It's all about features

- Feature selection
- Cluster analysis with k-means
- Prediction with k-nearest-neighbor

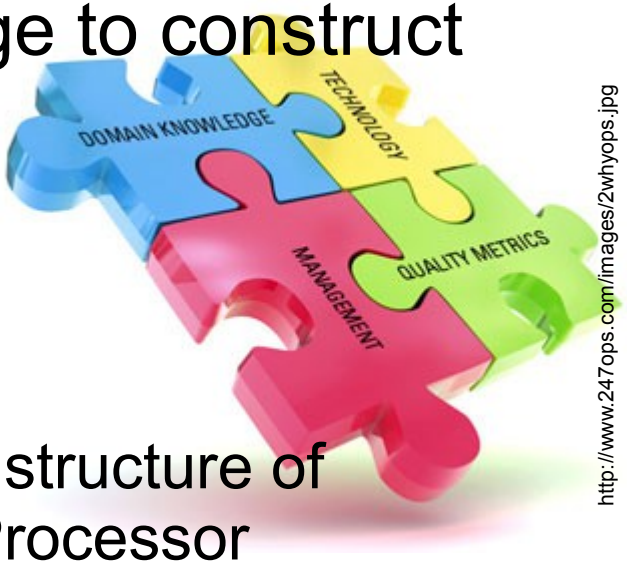
Some Data Processing Phases

- Data collection (raw Features)
- Feature computation
- Feature selection

http://historyofeconomics.files.wordpress.com/2008/10/data-mining.jpg

Feature computation

- Increase the abstraction of features
  - Use background/domain knowledge to construct features of higher value
- Examples
  - Image Processing
    - Insights into computation of hue and structure of ImageFeatureCreation and FeatureProcessor
  - Document Processing
    - Insights into body-parser (tokenizer + String.contains)



http://www.247ops.com/images/2whyops.jpg

Feature Selection

http://www.aldarin-electronics.com/images/market%20analysis.jpg

- Remove redundant features
  - Speeds up learning
  - Enhance generalization capability
- Analyze feature according to redundancy
  - Correlation between features
    - Which features can be replaced?
- Analyze features according to predictive value
  - Correlation with class attribute (continuous attribute)
  - Attribute with least entropy (nominal attribute)

Redundant features

- Given:
  - 5 features (dogs)
- Question:
  - Which of the features are just “boring” copies of other features and can be removed without fears?

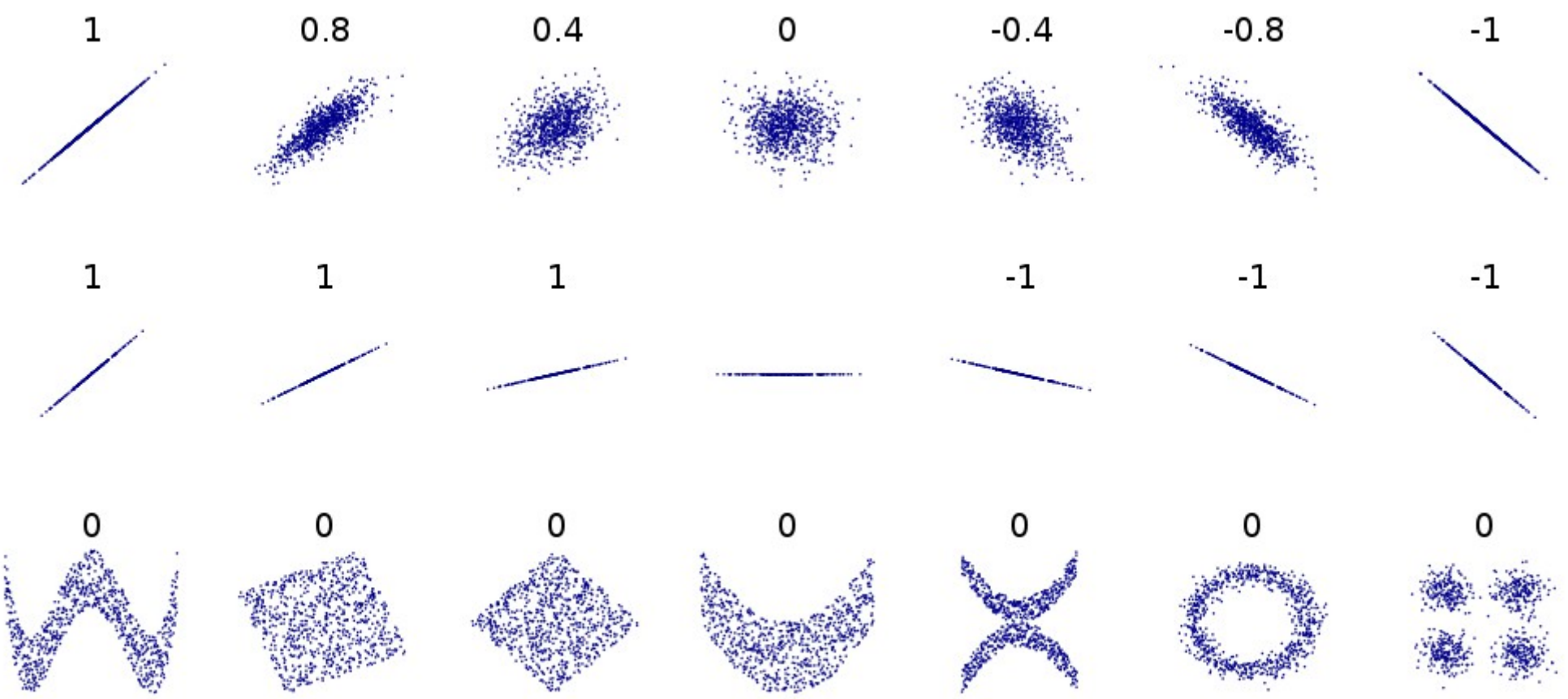
http://visual.merriam-webster.com/images/animal-kingdom/carnivorous-mammals/dog-breeds_2.jpg

http://www.finedogbreeds.com/mydalmation.jpg

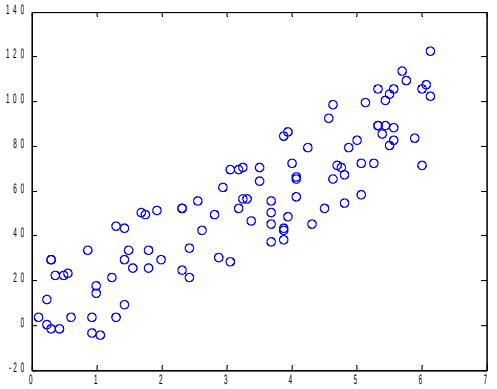
Pearson Correlation

- Average μ
- Standard-deviation δ

$$\rho_{XY} = \sum_{c=1}^{numberColumns} \frac{(X_c - \mu_X) \cdot (Y_c - \mu_Y)}{\delta_X \delta_Y}$$



Feature selection example 1



- Practice
  - Load the data matrix from the material section (material.zip folder dataMatrix)
  - This folder contains a matrix “mat” that has 12 columns=attributes; the last column is the class attribute
  - Find out which features can be replaced
    - use the corrcoeff-function to have an overview on the pair-wise correlations (high correlations are close to 1 or -1)
    - Visualize pairwise the dependent variables using the function “showCorrelation(mat,attr1,attr2)”

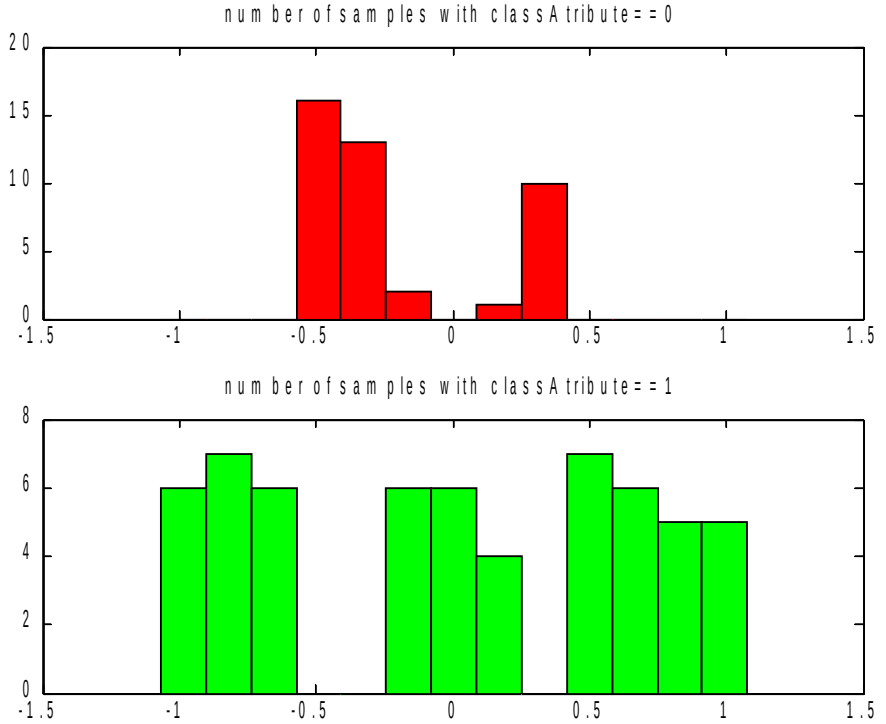
Predictive value of features

- Given:
  - A training database with features and 1 class
    - The features are continuous or ordinal
    - The class is binary (“0”,”1”)
- Question:
  - Which of the features should be used for prediction?

http://www.iconarch.com/assets/firm/meetus/people_web.png

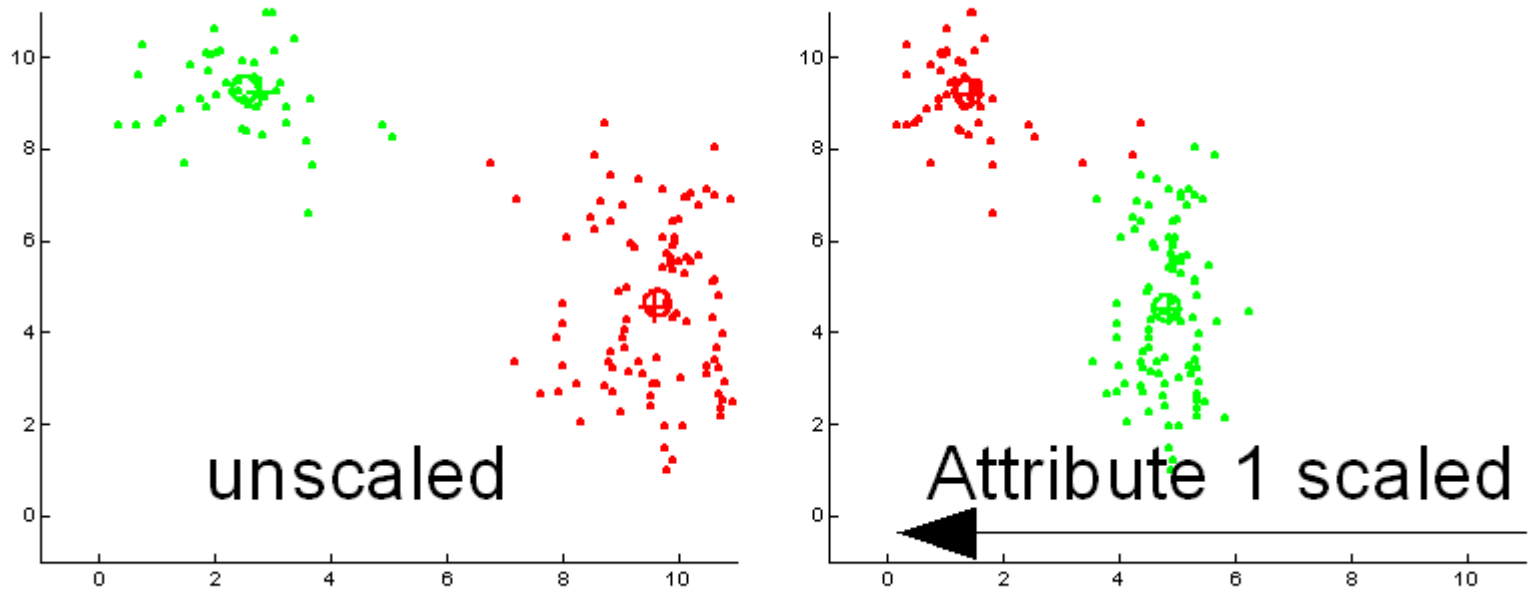
Practice

- Find the feature with maximal predictive value
  - Use the histogram function showClassHistogram(mat,attr) to decide which independent variable has the highest predictive power



Feature Weightening

- An alternative to removal of features
  - Why not completely remove features?
- Assign lower weights to unimportant features
  - Example: scale one dimension and watch the difference in cluster outcome (after re-scaling)

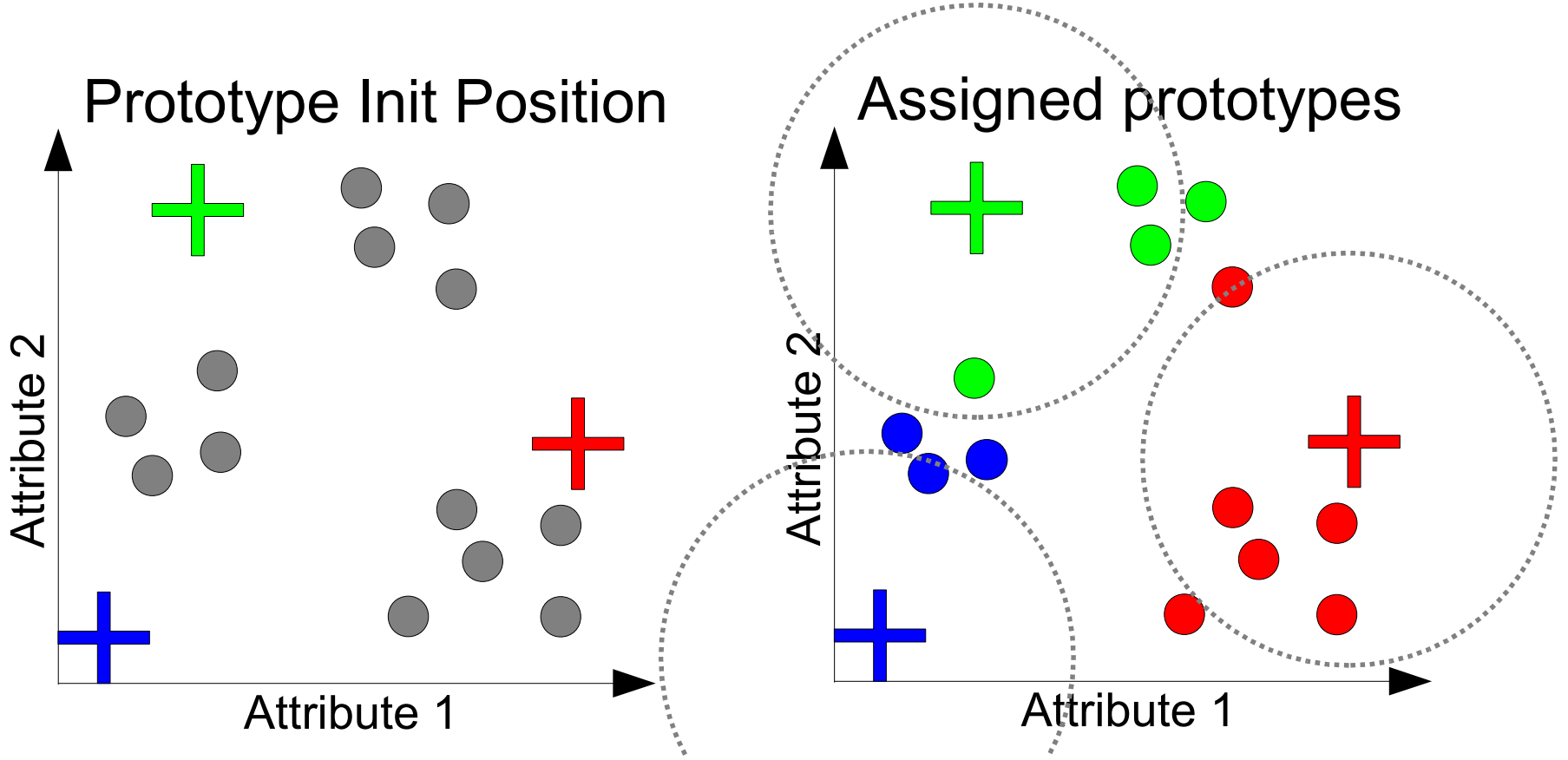


2. Cluster analysis

- K-means-clustering
  - Reduces the number of samples of input space to a few number of clusters
- Input (unsupervised learning)
  - Samples from n-dimensional input space
  - The number of cluster centers (also called prototypes p)
- Output
  - p Cluster centers approximating the data
  - Each data sample has a cluster assigned

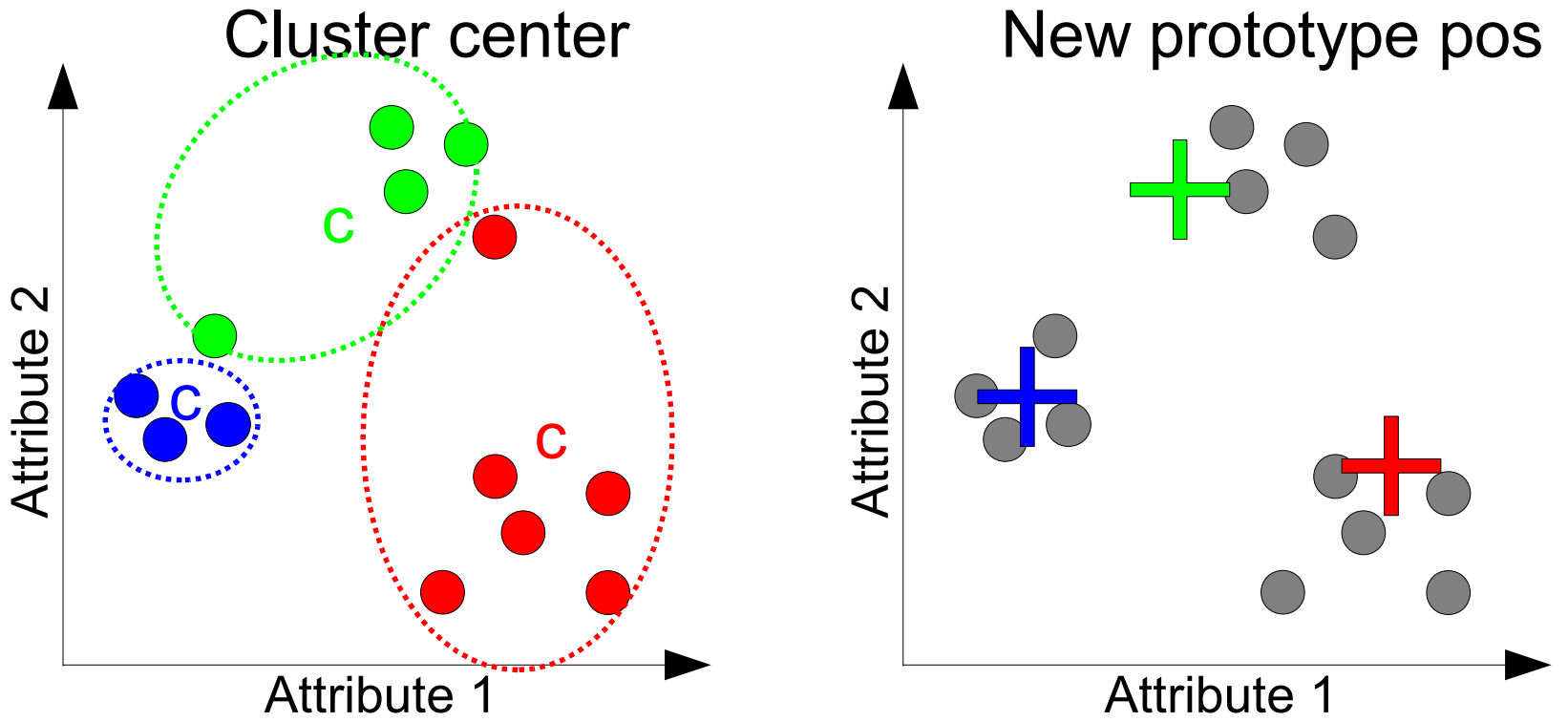
K-means learning Step 1

- For each sample
  - Compute the closest prototype



K-means learning Step 2

- For each prototype
  - compute the cluster center (gravity center of planets)
  - Set the prototype to the cluster center position



- Goto Step 1 followed by Step 2 until steady state

Practise K-means

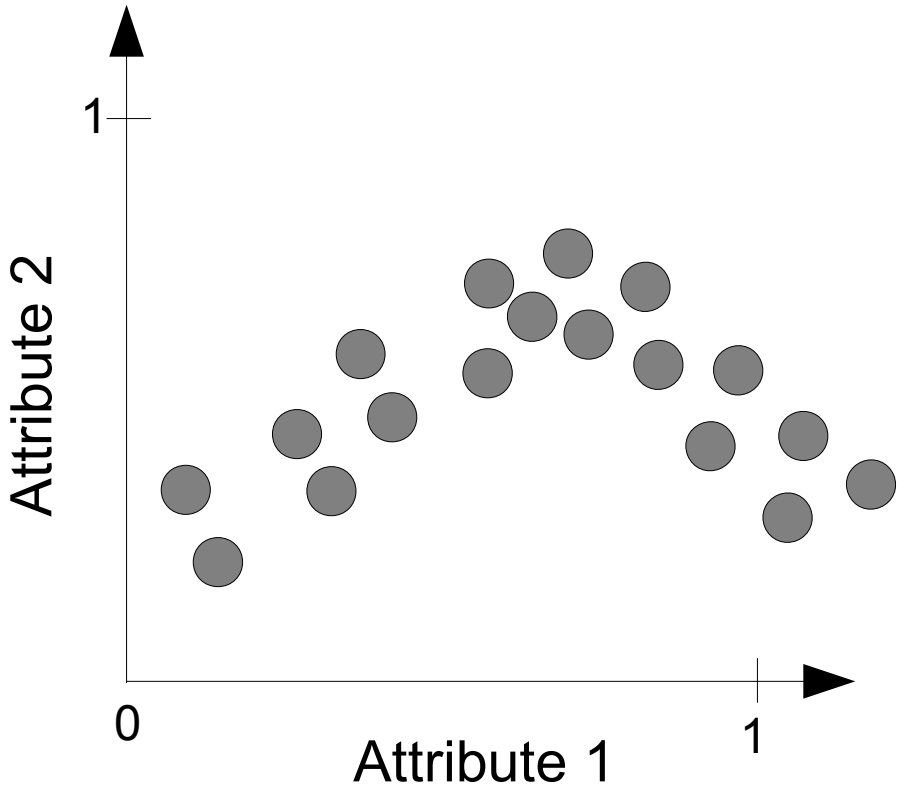
- K-means in 2-D input space with 2,3,4 clusters
- Open folder kmeans from Material.zip
  - data=generateData and draw the data
  - class=kmeans(data,3);

Properties of k-means

- Bad
  - Numbers of clusters required (this is often not known in advance)
  - Hard to recognize what makes a cluster
  - What happens with non-clustered data?
  - Requires computation (p times s) of distance from
    - Class prototypes (p) to
    - Samples (s)
- Good
  - simple
  - Works for any number of dimensions

Problem k-means

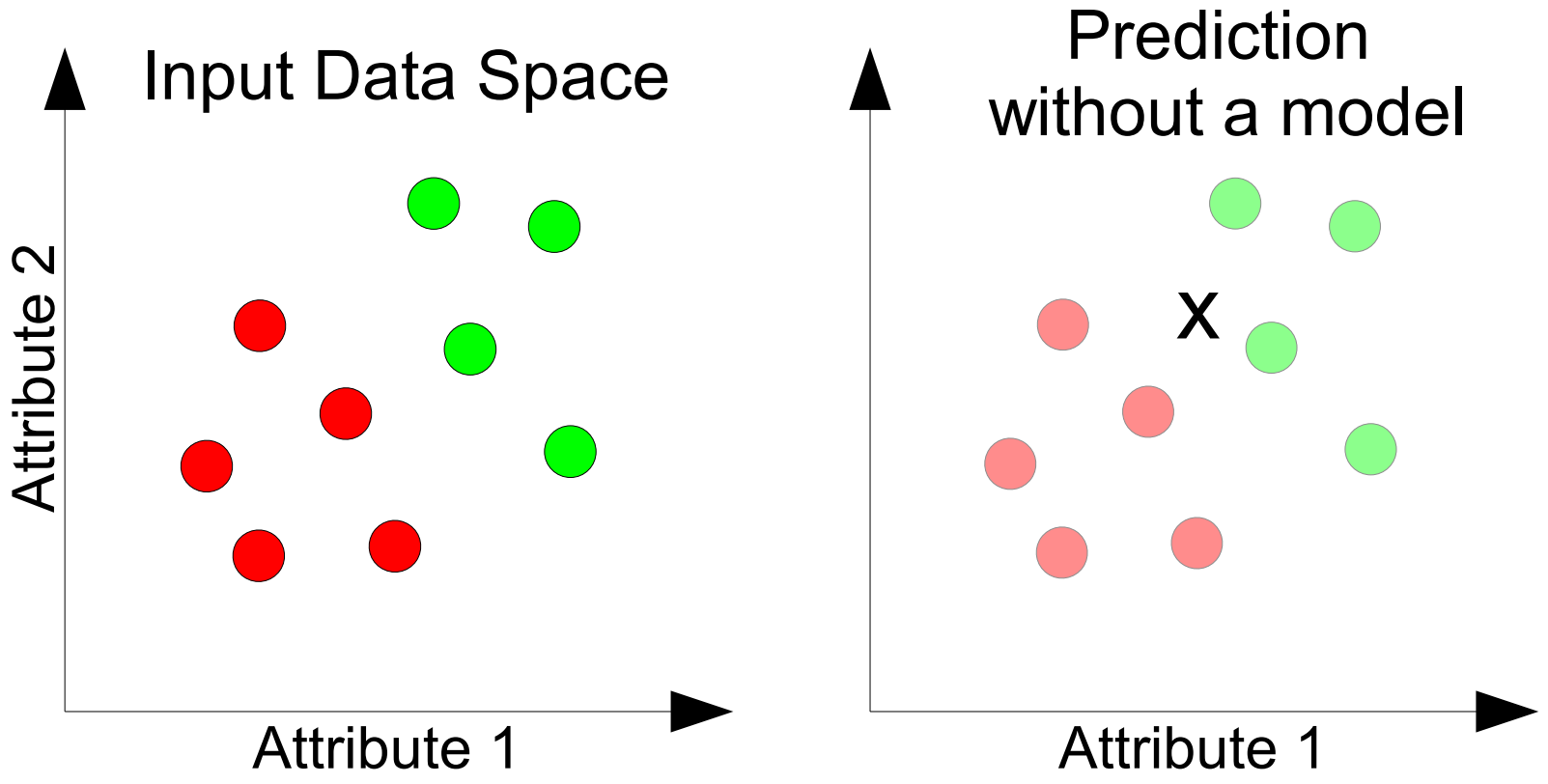
- The data clearly contains a simple rule but this is hard to find if the rule is not expressed as cluster
- Data “stripes”



3.The simplest prediction model

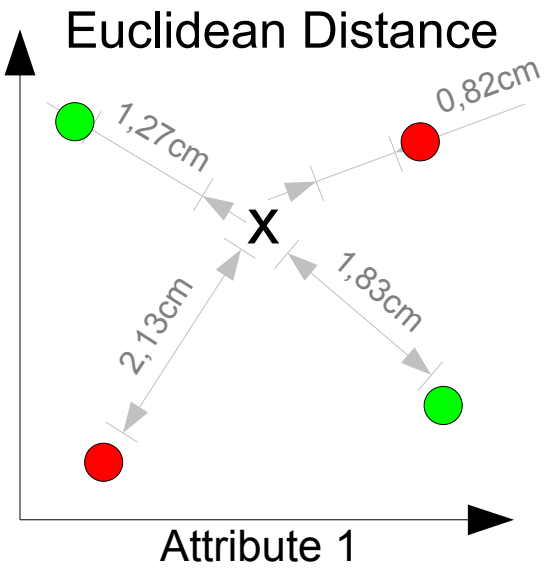
K-nearest-neighbor

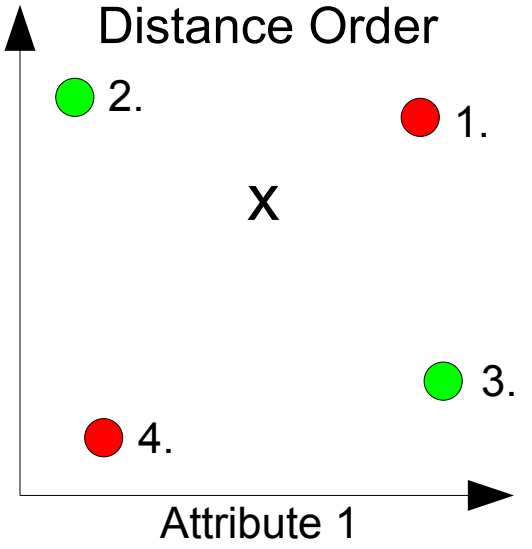
- In contrast to all other algorithms
  - Does not create a model but uses the samples directly to make predictions



Prediction algorithm

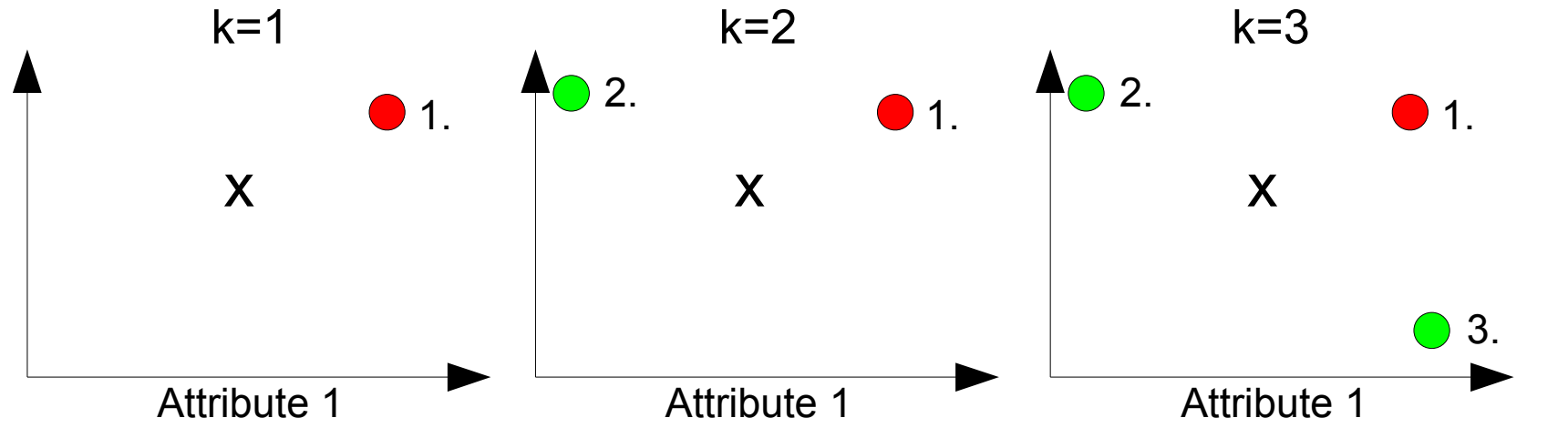
- Is based on distance measures
  - Calculate the distance between new sample and all samples in the training set
- Uses the least distant samples to predict the class
  - Majority vote





Example with different k's

- Consider the k nearest neighbors and use their class labels to predict the class for x



	Class red	Class green
Frequency	1	0

Predict: red

	Class red	Class green
Frequency	1	1

Predict: ?

	Class red	Class green
Frequency	1	2

Predict: green

Practise

- Use the matlab folder k-NN from the materials.zip
  - Use generateData to create supervised training samples with two independent attributes and a class label
  - Call kNN(data,targetX,targetY) to see how the prediction changes dependent on parameter k

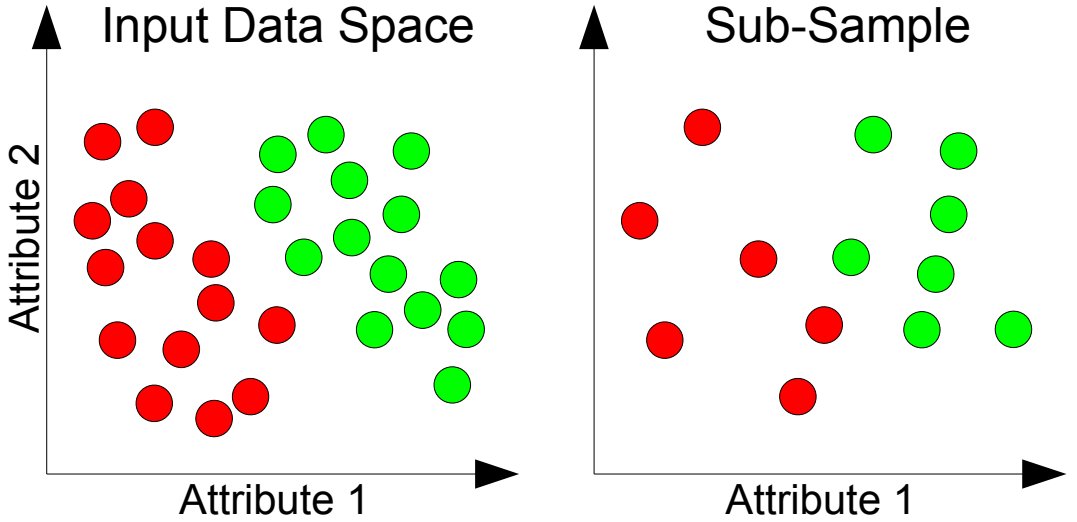
Properties of k-NN

http://www.marketcalls.in/2010/04/real-time-data-feed-to-amibroker.html

- Case based not model based
  - Or in other words: the samples are their own model
- Is not prone to become outdated
  - As soon as new training data arrives the prediction result can change
  - Contrasting Model-based prediction which require to learn the model again as soon as the training set is updated
- Is computationally very expensive
  - solution: sub-Sampling (next slide)

Sub-Sampling

- Reduce the amount of samples by selecting samples randomly



- Try the subSample function to reduce the number of samples for prediction
- Load oddData, try kNN before and after Sub-Sampling