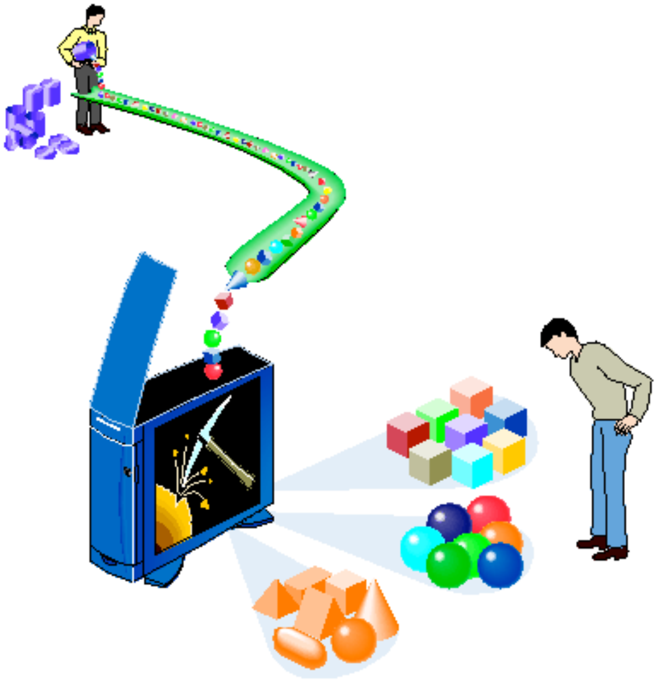
Supervised and Unsupervised Learning

Ciro Donalek

Ay/Bi 199 – April 2011

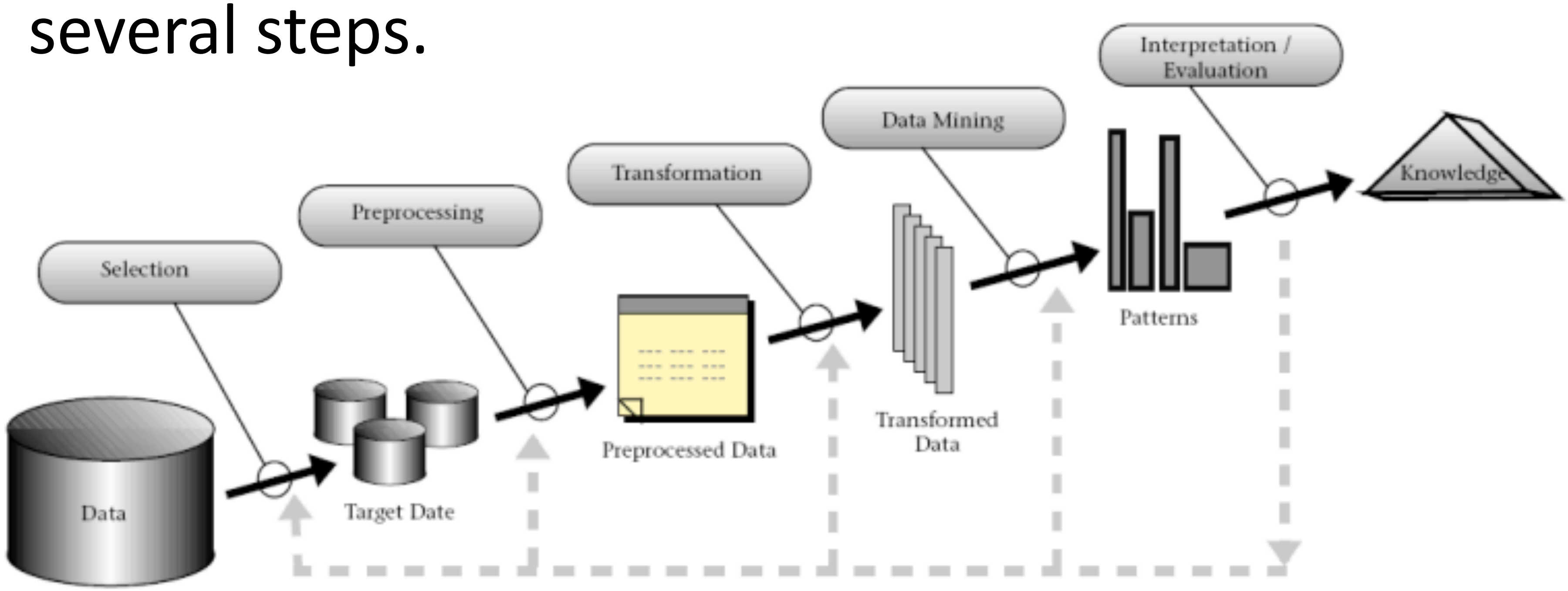
Summary

- KDD and Data Mining Tasks
- Finding the optimal approach
- Supervised Models
  - Neural Networks
  - Multi Layer Perceptron
  - Decision Trees
- Unsupervised Models
  - Different Types of Clustering
  - Distances and Normalization
  - Kmeans
  - Self Organizing Maps
- Combining different models
  - Committee Machines
  - Introducing a Priori Knowledge
  - Sleeping Expert Framework



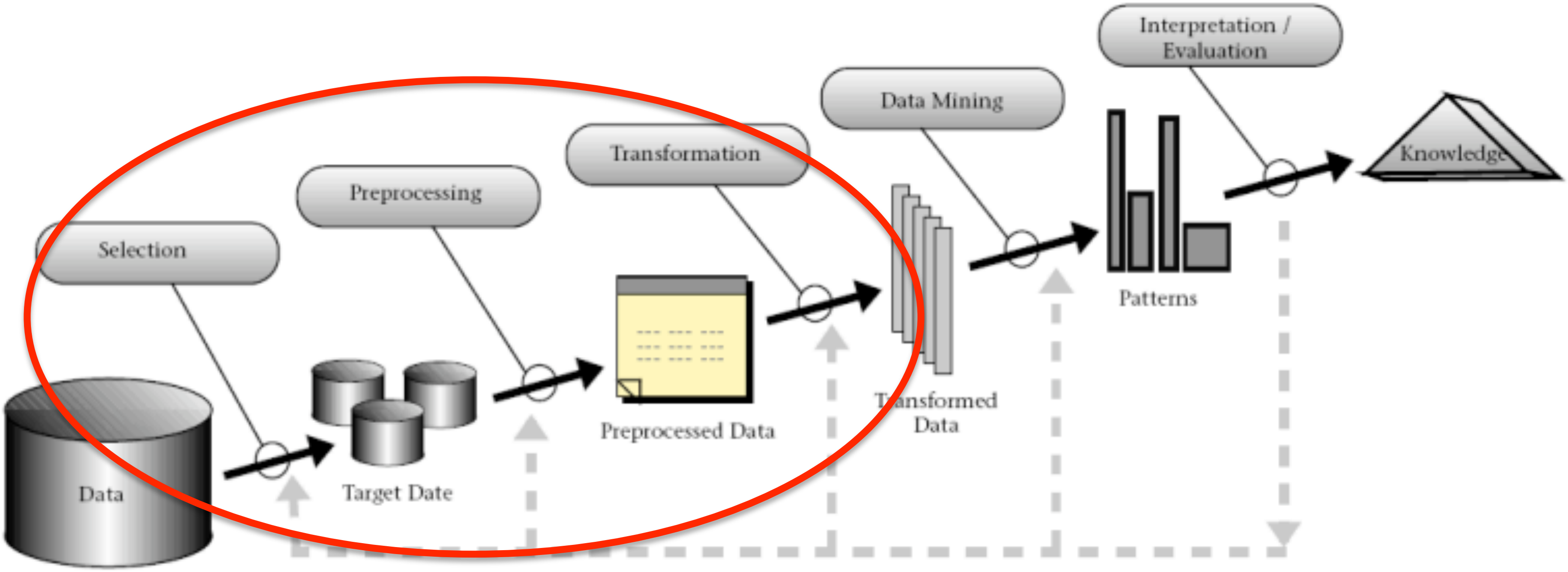
Knowledge Discovery in Databases

- KDD may be defined as: "*The non trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data*".
- KDD is an interactive and iterative process involving several steps.



You got your data: what’s next?

What kind of analysis do you need? Which model is more appropriate for it? …



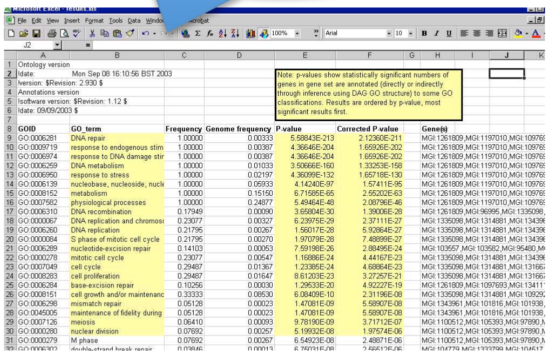
Selection
Preprocessing
Transformation
Data Mining
Interpretation / Evaluation
Data
Target Date
Preprocessed Data
Transformed Data
Patterns
Knowledge

Clean your data!

- Data preprocessing transforms the raw data into a format that will be more easily and effectively processed for the purpose of the user.

- Some tasks
  - *sampling*: selects a representative subset from a large population of data;
  - *Noise treatment*
  - strategies to handle ***missing data****:* sometimes your rows will be incomplete, not all parameters are measured for all samples.
  - *normalization*
  - *feature extraction*: pulls out specified data that is significant in some particular context.

Use standard formats!


Microsoft Excel - results.xls
File Edit View Insert Format Tools Data Windo Acrobat
100%
Arial
10
J2
=

Ontology version
!date: Mon Sep 08 16:10:56 BST 2003
!version: $Revision: 2.930 $
Annotations version
!software version: $Revision: 1.12 $
!date: 09/09/2003 $

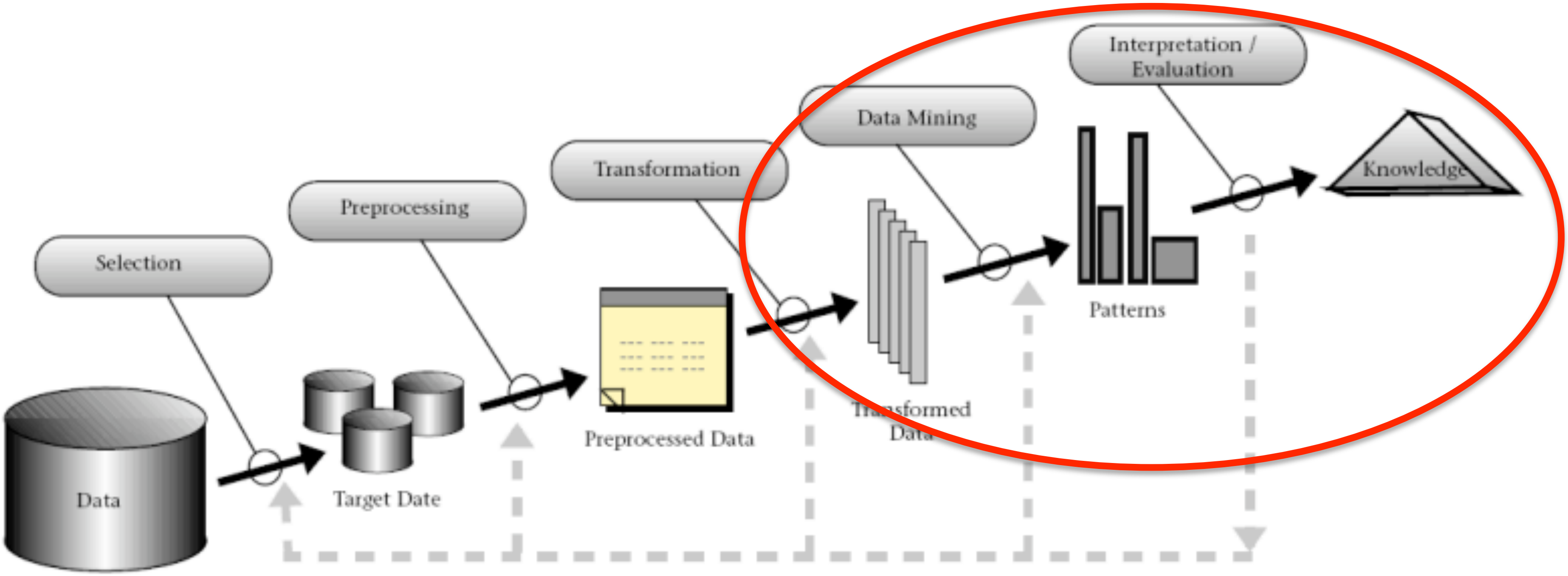
Note: p-values show statistically significant numbers of genes in gene set are annotated (directly or indirectly through inference using DAG GO structure) to some GO classifications. Results are ordered by p-value, most significant results first.

GOID	GO_term	Frequency	Genome frequency	P-value	Corrected P-value		Gene(s)
GO:0006281	DNA repair	1.00000	0.00333	5.58843E-213	2.12360E-211		MGI:1261809,MGI:1197010,MGI:10976
GO:0009719	response to endogenous stim	1.00000	0.00387	4.36646E-204	1.65926E-202		MGI:1261809,MGI:1197010,MGI:10976
GO:0006974	response to DNA damage stir	1.00000	0.00387	4.36646E-204	1.65926E-202		MGI:1261809,MGI:1197010,MGI:10976
GO:0006259	DNA metabolism	1.00000	0.01033	3.50666E-160	1.33253E-158		MGI:1261809,MGI:1197010,MGI:10976
GO:0006950	response to stress	1.00000	0.02197	4.36099E-132	1.65718E-130		MGI:1261809,MGI:1197010,MGI:10976
GO:0006139	nucleobase, nucleoside, nucle	1.00000	0.05933	4.14240E-97	1.57411E-95		MGI:1261809,MGI:1197010,MGI:10976
GO:0008152	metabolism	1.00000	0.15150	6.71585E-65	2.55202E-63		MGI:1261809,MGI:1197010,MGI:10976
GO:0007582	physiological processes	1.00000	0.24877	5.49464E-48	2.08796E-46		MGI:1261809,MGI:1197010,MGI:10976
GO:0006310	DNA recombination	0.17949	0.00090	3.65804E-30	1.39006E-28		MGI:1261809,MGI:96995,MGI:1335098,
GO:0000067	DNA replication and chromoso	0.23077	0.00327	6.23975E-29	2.37111E-27		MGI:1335098,MGI:1314881,MGI:13439
GO:0006260	DNA replication	0.21795	0.00267	1.56017E-28	5.92864E-27		MGI:1335098,MGI:1314881,MGI:13439
GO:0000084	S phase of mitotic cell cycle	0.21795	0.00270	1.97079E-28	7.48899E-27		MGI:1335098,MGI:1314881,MGI:13439
GO:0006289	nucleotide-excision repair	0.14103	0.00053	7.59198E-26	2.88495E-24		MGI:103557,MGI:103582,MGI:95480,M
GO:0000278	mitotic cell cycle	0.23077	0.00547	1.16886E-24	4.44167E-23		MGI:1335098,MGI:1314881,MGI:13439
GO:0007049	cell cycle	0.29487	0.01367	1.23385E-24	4.68864E-23		MGI:1335098,MGI:1314881,MGI:13166
GO:0008283	cell proliferation	0.29487	0.01647	8.61203E-23	3.27257E-21		MGI:1335098,MGI:1314881,MGI:13166
GO:0006284	base-excision repair	0.10256	0.00030	1.29533E-20	4.92227E-19		MGI:1261809,MGI:1097693,MGI:13411
GO:0008151	cell growth and/or maintenanc	0.33333	0.08530	6.08409E-10	2.31196E-08		MGI:1335098,MGI:1314881,MGI:10929
GO:0006298	mismatch repair	0.05128	0.00023	1.47081E-09	5.58907E-08		MGI:1343961,MGI:101816,MGI:101938,
GO:0045005	maintenance of fidelity during	0.05128	0.00023	1.47081E-09	5.58907E-08		MGI:1343961,MGI:101816,MGI:101938,
GO:0007126	meiosis	0.06410	0.00093	9.78190E-09	3.71712E-07		MGI:1100512,MGI:105393,MGI:97890,M
GO:0000280	nuclear division	0.07692	0.00257	5.19932E-08	1.97574E-06		MGI:1100512,MGI:105393,MGI:97890,M
GO:0000279	M phase	0.07692	0.00267	6.54923E-08	2.48871E-06		MGI:1100512,MGI:105393,MGI:97890,M
GO:0006302	double-strand break repair	0.03846	0.00013	6.75031E-08	2.56512E-06		MGI:104779,MGI:1333799,MGI:104517

Missing Data

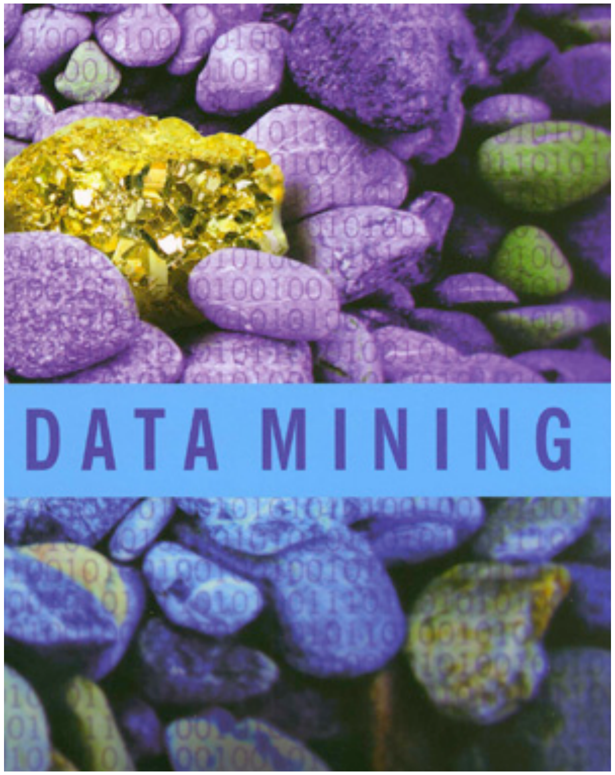
- Missing data are a part of almost all research, and we all have to decide how to deal with it.
- Complete Case Analysis: use only rows with all the values
- Available Case Analysis
- Substitution
  - Mean Value: replace the missing value with the mean value for that particular attribute
  - Regression Substitution: we can replace the missing value with historical value from similar cases
  - Matching Imputation: for each unit with a missing y, find a unit with similar values of x in the observed data and take its y value
  - Maximum Likelihood, EM, etc

- Some DM models can deal with missing data better than others.
- Which technique to adopt really depends on your data


Selection
Preprocessing
Transformation
Data Mining
Interpretation / Evaluation
Data
Target Date
Preprocessed Data
Transformed Data
Patterns
Knowledge

Data Mining

- Crucial task within the KDD
- Data Mining is about automating the process of searching for patterns in the data.
- More in details, the most relevant DM tasks are:
  - association
  - sequence or path analysis
  - **clustering**
  - **classification**
  - **regression**
  - visualization



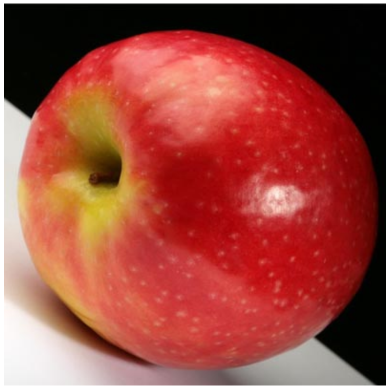
Finding Solution via Purposes

- You have your data, what kind of analysis do you need?

- Regression
  - predict new values based on the past, inference
  - compute the new values for a dependent variable based on the values of one or more measured attributes
- Classification:
  - divide samples in classes
  - use a trained set of previously labeled data
- Clustering
  - partitioning of a data set into subsets (clusters) so that data in each subset ideally share some common characteristics

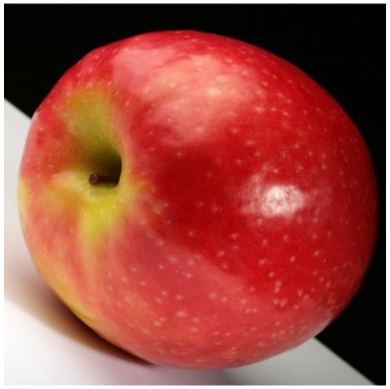
- Classification is in a some way similar to the clustering, but requires that the analyst know ahead of time how classes are defined.

Cluster Analysis

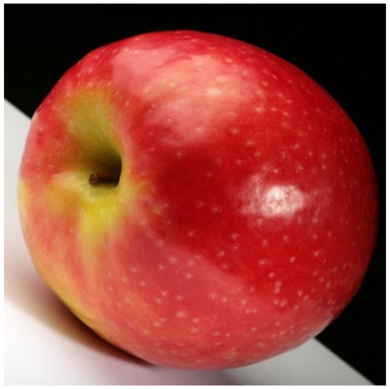


starshadow78/Flickr

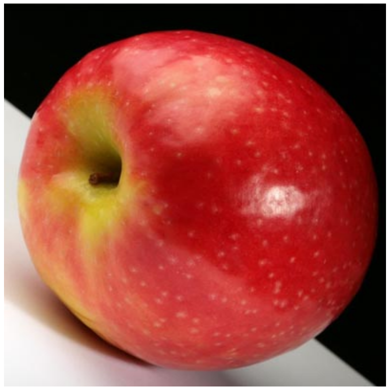
How many clusters do you expect?



starshadow78/Flickr

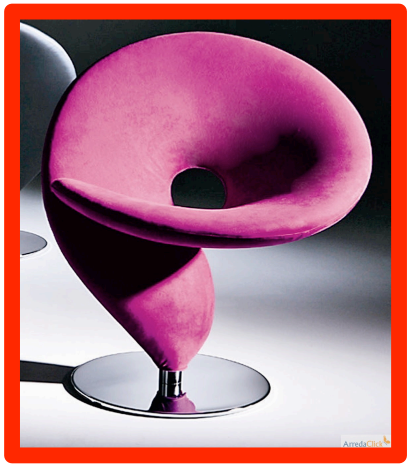


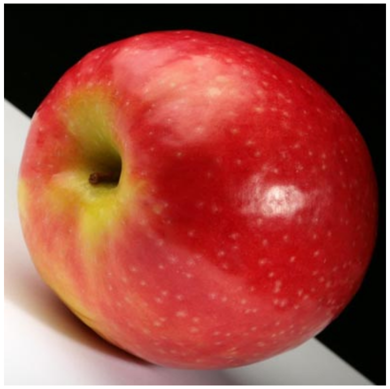
starshadow78/Flickr



starshadow78/Flickr

Search for Outliers





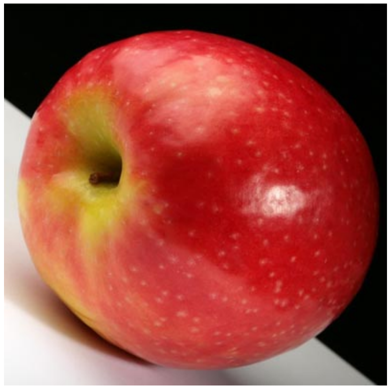
starshadow78/Flickr

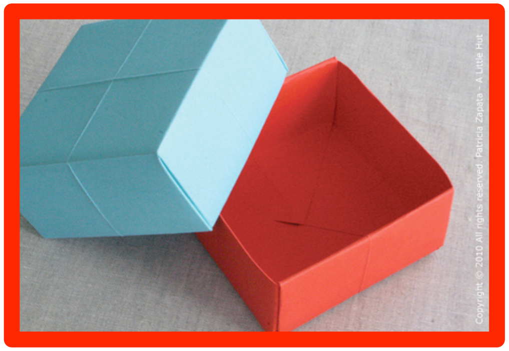
Classification

- Data mining technique used to predict group membership for data instances. There are two ways to assign a new value to a given class.

- **Crispy classification**
  - given an input, the classifier returns its label

- **Probabilistic classification**
  - given an input, the classifier returns its probabilities to belong to each class
  - useful when some mistakes can be more costly than others (give me only data >90%)
  - winner take all and other rules
    - assign the object to the class with the highest probability (WTA)
    - …but only if its probability is greater than 40% (WTA with thresholds)

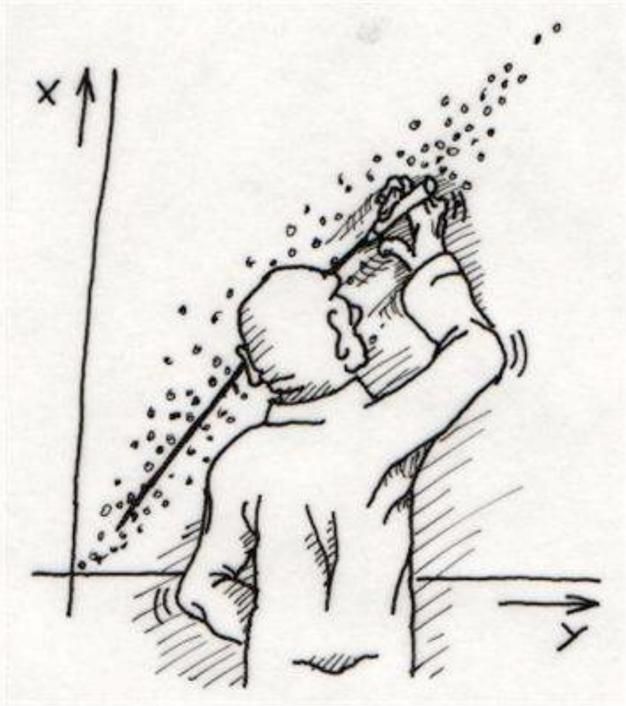



Copyright © 2010 All rights reserved. Patricia Zapata - A Little Hut

starshadow78/Flickr

Regression / Forecasting

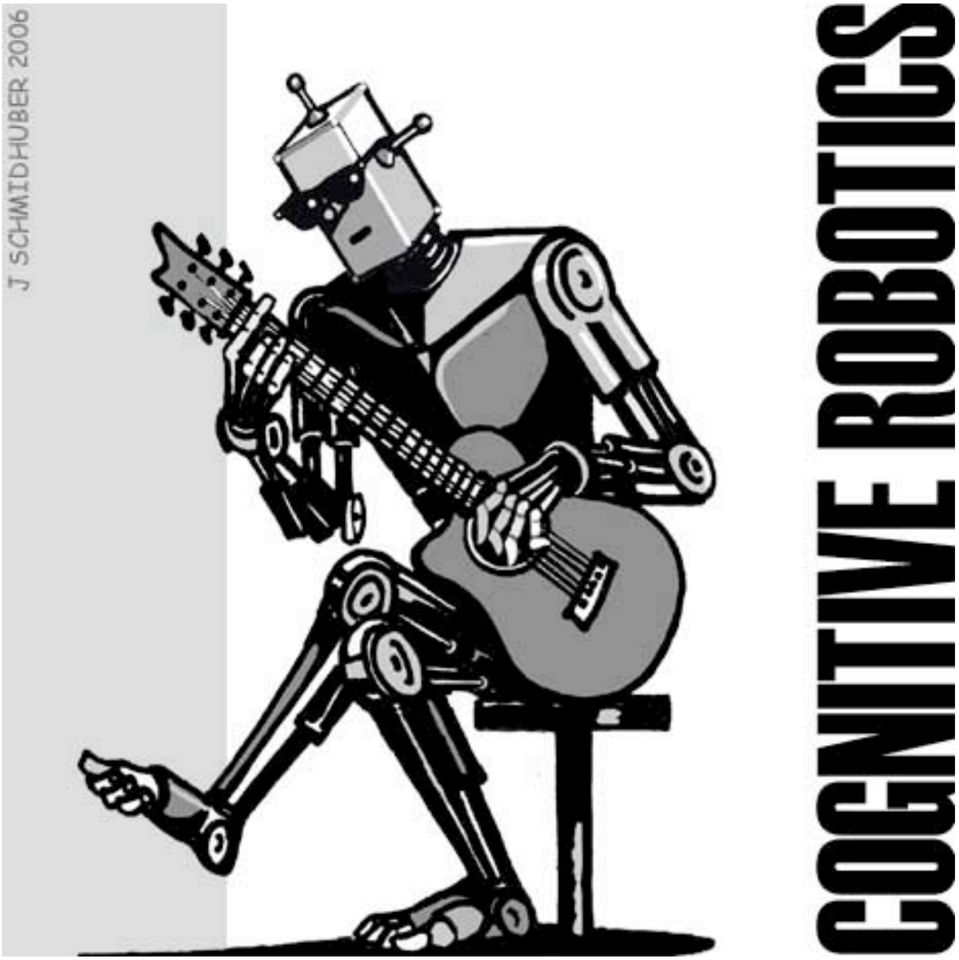
- Data table statistical correlation
  - – mapping without any prior assumption on the functional form of the data distribution;
  - – machine learning algorithms well suited for this.
- Curve fitting
  - – find a well defined and known function underlying your data;
  - – theory / expertise can help.



Machine Learning

- To learn: *to get knowledge of by study, experience, or being taught*.

- Types of Learning
  - Supervised
  - Unsupervised



Unsupervised Learning

- The model is not provided with the correct results during the training.
- Can be used to cluster the input data in classes on the basis of their statistical properties only.
- Cluster significance and labeling.
- The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.

Supervised Learning

- Training data includes both the input and the desired results.
- For some examples the correct results (targets) are known and are given in input to the model during the learning process.
- The construction of a proper training, validation and test set (Bok) is crucial.
- These methods are usually fast and accurate.
- Have to be able to **generalize**: give the correct results when new data are given in input without knowing a priori the target.

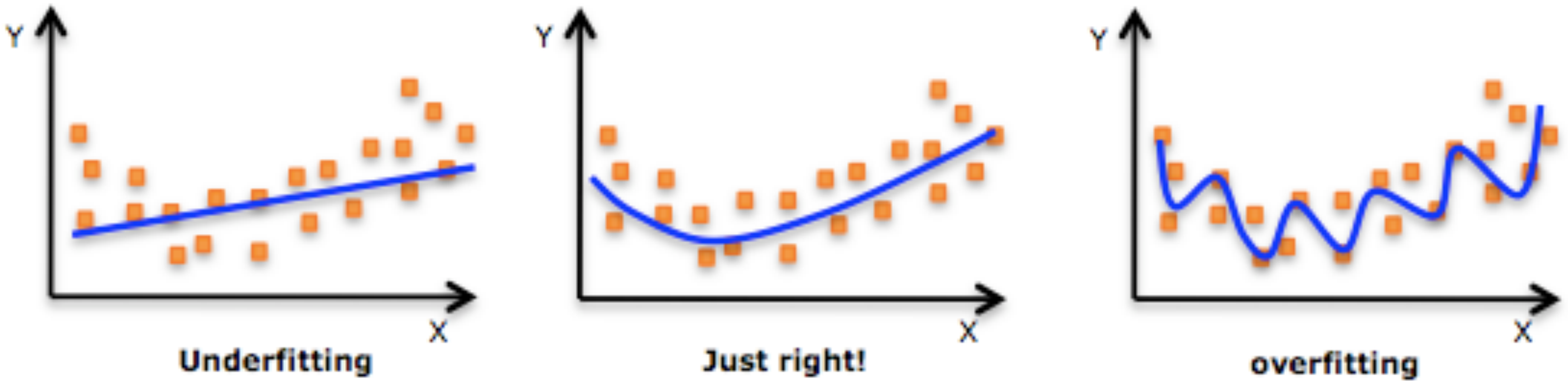
Generalization

- Refers to the ability to produce reasonable outputs for inputs not encountered during the training.

In other words: NO PANIC when "never seen before" data are given in input!

A common problem: OVERFITTING

- Learn the “data” and not the underlying function
- Performs well on the data used during the training and poorly with new data.



Underfitting

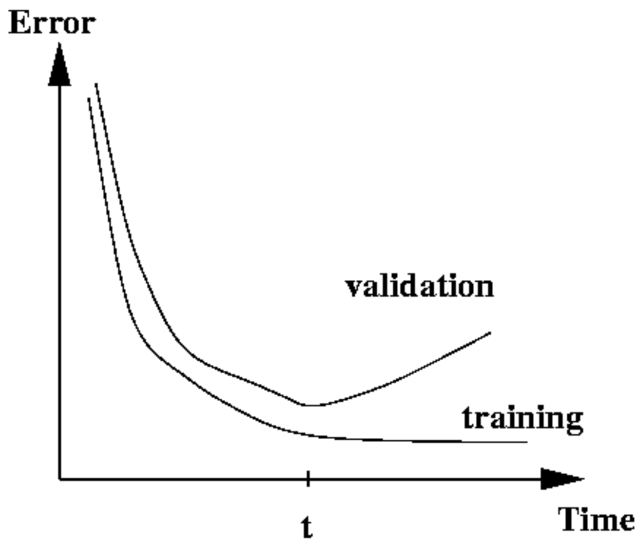
Just right!

overfitting

How to avoid it: use proper subsets, early stopping.

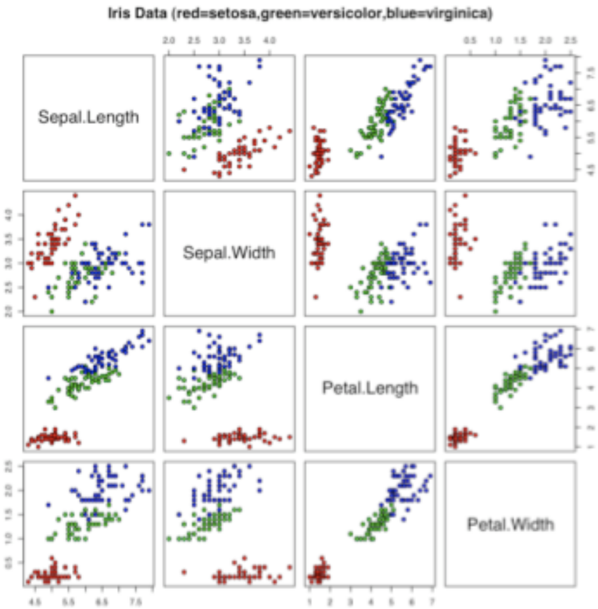
Datasets

- **Training set**: a set of examples used for learning, where the target value is known.
- **Validation set**: a set of examples used to tune the architecture of a classifier and estimate the error.
- **Test set**: used only to assess the performances of a classifier. It is never used during the training process so that the error on the test set provides an unbiased estimate of the generalization error.



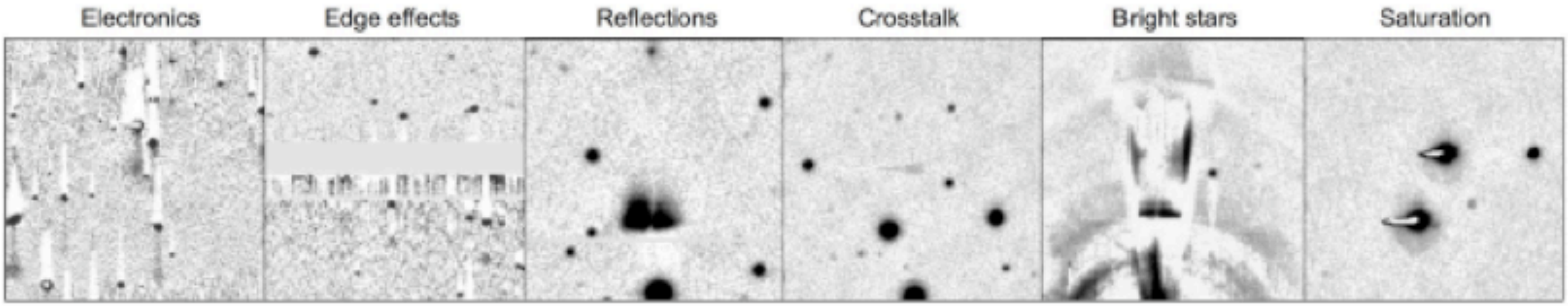
IRIS dataset

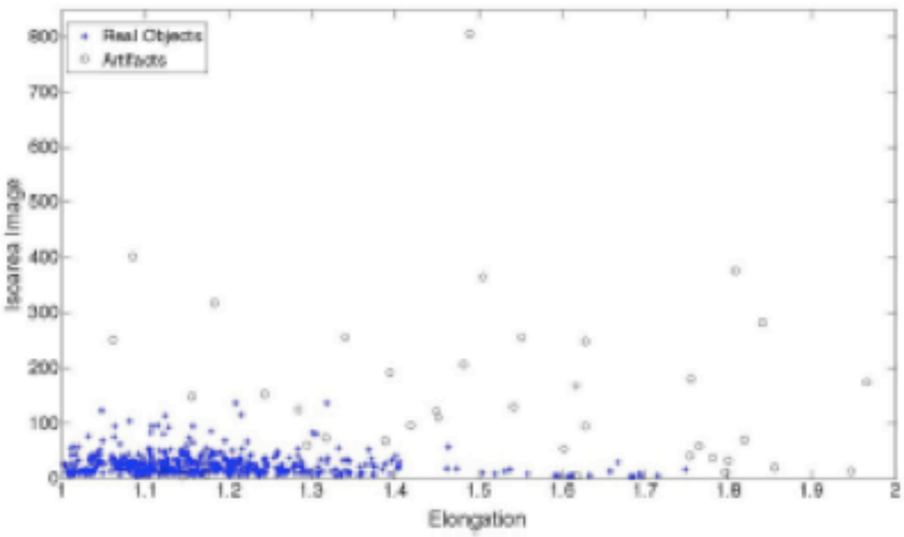
- IRIS
  - consists of 3 classes, 50 instances each
  - 4 numerical attributes (sepal and petal length and width in cm)
  - each class refers to a type of Iris plant (Setosa, Versicolor, Verginica)
  - the first class is linearly separable from the other two while the 2nd and the 3rd are not linearly separable

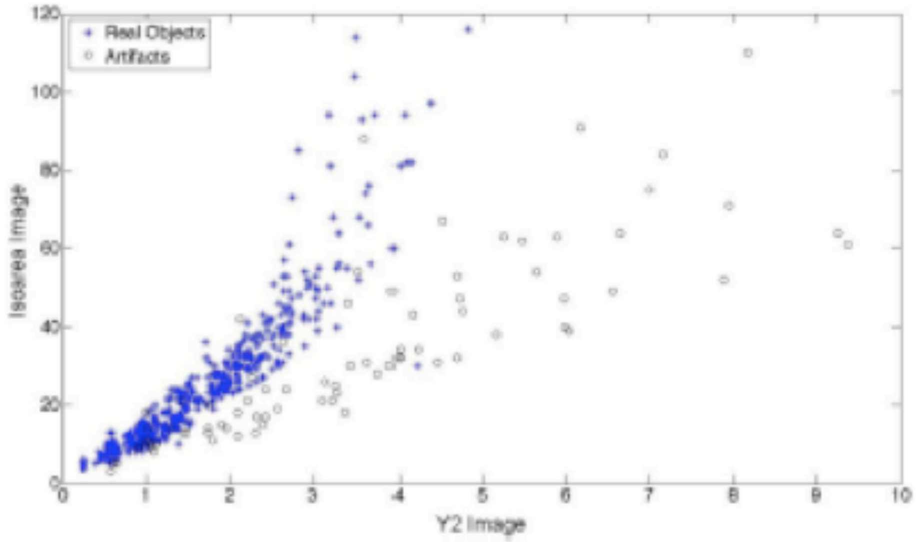


Artifacts Dataset

- PQ Artifacts
  - 2 main classes and 4 numerical attributes
  - classes are: true objects, artifacts







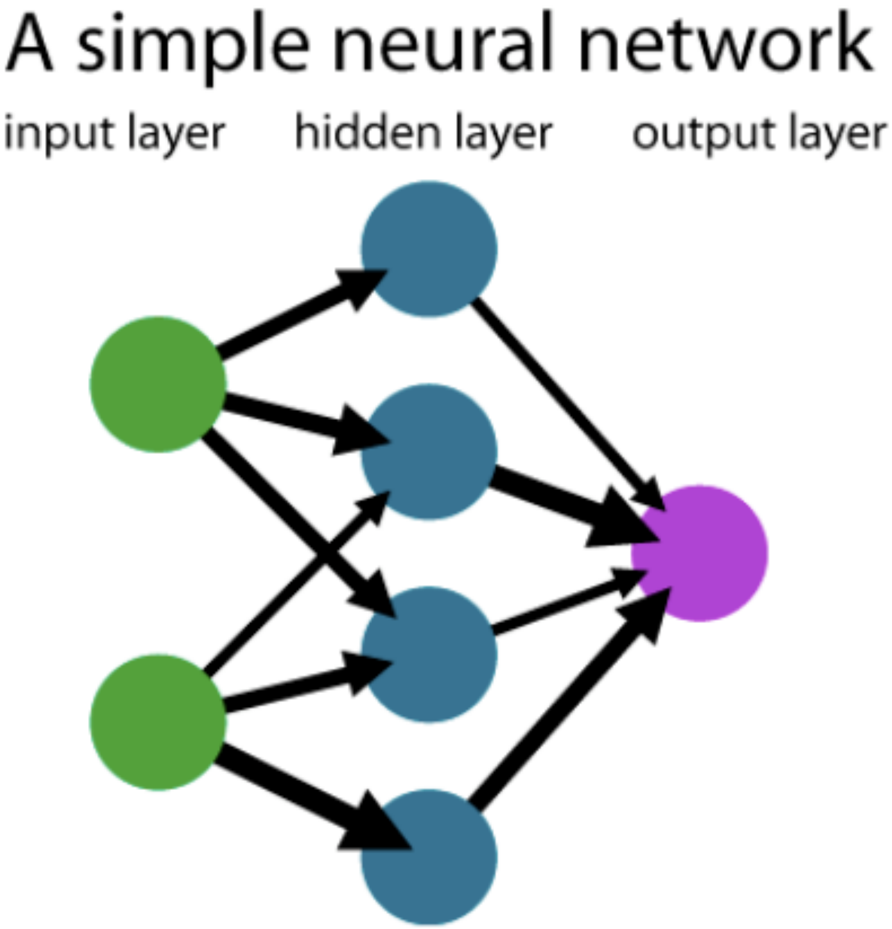
Data Selection

- "**Garbage in, garbage out** ": training, validation and test data must be representative of the underlying model
- All eventualities must be covered
- Unbalanced datasets
  - since the network minimizes the overall error, the proportion of types of data in the set is critical;
  - inclusion of a loss matrix (Bishop,1995);
  - often, the best approach is to ensure even representation of different cases, then to interpret the network's decisions accordingly.

Artificial Neural Network

An Artificial Neural Network is an information processing paradigm that is inspired by the way biological nervous systems process information:

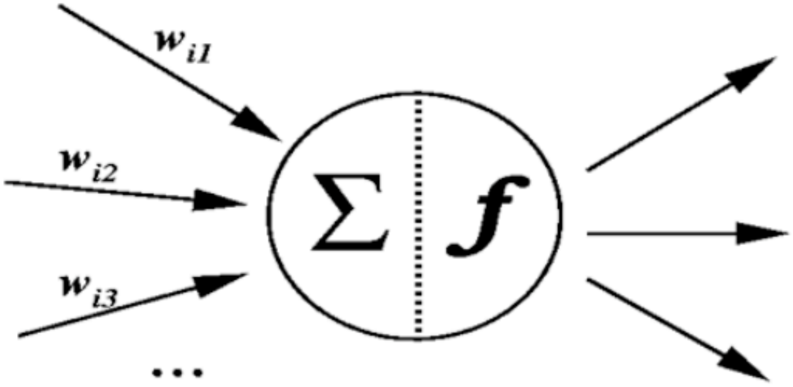
“a large number of highly interconnected simple processing elements (neurons) working together to solve specific problems”



A simple artificial neuron

- The basic computational element is often called a node or unit. It receives input from some other units, or from an external source.
- Each input has an associated weight **w**, which can be modified so as to model synaptic learning.
- The unit computes some function of the weighted sum of its inputs:

$$y_i = f(\sum_j w_{ij} y_j)$$



$$y_i = f(net_i)$$

Neural Networks

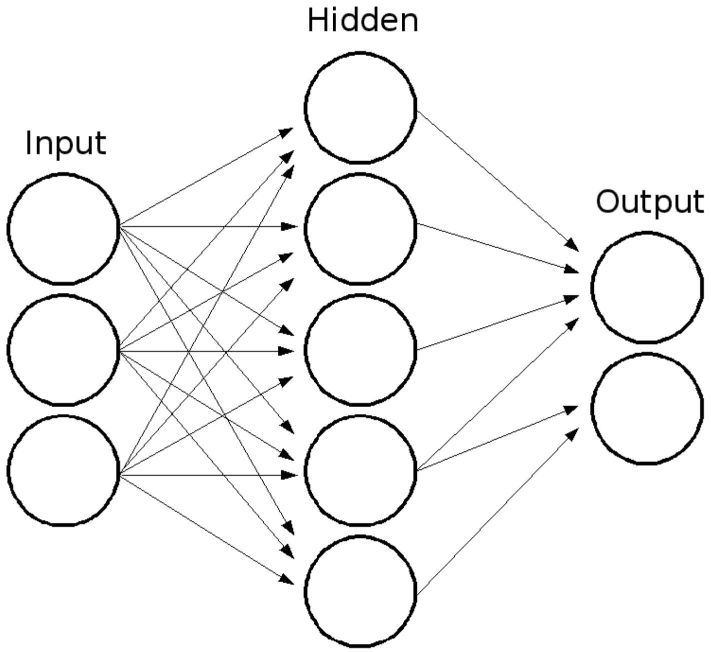
A Neural Network is usually structured into an input layer of neurons, one or more hidden layers and one output layer.

Neurons belonging to adjacent layers are usually fully connected and the various types and architectures are identified both by the different topologies adopted for the connections as well by the choice of the activation function.

The values of the functions associated with the connections are called "weights".

The whole game of using NNs is in the fact that, in order for the network to yield appropriate outputs for given inputs, the weight must be set to suitable values.

The way this is obtained allows a further distinction among modes of operations.



Neural Networks: types

Feedforward: Single Layer Perceptron, MLP, ADALINE (Adaptive Linear Neuron), RBF

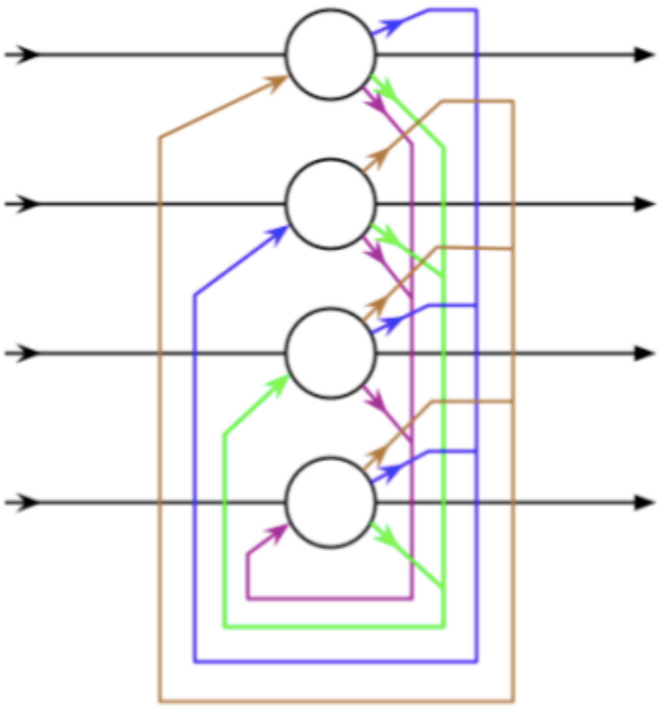
Self-Organized: SOM (Kohonen Maps)

Recurrent: Simple Recurrent Network, Hopfield Network.

Stochastic: Boltzmann machines, RBM.

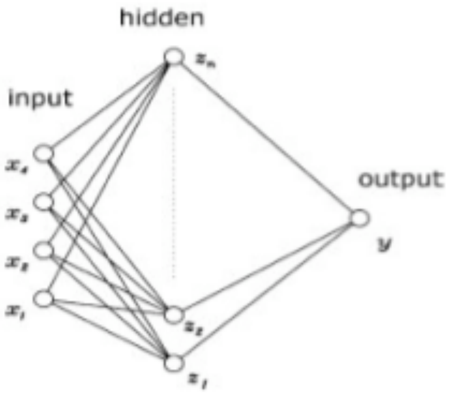
Modular: Committee of Machines, ASNN (Associative Neural Networks), Ensembles.

Others: Instantaneously Trained, Spiking (SNN), Dynamic, Cascades, NeuroFuzzy, PPS, GTM.

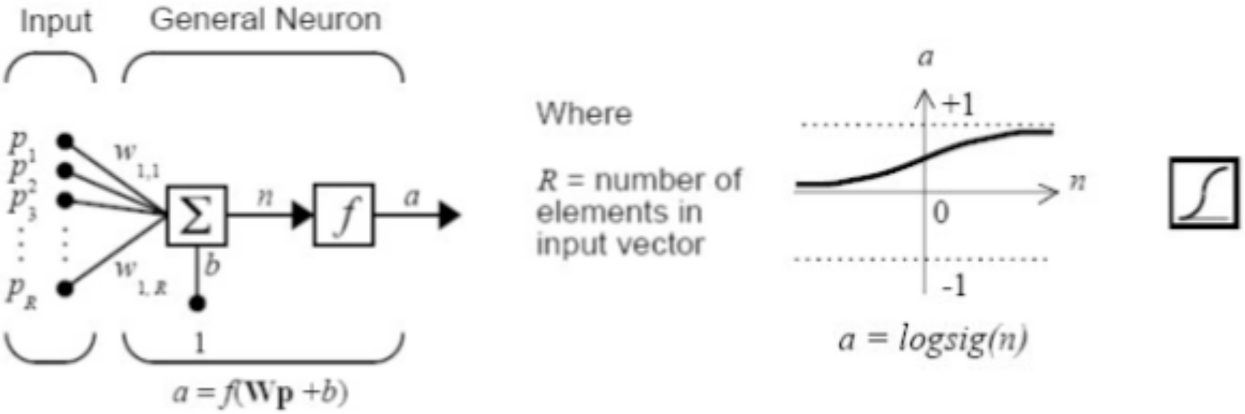


Multi Layer Perceptron

- The MLP is one of the most used supervised model: it consists of multiple layers of computational units, usually interconnected in a feed-forward way.
- Each neuron in one layer has direct connections to all the neurons of the subsequent layer.



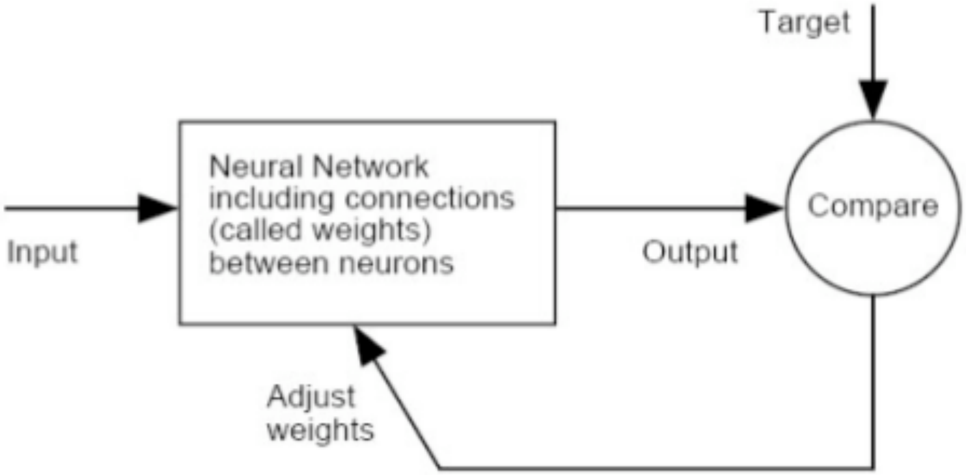
The architecture of a two layer MLP.



Learning Process

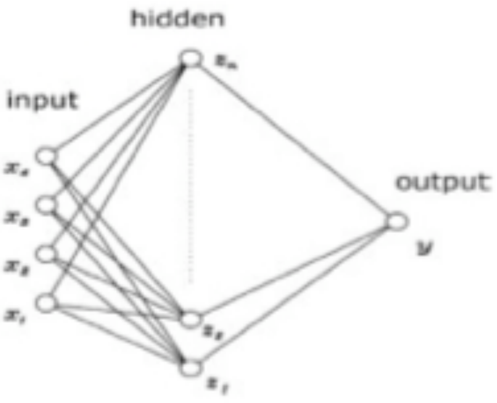
- Back Propagation
  - the output values are compared with the target to compute the value of some predefined error function
  - the error is then fedback through the network
  - using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function

After repeating this process for a sufficiently large number of training cycles, the network will usually converge.


Target
Neural Network including connections (called weights) between neurons
Input
Output
Compare
Adjust weights

Hidden Units

- The best number of hidden units depend on:
  - number of inputs and outputs
  - number of training case
  - the amount of noise in the targets
  - the complexity of the function to be learned
  - the activation function

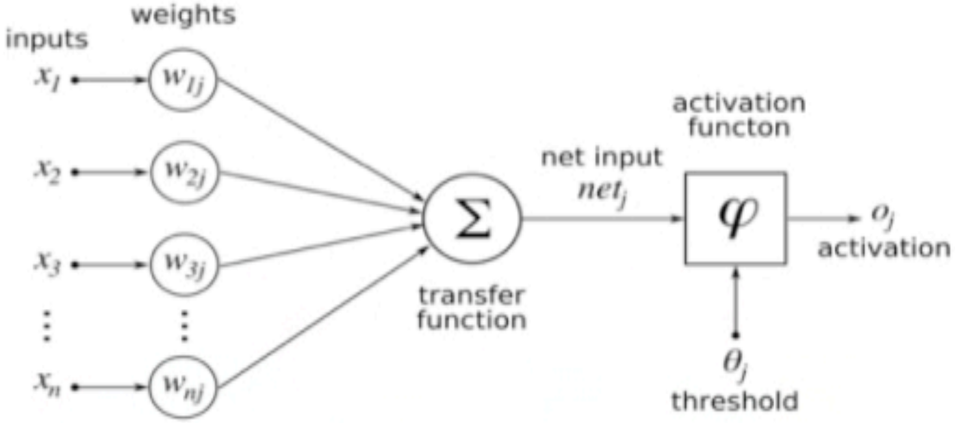


The architecture of a two layer MLP.

- Too few hidden units => high training and generalization error, due to underfitting and high statistical bias.
- Too many hidden units => low training error but high generalization error, due to overfitting and high variance.
- Rules of thumb don't usually work.

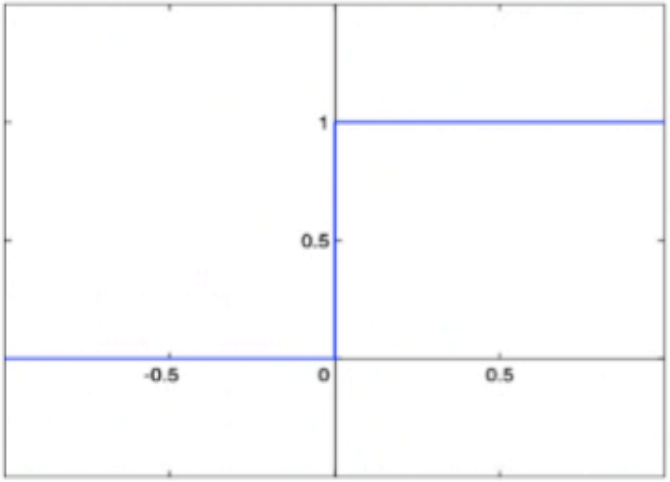
Activation and Error Functions

- **Error Functions**
  - measure of the discrepancy between the network output values and the target;
  - sum of the squared errors (SSE), cross entropy (CE), etc.
- **Activation Functions**
  - used by most units to transform their inputs;
  - needed to introduce non linearity into the network
  - linear, logistic, tanh, softmax...



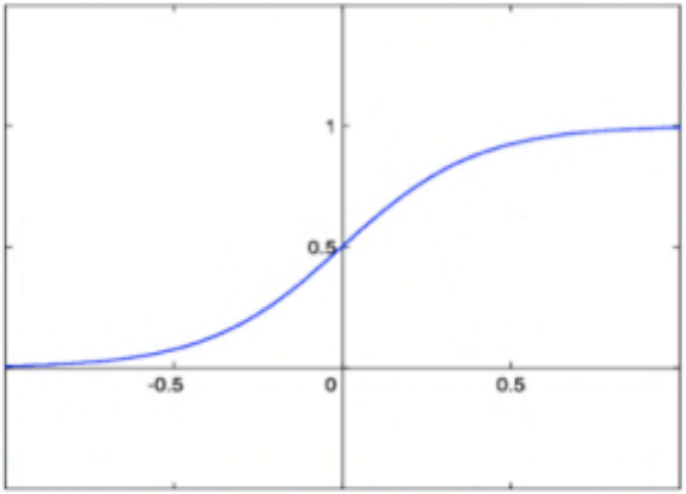
Using a Multilayer Perceptron with a softmax activation function and cross-entropy error, the network outputs can be interpreted as the conditional probabilities $p(C_1|\mathbf{x})$ and $p(C_2|\mathbf{x})$ where **x** is the input vector, C_1 the first class, C_2 the second class.

Activation Functions



Step function
The output is a certain value A1, if the input sum is above a certain threshold and A0 if the input sum is below a certain threshold.

When we want to classify an input pattern into one of two groups, we can use a binary classifier with a step activation function.



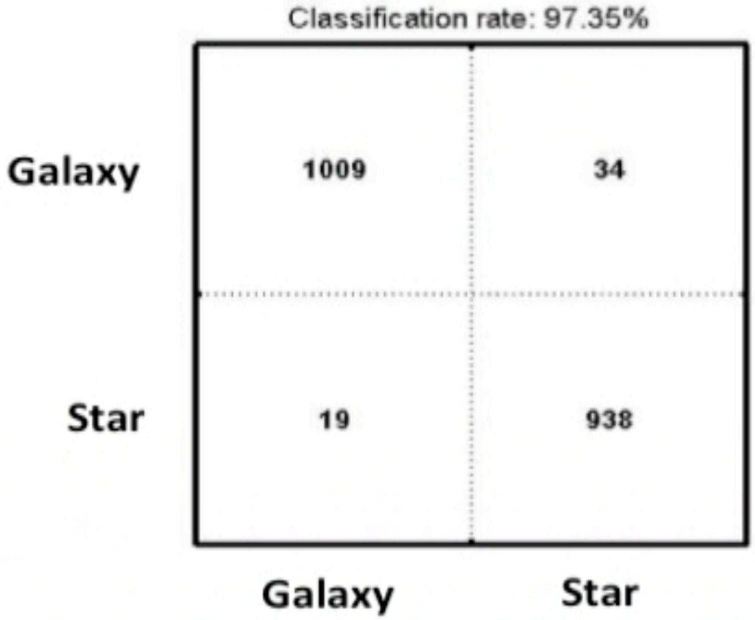
Sigmoid function
Has the property of being similar to the step function, but with the addition of a region of uncertainty.

Sigmoid functions in this respect are very similar to the input-output relationships of biological neurons.

Results: confusion matrix

In the confusion matrix the network prediction Y are compared with the target T: the rows represent the true classes and the columns the predicted classes.

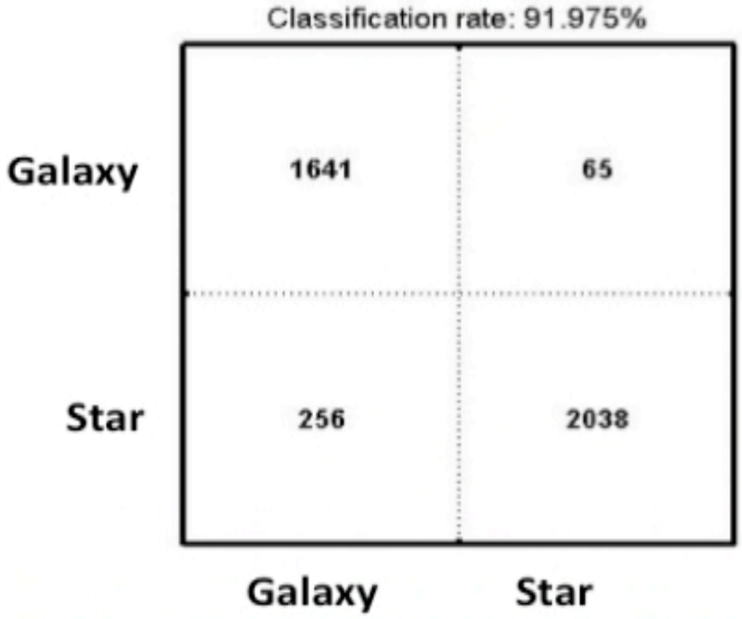
Training set



Classification rate: 97.35%

	Galaxy	Star
Galaxy	1009	34
Star	19	938

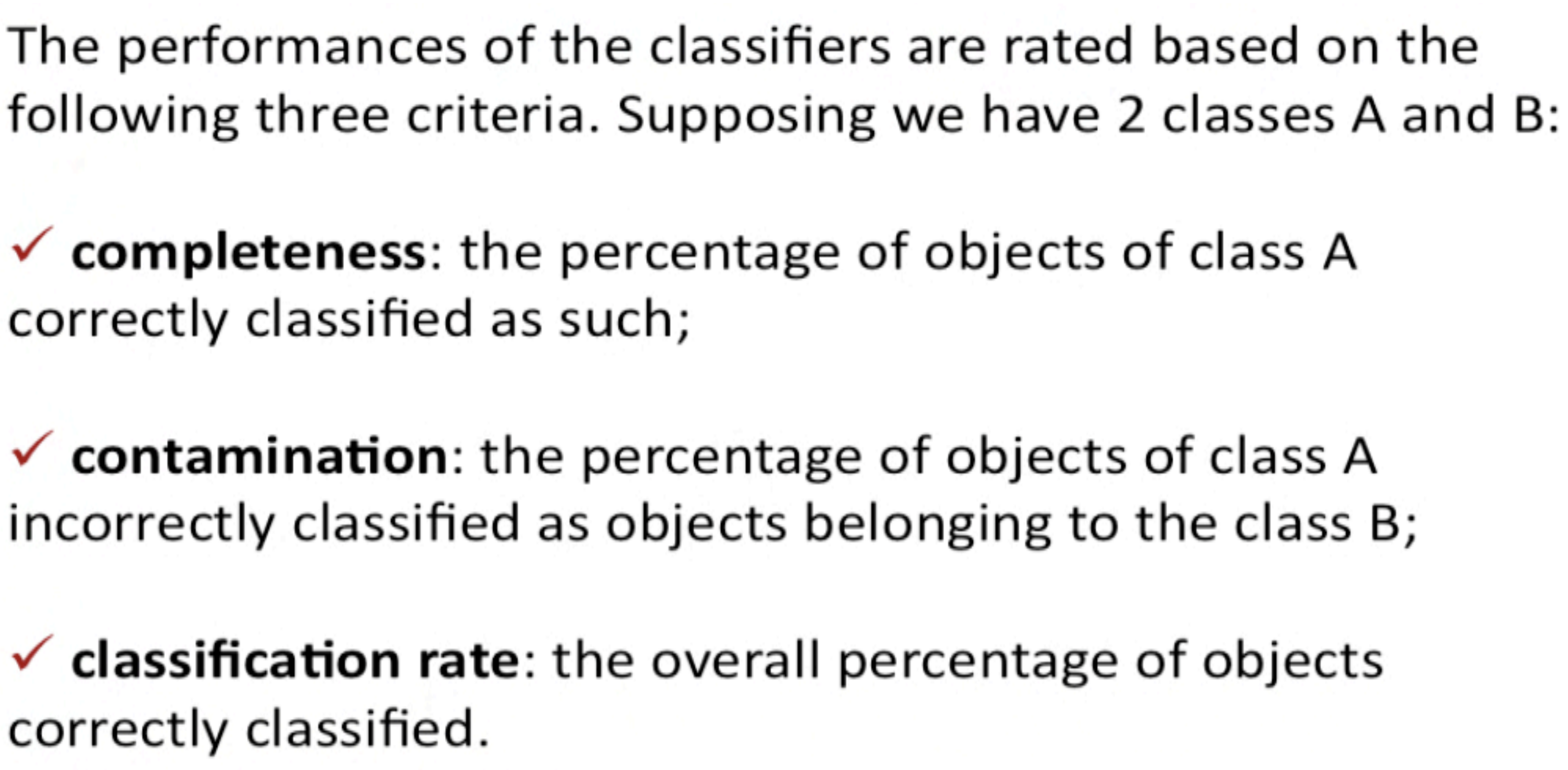
Test set



Classification rate: 91.975%

	Galaxy	Star
Galaxy	1641	65
Star	256	2038

Results: completeness and contamination



The performances of the classifiers are rated based on the following three criteria. Supposing we have 2 classes A and B:

✓ **completeness**: the percentage of objects of class A correctly classified as such;

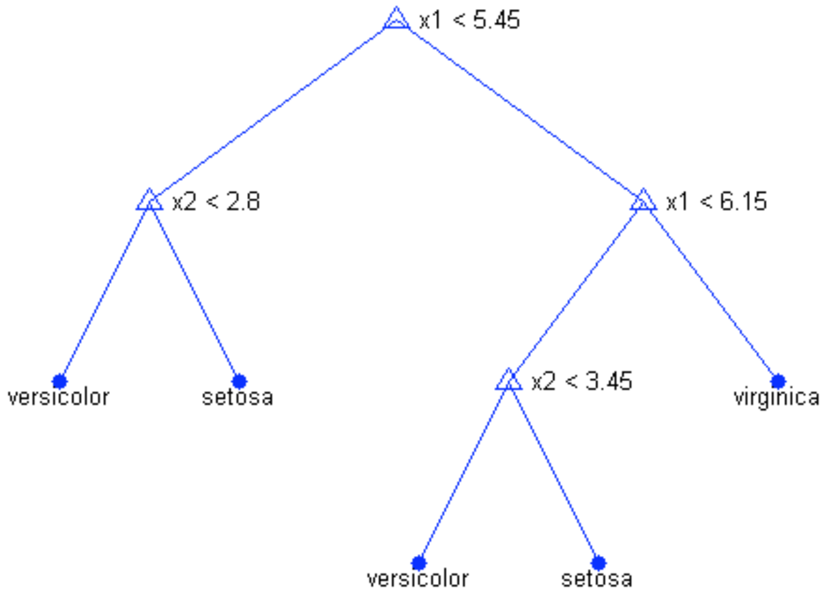
✓ **contamination**: the percentage of objects of class A incorrectly classified as objects belonging to the class B;

✓ **classification rate**: the overall percentage of objects correctly classified.

Exercise: compute completeness and contamination for the previous confusion matrix (test set)

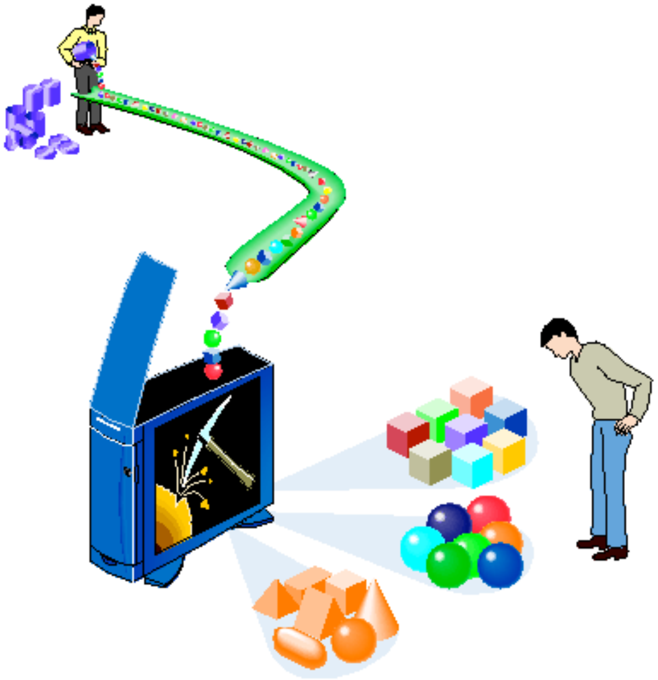
Decision Trees

- Is another classification method.
- A decision tree is a set of simple rules, such as "if the sepal length is less than 5.45, classify the specimen as setosa."
- Decision trees are also nonparametric because they do not require any assumptions about the distribution of the variables in each class.



Summary

- KDD and Data Mining Tasks
- Finding the optimal approach
- Supervised Models
  - Neural Networks
  - Multi Layer Perceptron
  - Decision Trees
- Unsupervised Models
  - Different Types of Clustering
  - Distances and Normalization
  - Kmeans
  - Self Organizing Maps
- Combining different models
  - Committee Machines
  - Introducing a Priori Knowledge
  - Sleeping Expert Framework

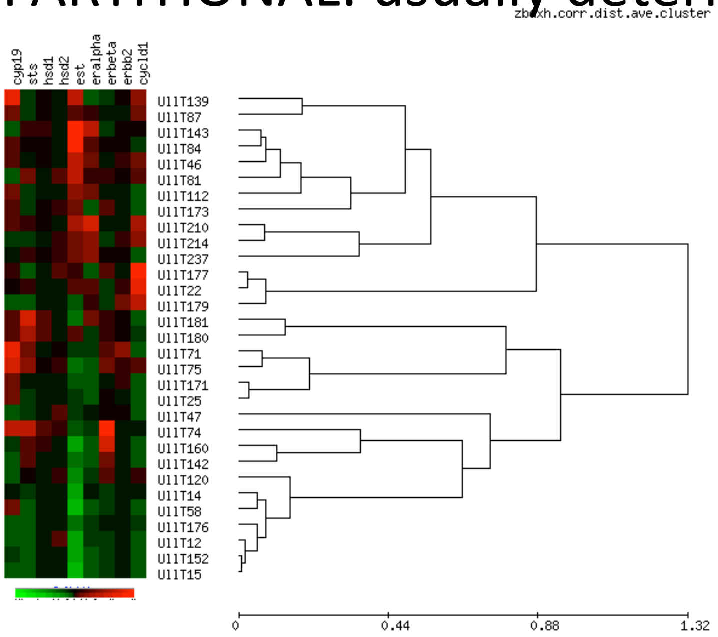


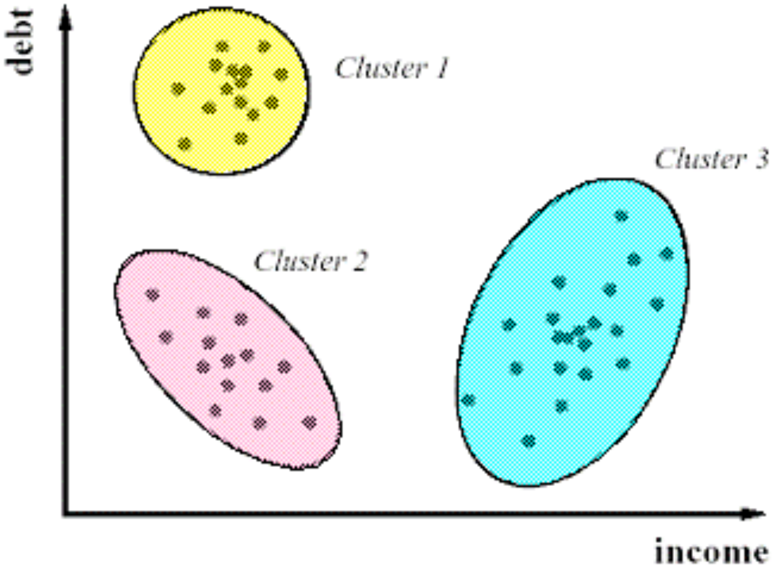
Unsupervised Learning

- The model is not provided with the correct results during the training.
- Can be used to cluster the input data in classes on the basis of their statistical properties only.
- Cluster significance and labeling.
- The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.

Types of Clustering

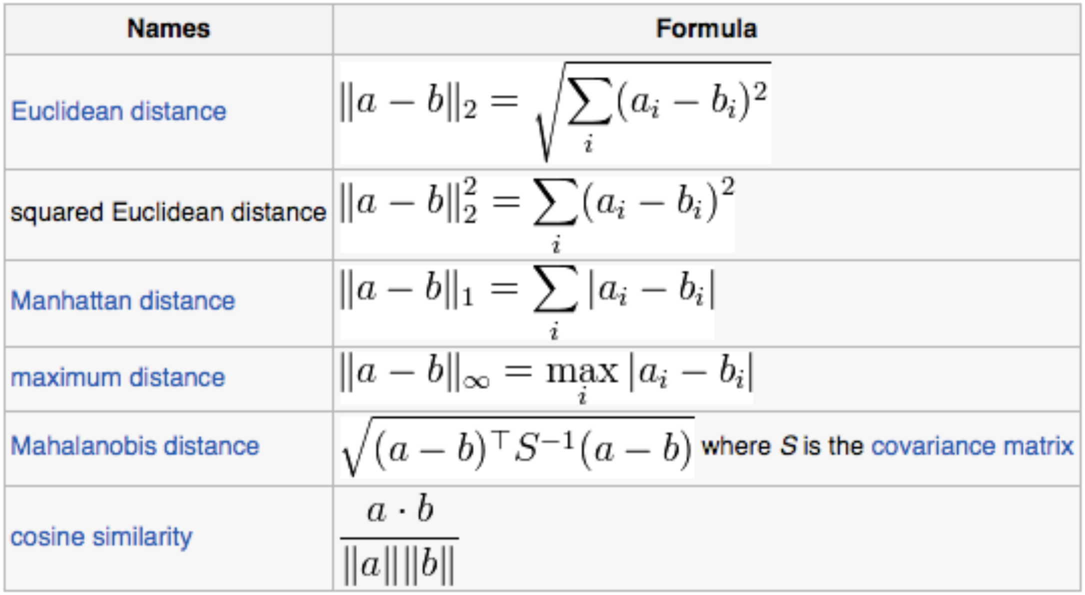
- Types of clustering:
  - HIERARCHICAL: finds successive clusters using previously established clusters
    - agglomerative (bottom-up): start with each element in a separate cluster and merge them accordingly to a given property
    - divisive (top-down)
  - PARTITIONAL: usually determines all clusters at once



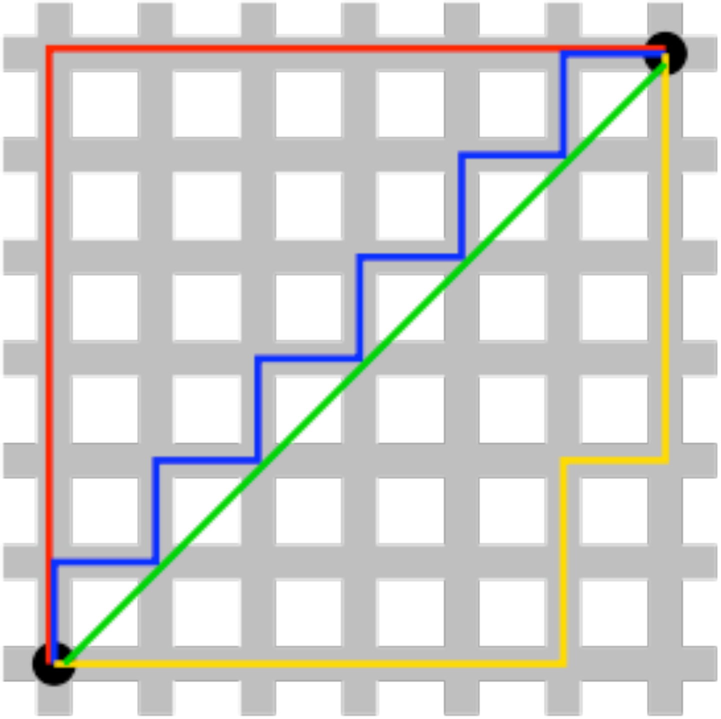


Distances

- Determine the similarity between two clusters and the shape of the clusters.



Names	Formula
Euclidean distance	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
squared Euclidean distance	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan distance	$\|a-b\|_1 = \sum_i \|a_i - b_i\|$
maximum distance	$\|a-b\|_\infty = \max_i \|a_i - b_i\|$
Mahalanobis distance	$\sqrt{(a-b)^\top S^{-1}(a-b)}$ where S is the covariance matrix
cosine similarity	$\frac{a \cdot b}{\|a\|\|b\|}$



In case of strings…

- The **Hamming distance** between two strings of equal length is the number of positions at which the corresponding symbols are different.
  - measures the minimum number of *substitutions* required to change one string into the other
- The **Levenshtein (edit) distance** is a metric for measuring the amount of difference between two sequences.
  - is defined as the minimum number of edits needed to transform one string into the other.

1001001

1000100

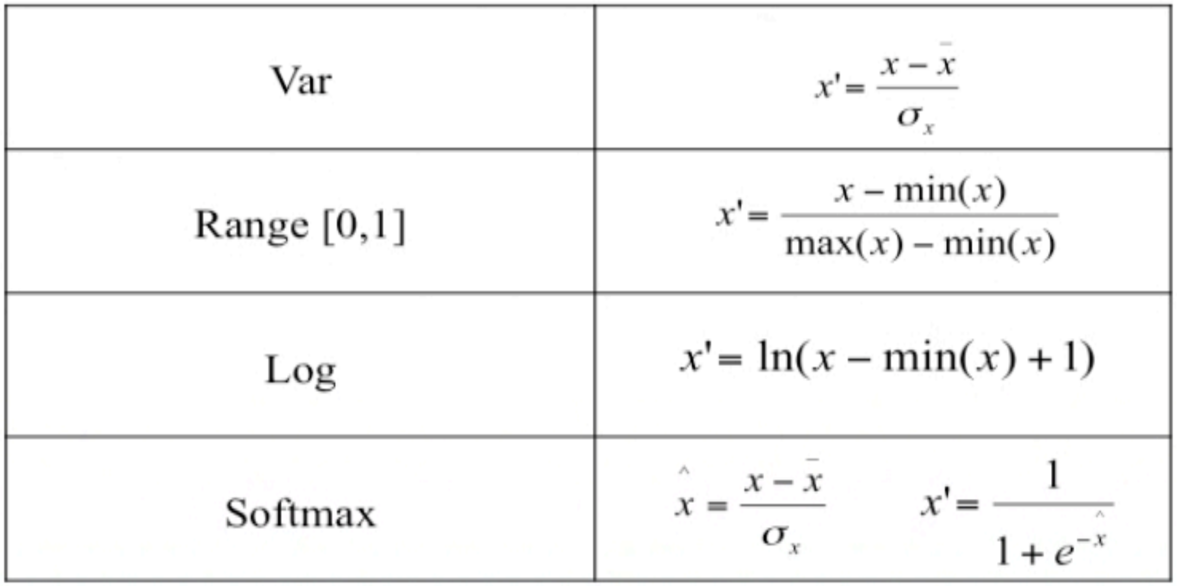
HD=3

LD(BIOLOGY, BIOLOGIA)=2

BIOLOG**Y** -> BIOLOG**I (substitution)**

BIOLOGI -> BIOLOGI**A (insertion)**

Normalization



Var	$x' = \frac{x - \bar{x}}{\sigma_x}$
Range [0,1]	$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$
Log	$x' = \ln(x - \min(x) + 1)$
Softmax	$\hat{x} = \frac{x - \bar{x}}{\sigma_x}$ $x' = \frac{1}{1 + e^{-\hat{x}}}$

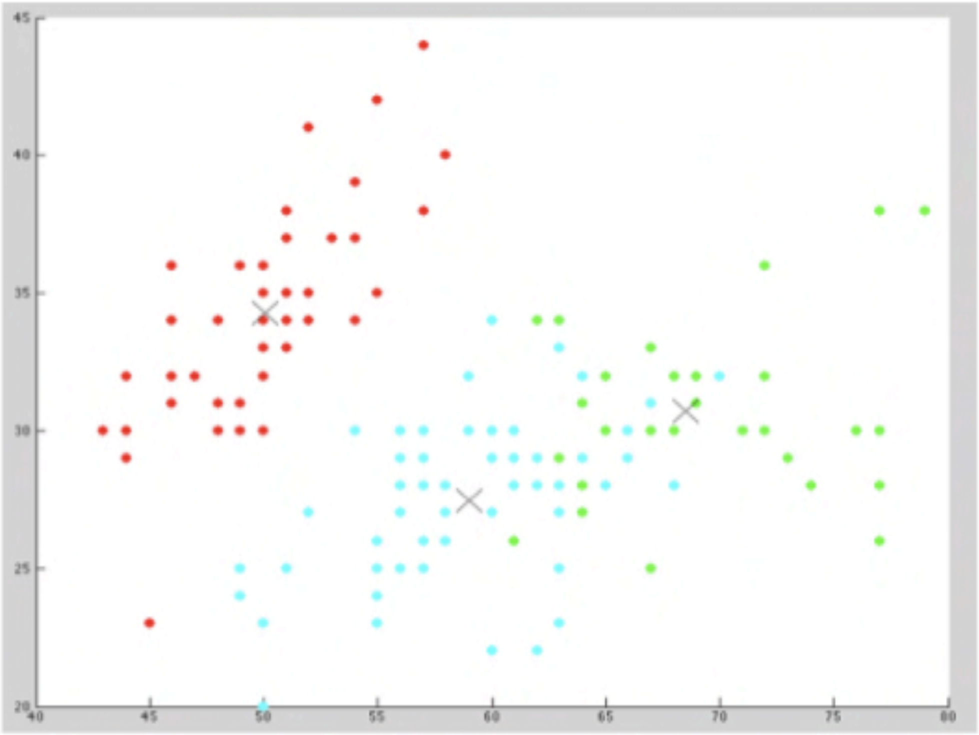
VAR: the mean of each attribute of the transformed set of data points is reduced to zero by subtracting the mean of each attribute from the values of the attributes and dividing the result by the standard deviation of the attribute.

RANGE (Min-Max Normalization): subtracts the minimum value of an attribute from each value of the attribute and then divides the difference by the range of the attribute. It has the advantage of preserving exactly all relationship in the data, without adding any bias.

SOFTMAX: is a way of reducing the influence of extreme values or outliers in the data without removing them from the data set. It is useful when you have outlier data that you wish to include in the data set while still preserving the significance of data within a standard deviation of the mean.

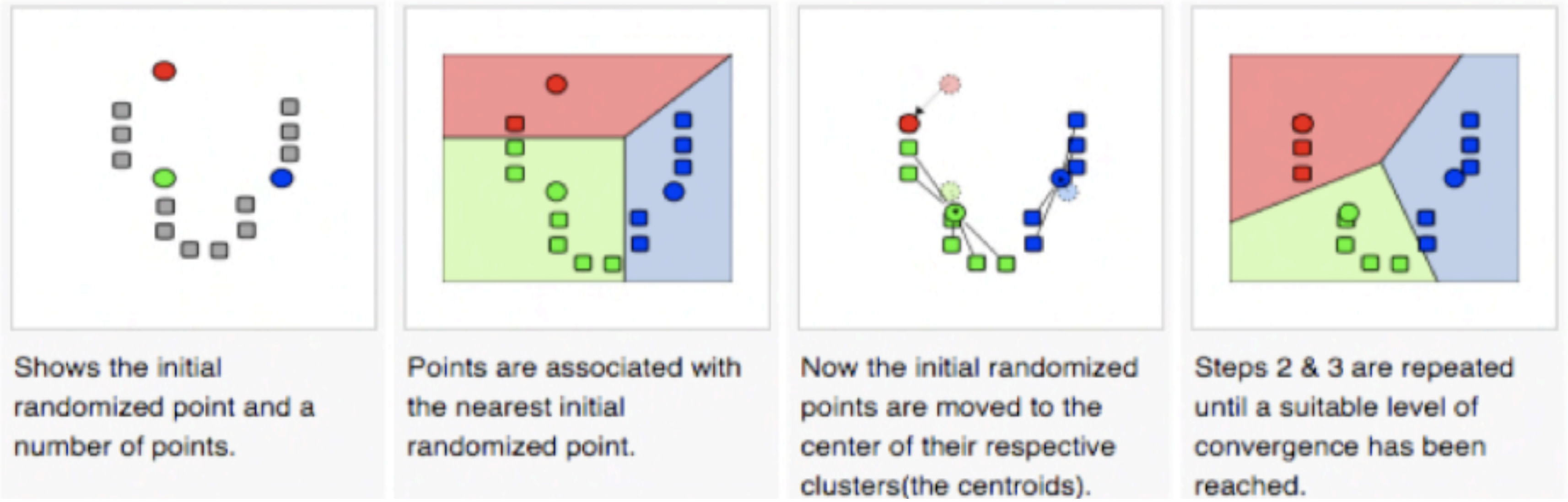
KMeans

- Unsupervised clustering method
- Partition the data into k clusters, based on their features
- Each cluster is represented by its centroid, defined as the center of the points in the cluster
- Each point is assigned to the cluster whose center is nearest
- Goal: minimize intra-cluster variance or the sum of squares of distances between data and the corresponding cluster centroid.



KMeans: how it works

- Partition the input points into k initial random generates clusters
- Build a new partition by associating each point with the closest centroid
- Computes the mean point, or centroid, of each set
- Repeat these two steps until convergence



Shows the initial randomized point and a number of points.

Points are associated with the nearest initial randomized point.

Now the initial randomized points are moved to the center of their respective clusters(the centroids).

Steps 2 & 3 are repeated until a suitable level of convergence has been reached.

Kmeans: Pro and Cons

Pro

- Simple
- Fast

Cons

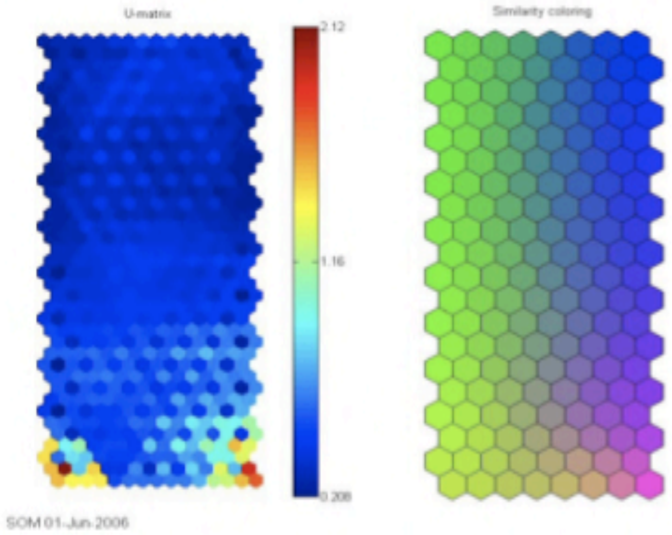
- does not yield to the same result with each run
- need to choose a priori the number of clusters
- it is not guaranteed to find the optimal configuration
  - Trick: place the first centroid on a data point, place the second centroid on a point that is far away as possible from the first one and so on...

Learning K

- Find a balance between two variables: the number of clusters (**K**) and the average variance of the clusters.
- Minimize both values
- As the number of clusters increases, the average variance decreases (up to the trivial case of k=n and variance=0).
- Some criteria:
  - BIC (Bayesian Information Criteria)
  - AIC (Akaike Information Criteria)
  - Davis-Bouldin Index
  - Confusion Matrix

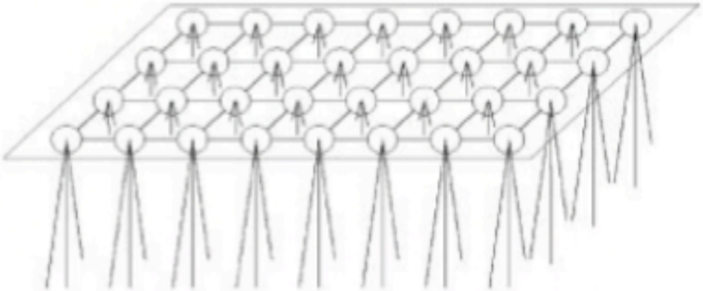
Self Organizing Maps

Self Organizing Maps learn to recognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors.



They project high dimensional data into a low dimensional output space.

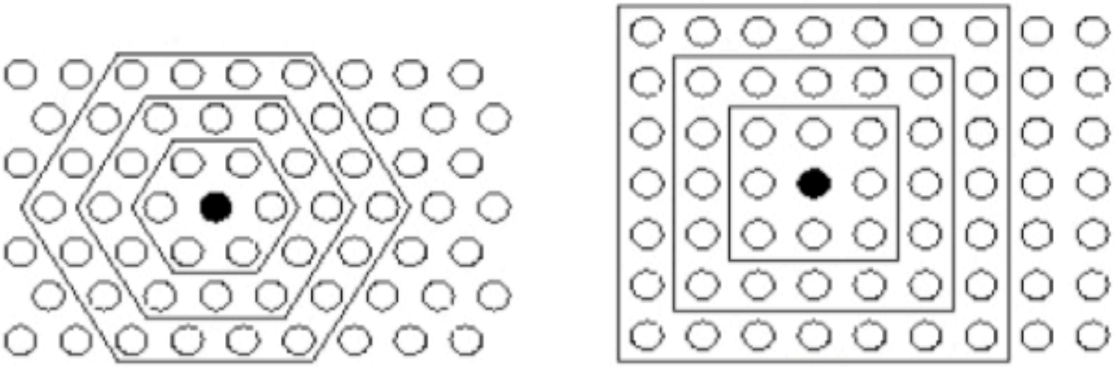
Used for clustering (or classification), data visualization, modeling, probability density estimation.



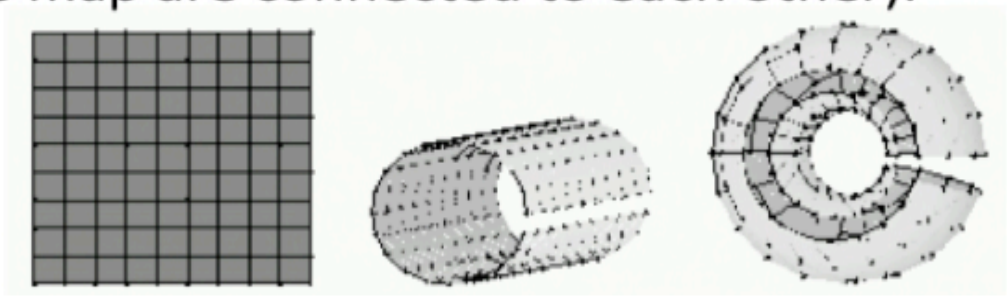
The architecture of a SOM with a two dimensional grid architecture and three inputs fed to all the neurons.

SOM topology

A SOM consists of neurons organized on a regular low dimension grid. Higher dimensional grids are possible but harder to visualize. Neurons can be organized on a rectangular or hexagonal lattice.



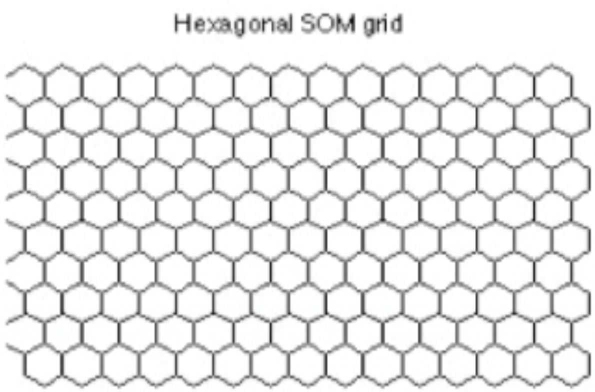
The global map shape can be rectangular, cylinder, toroid (the latter two if the sides of the map are connected to each other).

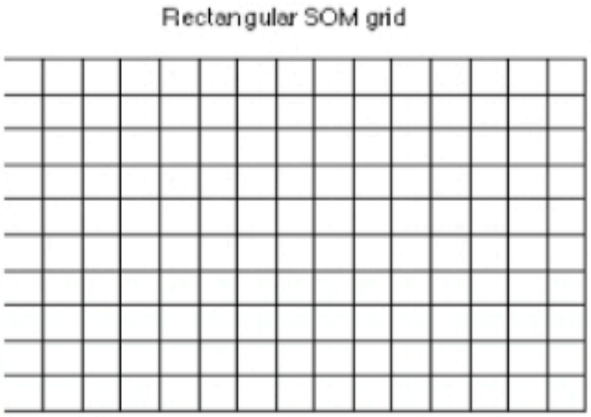


SOM Prototypes

Each neuron is represented by a d-dimensional weight vector:

$\mathbf{m_i}=[m_{i1}, \ldots , m_{id}]$ where d=dim(input_vector)



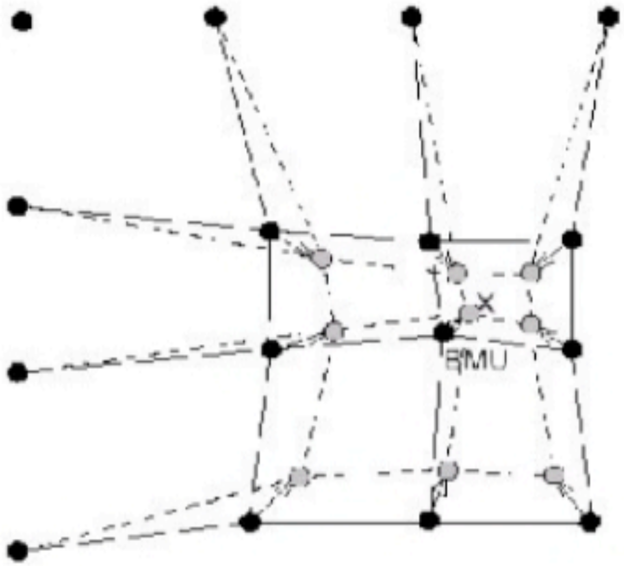


Each map unit can be thought as having two sets of coordinates:
- in the input space: the prototype vectors;
- in the output space: the position on the map.

SOM Training

In each training step, one sample **x** from the input data set is chosen; the distances between **x** and all the weight vectors of the SOM are computed;
the neuron whose weight vector is closest to the input vector is called the **Best Matching Unit** (BMU, $\mathbf{m}_c$):

$$\|x - m_c\| = \min_i \|x - m_i\|$$



Solid and dashed lines: situation before and after updating

After finding the BMU, the weight vectors are updated so that the BMU is moved closer to the input vector in the input space.

Also the topological neighbors of the BMU are treated similarly.

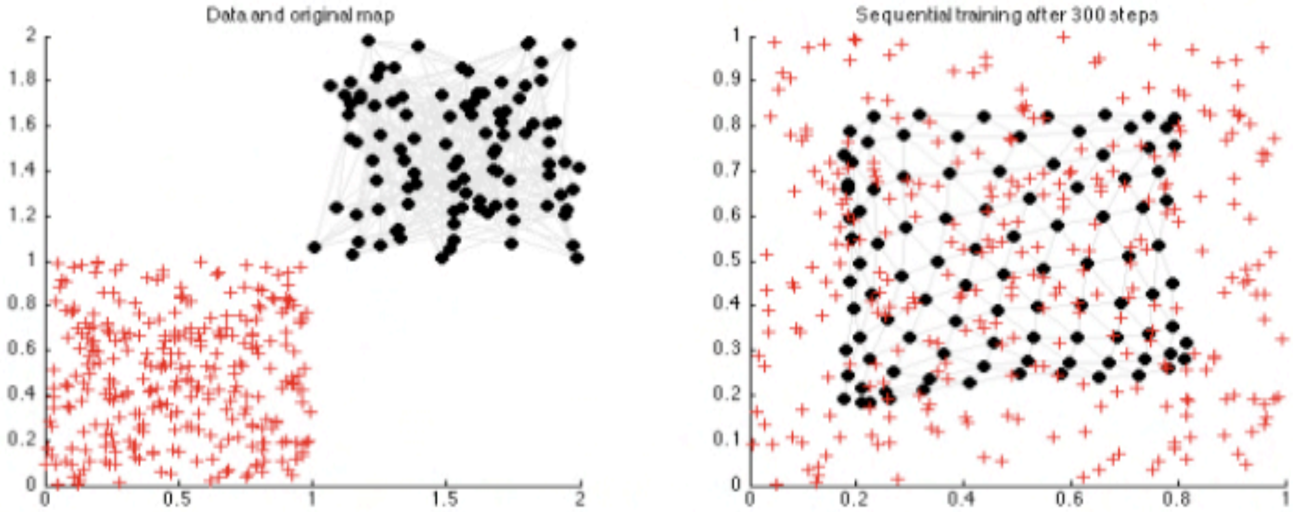
Competitive and Cooperative Learning

Competitive learning

The prototype vector most similar to a data vector is modified so that it is even more similar to it.

Cooperative learning

Not only the most similar prototype vector, but also its neighbors on the map are moved towards the data vector.



SOM Update Rule

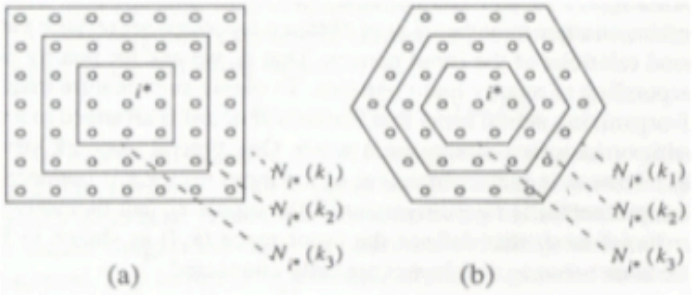
The SOM update rule for the weight vector of unit *i* is:

$$m_i(t+1) = m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)]$$

where:
- x(t) input vector chosen at time t;
- $h_{ci}(t)$ is the neighborhood function that define the kernel around the winner unit (BMU) c:

$$h_{ci}(t) = \exp(-\|r_c - r_i\|^2 / 2\sigma^2(t))$$

- σ(t) is the neighborhood radius at time t



where $k_1<k_2<k_3$

- α(t) is the learning rate at time t
  linear .α(t)= α_0(1-t/T) where α_0=initial learning rate; T=training length
  inversely proportional to time α(t)= A/(t+B) with A,B suitable constants

Parameters

- Map size and Topology
  - Default: 5*sqrt(n) where n = # of samples
- Weights initialization
  - random: with small random values
  - sample: with random samples drawn from the input
  - linear: along the linear subspace spanned by the two principal eigenvectors of the data set
- Batch and Sequential training
  - sequential: sample are presented to the map one a time and the algorithm gradually moves the weight vectors toward them
  - batch: the data set is presented as a whole and the new weight vectors are weighted averages of the data vectors
- Learning rate, neighborhood function, radius.

DM with SOM

How to find clusters?

Which components are the most important in discriminating among the classes?

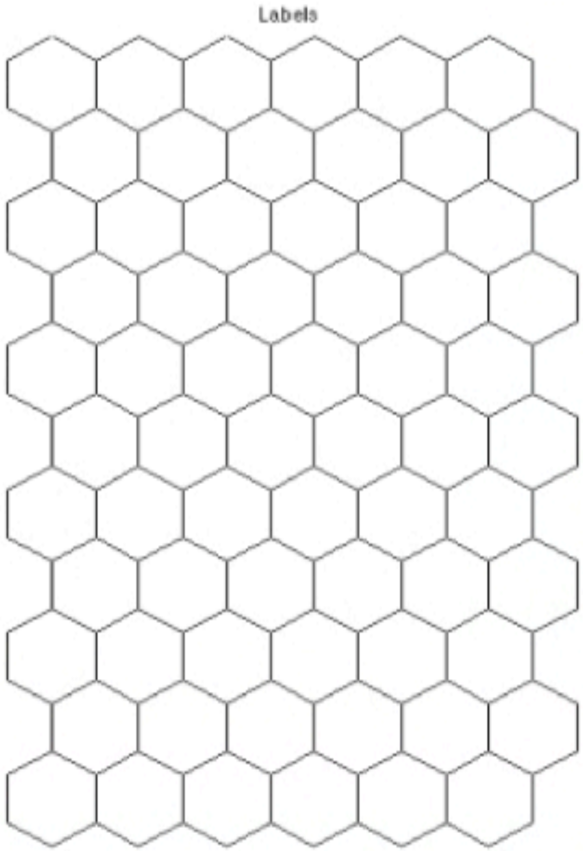
How do the parameters relate to the clusters?

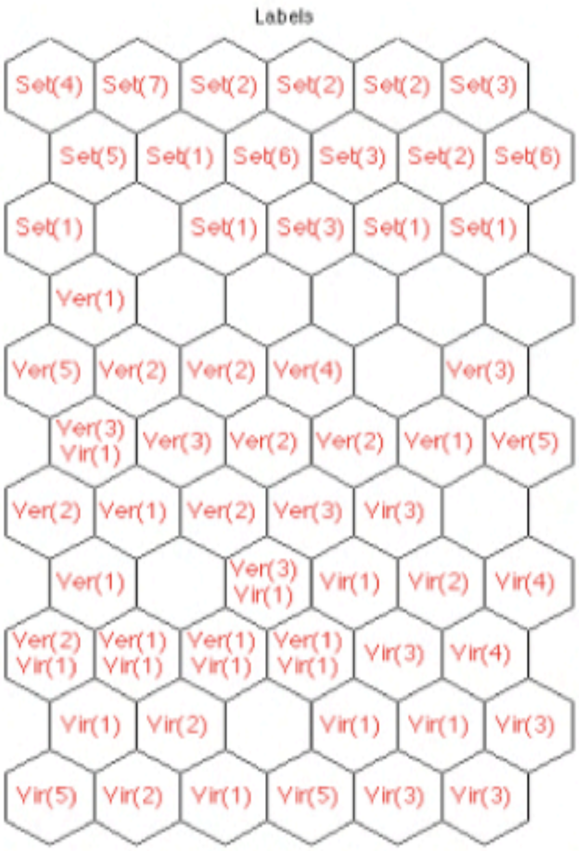
With the SOM we have several ways to visually (and not only) answer to the these question.

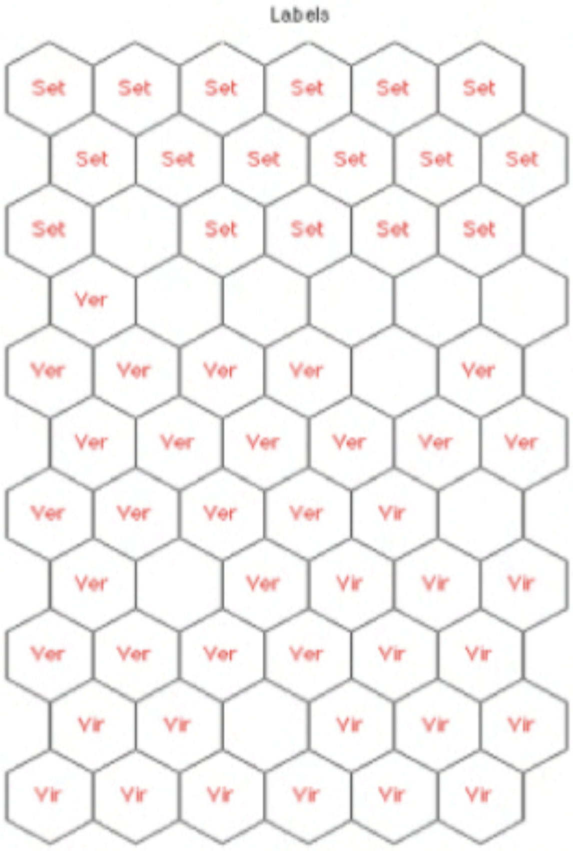
“Cool” visualizations looks good in the papers too!

SOM Labeling

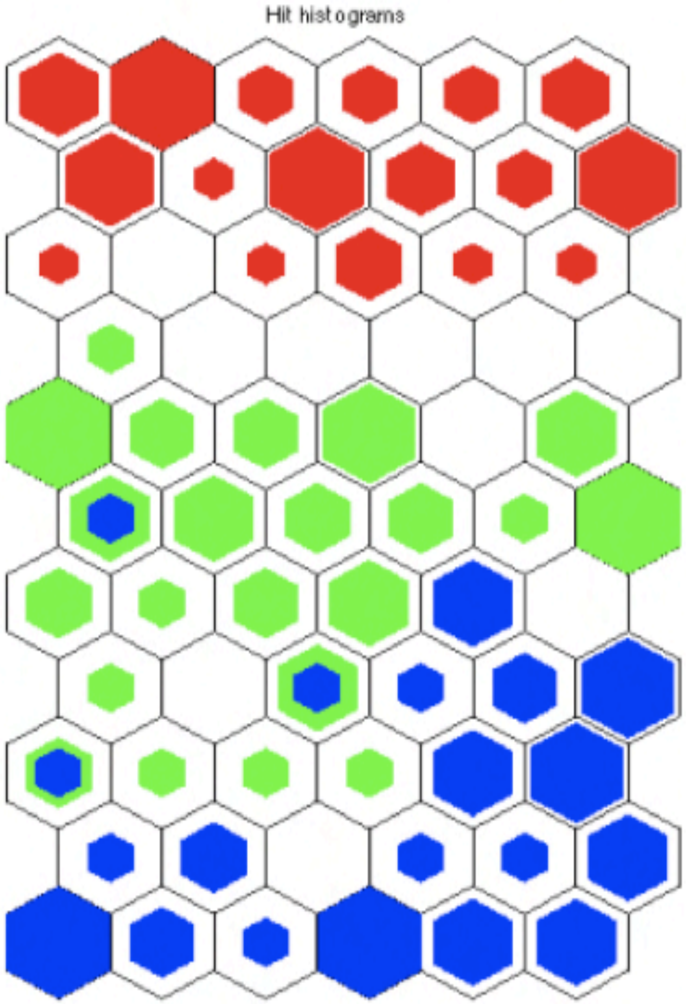
The BMU of each sample is found from the map, and the species label can be given to each map unit.




Labels
Set(4) Set(7) Set(2) Set(2) Set(2) Set(3)
Set(5) Set(1) Set(6) Set(3) Set(2) Set(6)
Set(1) Set(1) Set(3) Set(1) Set(1)
Ver(1)
Ver(5) Ver(2) Ver(2) Ver(4) Ver(3)
Ver(3) Vir(1) Ver(3) Ver(2) Ver(2) Ver(1) Ver(5)
Ver(2) Ver(1) Ver(2) Ver(3) Vir(3)
Ver(1) Ver(3) Vir(1) Vir(1) Vir(2) Vir(4)
Ver(2) Vir(1) Ver(1) Vir(1) Ver(1) Vir(1) Ver(1) Vir(1) Vir(3) Vir(4)
Vir(1) Vir(2) Vir(1) Vir(1) Vir(3)
Vir(5) Vir(2) Vir(1) Vir(5) Vir(3) Vir(3)



Localizing Data

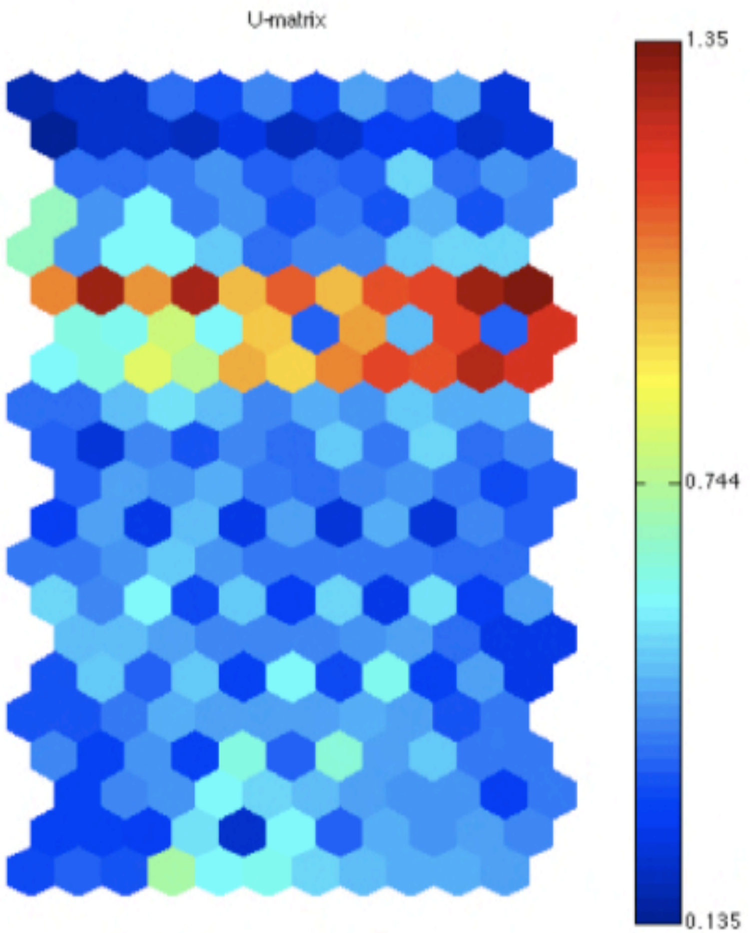


An important tool in data analysis using SOM are the so called **hit histogram**.

The hit histogram shows the distribution of the data set on the map.

They are formed by taking a data set, finding the BMU of each data sample from the map, and increasing a counter in a map unit each time it is the BMU.

Cluster Structure



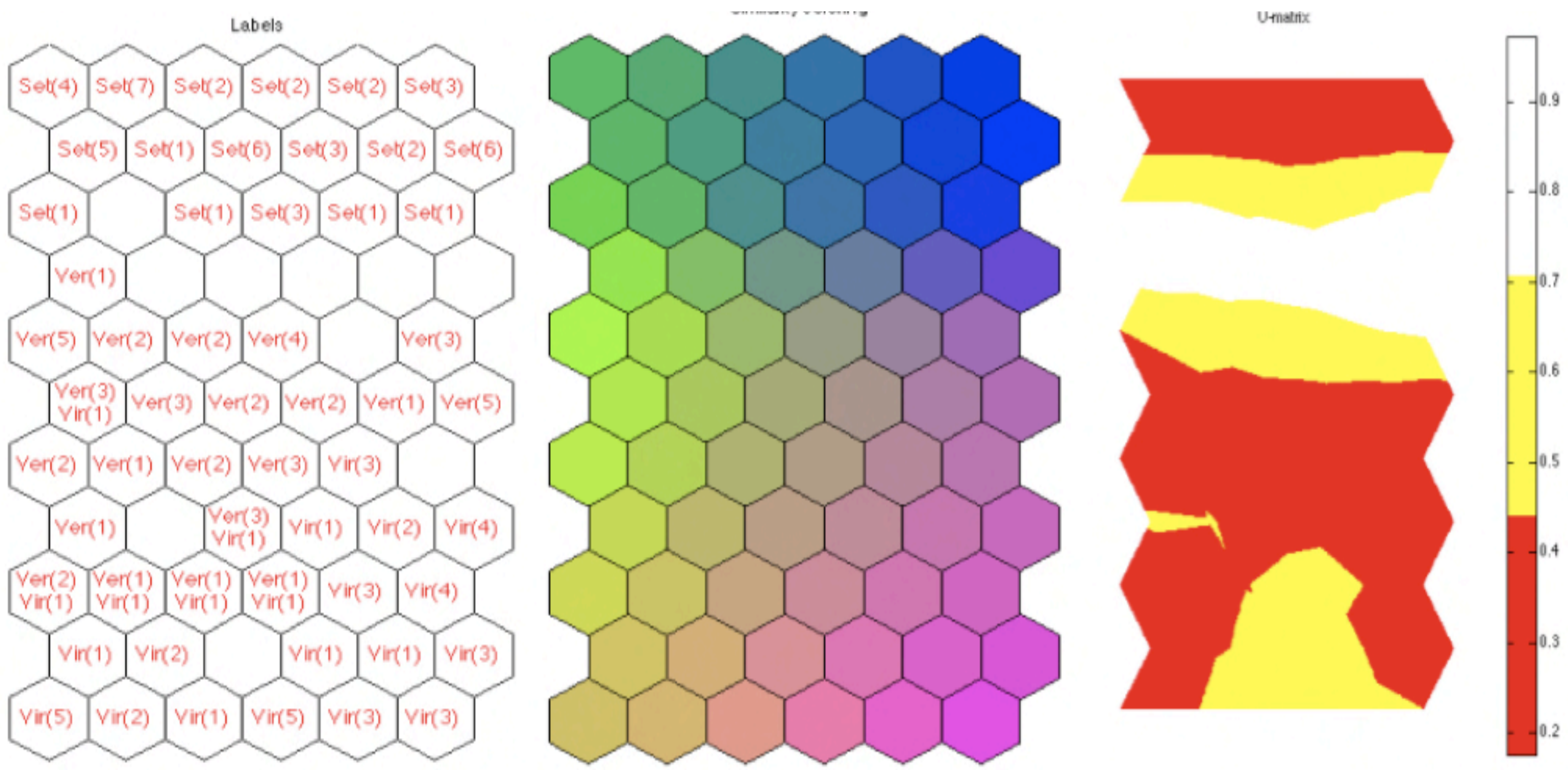
The **U-matrix** shows distances between neighboring units and thus visualizes the cluster structure of the map.

The U-matrix visualization has much more hexagons that the “real” map because distances between map units are shown.

High values on the U-matrix mean large distance between neighboring map units.

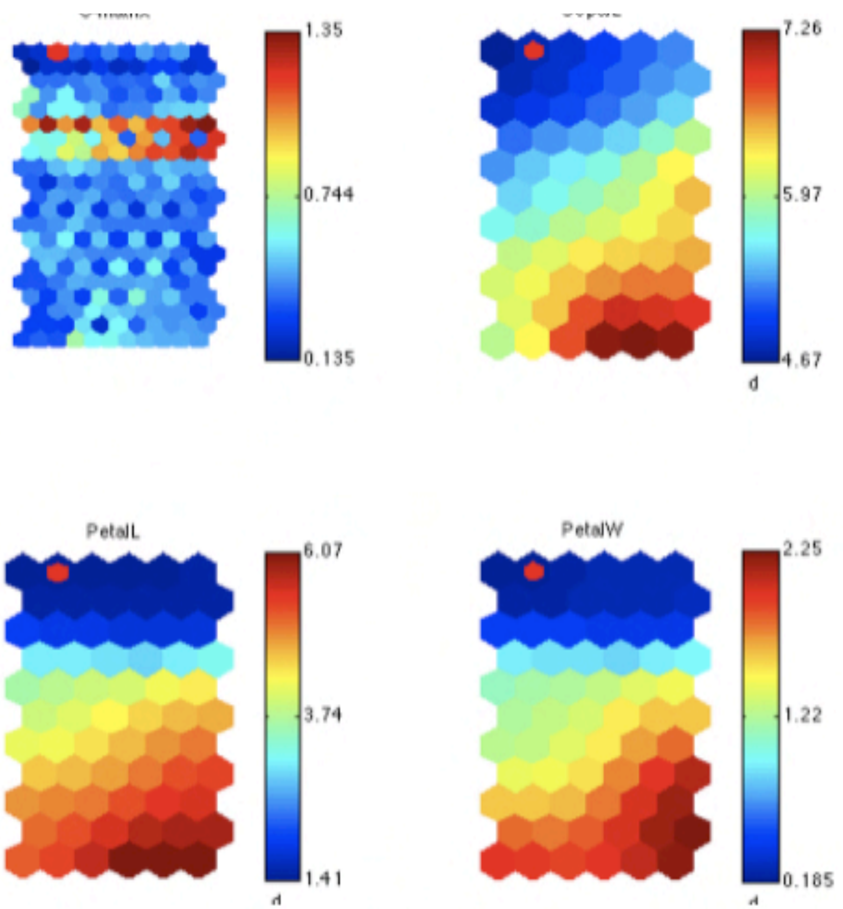
If we have five units 5x1: m(1), m(2), m(3), m(4), m(5)
U-Matrix is a 9x1 vector:
u(1), u(1,2), u(2), u(2,3), u(3), u(3,4), u(4), u(4,5), u(5)
Where:
u(i,j) is the distance between m(i) and m(j);
u(k) is the mean: u(k)=(u(k-1,k)+u(k,k+1))/2.

Cluster Structure - 2



Similarity coloring: assigns colors to the map units such that similar map units get similar colors.

Component Planes

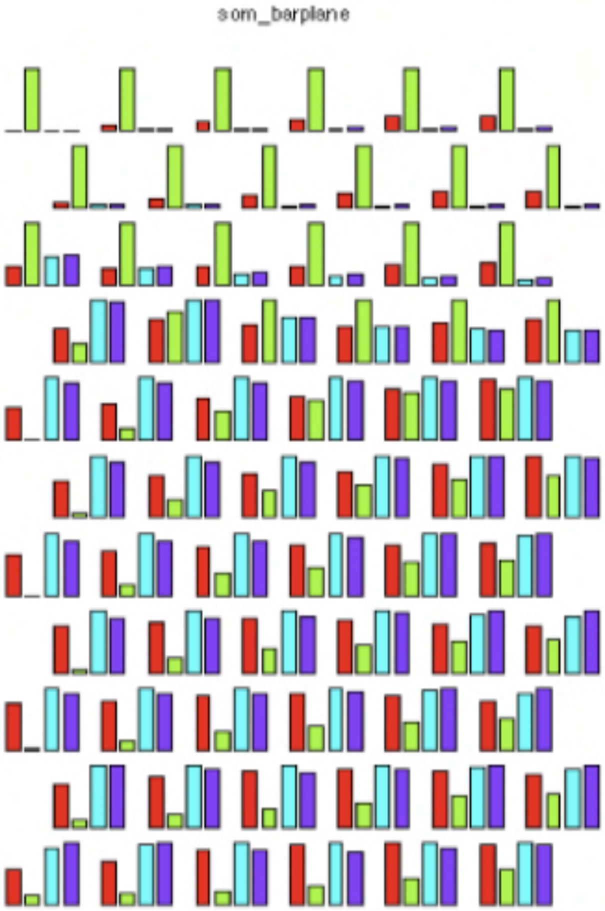


Visual inspection:

- PetalL and PetalW are highly correlated;
- SepalL is ~ correlated to PetalL and PetalW;
- Setosa has small petals and short wide sepals;
- the 2nd cluster is Virginica /Versicolor;
- Versicolor has smaller leaves than Virginica.

The **component planes** show the values of the prototype vectors for each parameter. Can be used for correlation hunting.

Relative Importance



Which components are the most important in discerning among the classes?

The significance of the components with respect to the clustering is usually harder to visualize.

One indication of importance is that on the borders of the clusters, values of important variables change very rapidly.

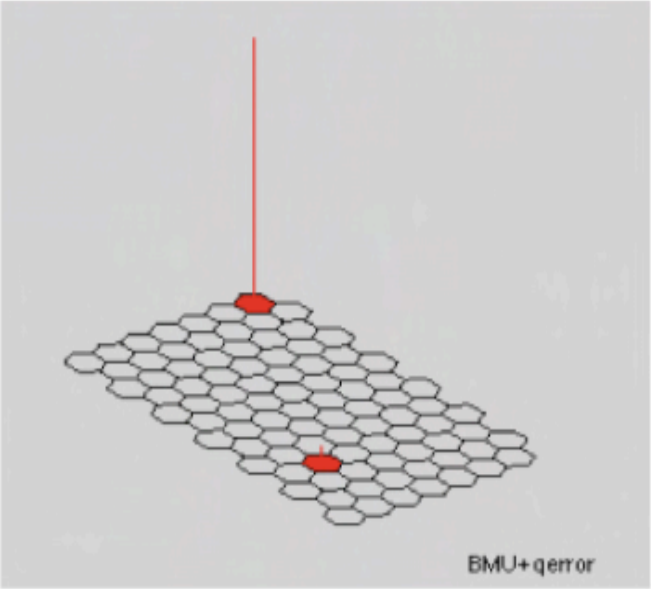
Each chart shows the relative importance of each variable in each map unit.

How accurate is your clustering

We have seen how to locate a sample on the map. But how accurate is that localization?

We can compute and plot how much a data sample is far from its BMU, its 2nd-BMU and WMU and so on.

Errors are also computed and are useful in determining the quality of the clustering.

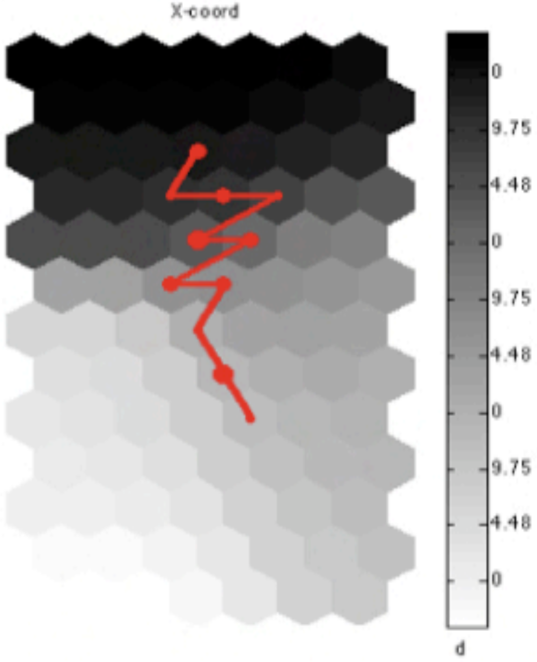


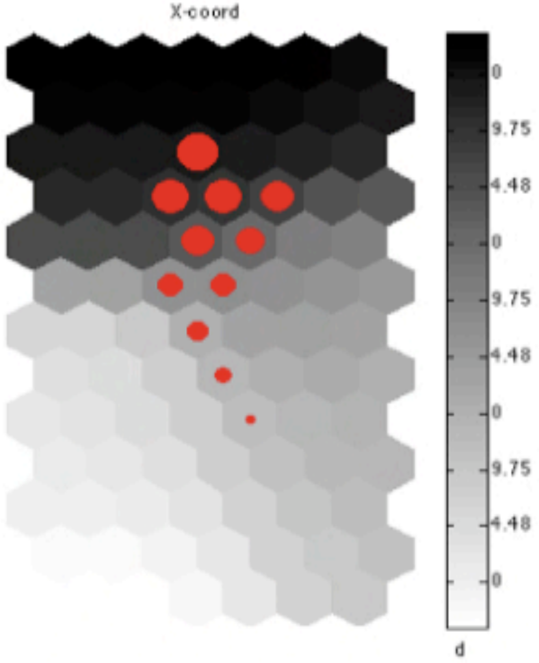
Average quantization error: measures the distance from each data vector to its BMU.

Topographic error measure: percentage of data vectors for which the BMU and the second-BMU are not adjacent units.

Trajectories

A special data mapping technique is trajectory. If the samples are ordered, forming a time-series for example, their response on the map can be tracked.

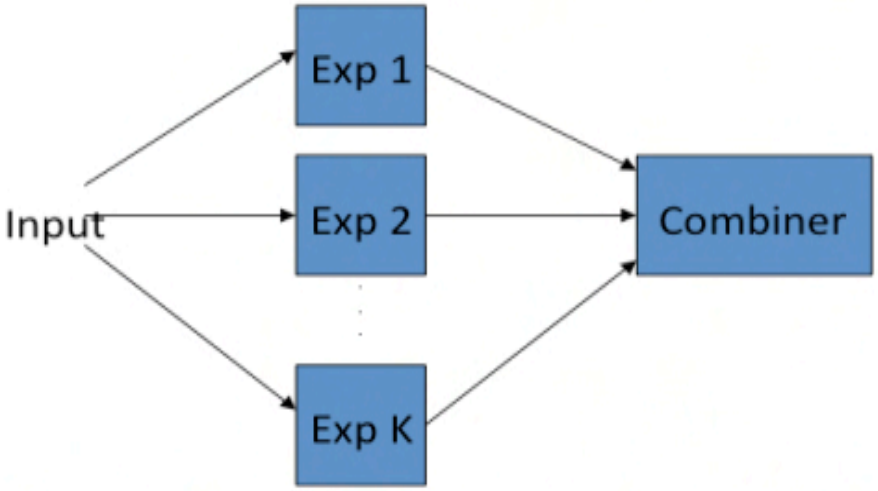




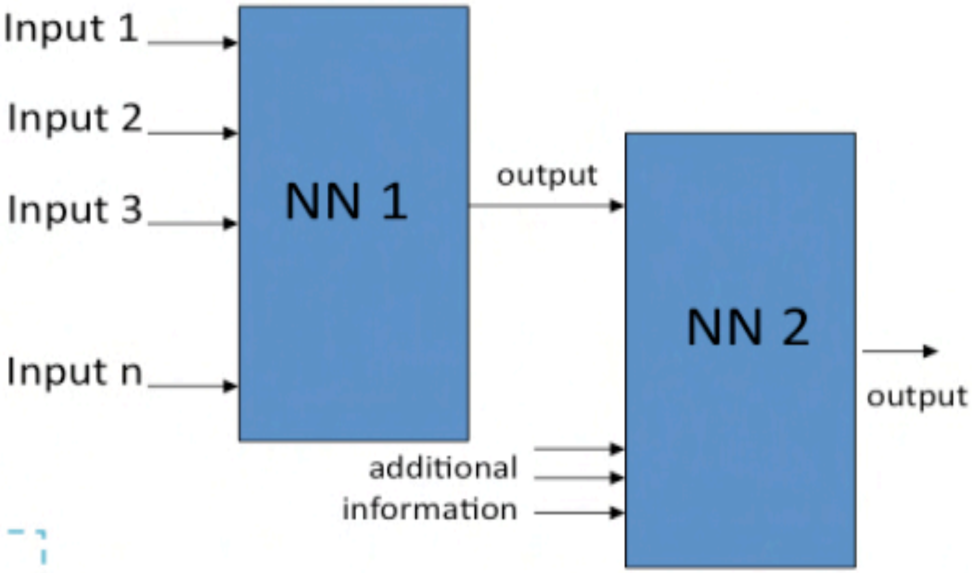
Combining Models

- It is often found that improved performance can be obtained by combining models together in some way, instead of using a single model in isolation. In this way, individual classifiers may be optimized or trained differently.
- Some classifiers could work better than others in recognizing some classes when certain input parameters are present while some others may be activated only in presence of certain inputs.
- It has been shown that using such a framework is a powerful way to decompose complex classification problems.

Committee Machines



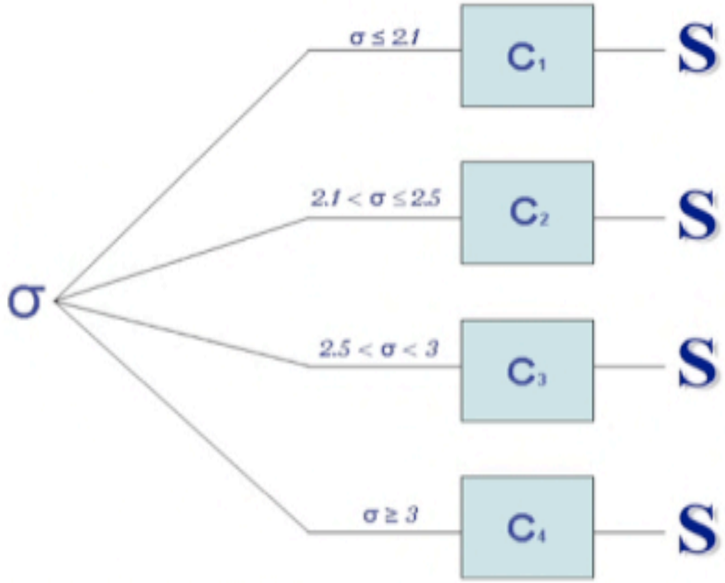
Committee Machines: combination of experts that "vote" together on a given example.

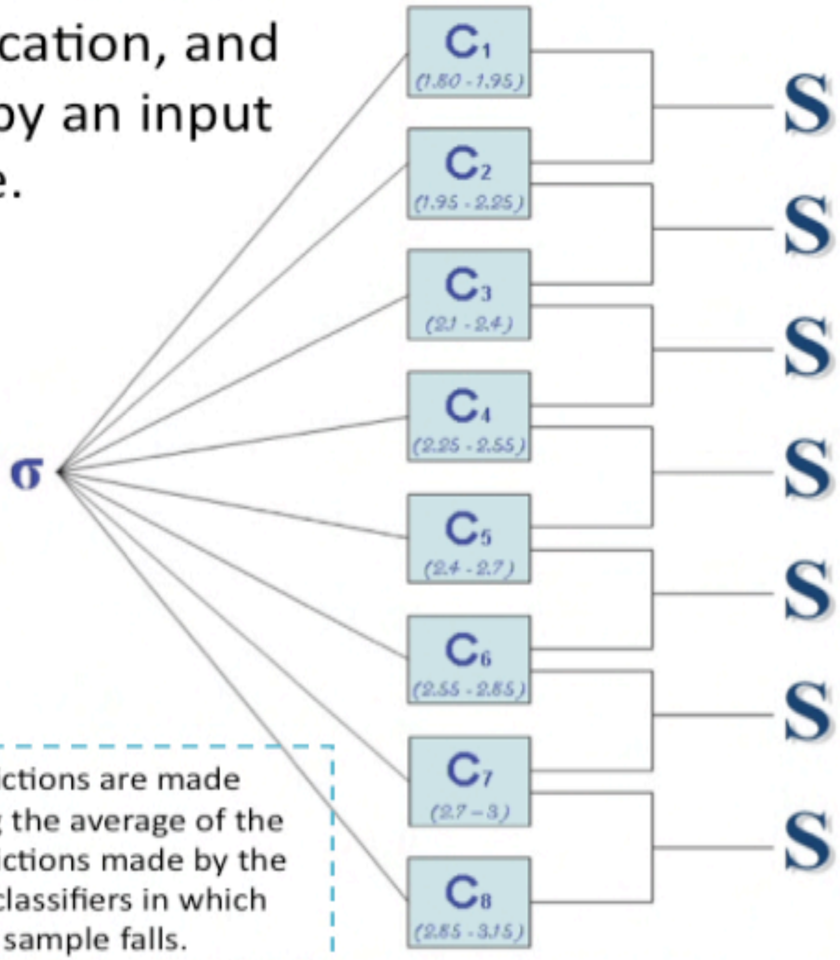


Two-level Network.

A priori knowledge

An alternative of model combinations is to select one of the models to make the classification, and let the choice of the model be driven by an input parameter or by an a priori knowledge.

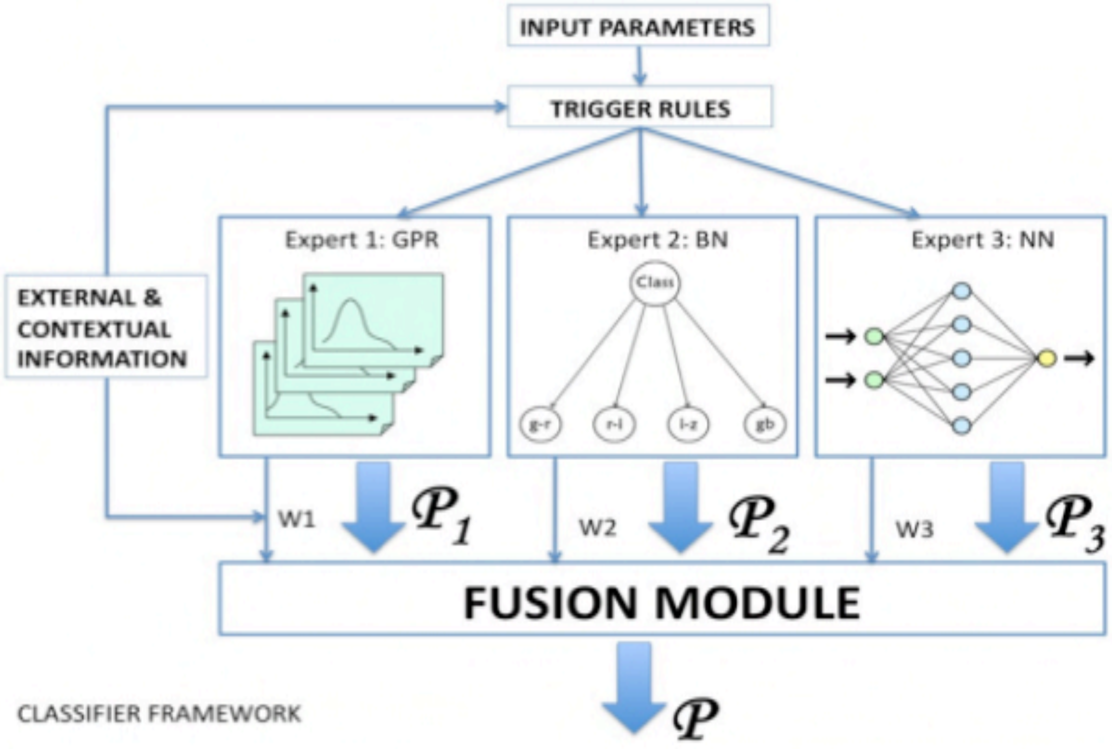




Predictions are made using the average of the predictions made by the two classifiers in which each sample falls.

Sleeping Experts

- Sleeping experts can be seen as a generalization of the IF-THEN rule: IF the condition is satisfied THEN activate this expert.
- Each classifier makes a prediction only when the instance to be predicted falls within its area of expertise.


INPUT PARAMETERS
TRIGGER RULES
EXTERNAL & CONTEXTUAL INFORMATION
Expert 1: GPR
Expert 2: BN
Class
g-r
r-i
i-z
gb
Expert 3: NN
W1
$\mathcal{P}_1$
W2
$\mathcal{P}_2$
W3
$\mathcal{P}_3$
FUSION MODULE
$\mathcal{P}$
CLASSIFIER FRAMEWORK