

Unit 5 : Classification

Dr. Latesh Malik

BE, MTech, PhD (Computer Science)

Associate Professor, Department of Computer Engineering

Govt Engineering College , Nagpur

Classification

- Naive Bayes classifier and its implementation in R
- Decision tree classifier and its implementation in R
- K-Nearest Neighbour method and its implementation in R
- K-means clustering technique and its implementation in R

Machine Learning

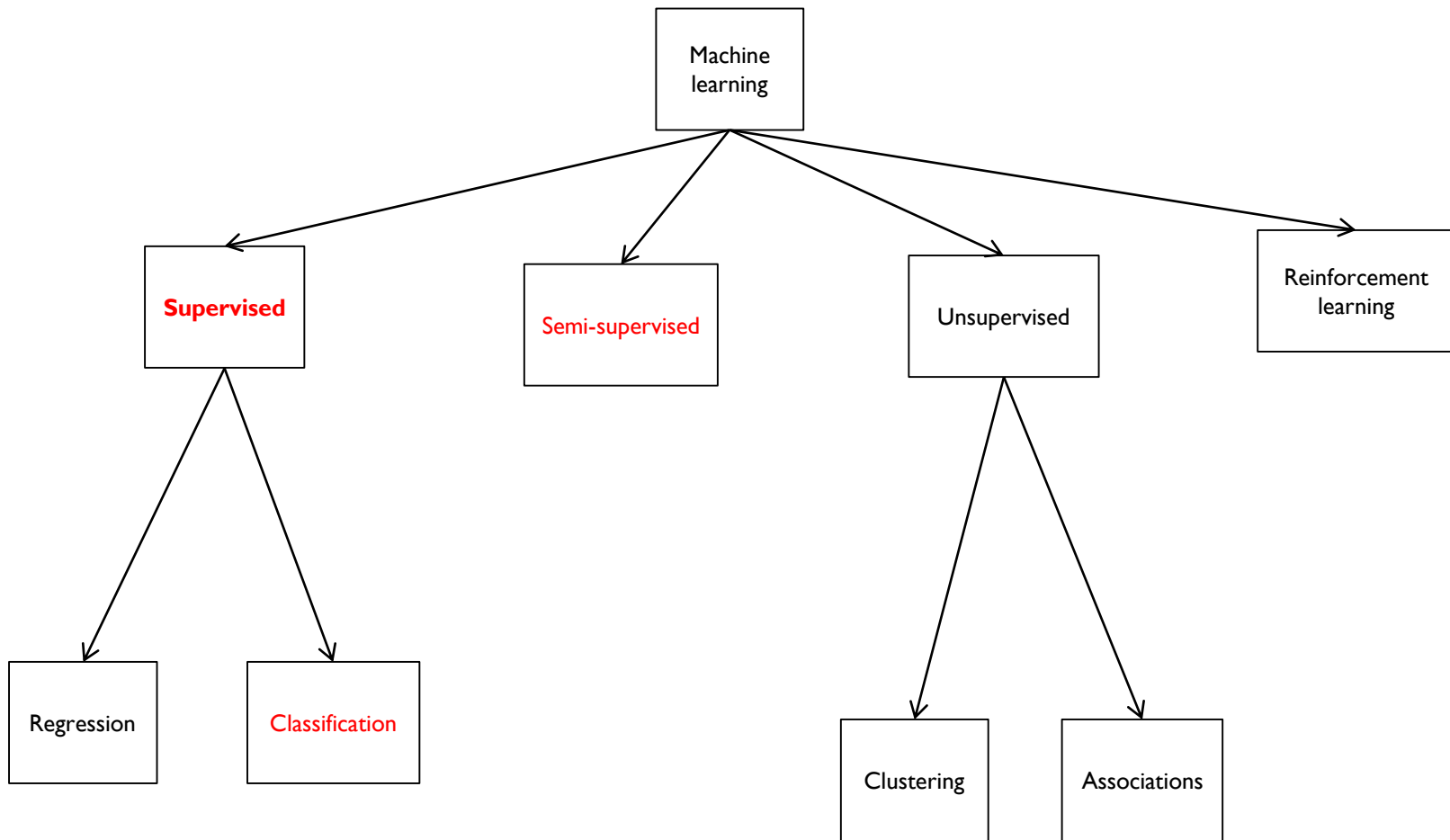
--Machine learning (ML) is the study and development of computer programs that can learn progressively from themselves to improve their performance of a particular task.

--Using machine learning algorithms, we can make machines behave like humans in different situations/environments.

Many machine learning algorithms such as linear regression, logistic regression, decision tree, support vector machine (SVM), k-nearest neighbor (kNN), naive Bayes and random forest are applied to almost any data problem.

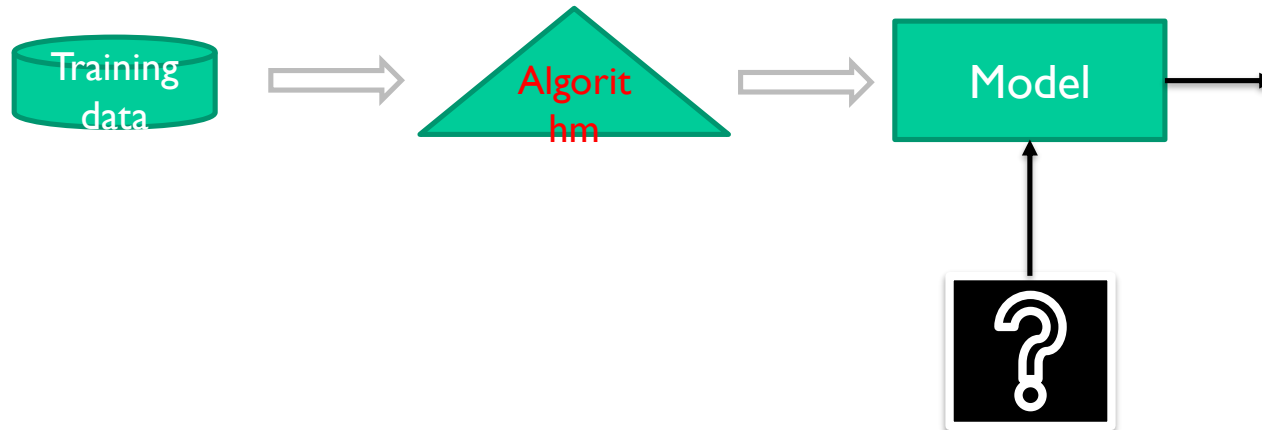


ML taxonomy

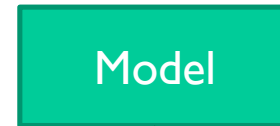


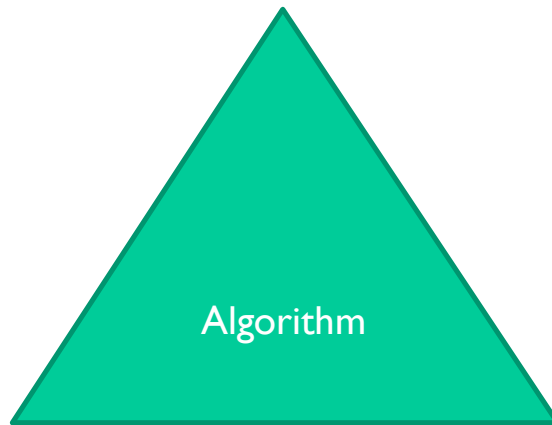
Supervised learning

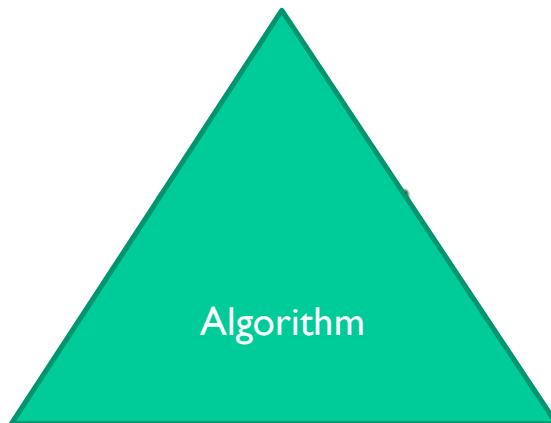
- In supervised learning, the algorithms are presented with a set of classified instances from which they learn a way of classifying unseen instances. When the attribute to be predicted is numeric rather than nominal it is called regression.



Classification

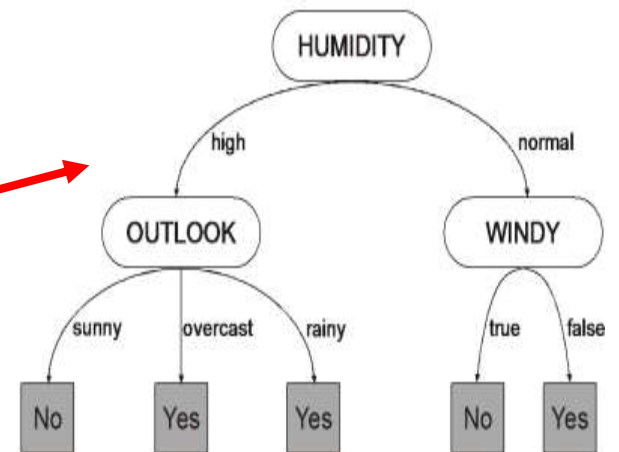






The resulting *model* is also called the *hypothesis*.

Given a model space and an optimality criterion, a *model* satisfying this criterion is sought.



Optimal tree!

Some criteria:

- Maximizing the prediction accuracy
- Minimizing the hypothesis' size
- Maximizing the hypothesis fitness to the input data
- Maximizing the hypothesis interpretability
- Minimizing the time complexity of prediction



Imbalanced data

- Random over/under sampling
- SMOTE
- Cost sensitive classification

You trained a model to **predict cancer** from image data
using a state of the art Hierarchical siamese CNN with
dynamic kernel activations...



-----Your model has an accuracy of 99.9%-----



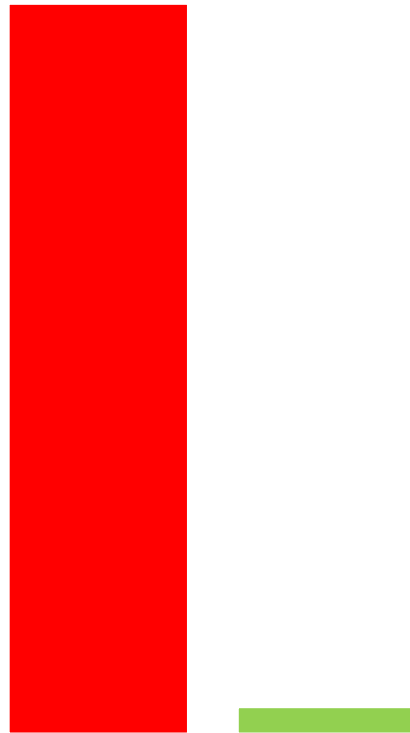
But... WTH!?



By looking at the confusion matrix you realize that the model **does not detect any of the positive examples.**



After plotting your class distribution you see that you have thousands of negative examples but just a couple of positives.



Classifiers try to reduce the overall error so they can be biased towards the majority class.

Negatives = 998

Positives = 2

By always predicting a negative class the **accuracy** will be 99.8% !!



Your dataset is imbalanced.

Now what??



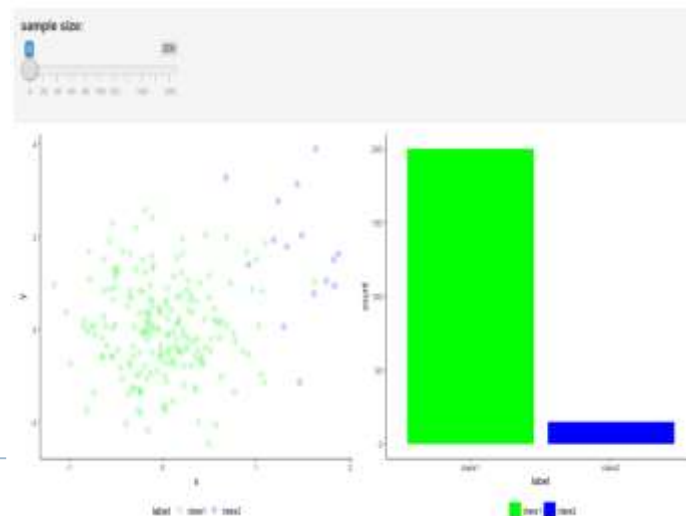
What can you do?

- Collect more data (difficult in many domains)
- Delete data from the majority class
- Create synthetic data
- Adapt your learning algorithm (cost sensitive classification)



Random over/under sampling

- **Random oversampling**: randomly duplicate data points from the minority class.
- **Random undersampling**: randomly delete data points from the majority class.



Problems with these approaches:

- Loss of information (in the case of under sampling)
- Overfitting and fixed boundaries (over sampling)



SMOTE

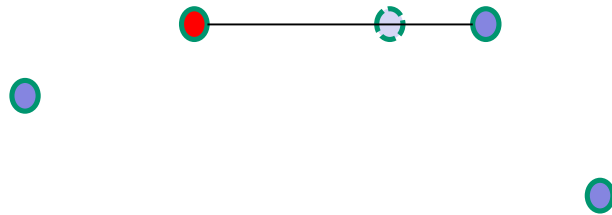
- Synthetic Minority Over-sampling Technique (Chawla).
- Creates new data points from the minority class.
- Operates in the feature space.



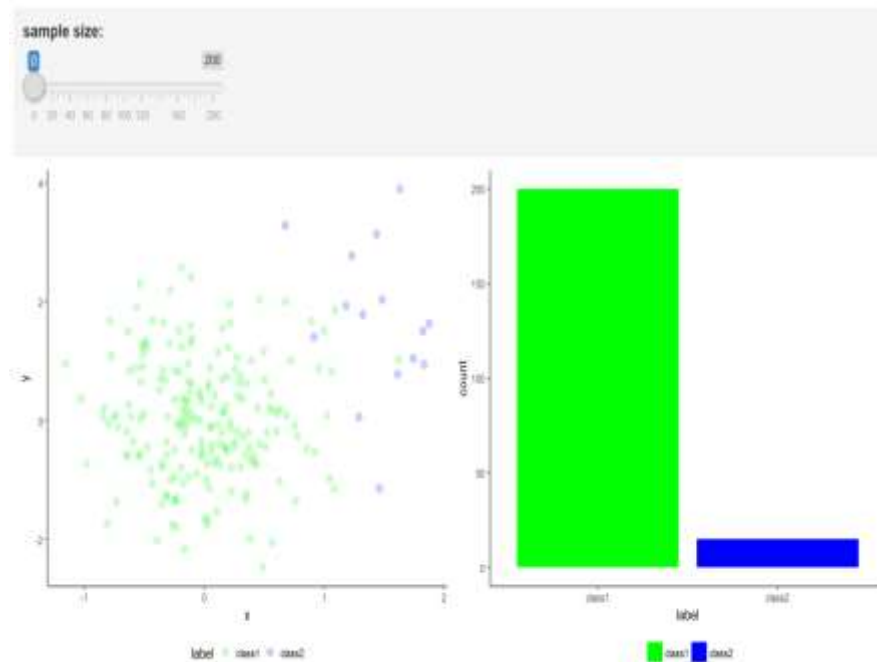
- Main steps:

1. Take the difference between a sample point and one of its nearest neighbors.
2. Multiply the difference by a random number between 0 and 1 and add it to the feature vector.

This causes the selection of a random point along the line segment between two specific features.



SMOTE DEMO



Danger of information injection and overfitting

Do not create synthetic points on the entire dataset before splitting into train/test sets.

- Perform the preprocessing just on the training data!!
- For k-fold cross validation, you have to do it for each fold (just on the training set).



For images:

- Augment training data by applying image transformations: rotate, scale, shift, etc.



- Keras provides functionalities for data augmentation:
<https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>



Type I error
(false positive)



Type II error
(false negative)



TP: true positives

FP: false positives

0	λ
μ	0

FN: false negatives

TN: true negatives

Classify as positive if: probability of positive $> \frac{\mu}{\mu + \lambda}$



Performance metrics

- Most of the time accuracy will not be enough to assess performance.

- $accuracy = \frac{TP+TN}{P+N}$

- $sensitivity = \frac{TP}{P}$

- $precision = \frac{TP}{TP+FP}$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$





Supporting materials: imbalanced data

- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.
- Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1), 25-36.
- Python imbalanced-learn library: <https://github.com/scikit-learn-contrib/imbalanced-learn>
- Weka also has oversampling methods and a cost sensitive meta classifier: <https://weka.wikispaces.com/CostSensitiveClassifier>
- Cost-sensitive classification video: <https://www.youtube.com/watch?v=I9muPldOG30>
- Performance metrics: https://en.wikipedia.org/wiki/Sensitivity_and_specificity



Naive Bayes Classifier

It is a supervised learning algorithm that uses the conditional probability formula as its base:

$$P(A | B) = P(A) * P(B | A) / P(B)$$

This learning model is used to predict the class of a given unknown observation. In naïve Bayes, we calculate the probability of each class for a given unknown observation and we assign the observation to the class that gives the highest probability value. This assumes that predictor variables are independent of each other. In naïve Bayes, $P(A|B)$ is called the posterior probability, $P(B|A)$ is the likelihood and $P(A)$ and $P(B)$ are prior probabilities of proposition and evidence, respectively.

Naïve Bayes states that:

$$\text{Posterior probability} = (\text{Likelihood of event}) \cdot (\text{Prior probability of Proposition}) / \text{Prior probability of Evidence}$$

Bayesian Classification

- A statistical classifier: performs *probabilistic prediction*, i.e., predicts class membership probabilities
- A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers
- Each training example can **incrementally increase/decrease** the probability that a hypothesis is correct — **prior knowledge can be combined with observed data**

Towards Naïve Bayesian Classifier

- Let D be a **training set of tuples** and their associated class labels, and each tuple is represented by an n -D attribute vector $\mathbf{X} = (x_1, x_2, \dots, x_n)$
- Suppose there are m classes C_1, C_2, \dots, C_m .
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\mathbf{X})$

- This can be derived from Bayes' theorem
$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

- Since $P(\mathbf{X})$ is constant for all classes, only needs to be maximized

Naïve Bayesian Classifier: Training Dataset

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data sample

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayesian Classifier: An Example

- $P(C_i)$: $P(\text{buys_computer} = \text{"yes"}) = 9/14 = 0.643$

$$P(\text{buys_computer} = \text{"no"}) = 5/14 = 0.357$$

- Compute $P(X|C_i)$ for each class

$$P(\text{age} = \text{"<=30"} | \text{buys_computer} = \text{"yes"}) = 2/9 = 0.222$$

$$P(\text{age} = \text{"<= 30"} | \text{buys_computer} = \text{"no"}) = 3/5 = 0.6$$

$$P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"yes"}) = 4/9 = 0.444$$

$$P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$$

$$P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"no"}) = 1/5 = 0.2$$

$$P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$$

- $X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$

$$P(X|C_i) : P(X|\text{buys_computer} = \text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$

$$P(X|\text{buys_computer} = \text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$$

$$P(X|C_i) * P(C_i) : P(X|\text{buys_computer} = \text{"yes"}) * P(\text{buys_computer} = \text{"yes"}) = 0.028$$

$$P(X|\text{buys_computer} = \text{"no"}) * P(\text{buys_computer} = \text{"no"}) = 0.007$$

Naïve Bayesian Classifier: Comments

- Advantages
 - Easy to implement
 - Good results obtained in most of the cases
- Disadvantages
 - Assumption: **class conditional independence**, therefore loss of accuracy
 - Practically, dependencies exist among variables
- E.g., hospitals: patients: Profile: age, family history, etc.
Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
- **Dependencies among these cannot be modeled by Naïve Bayesian Classifier**

Naive Bayes Classifier in R

Function and library used for naive Bayes classifier for predicting values

```
> naiveBayes(x1, y1, ...)          ## Default method  
> naiveBayes(formula1, data1, ..., subset1, na.action = na.pass)
```

where

x1 is a numeric matrix/data frame of categorical and/or numeric values or independent variables;

Y1 is a class vector or dependent variable;

formula1 is a formula of the form **class ~ x1 + x2 +**;

data1 is a data frame of predictors.

Other parameters deal with handling missing values or data preprocessing.

We need to install and load the naïveBayes library as follows:

```
>install.packages("naivebayes")  
>install.packages("e1071")  
>library(e1071)
```

Naive Bayes Classifier in R

In this example, we want to use the diabetes data set, which can be downloaded from the UCI repository :

```
>data<- read.csv("/Users/sandhy/Desktop/pima-indians-diabetes.csv")  
> str(data) # displays the structure of data set, variables it contains ,its type  
, values etc.
```

This data set contains eight independent variables: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, Age and DiabetesPedigreeFunction, and one dependent variable, Outcome. The Outcome variable is our target and contains value 0/1 (0 indicates no diabetes, 1 indicates diabetes). This data set has 768 objects of nine variables.

Naive Bayes Classifier in R

#Model building

```
> Naive_Bayes_Model=naiveBayes(Outcome ~., data=data)
> Naive_Bayes_Model # Displays the naive Bayes classifier parameters
> NB_Predictions=predict(Naive_Bayes_Model,data)
> table(NB_Predictions,data$Outcome)    #Prediction on the dataset
```

NB_Predictions False True

False 421 104

True 79 164

The accuracy of this model is about 76%, as calculated using the confusion matrix





What is classification?

A machine learning task that deals with identifying the class to which an instance belongs

A classifier performs classification

(Textual features :
Ngrams)

(Perceptive inputs)

Test instance

Classifier

Category of document?
{Politics, Movies, Biology}

Discrete-valued

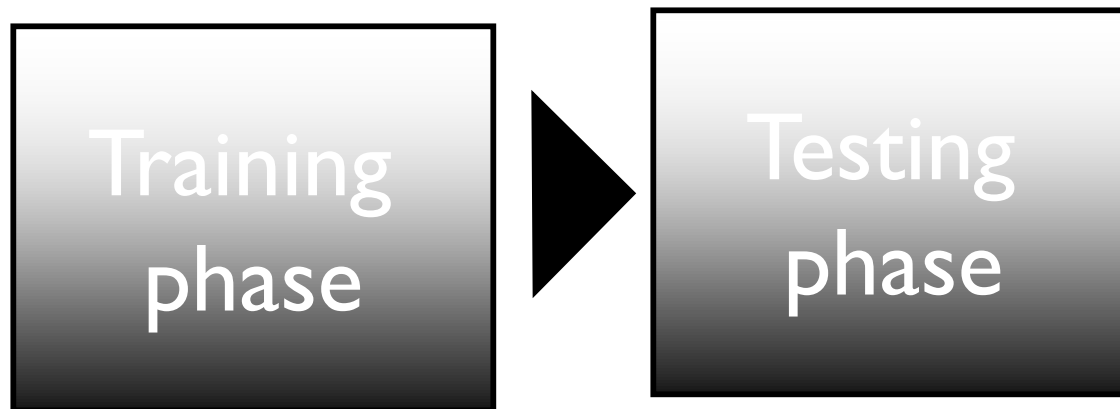
Class label

Attributes

(a_1, a_2, \dots, a_n)



Classification learning





Generating datasets

- Methods:
 - Holdout ($2/3^{\text{rd}}$ training, $1/3^{\text{rd}}$ testing)
 - Cross validation ($n - \text{fold}$)
 - Divide into n parts
 - Train on $(n-1)$, test on last
 - Repeat for different combinations
 - Bootstrapping
 - Select random samples to form the training set





Evaluating classifiers

- Outcome:
 - Accuracy
 - Confusion matrix
 - If cost-sensitive, the expected cost of classification (attribute test cost + misclassification cost)
- etc.



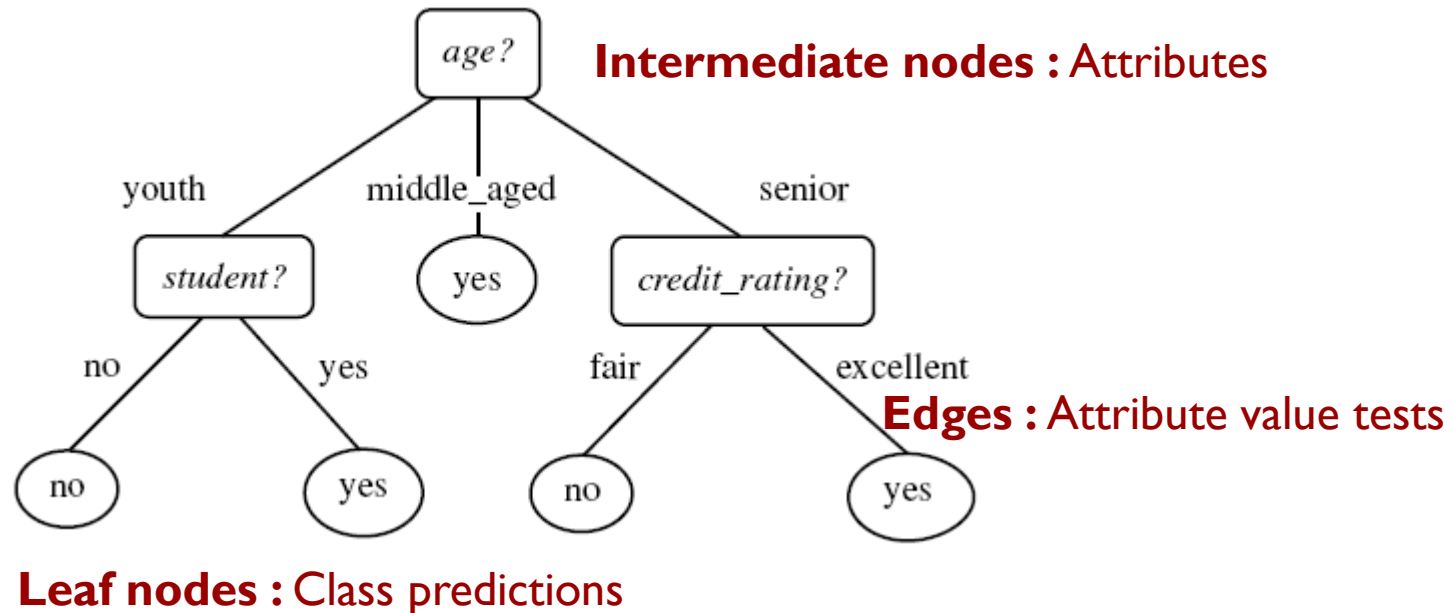


Decision Trees





Example tree

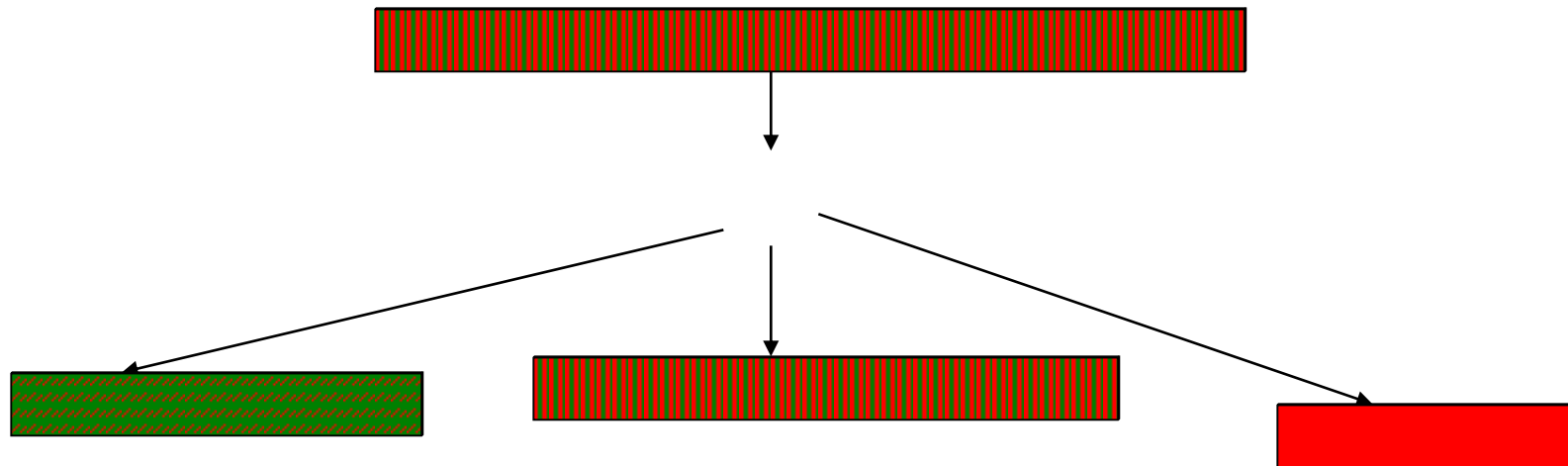


Example algorithms:





Decision Tree schematic



RED

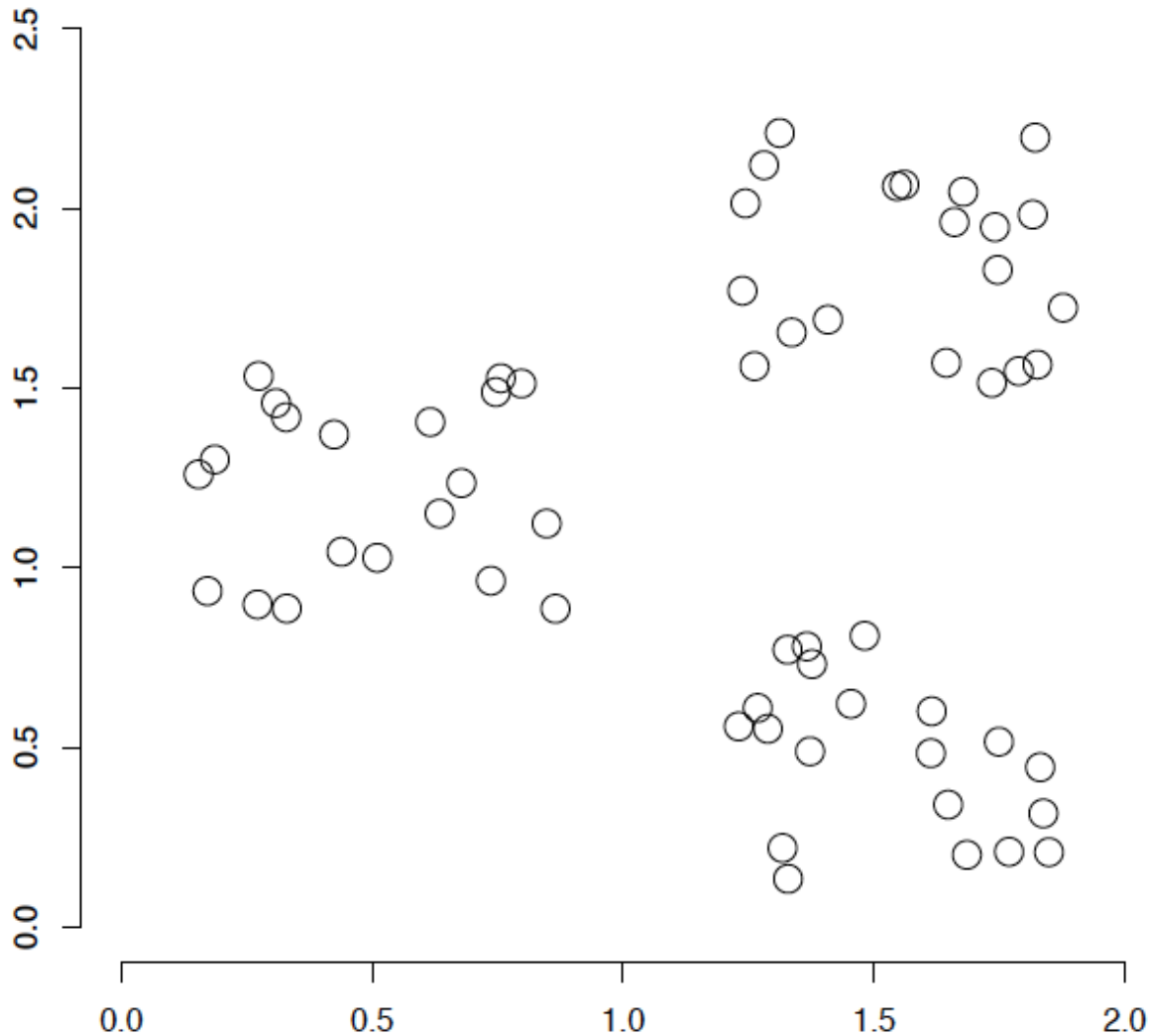


Clustering

- Clustering: the **process of grouping** a set of objects into classes of similar objects
- A common and important task that finds many applications in IR and other places

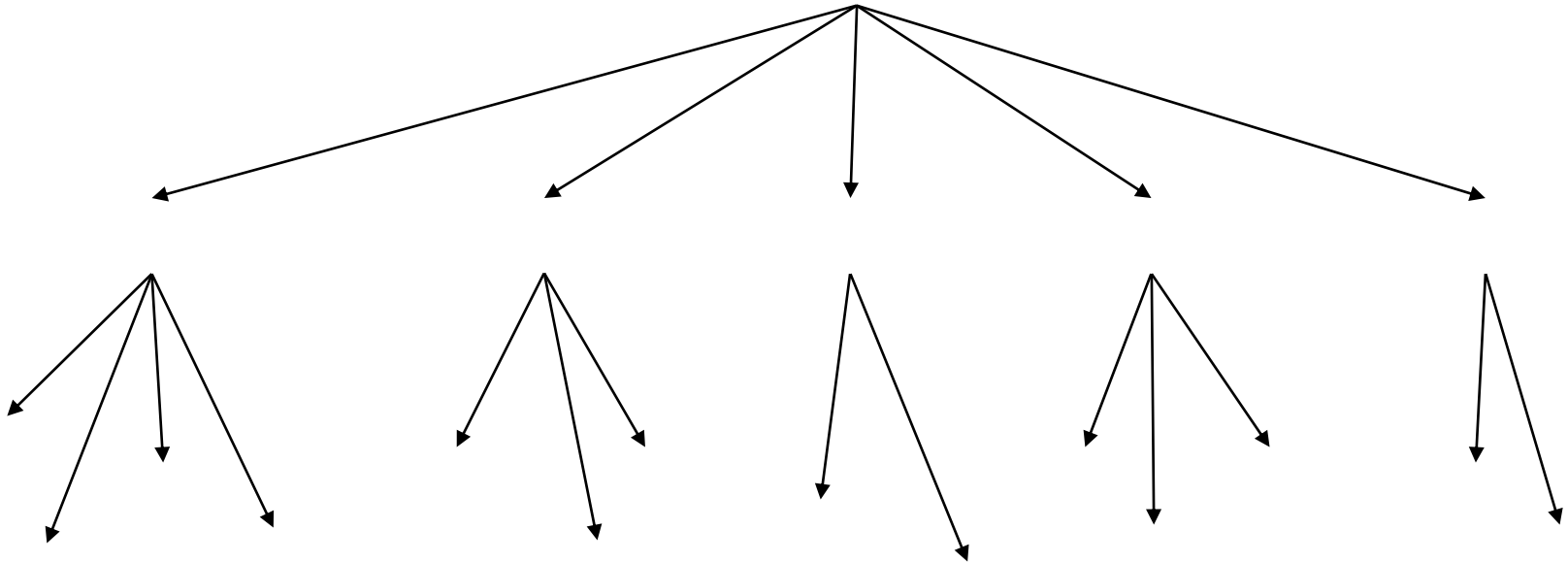


A data set with clear cluster structure



- How would you design an algorithm for finding the three clusters in this case?

Yahoo! Hierarchy *isn't* clustering but *is* the kind of output you want from clustering



Google News: automatic clustering gives an effective news presentation

Google News

http://news.google.com/


World » [edit](#) [X](#)

[Pirates Demand \\$25 Million Ransom for Hijacked Tanker \(Update1\)](#)

Bloomberg - 36 minutes ago

By Caroline Alexander and Hamsa Omar Nov. 20 (Bloomberg) -- Somali pirates are demanding \$25 million in ransom to release an oil-laden Saudi supertanker seized off the East African coast, and called on the ship's owners to pay up "soon."

[Somali pirates demand \\$25M for Saudi ship](#) United Press International
[African Union says Somali politicians fuel piracy](#) Washington Post
[BBC News - guardian.co.uk](#) - [Aljazeera.net](#) - [RIA Novosti](#)
[all 4,015 news articles »](#)




Pakistan protests over US missile strikes

Reuters - 2 hours ago

By Simon Cameron-Moore ISLAMABAD (Reuters) - Pakistan summoned US ambassador Anne Patterson on Thursday to protest over missile strikes launched by pilotless drone aircraft against militant targets in Pakistan.

[Pakistan protests US drone attacks, Taliban warns of reprisals](#) AFP
[Pakistan warns US over missile strike](#) CNN International
[Telegraph.co.uk](#) - [China Daily](#) - [Xinhua](#) - [PRESS TV](#)
[all 560 news articles »](#)




Nighttime attack on Thai antigovernment protesters wounds at least 20

Christian Science Monitor - 30 minutes ago

The government denied attacking demonstrators, who have called for the ouster of the prime minister. By Huma Yusuf One person has been killed and 23 others wounded in a grenade attack Thursday against antigovernment protesters occupying the Thai prime ...

[Blast Kills 1, Wounds 23 at Thai Prime Minister's Office](#) Washington Post
[Anti-government protestor in Thailand dies in grenade attack](#) International Herald Tribune
[Xinhua](#) - [United Press International](#) - [The Associated Press](#) - [AsiaOne](#)
[all 688 news articles »](#)




U.S. » [edit](#) [X](#)

[Top Court in California Will Review Proposition 8](#)

New York Times - 1 hour ago

By JESSE McKINLEY SAN FRANCISCO - Responding to pleas for legal clarity from those on both sides of the issue, the California Supreme Court said Wednesday that it would take up the case of whether a voter-approved ban on same-sex unions was ...

[California Supreme Court to decide fate of Prop. 8 same-sex ...](#)
[San Jose Mercury News](#)
[Prop. 8 gay marriage ban goes to Supreme Court](#) Los Angeles Times
[The Miami Herald](#) - [San Diego Union Tribune](#) - [Indiana Daily Student](#) - [San Francisco Chronicle](#)
[all 1,241 news articles »](#)




Drop That Cigarette, Today Is The Great American Smokeout

dBTechno - 1 hour ago

Washington (dbTechno) - Today marks the annual Great American Smokeout hosted by the American Cancer Society, and is trying to get people all across the US to drop their cigarettes for just one day.

[Great American Smokeout: Time to kick the habit](#) Capital Times
[National Smoke Out Day is Thursday; be a quitter](#) Las Cruces Sun-News
[MPNnow.com](#) - [eMaxHealth.com](#) - [Times Tribune of Corbin](#) - [ABC15.com \(KNXV-TV\)](#)
[all 338 news articles »](#)




Perino: Bush would sign jobless benefits extension

The Associated Press - 47 minutes ago

WASHINGTON (AP) - With weekly jobless claims benefits at a 16-year high, the White House said Thursday that President George W. Bush would quickly sign legislation pending in Congress to provide further unemployment benefits.

[Bush would sign measure to extend jobless benefits](#) Houston Chronicle
[Jobless claims show need for benefits extension: White House](#) AFP
[Washington Times](#) - [Wall Street Journal Blogs](#) - [WOL](#) - [Tampabay.com](#)
[all 599 news articles »](#)



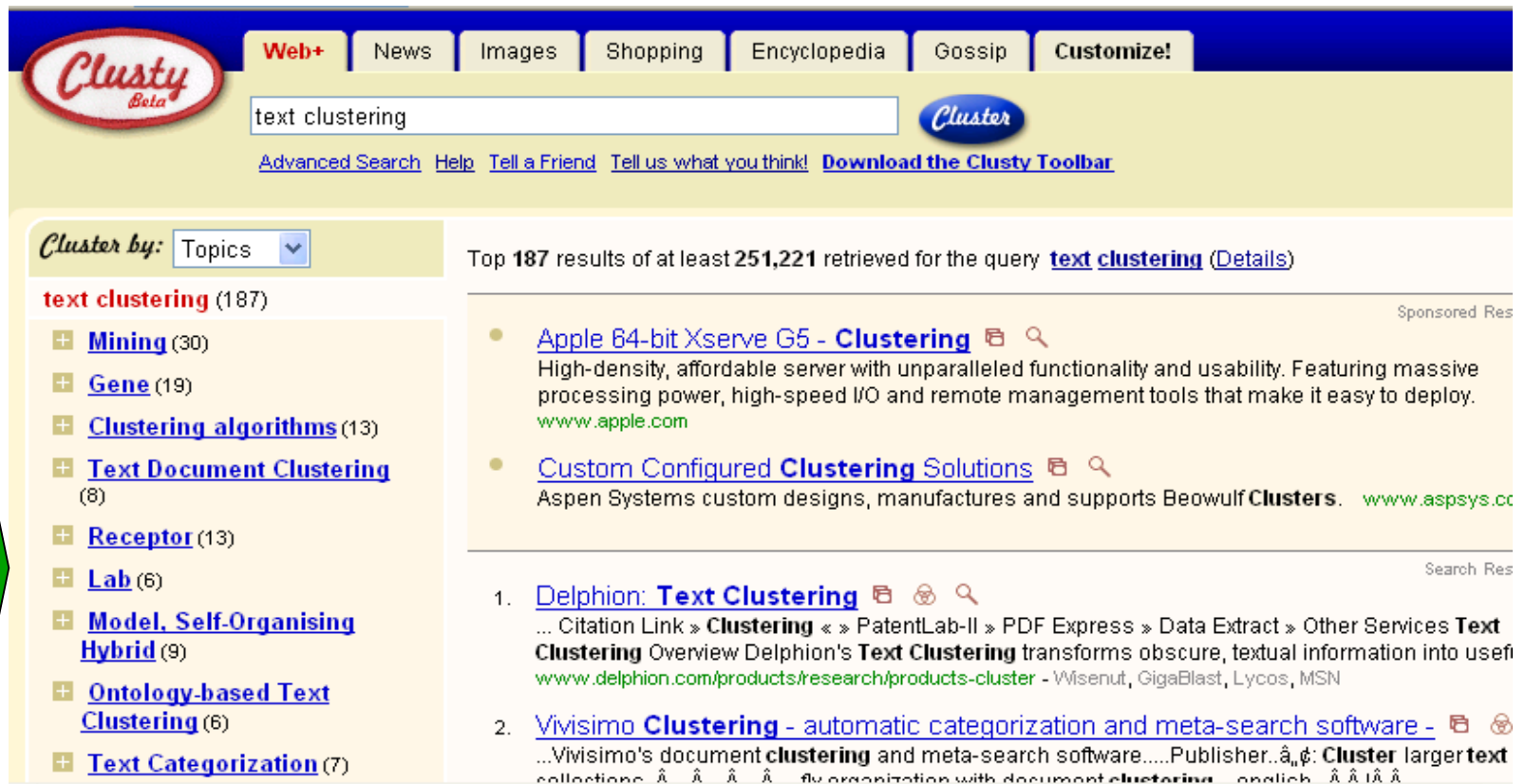
[Show more stories](#) [Show fewer stories](#)

[Show more stories](#) [Show fewer stories](#)

http://www.google.com/hostednews/ap/article/ALeqM5hGjNbxI6O23C8QzqZMY0pGPAik-AD94INLTG1

For better navigation of search results

- For grouping search results thematically
 - clusty.com / Vivisimo



The screenshot displays the Clusty Beta search engine interface. At the top, there is a navigation bar with links for Web+, News, Images, Shopping, Encyclopedia, Gossip, and Customize!. Below this is a search bar containing the text 'text clustering'. To the right of the search bar is a 'Cluster' button. Below the search bar are links for Advanced Search, Help, Tell a Friend, Tell us what you think!, and Download the Clusty Toolbar.

On the left side, there is a section titled 'Cluster by:' with a dropdown menu set to 'Topics'. Below this, a list of topics is shown, each with a plus icon and a count in parentheses:

- [Mining](#) (30)
- [Gene](#) (19)
- [Clustering algorithms](#) (13)
- [Text Document Clustering](#) (8)
- [Receptor](#) (13)
- [Lab](#) (6)
- [Model, Self-Organising Hybrid](#) (9)
- [Ontology-based Text Clustering](#) (6)
- [Text Categorization](#) (7)

A large green arrow points from the left towards the 'Text Document Clustering' topic.

On the right side, the search results are displayed. The top section shows 'Top 187 results of at least 251,221 retrieved for the query [text clustering](#) (Details)'. Below this, there are two sponsored results:

- [Apple 64-bit Xserve G5 - Clustering](#): High-density, affordable server with unparalleled functionality and usability. Featuring massive processing power, high-speed I/O and remote management tools that make it easy to deploy. [www.apple.com](#)
- [Custom Configured Clustering Solutions](#): Aspen Systems custom designs, manufactures and supports Beowulf Clusters. [www.aspsys.cc](#)

Below the sponsored results, there are two search results:

- [Delphion: Text Clustering](#): ... Citation Link » **Clustering** « » PatentLab-II » PDF Express » Data Extract » Other Services **Text Clustering** Overview Delphion's **Text Clustering** transforms obscure, textual information into usefu [www.delphion.com/products/research/products-cluster](#) - Wisenut, GigaBlast, Lycos, MSN
- [Vivisimo Clustering - automatic categorization and meta-search software -](#): ...Vivisimo's document **clustering** and meta-search software.....Publisher..â„¢: **Cluster** larger **text** collections. â„¢ â„¢ â„¢ â„¢ fly organization with document **clustering**... english. â„¢ â„¢ â„¢

Issues for clustering

- Representation for clustering
 - Need a notion of similarity/distance
- How many clusters?
 - Fixed a priori?
 - Completely data driven?

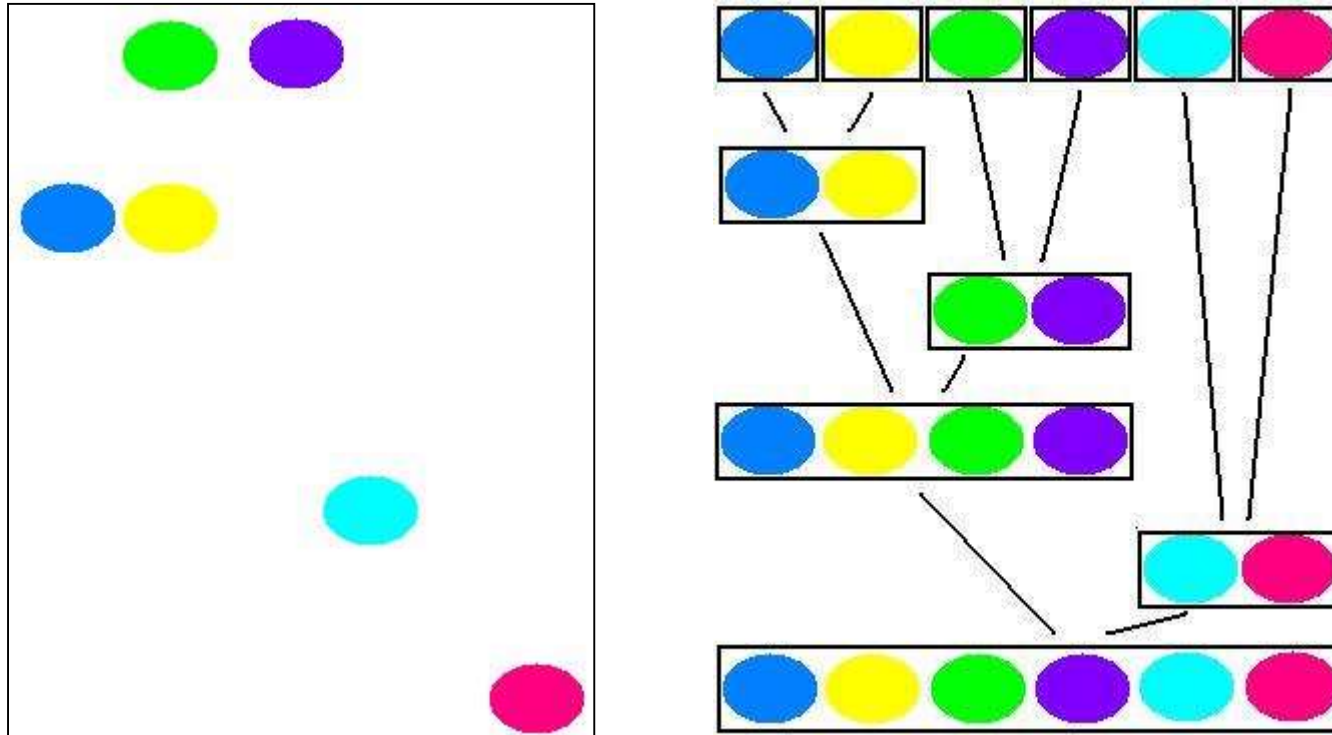


Clustering Algorithms

- Flat algorithms
 - Usually start with a random (partial) partitioning
 - Refine it iteratively
- Hierarchical algorithms
 - Bottom-up/agglomerative
 - (Top-down/divisive)



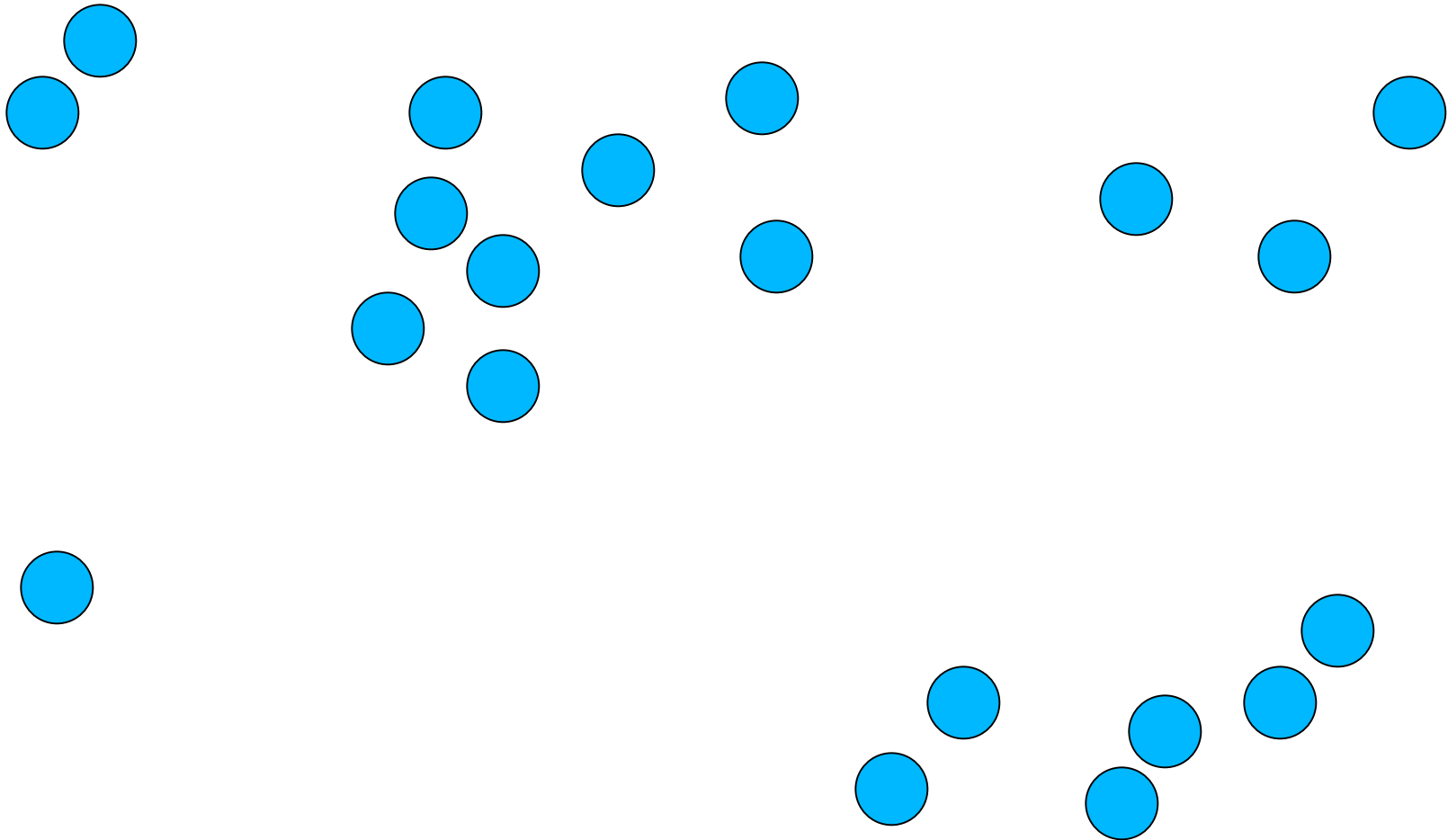
Hierarchical Clustering



- Builds or breaks up a hierarchy of clusters.

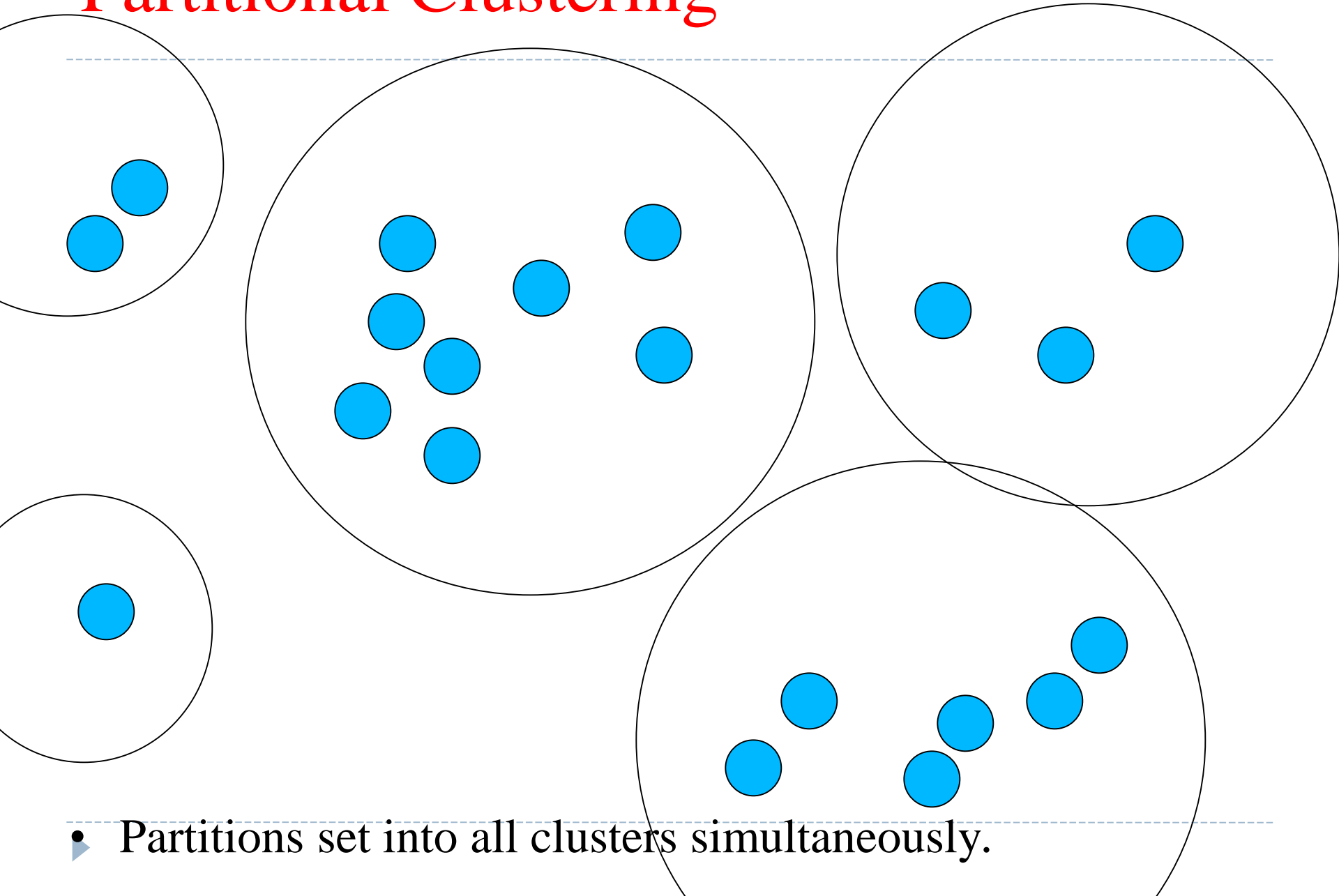


Partitional Clustering



-
- Partitions set into all clusters simultaneously.

Partitional Clustering



- Partitions set into all clusters simultaneously.

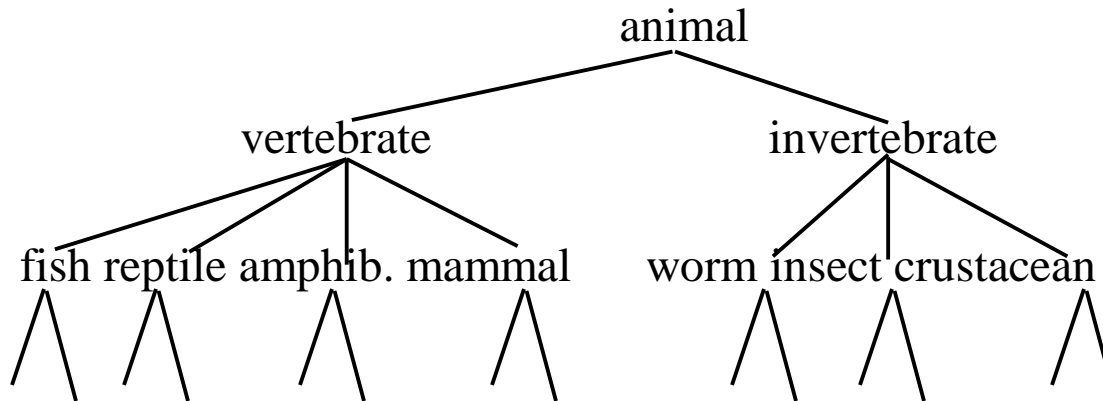
Hard vs. soft clustering

- Hard clustering: Each data belongs to exactly one cluster
- Soft clustering: A data can belong to more than one cluster.



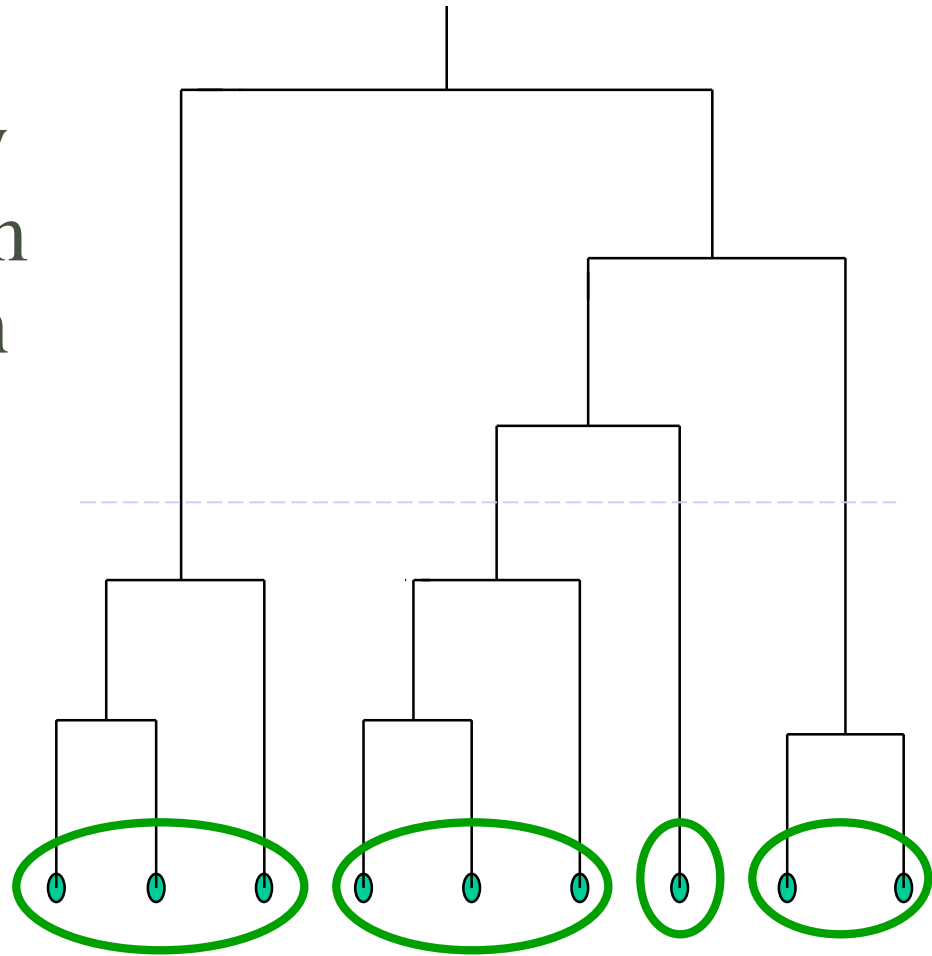
Hierarchical Clustering

- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of data.



Dendrogram: Hierarchical Clustering

- Clustering obtained by cutting the dendrogram at a desired level: each **connected** component forms a cluster.



Hierarchical Agglomerative Clustering (HAC)

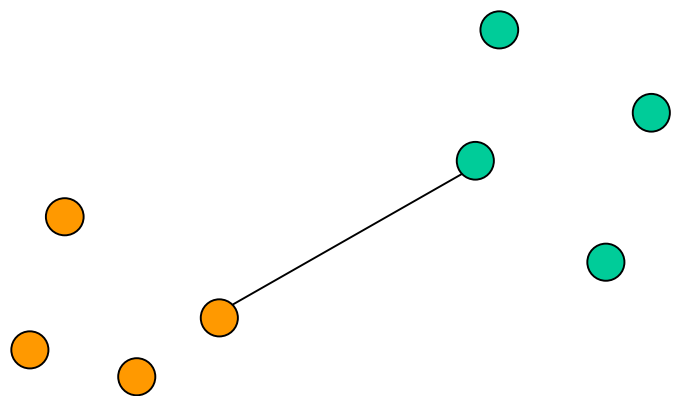
- Starts with each data in a separate cluster
 - then repeatedly joins the *closest pair* of clusters, until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.

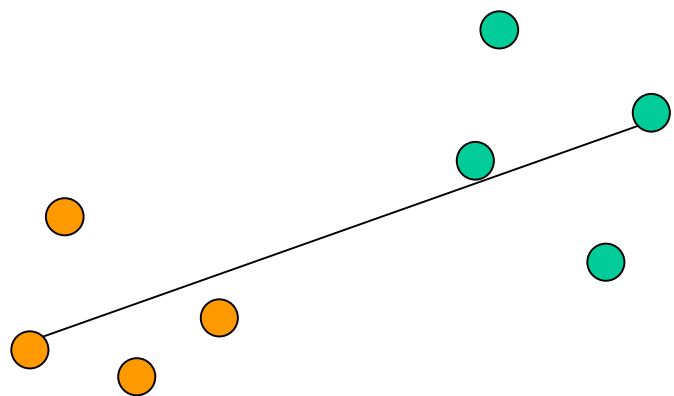


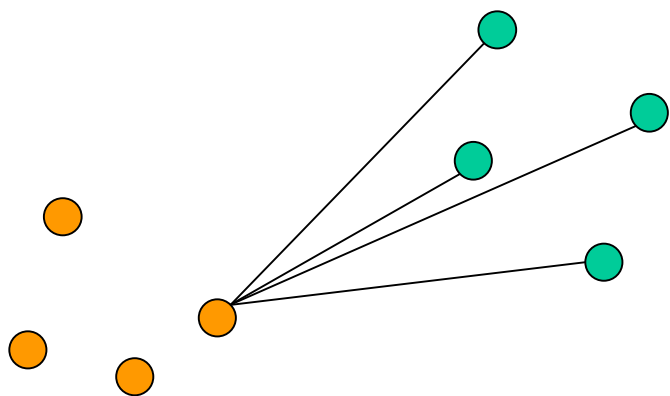
Closest pair of clusters

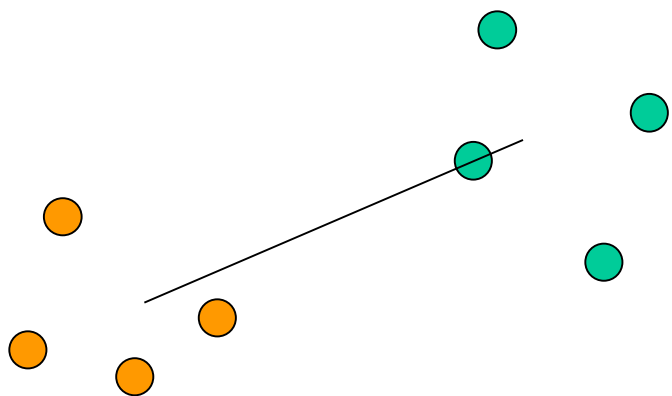
- Many variants to defining closest pair of clusters
- **Single-link**
 - Similarity of the *most* cosine-similar (single-link)
- **Complete-link**
 - Similarity of the “furthest” points, the *least* cosine-similar
- **Average-link**
 - Average cosine between pairs of elements











Single Link Agglomerative Clustering

- Use maximum similarity of pairs:

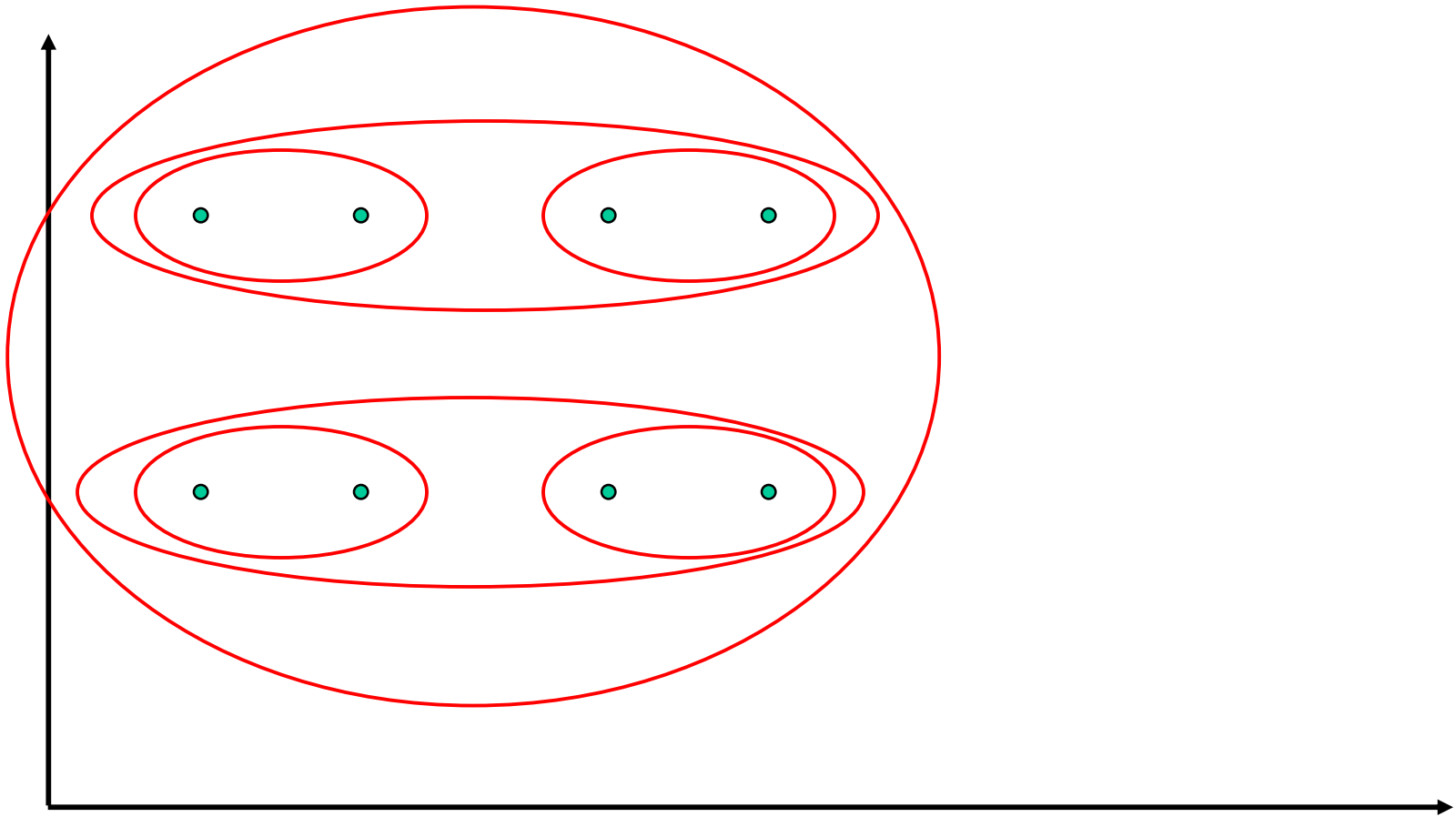
$$\textit{sim}(c_i, c_j) = \max_{x \in c_i, y \in c_j} \textit{sim}(x, y)$$

- Can result in “straggly” (long and thin) clusters due to chaining effect.
- After merging c_i and c_j , the similarity of the resulting cluster to another cluster, c_k , is:

$$\textit{sim}((c_i \cup c_j), c_k) = \max(\textit{sim}(c_i, c_k), \textit{sim}(c_j, c_k))$$



Single Link Example



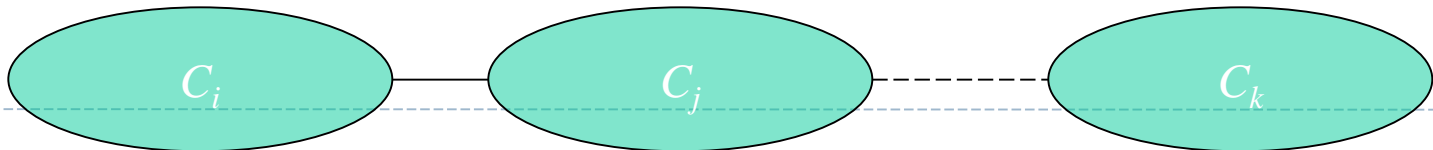
Complete Link Agglomerative Clustering

- Use minimum similarity of pairs:

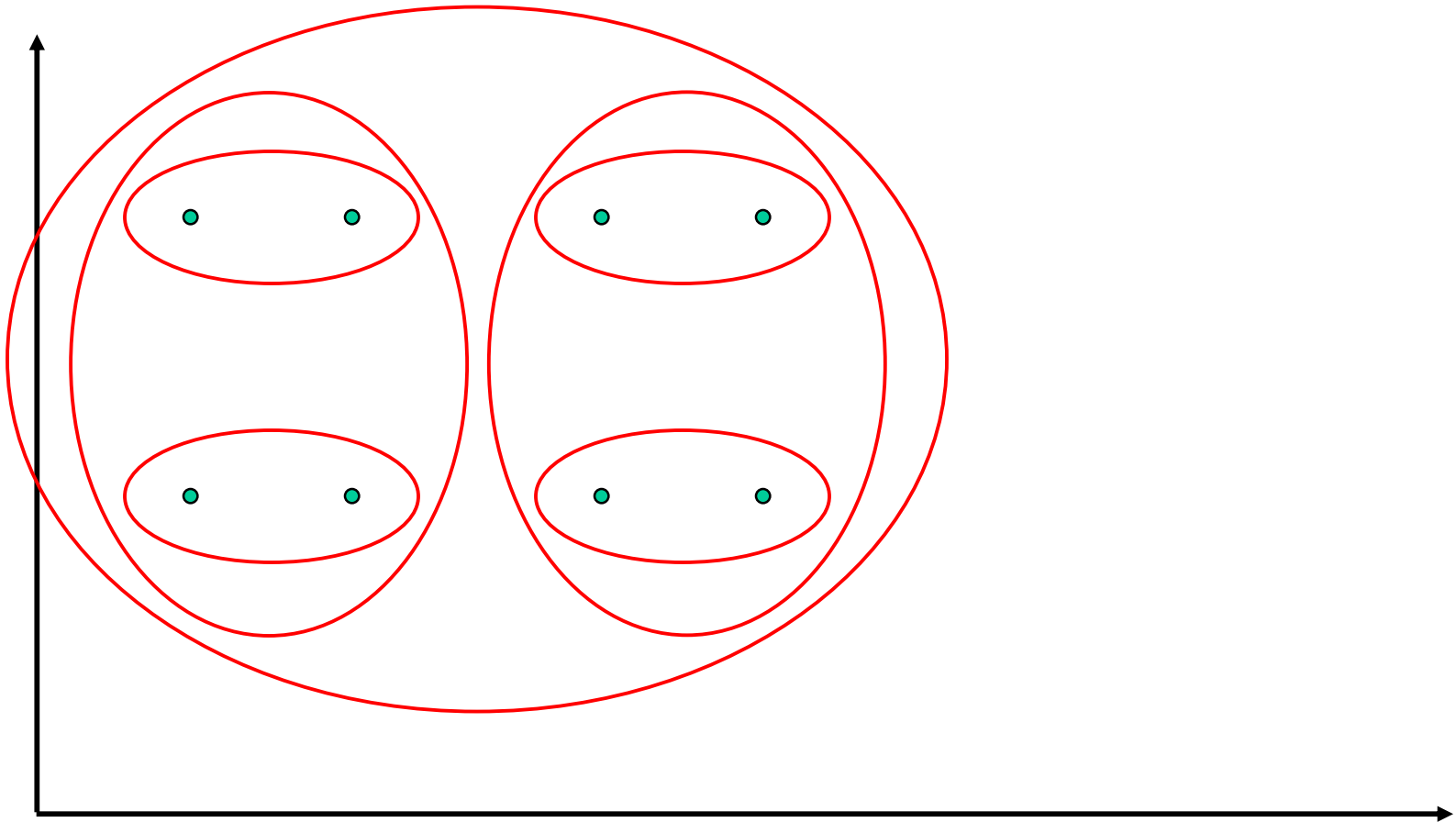
$$\text{sim}(c_i, c_j) = \min_{x \in c_i, y \in c_j} \text{sim}(x, y)$$

- Makes “tighter,” spherical clusters that are typically preferable.
- After merging c_i and c_j , the similarity of the resulting cluster to another cluster, c_k , is:

$$\text{sim}((c_i \cup c_j), c_k) = \min(\text{sim}(c_i, c_k), \text{sim}(c_j, c_k))$$



Complete Link Example



Partitioning Algorithms

- Partitioning method: Construct a partition of n data into a set of K clusters
- Given: a set of points and the number K
- Find: a partition of K clusters that optimizes the chosen partitioning criterion



K-Means

- Assumes data are real-valued vectors.
- Clusters based on *centroids* of points in a cluster, c :
- Reassignment of instances to clusters is based on distance to the current cluster centroids.
$$\vec{\mu}(c) = \frac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$$



K-Means Algorithm

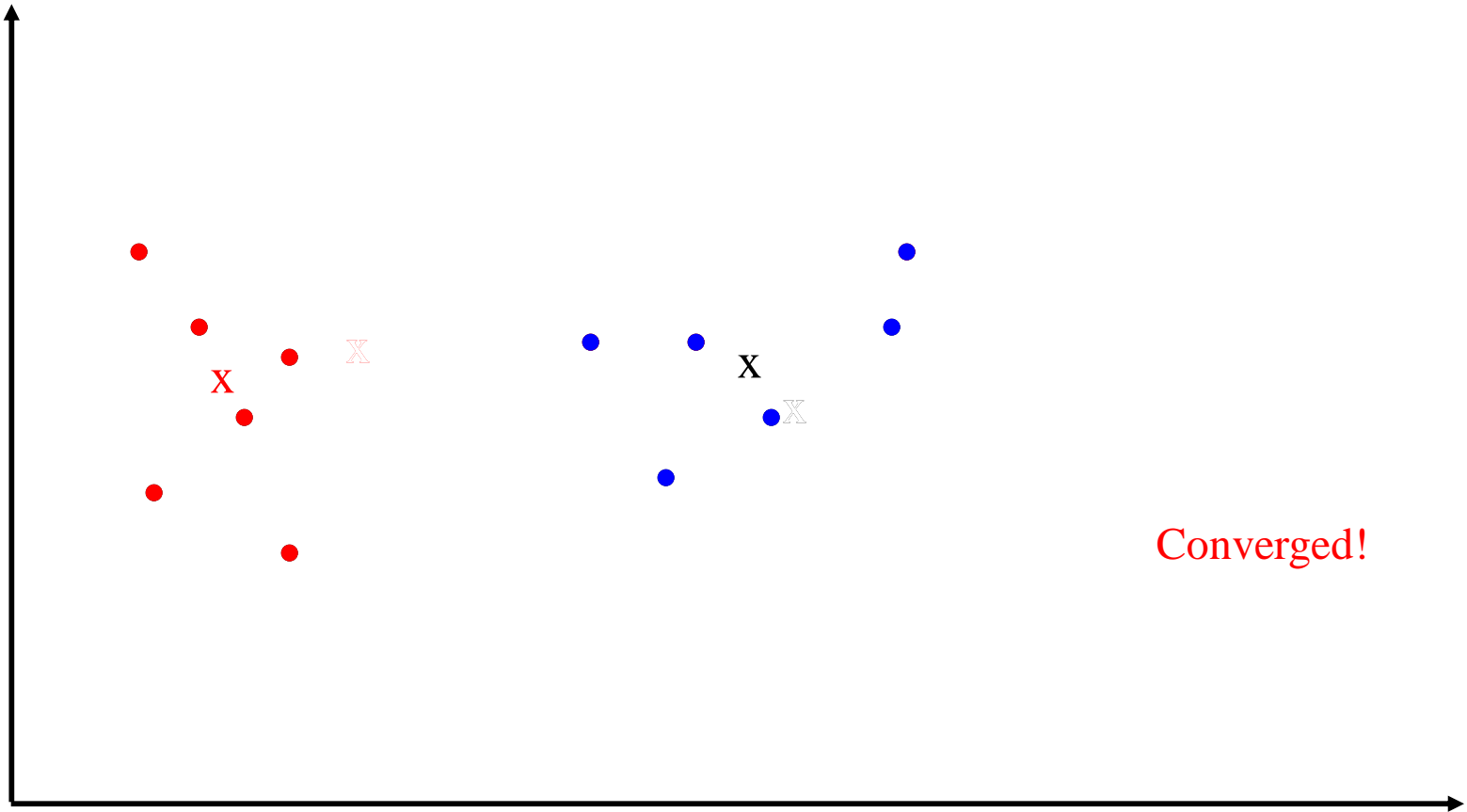
K random data $\{s_1, s_2, \dots, s_K\}$ as seeds.

(Update the seeds to the centroid of each cluster)



K Means Example

($K=2$)



Convergence

- K -means algorithm ever reach a *fixed point*?
 - A state in which clusters don't change.



Seed Choice

- Results can vary based on random seed selection.
 - Select good seeds using a heuristic (e.g., data least similar to any existing mean)



What Is A Good Clustering?

- Internal criterion: A good clustering will produce high quality clusters in which:
 - the intra-class (that is, intra-cluster) similarity is high
 - the inter-class similarity is low
 - The measured **quality** of a clustering depends on both the **data representation** and the **similarity measure** used



Decision Tree Classifier

Decision tree is a type of supervised learning algorithm that can be used for categorical or continuous data and where the data is continuously split according to a certain parameter. The parameters are determined based on information gain. The leaves of the tree are the decisions or final outcomes. The decision nodes are where the data is split on the basis of a parameter.

Decision Tree Classifier in R

To build the model using a decision tree, we can use the `rpart()` function, whose syntax is as follows:

```
>rpart(formula, data=, method='')
```

where `formula` indicates the function for prediction;

`data` represents the data specified; and

`method` could be

-class (for a classification tree for categorical data) or

-anova (for a regression tree for contiguous data).

Example here illustrate the use of decision trees, for the prediction of the **birth weight of infants**. The data set **birthwt** is part of the **MASS** library. We need to install and load the **MASS** and **rpart** libraries for this problem.

```
>install.package("rpart")
```

```
>library(MASS)
```

```
>library(rpart)
```

Decision Tree Classifier in R

Implementation in R:

```
>head(birthwt)      # displays first 6 records of data set  
>str(birthwt)       # displays the structure of birthwt data set
```

It has columns such as low, age, lwt, race, smoke, ptl, ht, uti, ftv and bwt. These indicate whether the birth weight is <2.5 kg (1 indicates weight <2.5 kg, 0 indicates weight >2.5 kg), the age of the mother (in years), the weight of the mother, the race of the mother, the smoking status of the mother, number of instances of premature labour of the mother, hypertension, presence of uterine irritability, physicians visited and birth weight.

Decision Tree Classifier in R

Implementation in R:

Let us try to build a prediction model or decision tree model for this data. First, we must convert all the categorical variables to factors, as all the variables are numeric.

```
>cols <- c('low', 'race', 'smoke', 'ht', 'ui')  
>birthwt[cols] <- lapply(birthwt[cols], as.factor)
```

Next, we need to split the data set into training and testing sets.

```
>set.seed(2)  
>train <- sample(1:nrow(birthwt), 0.80 * nrow(birthwt))
```

Decision Tree Classifier in R

Implementation in R:

```
>bTree <- rpart(low ~ . - bwt, data = birthwt[train, ], method = 'class') #build the model
```

Exclude bwt from the model as it is our dependent variable for the purposes of classification. We must specify method = 'class' for the classification task. We can then visualize the tree and associate it with proper text for appropriate interpretation.

```
>plot(bTree)
```

```
>text(bTree, pretty = 0)
```

Decision Tree Classifier in R

Implementation in R:

#Model performance

```
>table(bPred, birthwt[-train, ]$low)
```

```
bPred 0 1
```

```
0 25 6
```

```
1 3 4
```

Hence, the accuracy is $(25 + 4) / (25 + 4 + 6 + 3) = 76.31\%$. Here, the accuracy is measured using the formula:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The k-Nearest Neighbour Method

The k-Nearest Neighbour method is a supervised form of the classification algorithm. It takes labelled training cases as input and classifies new cases based on similarities that can be measured using many different methods such as Euclidean, Manhattan distance, and so on.

For a new instance, the classification is carried out by searching the entire training set and identifying the k most similar cases (called neighbours of the new case). **The new instance will be assigned to the most probable class of these k neighbours.** The k value is usually considered as an odd value, which helps in summarising the class of the new data.

The k-Nearest Neighbour Method

knn() function is used to model KNN in R. The syntax of the knn() function is:

```
>knn(trainset, testset, cl, kneighbours = 1, l = 0, prob = FALSE, use.all = TRUE)
```

where

trainset is the data frame or matrix of the training data,

testset is the data frame or matrix of the test data; a single test data is specified as a row vector,

cl is the factor of true classifications of the training set,

kneighbours represents the number of neighbours to be considered for classification,

l is the minimum vote for a definite decision (otherwise doubt),

prob is a logical value (TRUE indicates that the proportion of the votes for the winning class are returned as attribute prob) and

use.all is a logical value used to handle ties in decision making.

The k-Nearest Neighbour Method

knn() function is used to model KNN in R. We need FNN package installed for this. Here in this example we are making use of iris dataset

```
>install.packages("FNN")  
>library(class)
```

#To generate a random_sample value which has 90% of data set values.

```
> random_sample <- sample(1:nrow(iris), 0.9 * nrow(iris))
```

#iris_norm1 contains 1 to 4 columns from iris data set (90% samples of iris data set) as training samples

```
> iris_norm1 <- as.data.frame(iris[,c(1,2,3,4)])  
> iris_train <- iris_norm1[random_sample,]
```

#iris_test contains remaining 10% samples as test samples

```
> iris_test <- iris_norm1[-random_sample,]
```

The k-Nearest Neighbour Method

#iris_target_category contains the target class of training samples. Target class is in 5th column of training data samples

```
> iris_target_category <- iris[random_sample,5]
```

#iris_test_category contains the target class of training samples. Test class is in 5th column of test data samples

```
> iris_test_category <- iris[-random_sample,5]
```

#build knn model

```
> knnmodel <- knn(iris_train,iris_test,cl=iris_target_category,k=13)
```

The k-Nearest Neighbour Method

```
#calculate the accuracy of model for test data samples  
> tabpred <- table(knnmodel,iris_test_category)  
> accuracy <- function(d){sum(diag(d)/(sum(rowSums(d)))) * 100}  
> accuracy(tabpred)
```

[1] 96.66667

The accuracy of this model is 96.66%. We can normalise columns 1 to 4 as they are the predictor variables. The kNN method works on numeric as well as categorical variables.

K-means Clustering

This **unsupervised method clusters input data in different groups based on some similarity**. The number of clusters in which we want our data to be grouped is specified. Initially, each observation is randomly assigned to a cluster, and the centroid of each cluster is calculated. Then, the algorithm iterates through the following steps:

- 1) Calculate the distance of each data point with the calculated cluster centroid and reassign the data point to the closest cluster.
- 2) Calculate the new centroid of the cluster.

These two steps are repeated till there are no further changes in the cluster; that is, the within- cluster variation cannot be reduced any further.

K-means Clustering

The kmeans() function is used for clustering using k-means in R.

The syntax of this function is as follows:

```
> kmeans(x1, k, iter.max = 10, nstart=1,.....)
```

where

x1 represents the matrix or data frame containing the numeric values,

k represents the number of clusters,

iter.max specifies the maximum number of iterations and

nstart represents how many random sets should be chosen.

K-means Clustering

For example, we are interested in the price and sales of an item. We can see that if price is high, the sales is low, and vice-versa. Hence, the data can be classified within two categories. So, if we apply K-means for two clusters, we obtain the following results:

```
> price <- c(10,20,30,102,105,119)
```

```
> sale <- c(5,6,4,1,2,1)
```

```
> df <- data.frame(price,sale)
```

```
> df
```

```
  price sale
```

```
1   10    5
```

```
2   20    6
```

```
3   30    4
```

```
4  102    1
```

```
5  105    2
```

```
6  119    1
```

K-means Clustering

```
> knnmodel <- kmeans(df,2,nstart=25)
> knnmodel
```

K-means clustering with two clusters of sizes 3 and 3.

Cluster:

	price	sale
1	108.6667	1.333333
2	20.0000	5.000000

Clustering vector:

```
[1] 2 2 2 1 1 1
```

Within-cluster sum of squares by cluster:

```
[1] 165.3333 202.0000
```

(between_SS / total_SS = 97.0 %)

Available components:

[1]	"cluster"	"centers"	"totss"	"withinss"	"tot.withinss"
[6]	"betweenss"	"size"	"iter"	"ifault"	