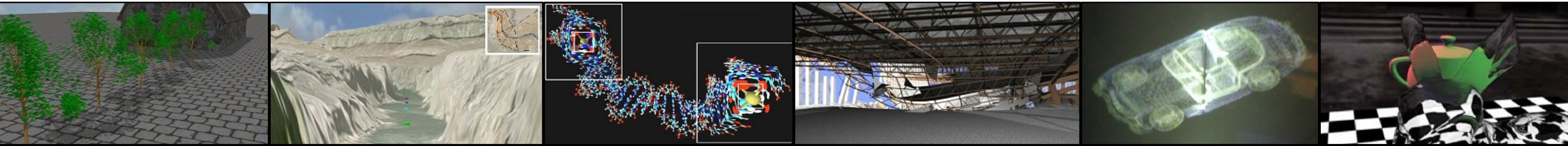
CIS 4930/6930-002
DATA VISUALIZATION

Classifiers and Clustering

Paul Rosen
Assistant Professor
University of South Florida

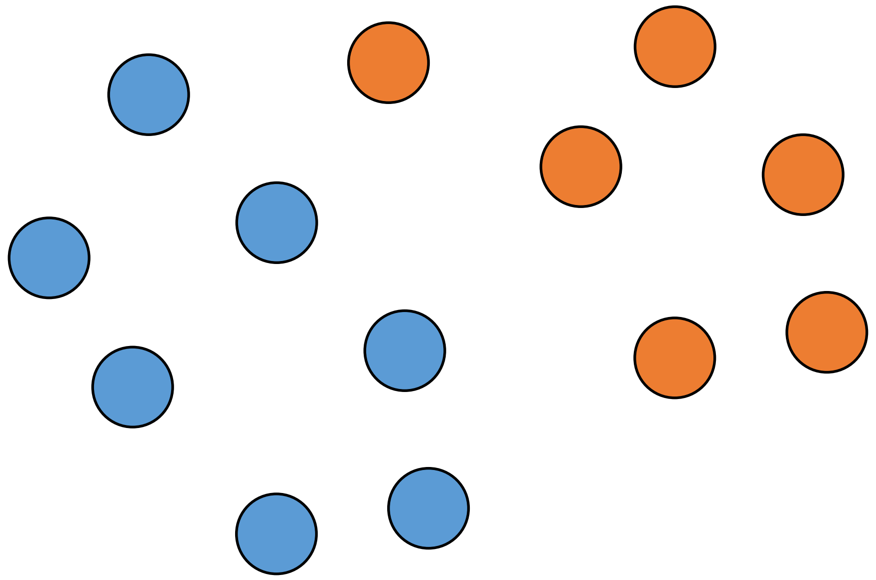


CLASSIFIERS & CLUSTERING

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

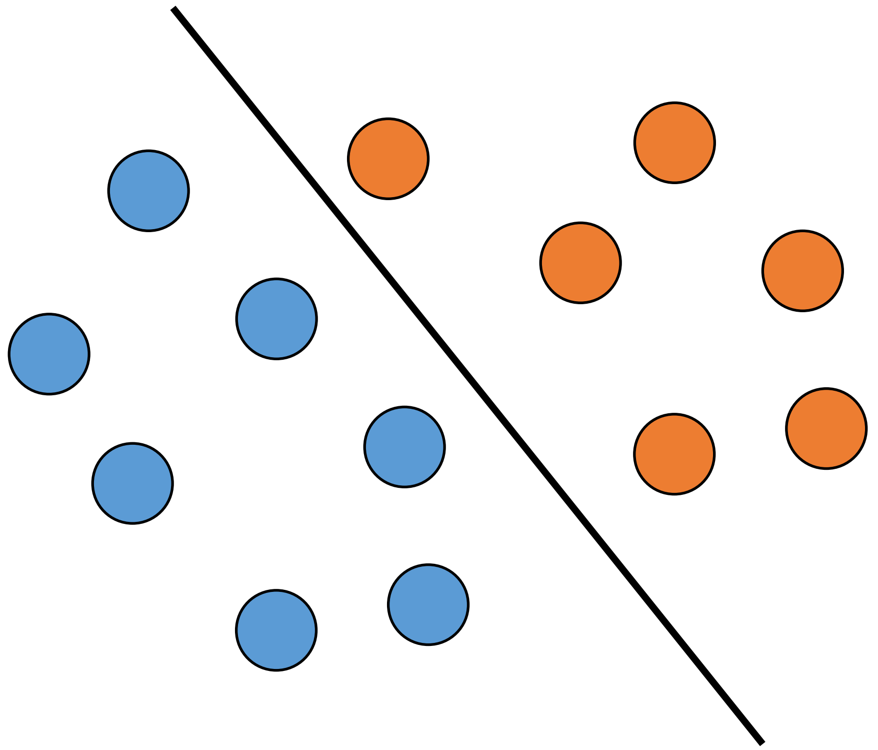
CLASSIFIERS

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



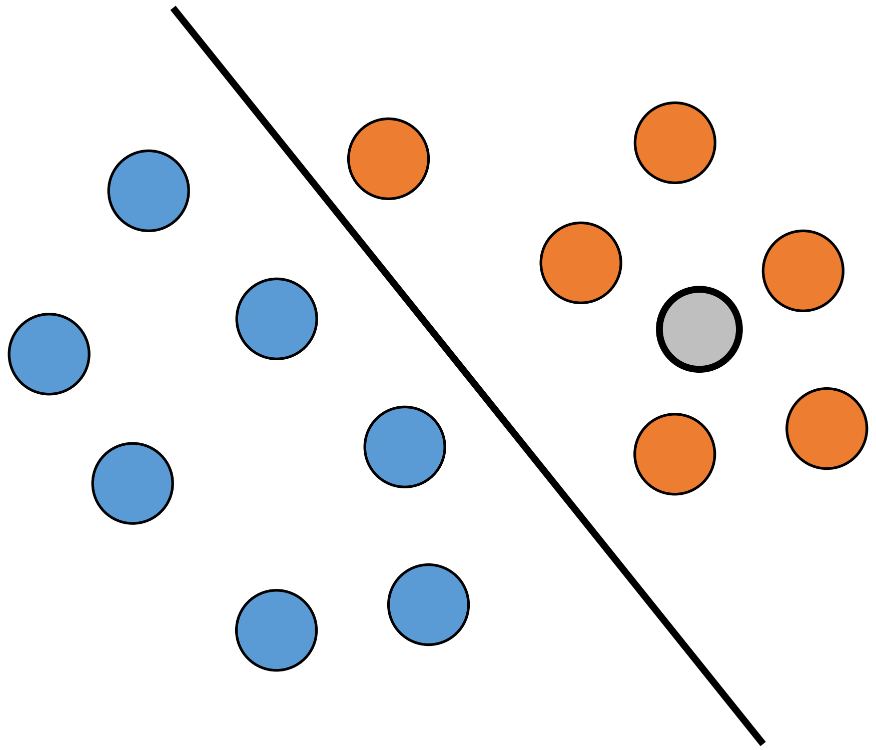
CLASSIFIERS

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



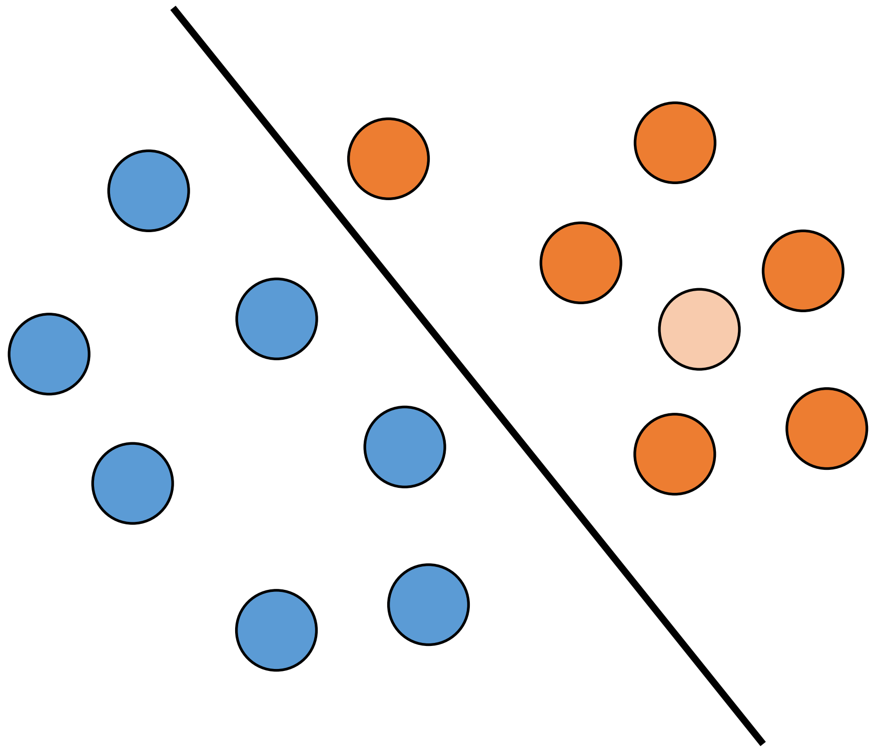
CLASSIFIERS

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



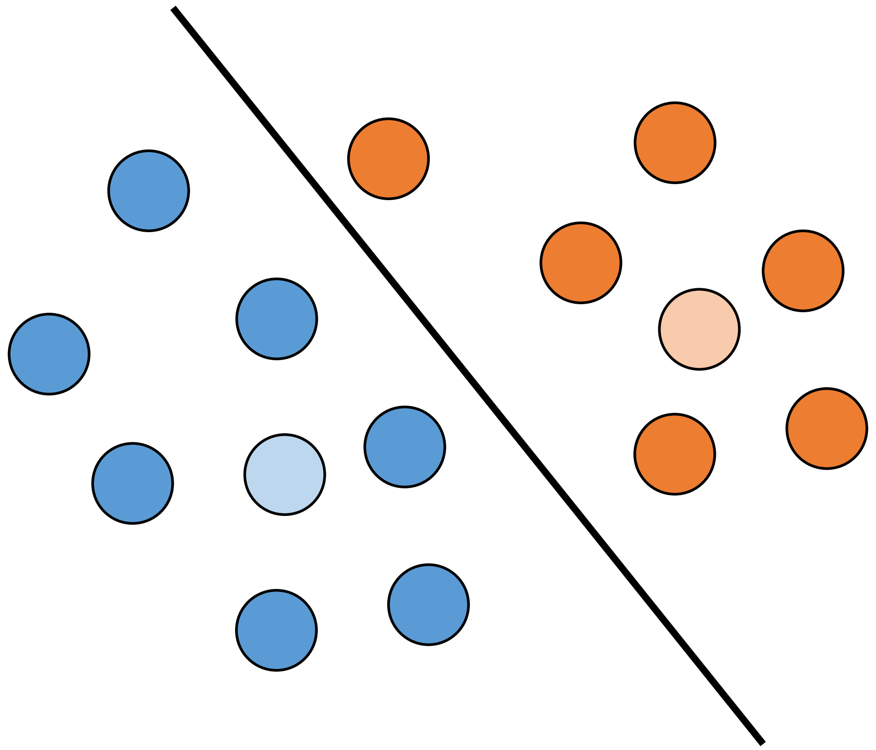
Classifiers

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



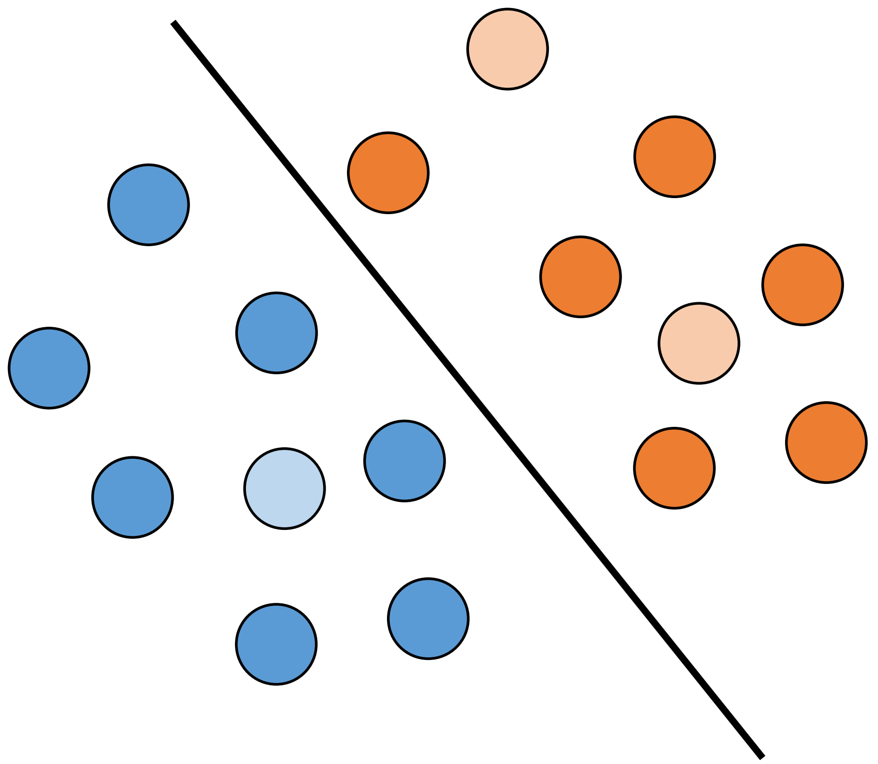
CLASSIFIERS

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…

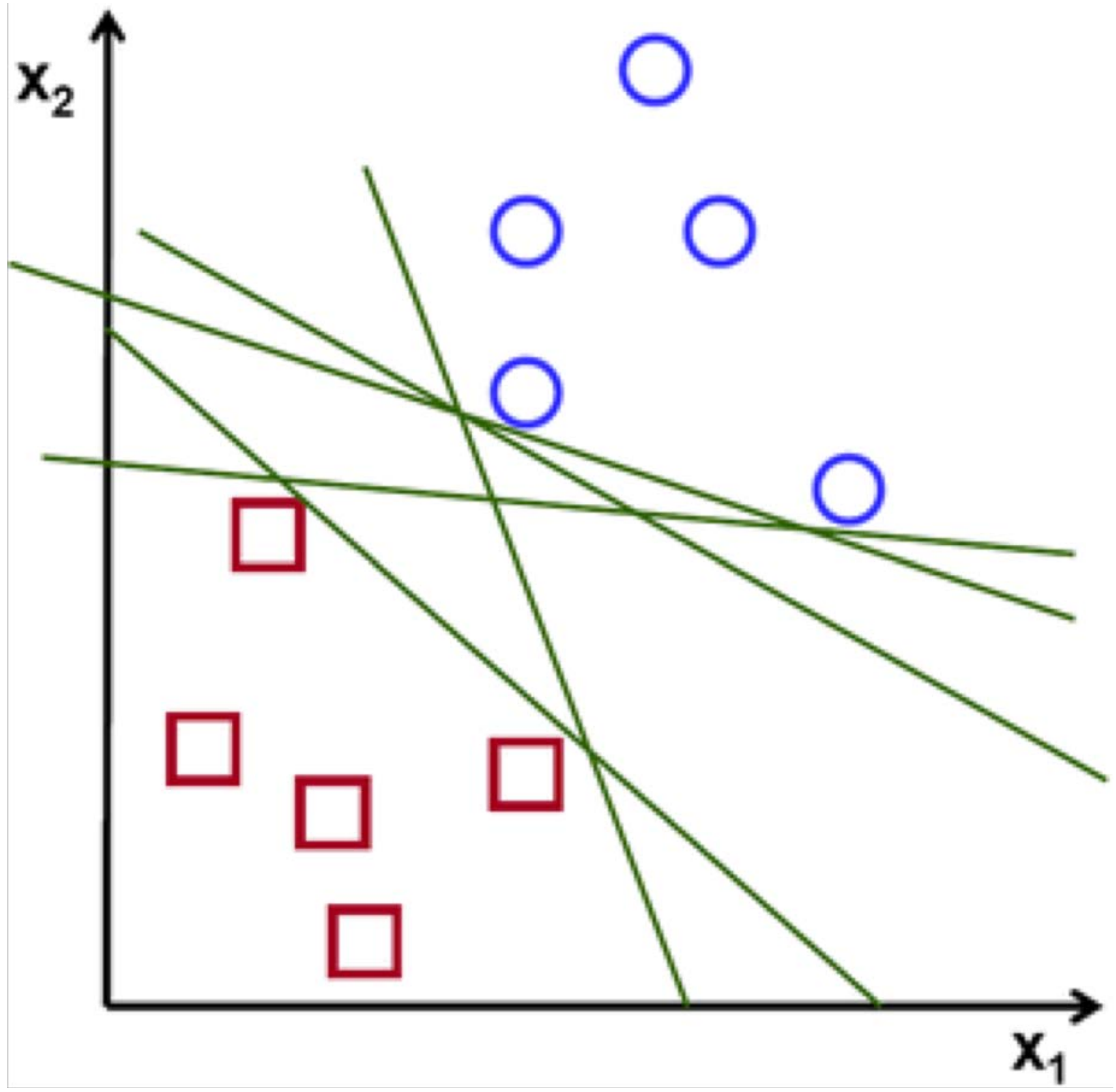
Binary classifiers : **Kernel Support Vector Machine**, Linear Discriminant Analysis, **Linear Support Vector Machine**

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/

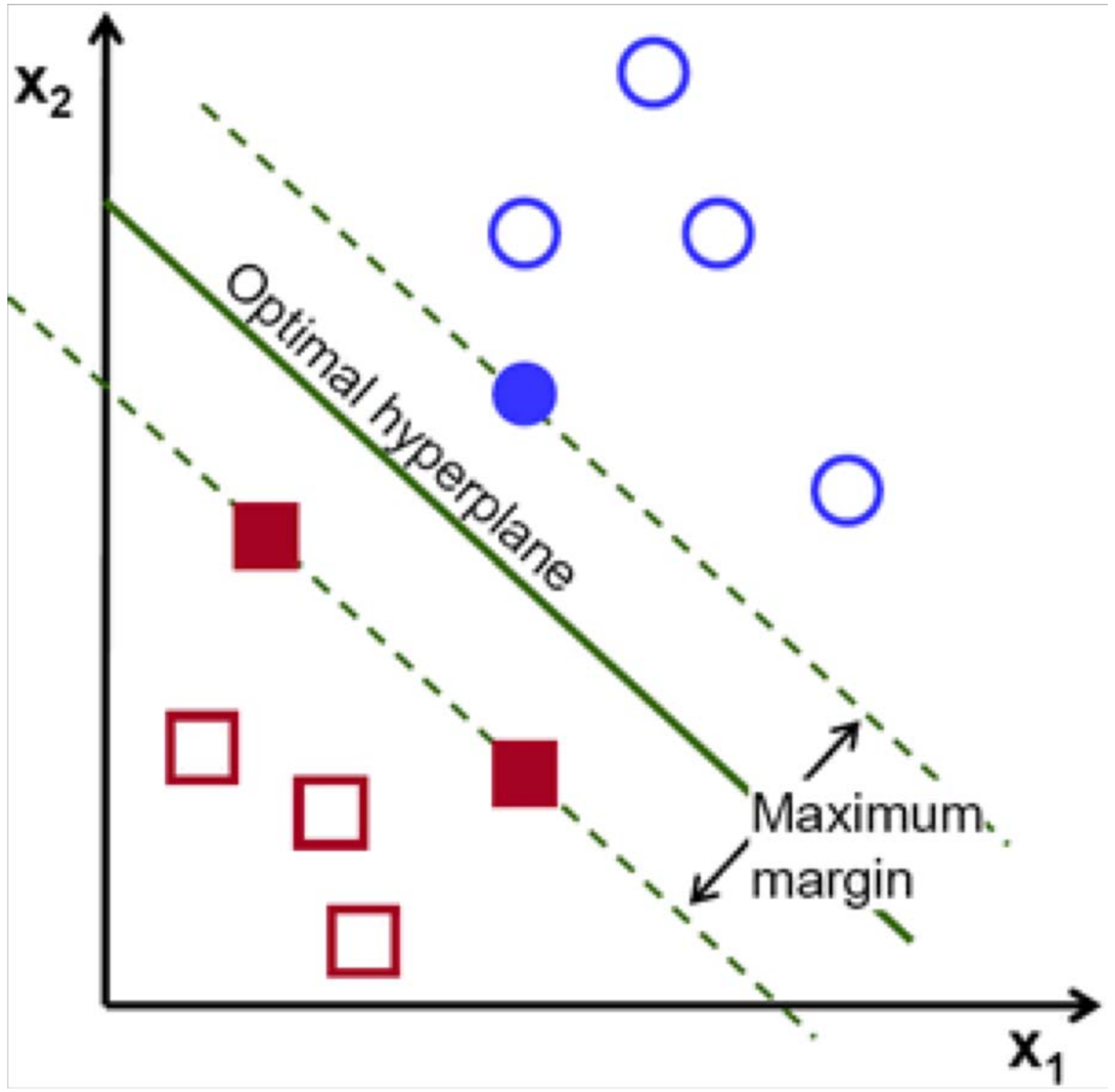
Linear Support Vector Machine (SVM)

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



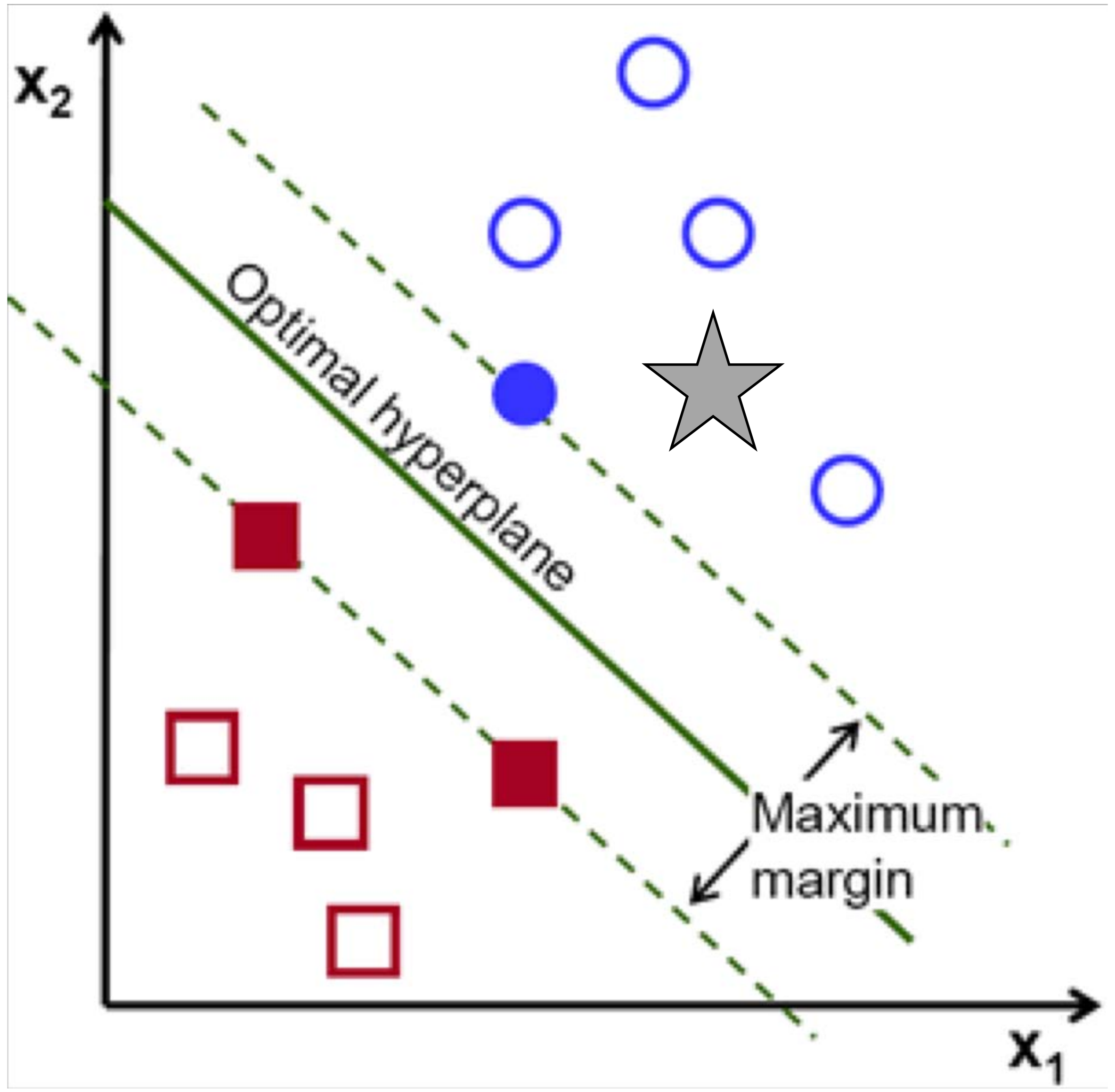
LINEAR SUPPORT VECTOR MACHINE (SVM)

Select the optimal hyperplane that maximizes the margin between classes



LINEAR SUPPORT VECTOR MACHINE (SVM)

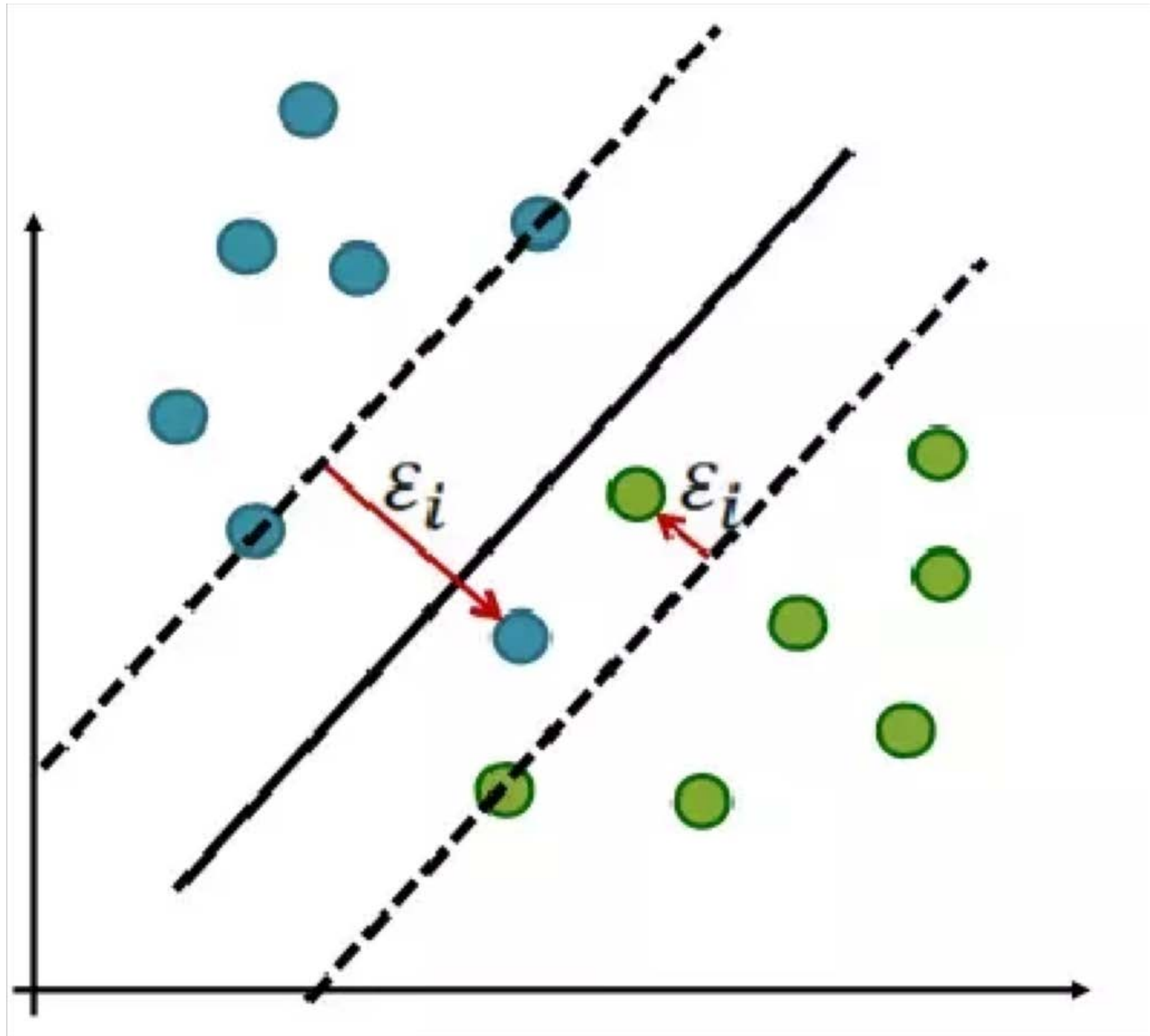
Now testing points can be classified



Linear Support Vector Machine (SVM)

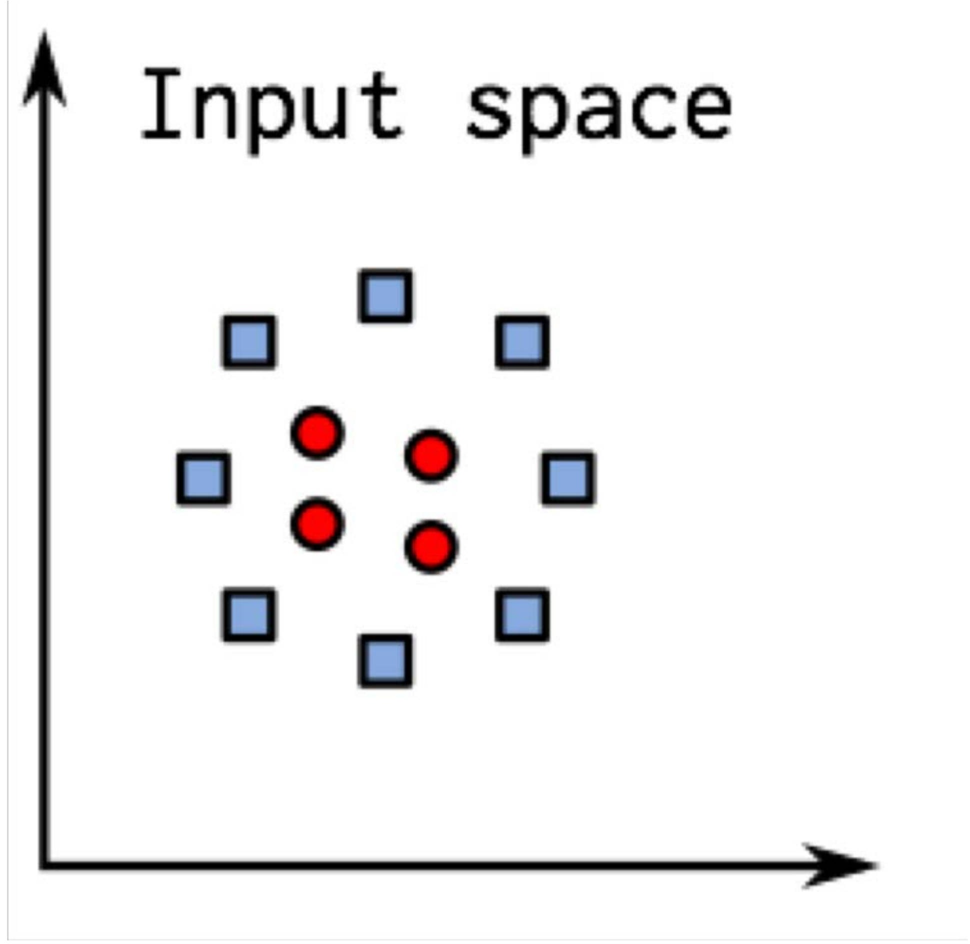
Data is often not linearly separable

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



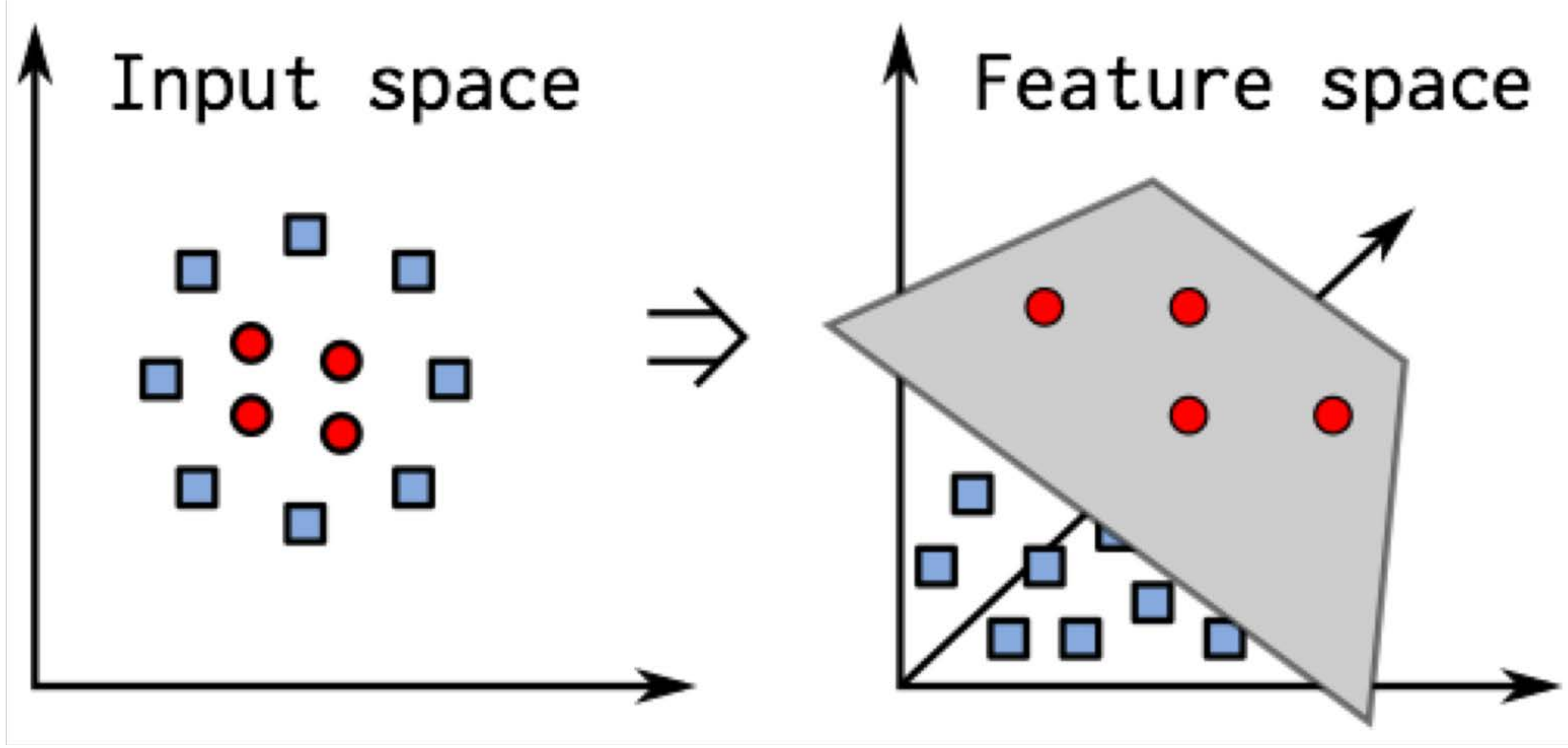
KERNEL SVM

Often data cannot be easily separated linearly



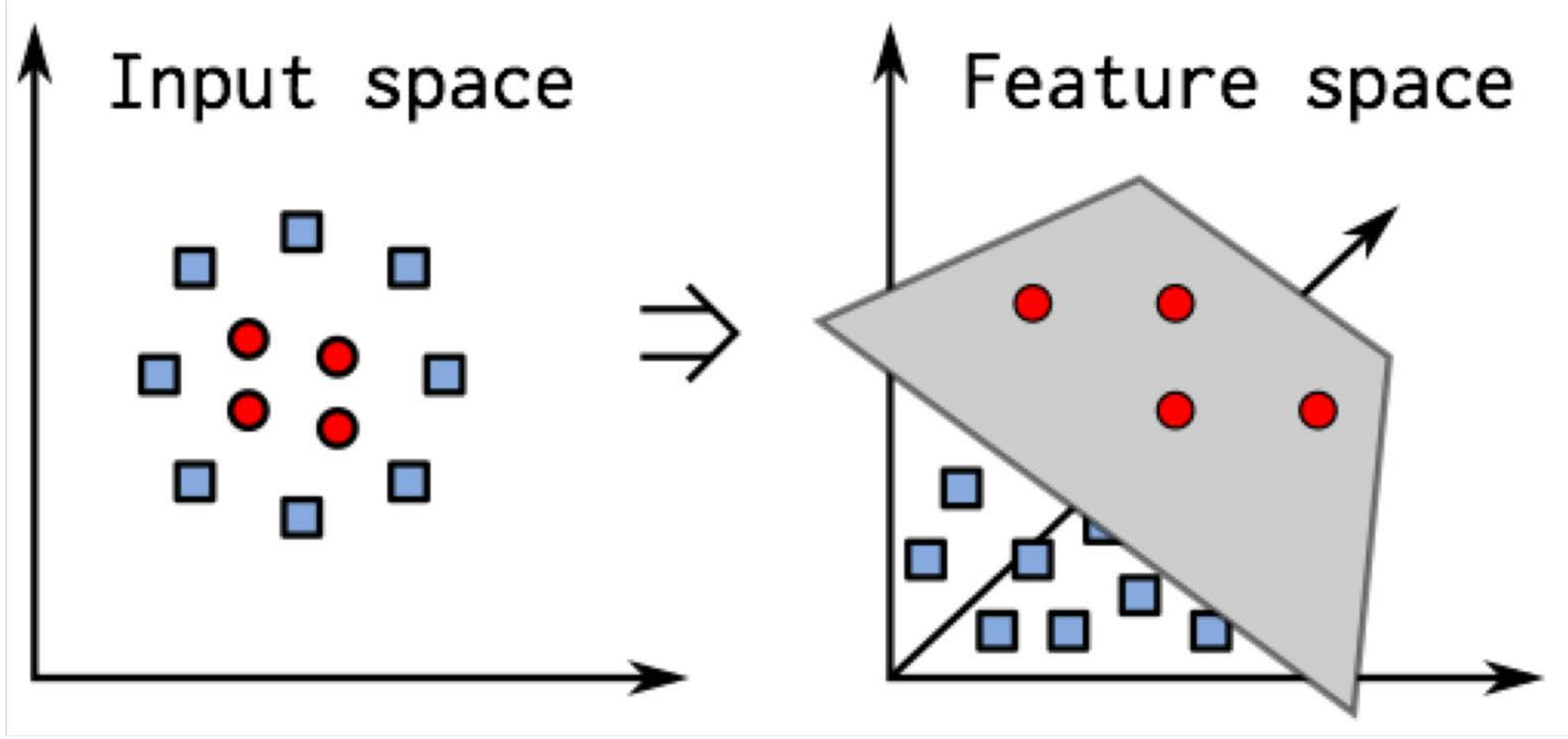
KERNEL SVM

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



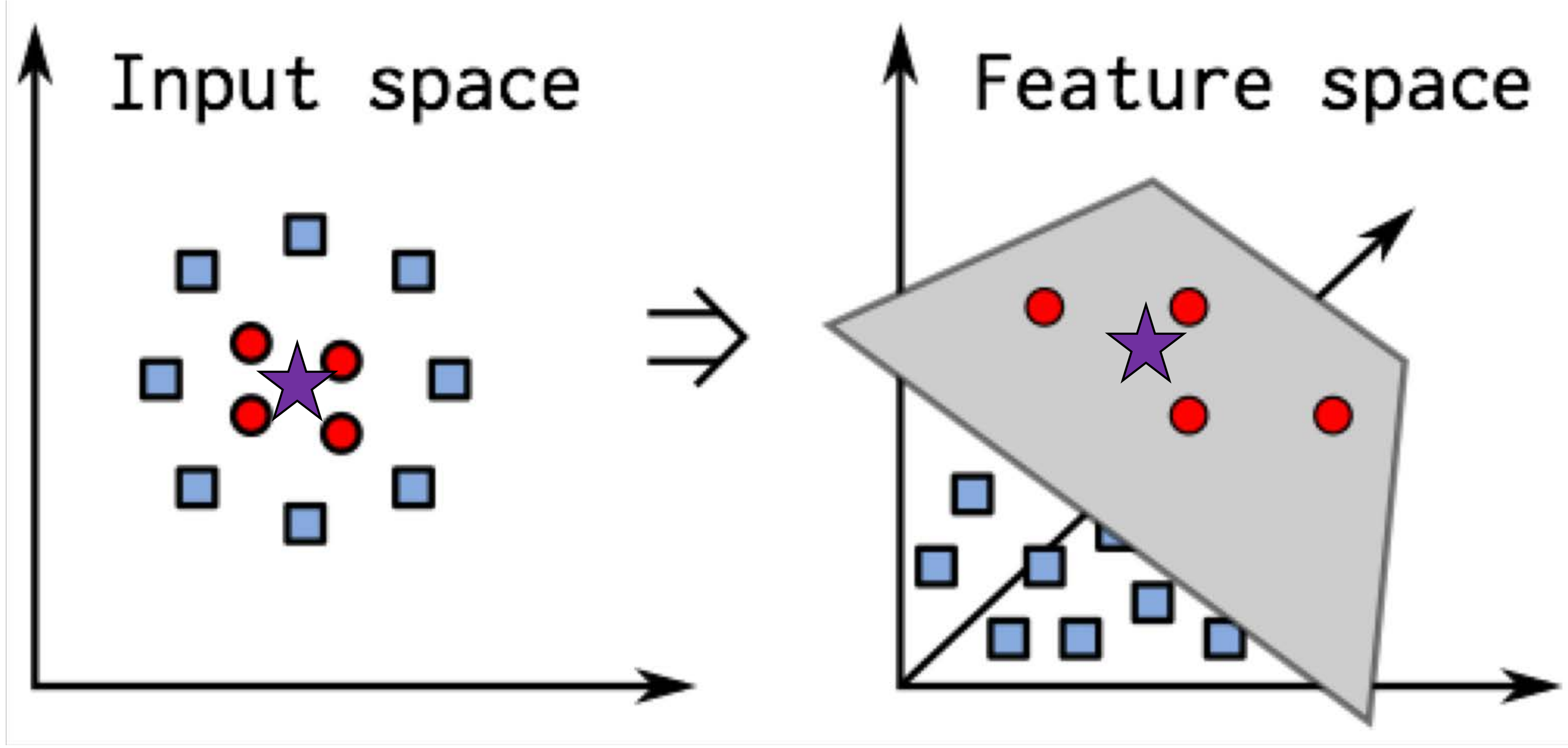
Kernel SVM

Run SVM on the feature space



KERNEL SVM

Test a point by converting it to feature space

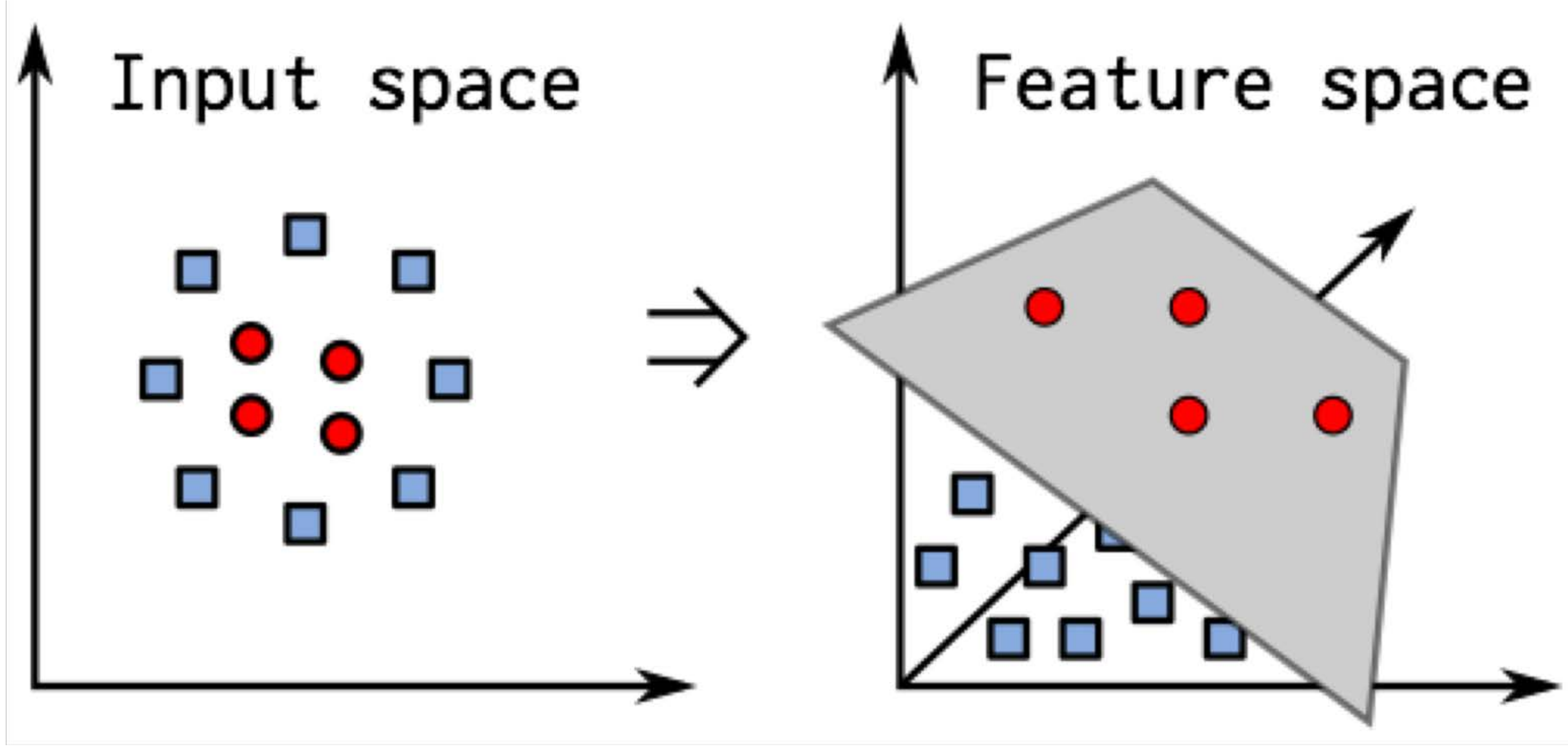


Kernel SVM

Wide variety of kernels available

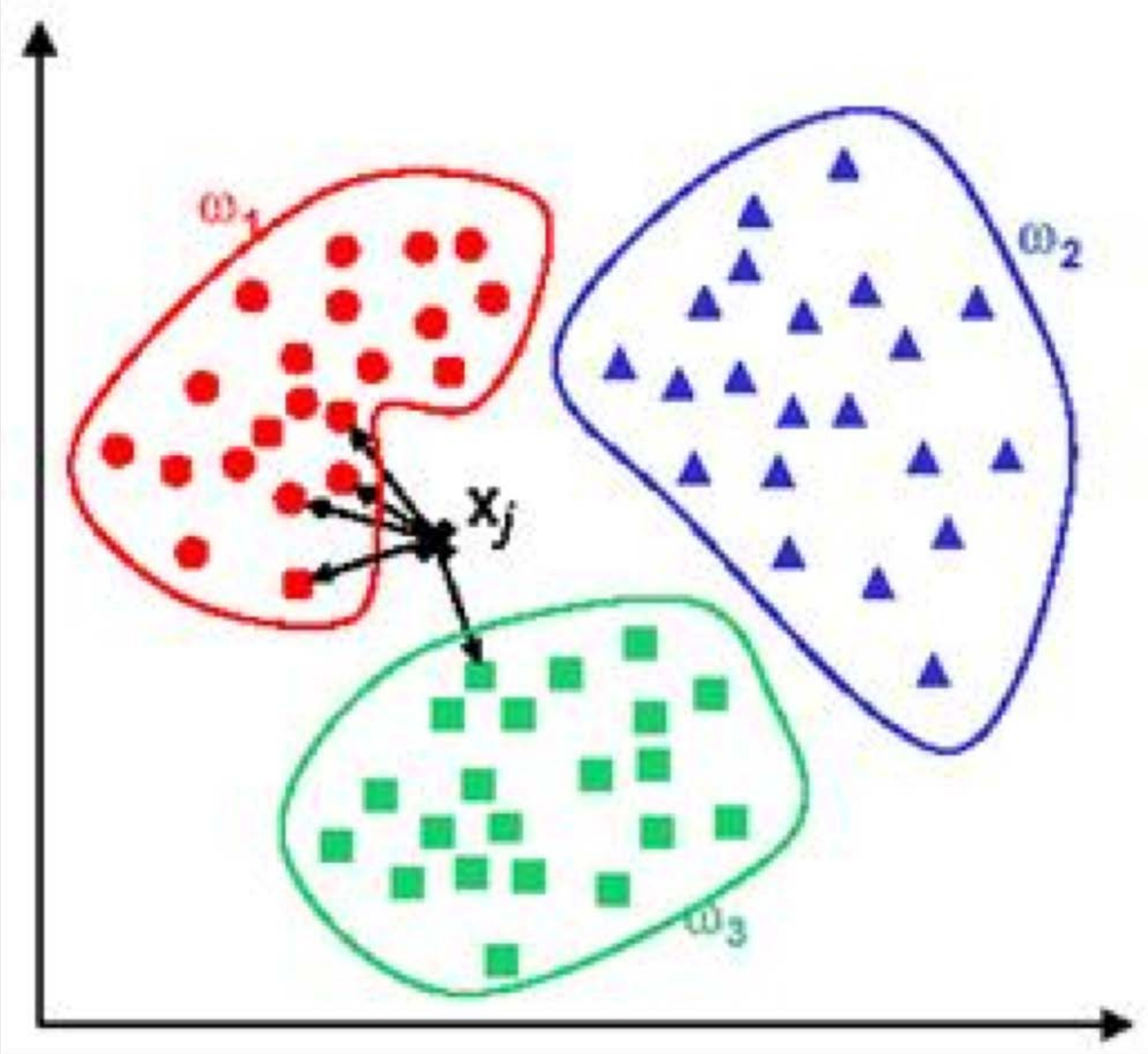
Gaussian, Fisher, Graph, RBF, Polynomial,

Which to use on your data?



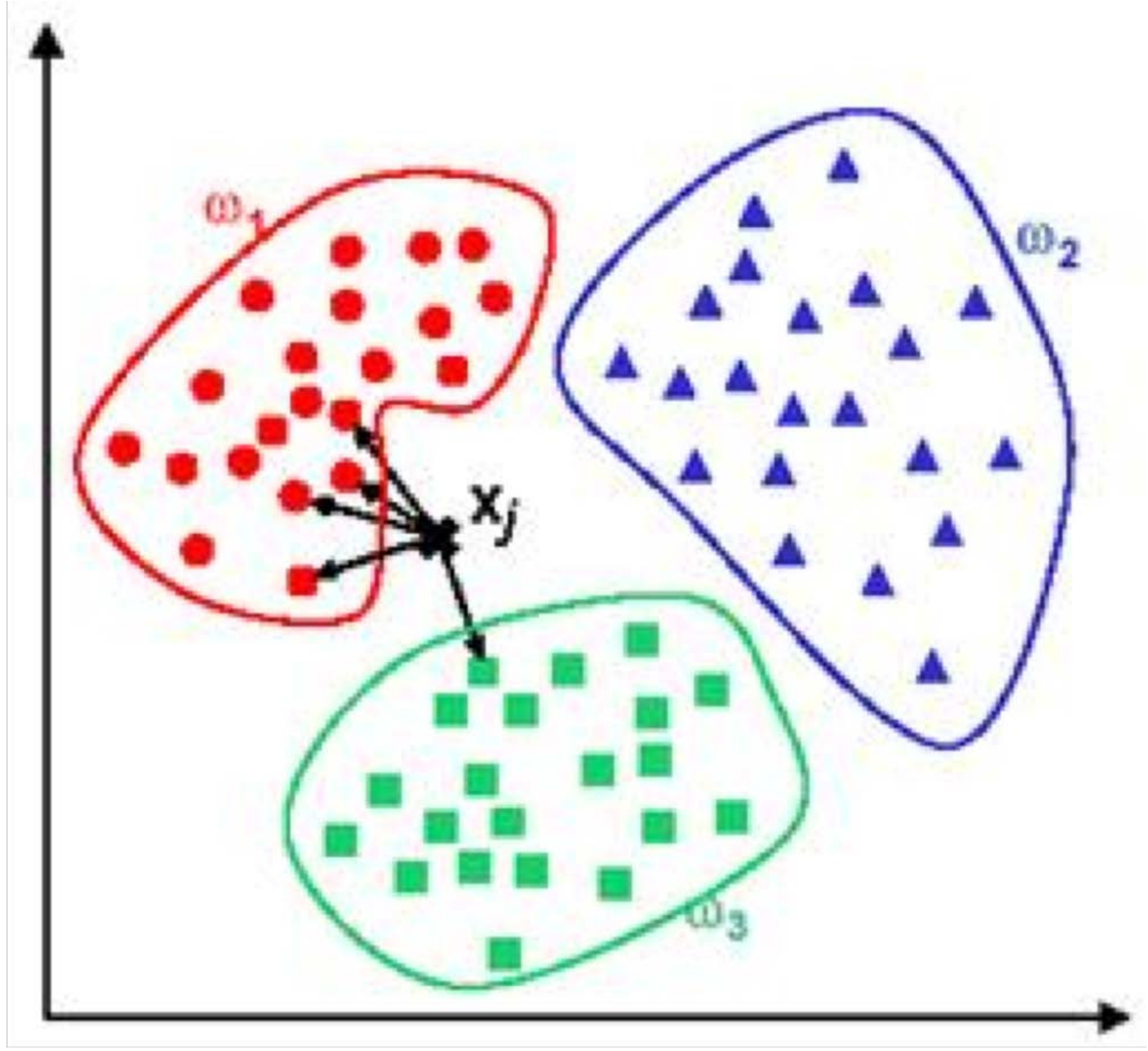
<u>K NEAREST NEIGHBORS CLASSIFIER</u>

Multiclassifier based upon proximity



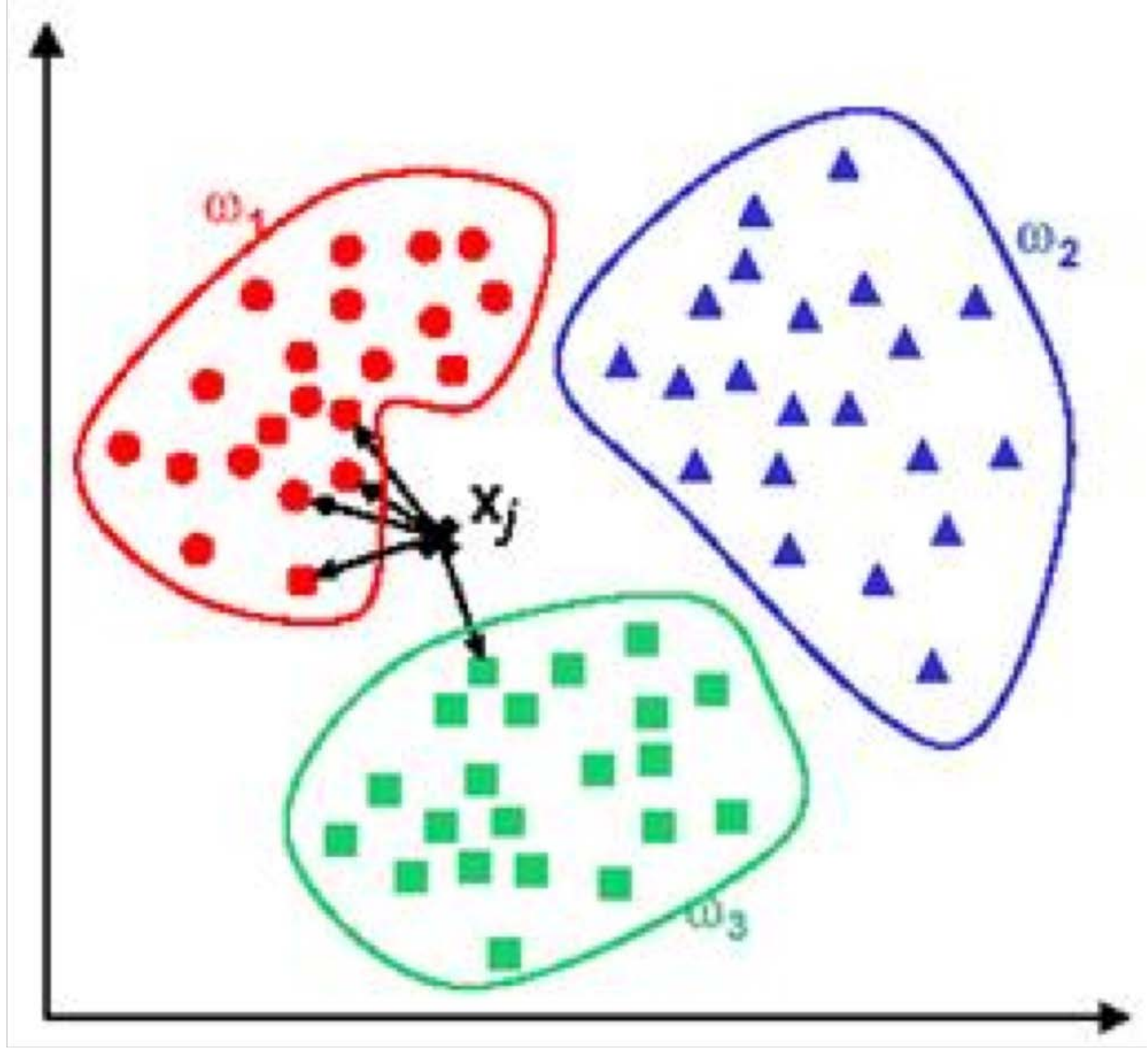
K NEAREST NEIGHBORS CLASSIFIER

Finding the closest k training points



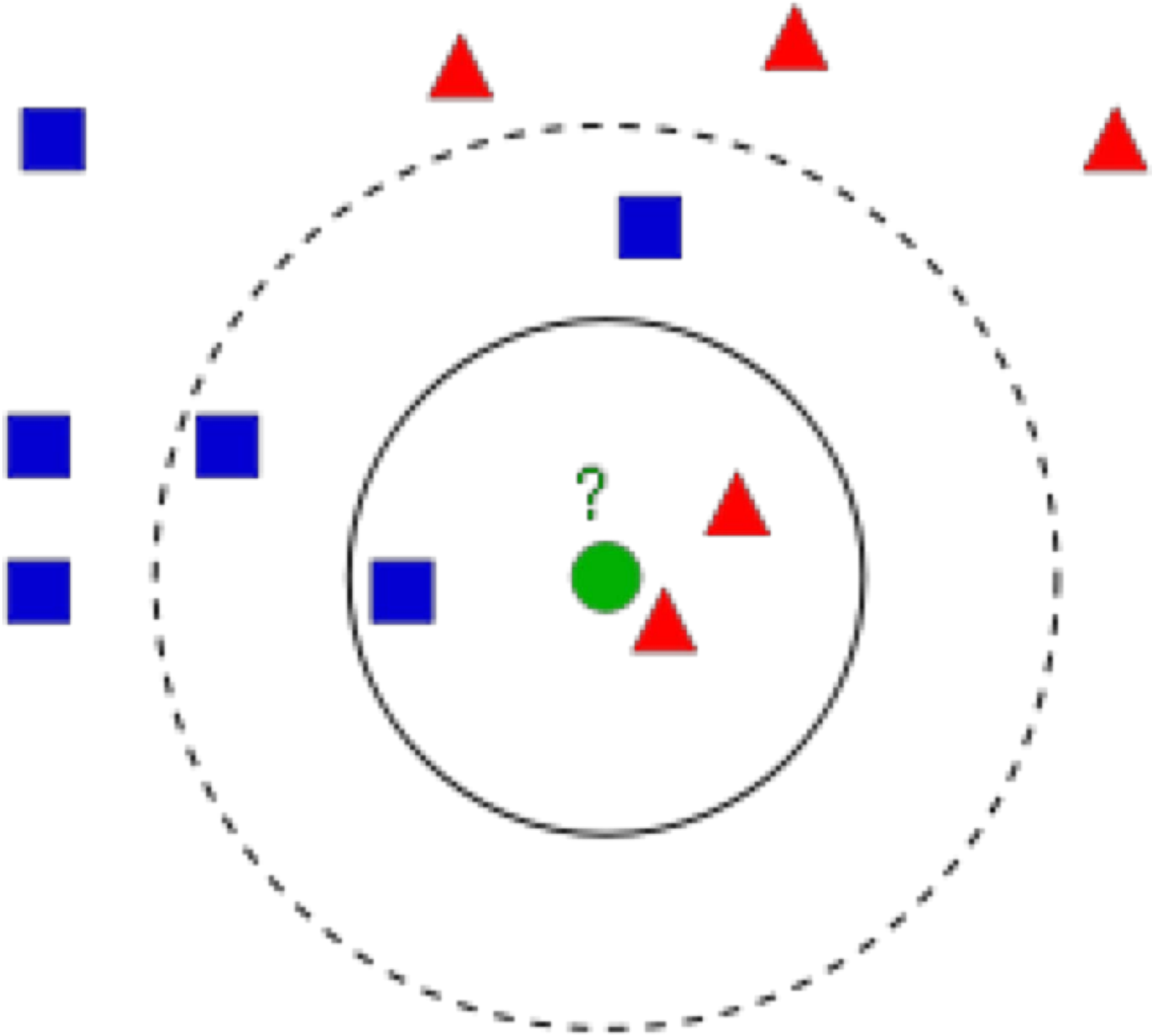
K NEAREST NEIGHBORS CLASSIFIER

those points vote on the class of the testing point



K NEAREST NEIGHBORS CLASSIFIER

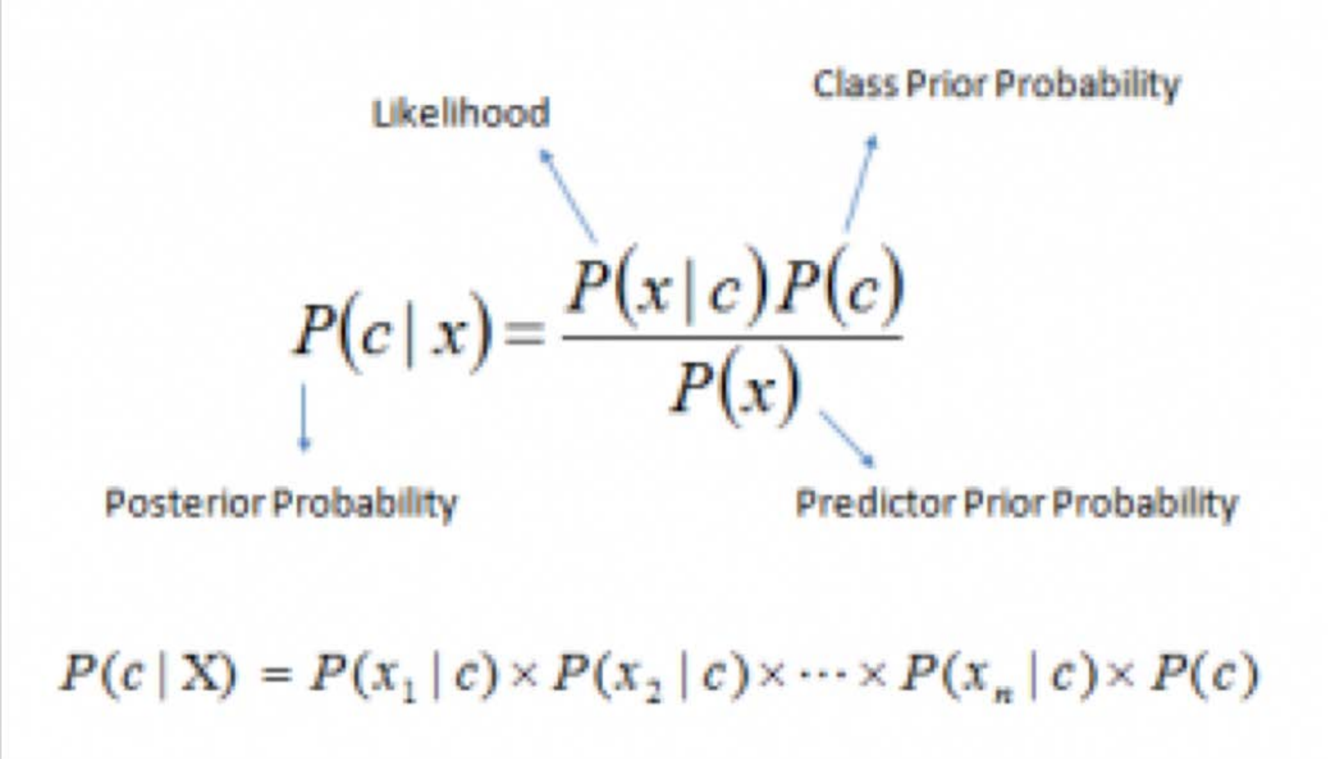
Problem: selecting k is hard



(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

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

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

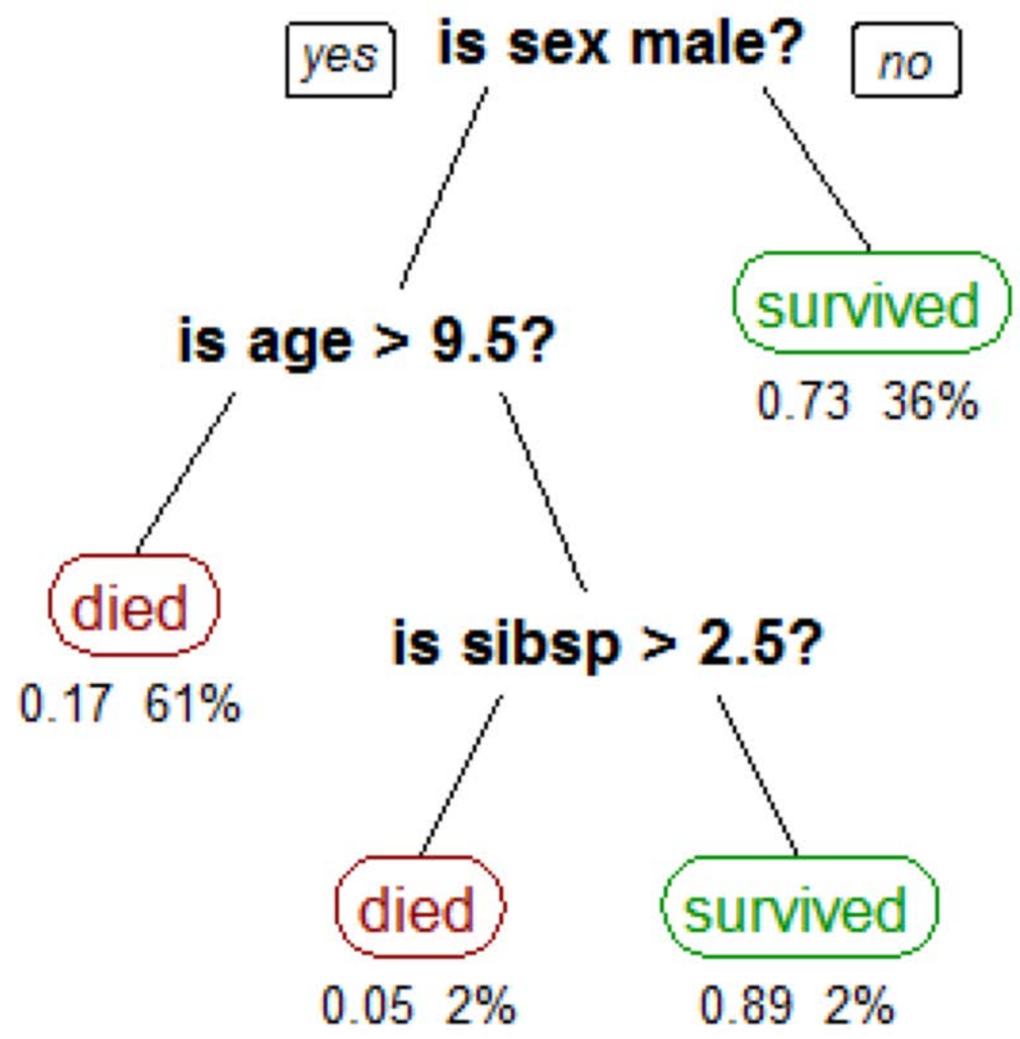
Gaussian Naive Bayes

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

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

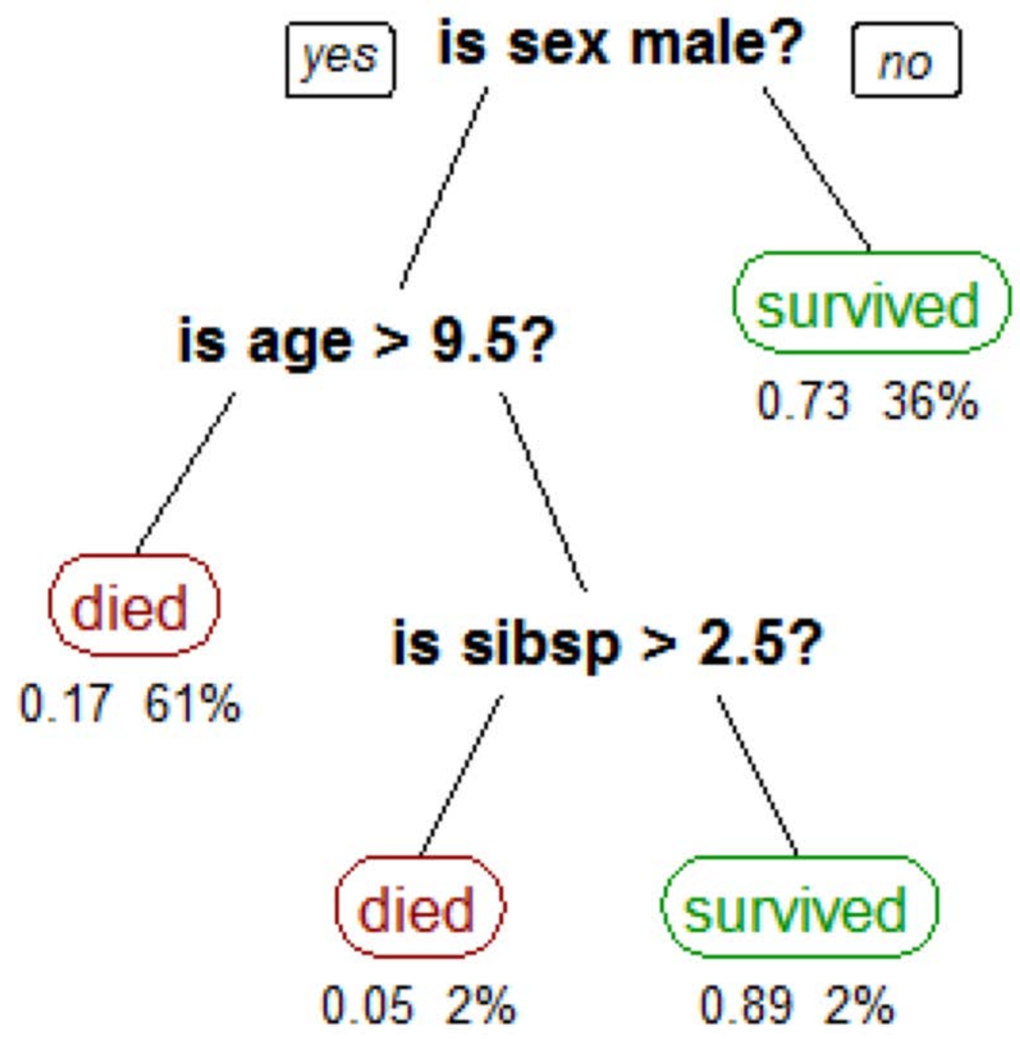
DECISION TREE AND RANDOM FOREST

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



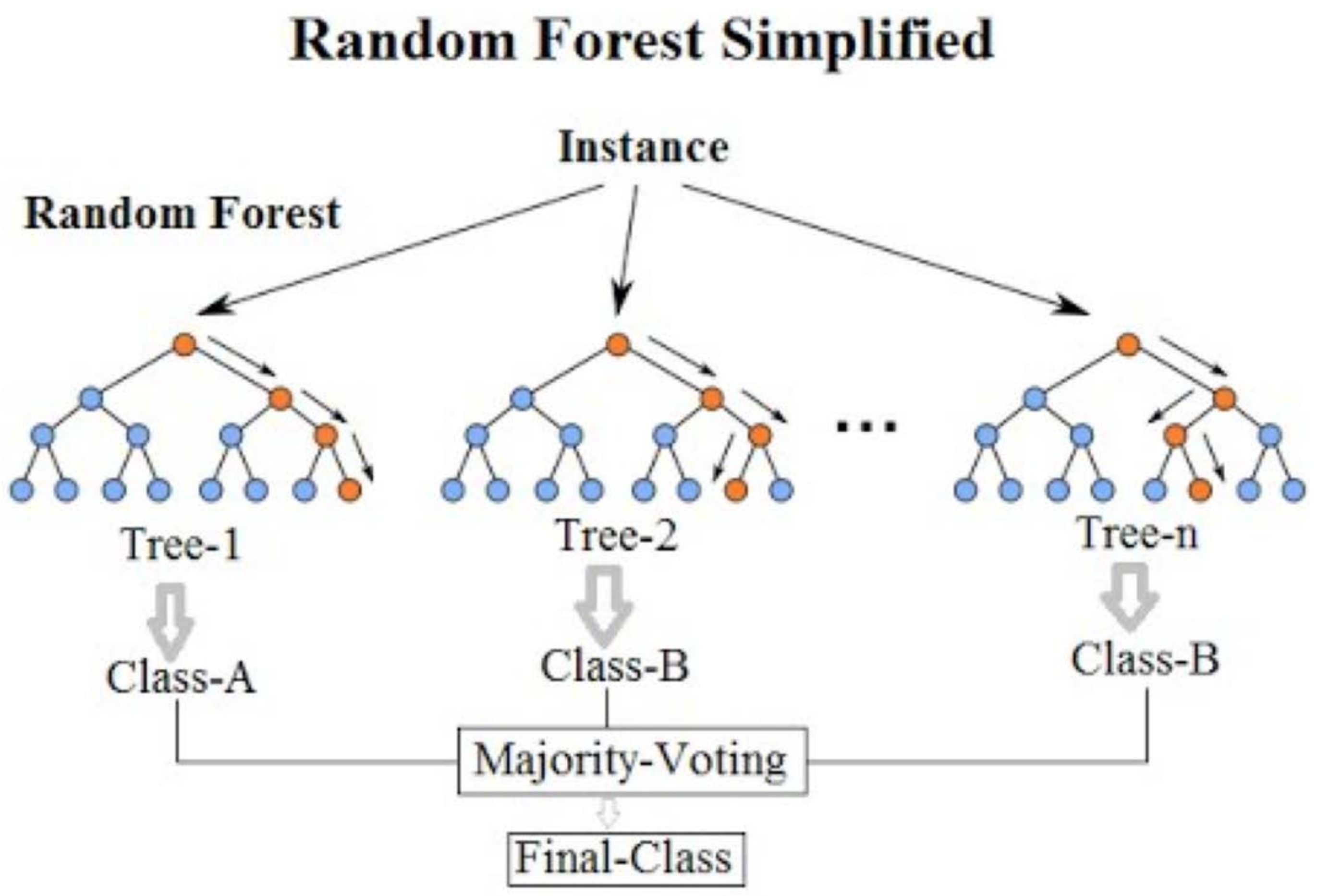
Decision Tree and Random Forest

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

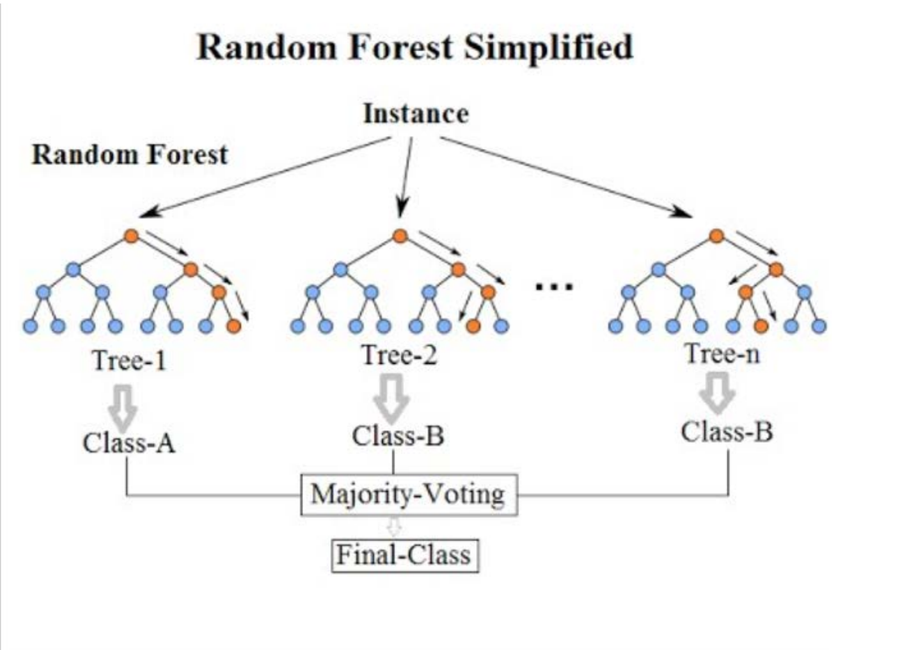


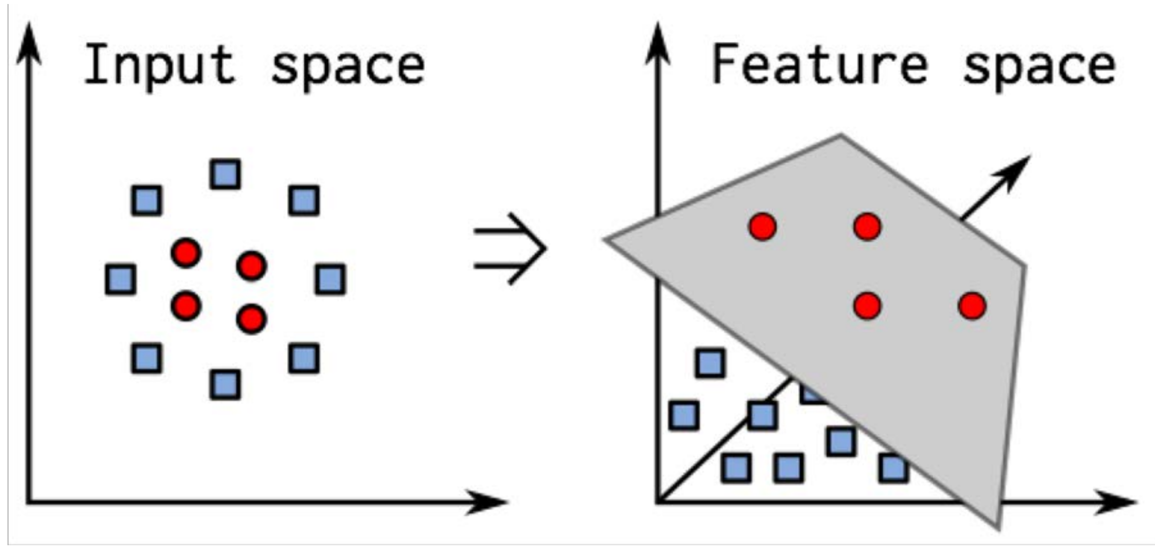
Decision Tree and Random Forest

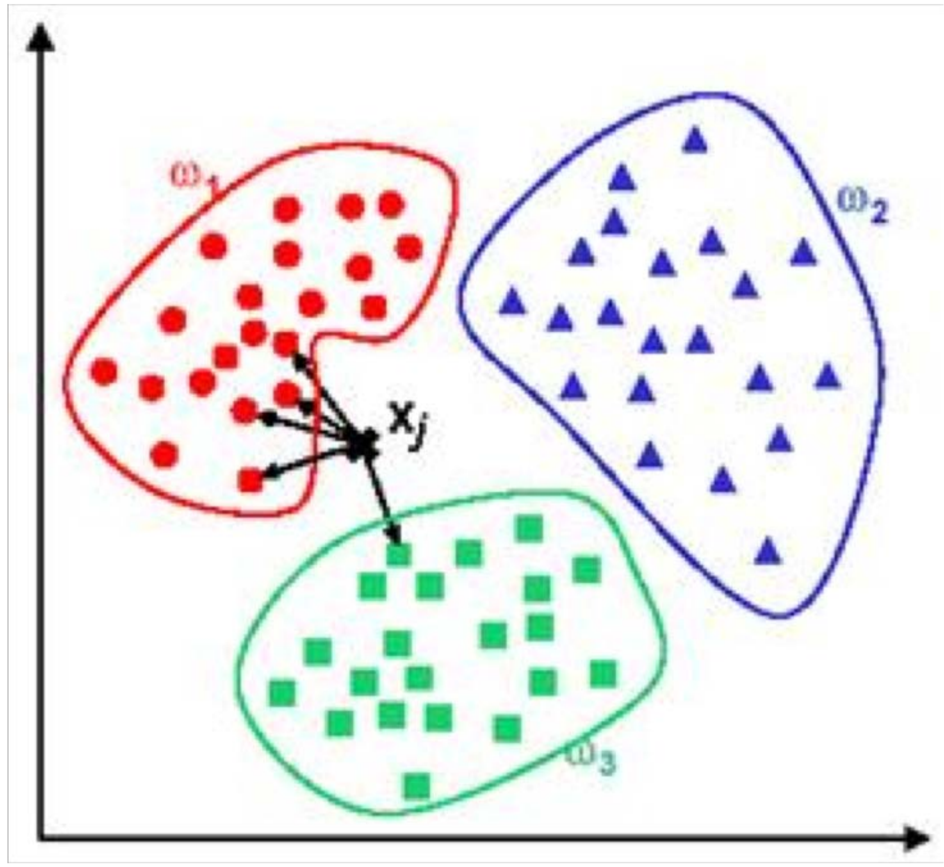
Random forest uses multiple decision trees to vote on a classification testing data

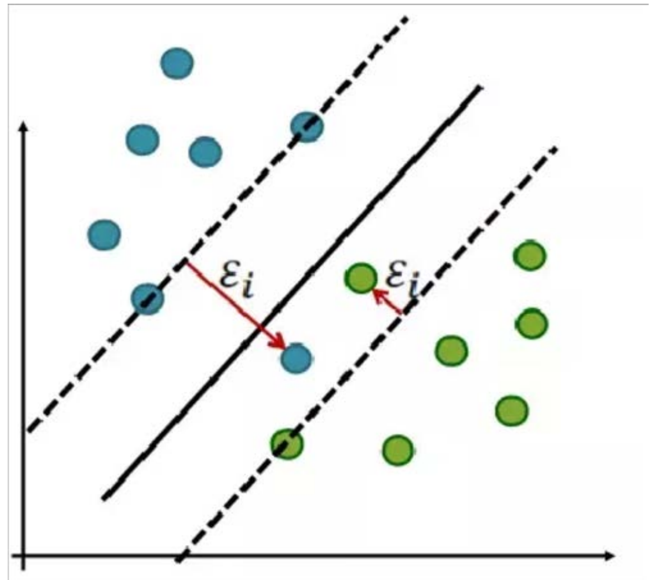


So… When/How can we use these in our visualizations?









Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

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

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

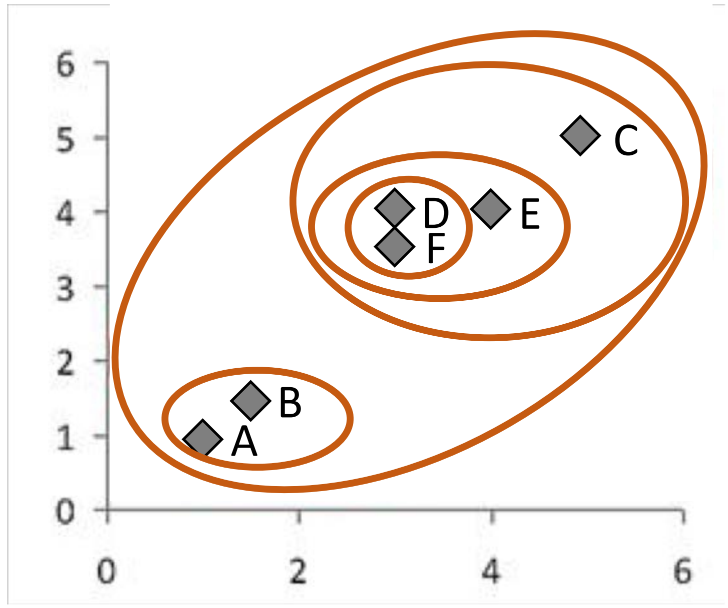
Gaussian Mixture Models, Hierarchical Clustering, K-means Clustering

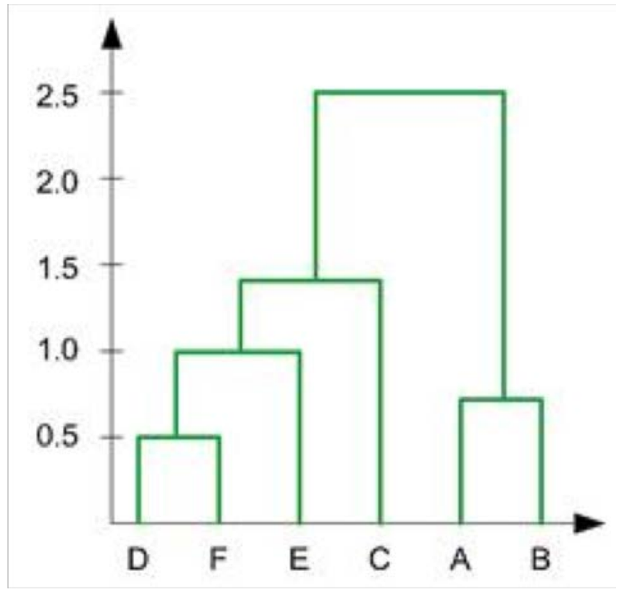
<u>FINDING CLUSTERS IN DATA</u>

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

<u>HIERARCHICAL CLUSTERING</u>

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

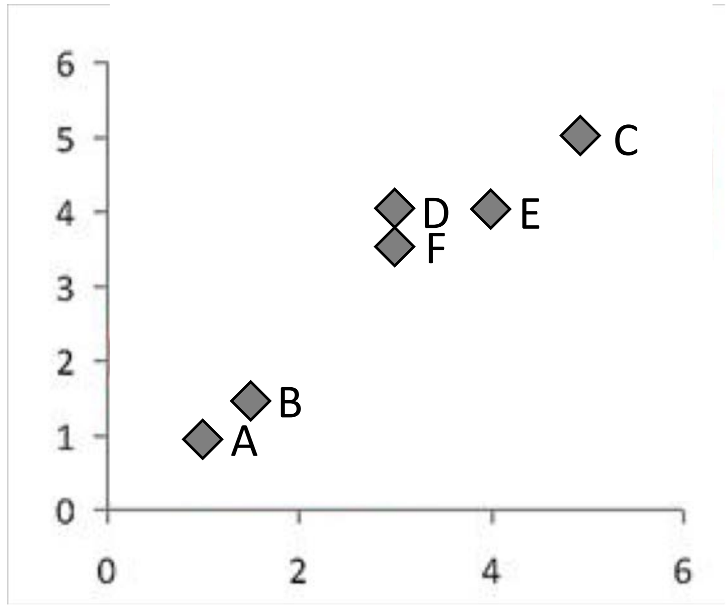


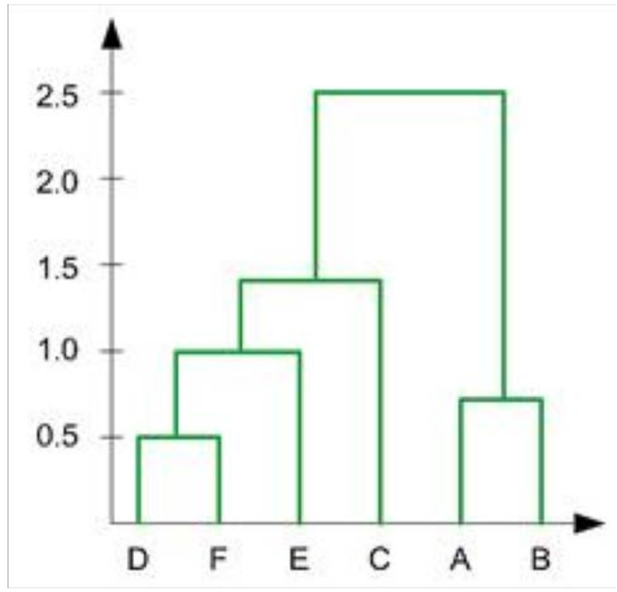


HIERARCHICAL CLUSTERING

Given a pairwise distance matrix

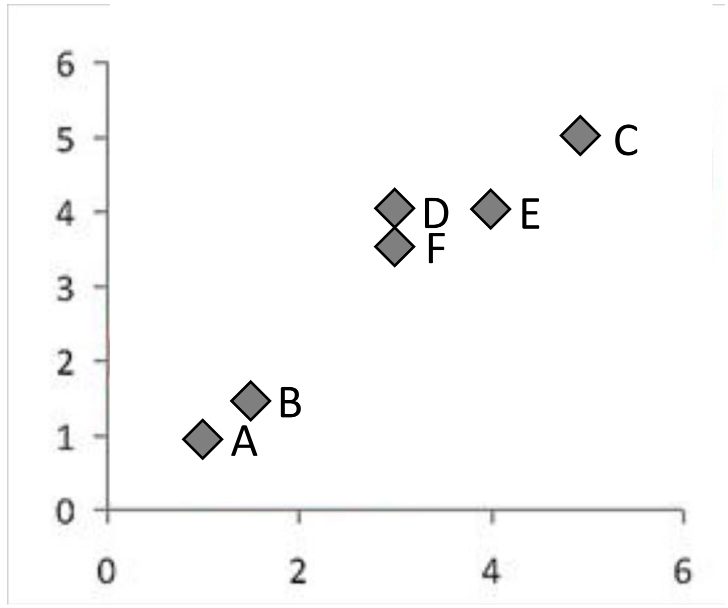
Sort the pairwise distances from smallest

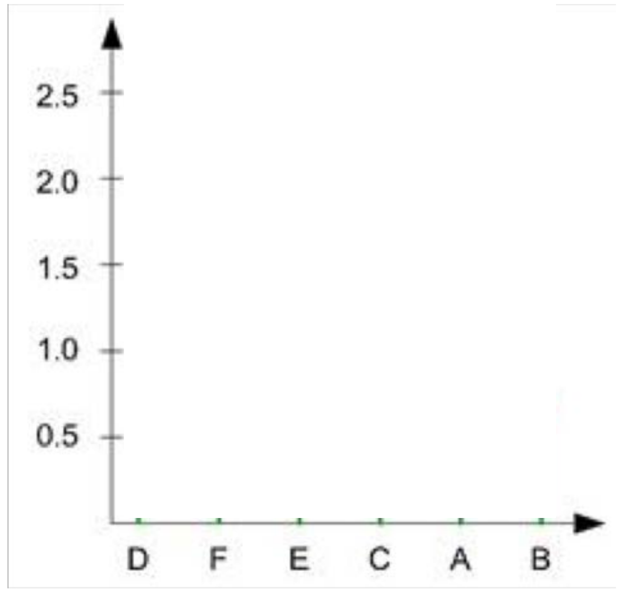




Hierarchical Clustering

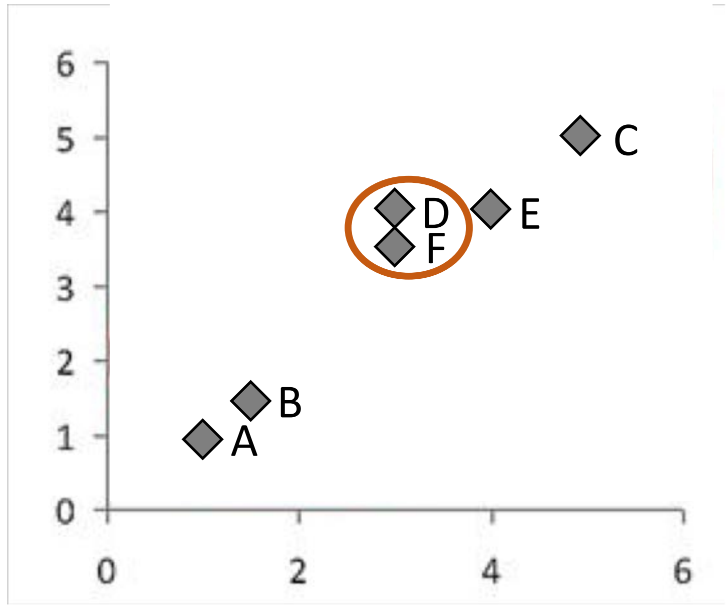
Start with each point as its own cluster

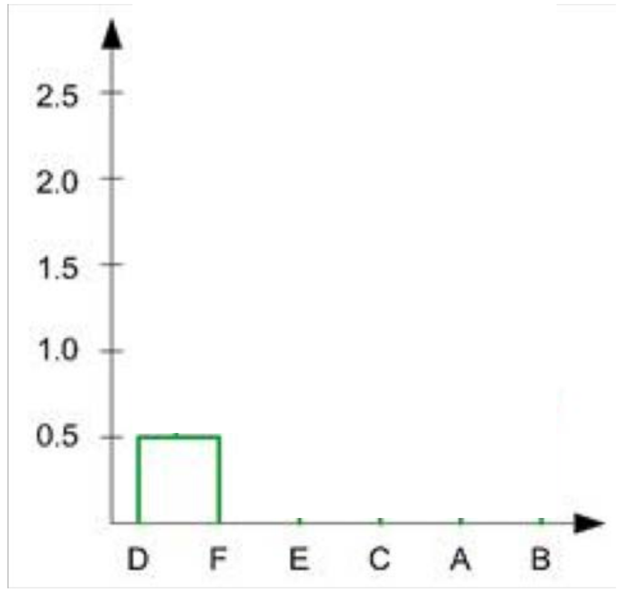




HIERARCHICAL CLUSTERING

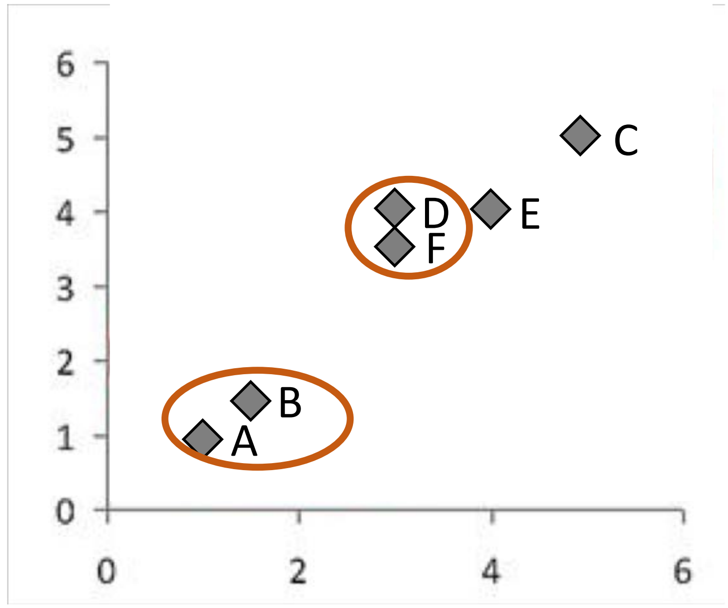
Step through sorted distances joining clusters

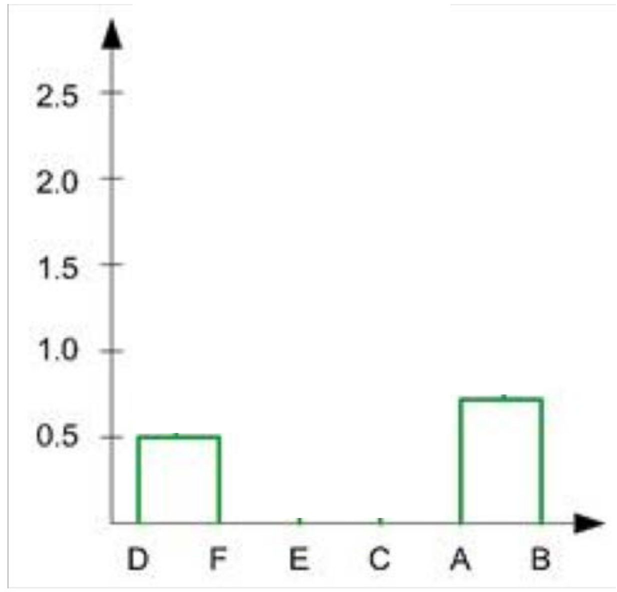




HIERARCHICAL CLUSTERING

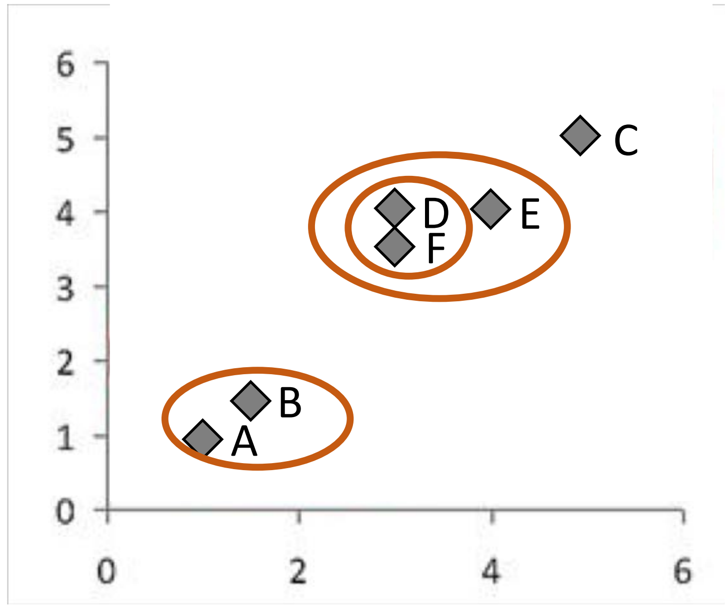
Step through sorted distances joining clusters

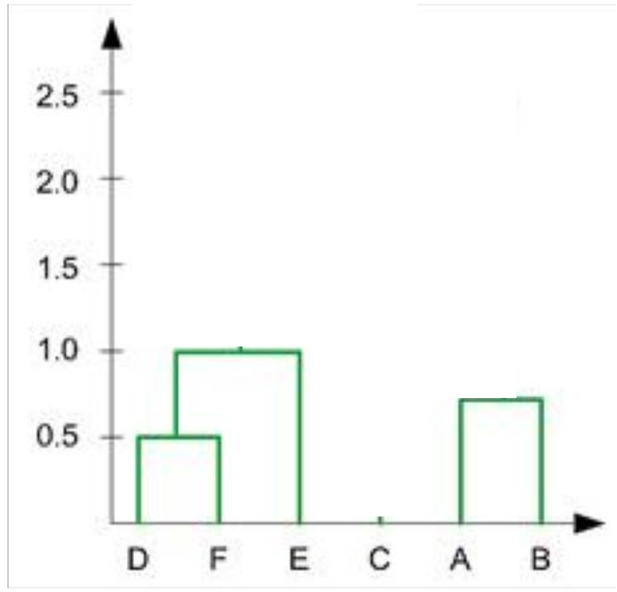




HIERARCHICAL CLUSTERING

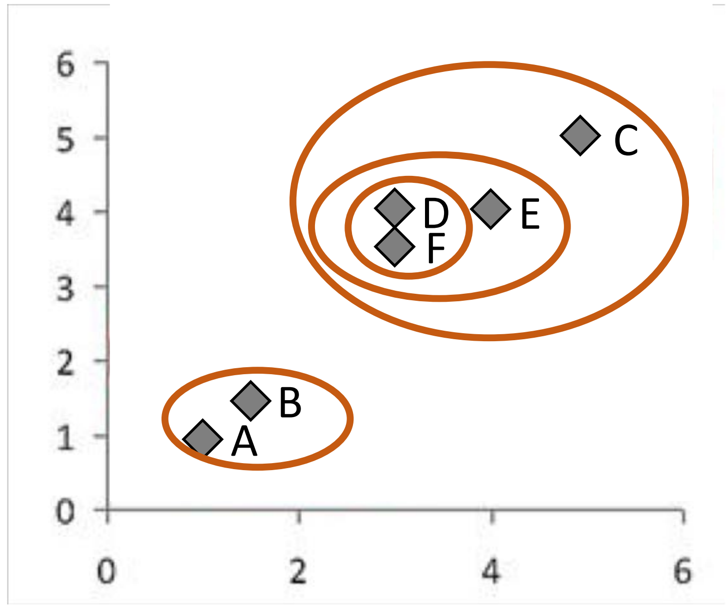
Step through sorted distances joining clusters

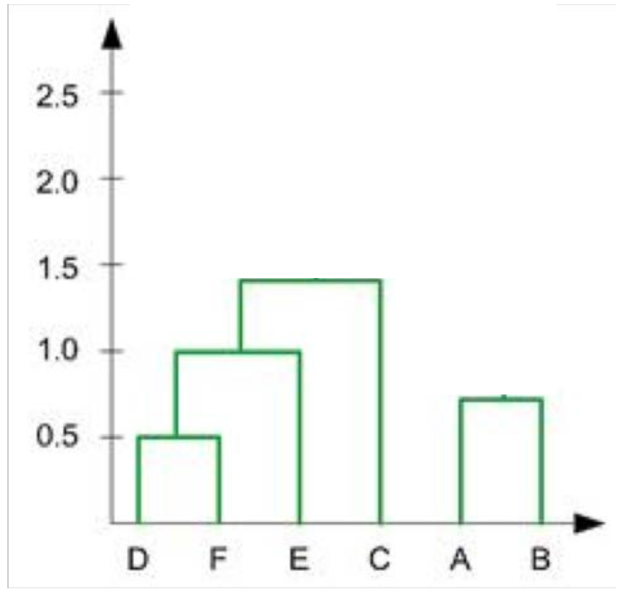




<u>HIERARCHICAL CLUSTERING</u>

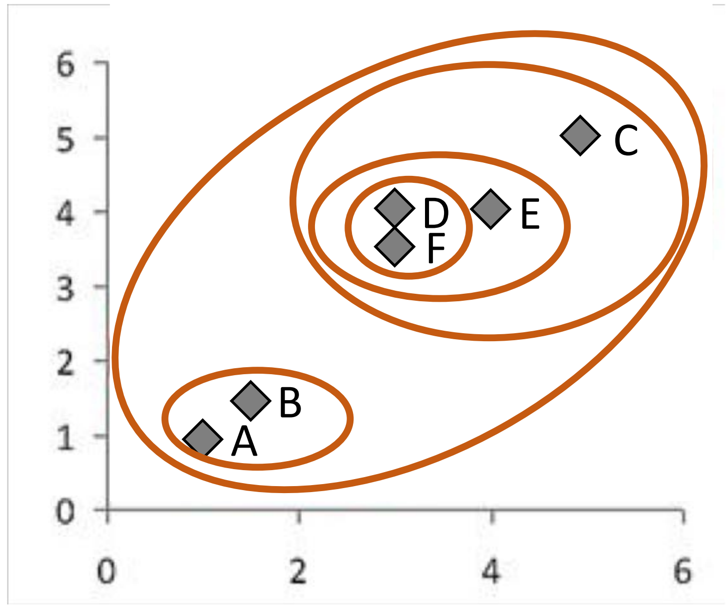
Step through sorted distances joining clusters

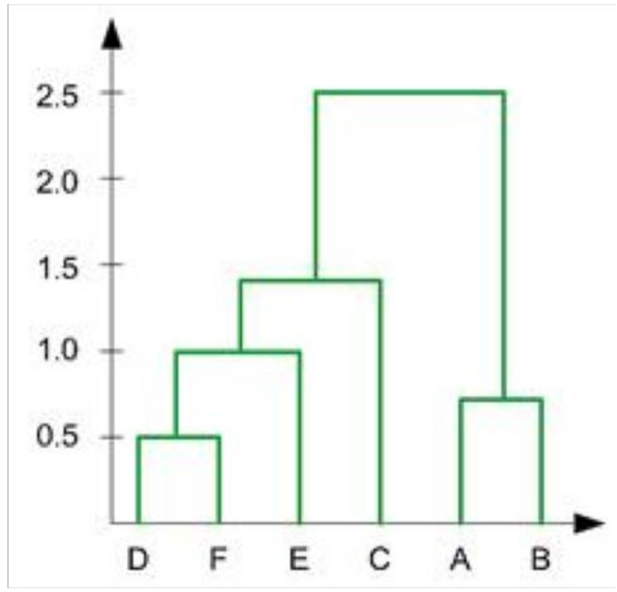




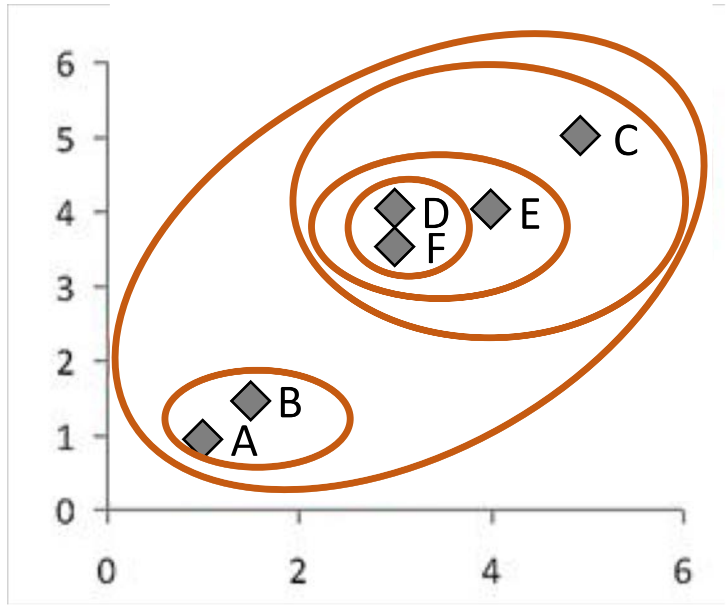
HIERARCHICAL CLUSTERING

Step through sorted distances joining clusters

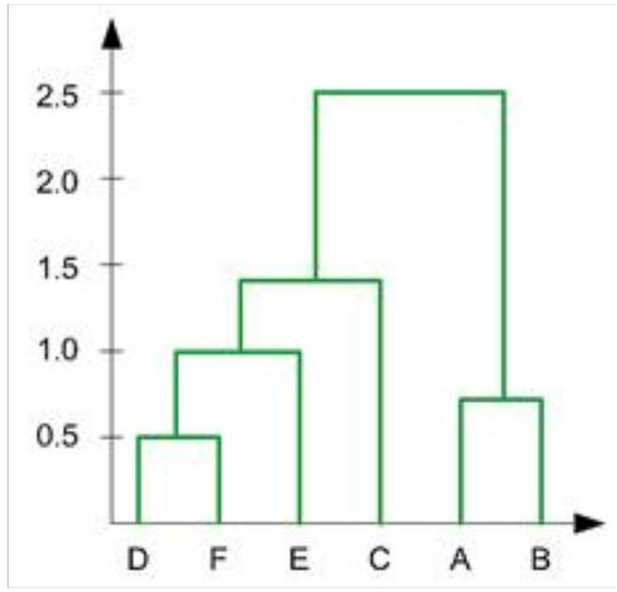




HIERARCHICAL CLUSTERING

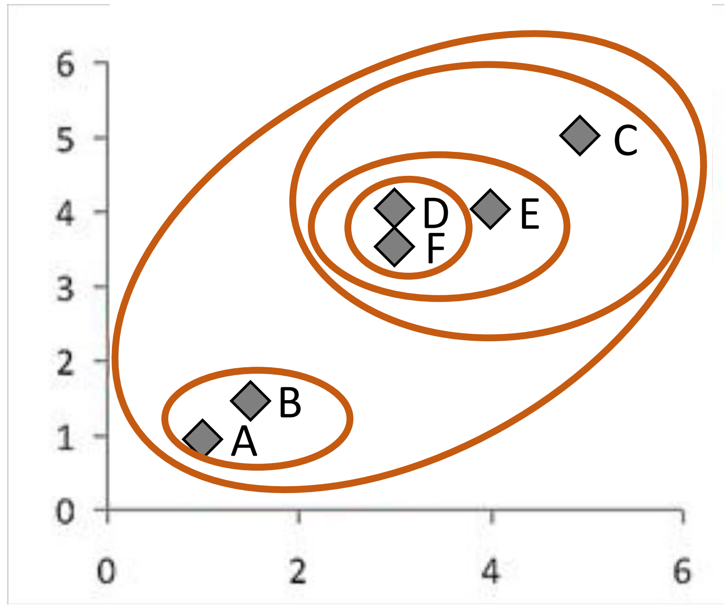


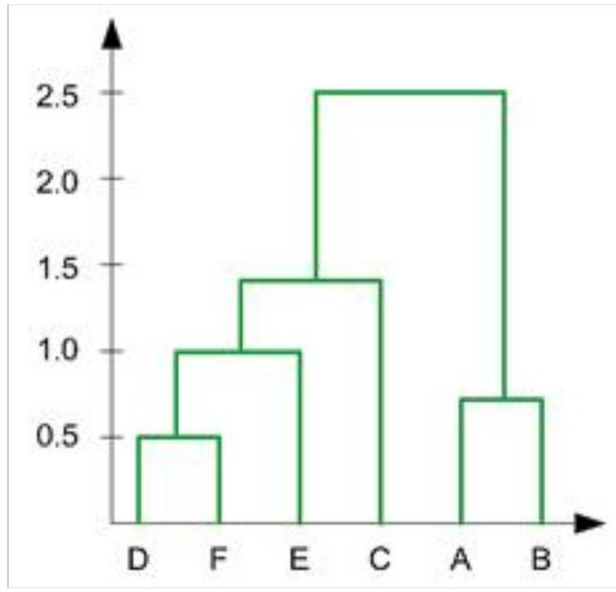
Advantages: easy to implement, works on metric space data, tells you all possible clustering events



Disadvantages: ?

HIERARCHICAL CLUSTERING



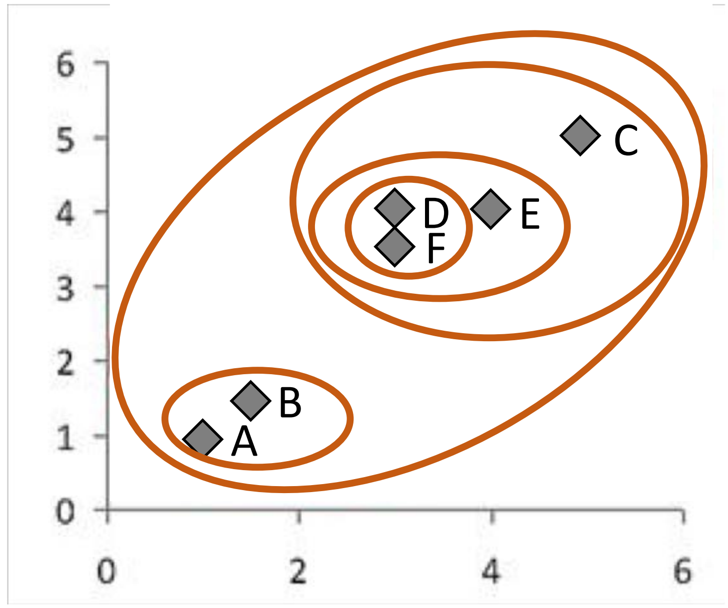


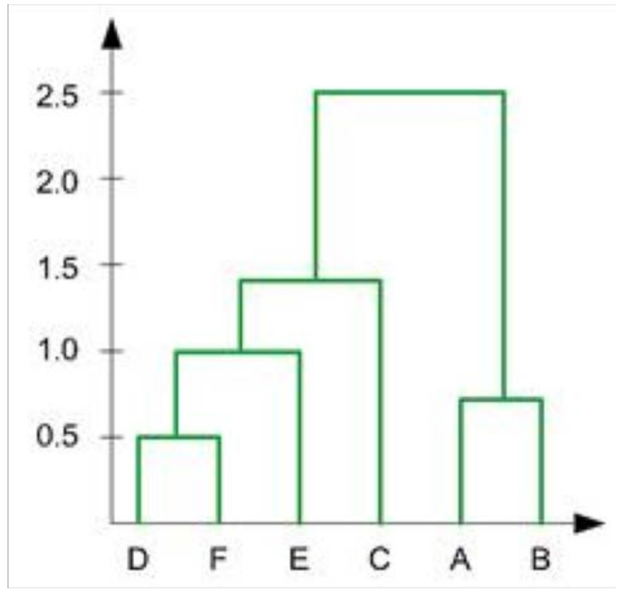
Advantages: ?

Hierarchical Clustering

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

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

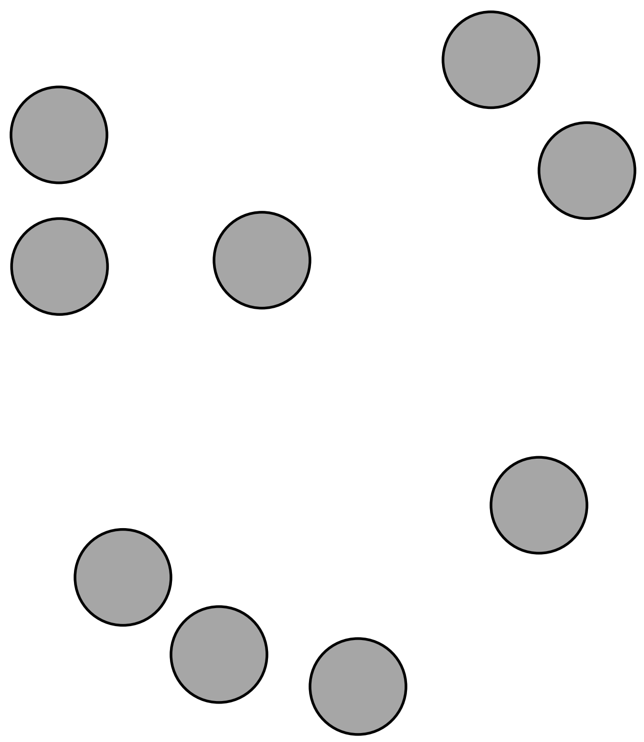




K-means Clustering

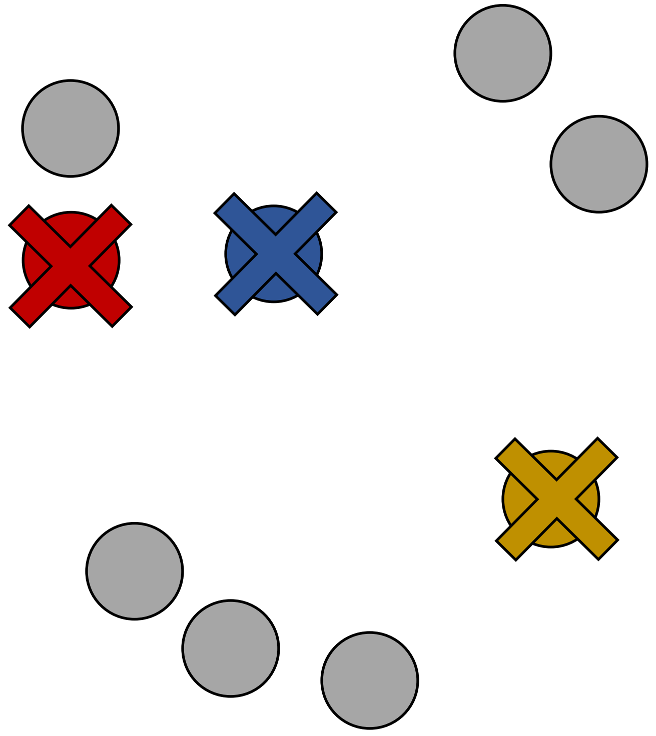
Hierarchical clustering is expensive

For "big data" we prefer something faster (approximate)



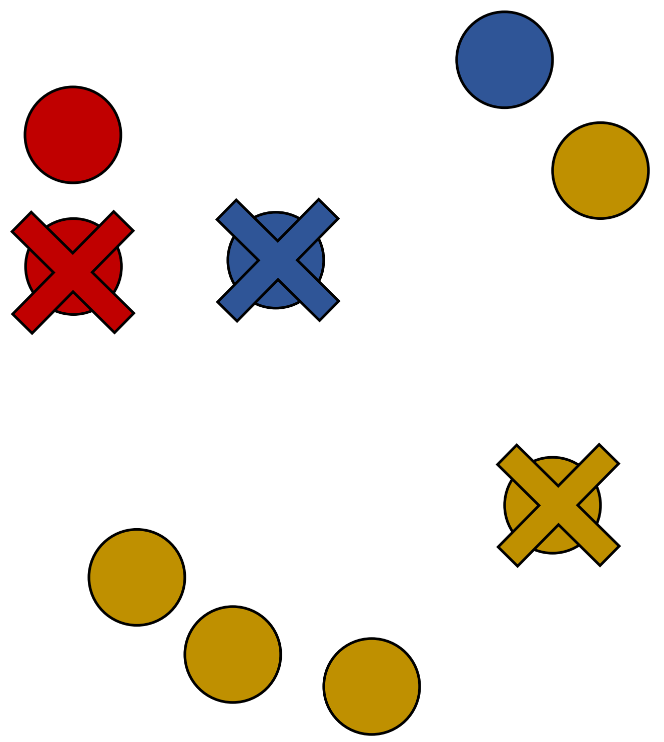
K-MEANS CLUSTERING

Select k points at random as initial cluster means



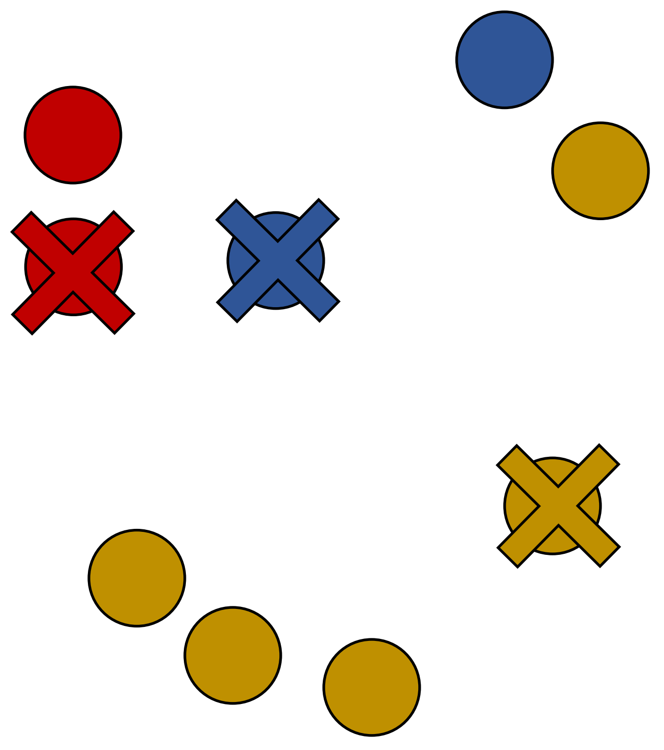
K-MEANS CLUSTERING

Points "join" the cluster they are closest to



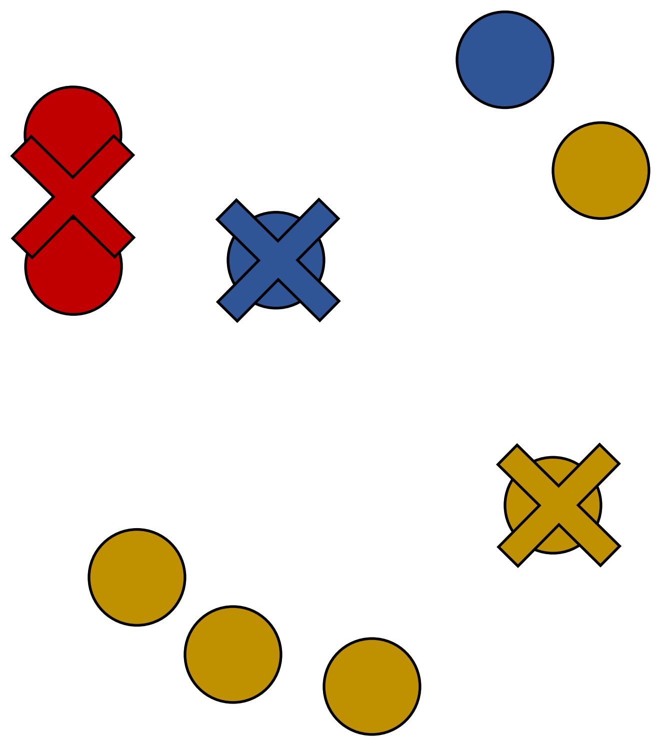
K-MEANS CLUSTERING

New means are calculated based upon cluster elements



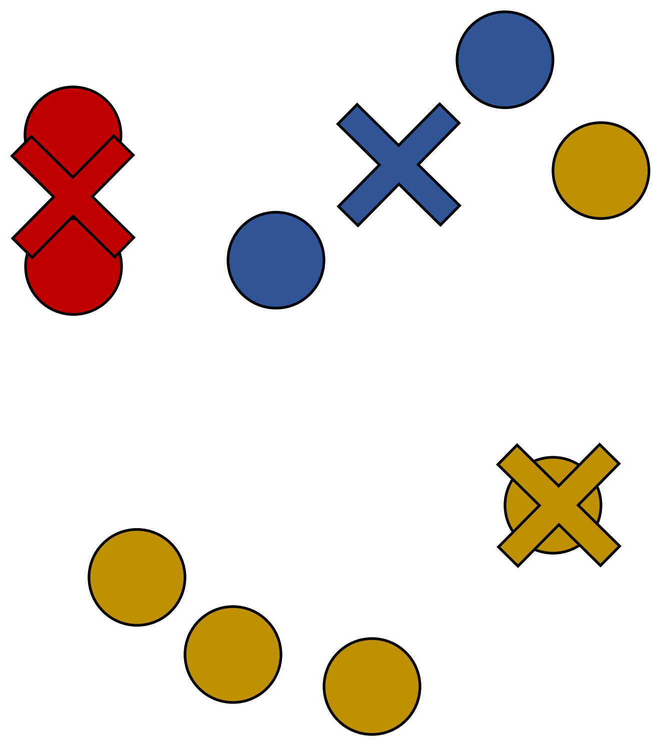
K-means Clustering

New means are calculated based upon cluster elements



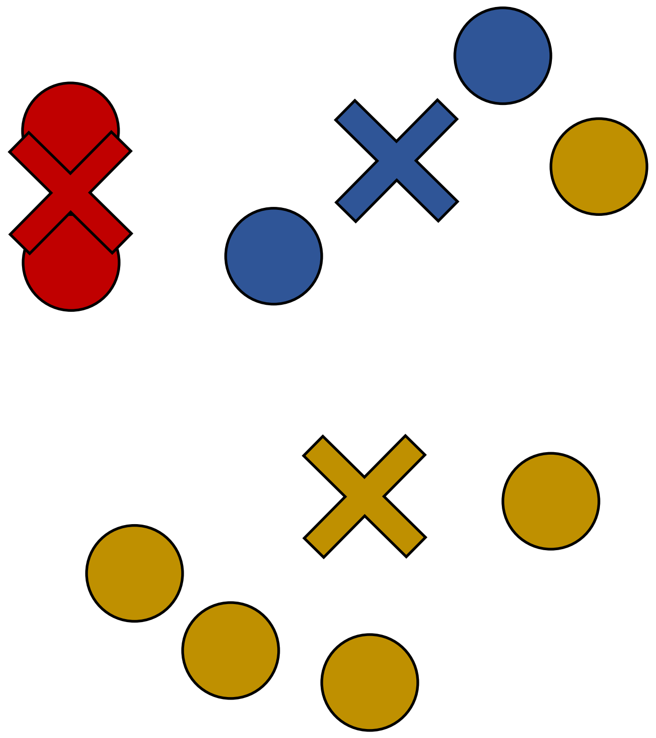
K-means Clustering

New means are calculated based upon cluster elements



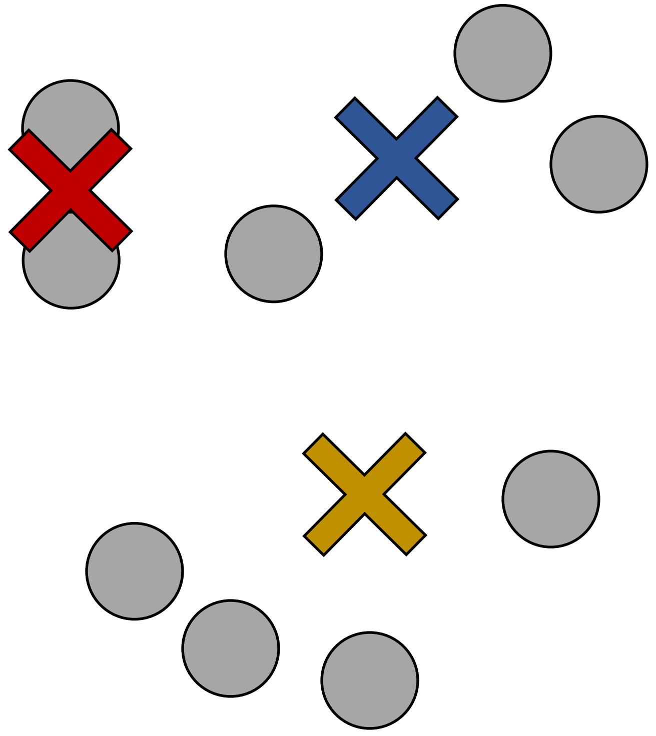
K-MEANS CLUSTERING

New means are calculated based upon cluster elements



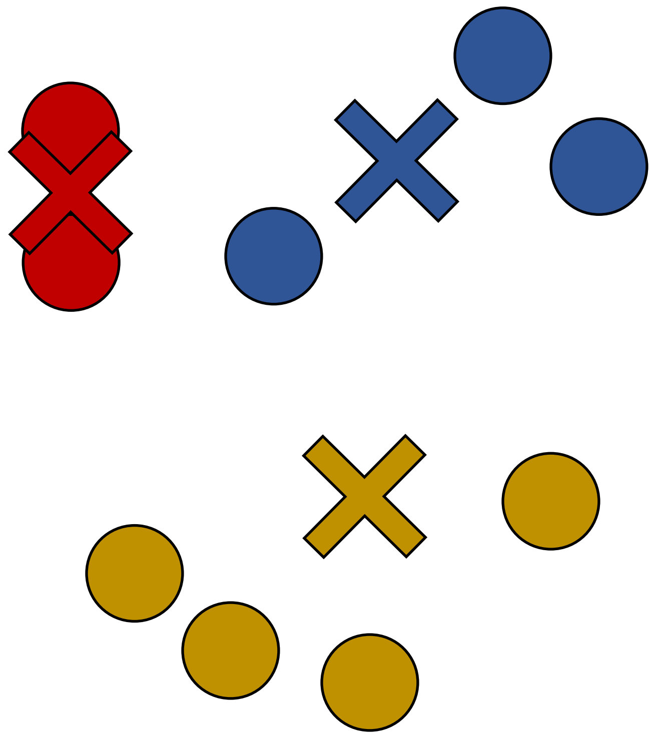
K-means Clustering

Recluster points



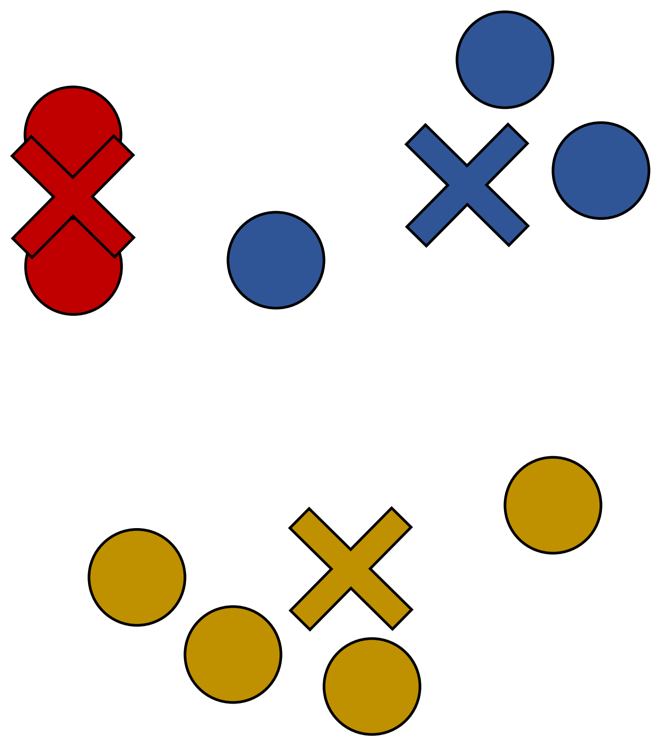
K-means Clustering

Recluster points



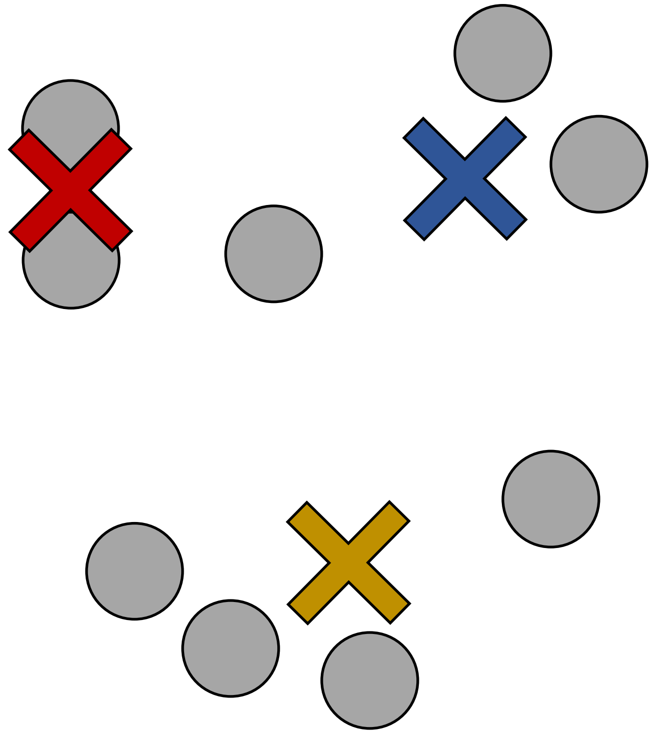
K-means Clustering

Recalculate means



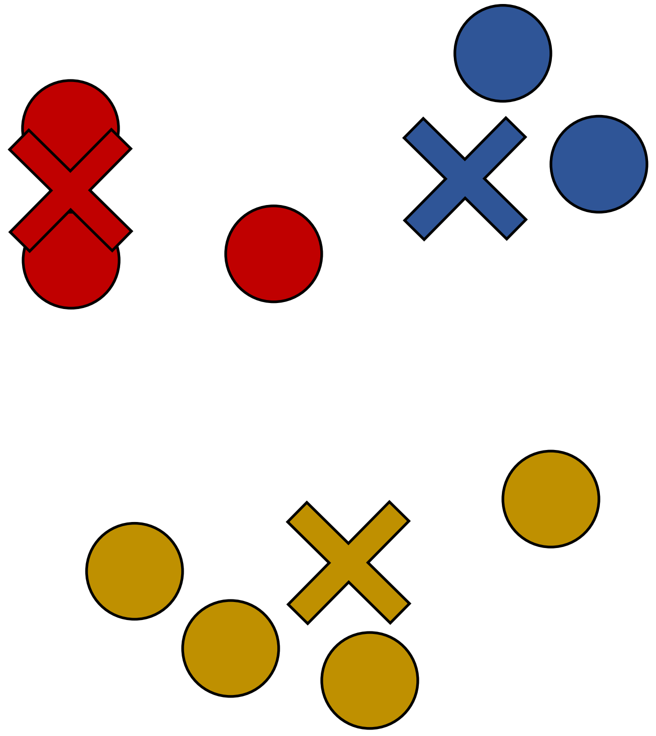
K-MEANS CLUSTERING

Recluster



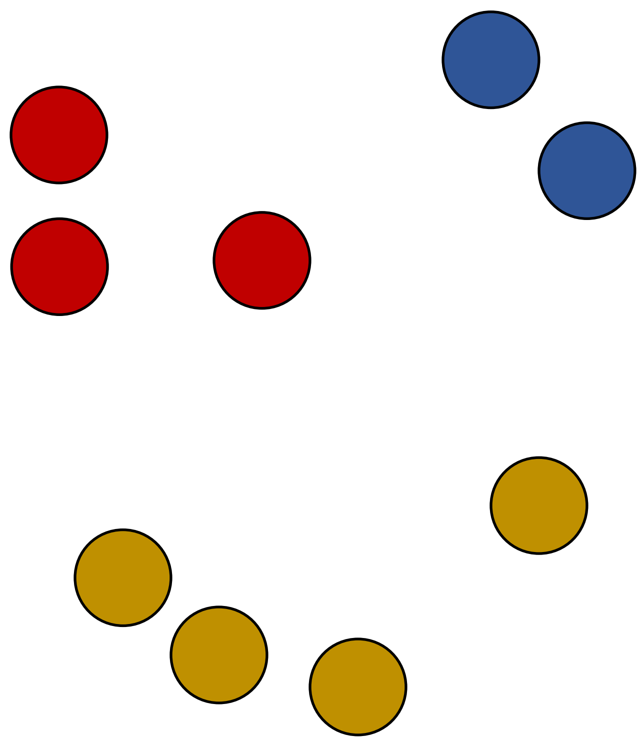
K-MEANS CLUSTERING

Recluster



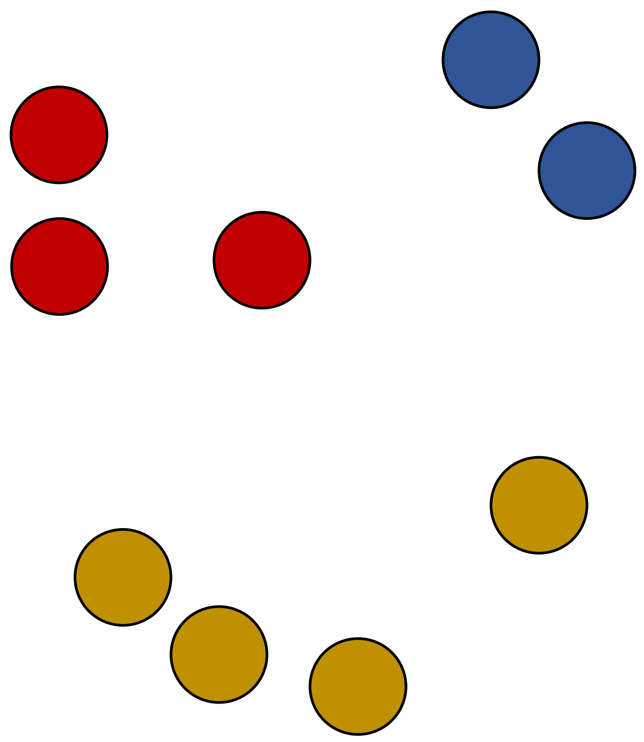
K-means Clustering

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

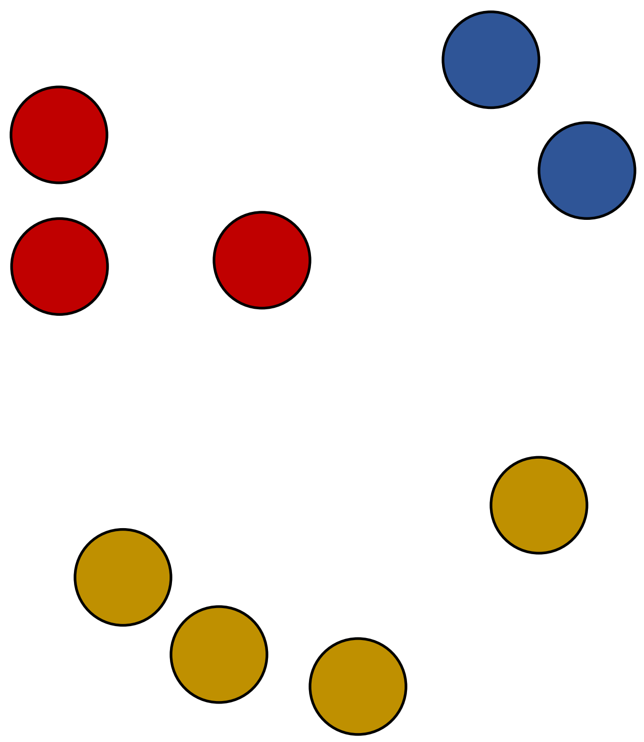
Advantages: ?



K-means Clustering

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

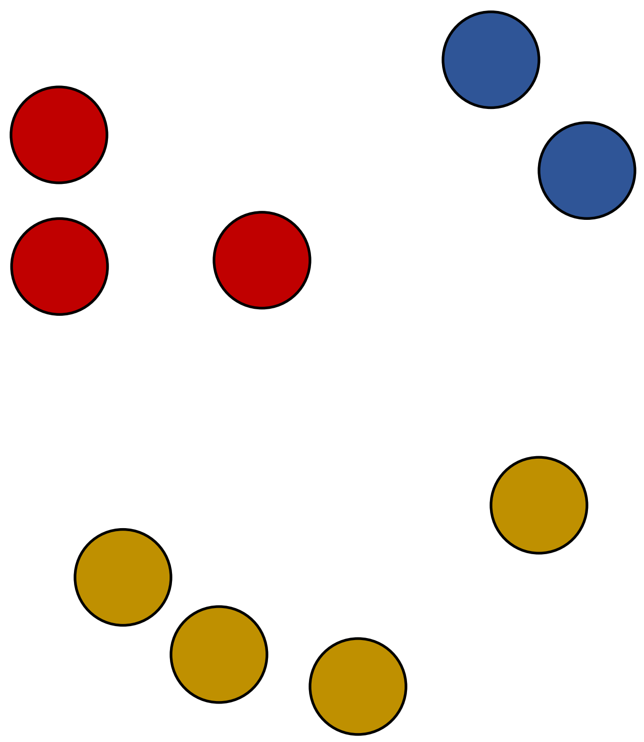
Disadvantages: ?



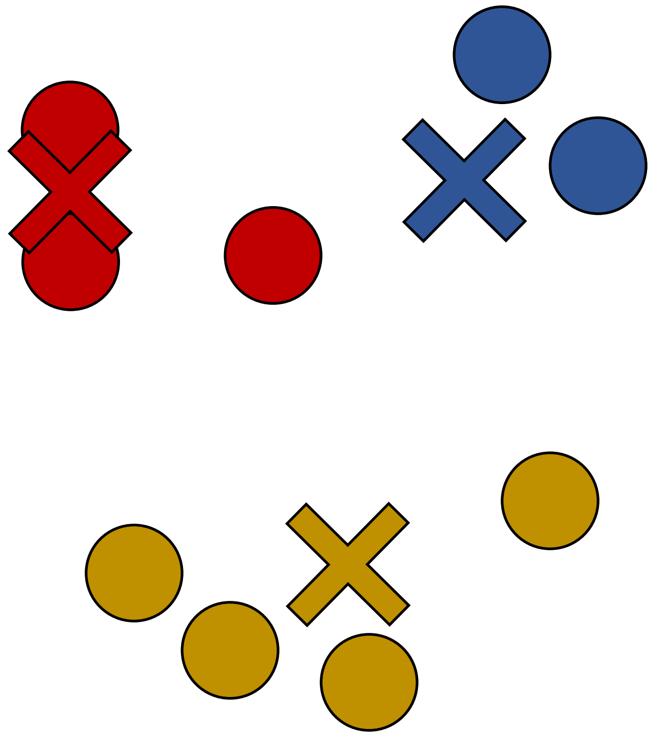
K-MEANS CLUSTERING

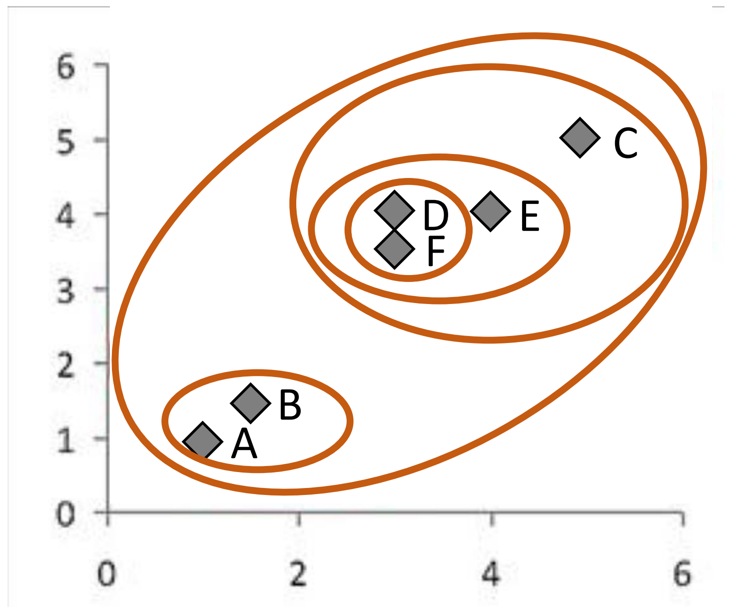
Advantages: Finds good clusters in many datasets, each iteration is only $O(kn)$

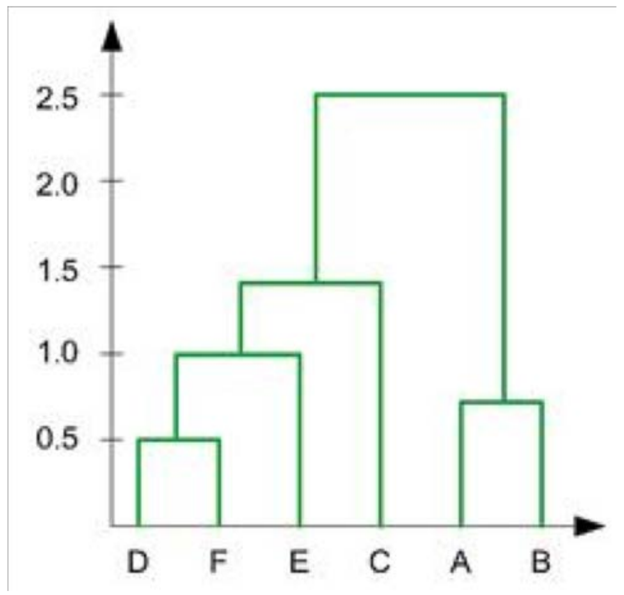
Disadvantages: How to pick k, when is it good enough to stop, initial point selection can change clusters



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.

Upcoming Classes

Statistics + Visualization

Computational Topology as a data mining tool