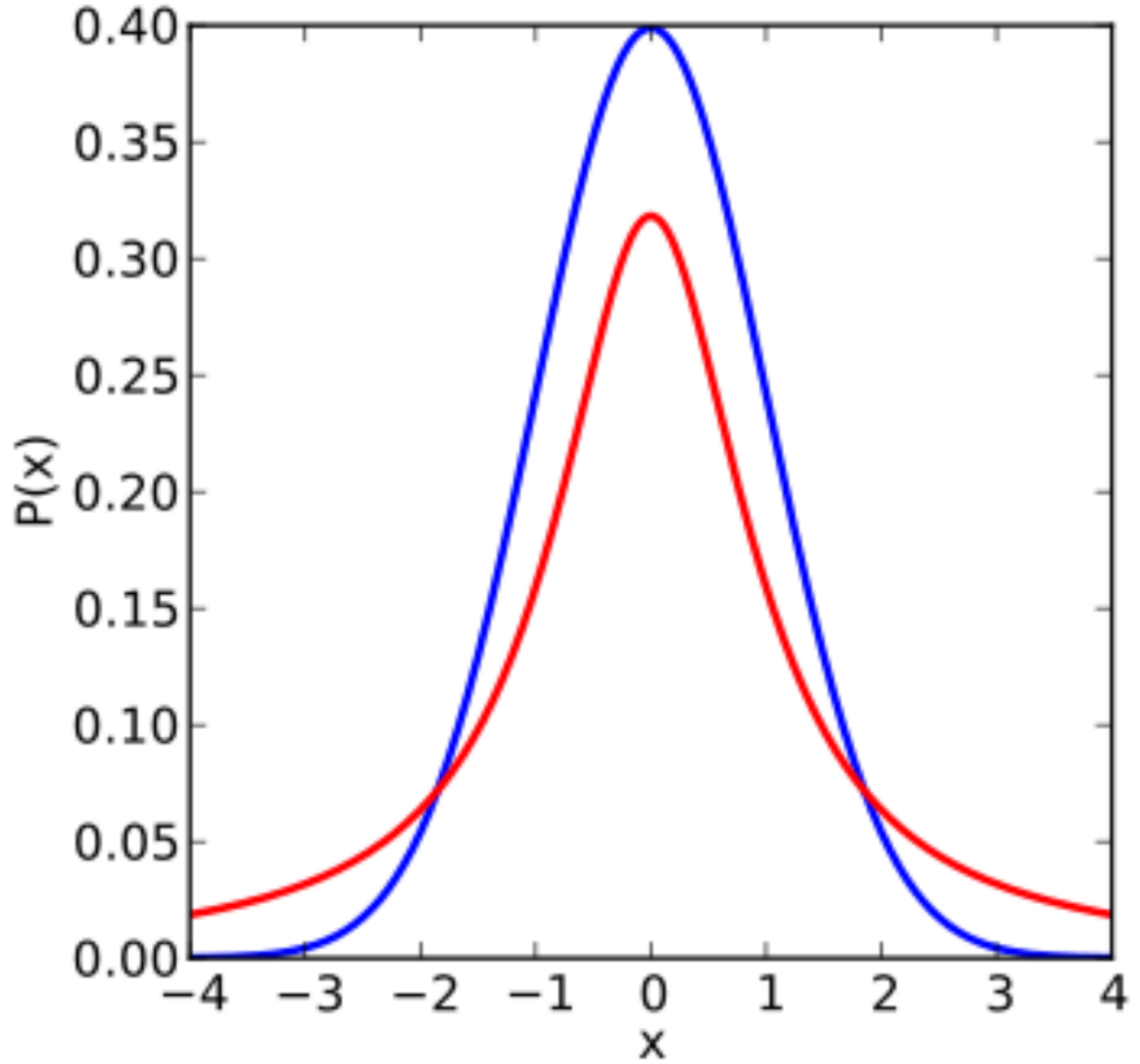
$$\frac{\partial C}{\partial y_i} = 2 \sum_j (y_i - y_j)(p_{ij} - q_{ij} + p_{ji} - q_{ji})$$

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

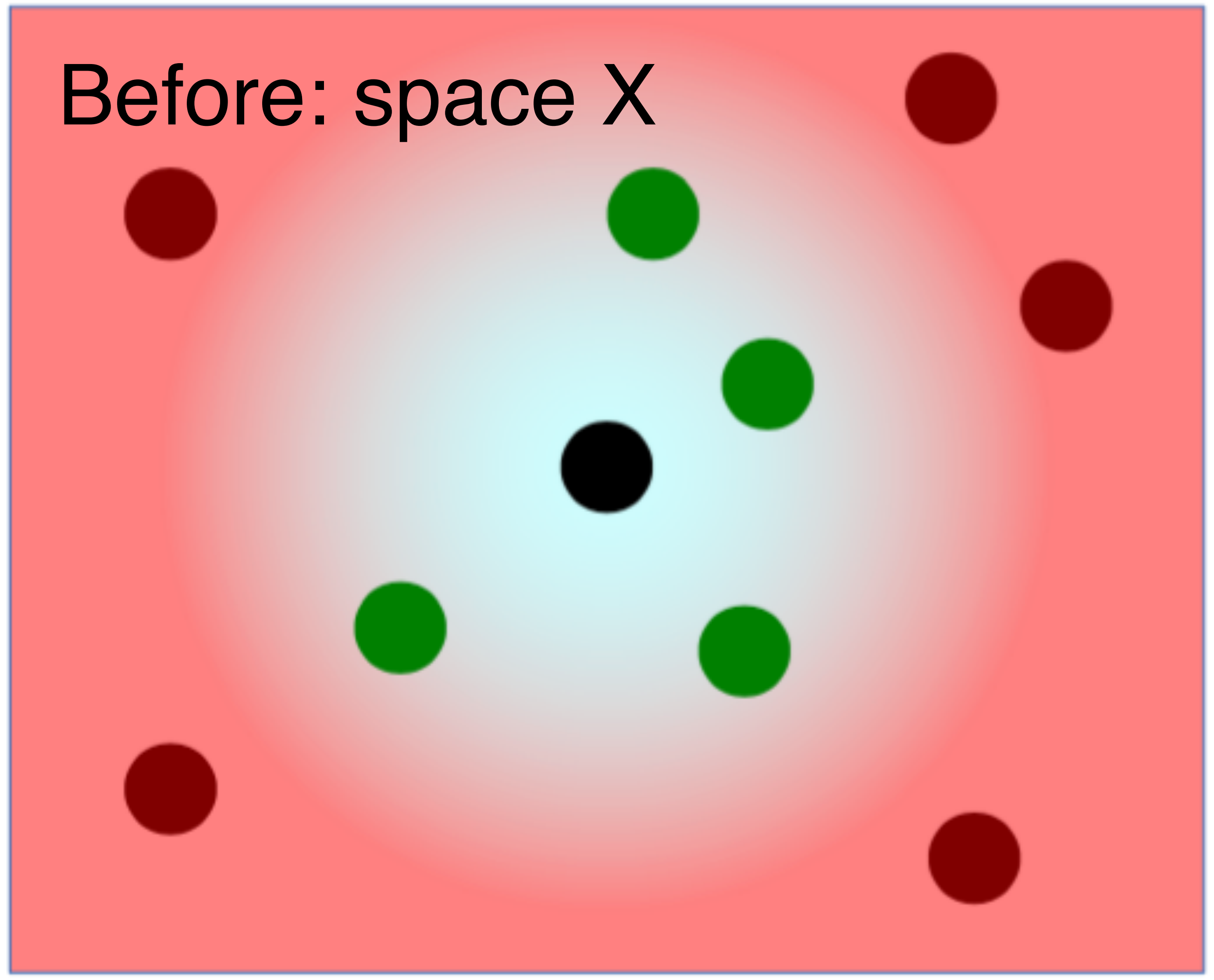
t-distributed SNE

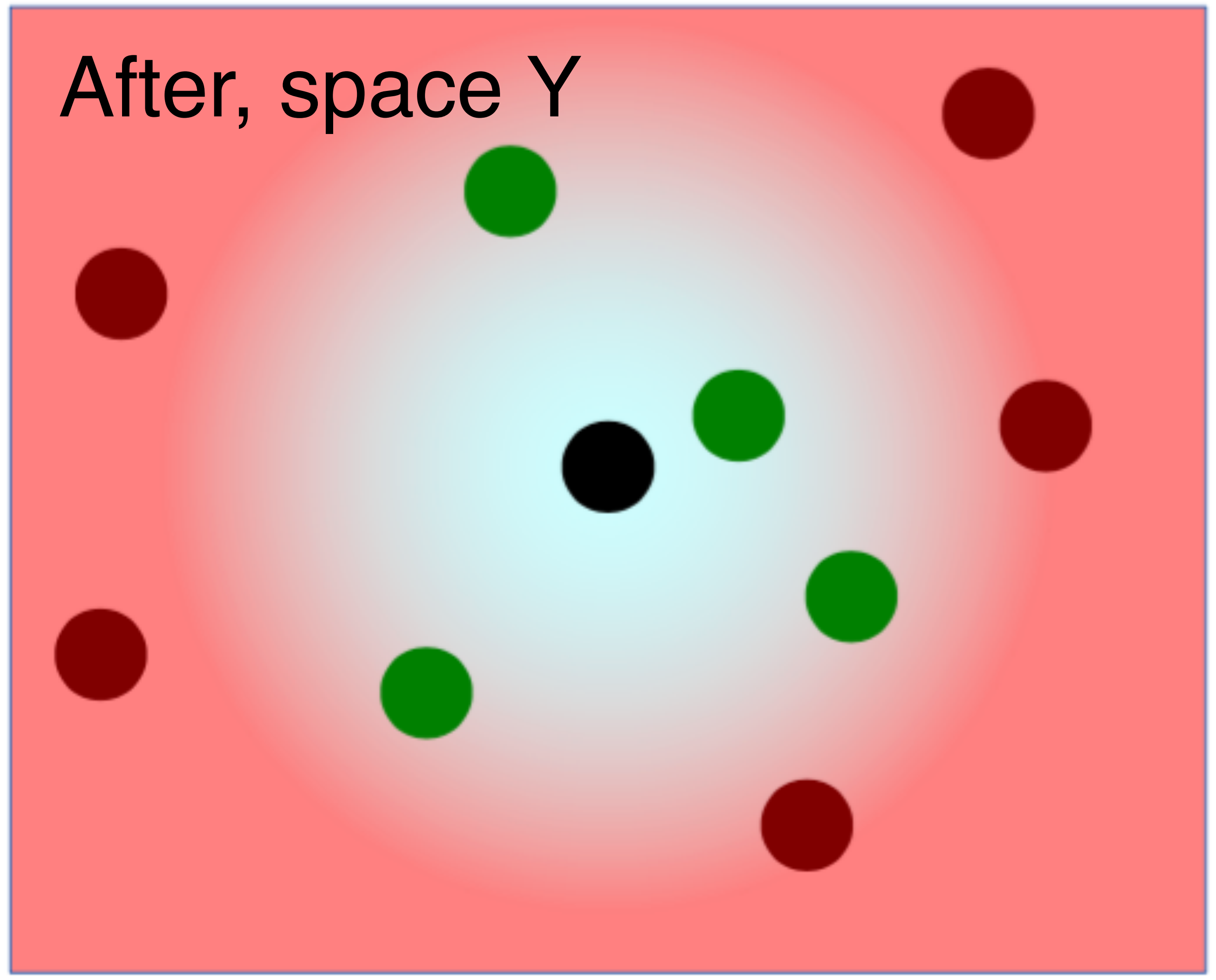
- Avoids crowding problem by using a more heavy-tailed neighborhood distribution in the low-dim output space than in the input space.
- Neighborhood probability falls off less rapidly; less need to push some points far off and crowd remaining points close together in the center.
- Use student-t distribution 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





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

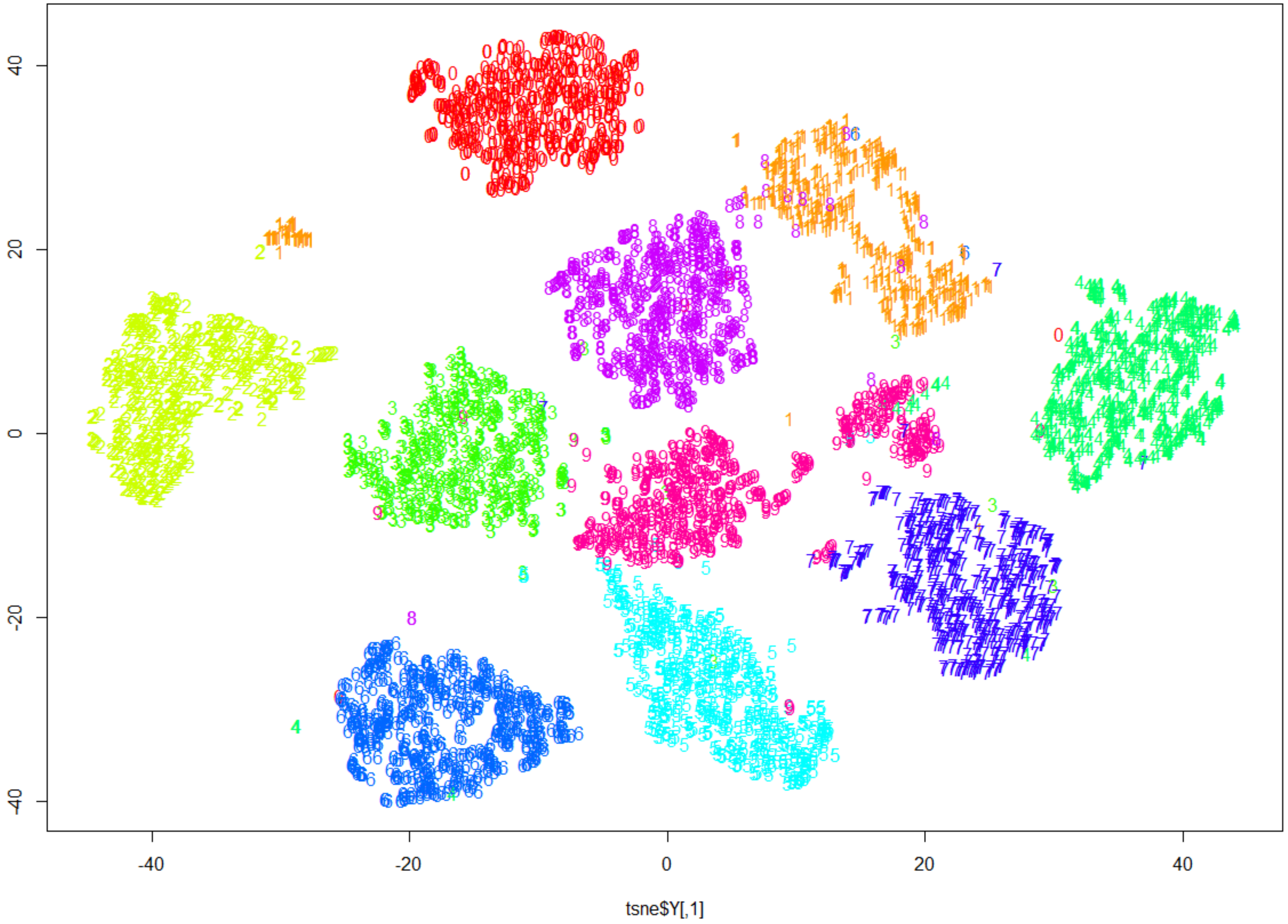
$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

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

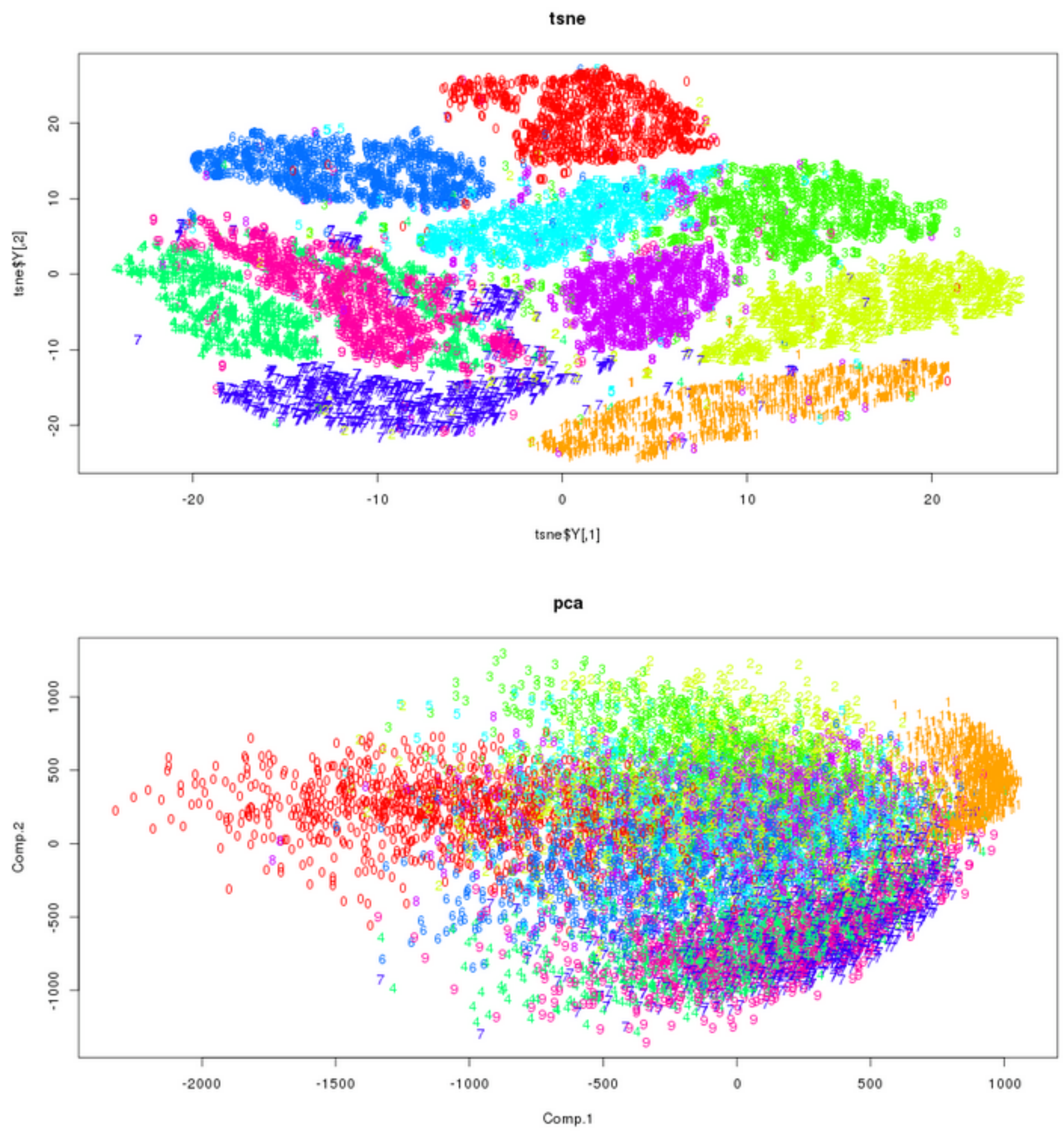
$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$$

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

Classic t-SNE result



t-SNE vs PCA



tsne
tsne$Y[,2]
tsne$Y[,1]
-20
-10
0
10
20
pca
Comp.2
Comp.1
-2000
-1500
-1000
-500
0
500
1000

t-SNE

- 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)

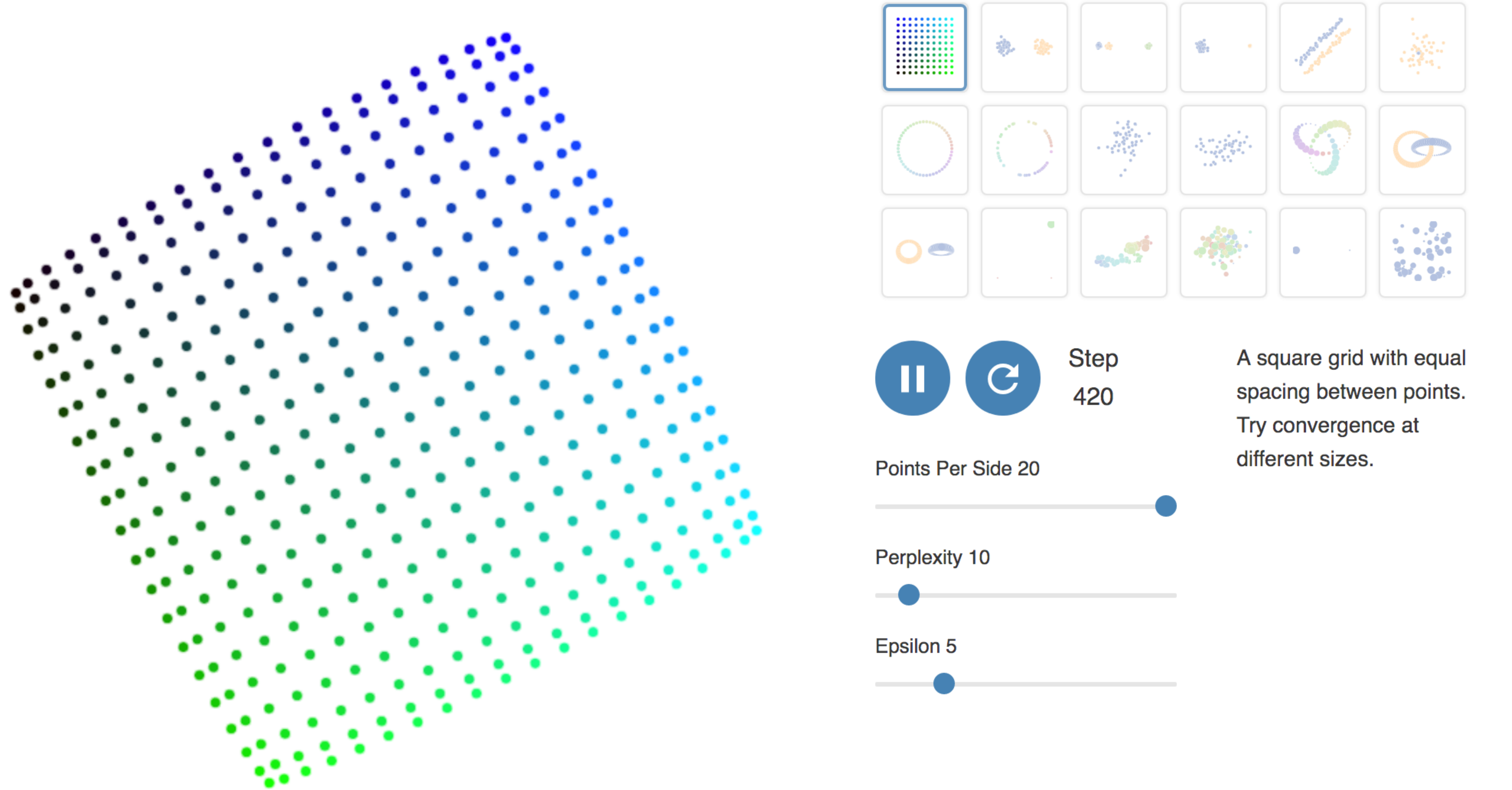
Source: https://distill.pub/2016/misread-tsne/

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

- 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/

Getting ready for Project 1

- Scikit-learn tutorial:
  - http://scikit-learn.org/stable/tutorial/basic/tutorial.html
- UMAP:
  - https://umap-learn.readthedocs.io/en/latest/
- Install and read the documentation of kepler-mapper:
  - https://github.com/MLWave/kepler-mapper
- Interactive Data Visualization for the Web, 2nd Ed.
  - http://alignedleft.com/work/d3-book-2e

Potential Final Projects

- Inspired by:
  - http://setosa.io/ev/principal-component-analysis/
  - https://distill.pub/2016/misread-tsne/
- ExtendingEmbedding 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?

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