COMP 4432 Machine Learning

Live session 2: Big picture

Yuanyuan Li
yuanyuan.li456@du.edu
09/21/2022

Agenda

- Week 2 schedule
- Machine learning big picture
- Quality control (QC)
- Demo

Week 2 schedule

- Lesson 2: End to end ML project
- Reading Geron chapter 2: Regressors vs. classifiers
- Assignment 1 due
- Start working on assignment 2

Datasets

- Some are built in to sklearn
- GitHub - arjayit/cs4432_data

Demos from async

- GitHub - arjayit/cs4432_demos: demos for course
- Dr. Harmon's demos and mine will be posted in **Files**

Assignment 2 overview

- **Assignment 2 (due by midnight MST the day prior to Live Session 4)**

- **Assignment 2, Part 1: Data Exploration.** Load the scikit-learn diabetes bunch object into a variable. Print out the description of the dataset. Load the diabetes features into a pandas dataframe with the proper column names. Add the target variable to this same dataframe. Run a command to look at the data types of your dataframe to see if there is any missing data. Perform descriptive statistics on the numeric columns of your dataframe. Plot histograms of your data to get a feel for each column's distribution. Split your dataframe into a training and test set with 20% of your data being in the test set. Define a correlation matrix. Look at values highly correlated with the target. Plot the correlation matrix with a Seaborn heatmap. Use a Seaborn pairplot to look at the scatter plots of the three values with the highest target correlation. Prepare a feature set by dropping the target from your training dataframe. Copy your training target into a new dataframe.

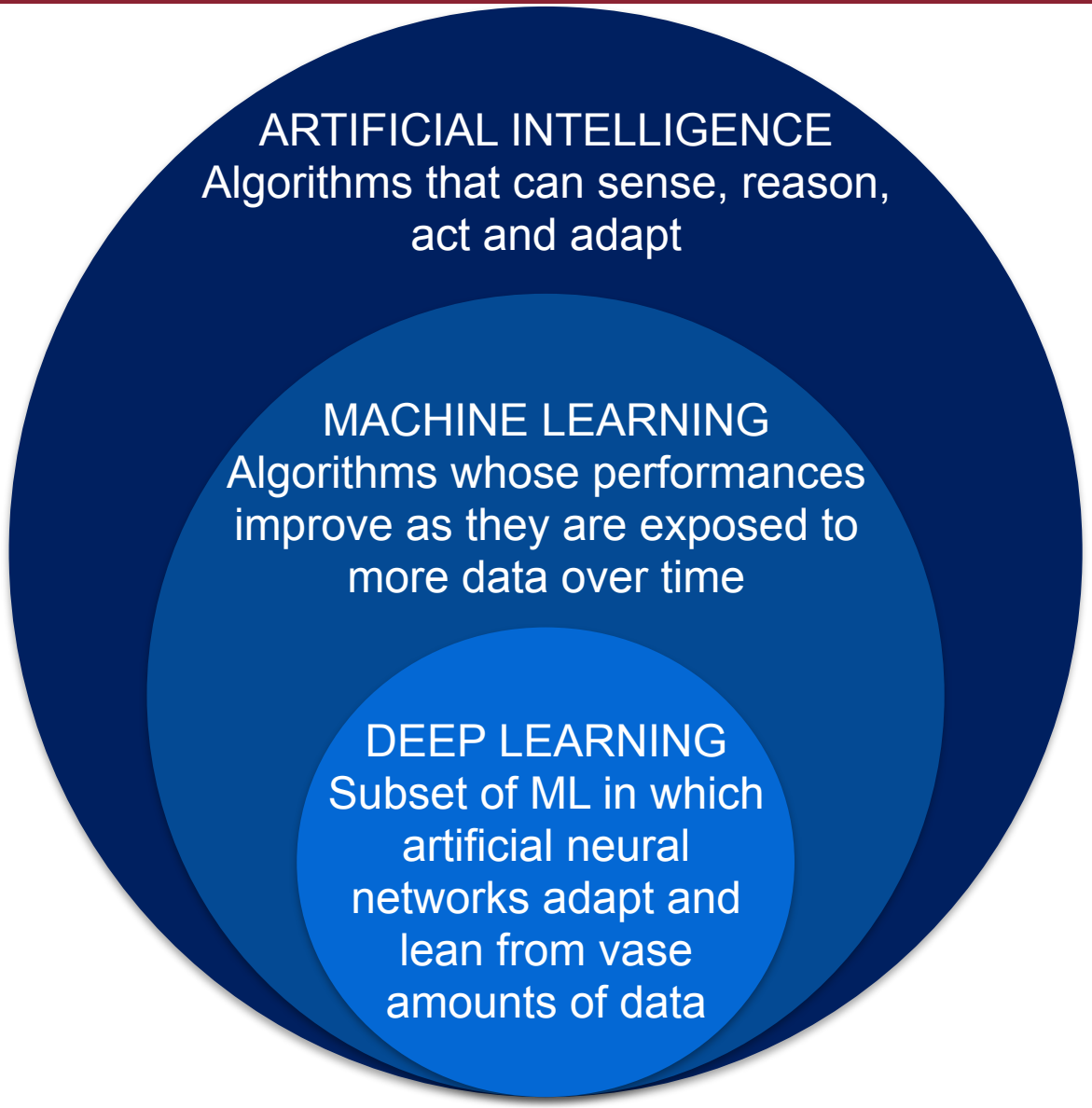
Assignment 2 overview

- **Assignment 2, Part 2: Model Training.** Train a linear regression model using your training set. Print the RMSE of your regression model on your training set. Implement a cross_val_score on a decision tree regressor on your training set. Print out root mean and standard deviation of the cross-validation scores. Do the same for a RandomForestRegressor. Record which model performs better.

- **Assignment 2, Part 3: Model Tuning.** Print out the parameters of your random forest model. Do a grid search cross-validation with the following values: n_estimators: 3,10,30 and max_features: 2,4,6,8, as well as the following experiment: bootstrap: False, n_estimators: 3,10 and max_features: 2,3,4. Print out the best parameters and the best performing model based on this grid search. Using the cv_results dictionary, print out the rmse of each feature combination for comparison. Also print out the feature importances of the best performing grid search model. Describe how it compares with the correlation matrix we implemented earlier.

- **Assignment 2, Part 4: Model Evaluation.** Document the best-performing model between the single feature model you trained in Assignment 1, and the models you trained in part 2 and 3 of this assignment. Evaluate the best performing model against your test set. Save your model for future use.

Introduction to machine learning



Machine learning emphasizes high dimensional prediction problems. [Wasserman, 2012]

Figure modified from: https://www.codeproject.com/articles/1185501/How-to-Get-Started-as-a-Developer-in-AI

Types of machine learning

- Features engineering
  - Feature selection
  - Feature extraction

- Unsupervised: outcome is unknown
  - Clustering
  - Kernel density estimation

- Supervised: outcome is known
  - Classification
  - Regression

Feature Engineering

Feature selection

- Select a subset of feature from the orginal feature set
- Suitable for high dimensional data
  - Image processing
  - Text processing
  - Biomedical data
- Find the optimal subset is NP-hard
- Main methods
  - Filtering
  - Wrap around a classifier/regressor
  - Embedded within a classifier/regressor

Filter feature selection method

- Example method
  - Statistical testing (T-test, F-test, ANOVA)
  - Correlation (Pearson, Spearsman)
  - Information gain
  - Variance threshold
- Is independent of the machine learning model
- Is generally univariate or low variate
- Is simple and fast
- Scales well to high dimensional data

Filter: statistical hypothesis testing

- The null hypothesis is that a given feature is not different between the two classes (normal vs disease).
- The alternative hypothesis is that the feature is different between the two classes.
- The hypothesis testing is performed by calculating a statistic (eg, the t-statistic) on the values of the feature of interest measured in the two classes.
- The computed value of the statistic is then compared with a threshold, calculated from a model (eg, the t-distribution) & a desired significance level (eg, 5% or 1%).

Filter: statistical hypothesis testing

Table of error types		Null hypothesis (H_0) is	
		True	**False**
Decision About Null Hypothesis (H_0)	**Reject**	Type I error (False Positive)	Correct inference (True Positive)
	Accept (not rejected)	Correct inference (True Negative)	Type II error (False Negative)

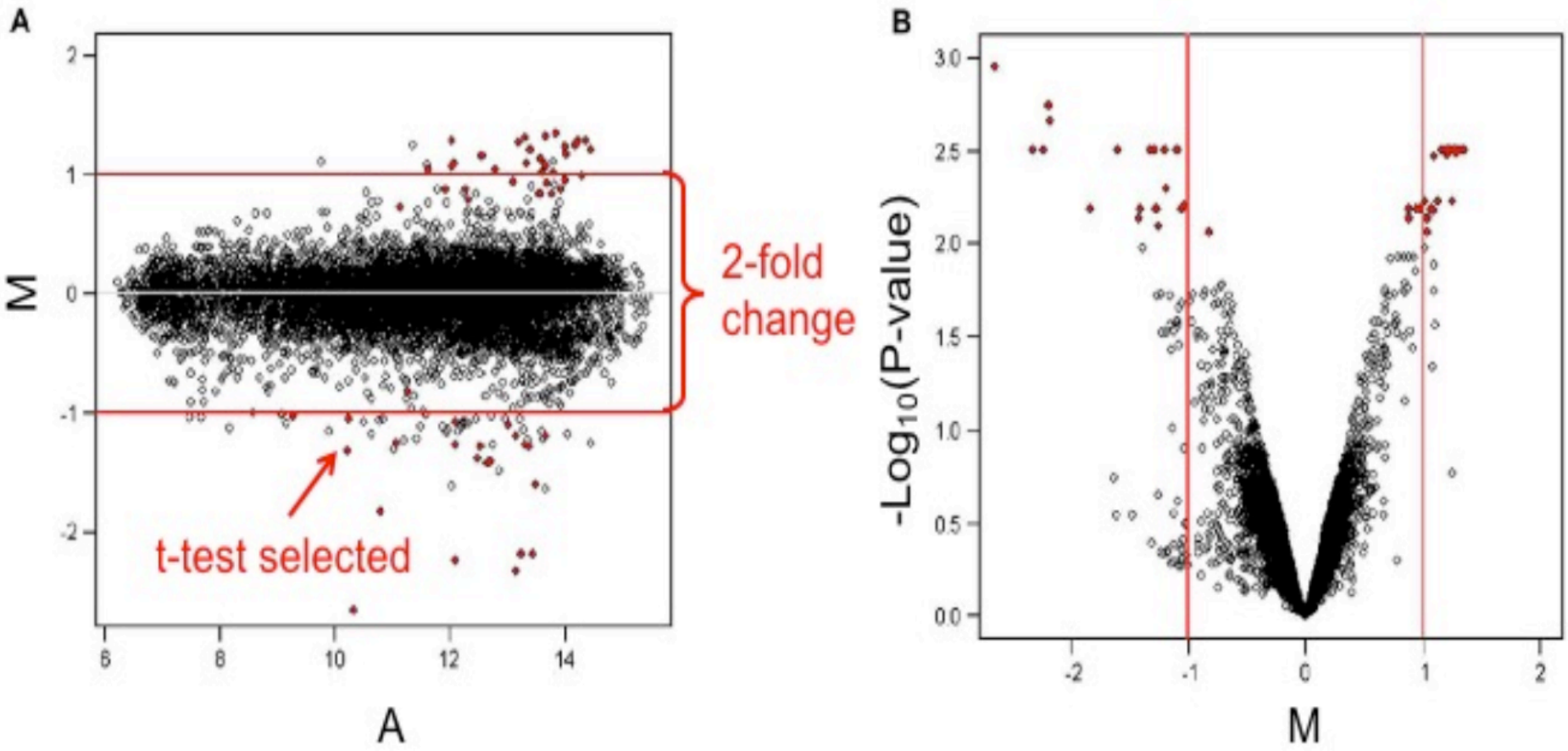
- Significance level chosen before the test, and represents % of Type I error that we are prepared to accept.
  - Eg, significance level at 1% means that, on average, there will be one false positive gene for every 100 genes identified as differentially expressed.
- The statistical power of a technique is a measure of its ability to identify true positives.

Filter: statistical hypothesis testing

- Formulate a null & alternative hypothesis for every feature
- T-test: difference in means between 2 classes divided by the standard deviation
  - compares the difference in the mean values between 2 classes
  - taking into account the variability of the data
- F-test: between-group variability divided by within-group variability considering a decomposition of the variability in a collection of data in terms of sums of squares (SS)
- These SS are constructed so that the statistic tends to be greater when the null hypothesis is not true.
- ANOVA F-test: used when comparing > 2 groups

Adjust for multiple testing

T-test for gene selection application



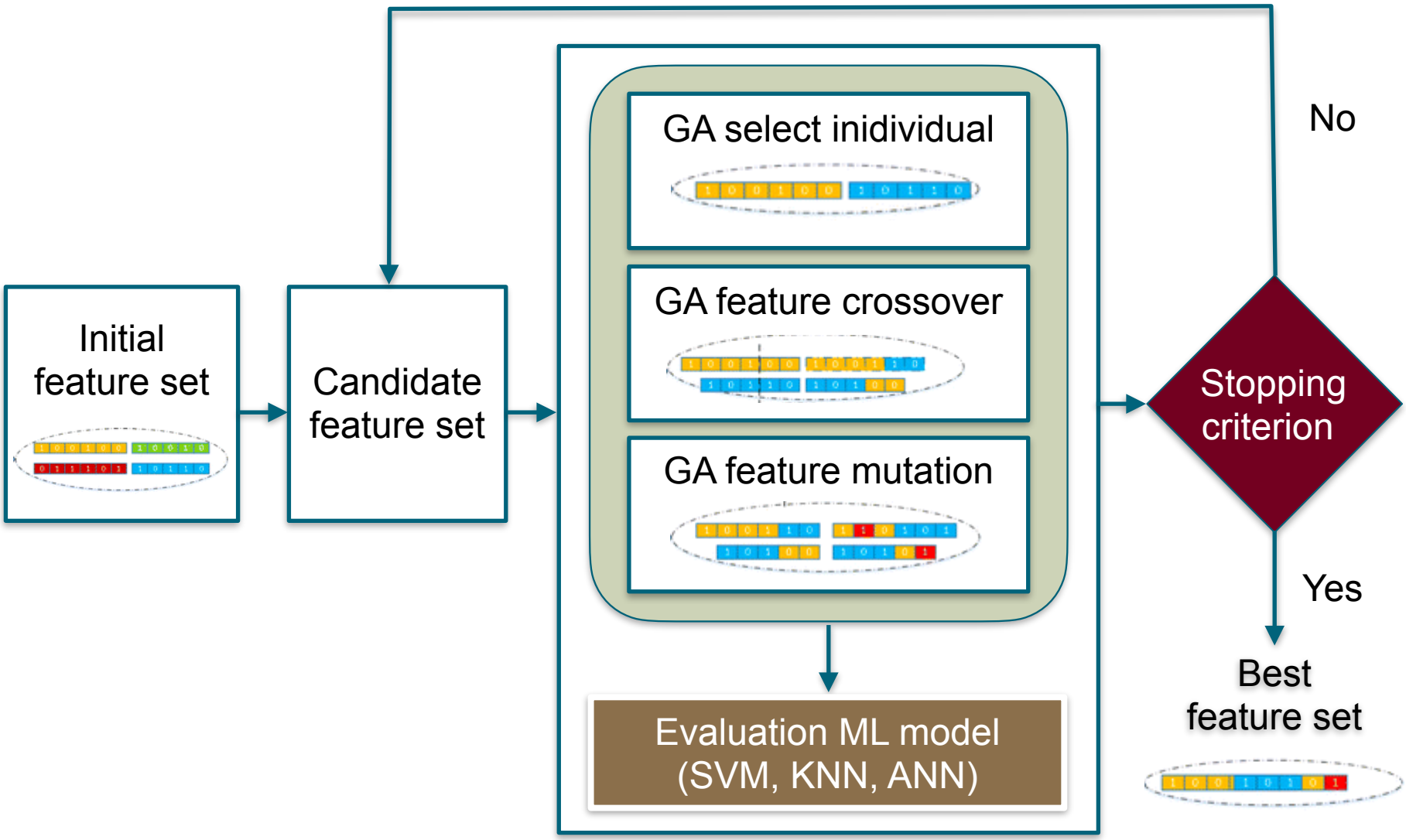
MA plot: the difference of the log-intensity of the two channels (log-ratio) against the average log-intensities (mean average).

Source: Cowley & Ying, 2011

Wrapper feature selection method

- Example method
- Sequential Forward Selection (SFS)
  - Genetic Algorithm (GA)
- Combines feature selection algorithm with a classifier/ regressor to evaluate selected feature(s)
  - First, selected subset of feature(s), then pass down to a machine learning model for evaluation
- Is a search-based algorithm
  - Forward search (iteratively add features)
  - Backward search (iteratively remove features)
- Finding the optimal subset is NP-hard
- Selected features directly ties with model performances

Wrapper: Genetic algorithm (GA)

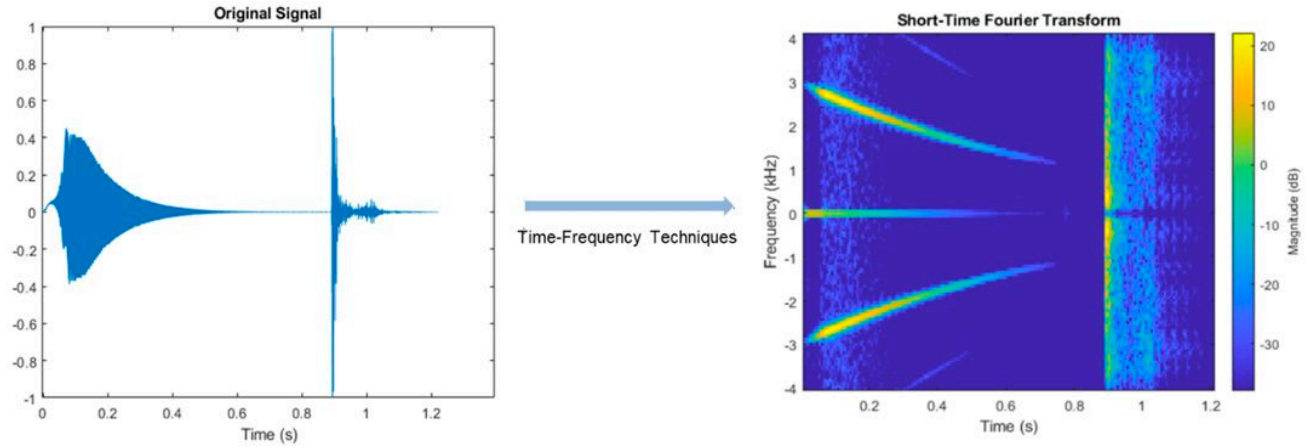


Embedded feature selection method

- Main methods
  - L1 (LASSO) regularization
  - Decision trees
- The machine learning model itself has feature selection procedure inside

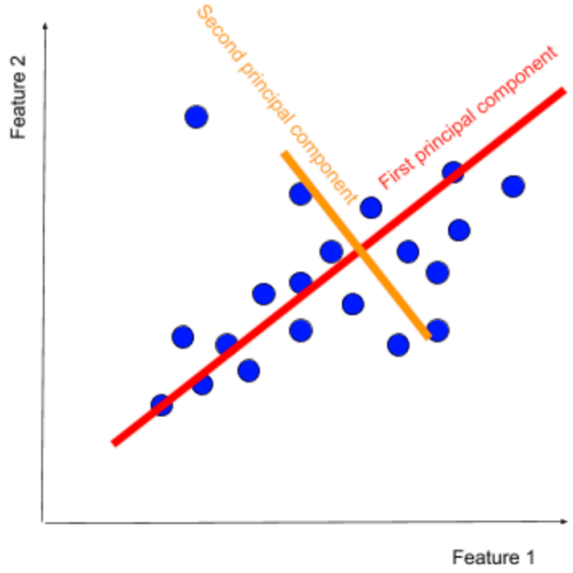
Feature extraction

- Generate new features from extisting set
- Add additional helpful information for prediction
  - Image processing
  - Audio/speech processing
  - Medical data like EEG, EKG, and EMG
- Principal component analysis (PCA)
- Independent component analysis (ICA)



Principal component analysis (PCA)

- The principal components are vectors.
- The original data can be represented as feature vectors.
- PCA allows us to go a step further and represent the data as linear combinations of principal components.
- Principal components remove noise by reducing a large number of features to just a couple of principal components.
  - Principal components are orthogonal projections of data onto lower-dimensional space.
- In theory, PCA produces the same number of principal components as there are features in the training dataset. In practice, picking just a few of the first componets sufficently approximates the original dataset.
- The result is a new set of features in the form of principal components, which have many practical applications.



Source: https://www.keboola.com/blog/pca-machine-learning

Principal component analysis (PCA)

1. Feature standardization (a mean of 0 and a variance of 1).
2. Obtain the covariance matrix computation.
3. Calculate the eigendecomposition of the covariance matrix. We calculate the eigenvectors (unit vectors) and their associated eigenvalues (scalars by which we multiply the eigenvector) of the covariance matrix.
4. Sort the eigenvectors from the highest eigenvalue to the lowest.
5. Select the number of principal components. Select the top N eigenvectors (based on their eigenvalues) to become the N principal components. The optimal number of principal components is both subjective and problem-dependent.

Source: https://www.keboola.com/blog/pca-machine-learning

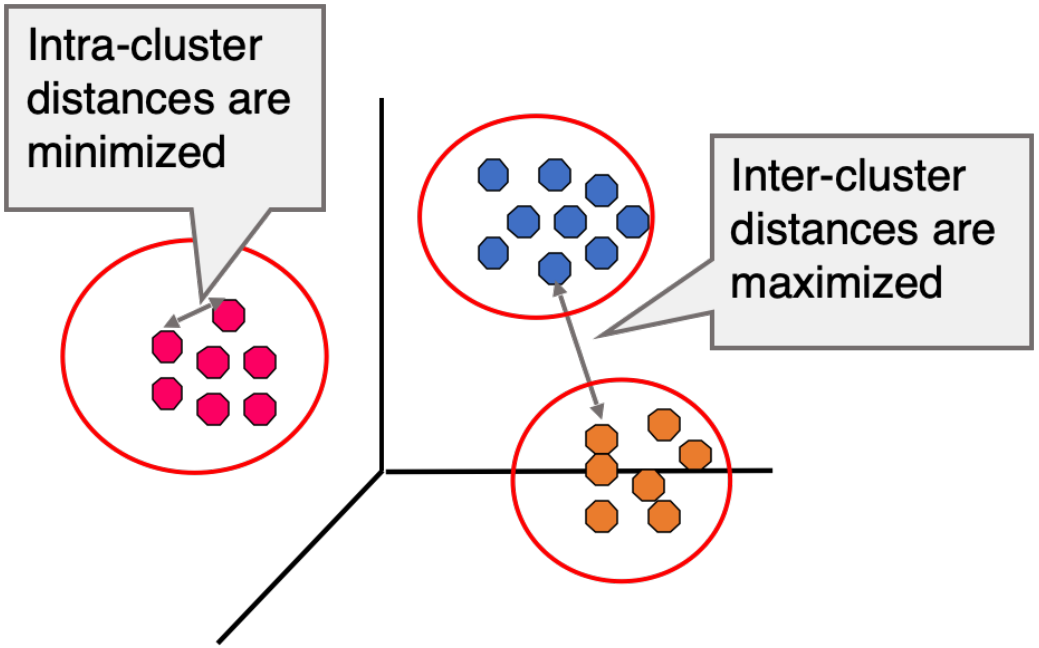
Principal component analysis (PCA)

- Assumes a correlation between features
- Is sensitive to the scale of the features
- Is not robust against outliers
- Assumes a linear relationship between features
- Cannot handle missing values

Unsupervised learning

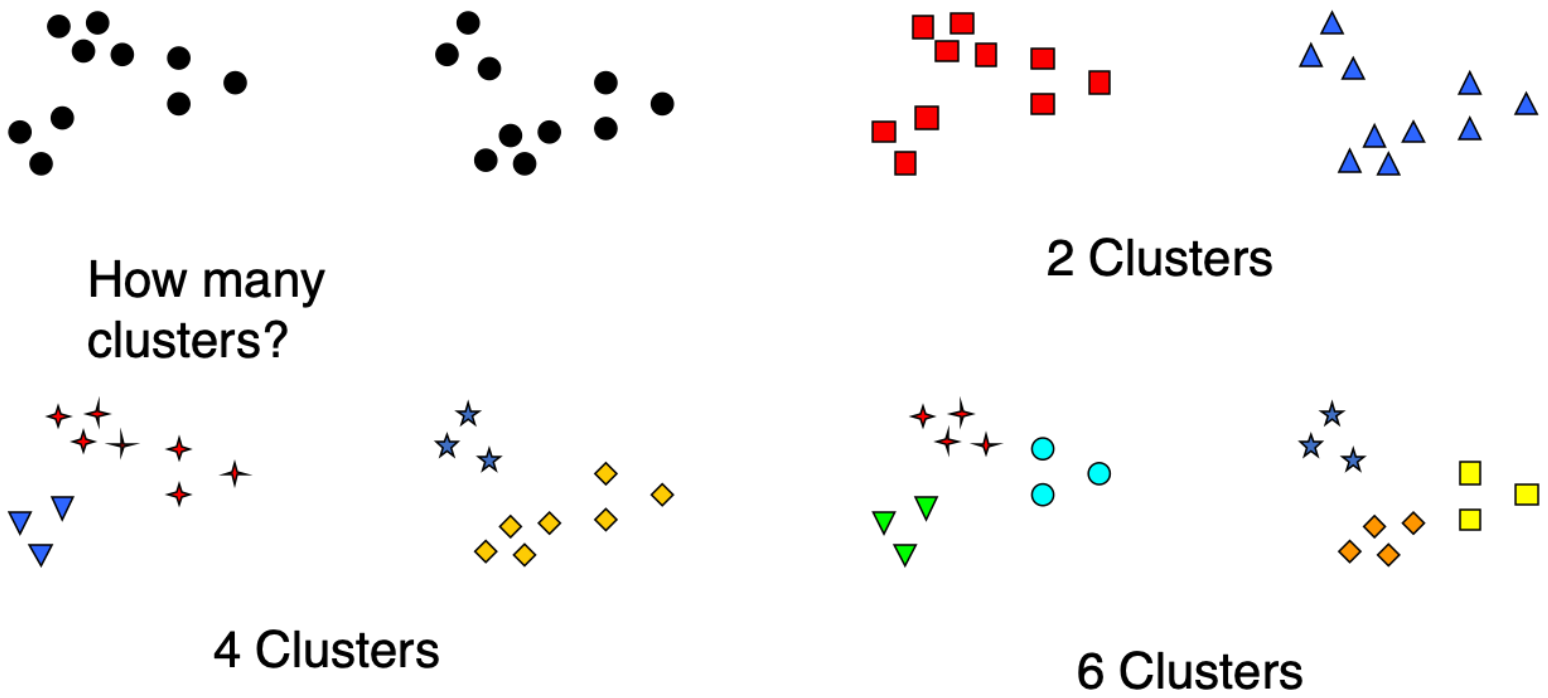
What is cluster analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another & different from (or unrelated to) the objects in other groups.



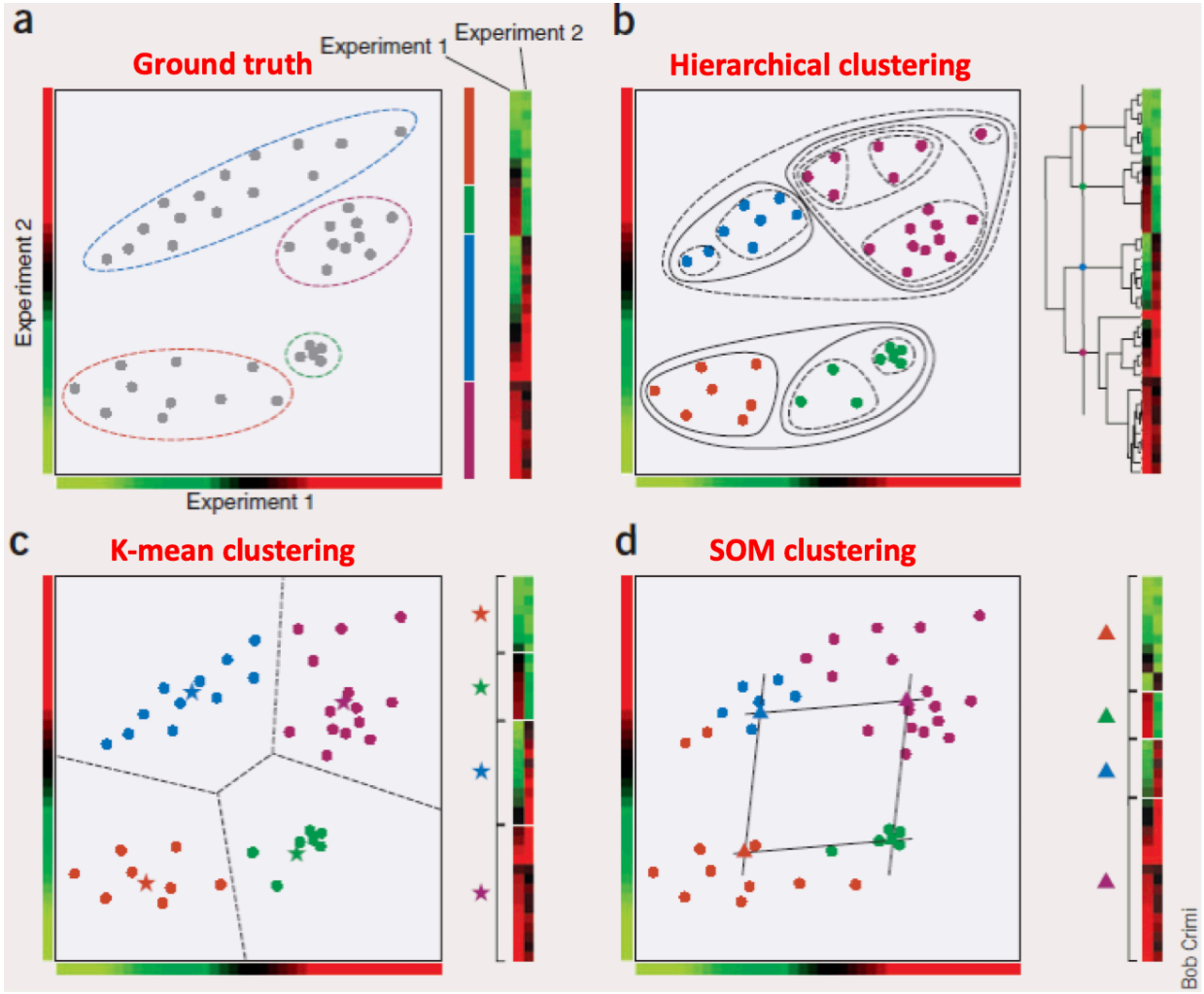
Source: Tan, Steinbach, & Kumar, 2004

Notion of a cluster can be ambiguous



Source: Tan, Steinbach, & Kumar, 2004

Example of clustering algorithms



Source: D'haeseleer, 2005

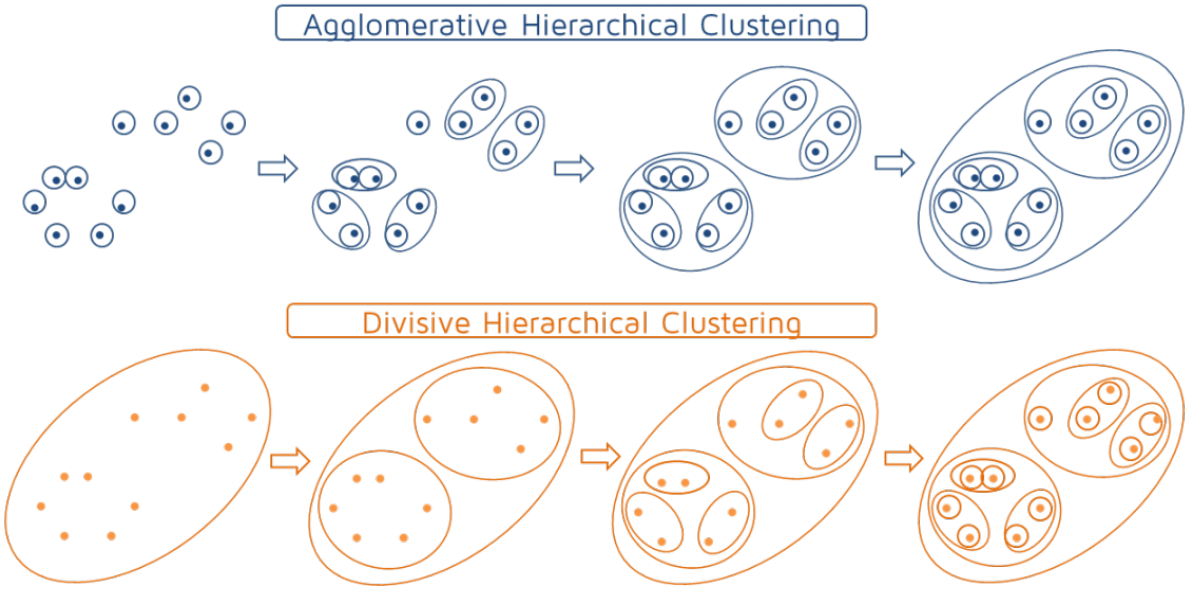
(Dis)Similarity measures

Manhattan distance (L1 norm)	$d_{fg} = \sum_c \lvert e_{fc} - e_{gc} \rvert$
Euclidean distance (L2 norm)	$d_{fg} = \sqrt{\sum_c (e_{fc} - e_{gc})^2}$
Mahalanobis distance	$d_{fg} = (e_f - e_g)' \Sigma^{-1} (e_f - e_g)$, where Σ is the covariance matrix of the data
Pearson correlation (centered)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2 \sum_c (e_{gc} - \bar{e}_g)^2}}$
Uncentered correlation (cosine angle)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c e_{fc} e_{gc}}{\sqrt{\sum_c e_{fc}^2 \sum_c e_{gc}^2}}$
Spearman's rank correlation	As Pearson correlation, but replace e_{gc} with the rank of e_{gc} within the expression values of gene g across all conditions $c = 1...C$
Absolute or squared correlation	$d_{fg} = 1 - \lvert r_{fg} \rvert$ or $d_{fg} = 1 - r_{fg}^2$

d_{fg}, distance between expression patterns for genes f and g. e_{gc}, expression level of gene g under condition c.

Source: D'haeseleer, 2005

Hierarchical clustering



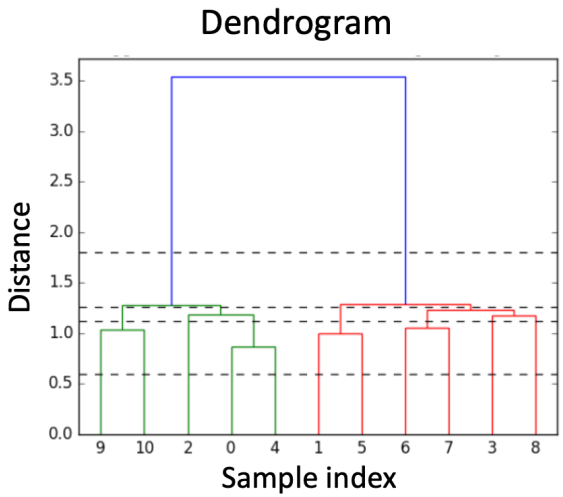
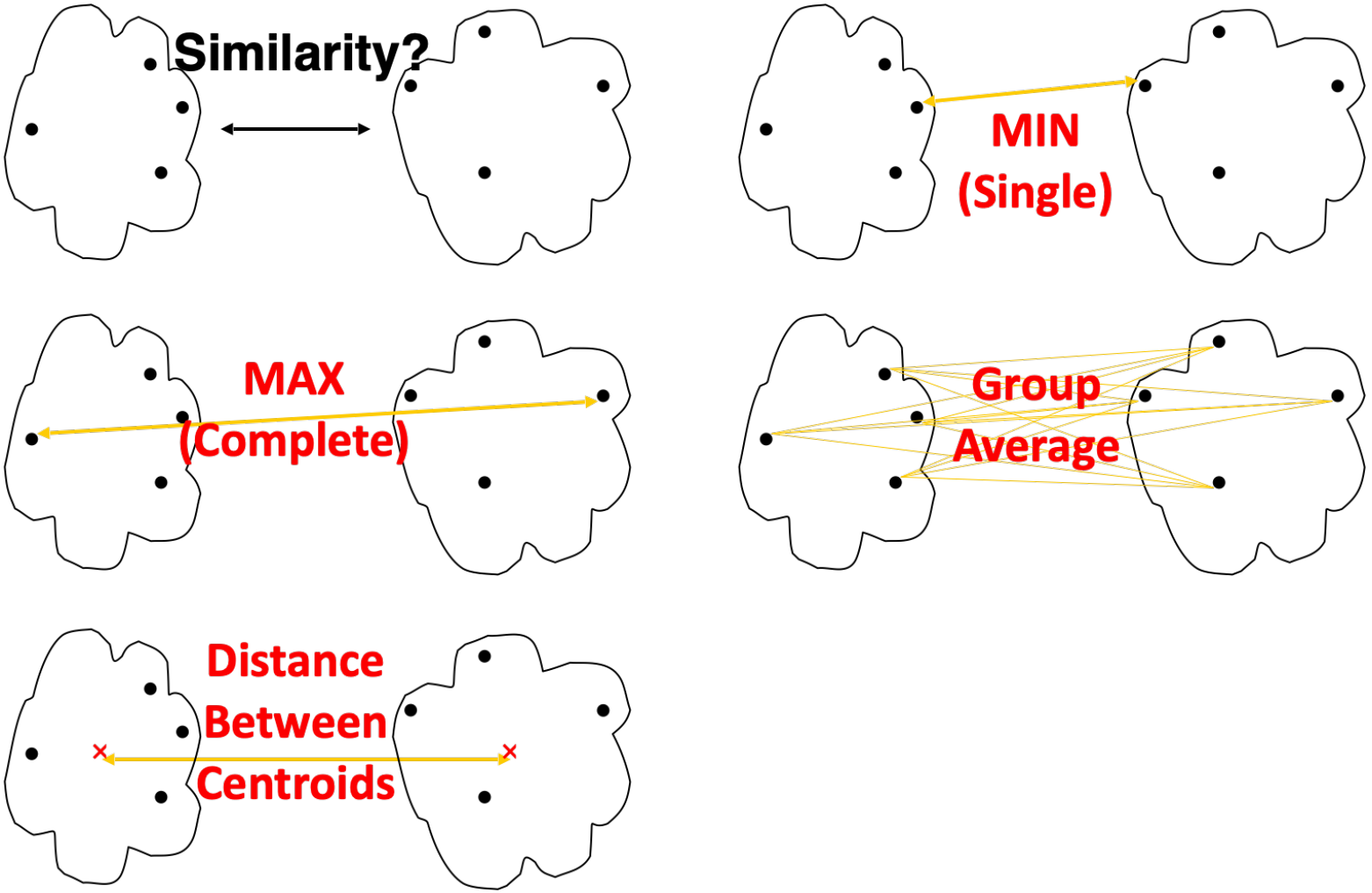


Figure source: https://quantdare.com/hierarchical-clustering/

- **Agglomerative**: start with the points as individual clusters, and merge the closest pair of clusters until only 1 (or *k*) cluster left
- **Divisive**: start with one cluster, & split a cluster until each cluster contains a point (or *k* clusters)
- Traditional hierarchical algorithms use a similarity measure to merge/split one cluster at a time.

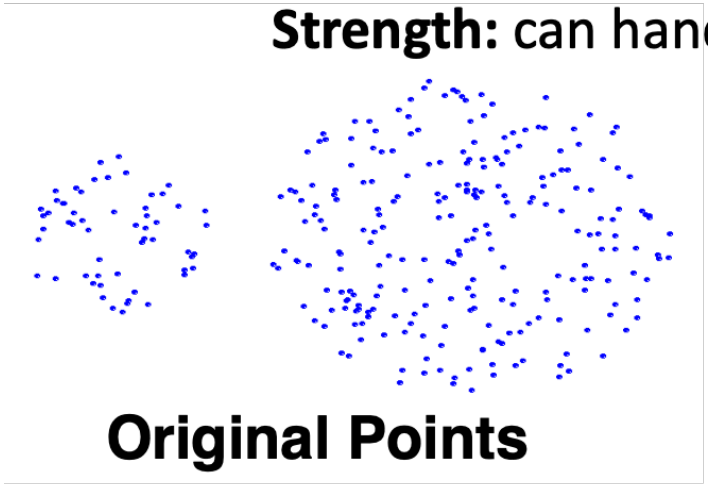
How to define inter-cluster similarity?

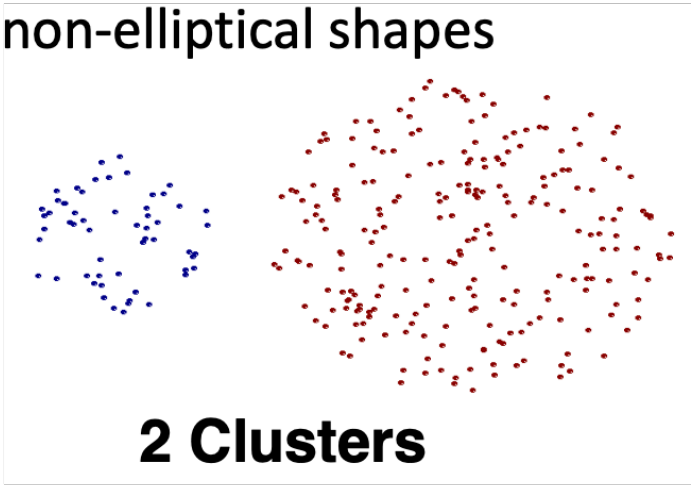


Source: Tan, Steinbach, & Kumar, 2004

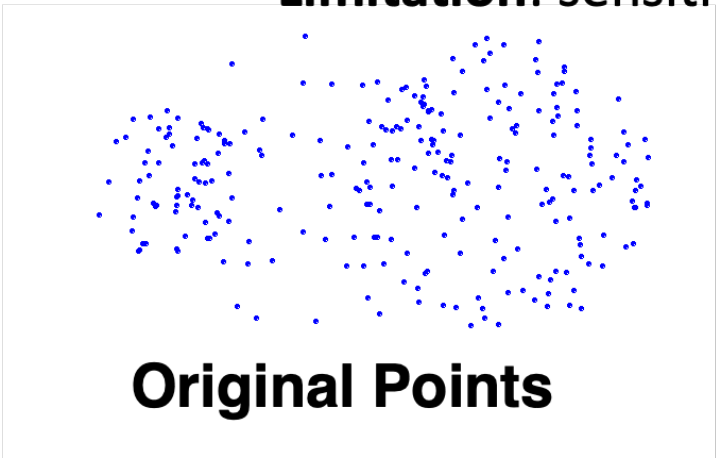
Strength and limitation of MIN (single) linkage

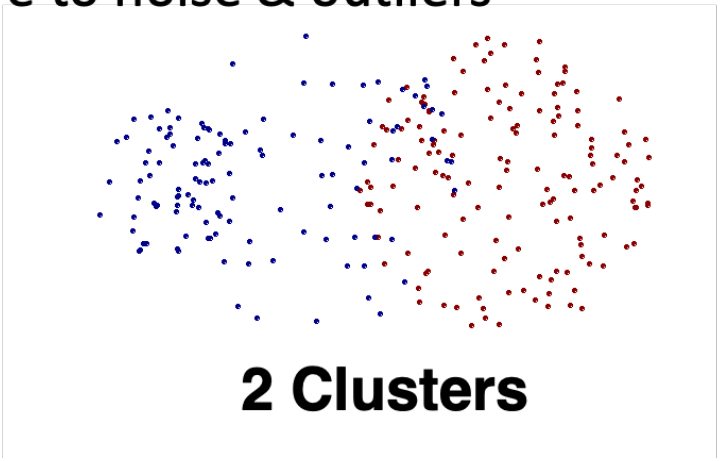
Strength: can handle non-elliptical shapes





Limitation: sensitive to noise & outliers

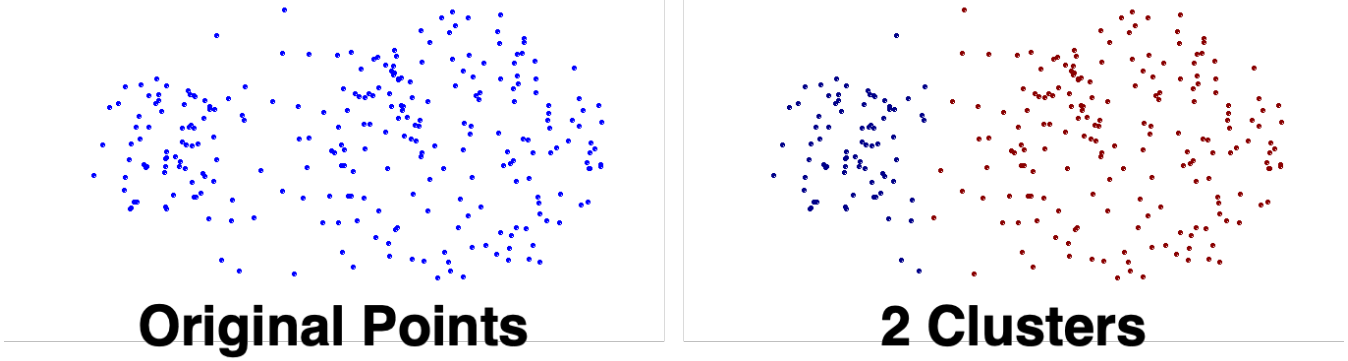




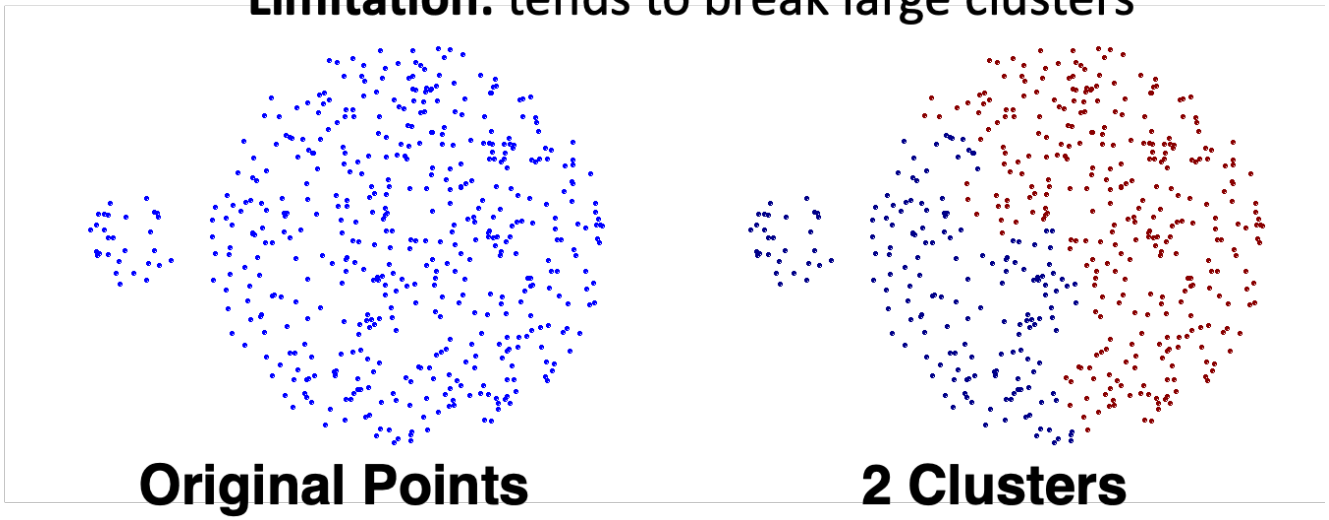
Source: Tan, Steinbach, & Kumar, 2004

Strength and limitation of MAX (complete) linkage

Strength: less susceptible to noise & outliers



Limitation: tends to break large clusters



Source: Tan, Steinbach, & Kumar, 2004

Hierarchical clustering: problems and limitations

- Once a decision is made to combine two clusters, it cannot be undone.
- No objective function is directly minimized.
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

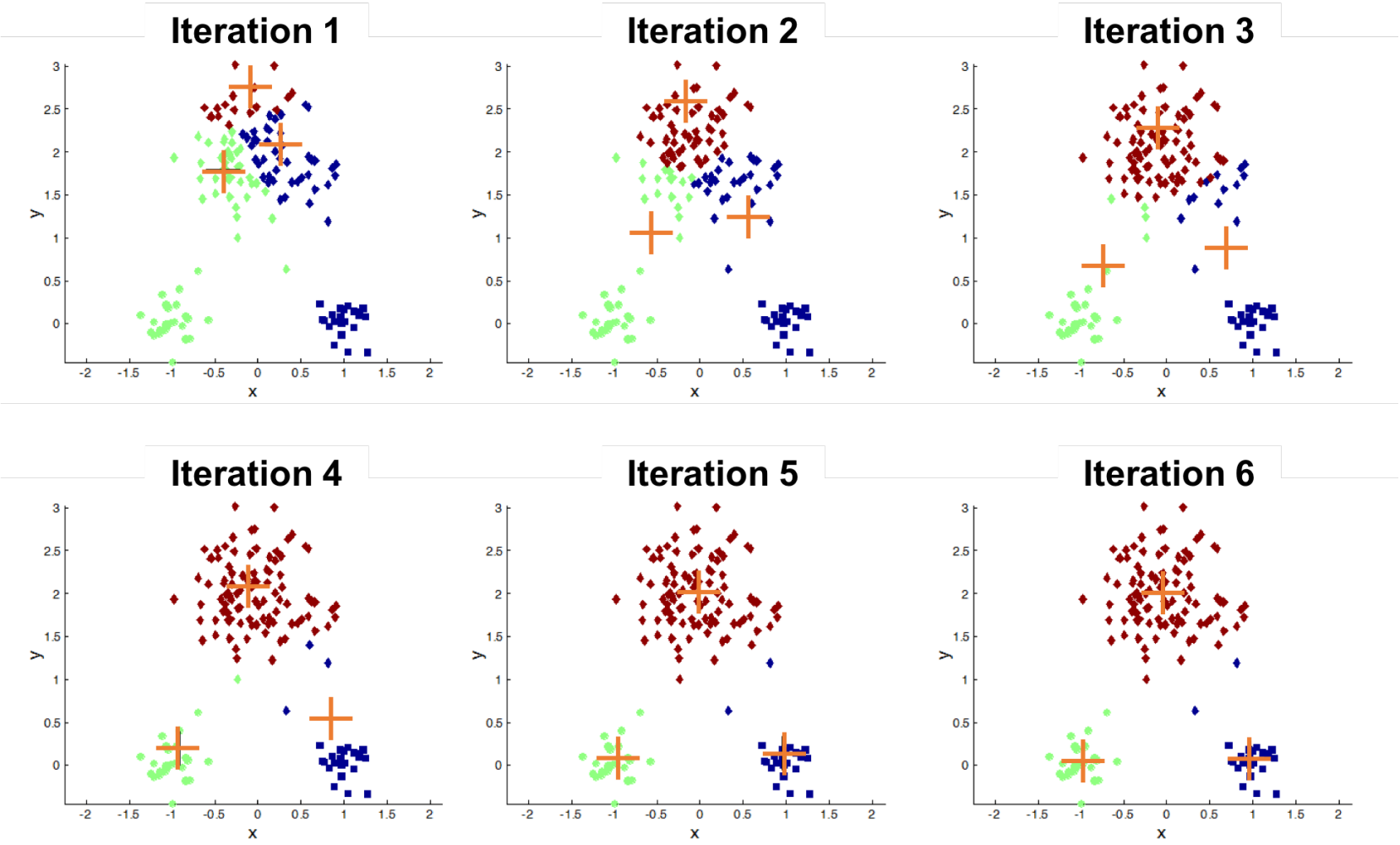
Source: Tan, Steinbach, & Kumar, 2004

K-means clustering algorithm

1. Select K points as initial centroids.
2. **repeat**
3. From K clusters by assigning all points to the closest centroid.
4. Recompute the centroid of each cluster.
5. **until** The centroids don't change.

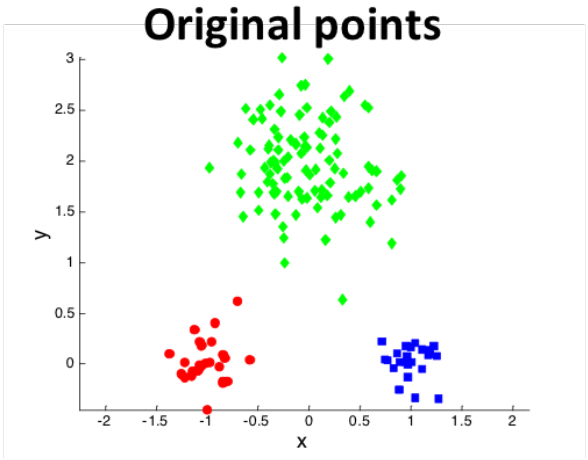
Source: Tan, Steinbach, & Kumar, 2004

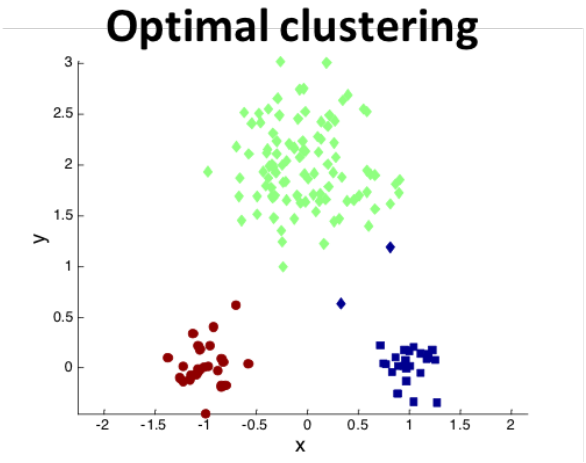
K-means clustering

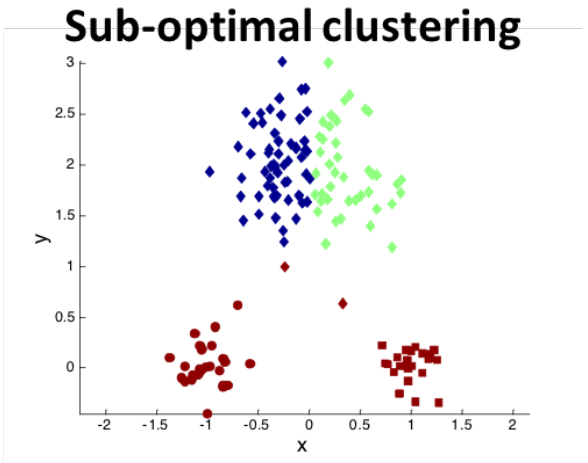


Source: Tan, Steinbach, & Kumar, 2004

Importance of choosing initial centroids





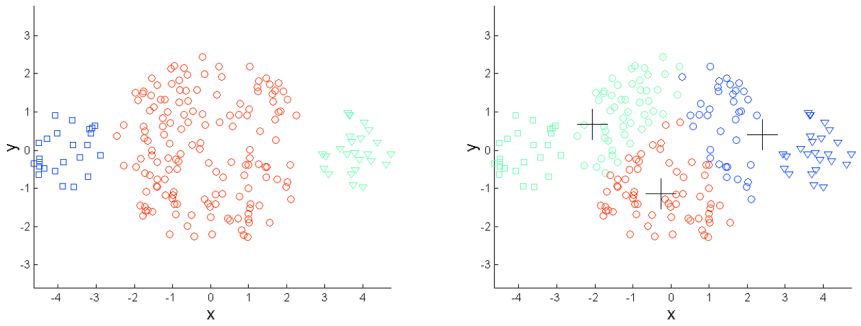


Solution: run K-means algorithm MULTIPLE times

Source: Tan, Steinbach, & Kumar, 2004

Limitations of *K*-means clustering

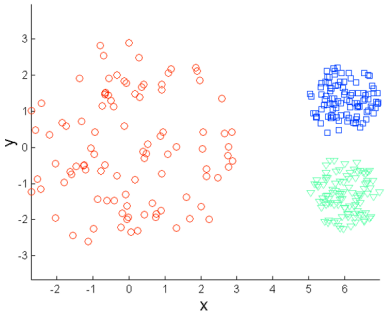
Different sizes

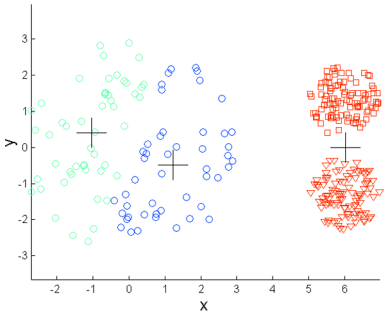


Original Points

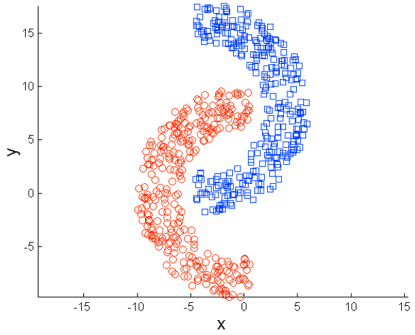
K-means (3 Clusters)

Different densities

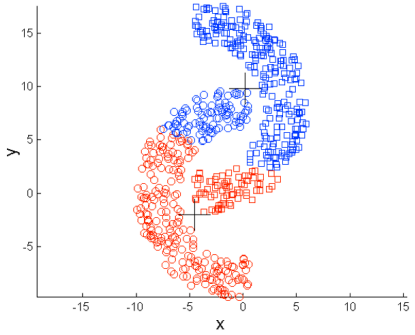




Non-globular shapes



Original Points

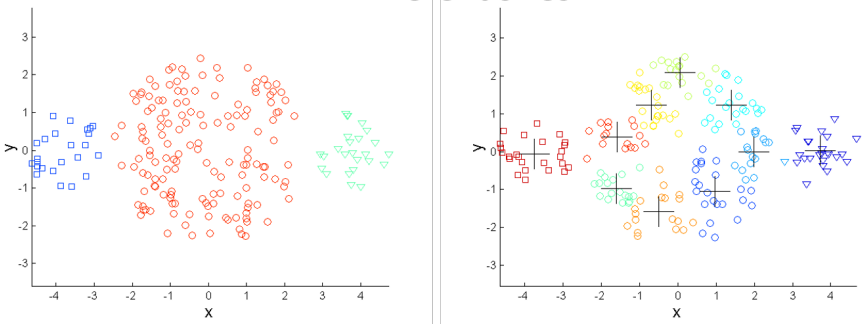


K-means (2 Clusters)

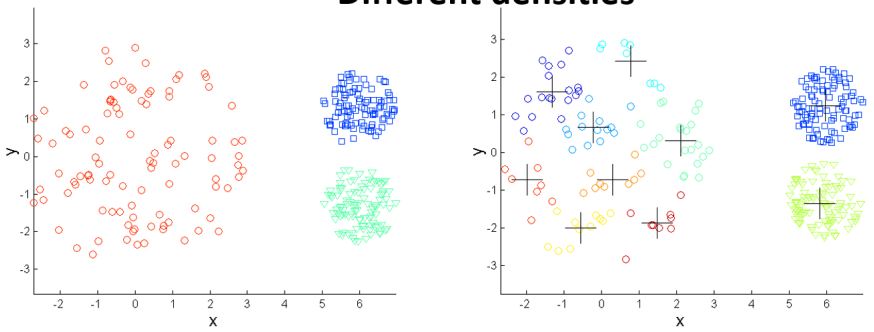
Source: Tan, Steinbach, & Kumar, 2004

Overcoming *K*-means clustering limitations

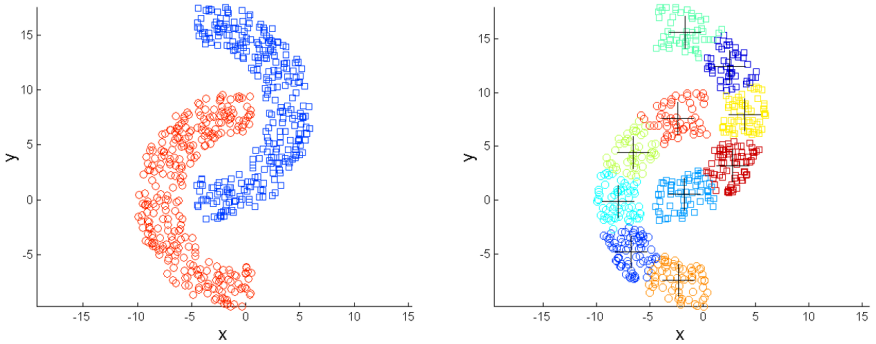
Different sizes



Different densities



Non-globular shapes

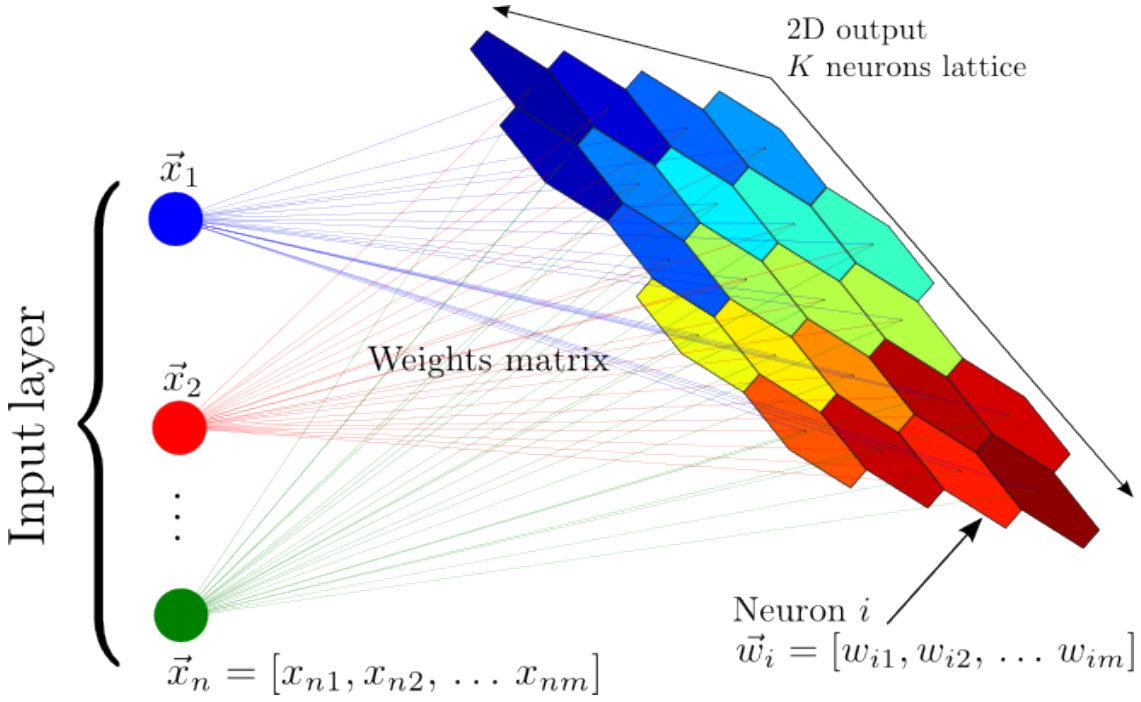


Solution: use many clusters, find parts of clusters, & put together

Source: Tan, Steinbach, & Kumar, 2004

Self-Organizing Map (SOM) clustering

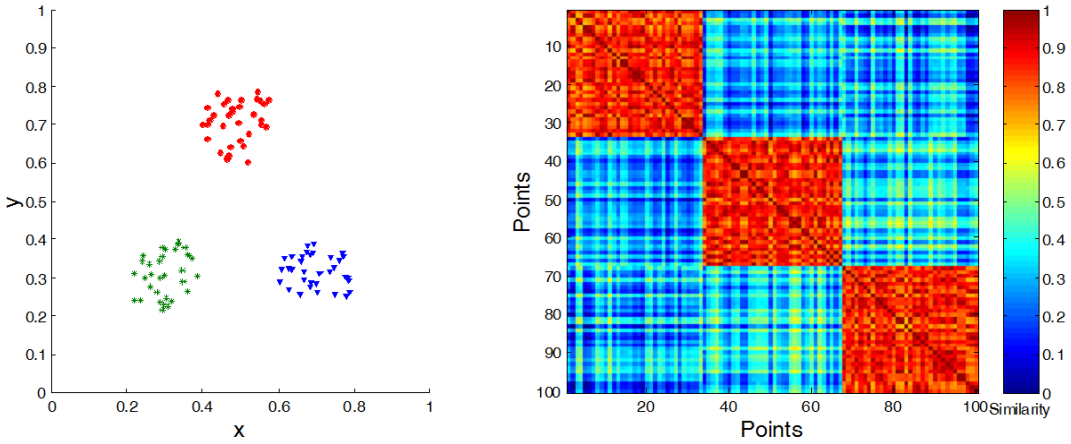
- is a type of artificial neural network (ANN)
- produces a low-dimensional (typically 2D), discretized representation of the input space



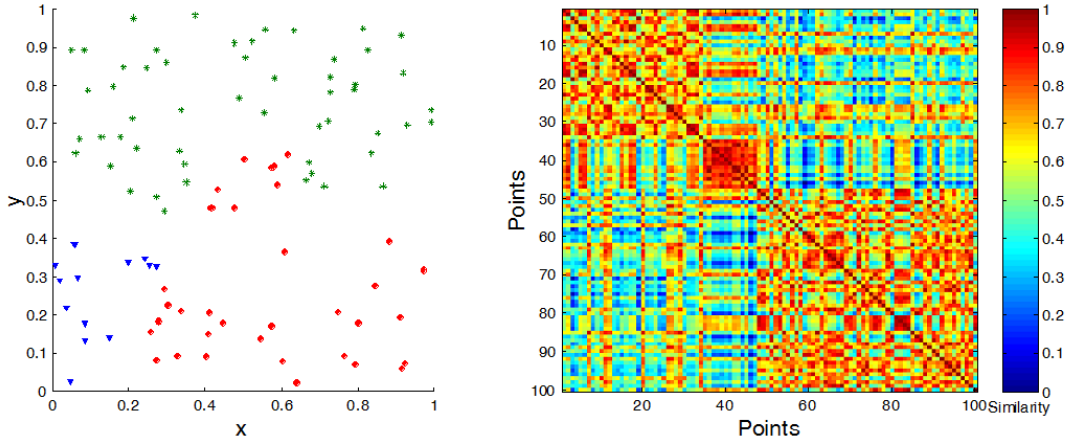
Source: http://www.cs.us.es/~fsancho/?e=136

Using similarity matrix for cluster validation

Order the similarity matrix with respect to cluster labels and inspect visually

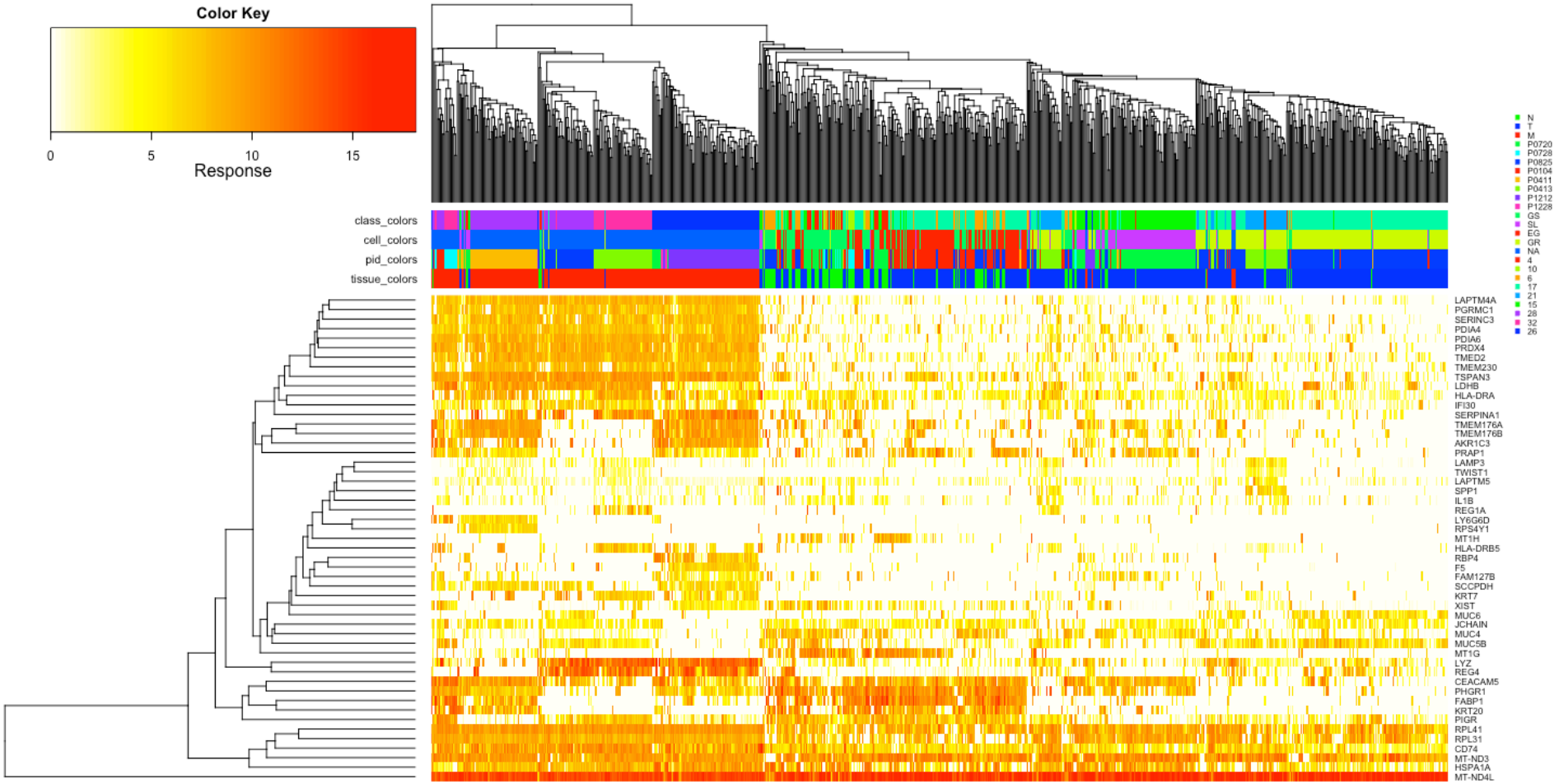


Clusters in random data are not so crisp



Source: Tan, Steinbach, & Kumar, 2004

Use clustering to visualize high-dimensional data



Supervised learning

Brief history of supervised machine learning

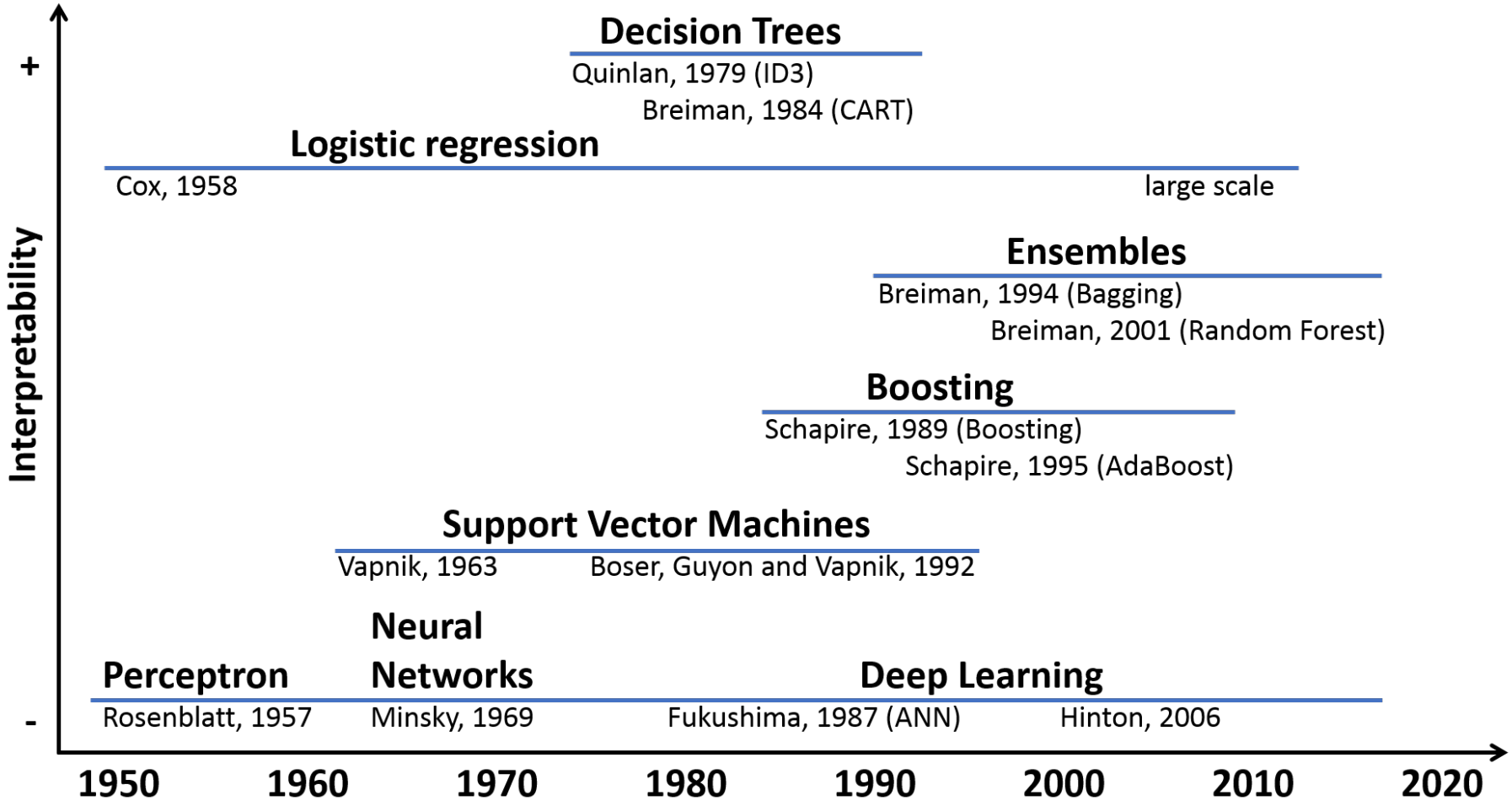
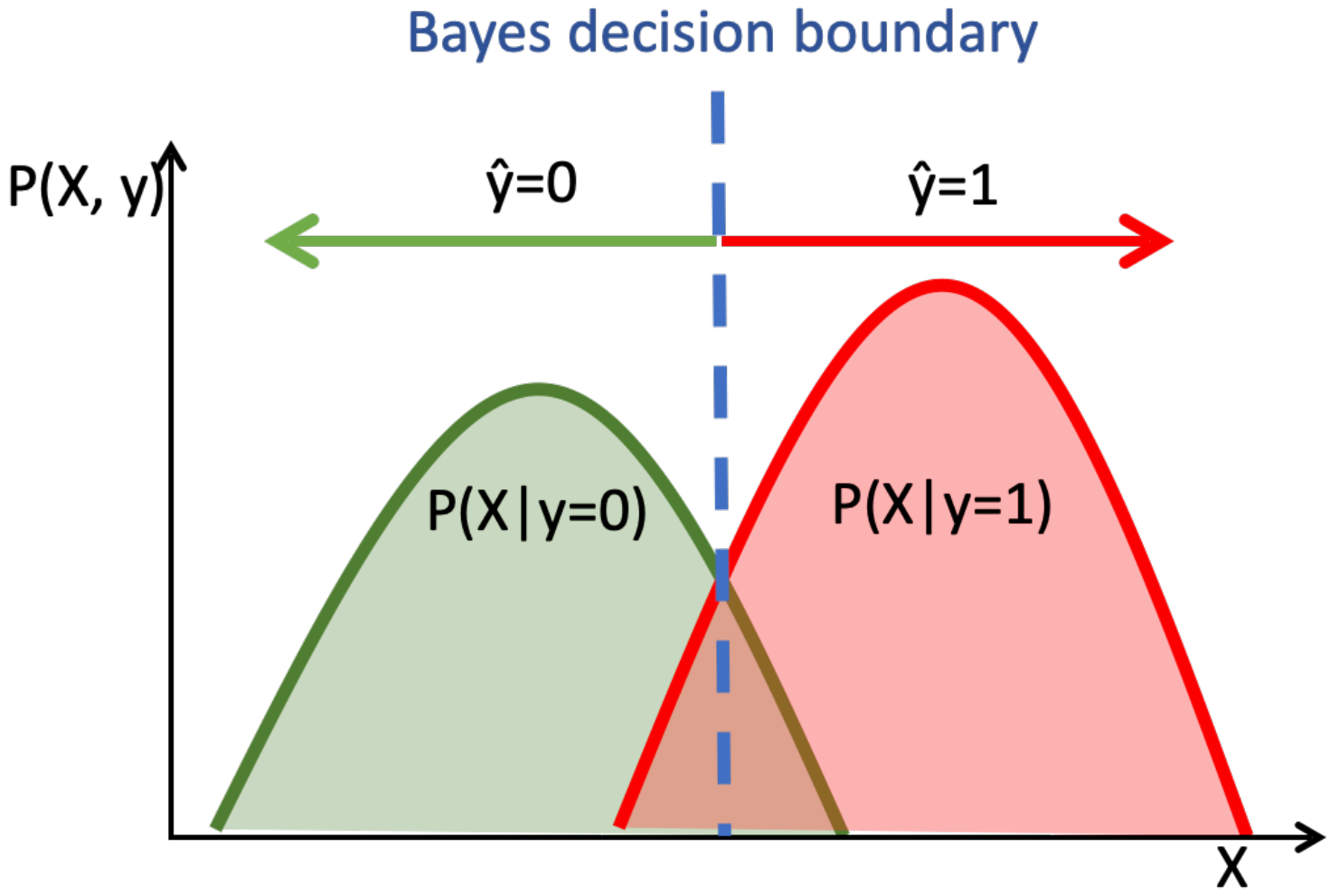


Figure: modified from Petersen, 2015

Classification



No Free Lunch (NFL) theorem

- No algorithm is the best
- No universal best algorithm

The "no free lunch" (NFL) theorem of David Wolpert and William Macready appears in the 1997 "No Free Lunch Theorems for Optimization". Wolpert had previously derived no free lunch theorems for machine learning (statistical inference). In 2005, Wolpert and Macready themselves indicated that the first theorem in their paper "state[s] that any two optimization algorithms are equivalent when their performance is averaged across all possible problems".

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

Supervised learning methods

- K-nearest neighbors (*K*NN)
- Support vector machine (SVM)

- Logistic regression
- Artificial neural networks (ANN)

K-nearest neighbor (*K*NN)

- **Classification**: an object is classified by a majority vote of its *K*NNs Use odd number to break ties in binary classification

- **Regression**: an object's value is the average values of its *K*NNs Suppose: minimizes variance (squared error loss)

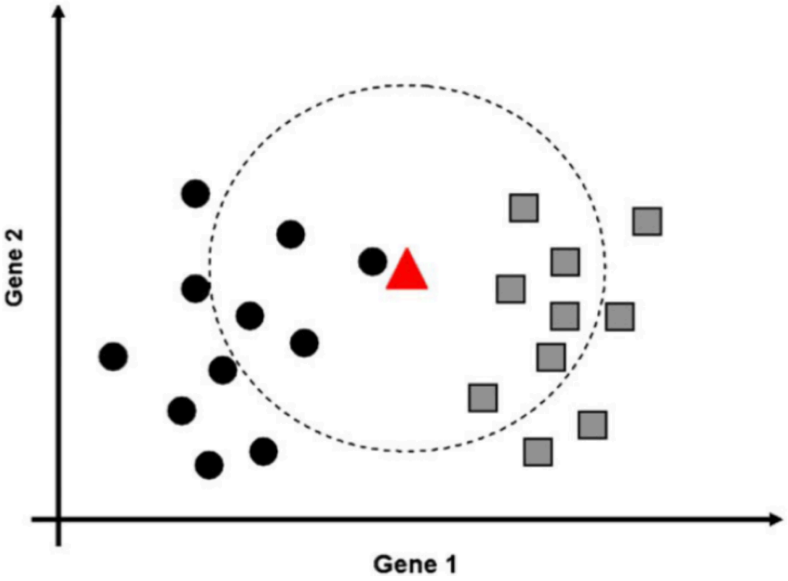
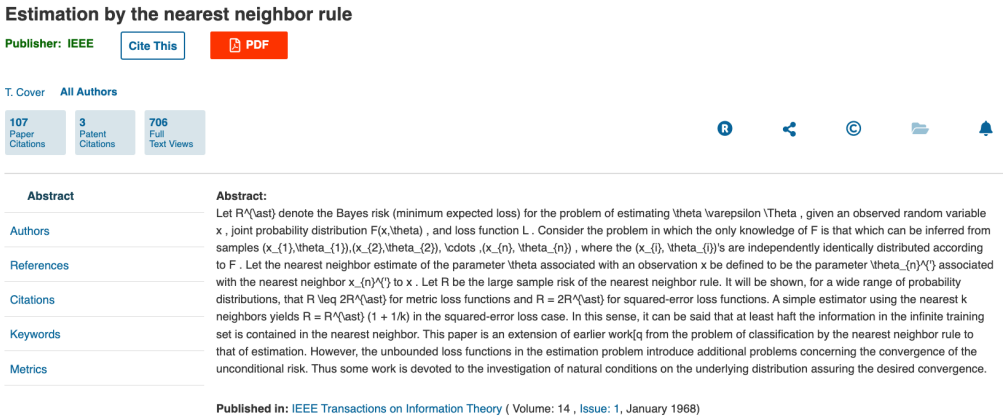


Figure source: Tarca et al., 2008



Estimation by the nearest neighbor rule

Publisher: IEEE | Cite This | PDF

T. Cover All Authors

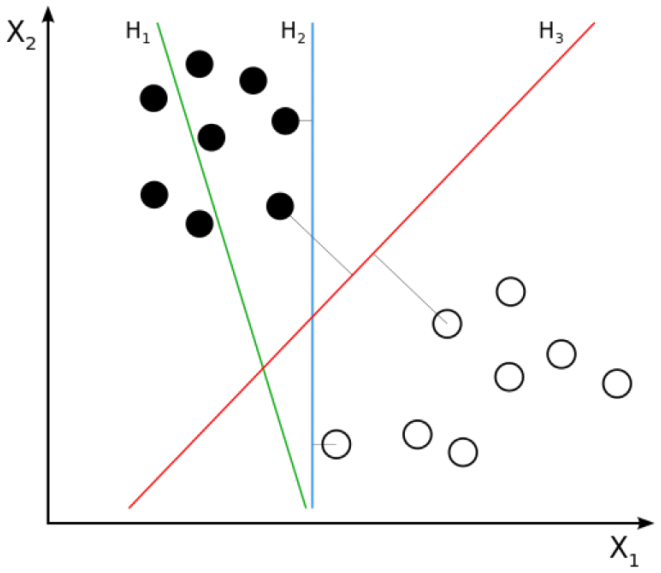
107 Paper Citations | 3 Patent Citations | 706 Full Text Views

Abstract | Authors | References | Citations | Keywords | Metrics

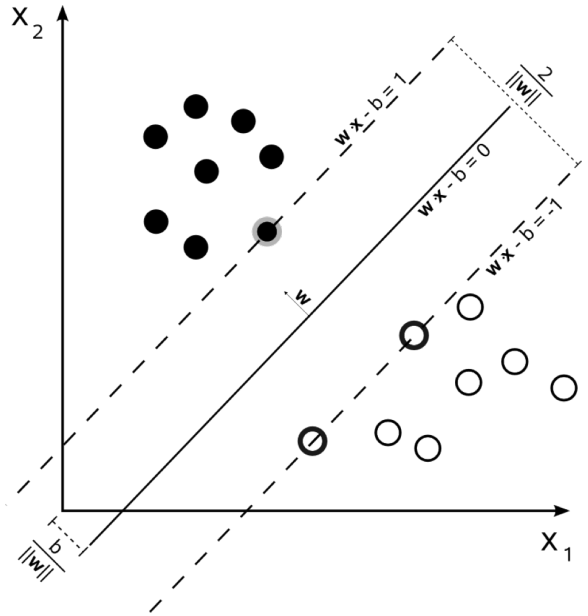
Abstract:
Let R^{\ast} denote the Bayes risk (minimum expected loss) for the problem of estimating \theta \varepsilon \Theta , given an observed random variable x , joint probability distribution F(x,\theta) , and loss function L . Consider the problem in which the only knowledge of F is that which can be inferred from samples (x_{1},\theta_{1}),(x_{2},\theta_{2}), \cdots ,(x_{n}, \theta_{n}) , where the (x_{i}, \theta_{i})'s are independently identically distributed according to F . Let the nearest neighbor estimate of the parameter \theta associated with an observation x be defined to be the parameter \theta_{n}^{'} associated with the nearest neighbor x_{n}^{'} to x . Let R be the large sample risk of the nearest neighbor rule. It will be shown, for a wide range of probability distributions, that R \leq 2R^{\ast} for metric loss functions and R = 2R^{\ast} for squared-error loss functions. A simple estimator using the nearest k neighbors yields R = R^{\ast} (1 + 1/k) in the squared-error loss case. In this sense, it can be said that at least haft the information in the infinite training set is contained in the nearest neighbor. This paper is an extension of earlier work[q from the problem of classification by the nearest neighbor rule to that of estimation. However, the unbounded loss functions in the estimation problem introduce additional problems concerning the convergence of the unconditional risk. Thus some work is devoted to the investigation of natural conditions on the underlying distribution assuring the desired convergence.

Published in: IEEE Transactions on Information Theory (Volume: 14 , Issue: 1, January 1968)

Support vector machines (SVM)



- H1 does not separate the classes
- H2 does, but only with a small margin
- H3 separates them with the max margin

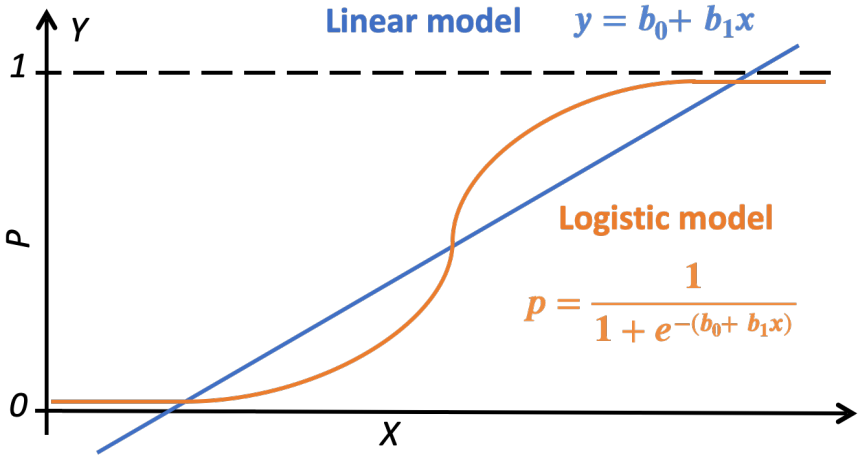


- Maximum-margin hyperplane for an SVM trained with samples from 2 classes
- Samples on the margin are called the support vectors

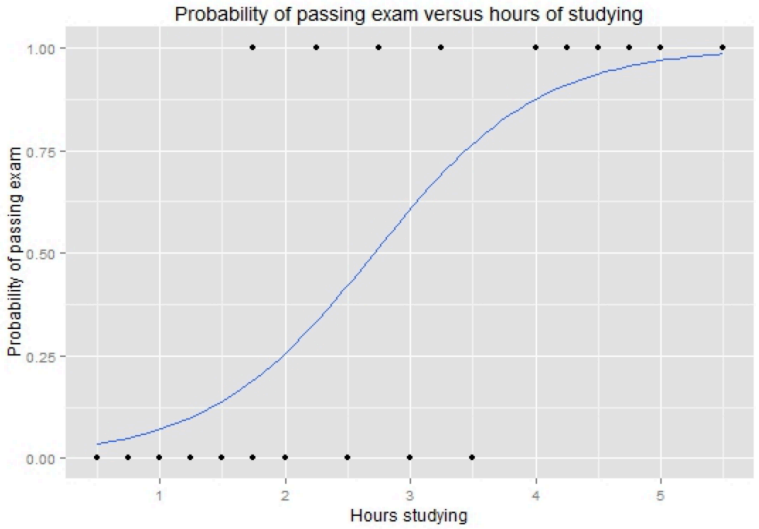
Normalize predictors between [-1, +1]

Source: wikipedia

Logistic regression (classification)



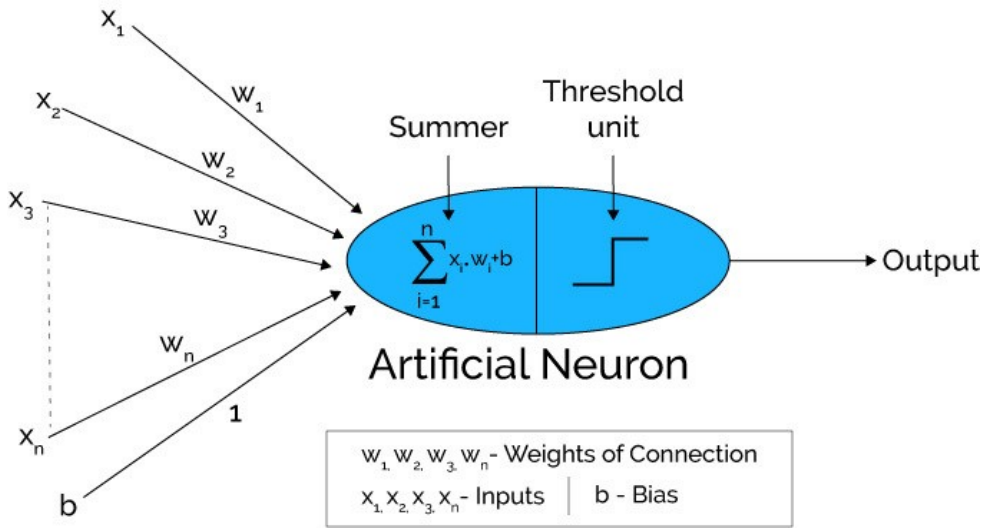
- Logit model: $ln[\frac{p}{(1-p)}] = b_0 + b_1 X$
- p is the probability that event Y occurs
  - $\frac{p}{(1-p)}$ is the "odds ratio"
  - $ln[\frac{p}{(1-p)}]$ is the log odds ratio, or logit



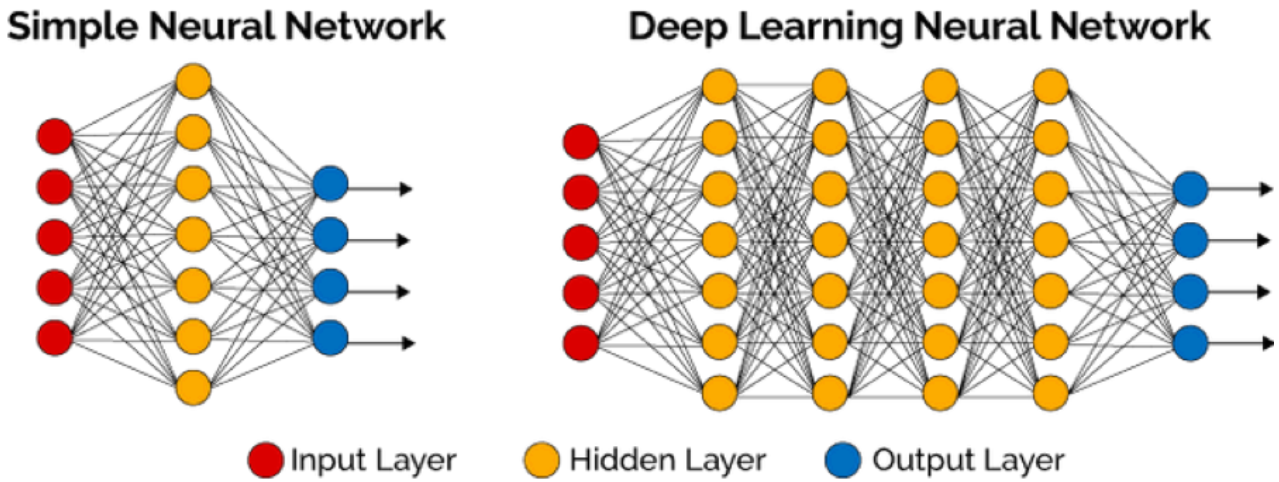
- Estimated probability: $p = \frac{1}{1 + e^{-(b_0 + b_1 X)}}$
  - If $b_0 + b_1 X = 0$, then $p = 0.5$
  - As $b_0 + b_1 X$ gets large, $p -> 1$
  - As $b_0 + b_1 X$ gets small, $p -> 0$

Figure source: wikipedia

Artificial neural networks (ANN)



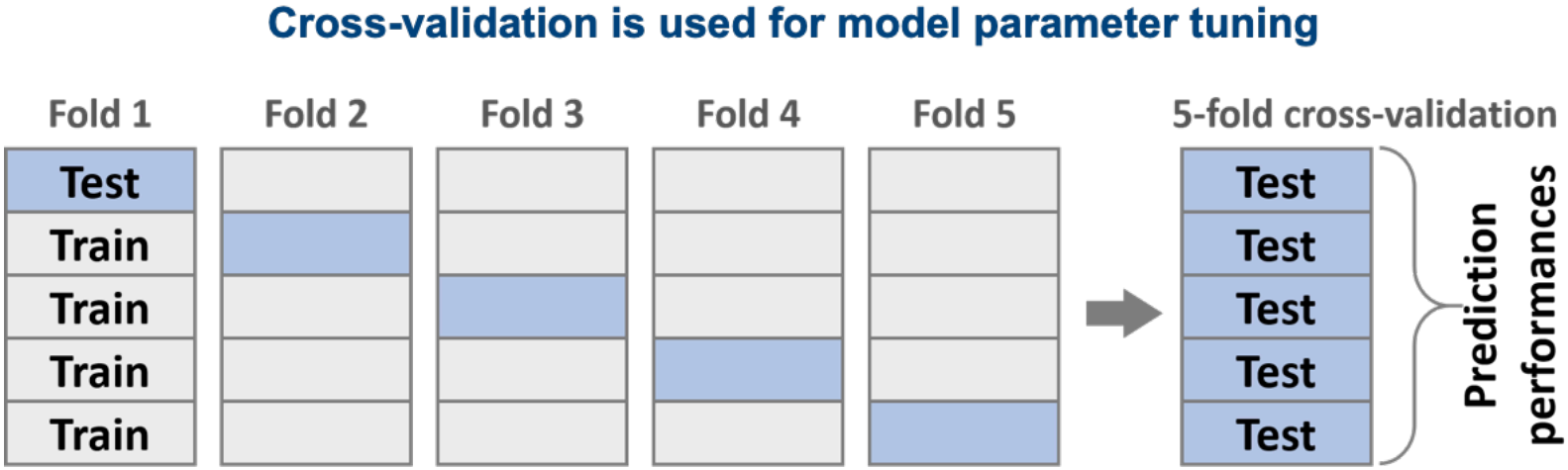
Source: Kafetzopoulo et al, 2013



Source: https://towardsdatascience.com/mnist-vs-mnist-how-i-was-able-to-speed-up-my-deep-learning-11c0787e6935

A three-phase process for supervised learning

1. **Training**: a model is constructed from the training samples
   - Algorithm finds relationships between predictors and outcome
   - Relationships are summarized in a model
2. **Testing**: fit the model on a test set whose outcomes are known but not used for training the model

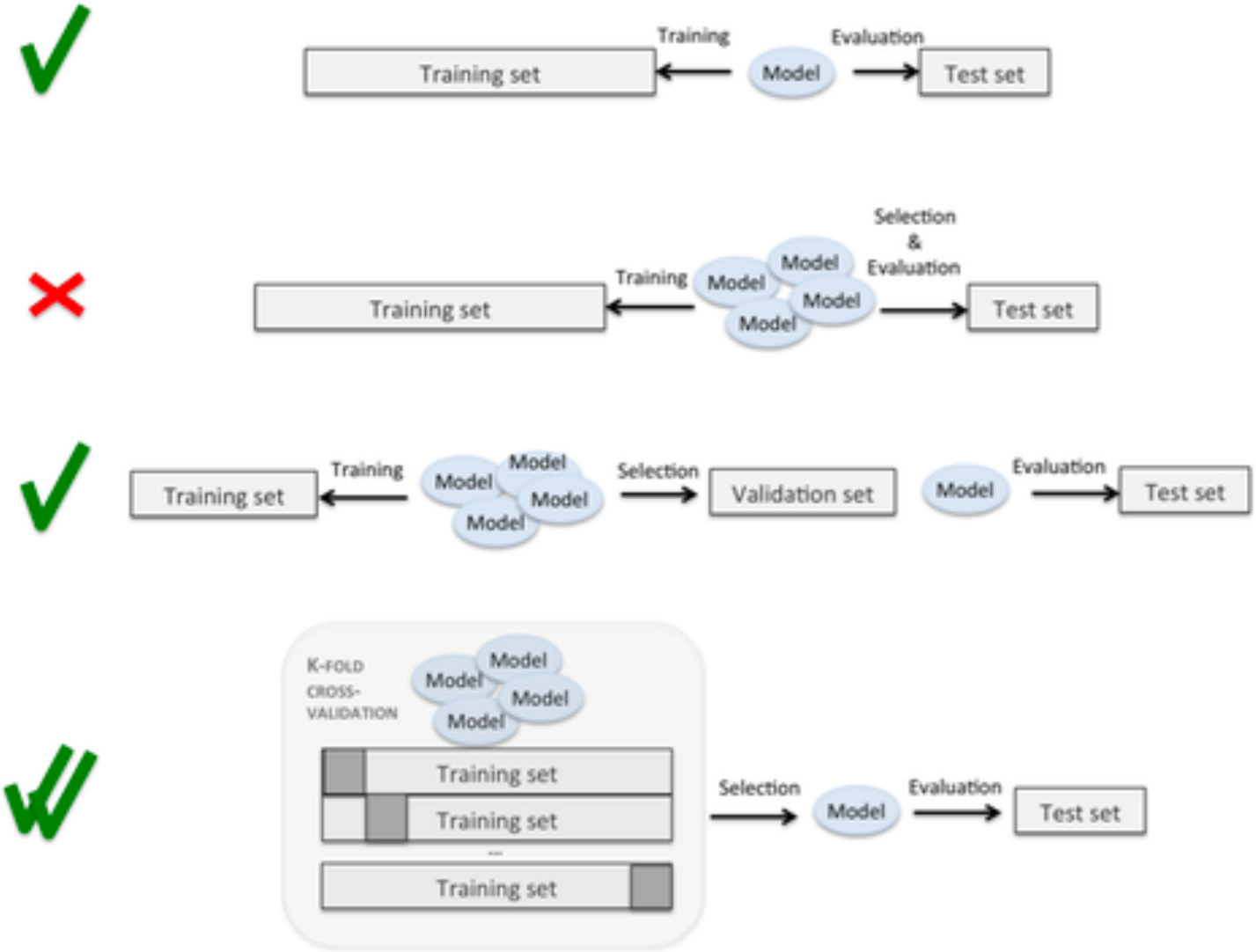


3. **Prediction**: apply the model the new data whose outcomes are unknown

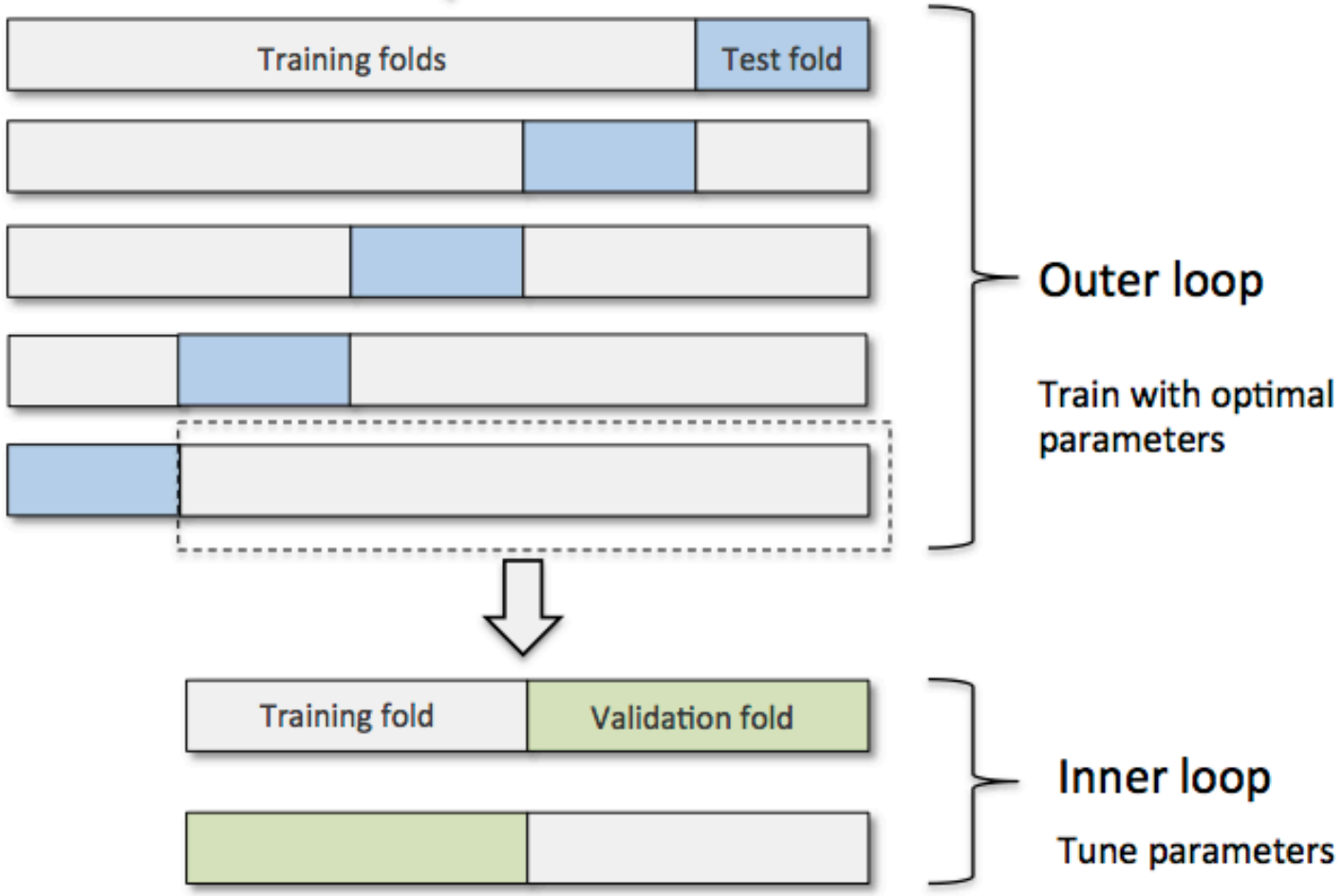
How to evaluate a model?

- Split data into training and testing
- User the training data to decide a model
- Use test set to verify model
  - Error
  - Accuracy
  - ROC

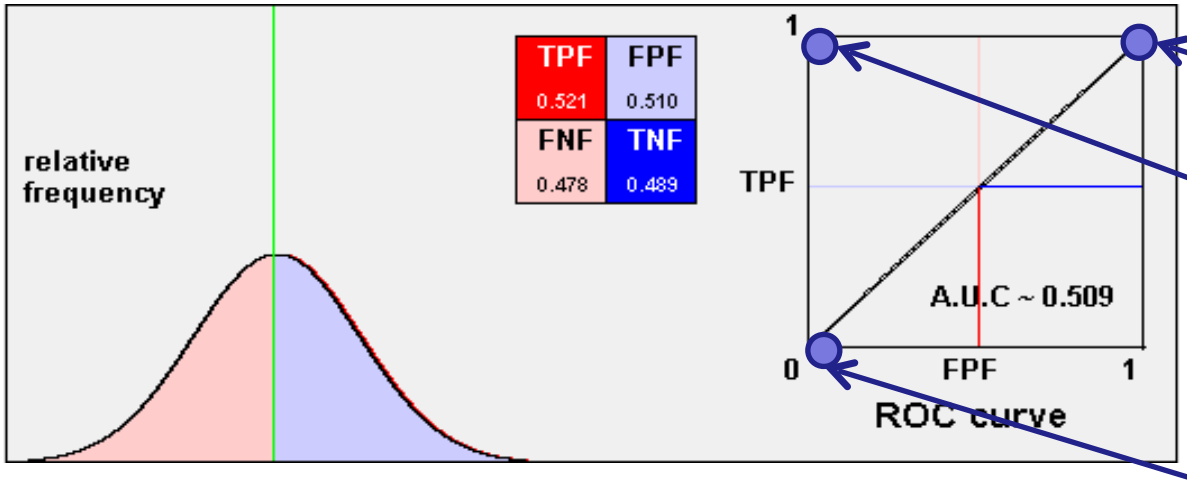
Train a model & tune its hyperparameters



Nested-cross validation



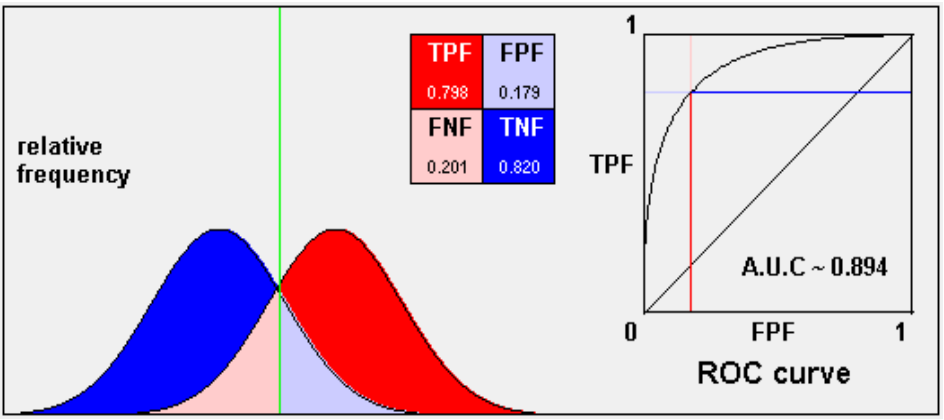
Receiver Operating Characteristic (ROC) curve

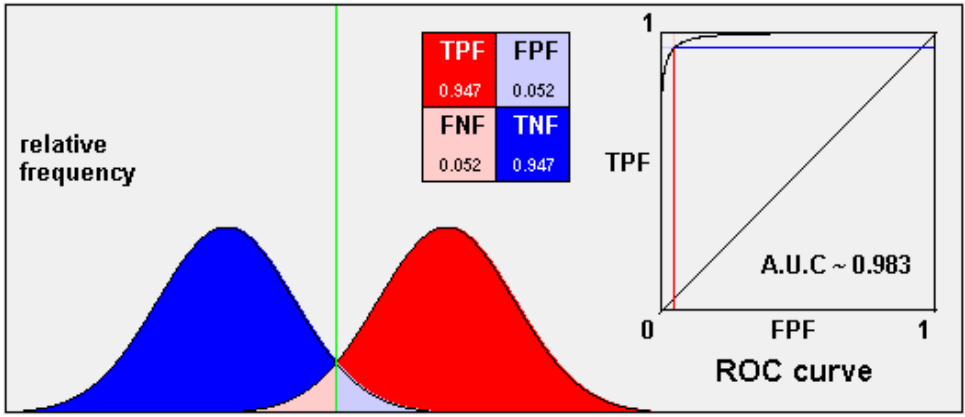


Predicts every instance to be positive class

The ideal model

Predicts every instance to be a negative class

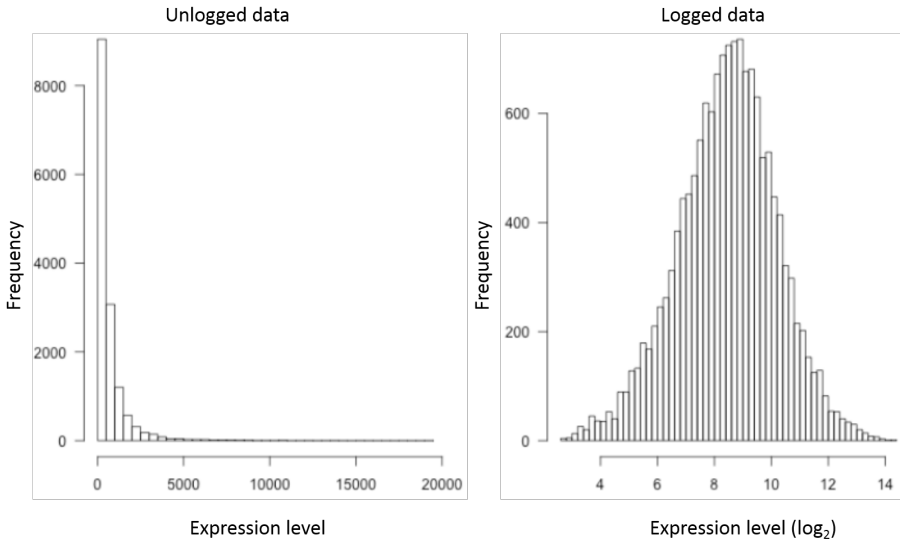


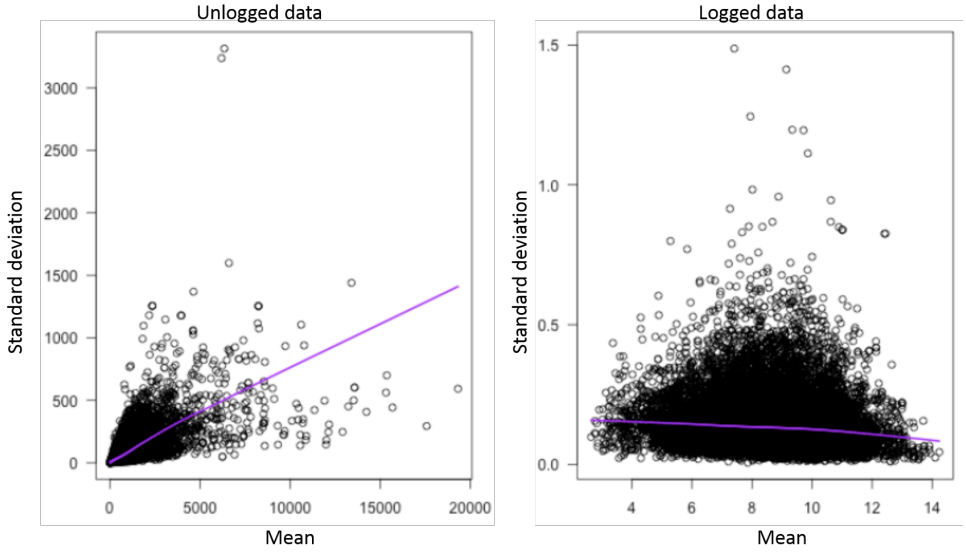


Source: http://www.anaesthetist.com/mnm/stats/roc/Findex.htm

Pre-processing: log2 transformation

- Improves the characteristics of the data distribution
- Allows the use of classical parametric statistics for analysis
- Stabilizes the variance
- Compresses the range of data

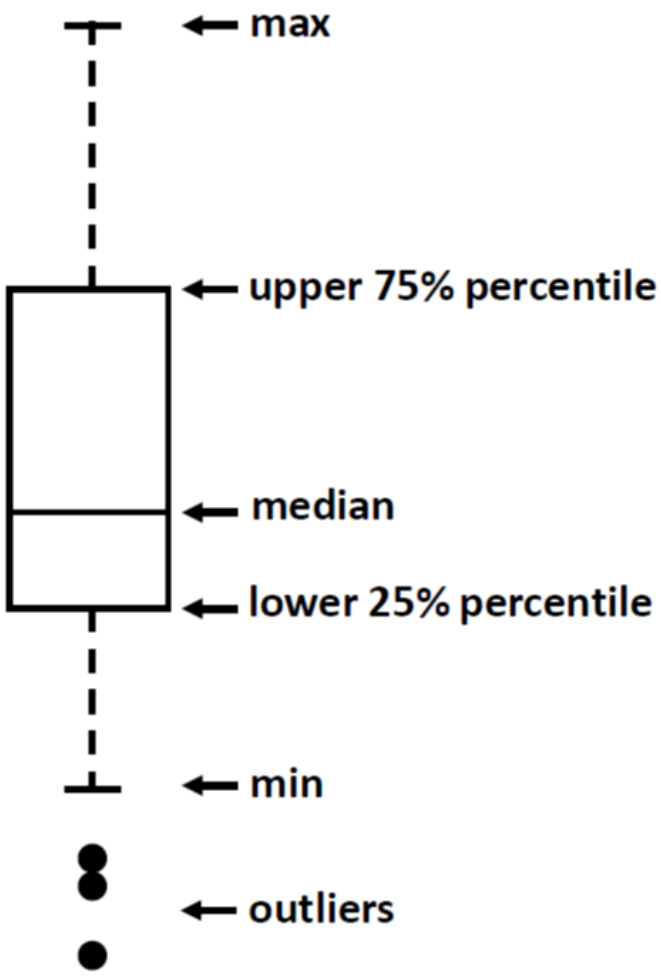




Source: Cowley & Ying, 2011

Quality control (QC)

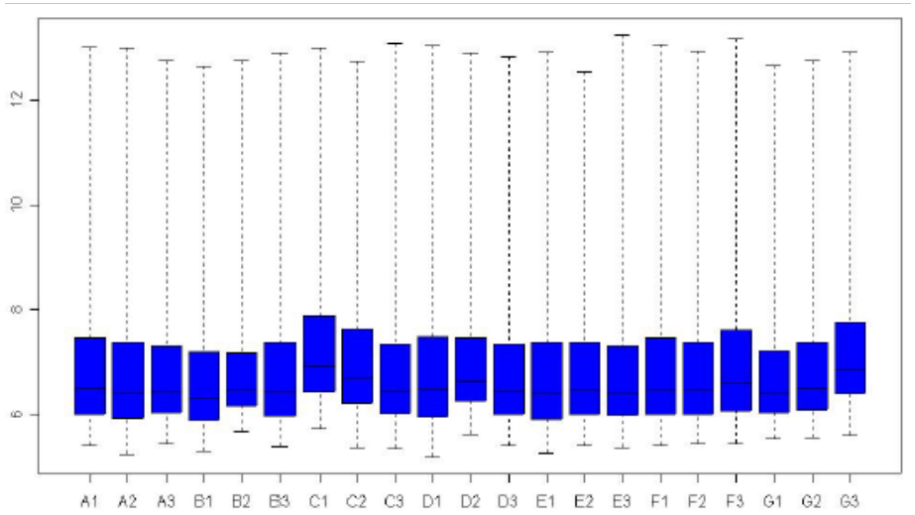
Boxplots


max
upper 75% percentile
median
lower 25% percentile
min
outliers

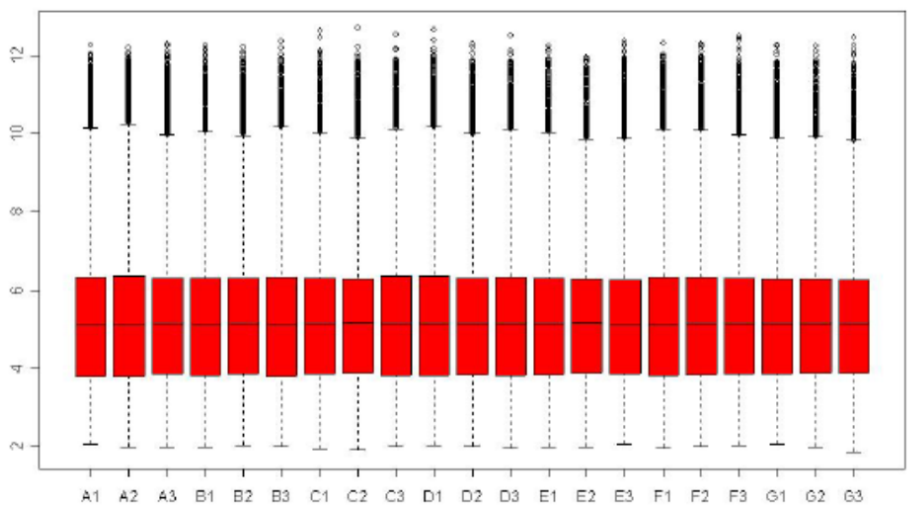
QC: boxplots

signal intensity data

Before



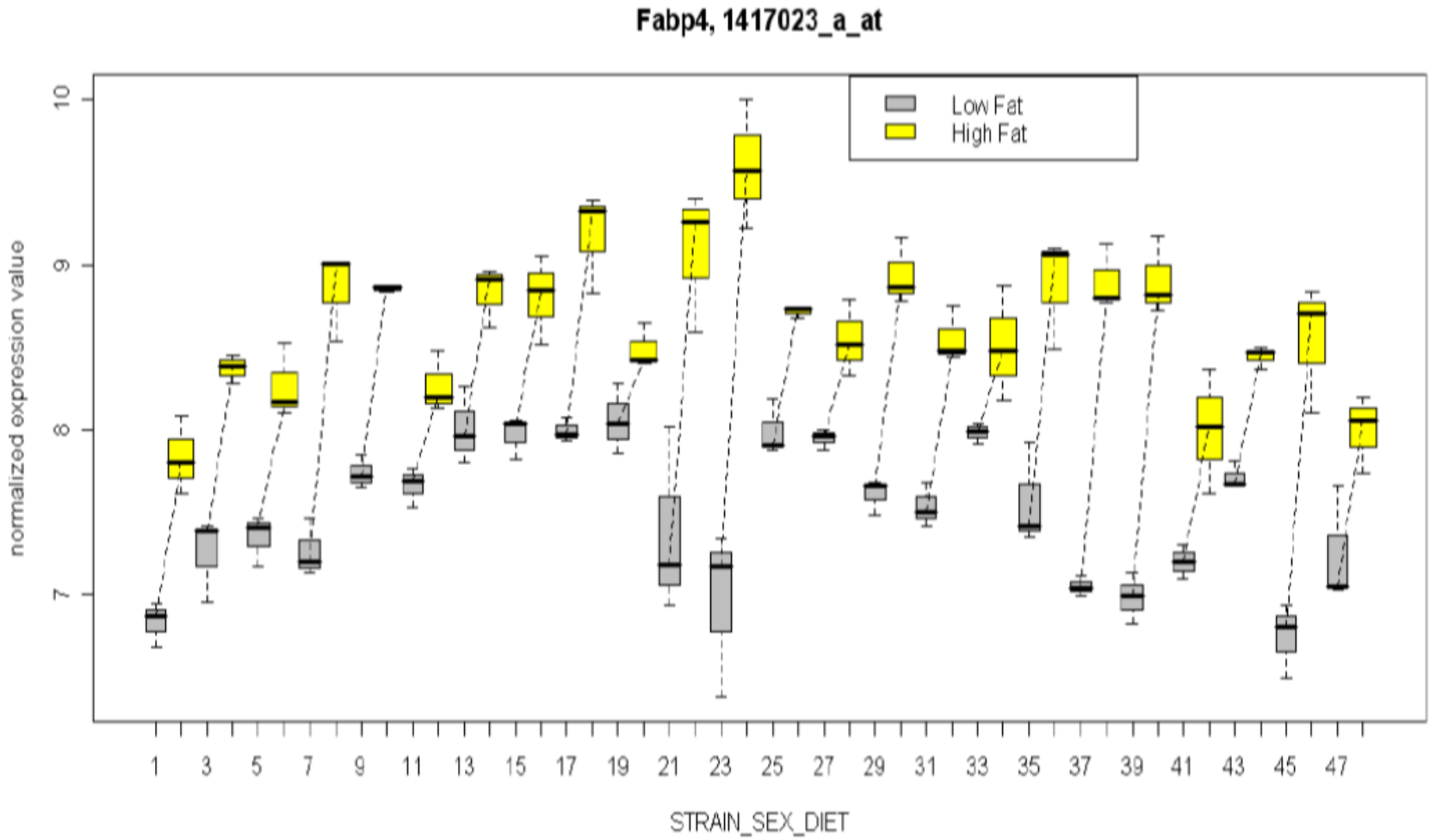
After



Need normalization

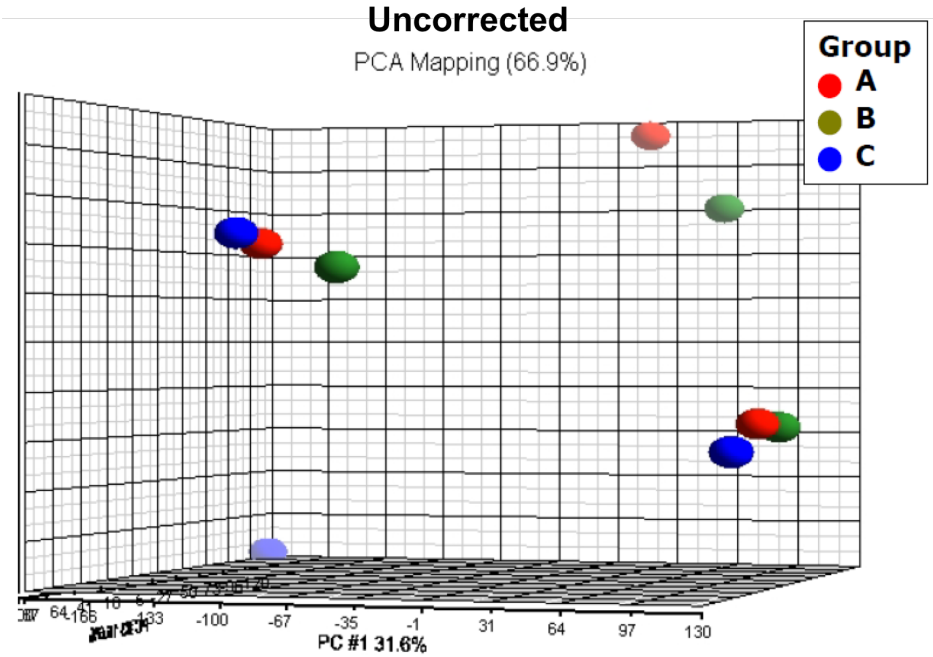
Source: Shockley

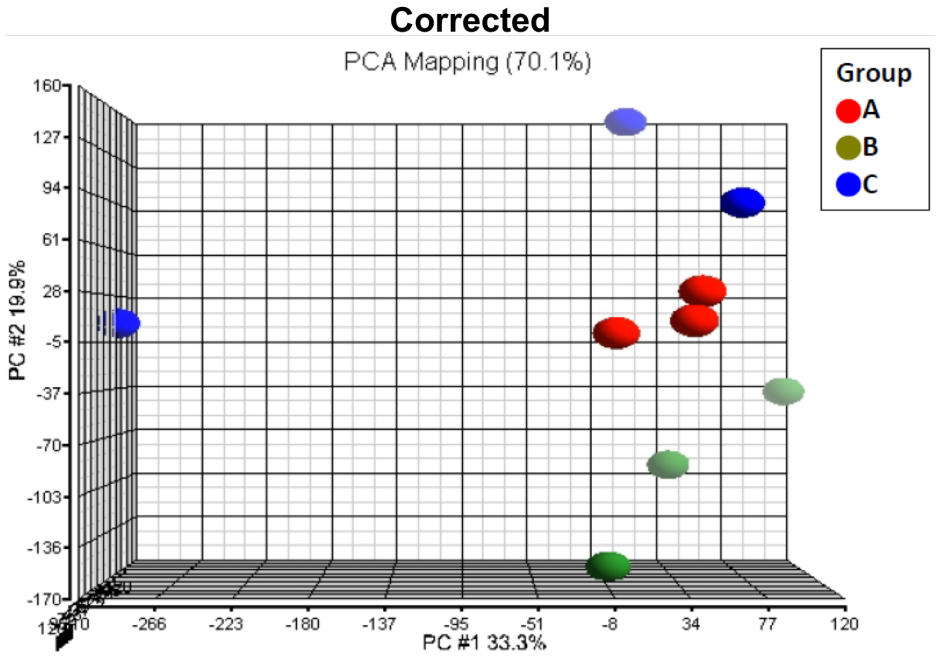
QC: Use known knowledge



Source: Shockley

QC: PCA

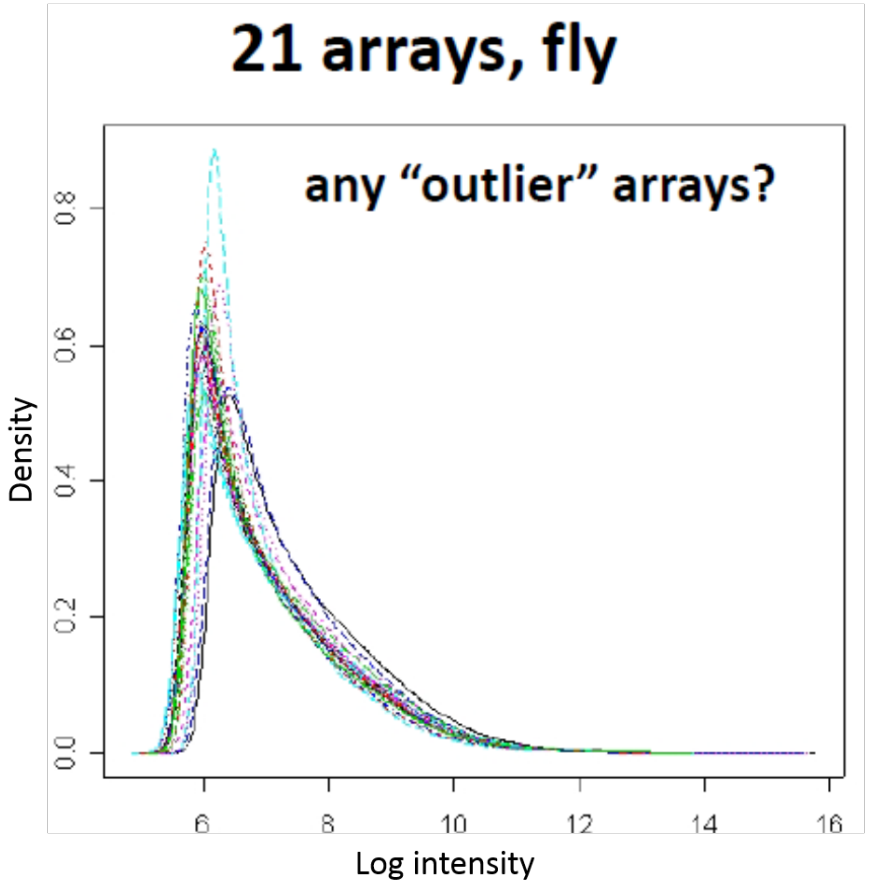


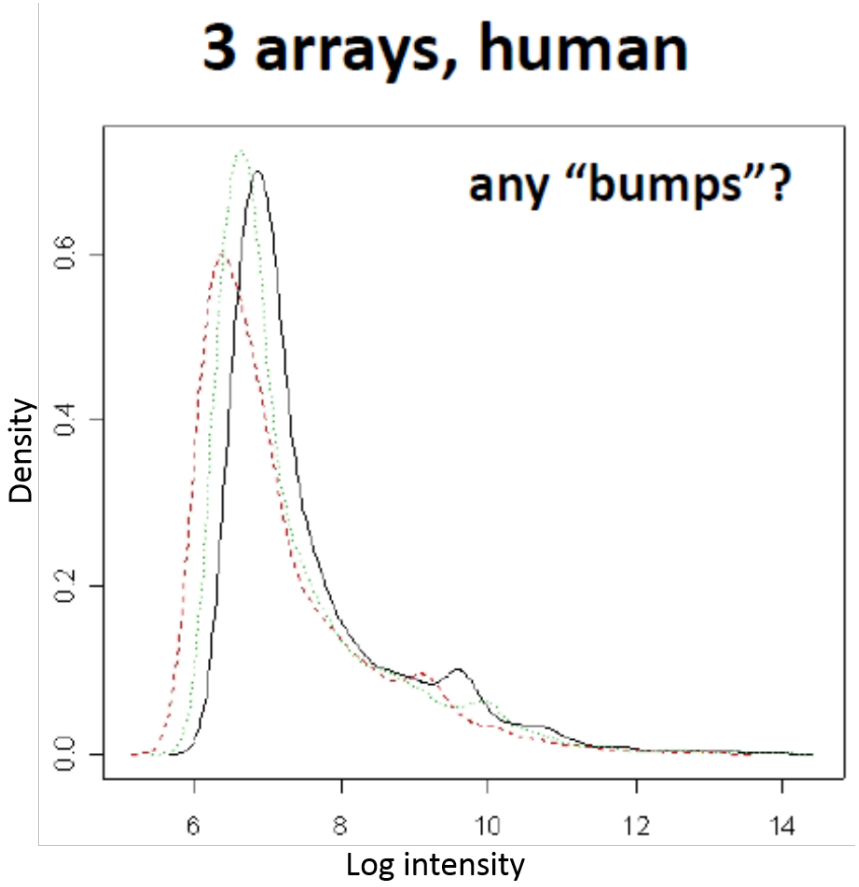


Need batch effect correction

Source: Shockley

QC: Kernal Density Estimation (KDE)

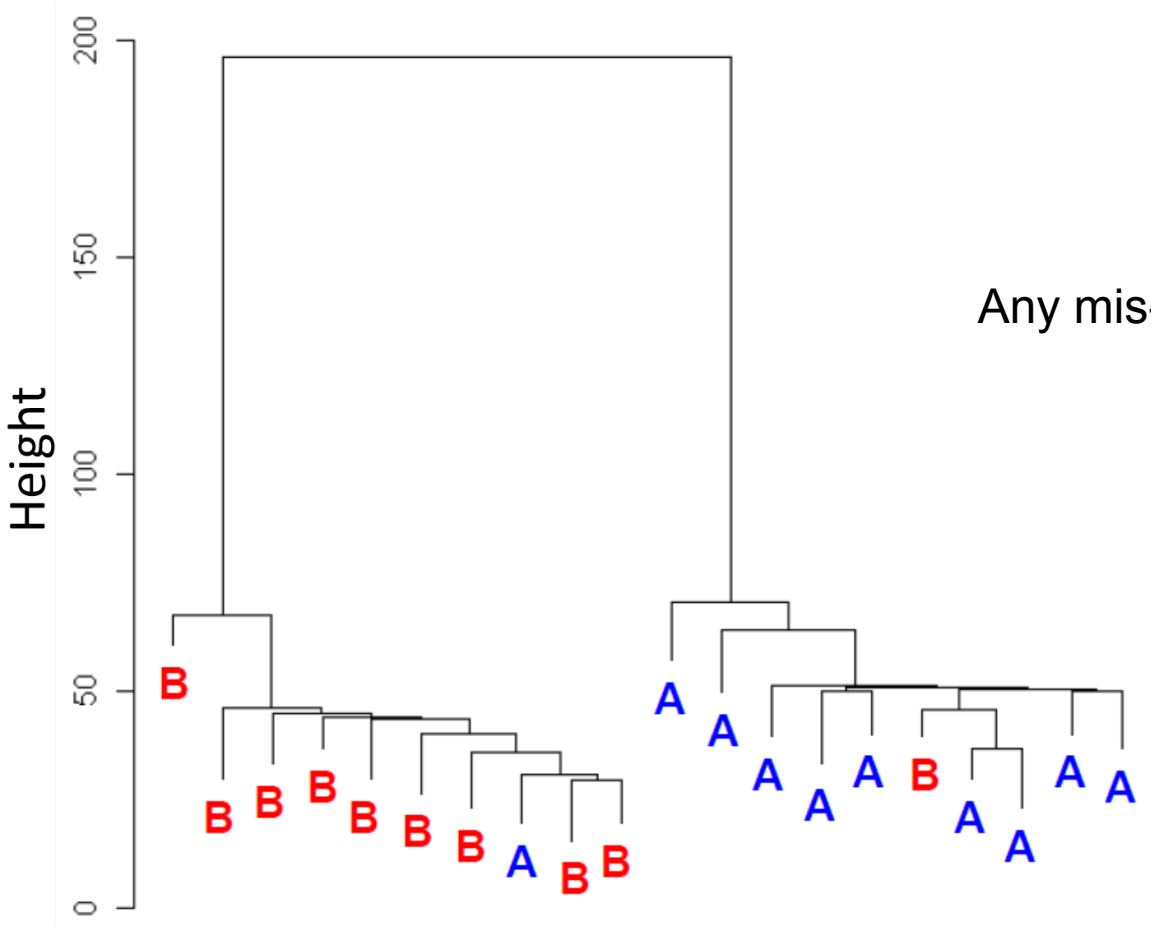




Minor bumps/shifts can be corrected by normalization

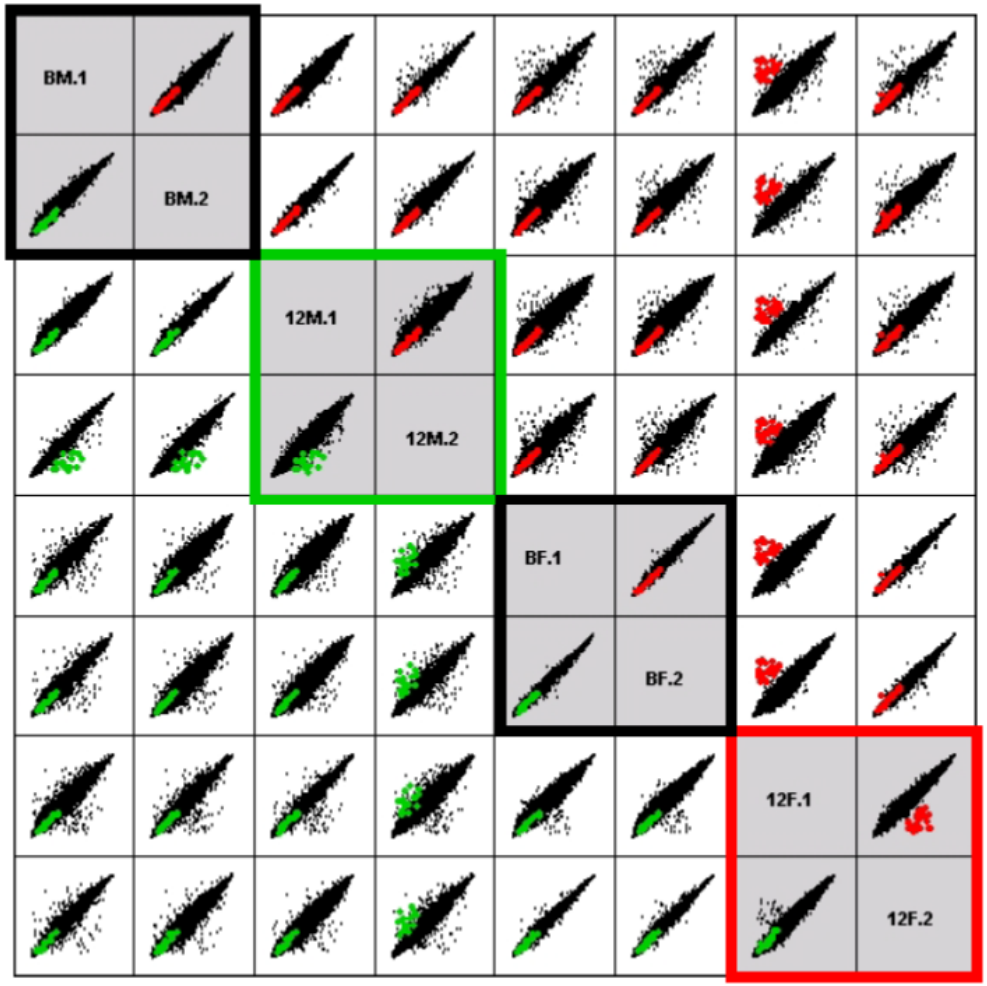
Source: Shockley

QC: cluster analysis



Source: Shockley

QC: Scatterplot



Muscle contraction,
Muscle development,
Hypoxia

Digestion of carbohydrates and proteins

Shockley and Churchill, 2006

Demos

Demos

- sklearn_toy.py
- c4432_1.py
- one_hot_simple.py
- boston_lin_reg.py
- geron1.py
- geron2.py
- best_corr_heatmap.py

- dendrogram_demo.py
- clustering_demo.py
- pca_demo.py