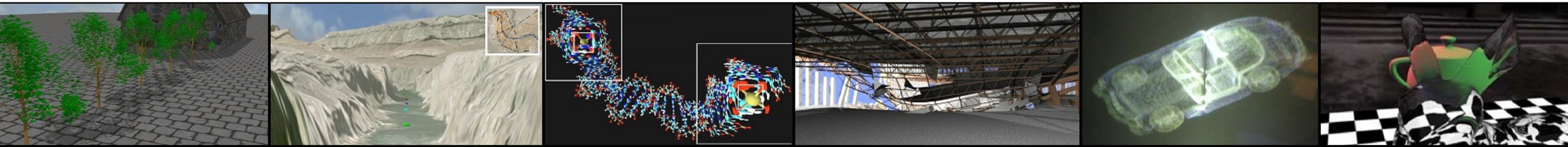
CIS 4930/6930-002
DATA VISUALIZATION

Machine Learning & Data Mining

Mustafa Hajij
University of South Florida



REMINDERS

4/11/2018 – Paper 4 Review Due

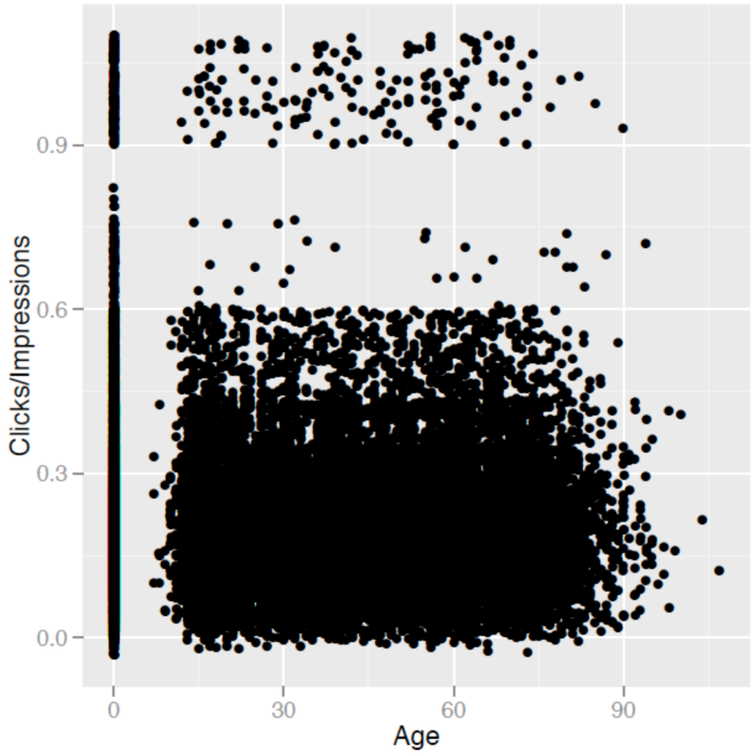
4/16/2018 – Project 7 Peer Review Due

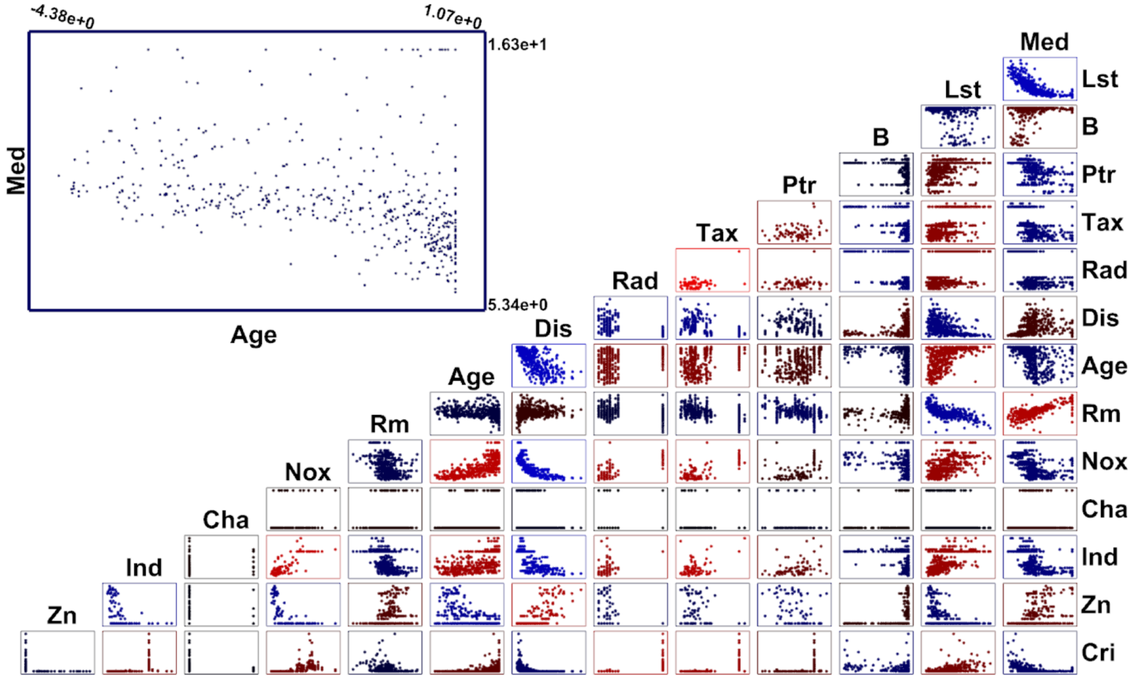
4/23/2018 – Project 8 Due

THE PROBLEM, ONCE AGAIN

Number of data points and attributes are large

Limited human visual capacity





THE SOLUTION

use the machine to do most of the work

BUT HOW?

STATISTICS

As we saw, statistics can help in a number of ways, but they don't give you everything you need/want

DATA MINING TOOLS

Help to interpret data by (hopefully) reducing them to their core components/features

WE'LL (BRIEFLY) TALK ABOUT

Dimensionality Reduction

Classifiers

Regression

Clustering

Dimensionality Reduction

Comes in 2 basic flavors, linear and nonlinear

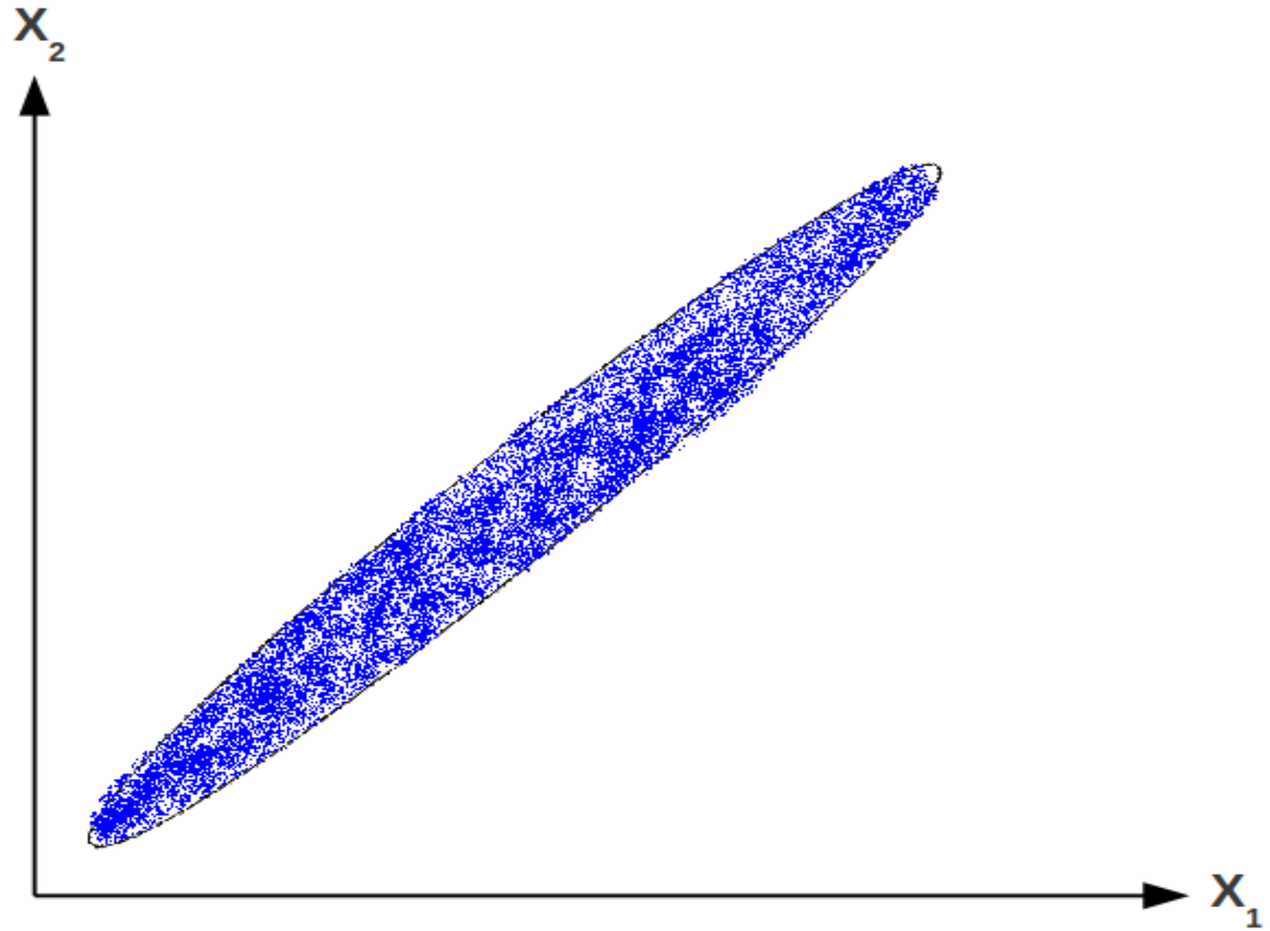
Linear Dimensionality reduction: Principal Component Analysis (PCA)

Given a set of data points embedded in any number of dimensions

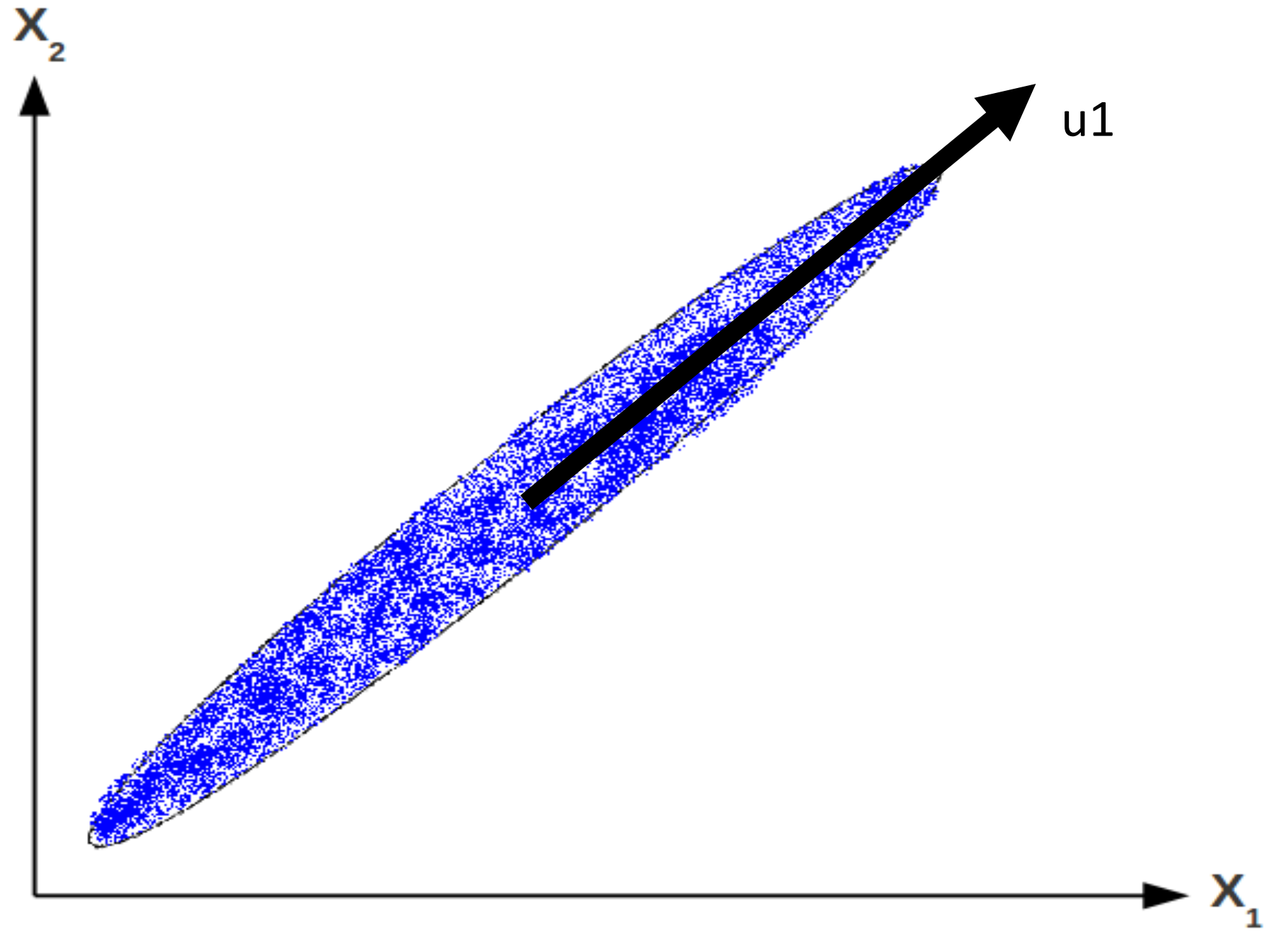
Find the vector (direction) with the largest variance

Repeat for additional vectors, finding the next largest variance orthogonal to all previous principal components

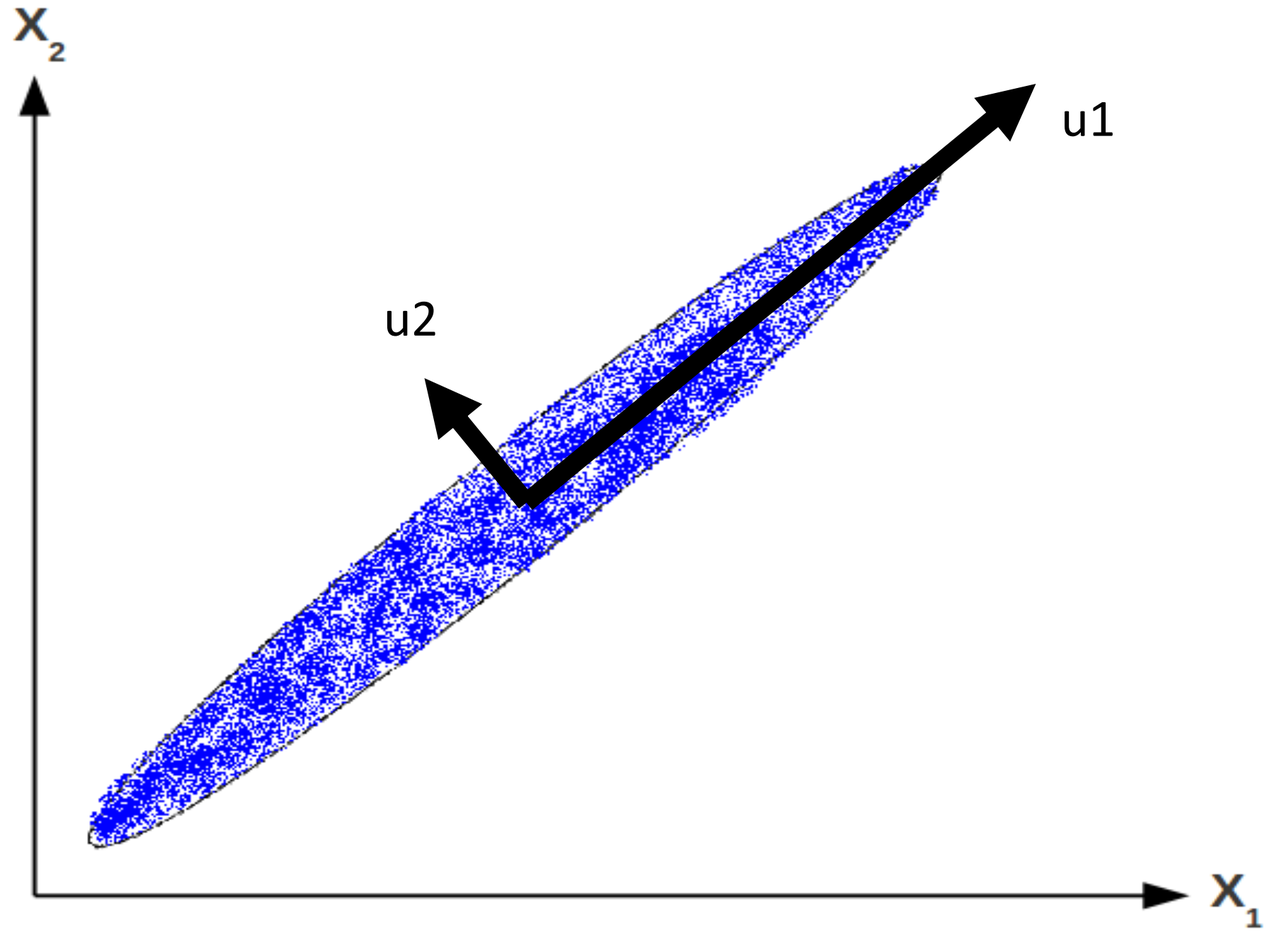
PCA: AN EXAMPLE



PCA: AN EXAMPLE



PCA: AN EXAMPLE

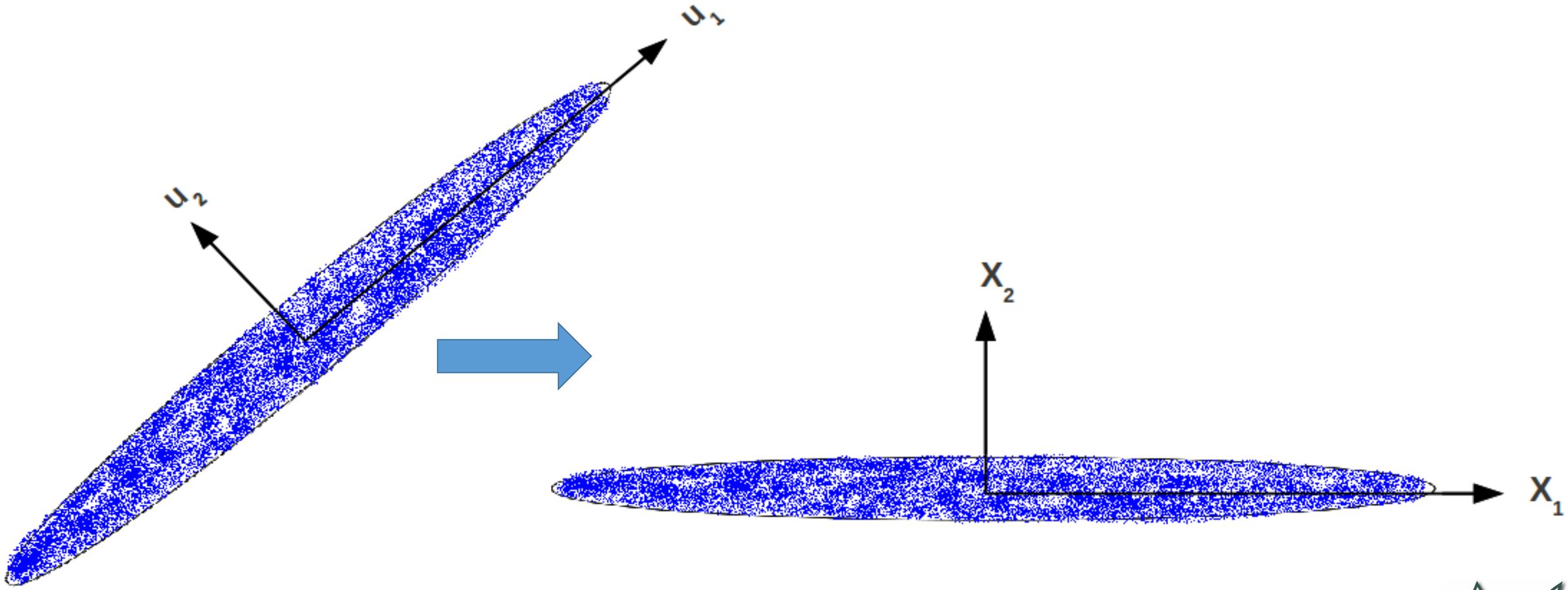


PCA Provides

A new basis that maximizes variance, where the length of the vectors indicates variance in that direction

Common practice is to reproject points by subtracting the mean and adjusting basis

PCA: AN EXAMPLE



PCA Provides

If data is higher than 2D, the first 2 principal components can be selected (or 3 for 3D)

WHAT DO WE GAIN/LOSE…

WHEN IS THIS A GOOD/BAD IDEA…

In the 2D case?

In an nD case?

Principal Component Analysis (PCA)

Can be calculated by finding the eigenvectors $O(d^3)$ of the covariance matrix $O(nd^2)$

Practically speaking, the covariance matrix finding dominates computation, since $n \gg d$

Subsampling points can improve performance, as long as the sampling is *good*

Nonlinear Dimensionality Reduction: Multidimensional Scaling (MDS)

Given pairwise distances/dissimilarities

Find a map that preserves distances

MDS

Many “flavors”

We’ll look at classical MDS

Basic MDS Process

Given a dissimilarity (distance) matrix $D = (d_{ij})$

MDS seeks to find an embedding in P dimensions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in R^P$, so that $d_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$

For certain types of dissimilarity matrices, this exists for some large P

More generally, we want $d_{ij} \approx \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$, as close as possible

Basic MDS Process

The mapping from matrix $D = (d_{ij})$ to $d_{ij} \approx \|\boldsymbol{x_i} - \boldsymbol{x_j}\|$ is found using eigenvector/eigenvalue decomposition of the dissimilarity matrix

Although this is a linear operation, the effect to data is nonlinear

METRIC VS. NON-METRIC SPACE

Distance and dissimilarity are defined for any pair of objects in any space. A distance function is also called metric if it satisfies…

$d(x, y) \geq 0,$	non-negativity
$d(x, y) = 0 \; iff \; x = y$	identity
$d(x, y) = d(y, x)$	symmetry
$d(x, z) \leq d(x, y) + d(y, z)$	triangle inequality

Metric vs. Non-metric space

Given a set of dissimilarities, one can ask whether these values are distances, and whether the space they define is metric or not

Why does this matter?

Non-metric data requires special treatment

CLASSICAL MDS EXAMPLE: AIRLINE DISTANCES

TABLE 13.2. *Airline distances (km) between 18 cities. Source: Atlas of the World, Revised 6th Edition, National Geographic Society, 1995, p. 131.*

	Beijing	Cape Town	Hong Kong	Honolulu	London	Melbourne
Cape Town	12947					
Hong Kong	1972	11867				
Honolulu	8171	18562	8945			
London	8160	9635	9646	11653		
Melbourne	9093	10338	7392	8862	16902	
Mexico	12478	13703	14155	6098	8947	13557
Montreal	10490	12744	12462	7915	5240	16730
Moscow	5809	10101	7158	11342	2506	14418
New Delhi	3788	9284	3770	11930	6724	10192
New York	11012	12551	12984	7996	5586	16671
Paris	8236	9307	9650	11988	341	16793
Rio de Janeiro	17325	6075	17710	13343	9254	13227
Rome	8144	8417	9300	12936	1434	15987
San Francisco	9524	16487	11121	3857	8640	12644
Singapore	4465	9671	2575	10824	10860	6050
Stockholm	6725	10334	8243	11059	1436	15593
Tokyo	2104	14737	2893	6208	9585	8159

	Mexico	Montreal	Moscow	New Delhi	New York	Paris
Montreal	3728					
Moscow	10740	7077				
New Delhi	14679	11286	4349			
New York	3362	533	7530	11779		
Paris	9213	5522	2492	6601	5851	

CLASSICAL MDS EXAMPLE: AIRLINE DISTANCES

Negative eigenvalues indicate Airline distance is non-metric

Should only use at most first 3 eigenvectors (negative eigenvalues produce resulting in the imaginary domain)

TABLE 13.6. *Eigenvalues of* B *and the eigenvectors corresponding to the first three largest eigenvalues (in red) for the airline distances example.*

	Eigenvalues	Eigenvectors		
1	471582511	0.245	–0.072	0.183
2	316824787	0.003	0.502	-0.347
3	253943687	0.323	–0.017	0.103
4	–98466163	0.044	–0.487	-0.080
5	–74912121	–0.145	0.144	0.205
6	–47505097	0.366	–0.128	–0.569
7	31736348	–0.281	–0.275	–0.174
8	–7508328	–0.272	–0.115	0.094
9	4338497	–0.010	0.134	0.202
10	1747583	0.209	0.195	0.110
11	–1498641	–0.292	–0.117	0.061
12	145113	–0.141	0.163	0.196
13	–102966	–0.364	0.172	–0.473
14	60477	–0.104	0.220	0.163
15	–6334	–0.140	–0.356	–0.009
16	–1362	0.375	0.139	–0.054
17	100	–0.074	0.112	0.215
18	0	0.260	–0.214	0.173

CLASSICAL MDS EXAMPLE: AIRLINE DISTANCES

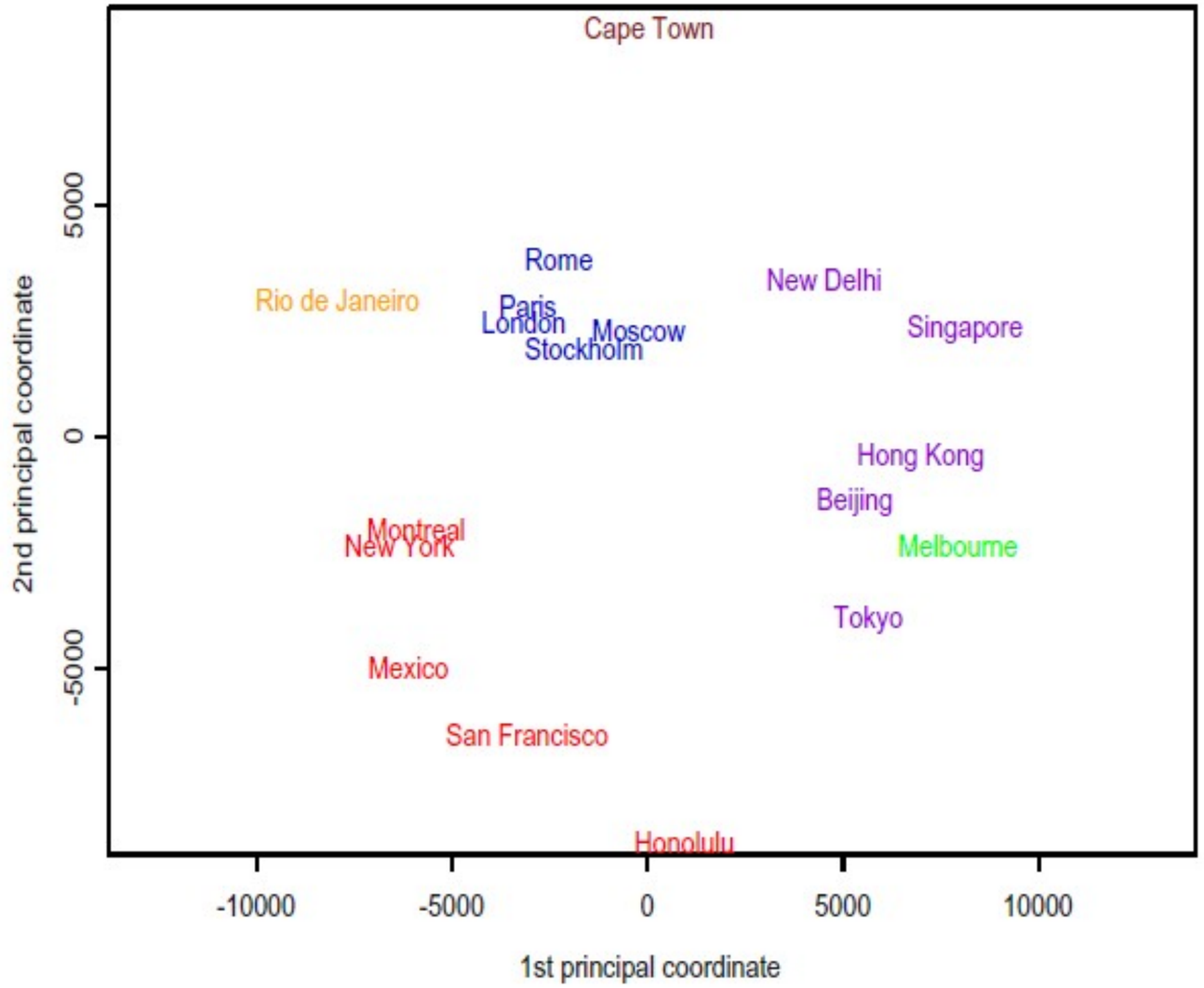


FIGURE 13.1. *Two-dimensional map of 18 world cities using the classical scaling algorithm on airline distances between those cities. The colors*

CLASSICAL MDS EXAMPLE: AIRLINE DISTANCES

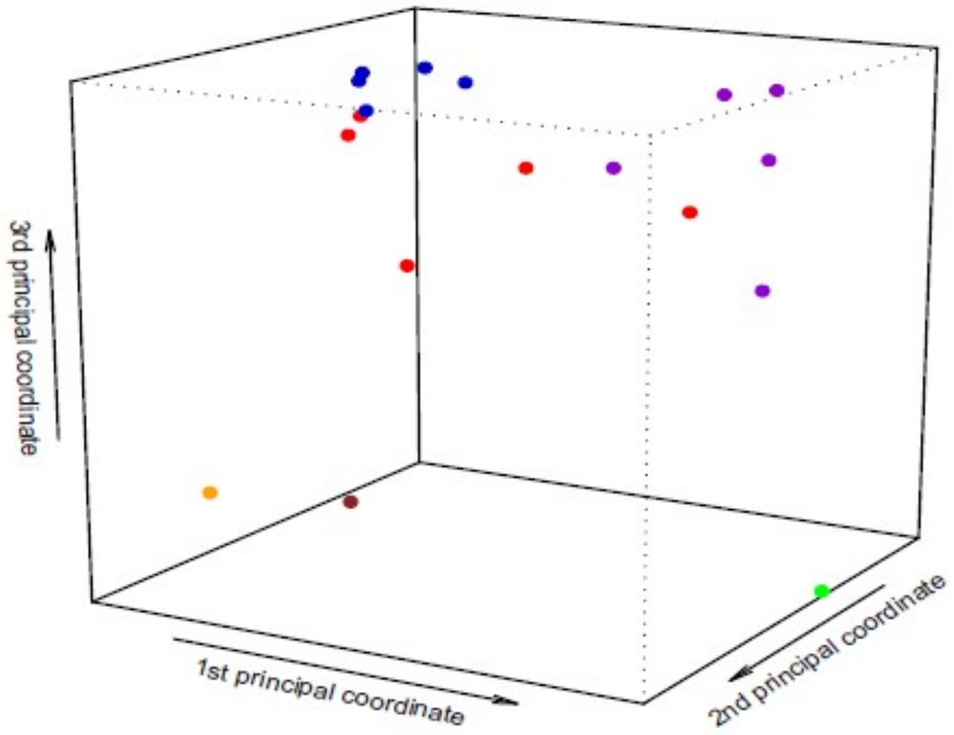


FIGURE 13.2. *Three-dimensional map of 18 world cities using the classical scaling algorithm on airline distances between those cities. The colors reflect the different continents: Asia (purple), North America (red), South America (yellow), Europe (blue), Africa (brown), and Australasia (green).*

CLASSICAL MDS EXAMPLE: AIRLINE DISTANCES

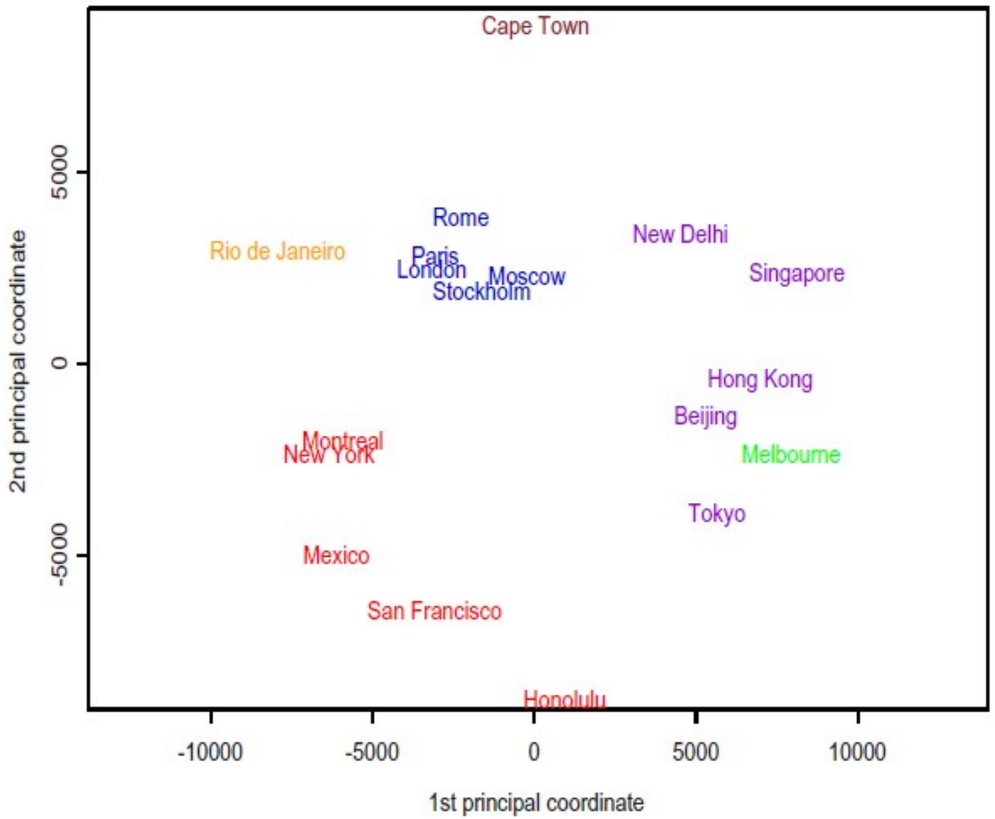


FIGURE 13.1. *Two-dimensional map of 18 world cities using the classical scaling algorithm on airline distances between those cities. The colors*

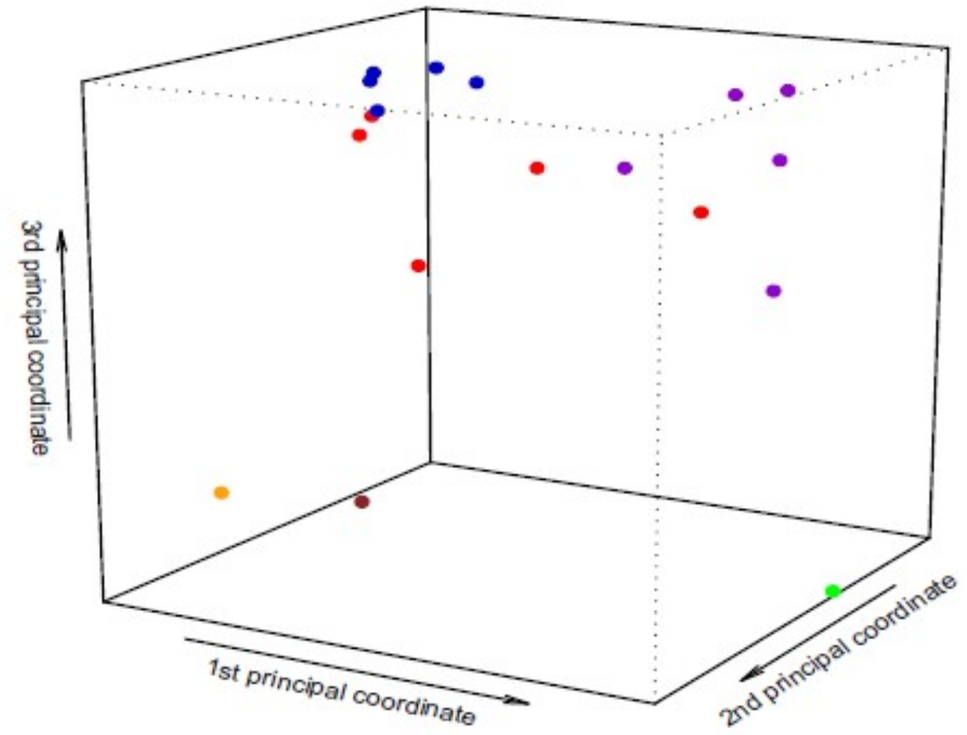


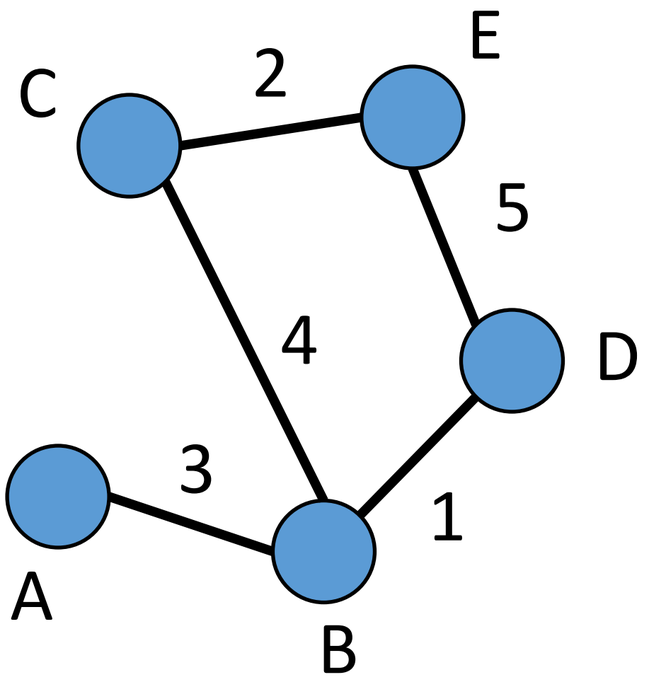
FIGURE 13.2. *Three-dimensional map of 18 world cities using the classical scaling algorithm on airline distances between those cities. The colors reflect the different continents: Asia (purple), North America (red), South America (yellow), Europe (blue), Africa (brown), and Australasia (green).*

How to use this in your project (Extra Credit)

Your input is a weighted graph

We would like to project node positions using MDS

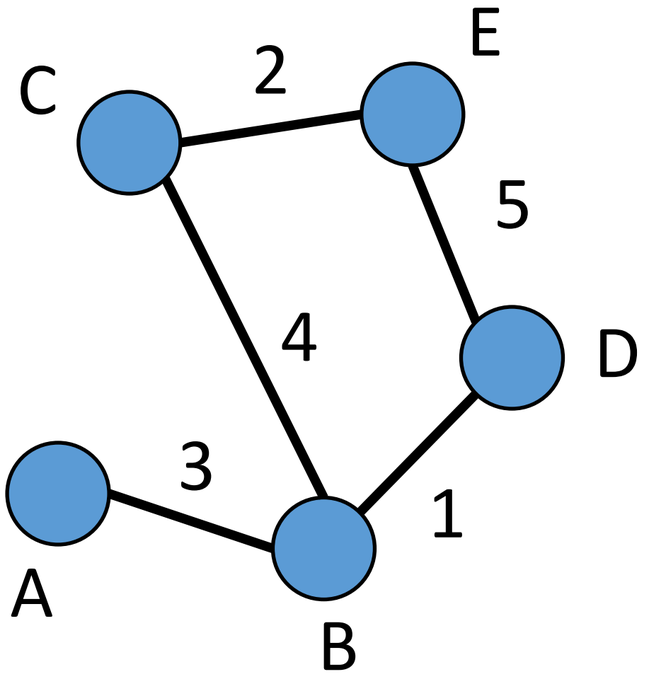
What do we need?



Graph to Dissimilarity Matrix

Convert graph using Dykstra's algorithm to find shortest path between nodes

$$A \rightarrow B = 3$$
$$A \rightarrow C = 7$$
$$A \rightarrow D = 4$$
$$A \rightarrow E = 9$$
$$\dots$$

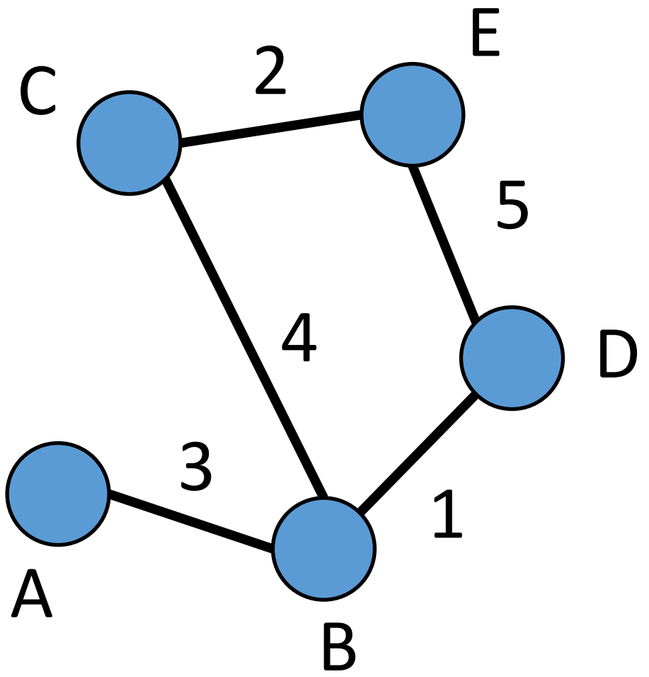


Graph to Dissimilarity Matrix

Dissimilarity matrix passed to MDSJ library's classical MDS function

Returns a list of output positions

Output nodes positions

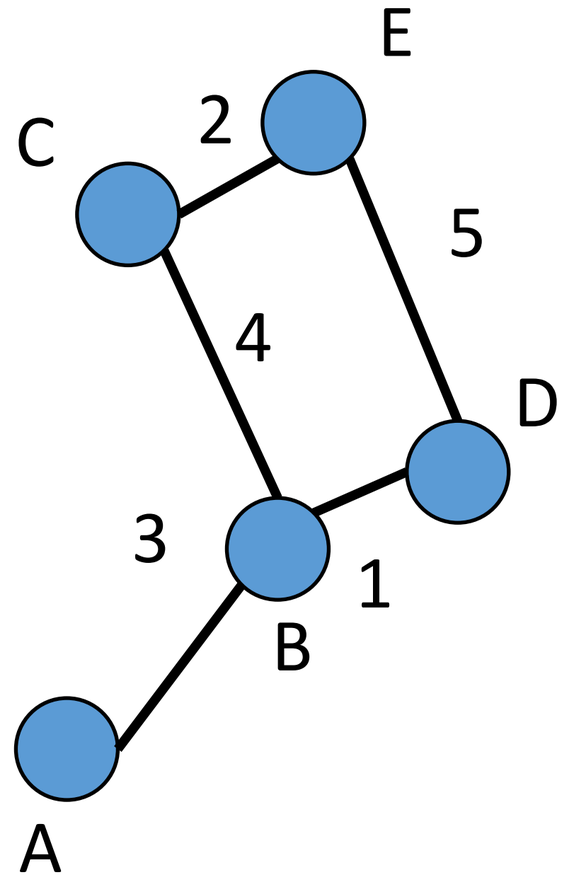


$$\begin{bmatrix} 0 & 3 & 7 & 4 & 9 \\ 3 & 0 & 4 & 1 & 6 \\ 7 & 4 & 0 & 7 & 2 \\ 4 & 1 & 7 & 0 & 5 \\ 9 & 6 & 2 & 5 & 0 \end{bmatrix}$$

Drawing the Graph

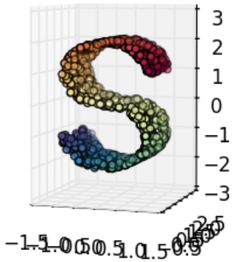
Node positions can be centered and (uniformly) scaled to fit the output display

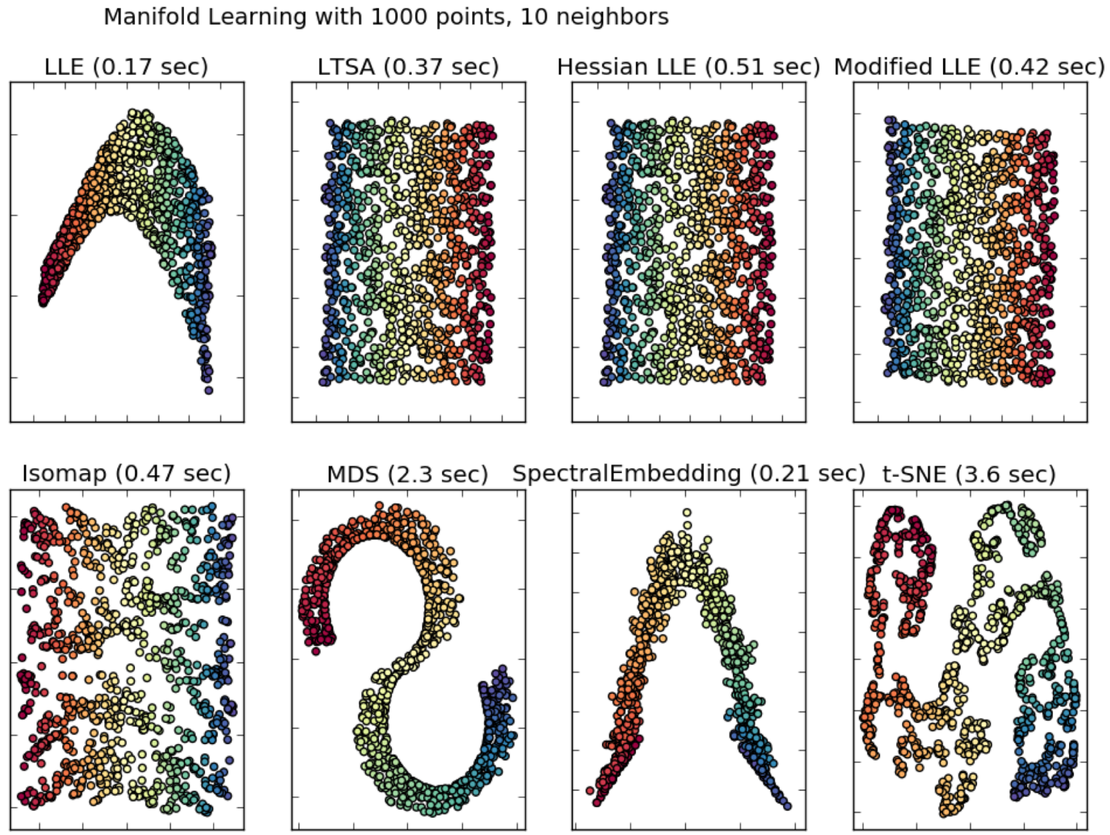
Then, draw a normal node-link diagram



Back to Nonlinear Dimension Reduction

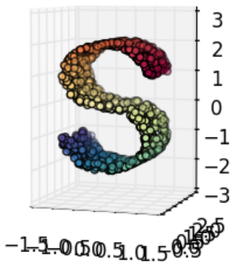
A lot of options, all preserve different features of the high dimensional space

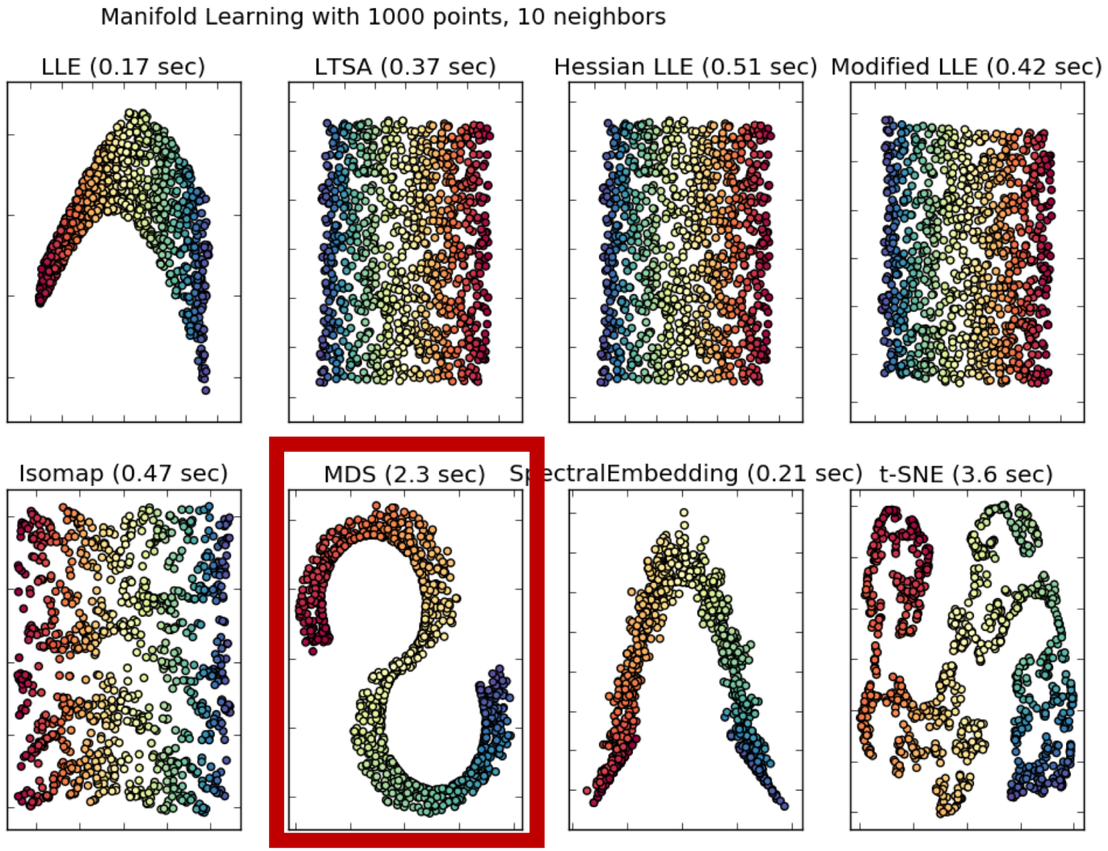




MULTIDIMENSIONAL SCALING (MDS)

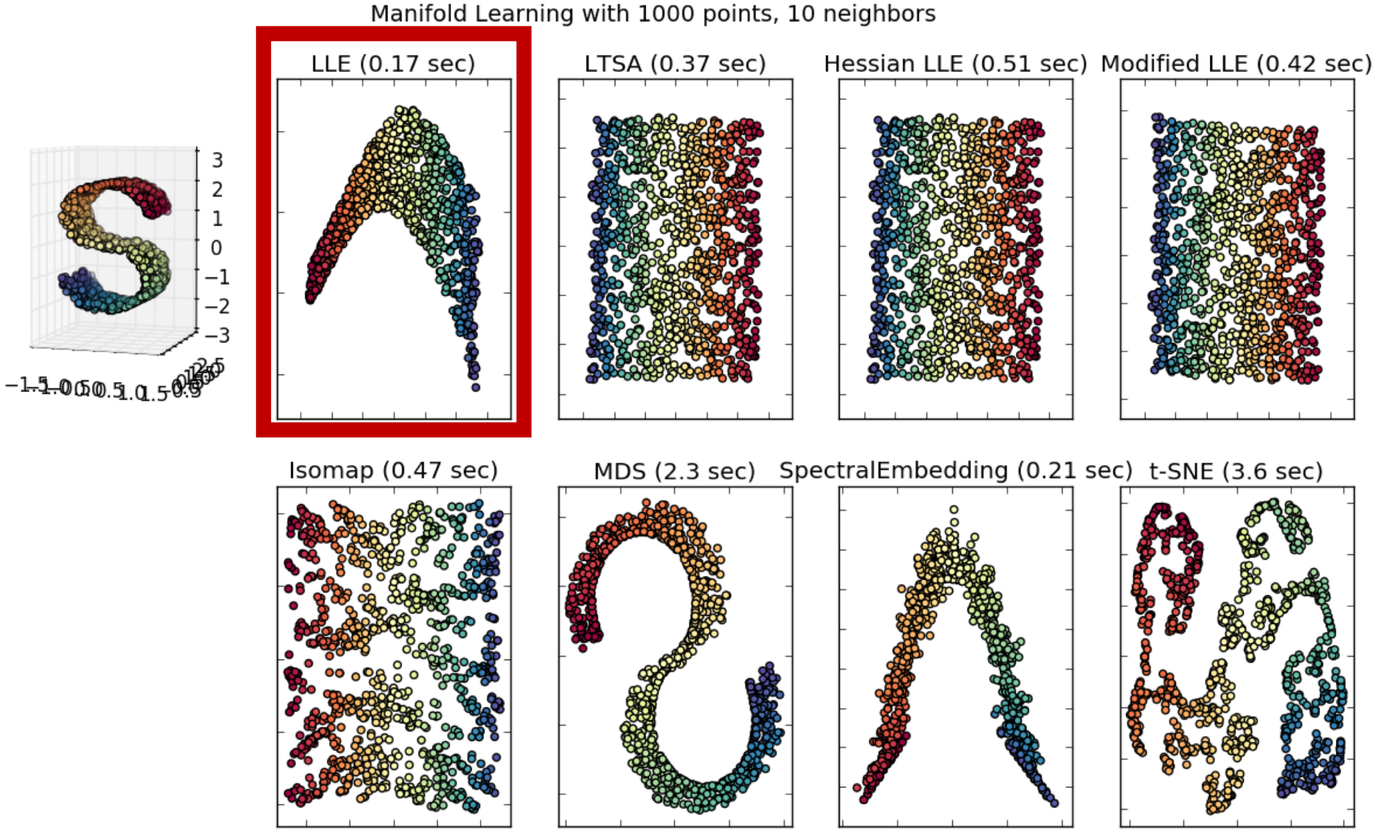
We've already seen





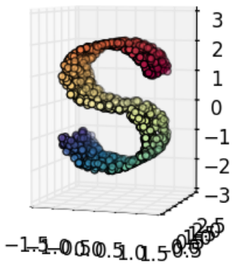
Local Linear Embedding (LLE)

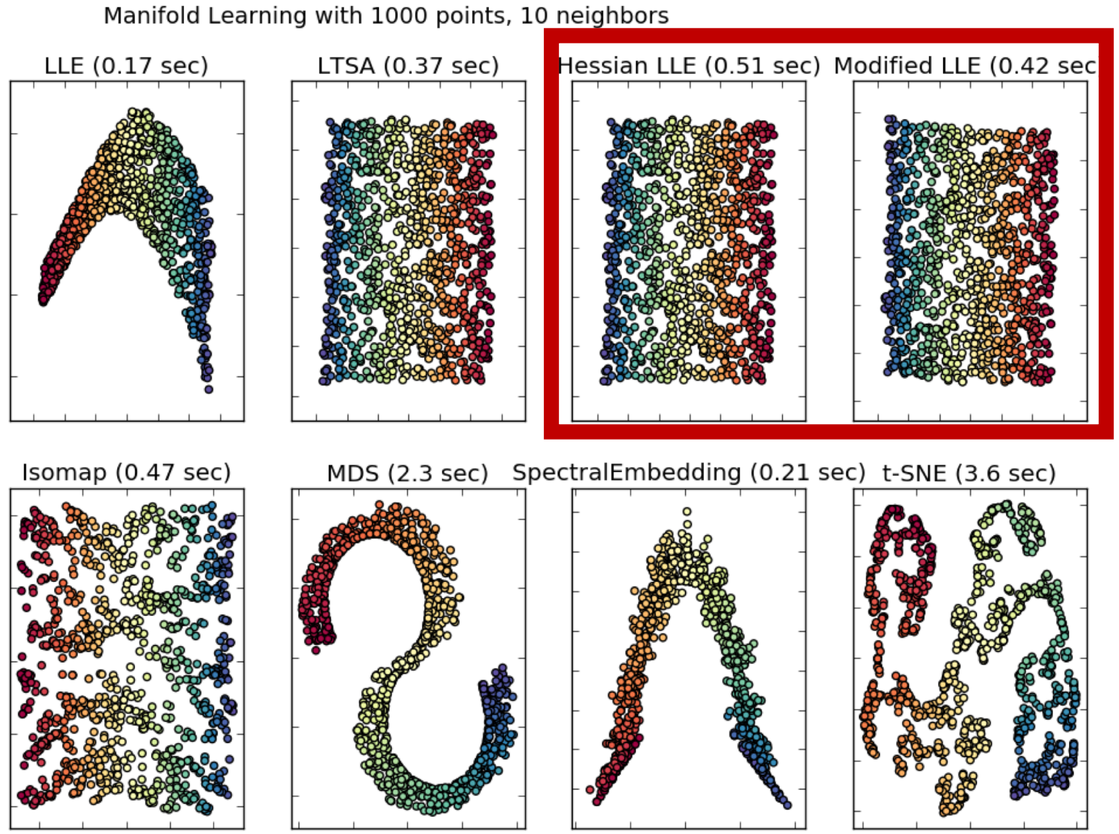
Tries to embed such that local neighborhood distances are preserved as well as possible



LLE Extensions

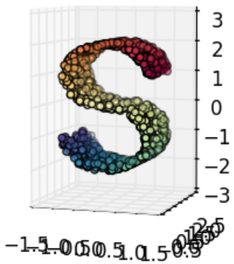
Fix various issues associated with regularization of the domain in LLE

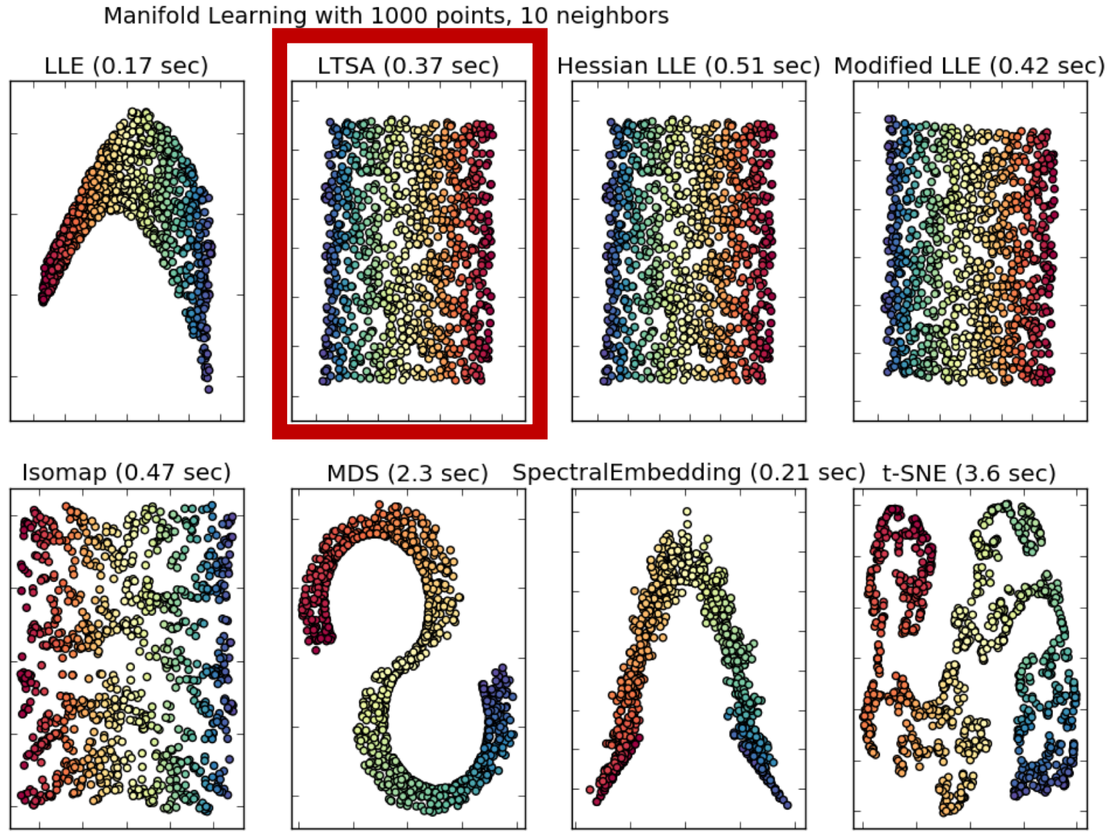




Local Tangent Space Alignment (LTSA)

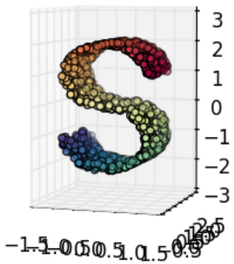
Tries to characterize geometry via local tangent directions

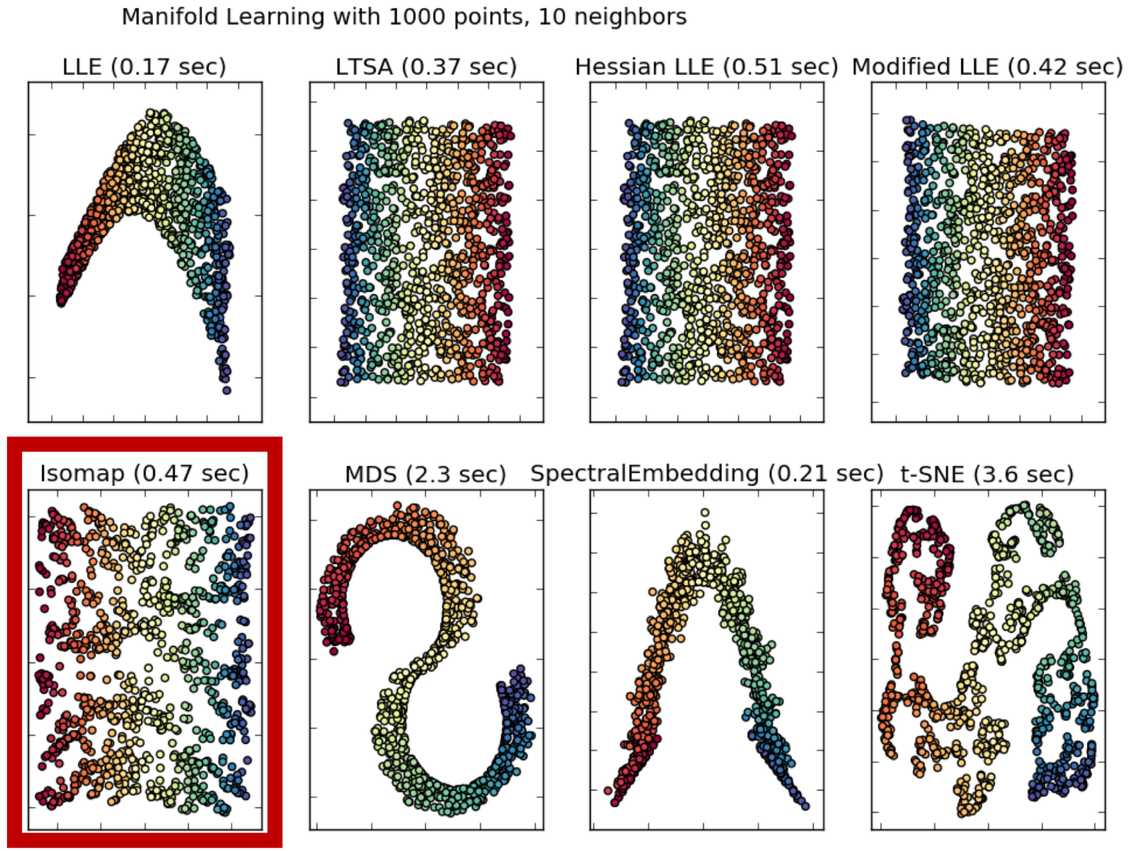




ISOMAP

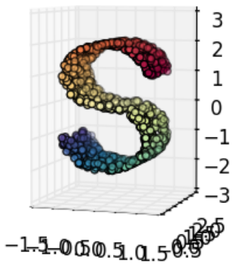
Extension of MDS. Tries to preserve geodesic distance as well as possible

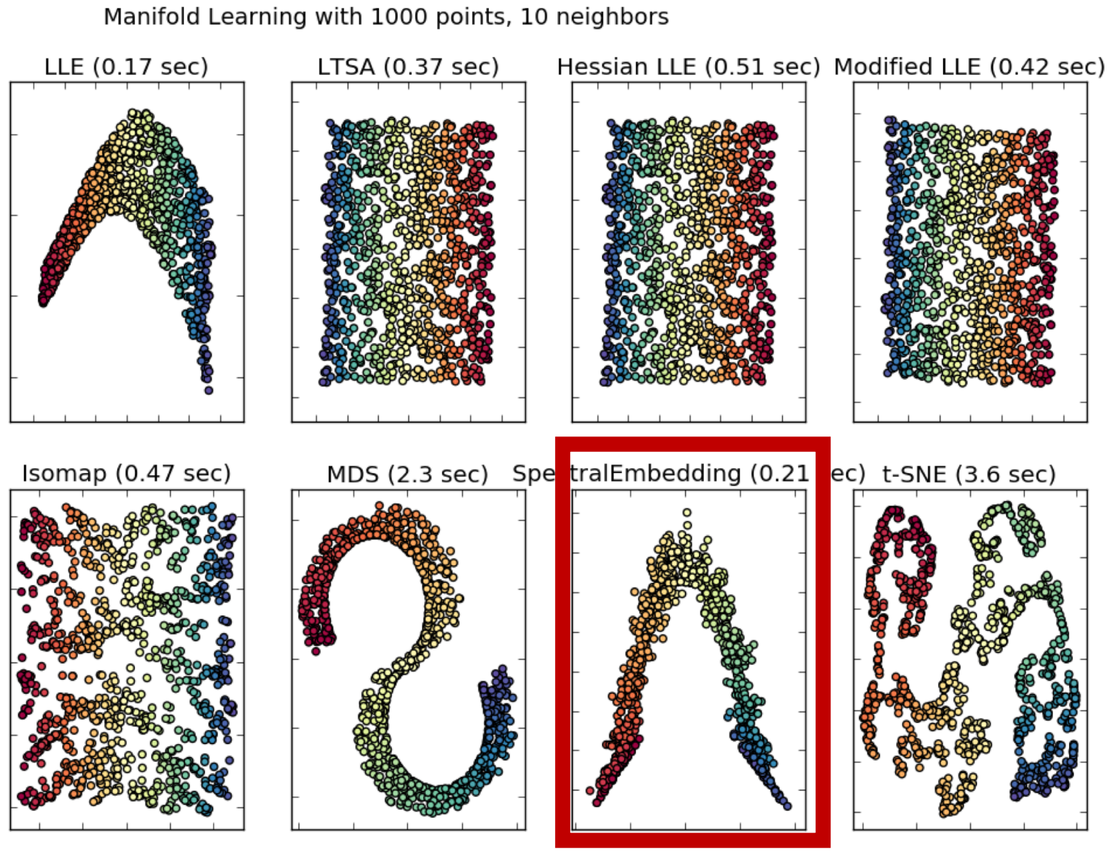




SPECTRAL EMBEDDING

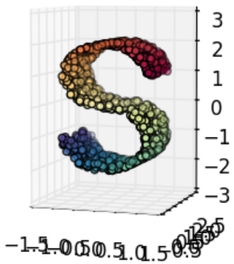
Uses a spectral decomposition of the graph Laplacian

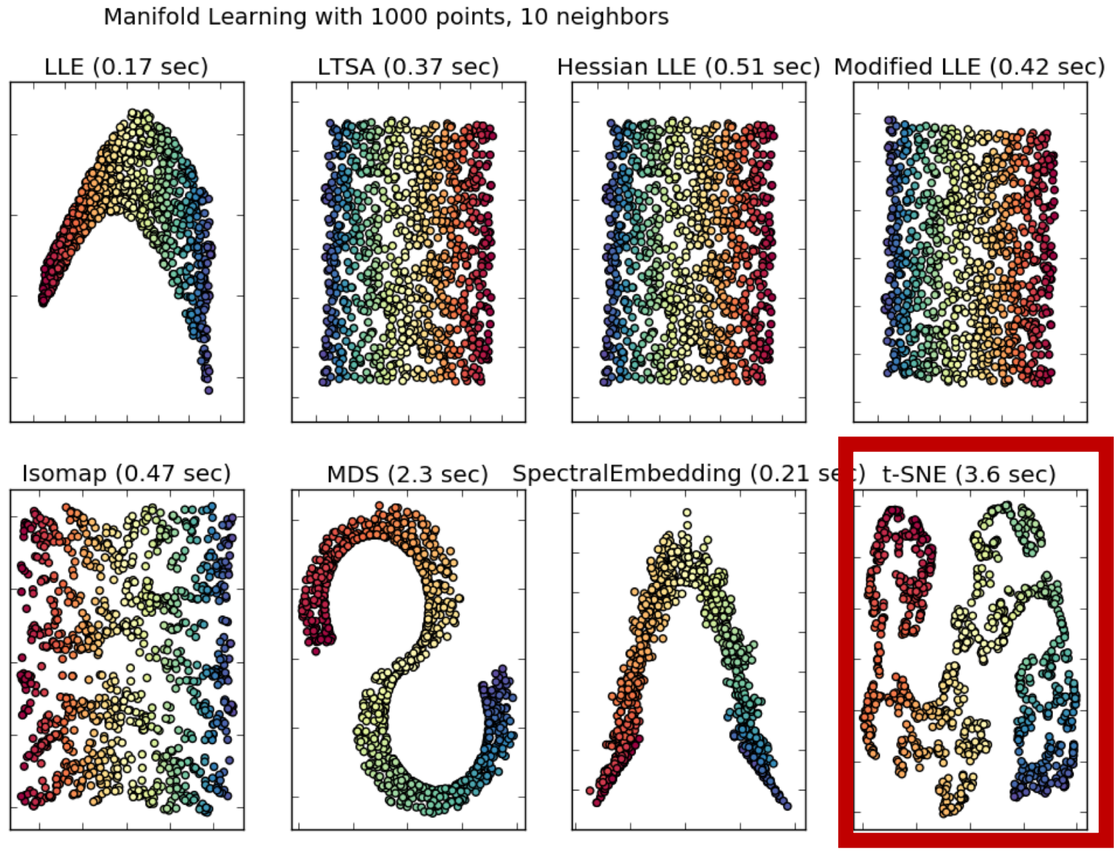




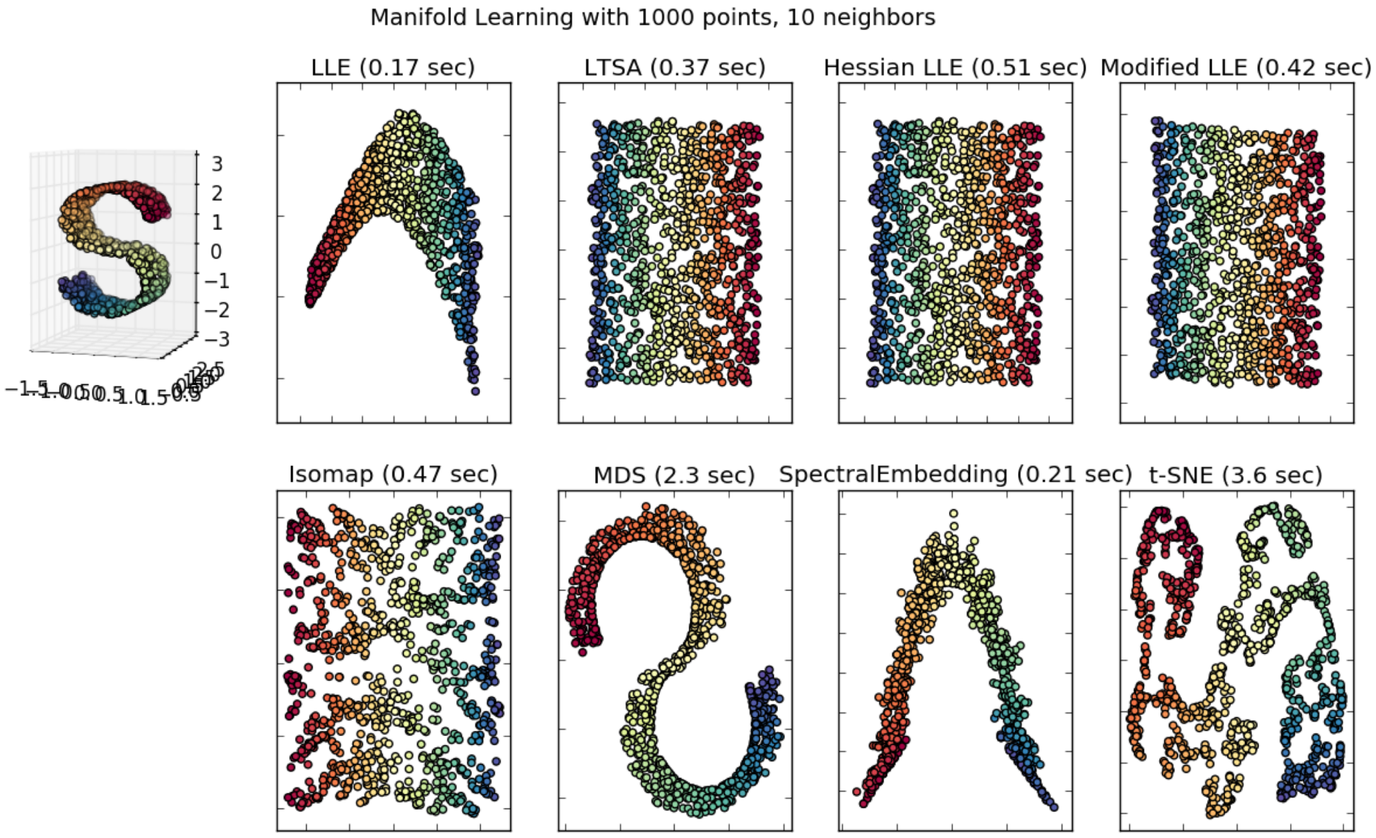
T-DISTRIBUTED STOCHASTIC NEIGHBOR EMBEDDING (T-SNE)

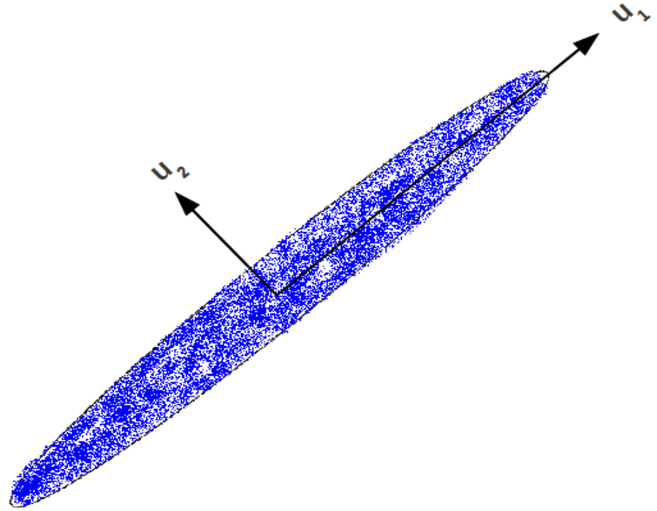
Converts the points probability distribution and samples that distribution





SO…VISUALIZING DATA, WHAT DO YOU USE AND WHEN?

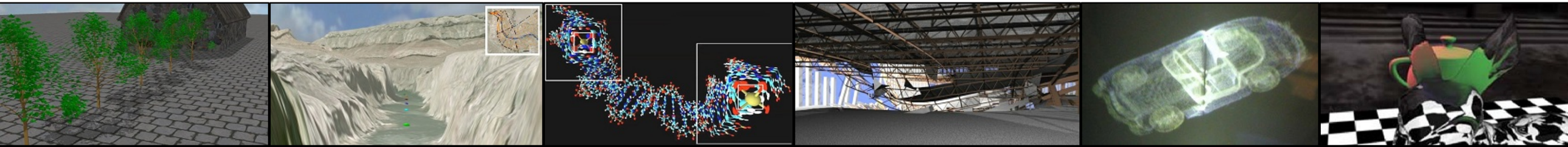




CIS 4930/6930-002
DATA VISUALIZATION

Classifiers and Clustering

Paul Rosen
Assistant Professor
University of South Florida



Reminders

4/16/2018 – Project 7 Peer Review Due

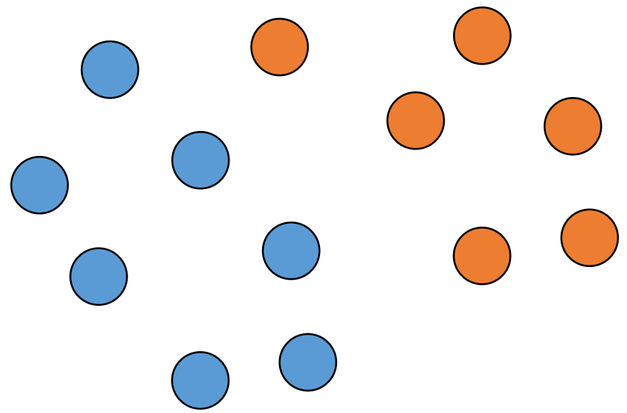
4/23/2018 – Project 8 Due

CLASSIFIERS & CLUSTERING

Goal: to produce a *new categorical data*
on a set of data points

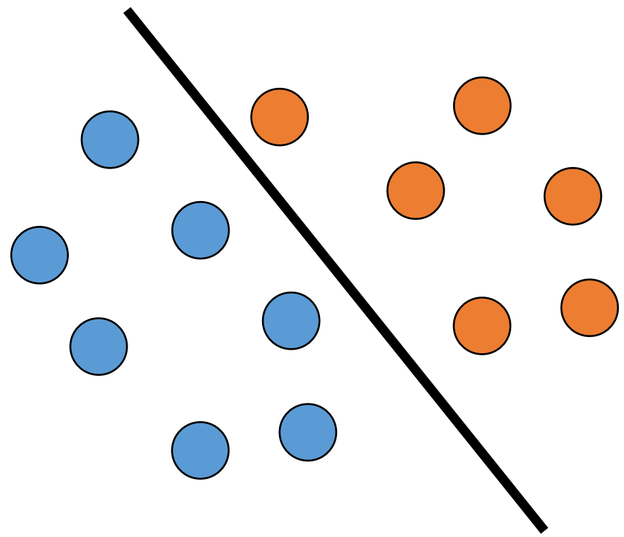
CLASSIFIERS

Given a set of *training* points, divide the domain to label *testing* points



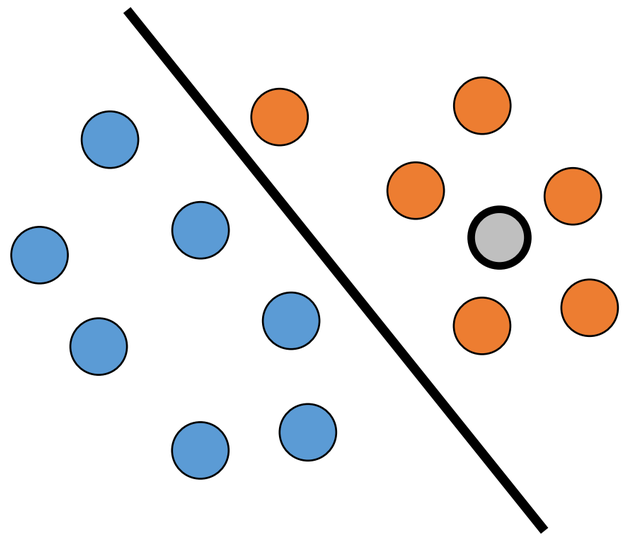
CLASSIFIERS

Given a set of *training* points, divide the domain to label *testing* points



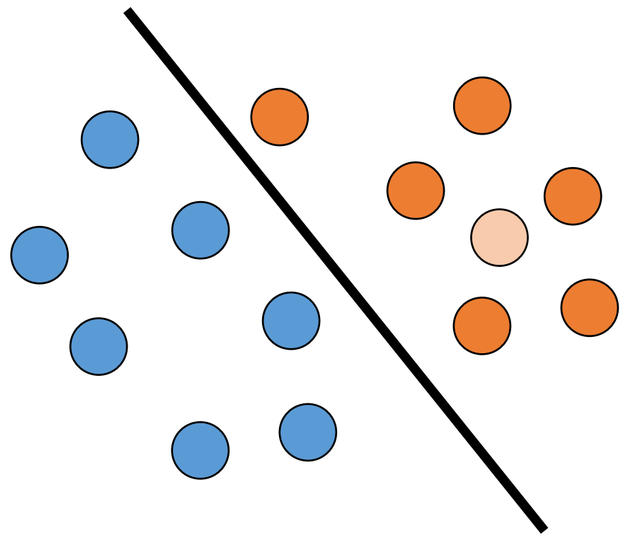
CLASSIFIERS

Given a set of *training* points, divide the domain to label *testing* points



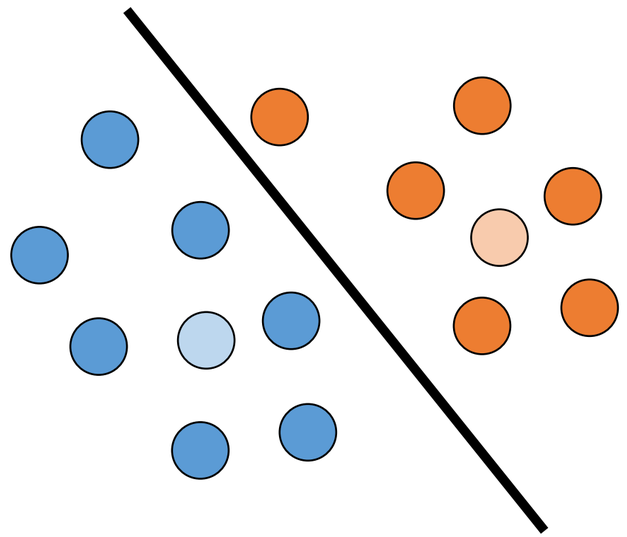
CLASSIFIERS

Given a set of *training* points, divide the domain to label *testing* points



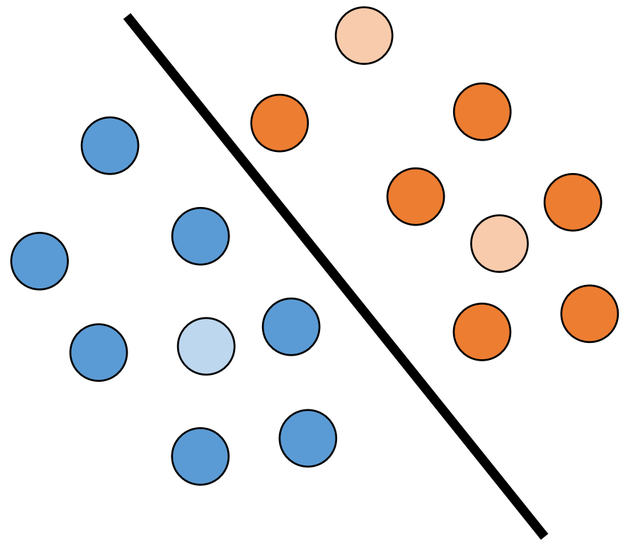
CLASSIFIERS

Given a set of *training* points, divide the domain to label *testing* points



CLASSIFIERS

Given a set of *training* points, divide the domain to label *testing* points



LOTS OF OPTIONS…

Binary classifiers : Kernel Support Vector Machine, Linear Discriminant Analysis, Linear Support Vector Machine

Multiclass classifier : Gaussian Naive Bayes, K Nearest Neighbors, Large Margin Nearest Neighbors, Linear Discriminant Analysis, Multi-class Error-Correcting Output Codes, Multi-class Linear Machine, Multi-class Logistic Regression, Quadratic Discriminant Analysis, Random Forest, Relaxed Tree, ShareBoost, Multi-class Support Vector Machine, Feedforward Neural Network for Classification, Gaussian Process Classifier

LIST FROM HTTP://WWW.SHOGUN-TOOLBOX.ORG/

LOTS OF OPTIONS…

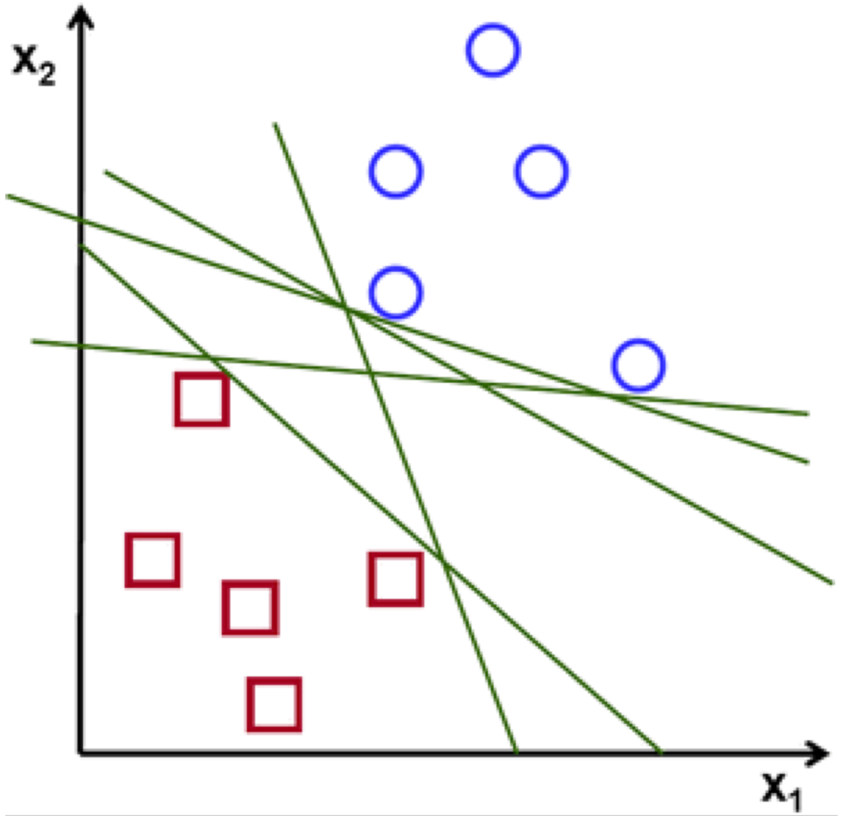
Binary classifiers : **Kernel Support Vector Machine**, Linear Discriminant Analysis, **Linear Support Vector Machine**

Multiclass classifier : **Gaussian Naive Bayes**, **K Nearest Neighbors**, Large Margin Nearest Neighbors, Linear Discriminant Analysis, Multi-class Error-Correcting Output Codes, Multi-class Linear Machine, Multi-class Logistic Regression, Quadratic Discriminant Analysis, **Random Forest**, Relaxed Tree, ShareBoost, Multi-class Support Vector Machine, Feedforward Neural Network for Classification , Gaussian Process Classifier

LIST FROM HTTP://WWW.SHOGUN-TOOLBOX.ORG/

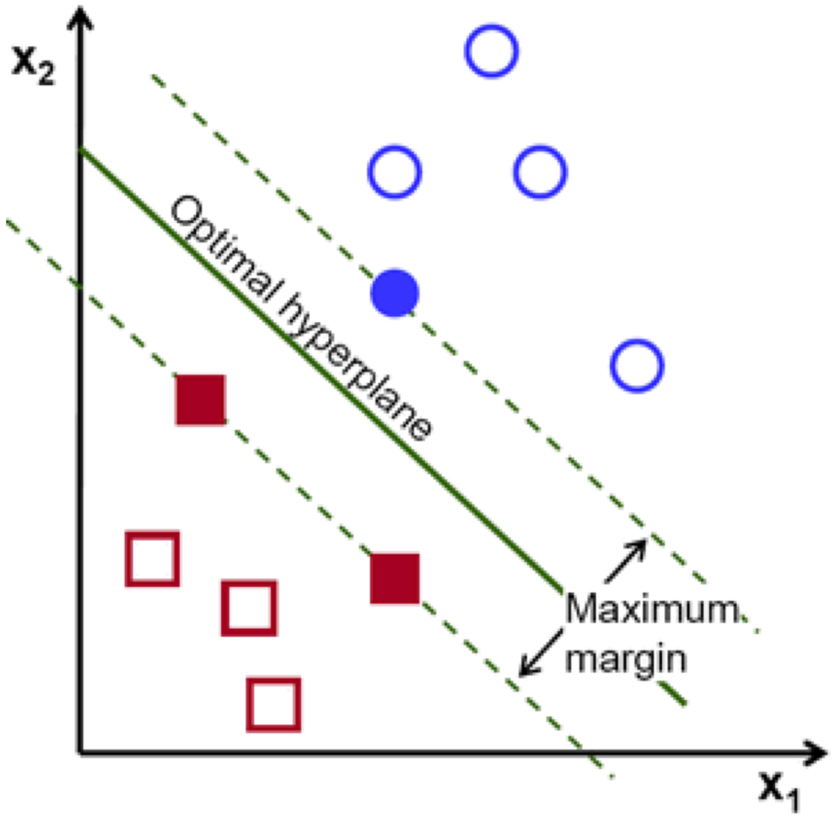
Linear Support Vector Machine (SVM)

Given a set of points with labels many hyperplanes could be defined to divide the sets



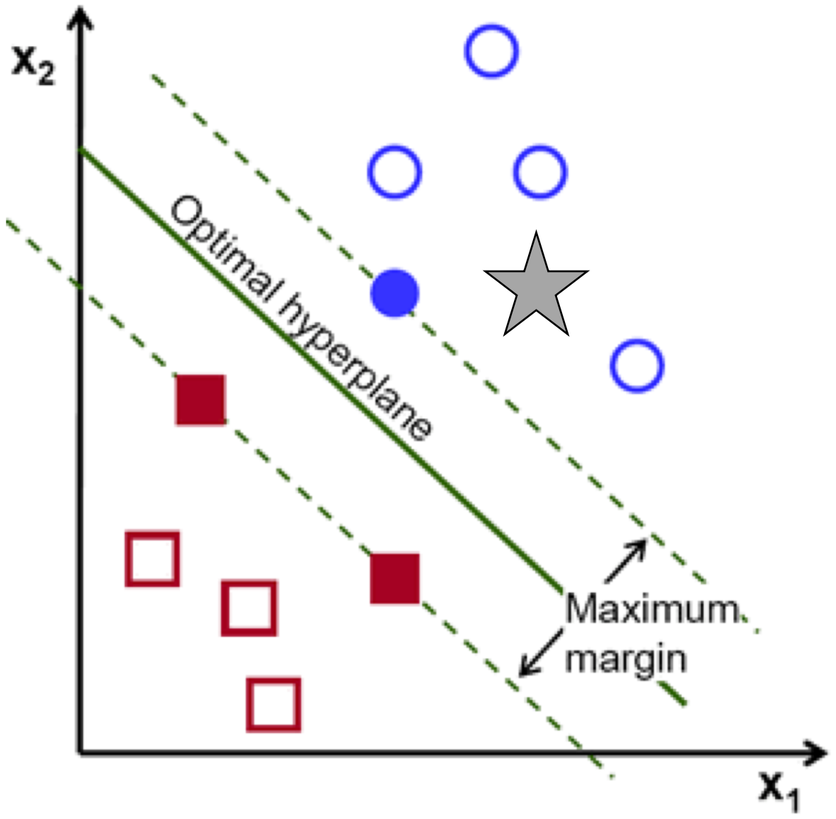
Linear Support Vector Machine (SVM)

Select the optimal hyperplane that maximizes the margin between classes



Linear Support Vector Machine (SVM)

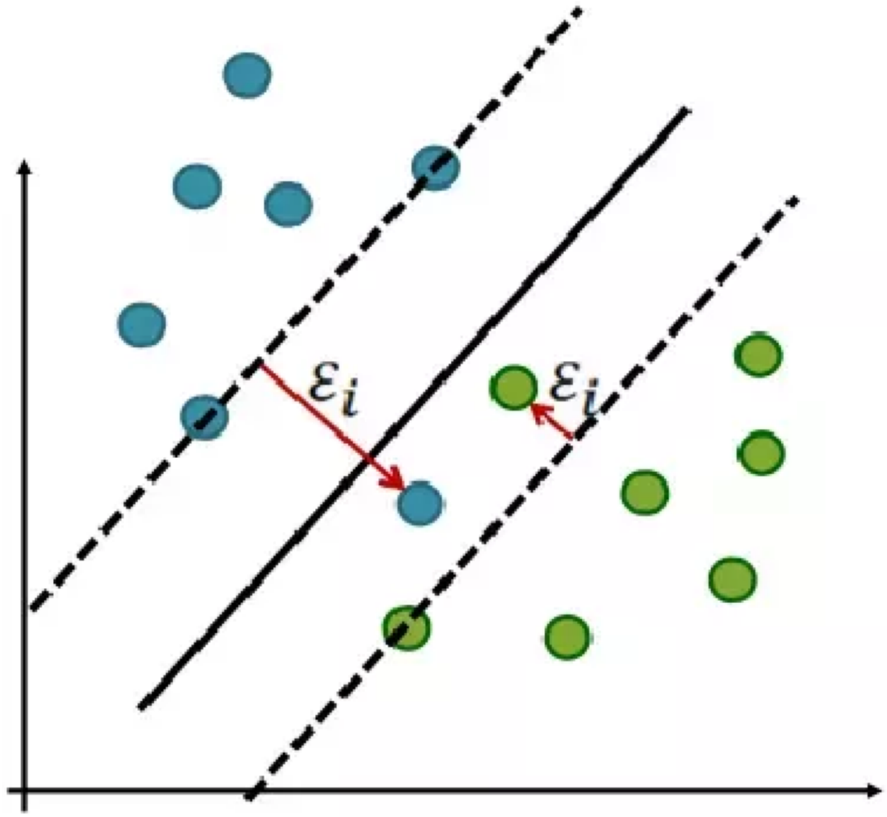
Now testing points can be classified



Linear Support Vector Machine (SVM)

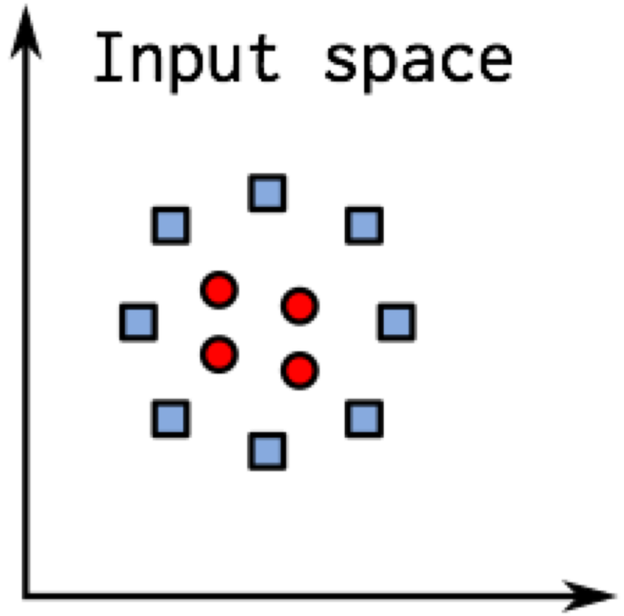
Data is often not linearly separable

Soft margin variation enables finding a hyperplane, but some points will be misclassified in this case



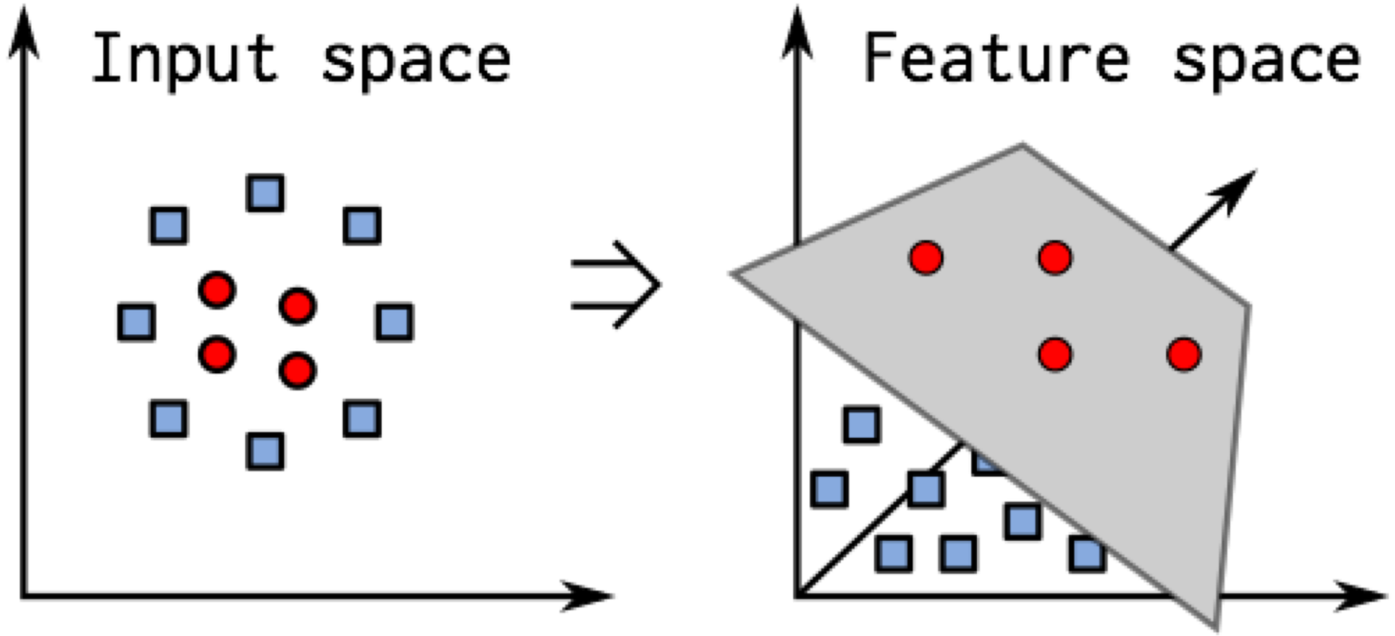
KERNEL SVM

Often data cannot be easily separated linearly



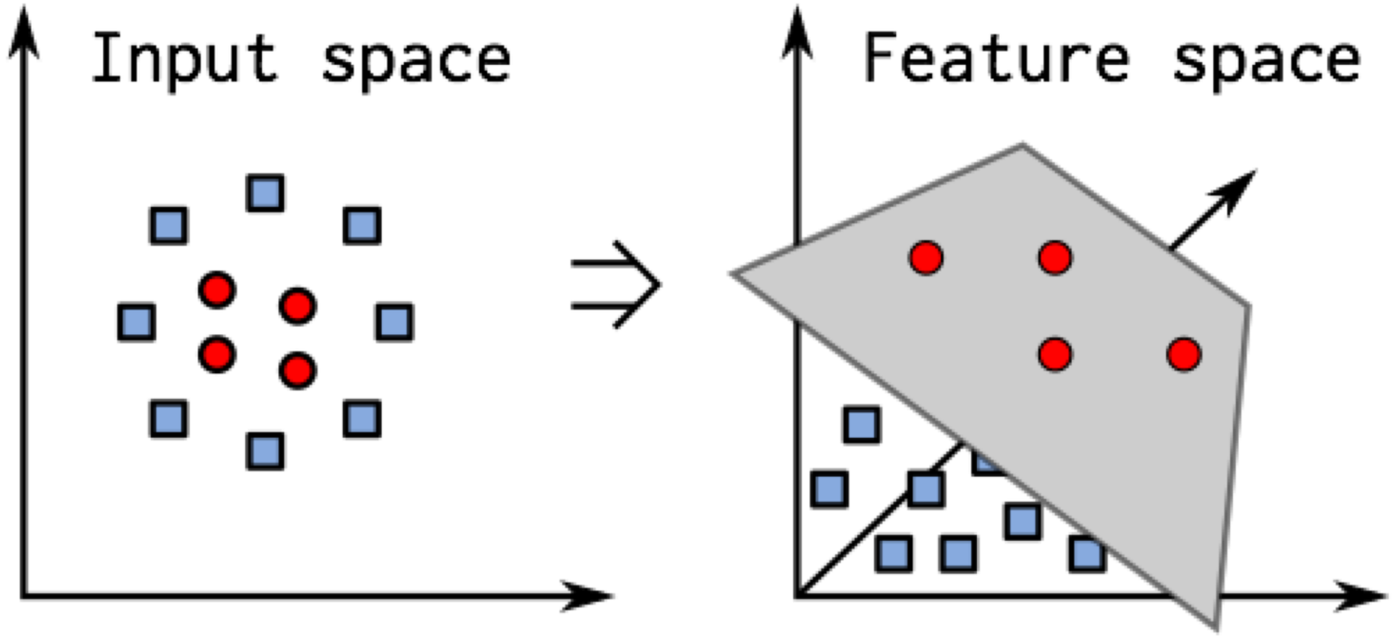
KERNEL SVM

Transform the data from its original domain into a feature space using a kernel (such as a Gaussian)



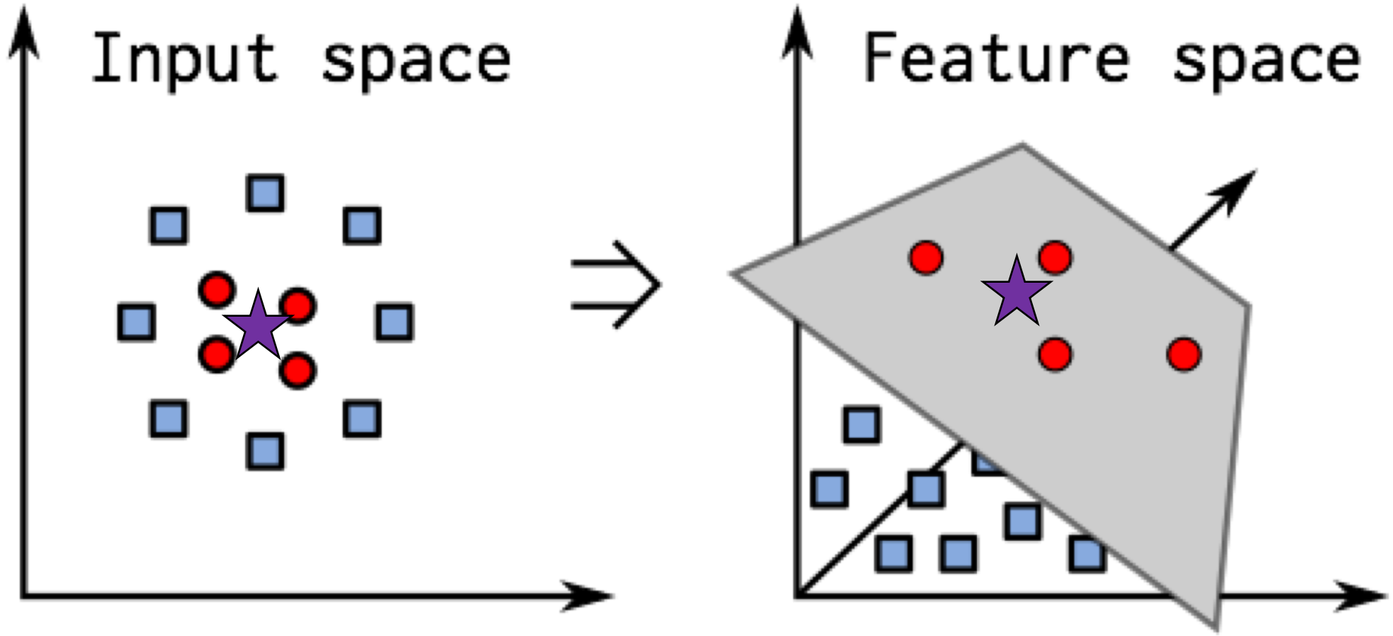
KERNEL SVM

Run SVM on the feature space



KERNEL SVM

Test a point by converting it to feature space

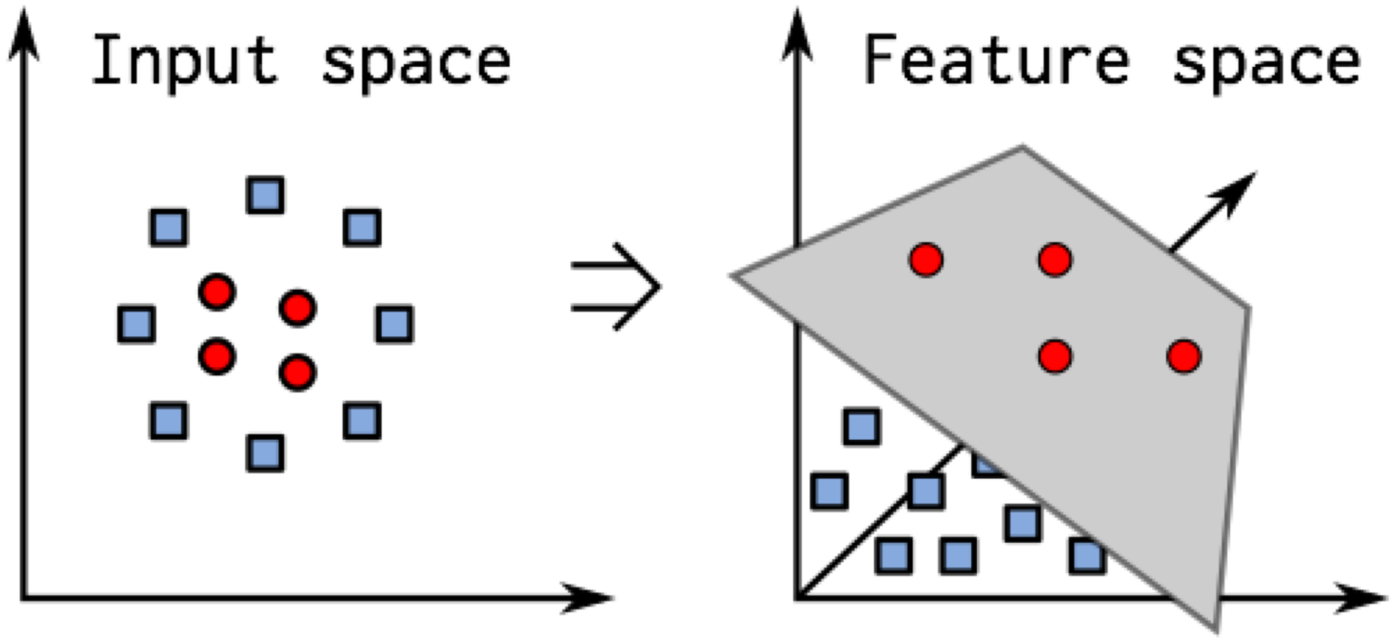


KERNEL SVM

Wide variety of kernels available

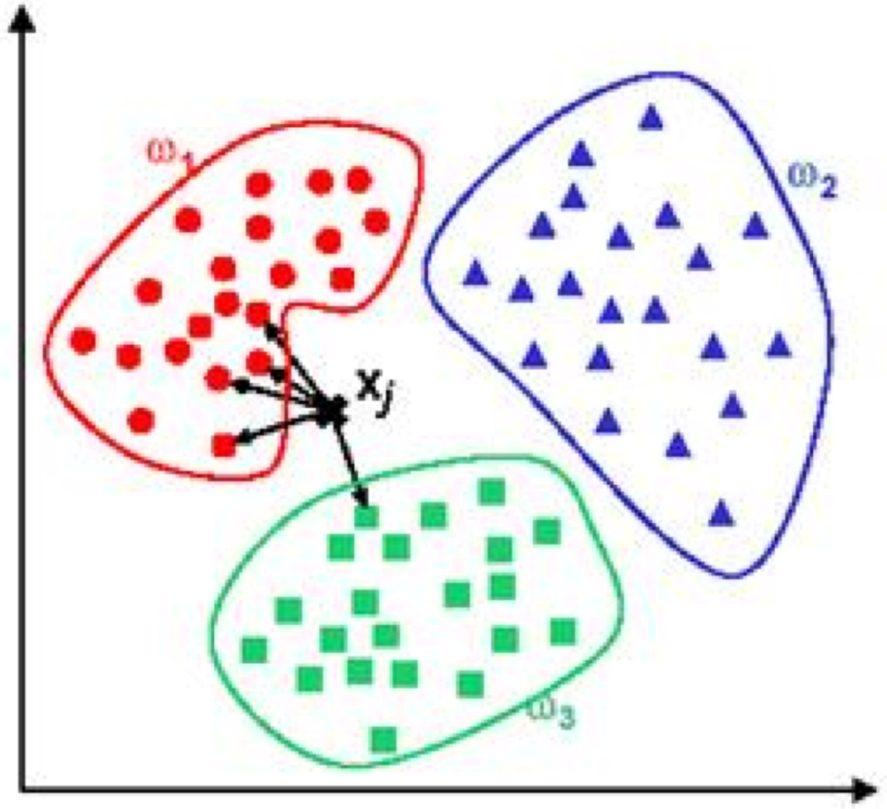
Gaussian, Fisher, Graph, RBF, Polynomial,

Which to use on your data?



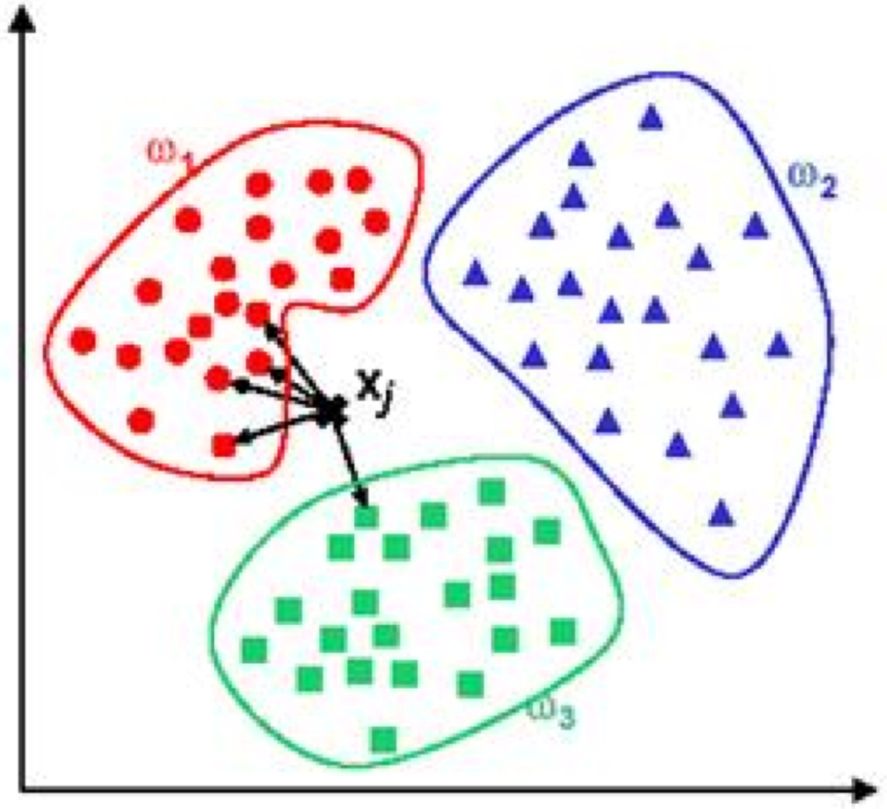
K NEAREST NEIGHBORS CLASSIFIER

Multiclassifier based upon proximity



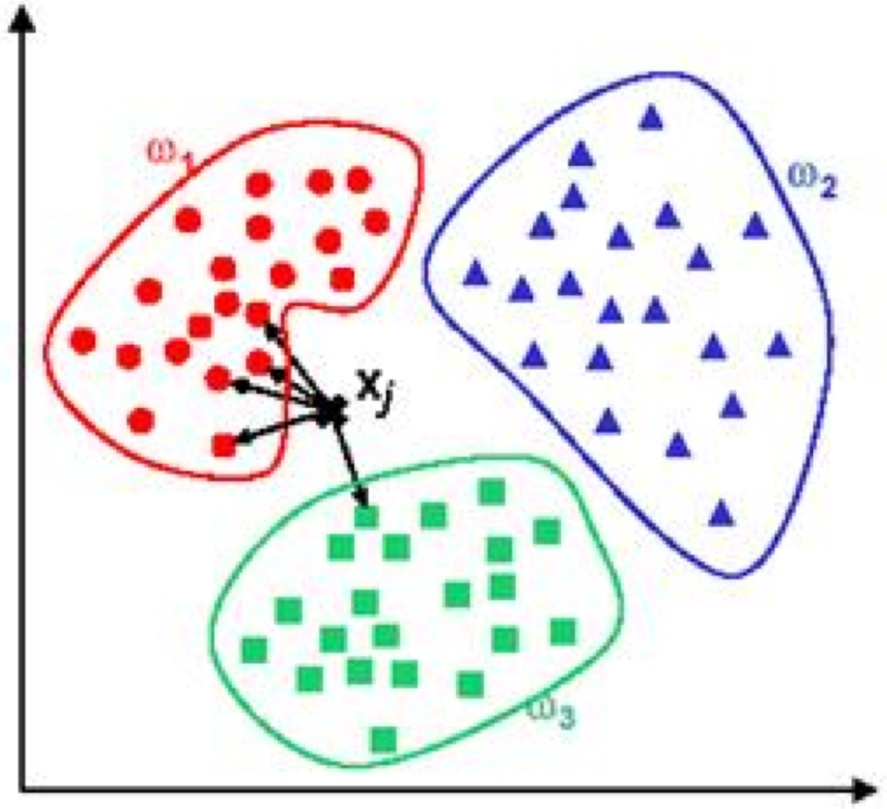
K NEAREST NEIGHBORS CLASSIFIER

Finding the closest k training points



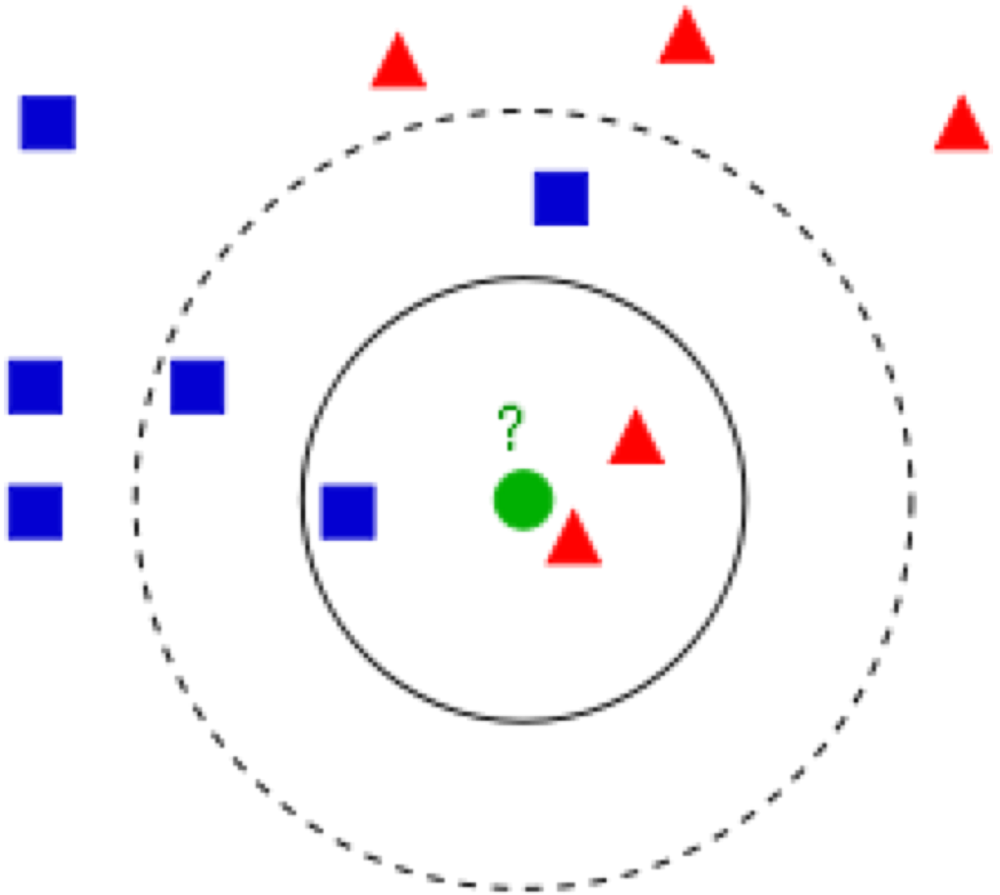
K NEAREST NEIGHBORS CLASSIFIER

those points vote on the class of the testing point



K NEAREST NEIGHBORS CLASSIFIER

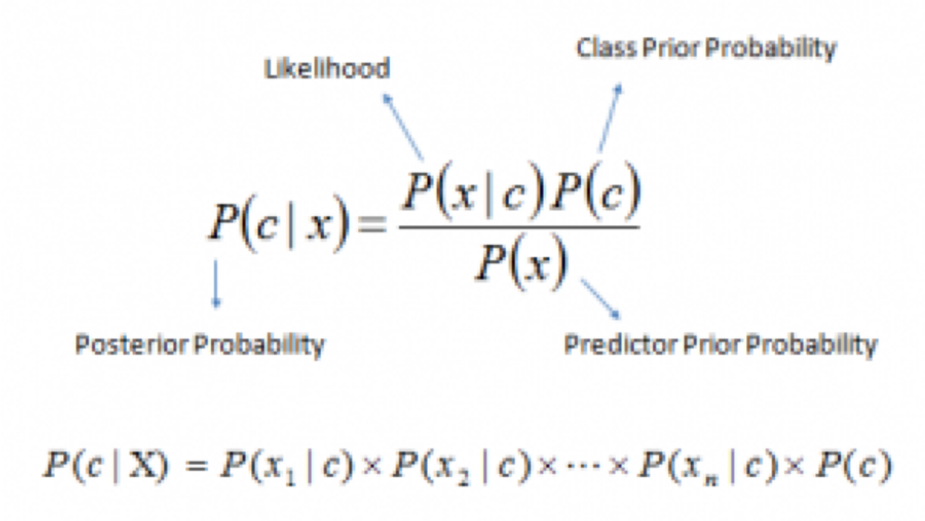
Problem: selecting k is hard



(Gaussian) Naive Bayes

Naïve Bayes uses conditional probability to classify a test point

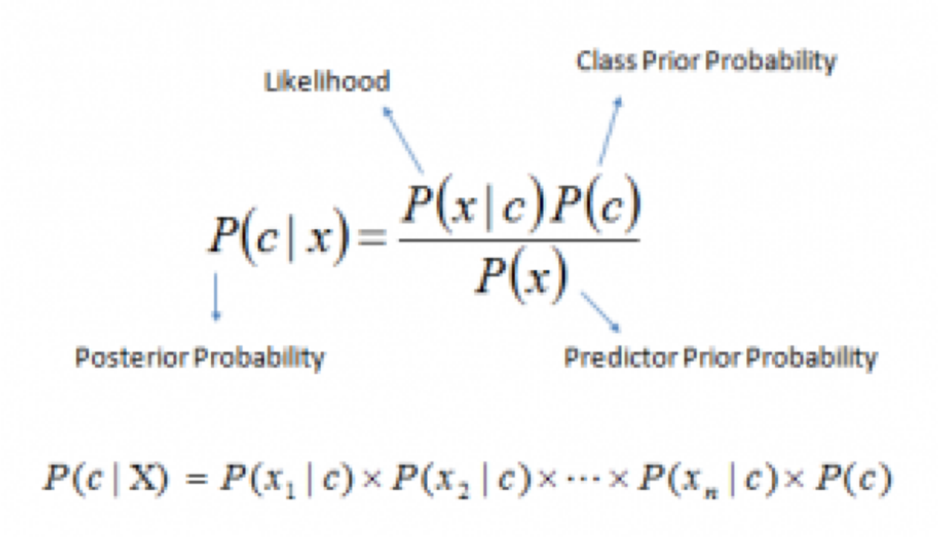
Output is not a decision, but a *probability for a decision*



To remind yourself about conditional probability, see: http://students.brown.edu/seeing-theory/compound-probability/index.html

(Gaussian) Naive Bayes

For categorical data Naïve Bayes can be used directly by counting how frequently events occur together



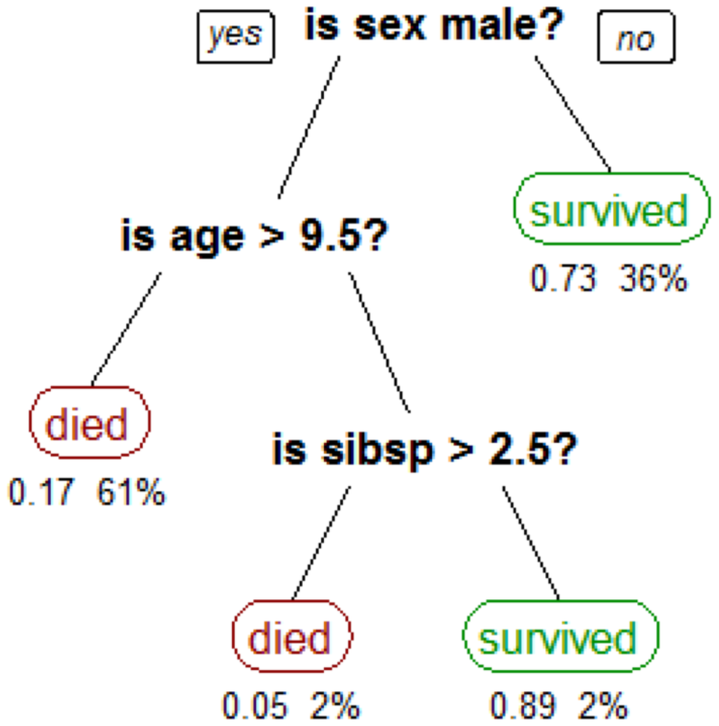
Gaussian Naive Bayes

For numeric data Gaussian Naïve Bayes determines the probability by assuming a Gaussian distribution for the data

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

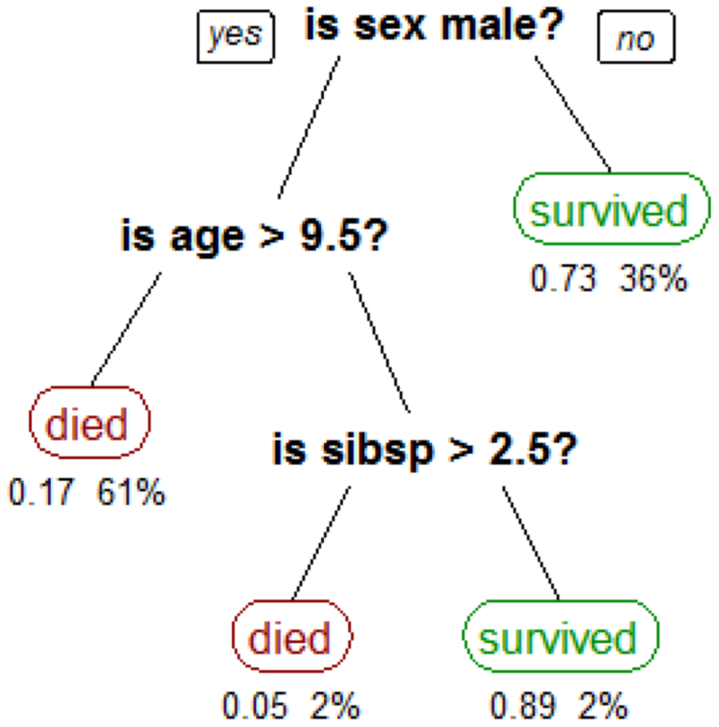
Decision Tree and Random Forest

Each level of a decision tree makes an observation about the data



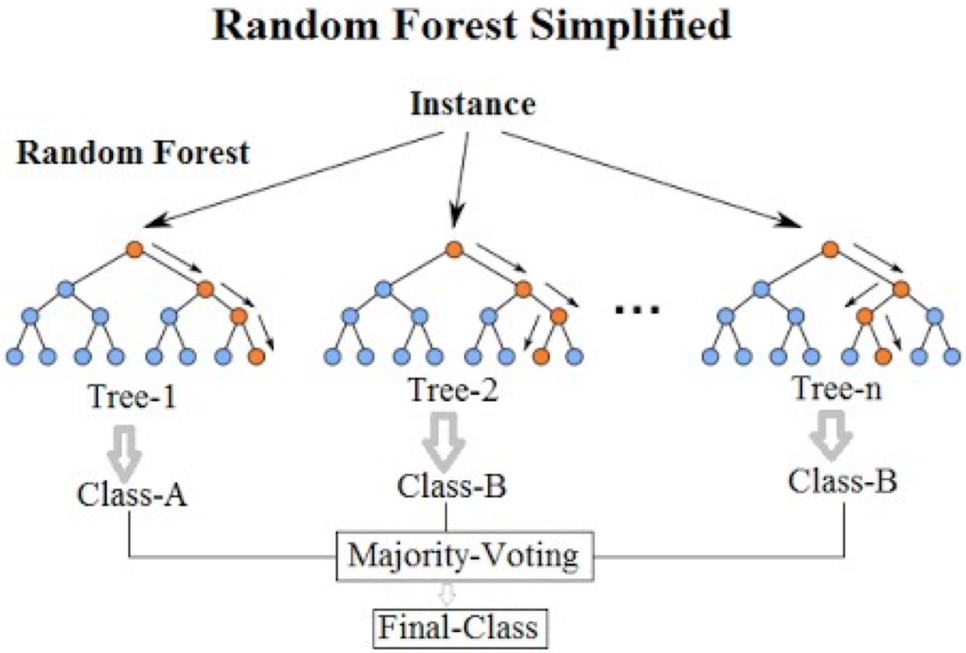
Decision Tree and Random Forest

Following the tree to a leaf provides a class and probability for that class

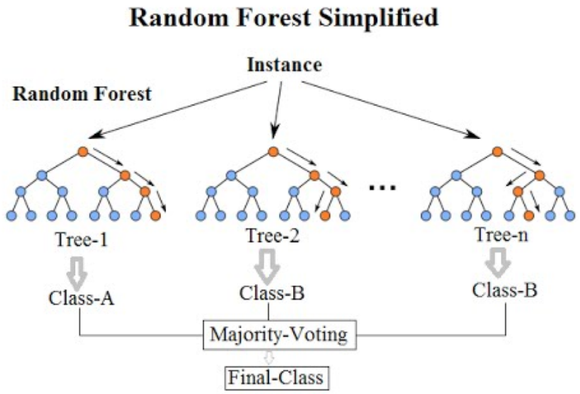


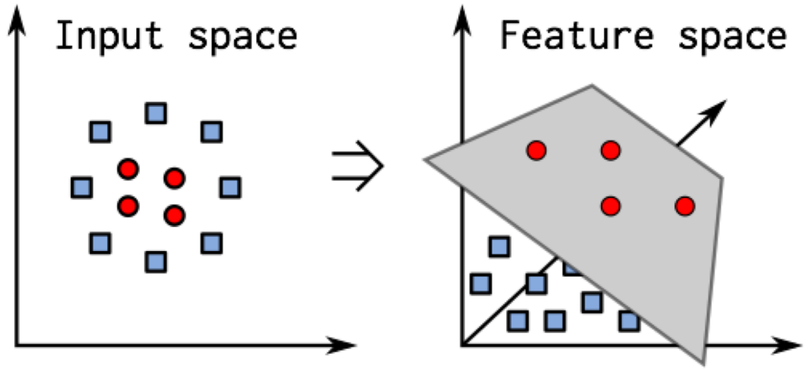
Decision Tree and Random Forest

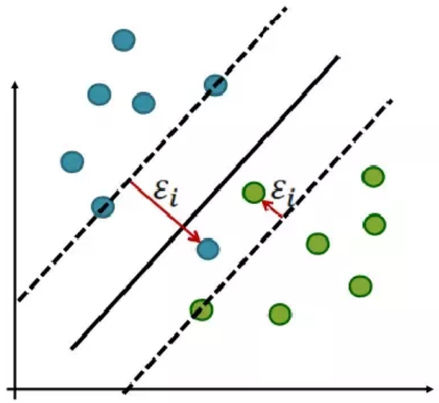
Random forest uses multiple decision trees to vote on a classification testing data

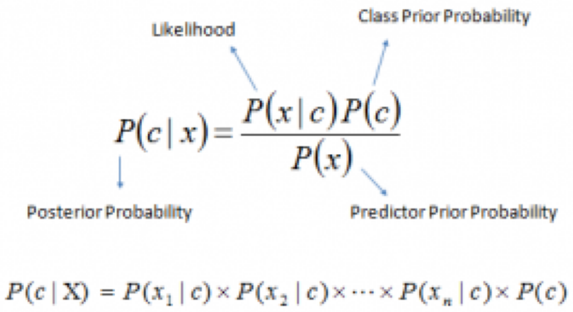


So… When/How can we use these in our visualizations?

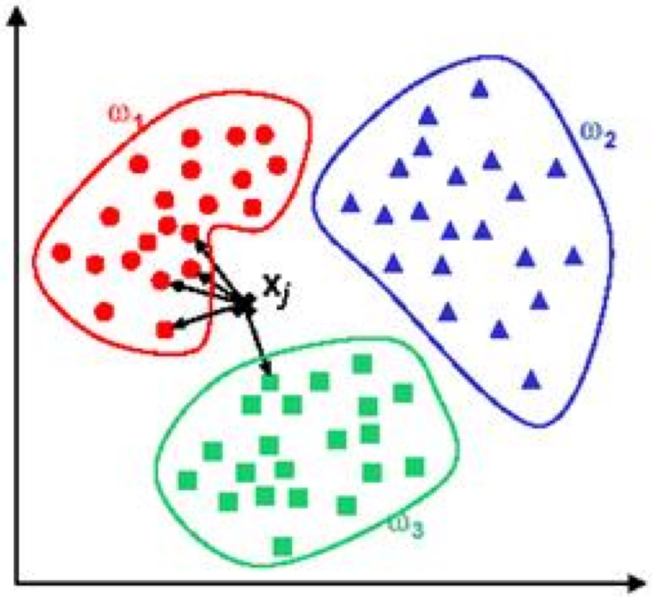








$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$



Regression

Many of the same tools used for classification can be used for regression analysis (finding the relationship between variables).

Examples: Kernel Ridge Regression, Linear Ridge Regression, Multiple Kernel Learning, Random Forest, Support Vector Regression, Feedforward Neural Networks for Regression, Gaussian Process Regression

Finding Clusters in Data

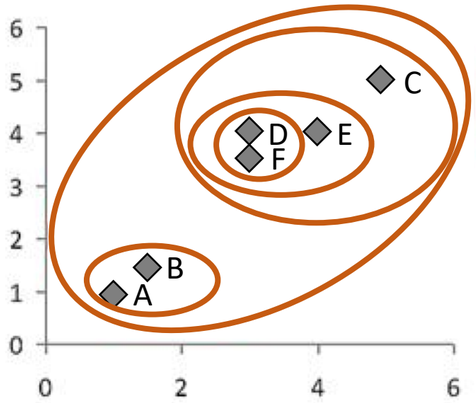
Gaussian Mixture Models, Hierarchical Clustering, K-means Clustering

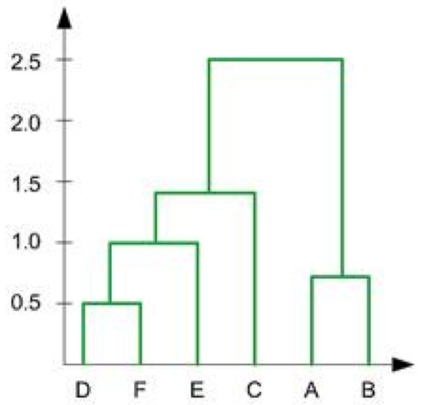
Finding Clusters in Data

Gaussian Mixture Models, **Hierarchical Clustering**, **K-means Clustering**

Hierarchical Clustering

Finds a series of clustering events and represents them in a dendogram

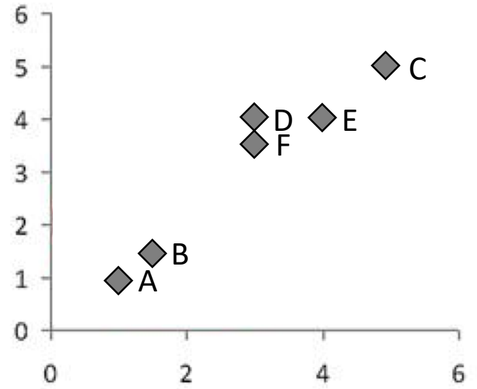


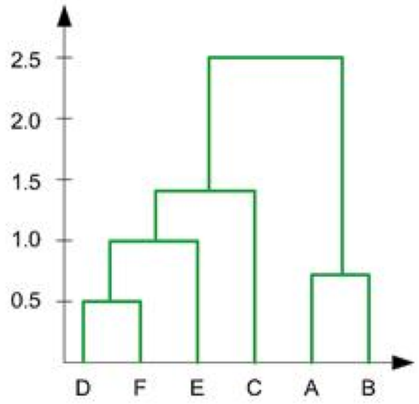


HIERARCHICAL CLUSTERING

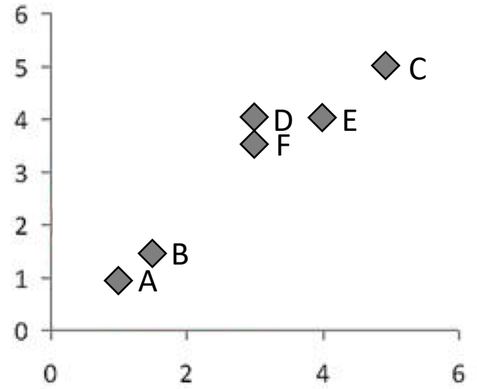
Given a pairwise distance matrix

Sort the pairwise distances from smallest

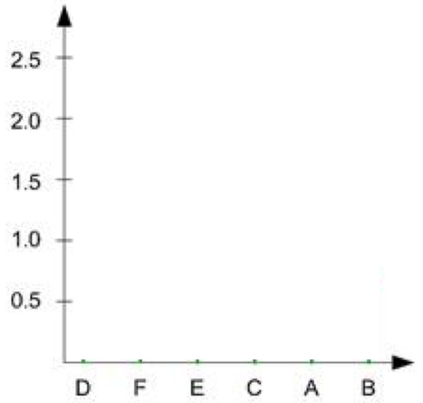




HIERARCHICAL CLUSTERING

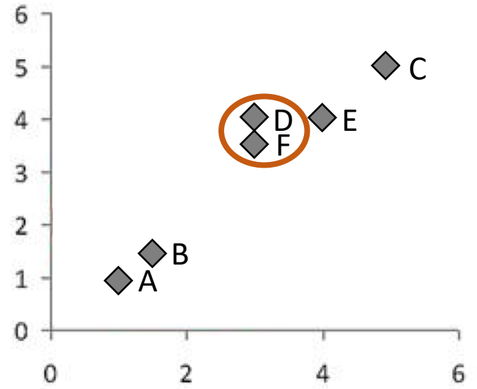


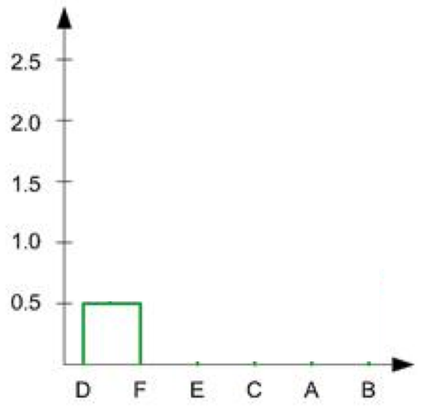
Start with each point as its own cluster



HIERARCHICAL CLUSTERING

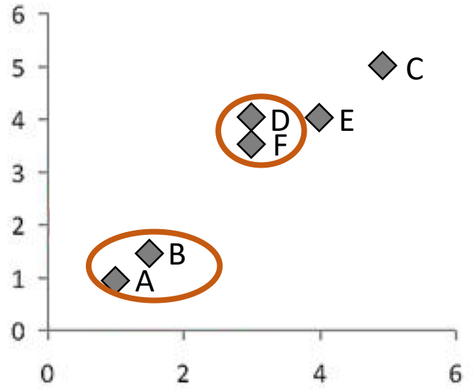
Step through sorted distances joining clusters

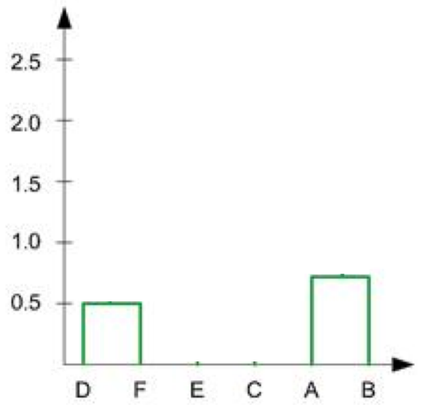




HIERARCHICAL CLUSTERING

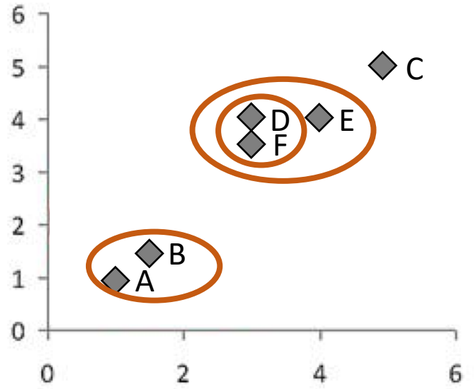
Step through sorted distances joining clusters

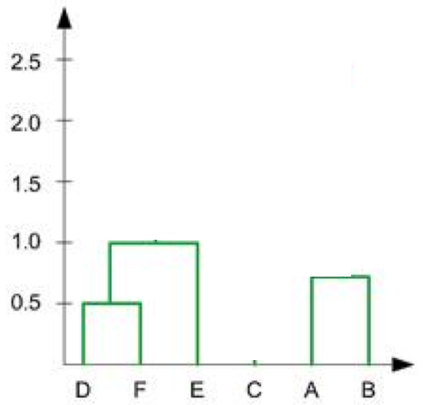




HIERARCHICAL CLUSTERING

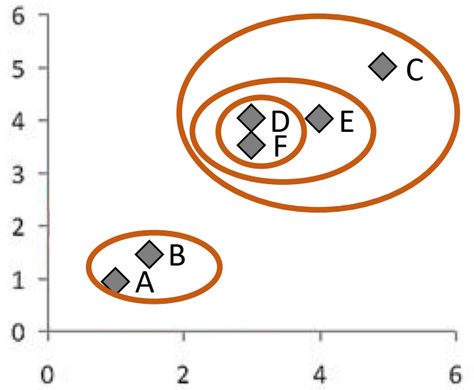
Step through sorted distances joining clusters

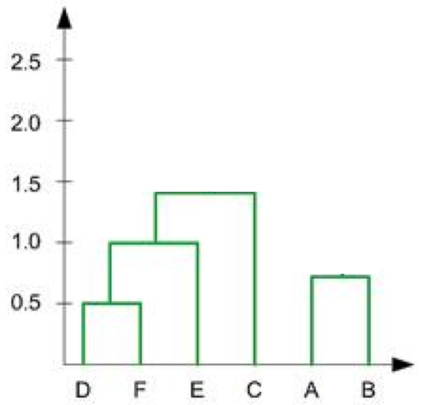




Hierarchical Clustering

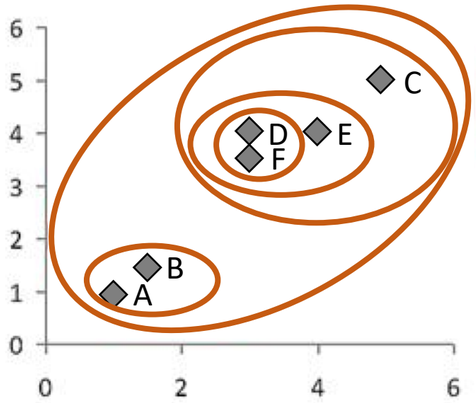
Step through sorted distances joining clusters

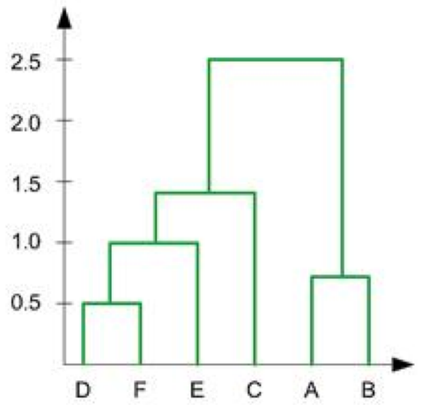




Hierarchical Clustering

Step through sorted distances joining clusters

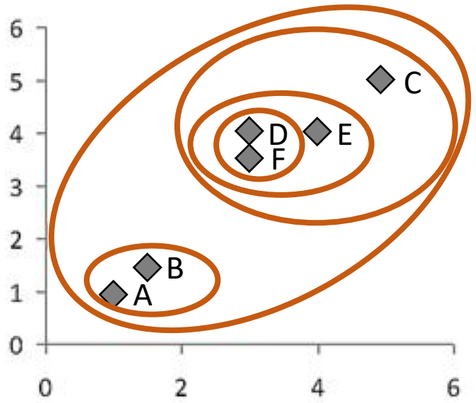


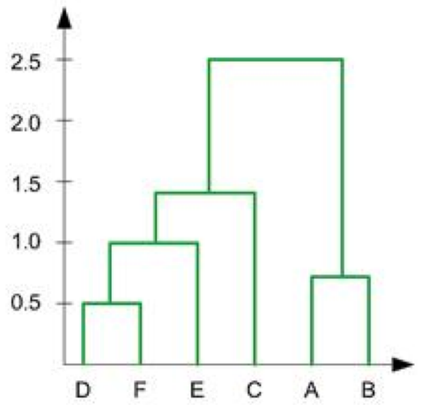


Hierarchical Clustering

Advantages: easy to implement, works on metric space data, tells you all possible clustering events

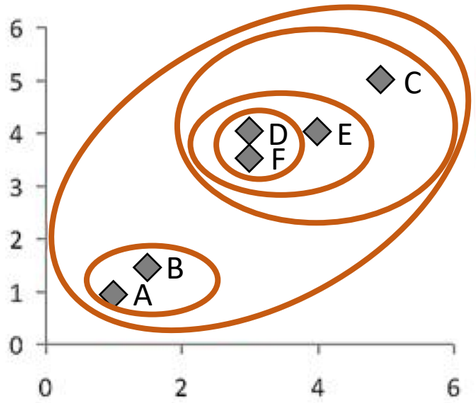
Disadvantages: ?

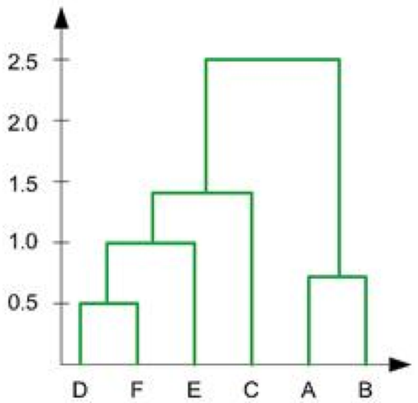




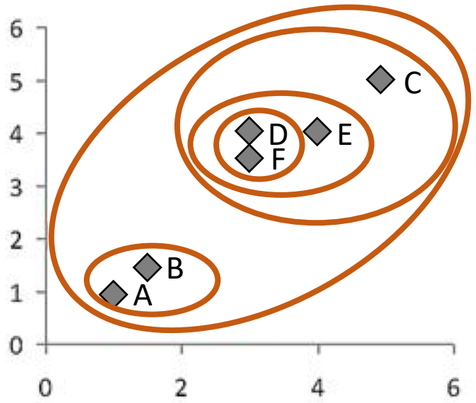
HIERARCHICAL CLUSTERING

Advantages: ?

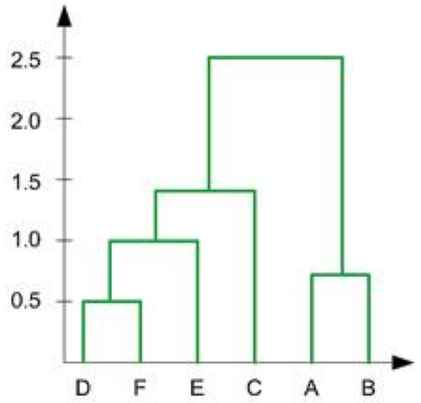




Hierarchical Clustering



Advantages: easy to implement, works on metric space data, tells you all possible clustering events

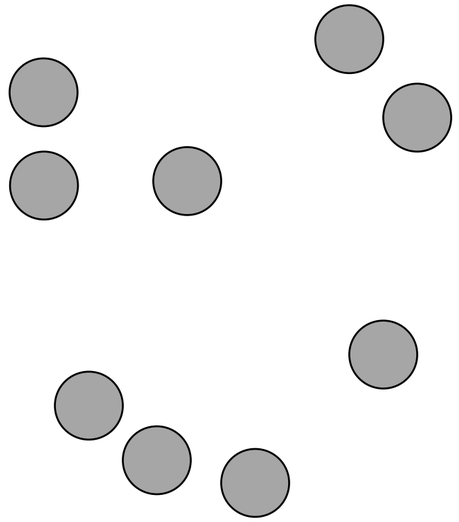


Disadvantages: "How many clusters are there?", takes $O(n^2 \log(n))$ to compute

K-means Clustering

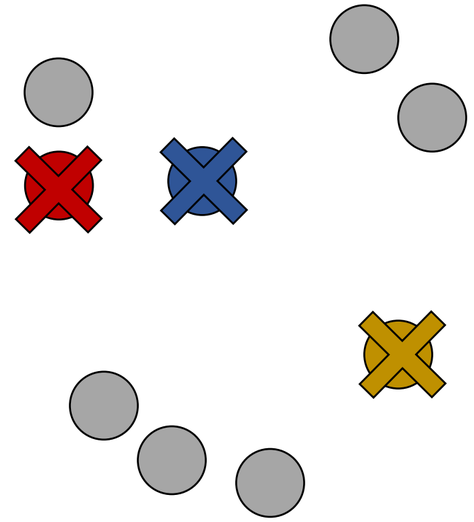
Hierarchical clustering is expensive

For “big data” we prefer something faster (approximate)



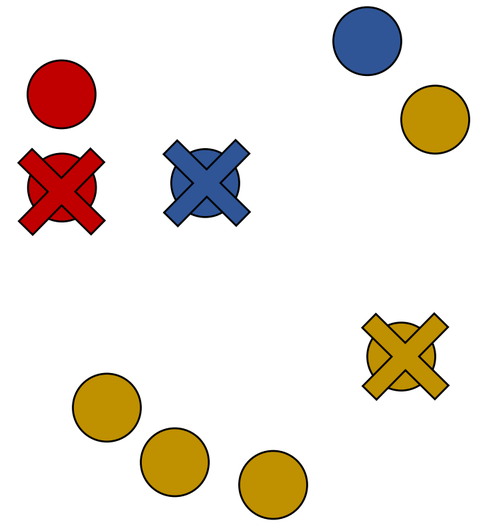
K-means Clustering

Select k points at random as initial cluster means



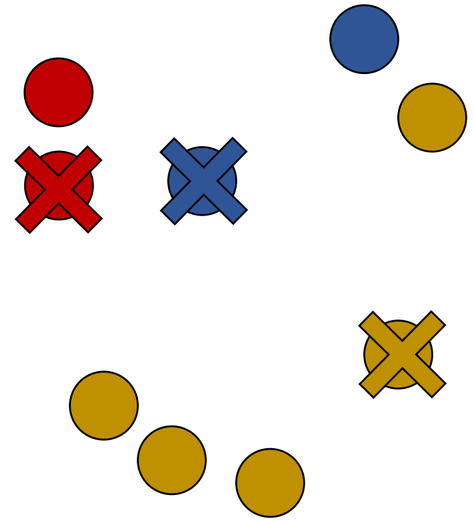
K-means Clustering

Points "join" the cluster they are closest to



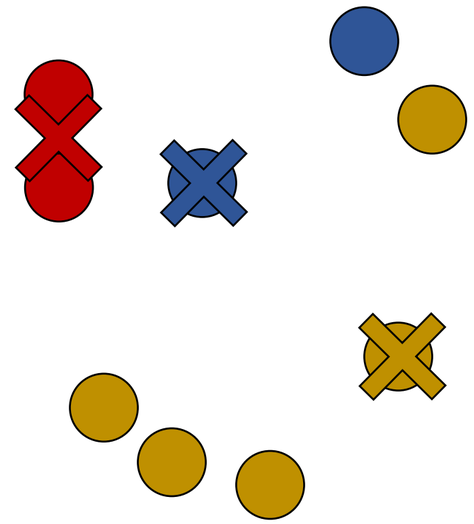
K-means Clustering

New means are calculated based upon cluster elements



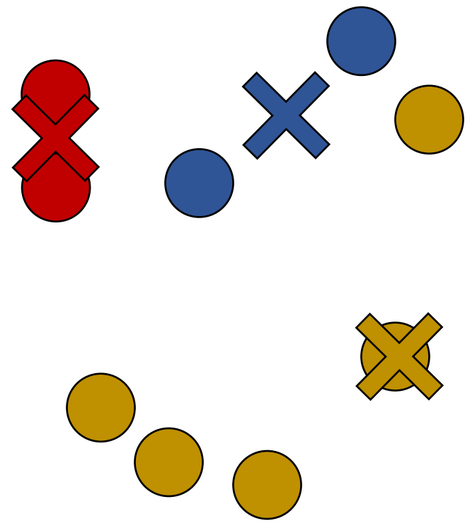
K-means Clustering

New means are calculated based upon cluster elements



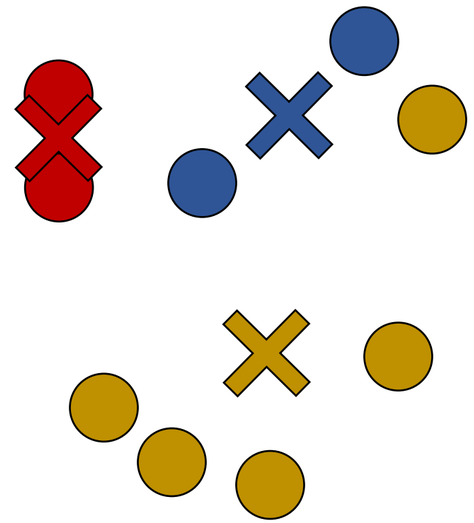
K-means Clustering

New means are calculated based upon cluster elements



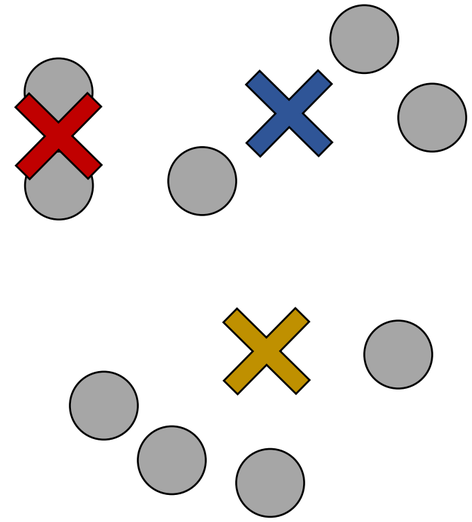
K-means Clustering

New means are calculated based upon cluster elements



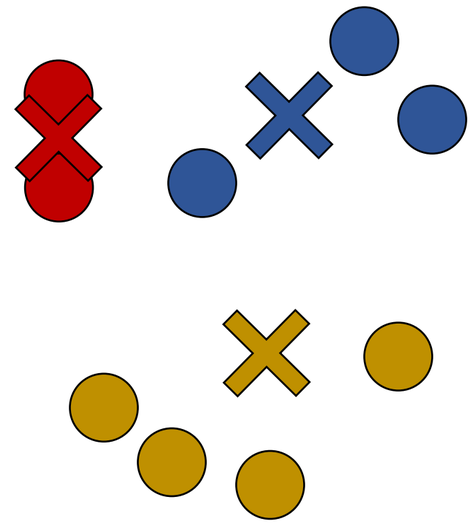
K-means Clustering

Recluster points



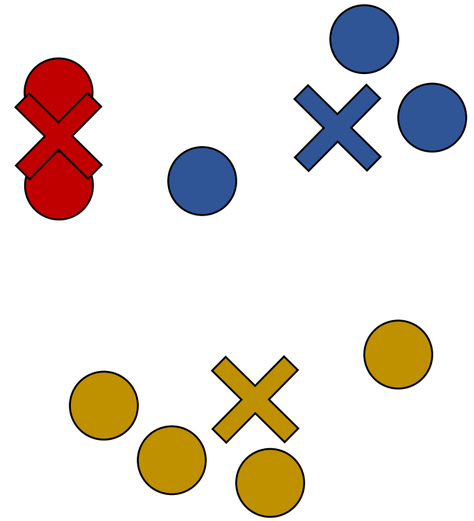
K-means Clustering

Recluster points



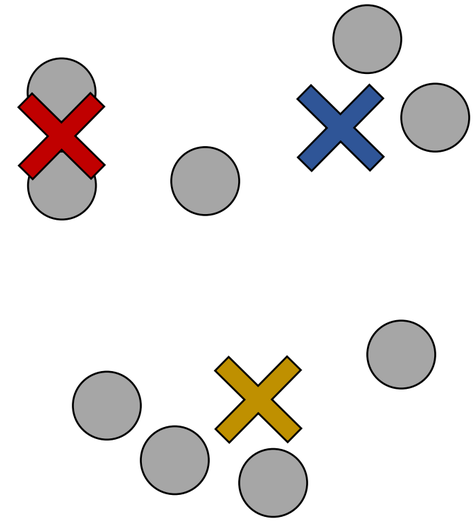
K-means Clustering

Recalculate means



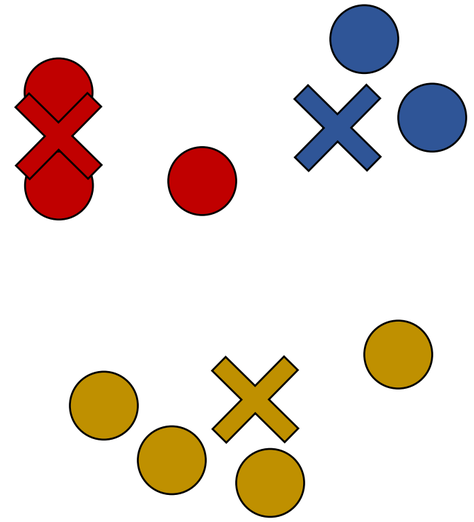
K-means Clustering

Recluster



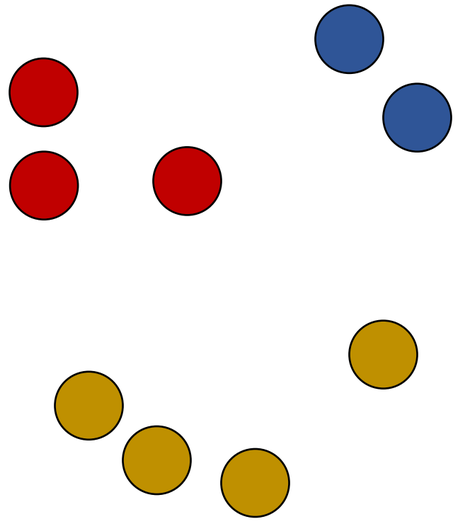
K-MEANS CLUSTERING

Recluster



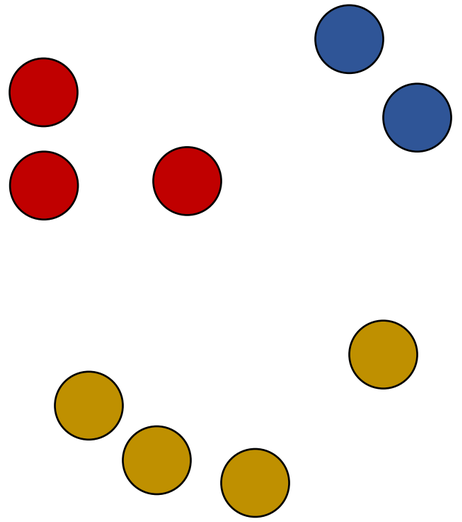
K-means Clustering

Process can be iterated until a stopping condition is reached, such as a fixed number of iterations or number of points changing clusters



K-means Clustering

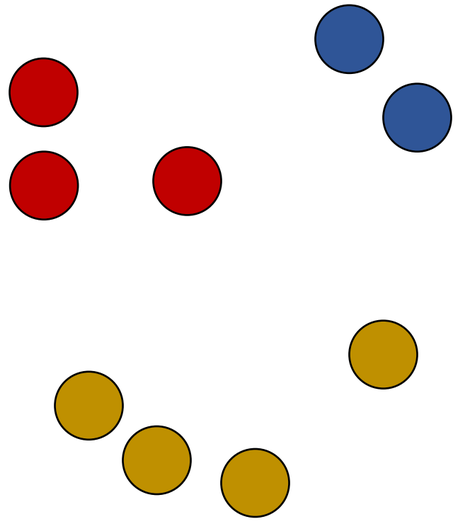
Advantages: ?



K-means Clustering

Advantages: Finds good clusters in many datasets, each iteration is only $O(kn)$

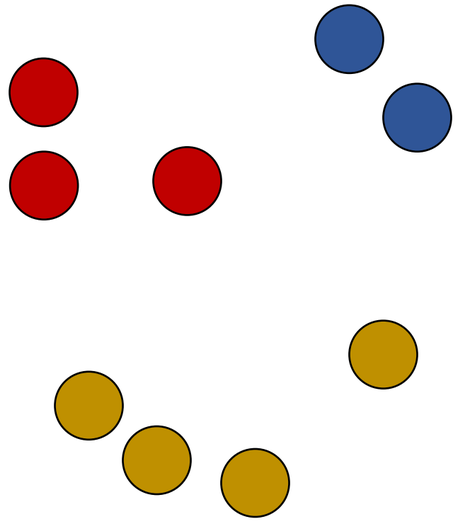
Disadvantages: ?



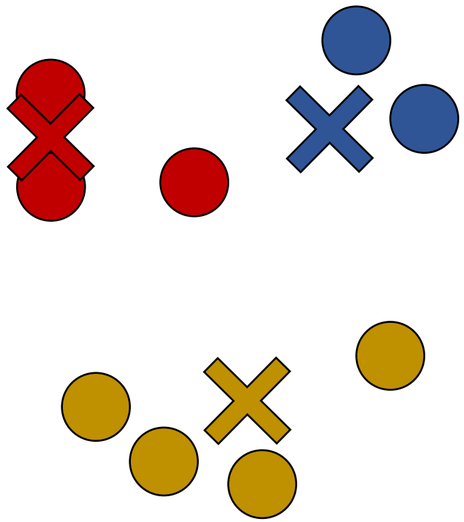
K-means Clustering

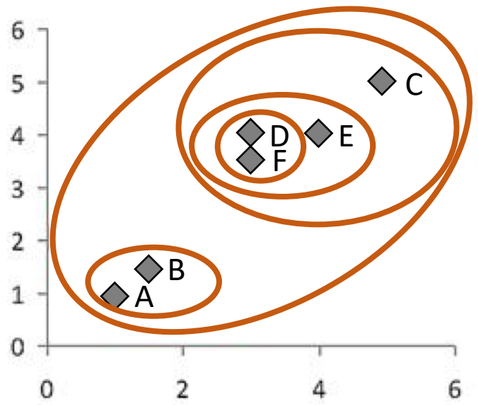
Advantages: Finds good clusters in many datasets, each iteration is only $O(kn)$

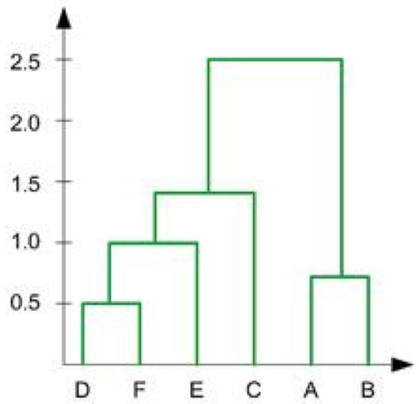
Disadvantages: How to pick k, when is it good enough to stop, initial point selection can change clusters



So… How do we use clustering in our visualizations?







SUMMARY

Lots of powerful tools available

The challenge/opportunity in knowing what data they provide and their limitations. Provide as much of this information to your consumer as possible.

SUMMARY

Try to think about the tools you pick within the context of the visualization task that needs to be performed.

Test multiple tools effectiveness, but avoid adding tools just for the sake of adding them.

Next Class

Computational Topology and its use as a data mining tool