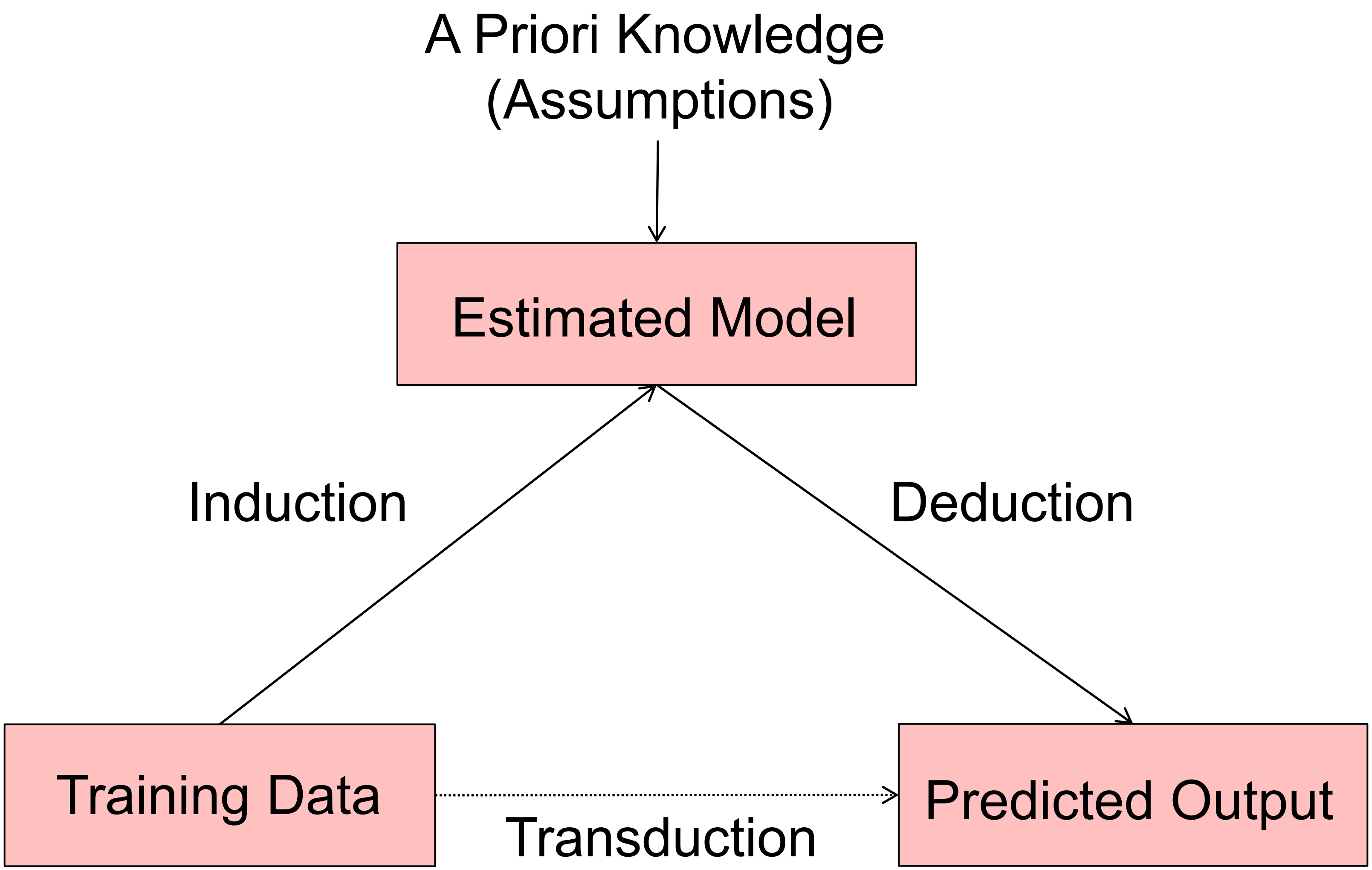
Data Mining

Lecture 4
Learning from Data towards Data Warehouses

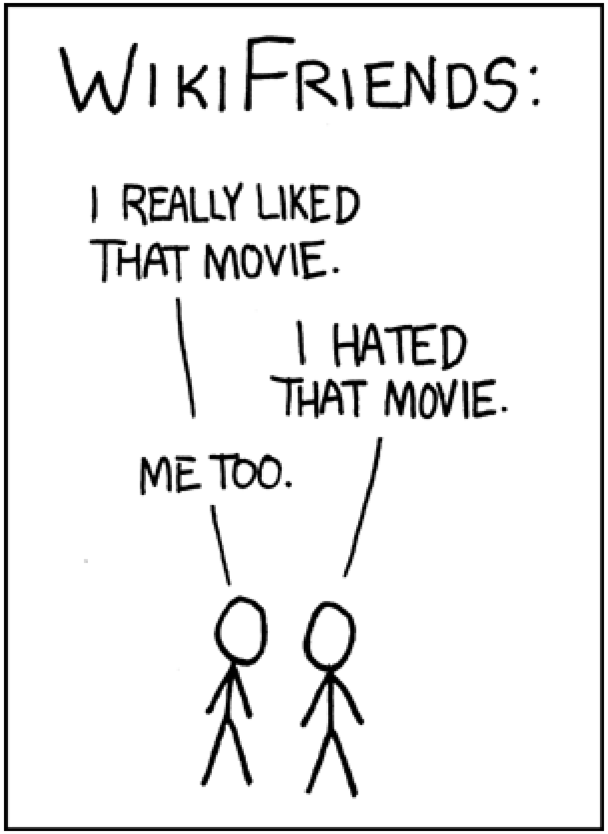
http://www.informatik.uni-hamburg.de/WTM/

Types of Inference: Induction, Deduction, Transduction



Machine Learning & Human Learning

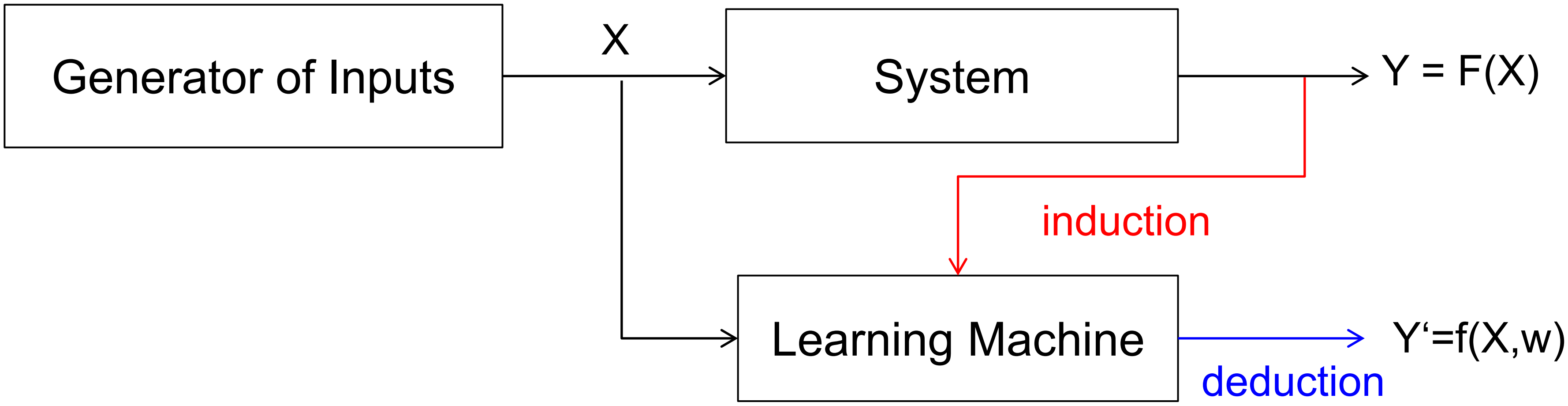
- Supervised, unsupervised, semi-supervised, reinforcement, active learning
- Learning from examples
- Case-based learning
- Learning by analogy
- Learning by doing
- Template-based learning
- …



Machine Learning Issues

- Static vs. dynamic data
- Centralized vs. distributed data
- Incremental (on-line) vs. batch learning
- Adaptive learning
- Life-long learning
- …

A Learning Machine



Given: observed samples {(X, Y)}

How to select f(X, w):

- Approximating function f?
- Parameters: w?
- Hyperparameters?

← A priori knowledge required!

Example

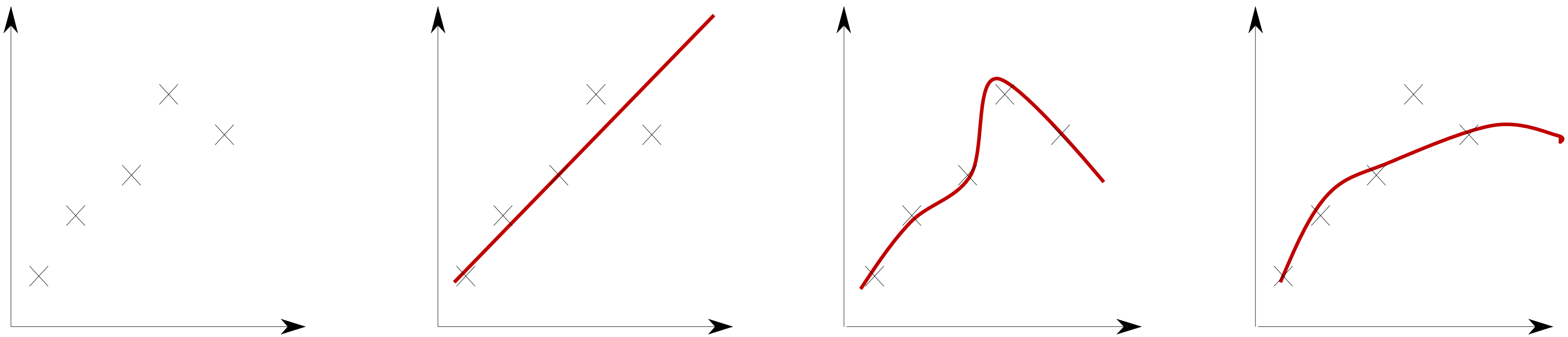
f: linear in parameters:

$$y = w_1 x^n + w_2 x^{n-1} + \cdots + w_0$$

nonlinear in parameters:

$$y = e^{-wx}$$

Hypotheses for a Given Data Set



Polynomial (linear, quadratic, etc.) or exponential model?

How to Learn with a Learning Machine? (1)

- Inductive principle:
  - Tell us ***what*** to do with the data (general prescription)
  - E.g. by defining a cost function such as: ERM, SRM, …
- Learning method:
  - Tell us ***how*** to obtain an estimate
  - I.e. a constructive implementation of an inductive principle

How to Learn with a Learning Machine? (2)

- Loss function (also: error function) L(y, f(X, w)):
  - Measure of a difference between y_i and f(X_i, w) for each sample.

With:

y:	The output produced by the system, and
***X*:**	a set of inputs, and
f(X, w):	The output produced by the learning machine for a selected approximating function, and
w:	the set of parameters in the approximating functions.

- Risk function R(w):
  - Measure of accuracy of the learning machine.

With:

R(w) = $\iint$ L(y, f[X, w]) p(X, y) dX dy

p(X,y): probability distribution of samples.

How to Learn with a Learning Machine? (3)

- Examples of loss function L(y, f(X, w)):
  - Classification error:
    - $L(y, f(X,w)) = \begin{cases} 0, if\ y = f(X,w) \\ 1, if\ y \neq f(X,w) \end{cases}$
  - Squared error measure for regression:
    - $L(y, f[X, w]) = (y - f[X, w])^2$

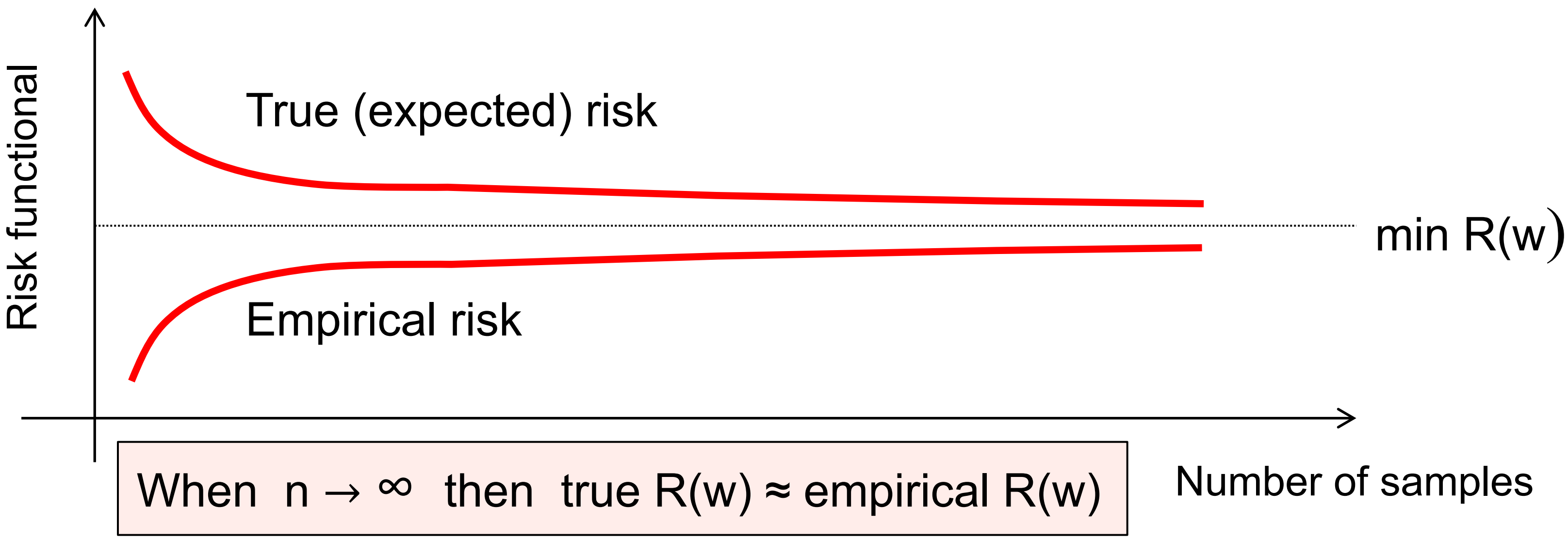
Statistical Learning Theory (SLT) (1)

- SLT = VC theory (**V**apnik **C**hervonenkis):
  for estimation with small (finite) sets of samples.
  - Optimal estimate = minimum of risk function R(w)
  - Exact distribution of data p(X, y) is unknown
  - Approximate computation of true R(w) with empirical R(w)
    - ***Empirical Risk Minimization (ERM)*** – the basic inductive principle
    - Implementation of ERM depends on selected L and f(X, w)

- SLT – formalizes many learning procedures developed in AI, ANN, statistics, Data Mining, Pattern Recognition.

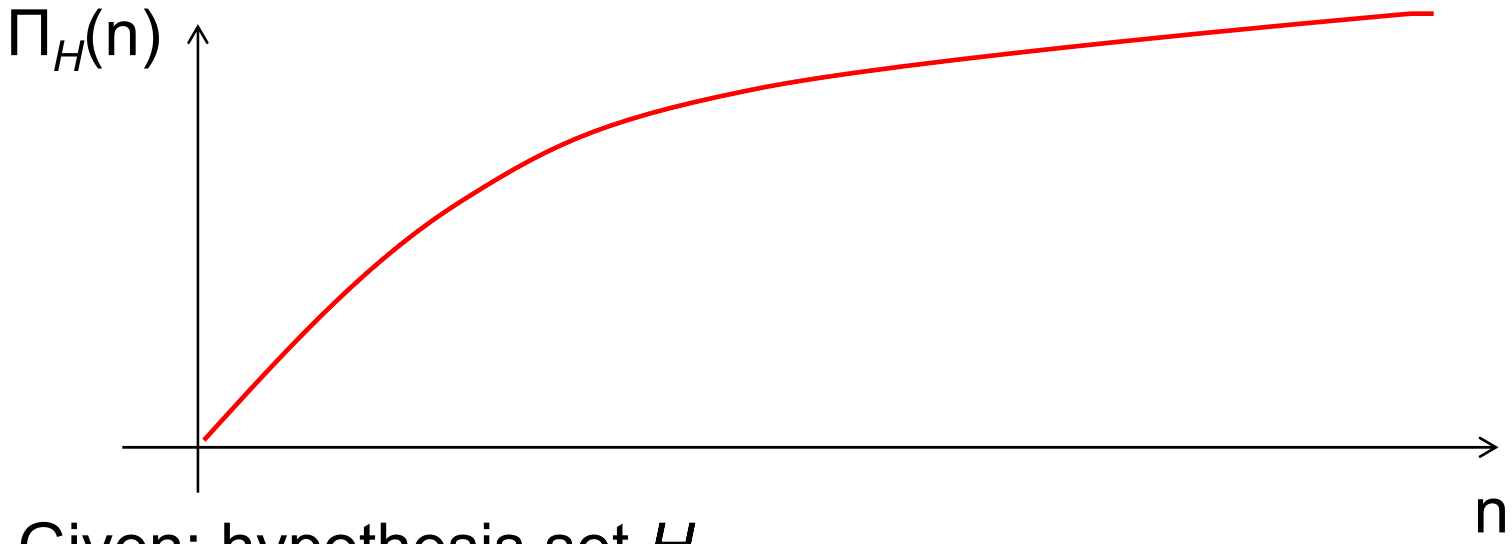
Statistical Learning Theory (2)

- Asymptotic Consistency of ERM:



- Nontrivial consistency: AC should hold for ALL classes of approximating functions.
- Approximating functions should be in a form of a ***growth function***.

Growth Function

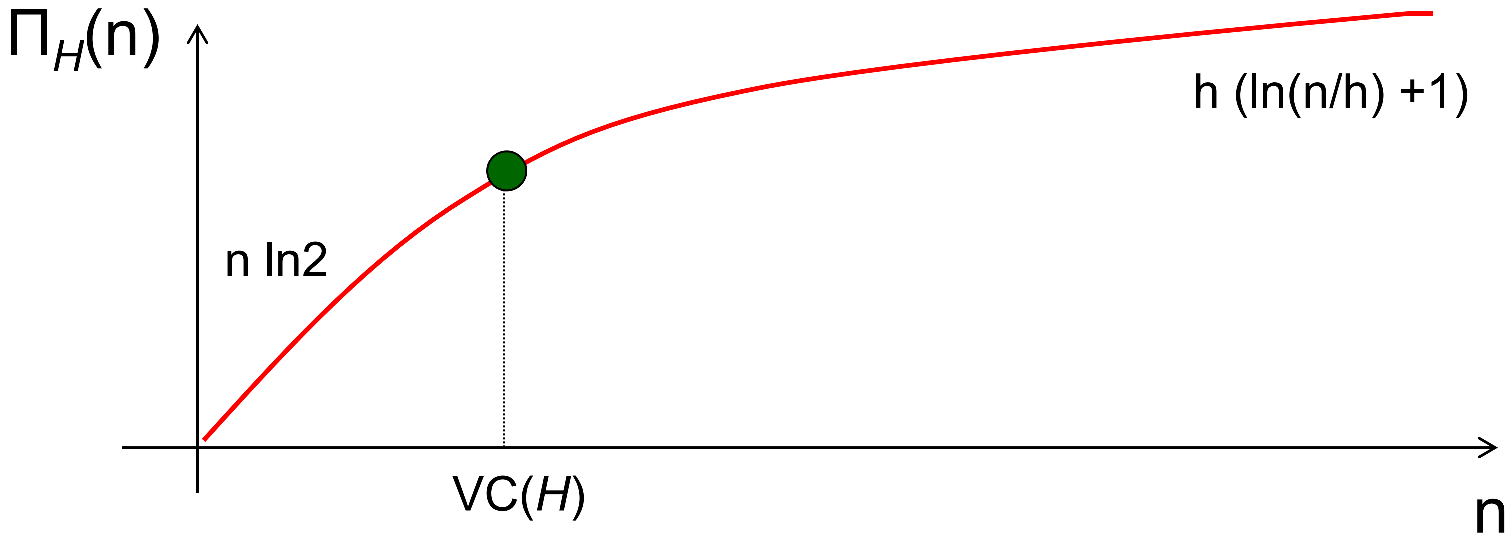


- Given: hypothesis set *H,*
  i.e. all the functions a learner can approximate
- A growth function is defined as

  $\Pi_H(n) = \max |\Pi_H(S)|$ over all input sets S of size n

  i.e. the maximum number of ways n points can be classified by *H*
- E.g. binary classification: $\Pi_H(n) \leq 2^N$

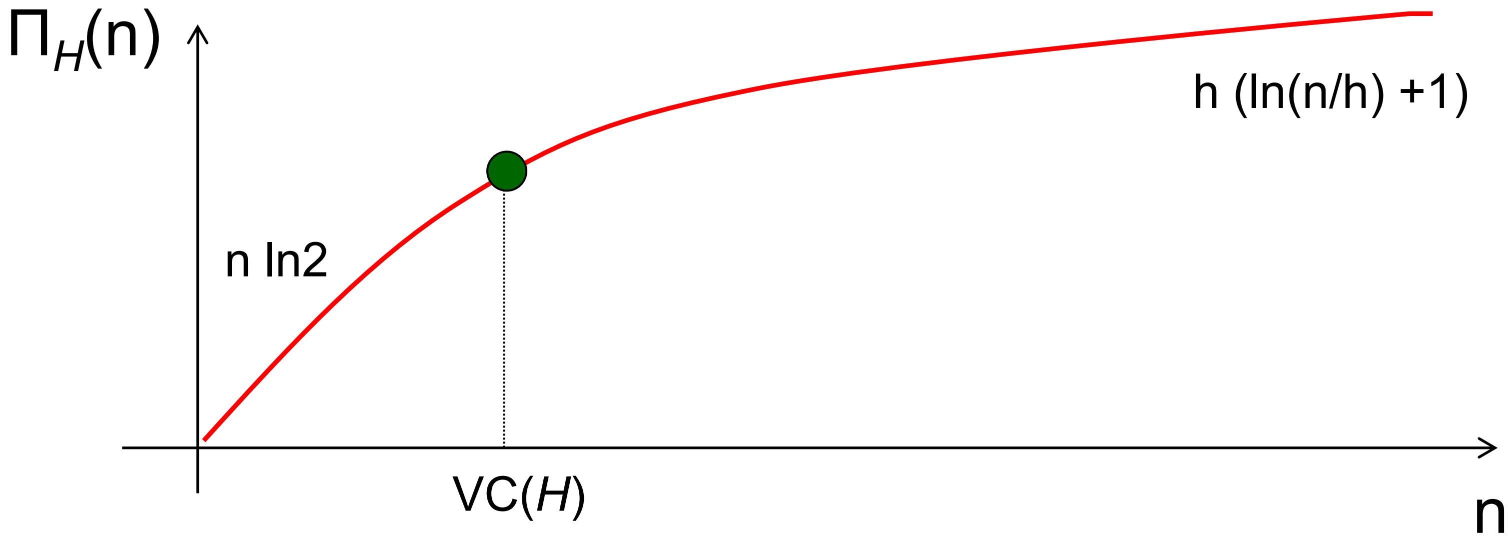
Vapnik Chervonenkis Dimension



Point n = VC where growth starts to slow down is called ***VC dimension***

- The VC dimension of *H* is the cardinality of the largest set S that can be fully represented by *H* (i.e. learned)
- VC is typically finite in good learners
- “saturating” growth function ensures Asymptotic Consistency of ERM

Vapnik Chervonenkis Dimension

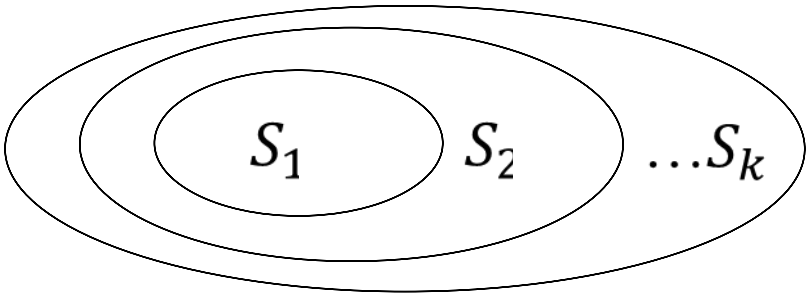


Point n = VC where growth starts to slow down is called ***VC dimension***

- **ERM** applicable for large n (n/VC > 20)
- Problem for small n (n/VC < 20) $\rightarrow$ need to constrain the structure of the learner $\rightarrow$ **SRM**

Structural Risk Minimization (SRM) (1)

- SRM requires a priori specification of a structure for sets of approximating functions.



Structure on a set of approximating Functions S_1, S_2,…S_k

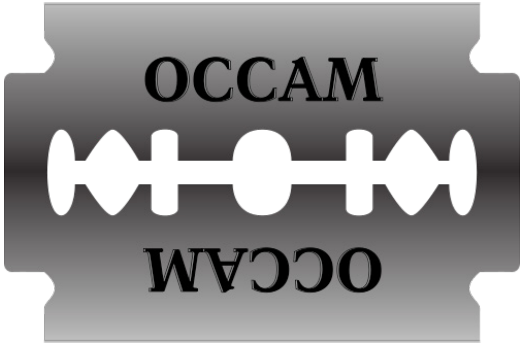
- ***SRM approach:***
  - Calculate or estimate VC-dimension for any element S_k of the structure
  - Minimize empirical risk R(w) for each element of the structure

Structural Risk Minimization (SRM) (2)

- SRM – a trade off between ***complexity*** (of approximating functions) and ***quality*** (of results)

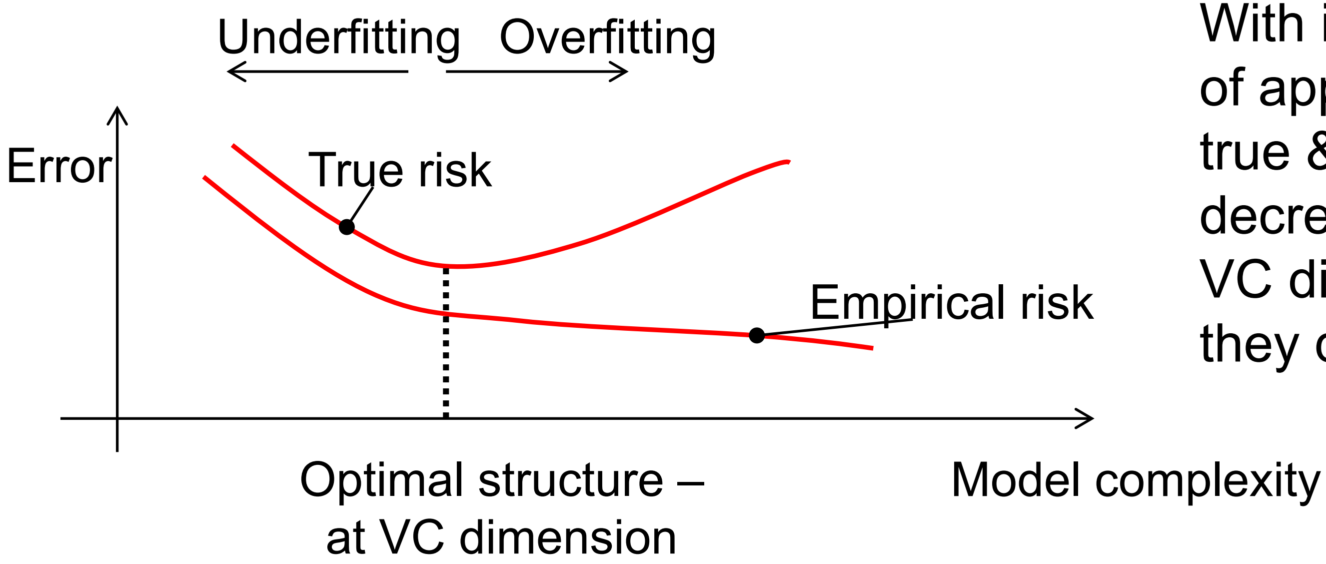
„As simple as possible,
but with enough quality.“

Occam‘s razor principle



- SRM – optimal model estimation:
  - Select an element of a structure with optimal complexity
  - Define the model based on selected approximating functions

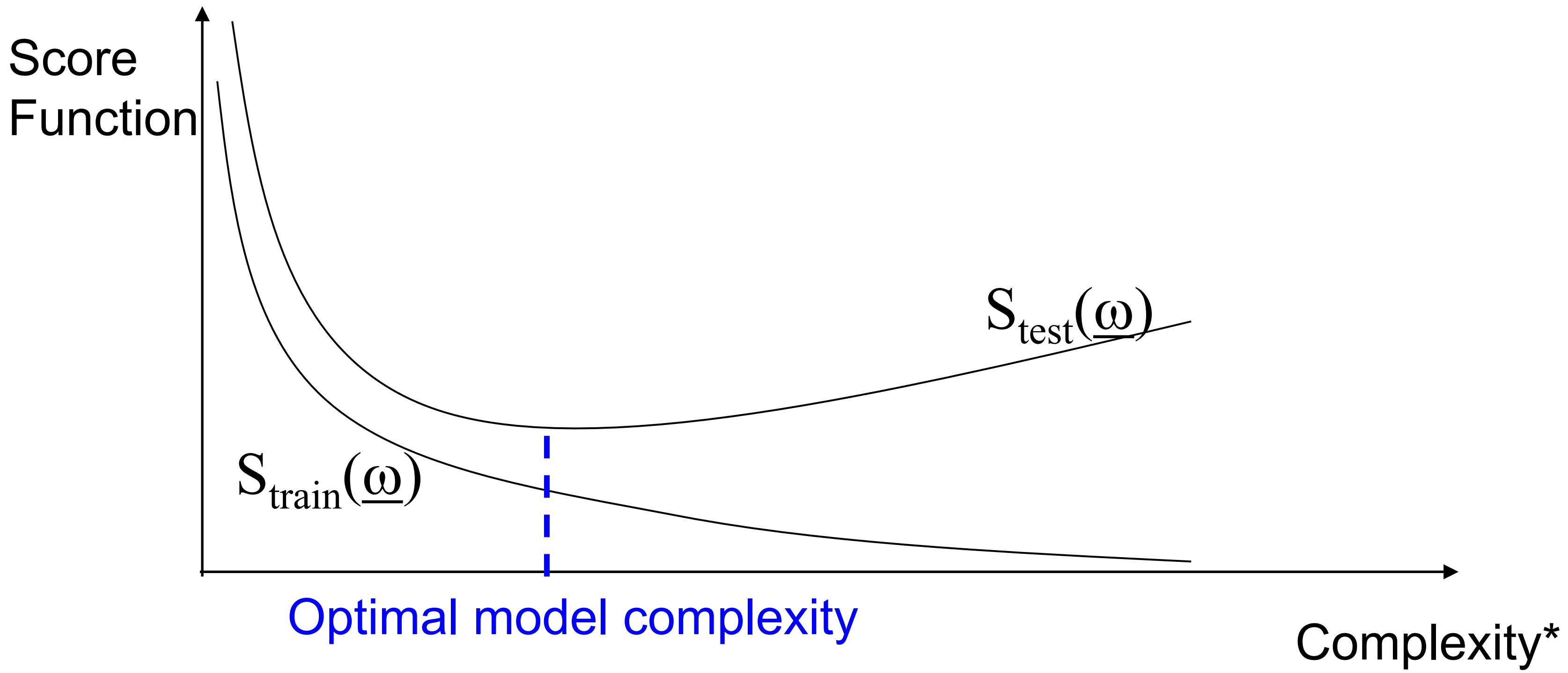
SRM Optimization Strategy



With increasing complexity of approximating functions true & empirical risk R(w) decrease until the value – VC dimension; thereafter they diverge.

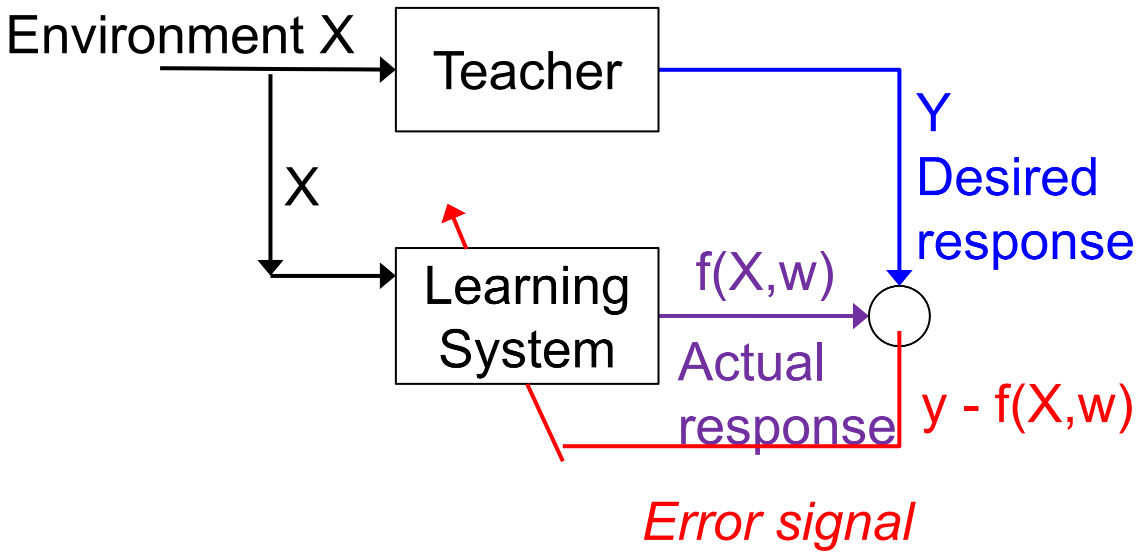
- ***Optimization:***
  - Stochastic approximation (or gradient descent)
  - Iterative methods
  - Greedy optimization

Complexity and Generalization



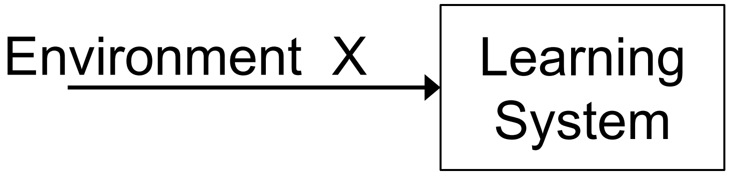
- *Complexity = degrees of freedom in the model,
  E.g.: number of variables.

Main Types of Inductive Learning



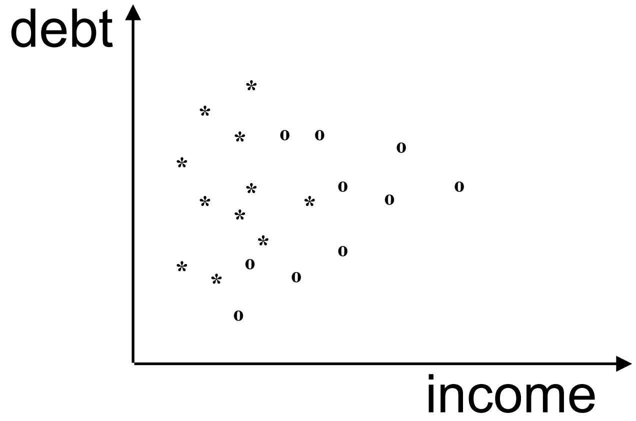
Supervised learning (with a teacher)

Unsupervised learning:
(without teacher)



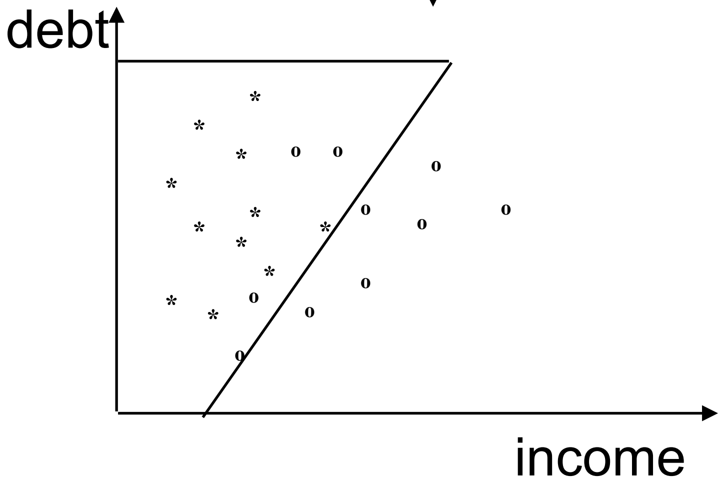
- goal is to discover "natural" structure in the data,
- requires task-independent measure of quality of representation

Supervised vs. Unsupervised Learning: Supervised



Problem: bank approval of credit (1)

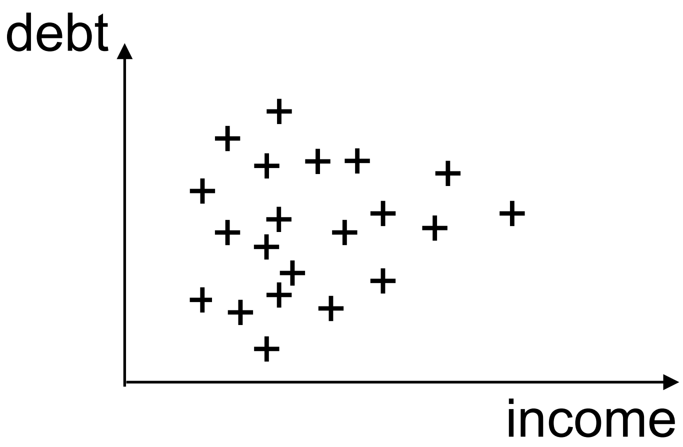
Supervised
Learning



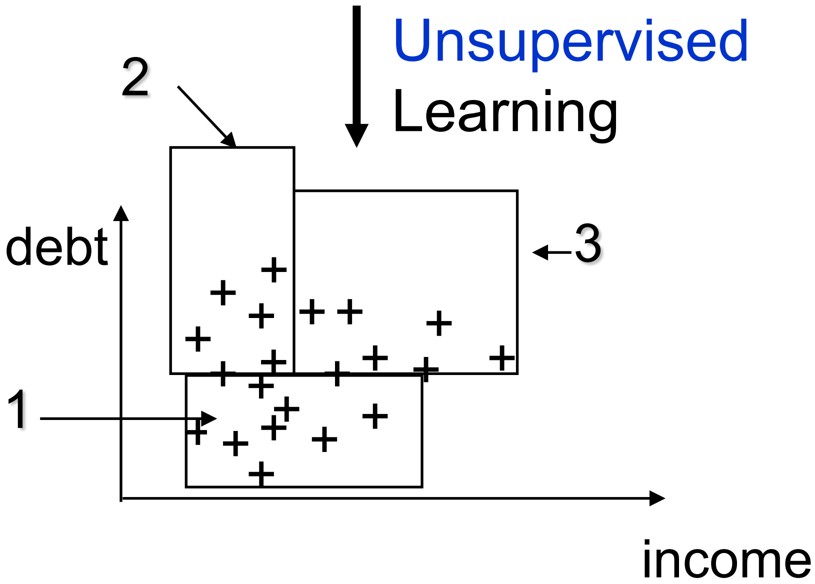
Previous customers with or without approval.
Learning:
Linear classification function:

1. above – reject
2. below - accept

Supervised vs. Unsupervised Learning: Unsupervised



Problem: bank approval of credit (2)



Approval unknown for previous customers.

Three classes of customers:

1. Low debt - approved
2. High debt +Low income - reject
3. High debt +High income - additional analysis

Using Data

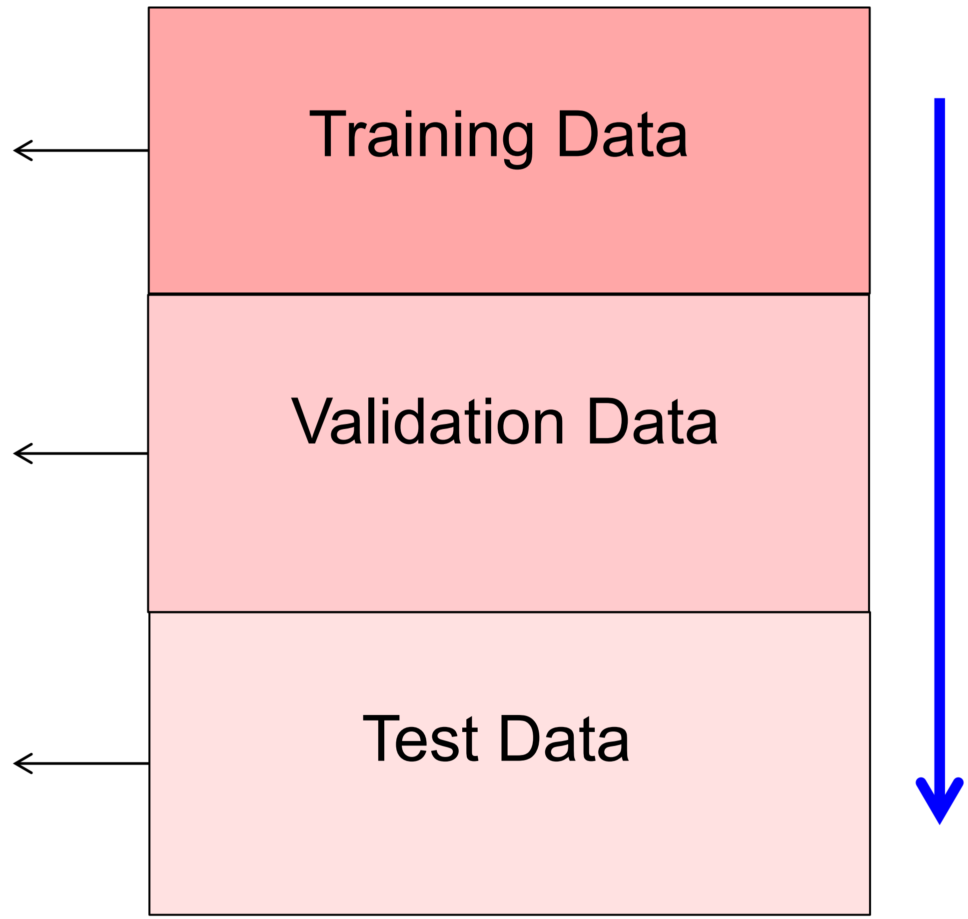
- Use this data to find the best $\underline{\omega}$ for each model

  $f_k(\underline{x}\ ;\ \underline{\omega})$.

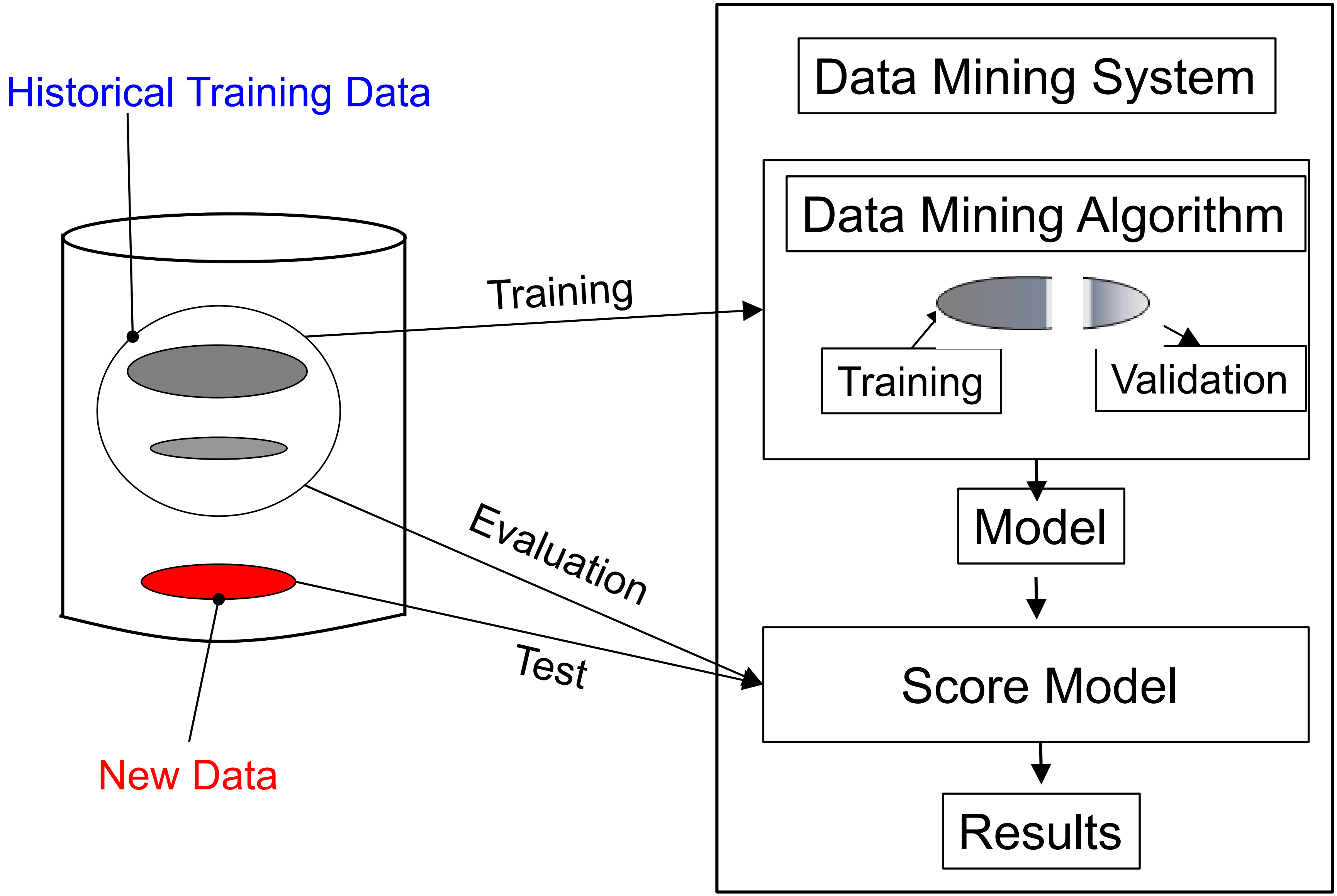
- Use this data to calculate an estimate of $S_k(\underline{\omega})$ for each $f_k(\underline{x}\ ;\ \underline{\omega})$ and select

  $k^* = \arg\min_k S_k(\underline{\omega})$

- Use this data to calculate an unbiased estimate of $S_{k^*}(\underline{\omega})$ for the selected model



The Data Mining Process



Model Estimation

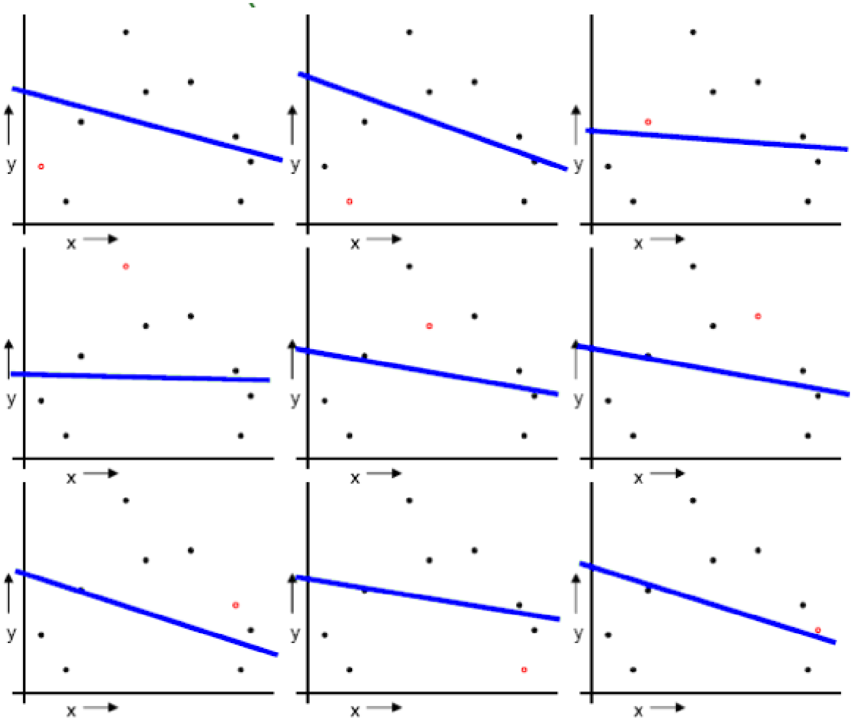
- Model ***Verification*** – model gives good representation of data-generating process.
  - Building the model <u>right</u>, i.e. it corresponds to the system.
- Model ***Validation*** – model behaves with satisfactory accuracy.
  - Building the <u>right</u> model, i.e. it corresponds to the data.
- V & V are performed through training (learning) and testing data sets.
- For extremely large amount of samples any V & V method is applicable.

Model & Parameters Estimation

- ***Resubstitution*** method
  - naïve strategy, training data = testing data; optimistically biased; not for small *n*.
- ***Holdout*** method
  - *x*% of data for training , (1-*x*)% for testing.
- ***Leave-one-out*** method (LOO)
  - *n*-1 training samples, one testing samples; repeat *n* times.
- ***Rotation*** method (*n*-fold cross validation)
  - total of *P* data segments, *P*-1 for training, one for testing; repeat *P* times.
- ***Bootstrap*** method
  - resample with replacement randomly to generate “fake” data sets of the same size for training and testing.

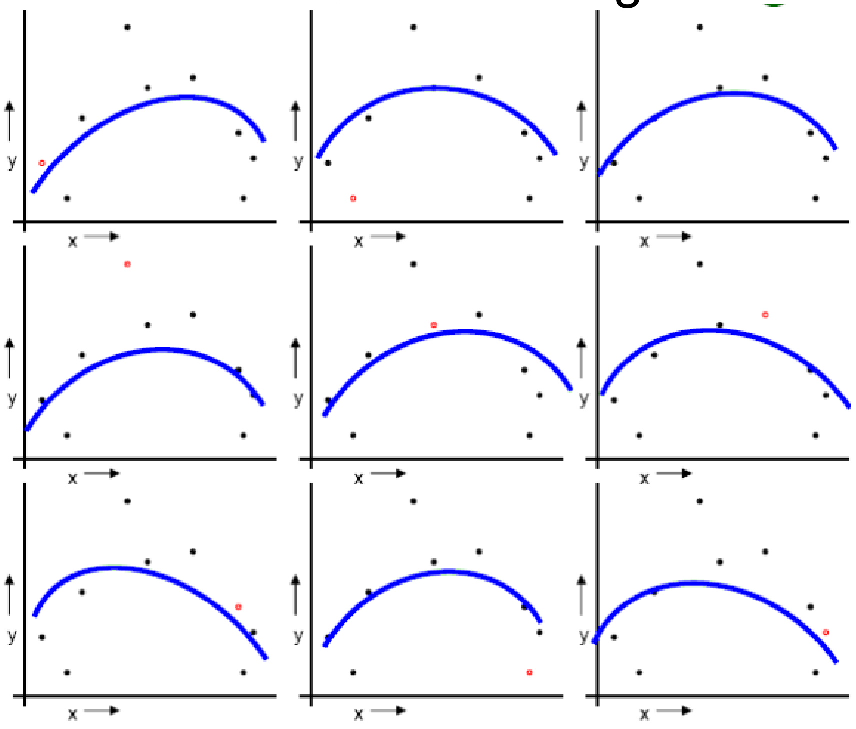
Examples: Leave One Out Cross Validation

LOOCV



$$\text{MSE}_{\text{LOOCV}} = 2.12$$

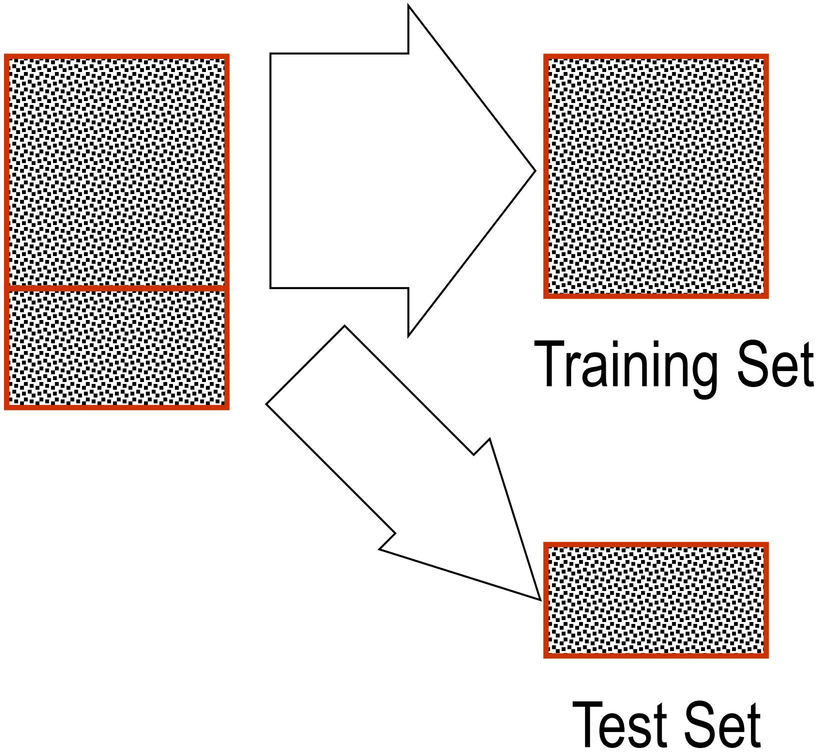
LOOCV for Quadratic Regression



$$\text{MSE}_{\text{LOOCV}} = 0.962$$

Hold-out Set

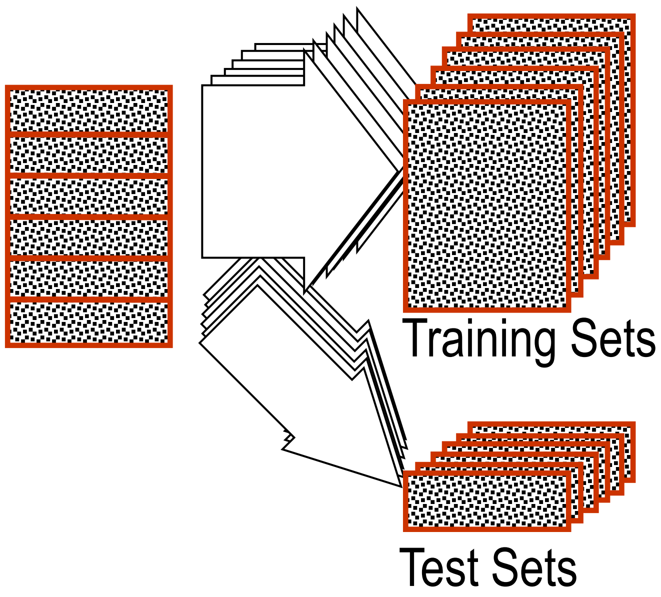
- ***Hold-out set***: Partition data into training and test sets

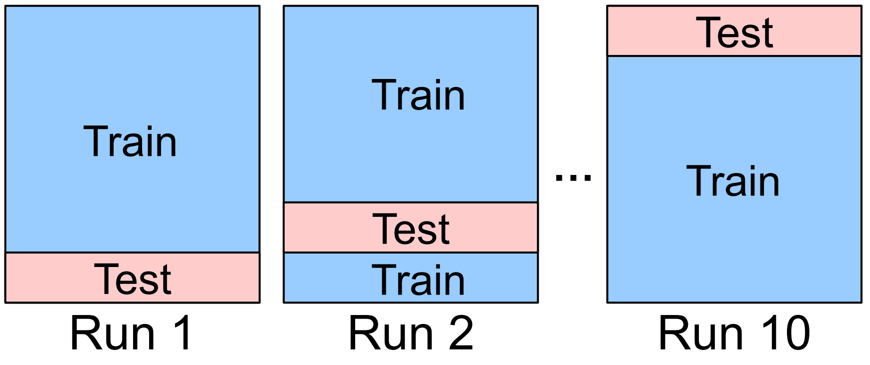


N-fold Cross Validation

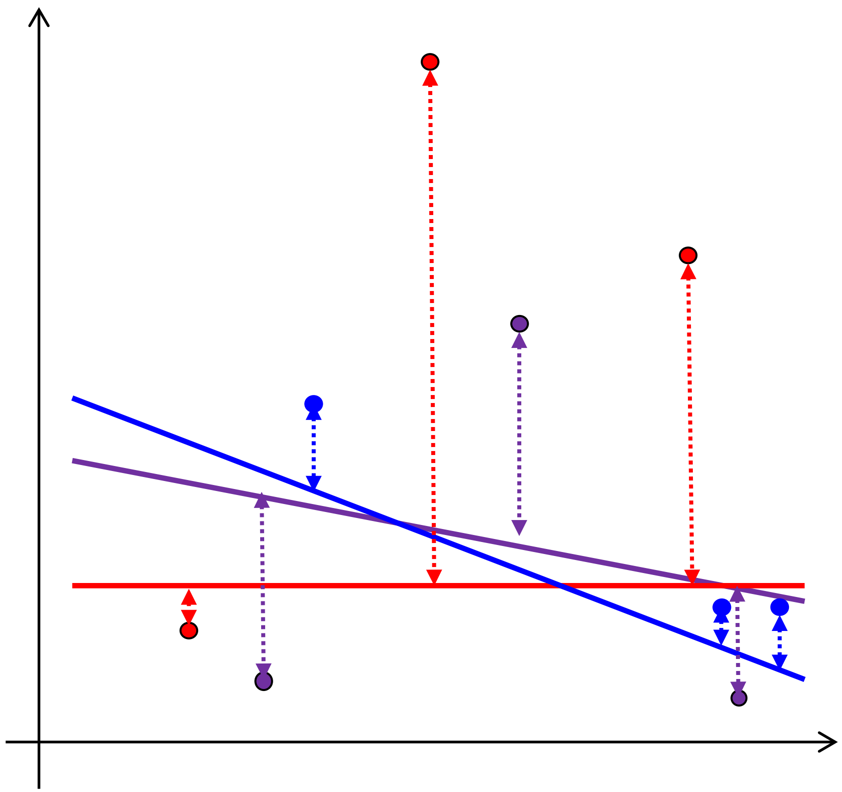
- ***N-fold Cross Validation:***
  create **N** equal partitions

- **Example:**
  10-fold cross validation:
  - Use the first 90% of the data set for training and then test on the final 10%
  - Then use the next 10% for testing etc.





K-fold Cross Validation



Linear Regression:
$MSE_{3FOLD} = 2.05$

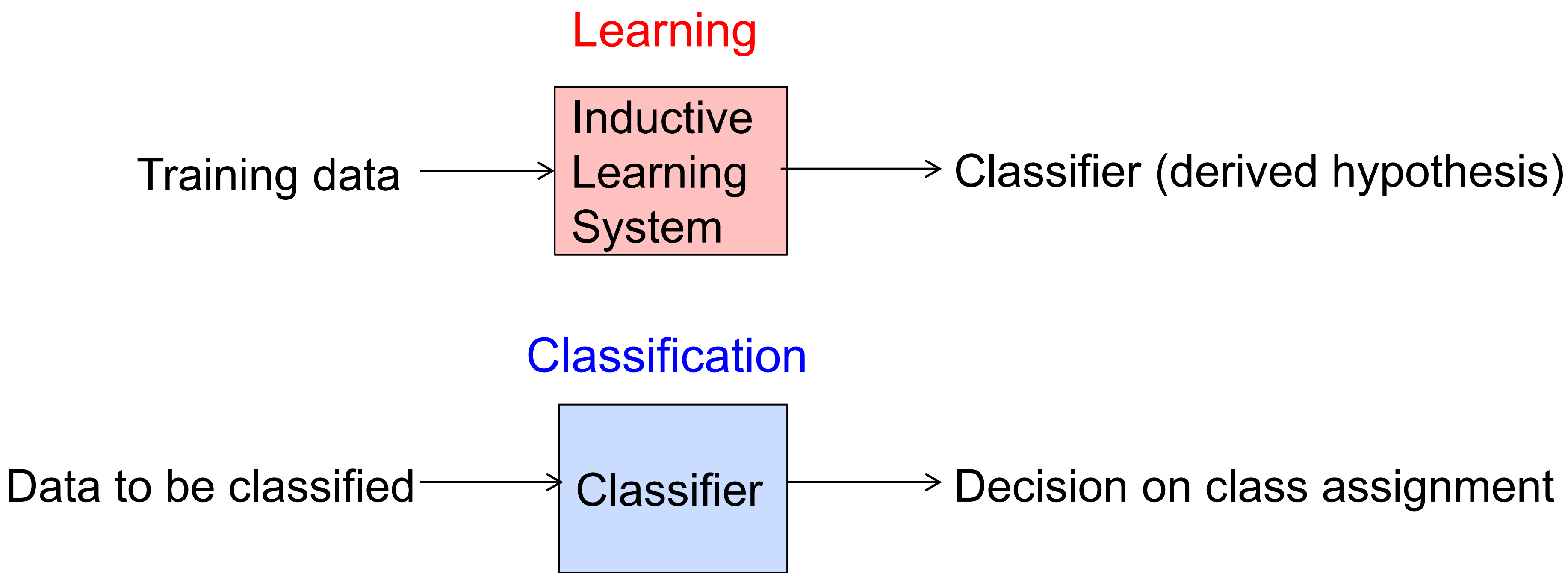
Randomly break the dataset into k partitions (here: k=3 – red, blue, purple).

- For the red partition: Train on the points not in the red partition. Find the test-sum of errors on the red points.
- For the blue partition: Train on the points not in the blue partition. Find the test-sum of errors on the blue points.
- For the purple partition: Train on the points not in the purple partition. Find the test-sum of errors on the purple points.

Then report the mean error!

Evaluation of Classification Systems (1)

- Task: Determine which of a fixed set of classes an example belongs to.
- Input: Training set of examples annotated with class values.
- Output: Induced hypotheses (model/concept description/classifiers).



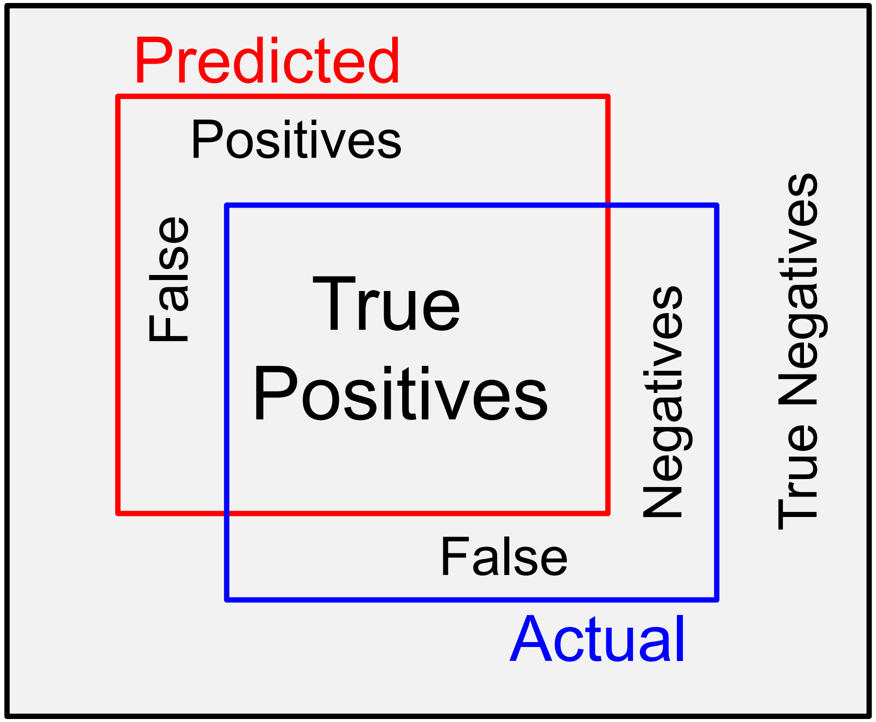
Evaluation of Classification Systems (2)

Evaluation criteria:

- ***Accuracy*** of the classification
- ***Interpretability***
  - E.g. size of a decision tree; insight gained by the user
- ***Efficiency***
  - … of model construction
  - … of model application
- ***Scalability*** for large datasets
  - for secondary storage data
- ***Robustness***
  - w.r.t. noise and unknown attribute values

Evaluation of Classification Systems (3)

- Training Set: examples with class values for learning.
- Test Set: examples with class values for evaluating.
- ***Evaluation***: Hypotheses are used to infer classification of examples in the test set; inferred classification is compared to known classification.
- ***Accuracy***: percentage of examples in the test set that is classified correctly.



The Confusion Matrix

Actual / **Predicted**	Class 1	Class 2
Class 1	A: True Positive	B: False Positive
Class 2	C: False Negative	D: True Negative

Evaluation metrics:

Accuracy	A = (A+D)/(A+B+C+D)	
True positive rate	TPr = A/(A+C) = 1- false negative rate	
False positive rate	FPr = B/(B+D) = 1- true negative rate	
Sensitivity	SE = TPr	
Specificity	SP = 1 - FPr	
Recall	R = A/(A+C)	*different in the*
Precision	P = A/(A+B)	*Kantardzic book!*
F-score	F = 2 P R / (P+R)	

Confusion Matrix for Three Classes

	True Class			
Classification Model	0	1	2	Total
0	28	**1**	**4**	33
1	**2**	28	**2**	32
2	**0**	**1**	24	25
Total	30	30	30	90

$$\textit{Error} = \frac{\textit{Sum of non diagonal}}{\textit{Total}} = 10 / 90 = 0.11 \ (11\%)$$

Accuracy = 1 – Error = 1 - 0.11 = 0.89 (89%)

Accuracy Unsuitable for Skewed Distributions

P\A	C1	C2
C1	0	0
C2	10	90

Accuracy	90/100
Recall	0/10
Precision	0/0
F-Score	0/0

P\A	C1	C2
C1	3	10
C2	7	80

Accuracy	83/100
Recall	3/10
Precision	3/13
F-Score	6/23

P\A	C1	C2
C1	8	42
C2	2	48

Accuracy	56/100
Recall	8/10
Precision	8/50
F-Score	4/15

Cost Matrix

	PREDICTED CLASS	
ACTUAL CLASS		
C(i|j)*	Class=Yes	Class=No
Class=Yes	C(Yes|Yes)	C(No|Yes)
Class=No	C(Yes|No)	C(No|No)

*C(i|j): Cost of misclassifying class j example as class i

Computing Cost of Classification

Cost Matrix	Predicted Class		
Actual Class	C(i|j)	+	-
	+	1	100
	-	1	0

Model M_1	Predicted Class		
Actual Class		+	-
	+	150	40
	-	60	250

Model M_2	Predicted Class		
Actual Class		+	-
	+	250	45
	-	5	200

Accuracy = 80%
Cost = 3910

Accuracy = 90%
Cost = 4255

Cost vs. Accuracy

Count		Predicted Class	
		Class=Yes	Class=No
Actual Class	Class=Yes	a	b
	Class=No	c	d

$$\textbf{\textit{Accuracy}} = \frac{a+d}{N}$$

With N = a + b + c + d

Cost		Predicted Class	
		Class=Yes	Class=No
Actual Class	Class=Yes	p	q
	Class=No	q	p

Cost = p (a+d) + q (b+c)

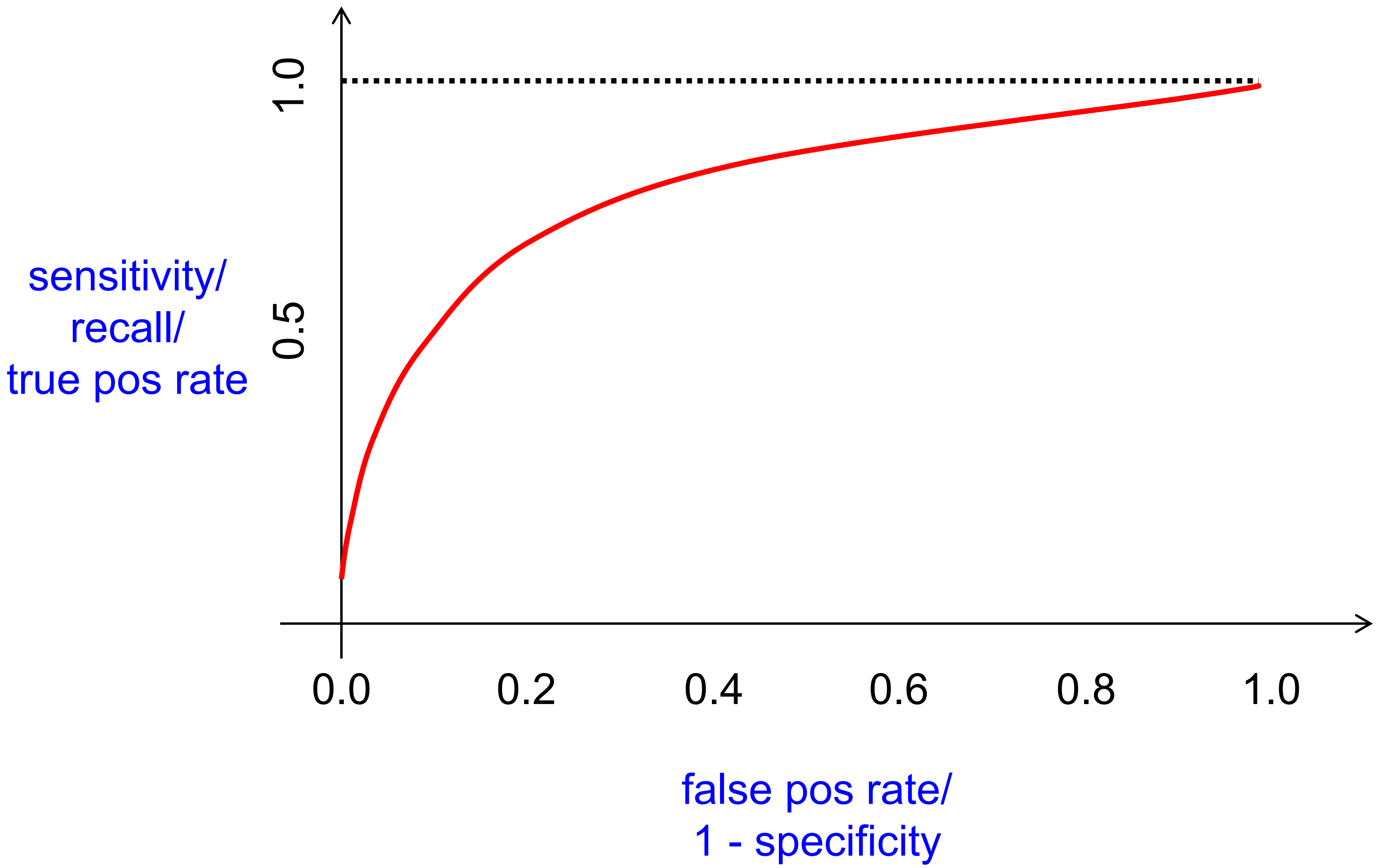
= p (a+d)+q (N–a –d)

= q N–(q –p)(a+d)

= N[q–(q-p)×Accuracy]

- Accuracy is proportional to cost if:
  - C(Yes|No) = C(No|Yes) = q
  - C(Yes|Yes) = C(No|No) = p

Receiver Operating Characteristic (ROC)



- ROC is a good measure of overall model performance

How to Construct ROC Curve? (1)

- A model is often tunable to different thresholds
- ROC computes sensitivity and specificity for all possible thresholds and plots them

- If threshold = minimum (=0)
  - $c = d = 0$ so sens = 1; spec = 0
- If threshold = maximum (=1)
  - $a = b = 0$ so sens = 0; spec = 1

	Actual outcome	
Predicted outcome	1	0
1	a	b
0	c	d

How to Construct ROC Curve? (2)

- Suppose we use a cutoff of ***0.5*** for our classifier…

	Actual Outcome	
Predicted Outcome	1	0
1	8	3
0	0	9

Sensitivity: 8/(8+0)

Specificity: 9/(9+3)

How to Construct ROC Curve? (3)

- Suppose we use a cutoff of ***0.8*** for our classifier…

	Actual Outcome	
Predicted Outcome	1	0
1	6	2
0	2	10

Sensitivity: 6/(6+2)

Specificity: 10/(10+2)

How to Construct ROC Curve – Automatization

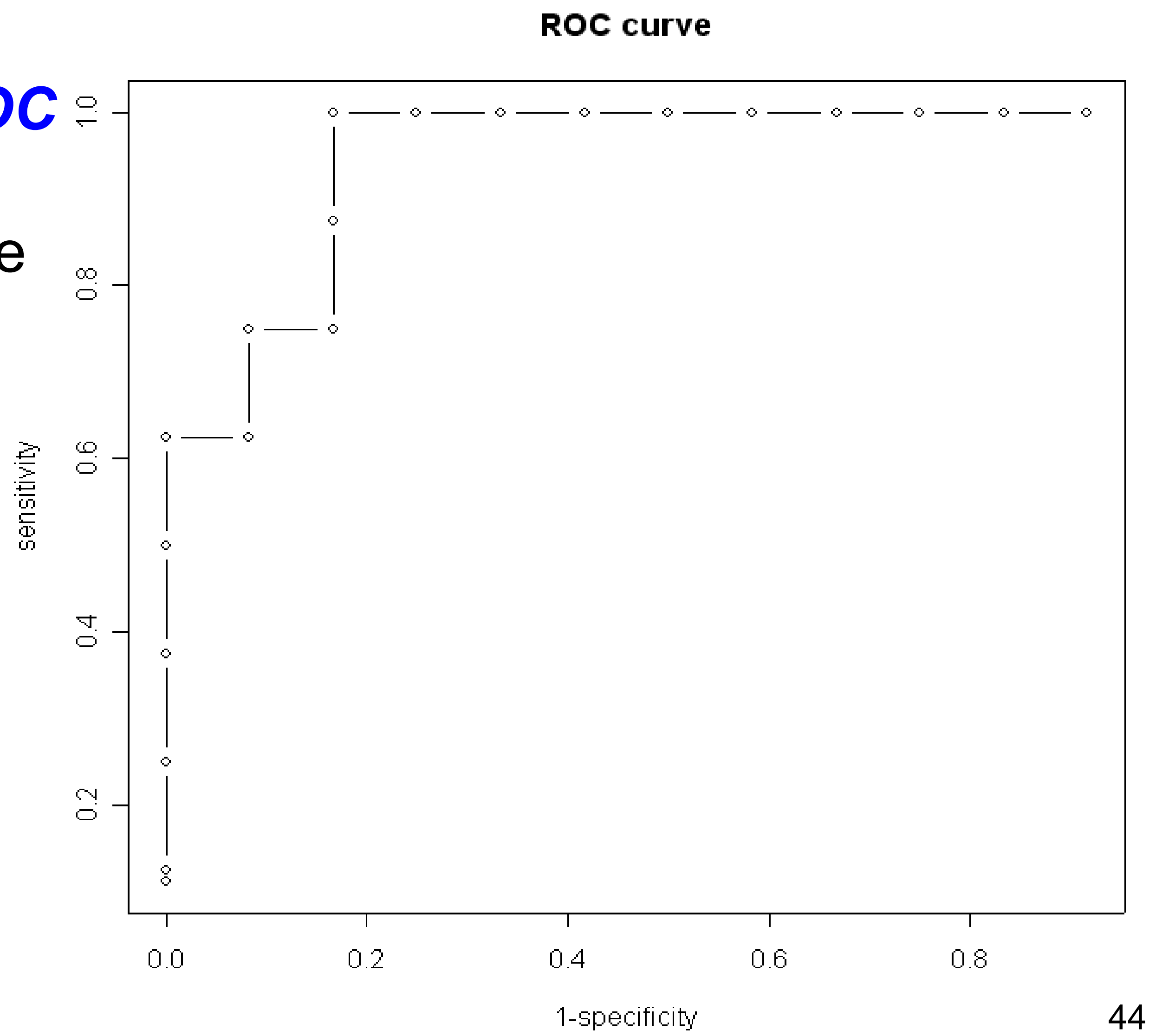
A1 *fx*

	A	C	D	E	F	G	H	I
1			a	b	c	d	sensitivity	specificity
2	0	0.005694	8	11	0	1	1	0.083333
3	0	0.009911	8	10	0	2	1	0.166667
4	0	0.025475	8	9	0	3	1	0.25
5	0	0.039375	8	8	0	4	1	0.333333
6	0	0.070495	8	7	0	5	1	0.416667
7	0	0.080184	8	6	0	6	1	0.5
8	0	0.099051	8	5	0	7	1	0.583333
9	0	0.346722	8	4	0	8	1	0.666667
10	0	0.493576	8	3	0	9	1	0.75
11	0	0.635592	8	2	0	10	1	0.833333
12	1	0.705922	7	2	1	10	0.875	0.833333
13	1	0.753097	6	2	2	10	0.75	0.833333
14	0	0.88035	6	1	2	11	0.75	0.916667
15	1	0.92832	5	1	3	11	0.625	0.916667
16	0	0.970674	5	0	3	12	0.625	1
17	1	0.97985	4	0	4	12	0.5	1
18	1	0.983794	3	0	5	12	0.375	1
19	1	0.984132	2	0	6	12	0.25	1
20	1	0.99631	1	0	7	12	0.125	1
21	1	0.999876	1	0	8	12	0.111111	1

```
sens<-c(1,1,1,1,1,1,1,1,1,1,0.875,0.75,0.75,0.625,0.625,0.5,0.375,0.25,0.125,0.11111
spec<-c(0.083333333,0.166666667,0.25,0.333333333,0.416666667,0.5,0.583333333,0.66666
33333,0.916666667,0.916666667,1,1,1,1,1,1)
plot(1-spec,sens,type="b",xlab="1-specificity",ylab="sensitivity",main="ROC curve")
```

Predictive Performance Measure

"Area under the ROC curve" is a common measure of predictive performance.



ROC (Receiver Operating Characteristic)

- ***ROC Space:*** Each classifier is represented by plotting its (FP, TP) pair

- ***Good model:***
  maximizing AUC
  (Area Under Curve)

- ***Interpolation:***
  a good model extends
  the ROC Convex Hull.

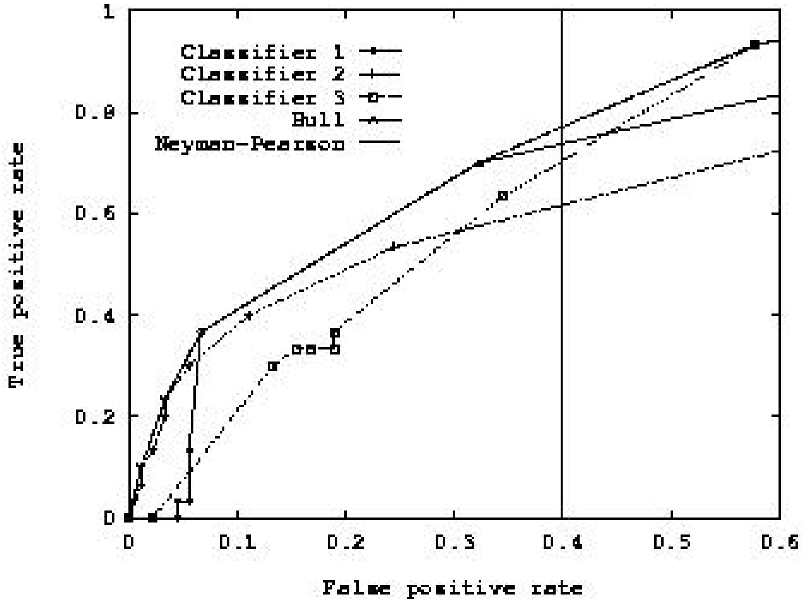
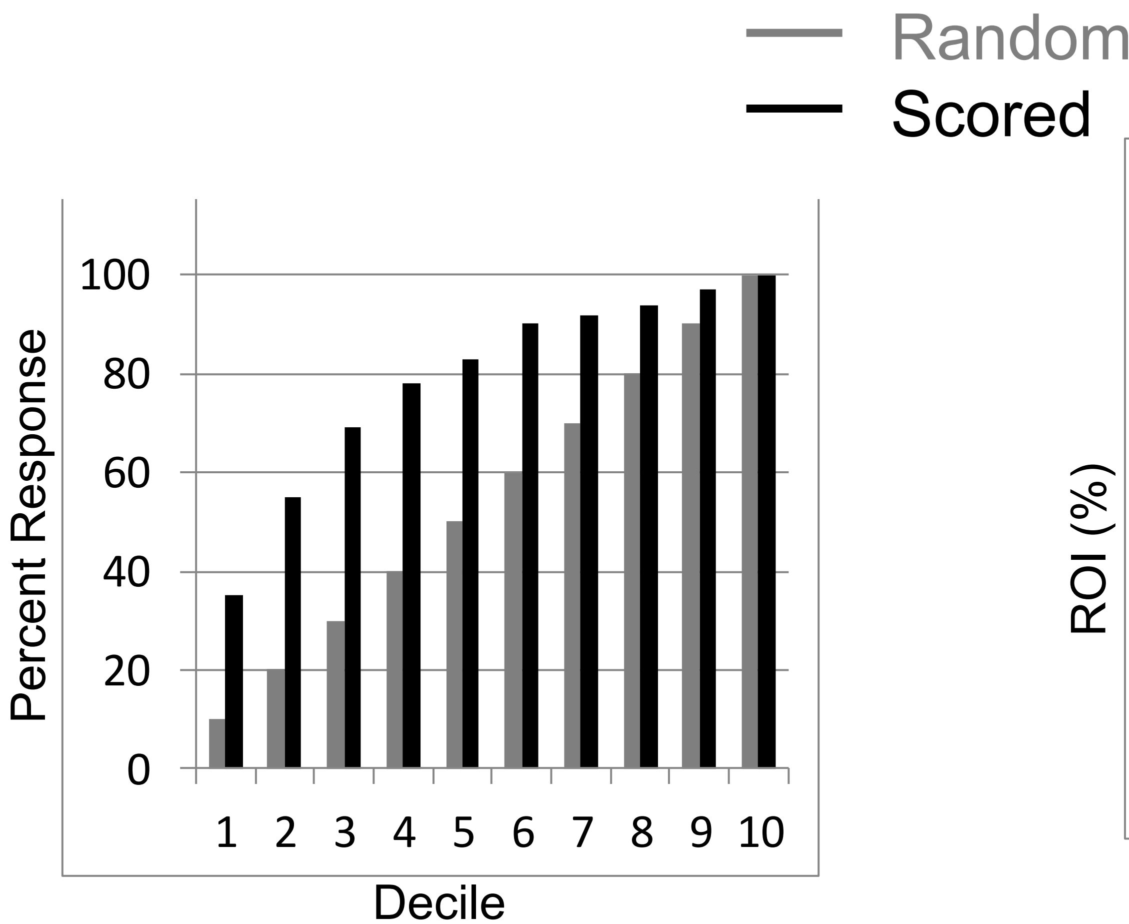
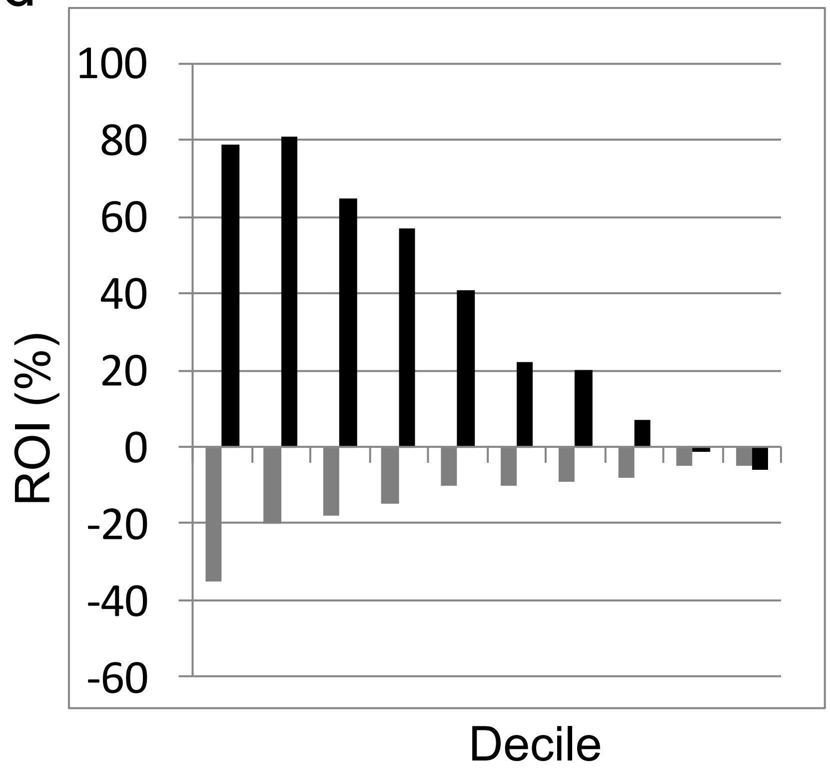


Figure 4: The ROC Convex Hull used to select a classifier under the Neyman-Pearson criterion

Assessing a Data Mining Model



Lift Chart



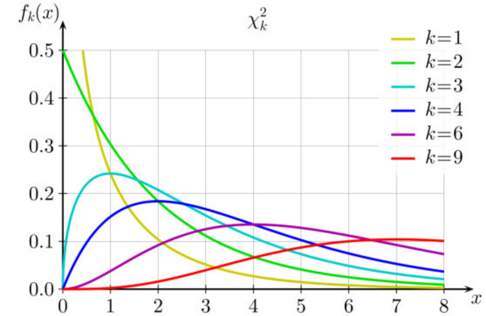
ROI Chart
(Return On Investment)

McNemar's test for comparison of two classifiers

e_{00}: Number of samples misclassified by both classifiers.	e_{01}: Number of samples misclassified by classifier 1, but not classifier 2
e_{10}: Number of samples misclassified by classifier 2, but not classifier 1	e_{11}: Number of samples correctly classified by both classifiers.

Apply the $\chi 2$ statistic with one degree of freedom for:

$$\frac{[\,(\,|e_{01} - e_{10}| - 1)^{2}\,]}{(e_{01} + e_{10})} \sim \chi 2$$



McNemar's test rejects the hypothesis that the two algorithms have the same error at the significance level α, if previous value is greater than $\chi 2_{\alpha,\,1}$

For example, for $\alpha = 0.05$, $\chi 2_{0.05,\,1} = 3.84$.

Test with K-fold Cross Validation (1)

- We compare the error percentages in two classification algorithms based on errors in K validation sets which are recorded for two models as:

  p_{1i} and p_{2i} , i = 1, … , k.

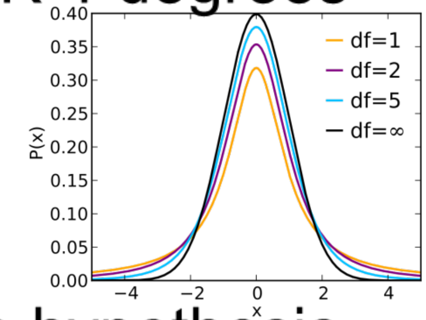
- The difference in error rates on fold *i* is $P_i = p_{1i} - p_{2i}$

- Compute:

$$m = \frac{\left[\sum_{i=1}^{k} P_i\right]}{K} \quad \text{and} \quad S^2 = -\frac{\left[\sum_{i=1}^{k} (P_i - m)^2\right]}{K-1}$$

Test with K-fold Cross Validation (2)

- We have a statistic which is t distributed with K-1 degrees of freedom, and the following test:

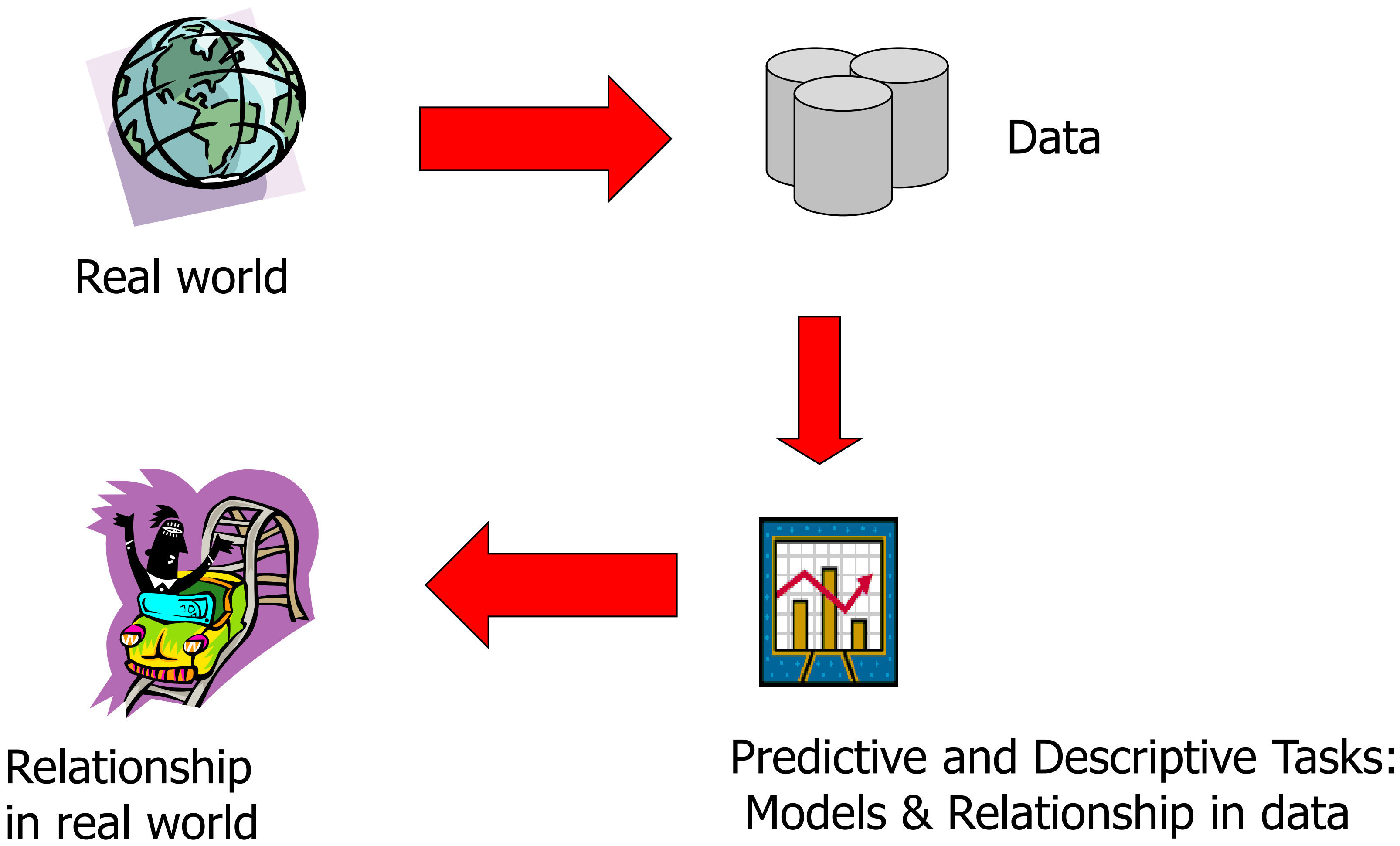


$$\frac{\sqrt{k} * m}{s} \sim t_{k-1}$$

- K-fold cross validation paired t-test rejects the hypothesis that two algorithms have the same error rate at significance level α, if previous value is outside interval: ($-t_{\alpha/2,K-1}$, $t_{\alpha/2,K-1}$)

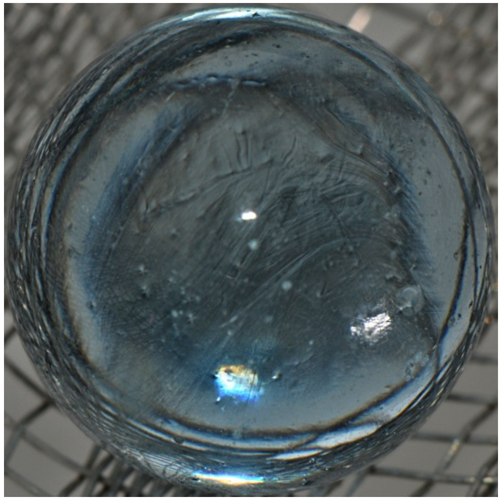
- For example, for $\alpha = 0.05$ and K = 10 or 30: $t_{0.025,\ 9} = 2.26$, and $t_{0.025,\ 29} = 2.05$.

Turning Data into Knowledge



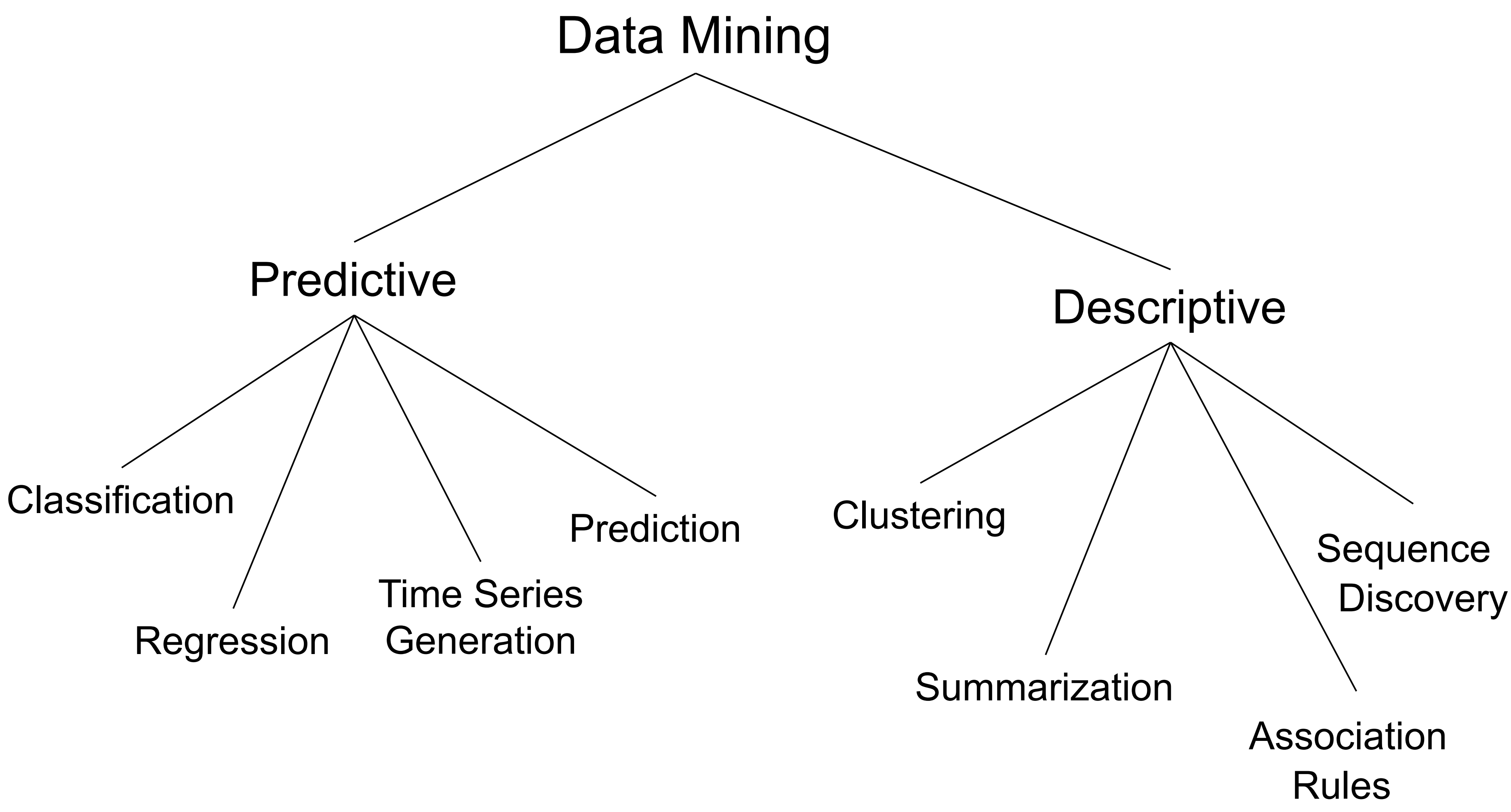
Data Mining Tasks Overview (1)

- ***Prediction*** tasks
  - Use some variables to predict unknown or future values of other variables.
    - Produce as a result the ***model***.
    - **Examples**:
      decision tree, artificial neural network, …



- ***Description*** tasks
  - Find human-interpretable patterns that describe the data.
    - produce as a result ***information***.
    - **Examples**: rule, graph, summary, …

Data Mining Tasks Overview (2)



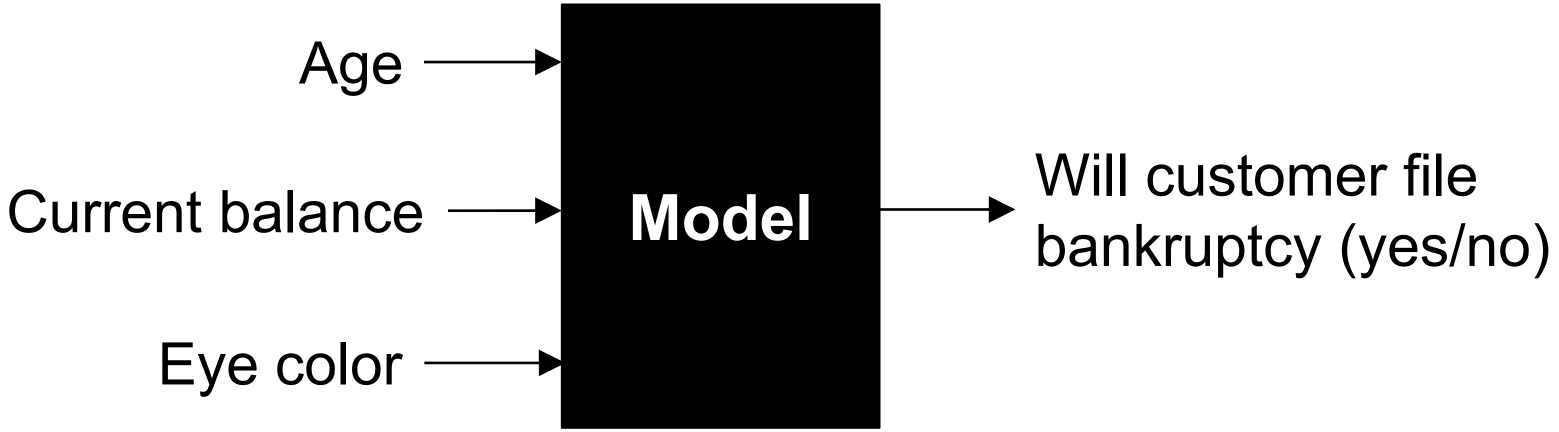
Data Mining Algorithms

A data mining algorithm is a well-defined procedure that takes data as input and produces output in the form of models or patterns.

- ***Well-defined:*** can be encoded in software
- ***Algorithm***: must terminate after some finite number of steps

Predicitve Modelling (1)

- A ***black box*** that makes predictions about the future based on information from the past and present.



- Large number of inputs is usually available to build the model.

Predicitve Modelling (2)

- Predict one variable Y given a set of other variables X
  - Here X could be a n-dimensional vector
- ***Classification***: Y is categorical
- ***Regression***: Y is real-valued

$$\boxed{y = f(x)}$$

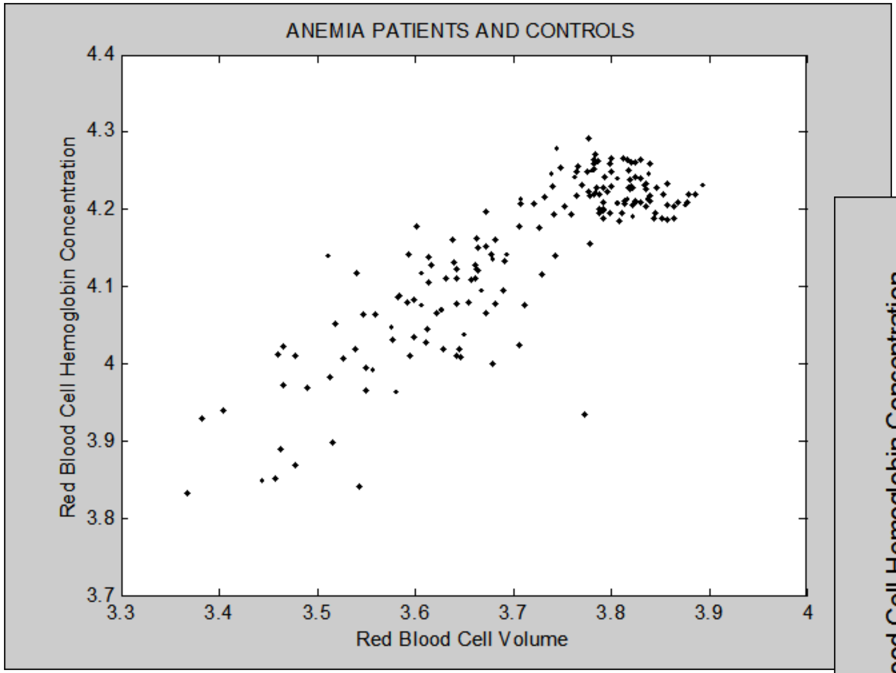
- In effect this is function approximation (F), learning the relationship between Y and X
- Many algorithms for predictive modeling in statistics and machine learning
- Often the emphasis is on predictive accuracy (~ERM), less emphasis on understanding the model (~SRM)

Descriptive Modelling

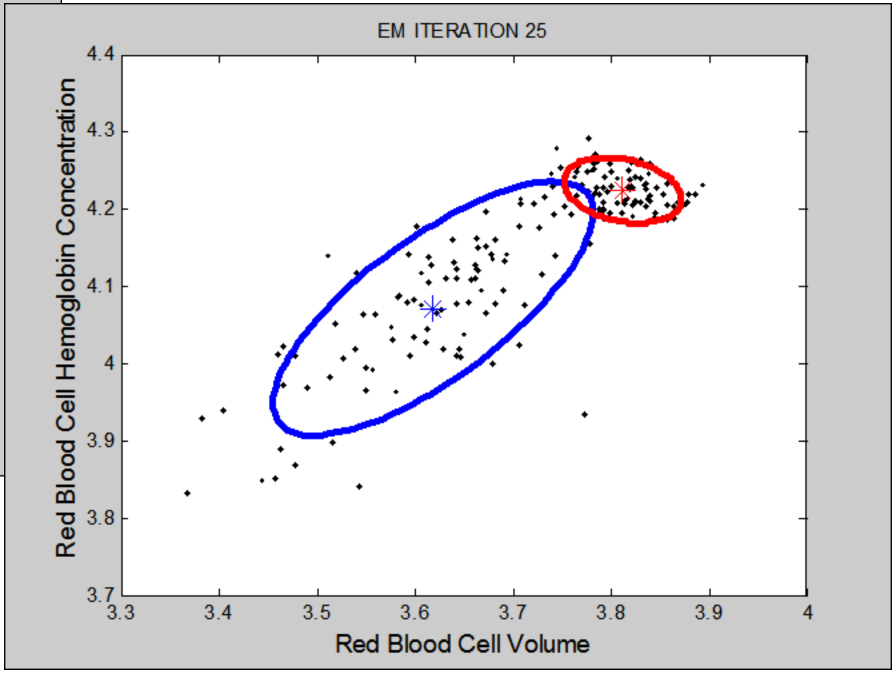
- Goal is to build a ***generative*** or ***descriptive*** model
  - E.g., a model that could simulate the data helping to understand basic characteristics of the process.

- **Examples:**
  - ***Density estimation***:
    - Estimate the joint distribution $P(x_1,\ldots\ldots x_p)$
  - ***Cluster analysis***:
    - Find natural groups in the data and describe them
  - ***Dependency models*** among variables
    - Learn a Bayesian network for the data

Example of Descriptive Modeling



Control Data



Main Groups

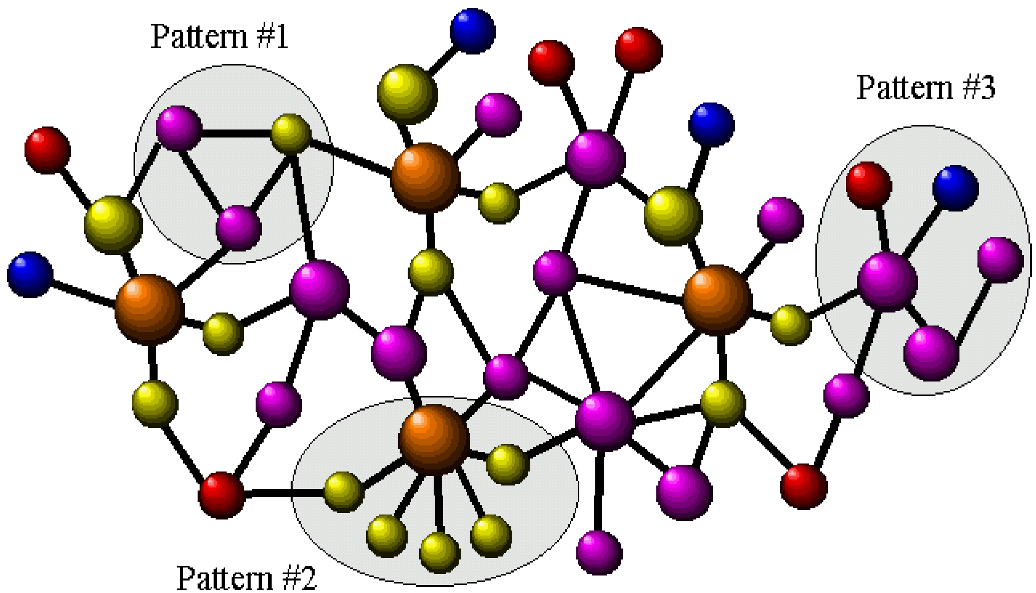
Pattern Discovery is a Descriptive Task

- Gene Analysis Example:

```
ADACABDABAABBDDBCADDDDBCDDBCCBBCCDADADAADABDBBDAB
ABBCDDDCDDABDCBBDBDBCBBABBBCBBABCBBACBBDBAACCADDA
DBDBBCBBCCBBBDCABDDBBADDBBBBCCACDABBABDDCDDBBABDB
DDBDDBCACDBBCCBBACDCADCBACCADCCCACCDDADCBCADADBAA
CCDDDCBDBDCCCCACACACCDABDDBCADADBCBDDADABCCABDAAC
ABCABACBDDDCBADCBDADDDDCDDCADCCBBADABBAAADAAABCCB
CABDBAADCBCDACBCABABCCBACBDABDDDADAABADCDCCDBBCDB
DADDCCBBCDBAADADBCAAAADBDCADBDBBBCDCCBCCCDCCADAAD
ACABDABAABBDDBCADDDDBCDDBCCBBCCDADADACCCDABAABBCB
DBDBADBBBBCDADABABBDACDCDDDBBCDBBCBBCCDABCADDADBA
CBBBCCDBAAADDDBDDCABACBCADCDCBAAADCADDADAABBACCBB
```

Another Example of Descriptive Modeling

- Learning Directed Graphical Models



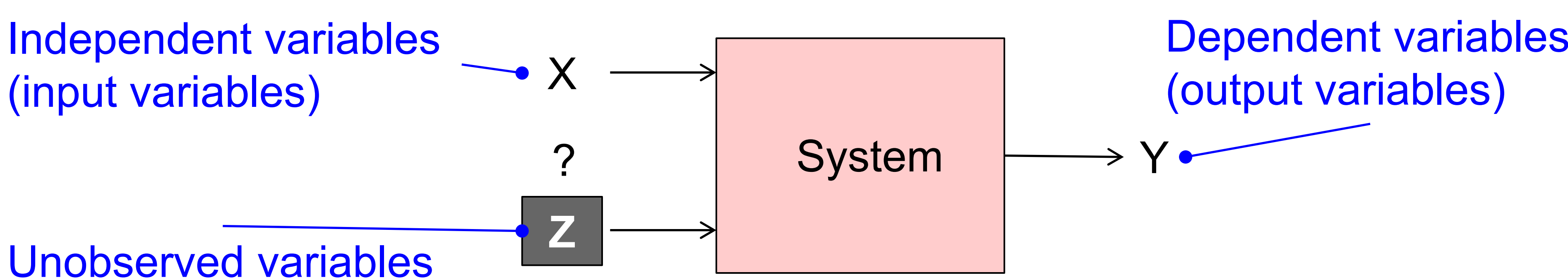
Tabular Representation of a Data Set

Features (=Variables, Attributes)

Samples

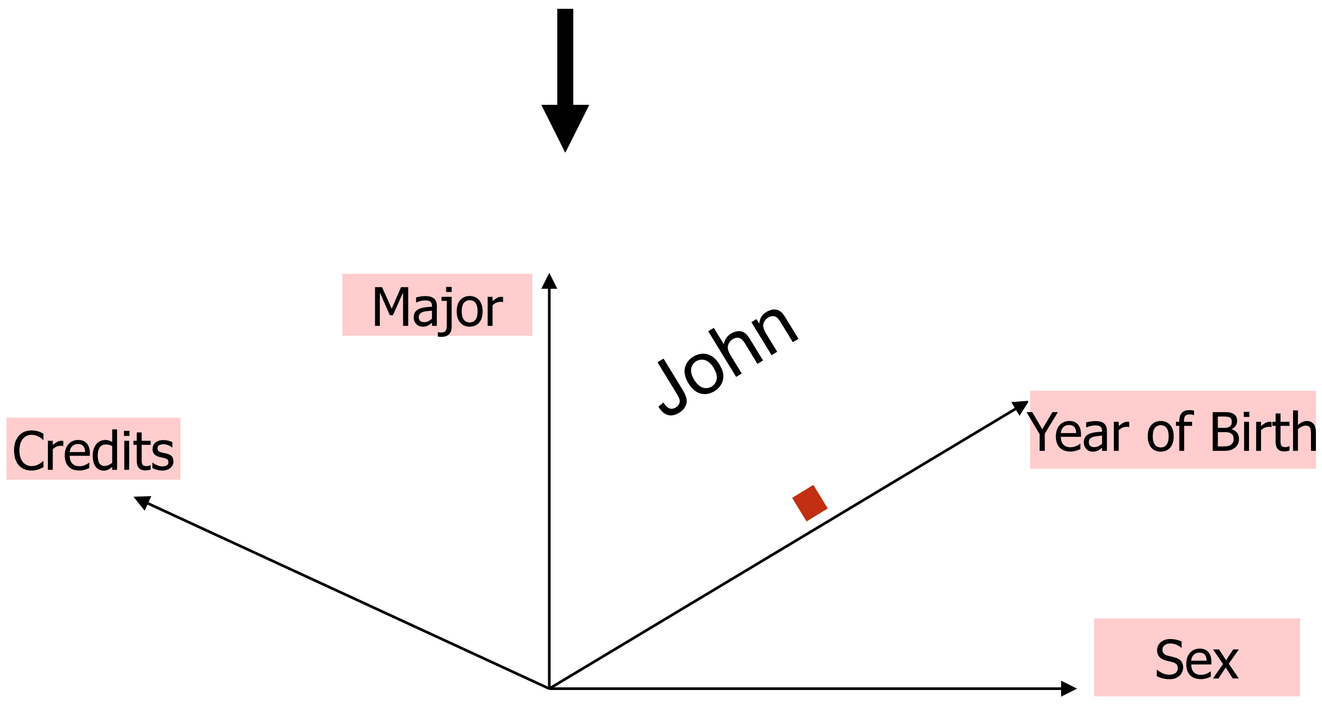
1				
2				
.				
.				
.	M	1983	CECS	132
n				

Feature's value for the given sample



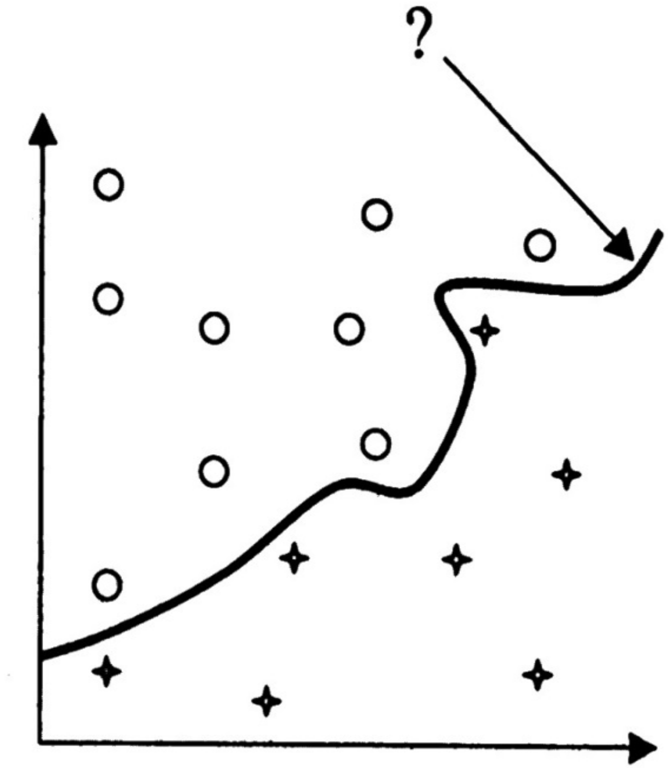
Points in n-Dimensional Space Representation of Data Samples

Name	*Sex*	*Year of Birth*	*Major*	*Credits*
John	M	1983	CECS	132



Data Mining Tasks: Classification

- Classification is a learning function that ***classifies*** a sample (nD) into one of several ***predefined*** classes.
  - Given a collection of samples – nD points (training set)
  - Find a ***model*** for class (***output***) attribute as a function of the other attributes
  - Goal: previously unseen samples should be assigned a class as accurately as possible (test set)

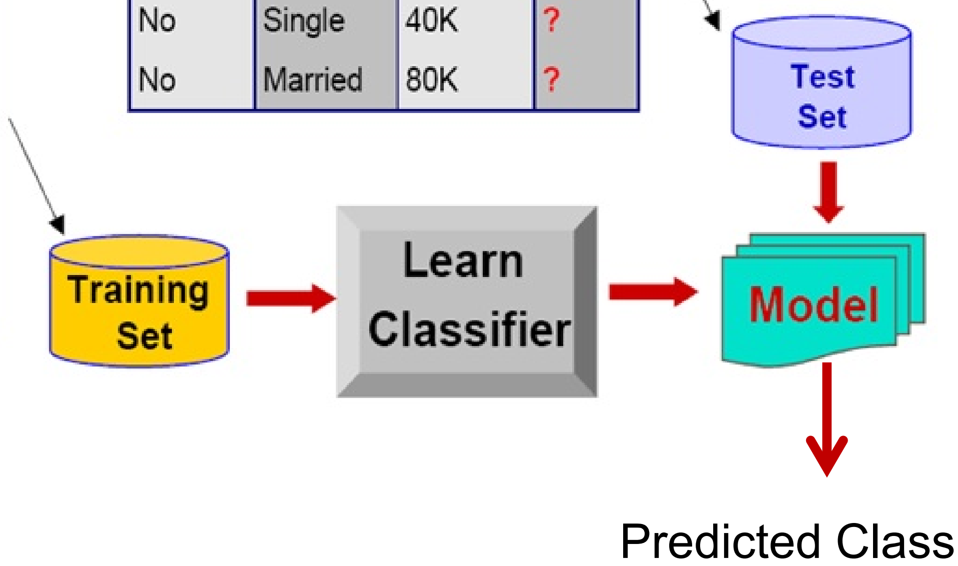


Classification Example

categorical categorical continuous class

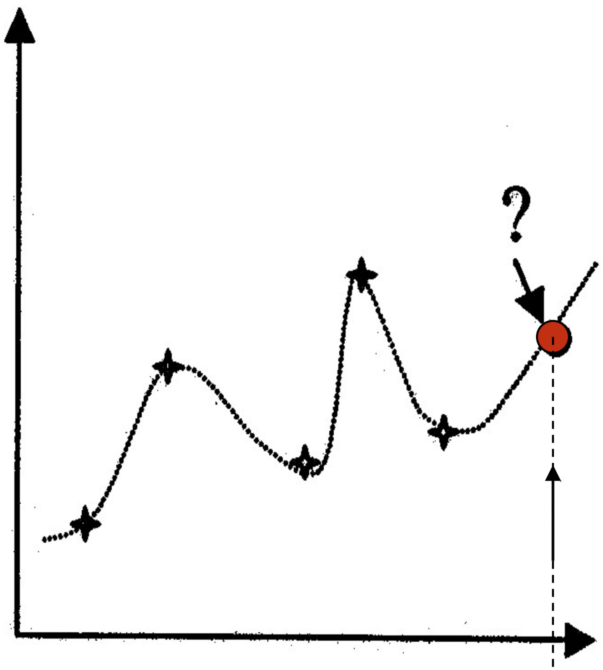
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?



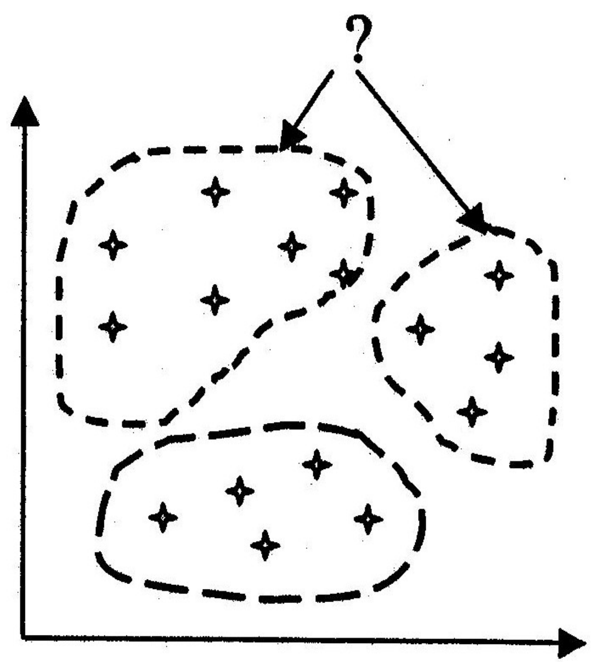
Data Mining Tasks: Prediction

- Prediction is a learning function that ***maps*** a ***sample*** to a real valued ***prediction attribute***
  - Given a collection of samples – nD points (*training set*)
  - Find *a* ***model*** for prediction ***attribute*** as a function of the other attributes
  - Goal: previously unseen sample should be assigned a value as accurately as possible (*test set*)



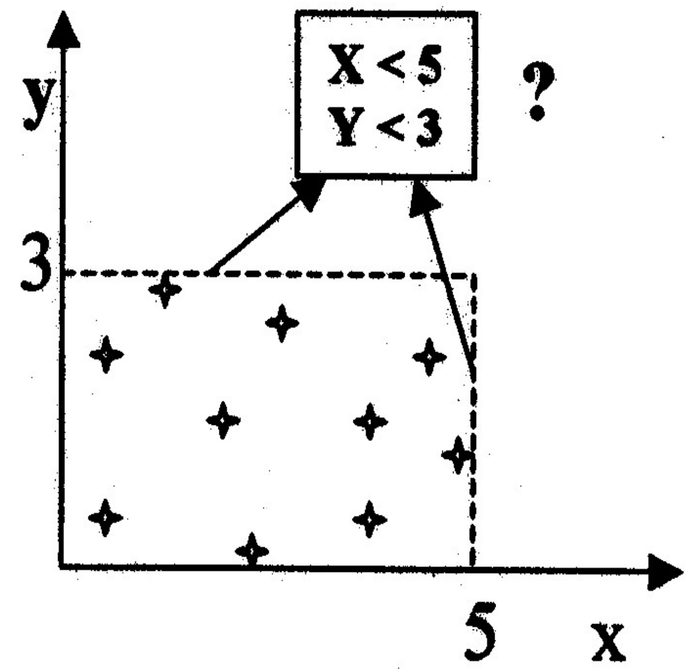
Data Mining Tasks: Clustering

- Clustering is a common descriptive task where one seeks to identify a finite ***set of categories*** (or clusters) to describe the data
  - Given a collection of samples – nD points
  - Find *a* ***model*** as a ***function*** of all attributes



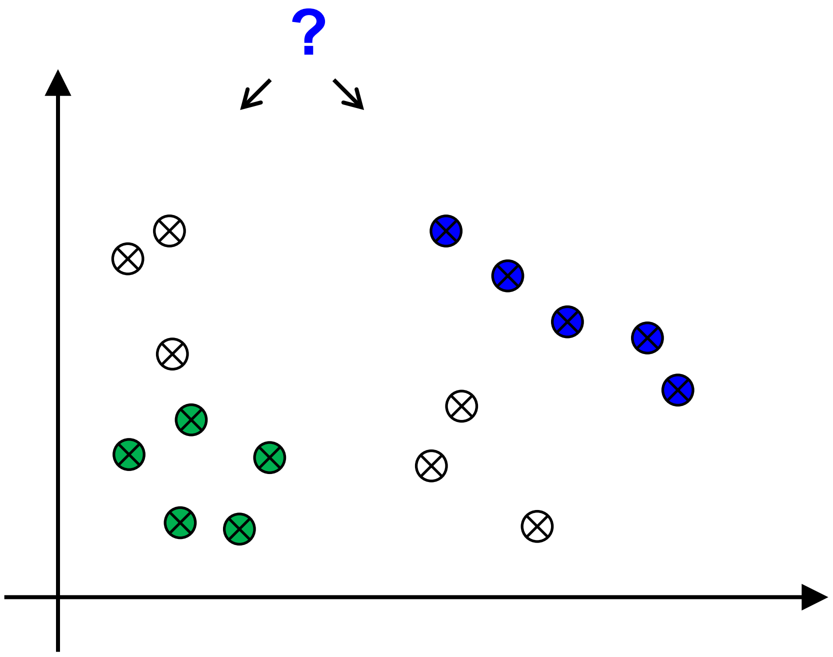
Data Mining Tasks: Summarization

- Summarization involves methods for finding *a* ***complete description*** for a set of samples.
  - Given a collection of samples – nD points
  - Find *a* short, ***simple descriptive model*** for samples as a function of all attributes.



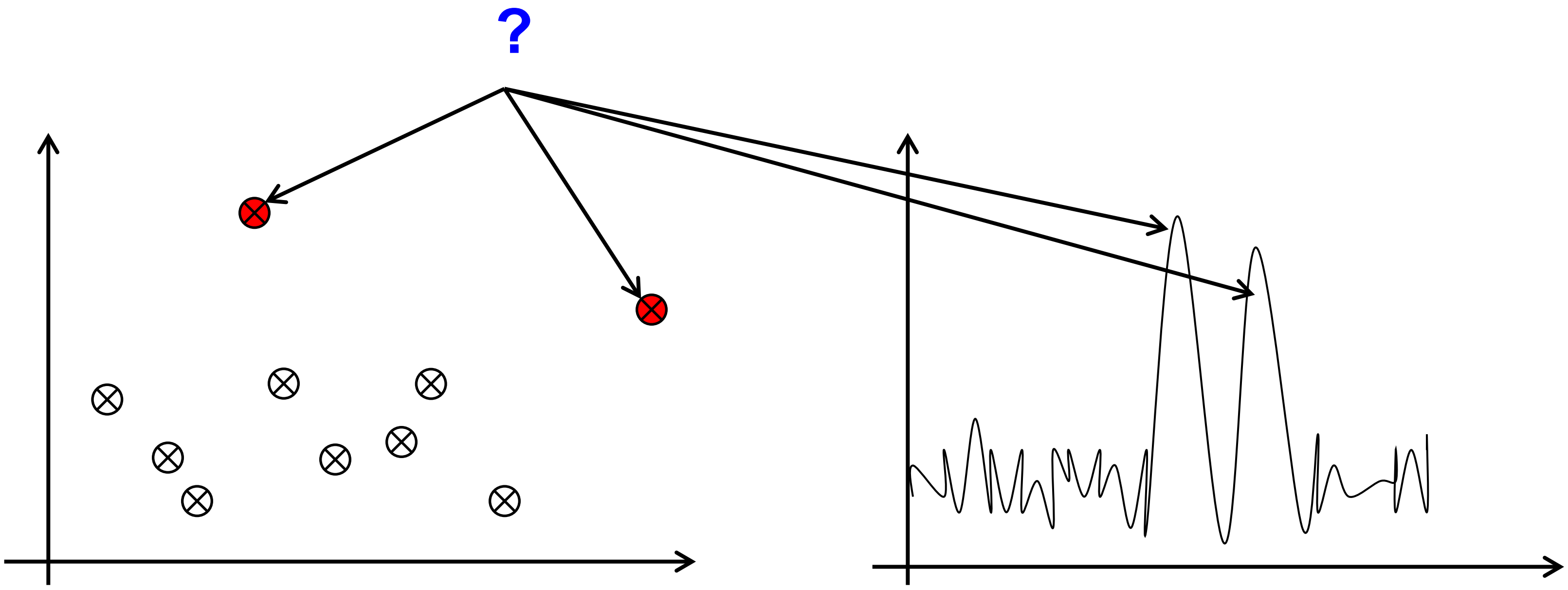
Data Mining Tasks: Dependency Modeling

- The task consists of finding a model that describes ***significant dependency*** in a set (subset) of samples.
  - Given a collection of samples – nD points
  - Find significant model(s) for set (subset) of samples – ***local models***



Data Mining Tasks: Change- & Deviation Detection

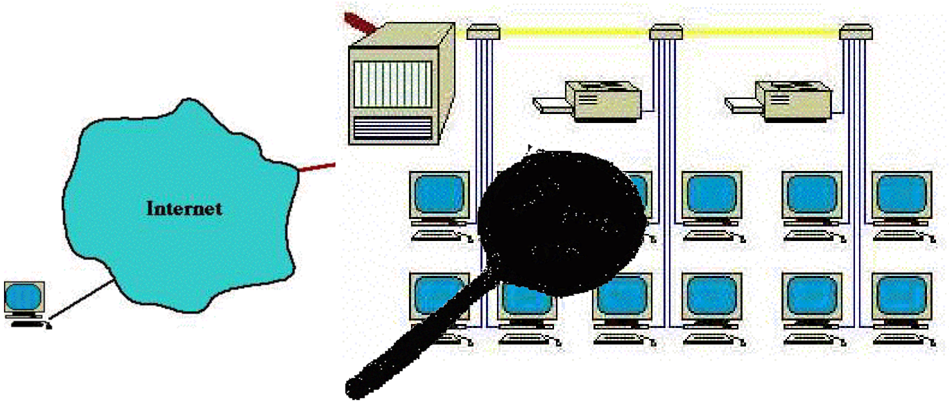
- Focuses on methods for discovering the most significant changes in large data sets.



Deviation Detection Example

- Detect significant deviations from normal behavior

- Applications:
  - Credit Card Fraud Detection
  - Network Intrusion Detection



Summarizing frequent Pattern Analysis as Association Rules

- ***Frequent pattern***: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of ***frequent itemsets*** and ***association rule mining***
- Motivation: Finding inherent regularities in data
  - What products were often purchased together?— Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?
- Applications
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis.

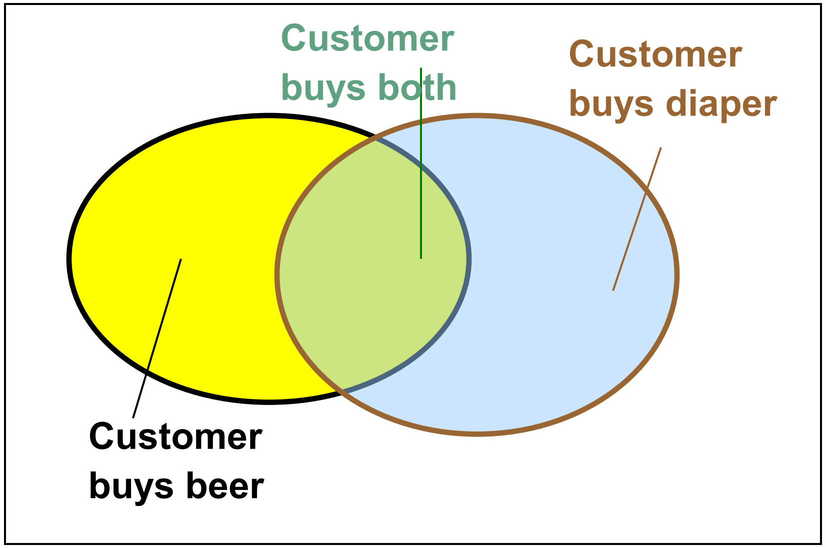
Why is freq. Pattern Mining important?

Why is freq. Pattern Mining important?

- Freq. pattern: An intrinsic and important property of datasets
- Foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: discriminative, frequent pattern analysis
  - Cluster analysis: frequent pattern-based clustering
  - Data warehousing
  - Semantic data compression
  - Broad applications

Basic Concepts: Frequent Patterns

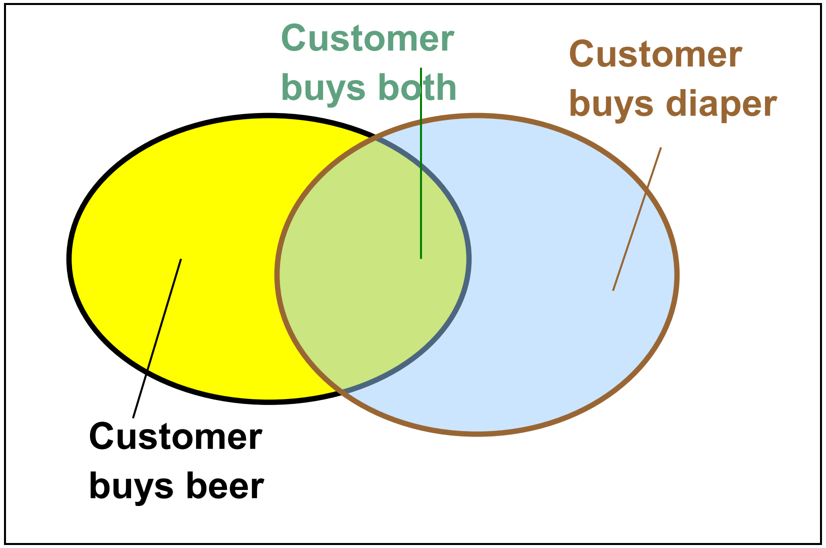
Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- ***itemset***: A set of one or more items
- ***k-itemset*** X = $\{x_1, \ldots, x_k\}$
- ***(absolute) support, or, support count*** of X: Frequency or occurrence of an itemset X
- ***(relative) support***, *s*, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X)
- An itemset X is ***frequent*** if X's support is no less than a *minsup* threshold

Basic Concepts: Association Rules

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - ***support***, *s*, probability that a transaction contains $X \cup Y$
  - ***confidence***, *c*, conditional probability that a transaction having X also contains *Y*

Let minsup = 50%, minconf = 50%

Freq. Pat.: Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

- Association rules: (many more!)
  - Beer ◊ Diaper (60%, 100%)
  - Diaper ◊ Beer (60%, 75%)

The downward Closure Property and scalable Mining Methods

- The ***downward closure*** property of frequent patterns
  - Any subset of a frequent itemset must be frequent
  - If **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}**
  - I.e., every transaction having {beer, diaper, nuts} also contains {beer, diaper}
- Scalable mining methods: Three major approaches
  - Apriori (Agrawal & Srikant@VLDB'94)
  - Freq. pattern growth (FPgrowth—Han, Pei & Yin @SIGMOD'00)
  - Vertical data format approach (Charm—Zaki & Hsiao @SDM'02)

Apriori: a Candidate Generation & Test Approach

- ***Apriori pruning principle***: If there is any itemset which is infrequent, its superset should not be generated/tested! [Agrawal & Srikant @VLDB'94, Mannila, et al. @ KDD' 94]
- Apriori name: use of prior knowledge of freq. itemset
- Method:
  - Initially, scan DB once to get frequent 1-itemset
  - ***Generate*** length (k+1) ***candidate*** itemsets from length k ***frequent*** itemsets
  - ***Test*** the candidates against DB
  - Terminate when no frequent or candidate set can be generated

The Apriori Algorithm (Pseudo-Code)

```
Ck: Candidate itemset of size k
Lk : frequent itemset of size k

L1 = {frequent items};
for (k = 1; Lk !=∅; k++) do begin
    Ck+1 = candidates generated from Lk;
    for each transaction t in database do
      increment the count of all candidates in Ck+1
      that are contained in t
    Lk+1  = candidates in Ck+1 with min_support
    end
return ∪k Lk;
```

The Apriori Algorithm – an Example

$Sup_{min} = 2$

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1[st] scan for count

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2[nd] scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3[rd] scan

L_3

Itemset	sup
{B, C, E}	2

Further Improvement of the Apriori Method

- Major computational challenges
  - Multiple scans of transaction database
  - Huge number of candidates
  - Tedious workload of support counting for candidates
- Improving Apriori: general ideas
  - Reduce passes of transaction database scans
  - Shrink number of candidates
  - Facilitate support counting of candidates

Partition: Scan Database only twice

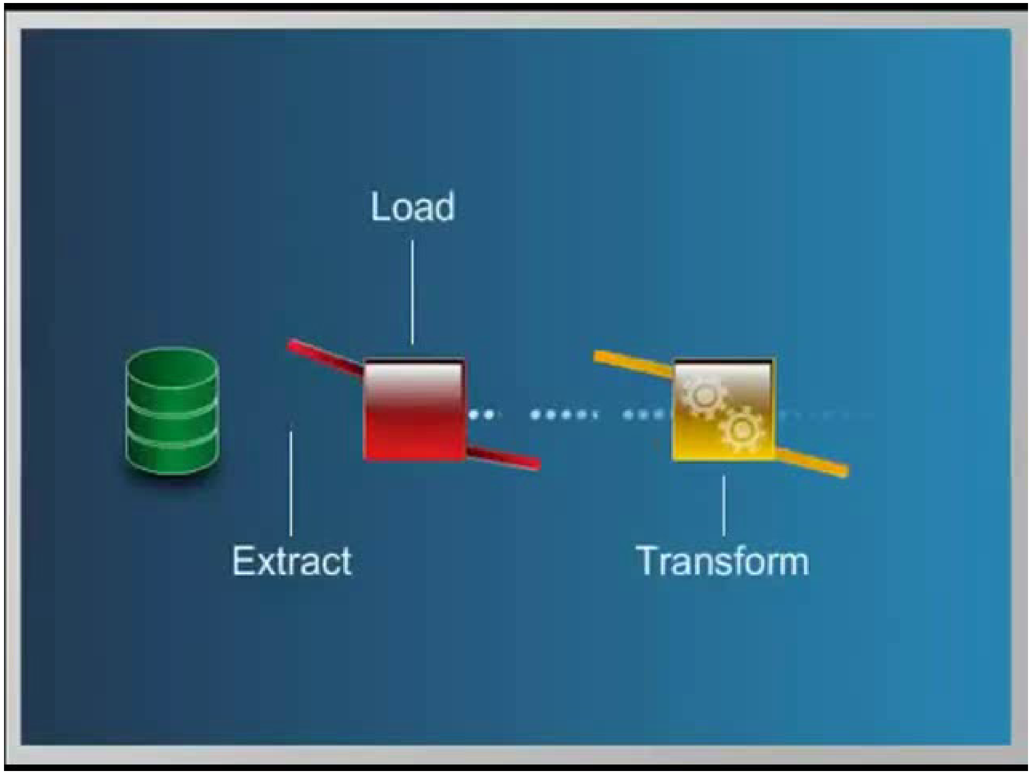
- Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
  - Scan 1: partition database and find local frequent patterns
  - Scan 2: consolidate global frequent patterns

[A. Savasere, E. Omiecinski and S. Navathe, *VLDB'95*]

DB_1 + DB_2 + . . . + DB_k = DB

$sup_1(i) < \sigma DB_1$ $\quad$ $sup_2(i) < \sigma DB_2$ $\quad$ $sup_k(i) < \sigma DB_k$ $\quad$ $sup(i) < \sigma DB$

Challenge of building a data warehouse



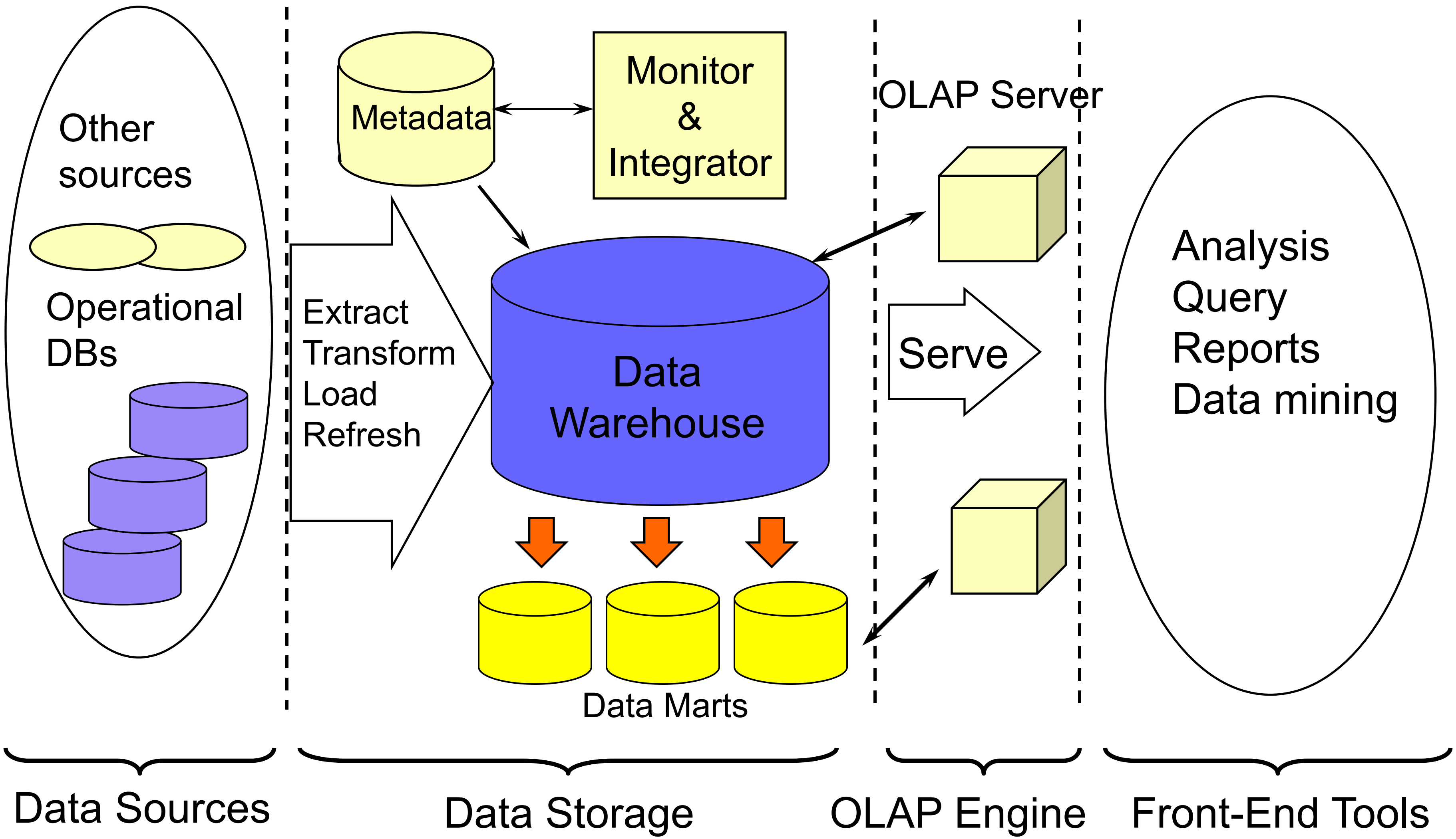
Associations as part of a Data Warehouse

- Defined in different ways....
  - A decision support database that is maintained ***separately*** from the organization's operational database
  - Support ***information processing*** by providing a solid platform of consolidated, historical data for analysis.
- "A data warehouse is a ***subject-oriented***, ***integrated***, ***time-variant***, and ***nonvolatile*** collection of data in support of management's decision-making process."—W. H. Inmon (nonvolatile=unvergänglich)
- Data warehousing:
  - The process of constructing and using data warehouses

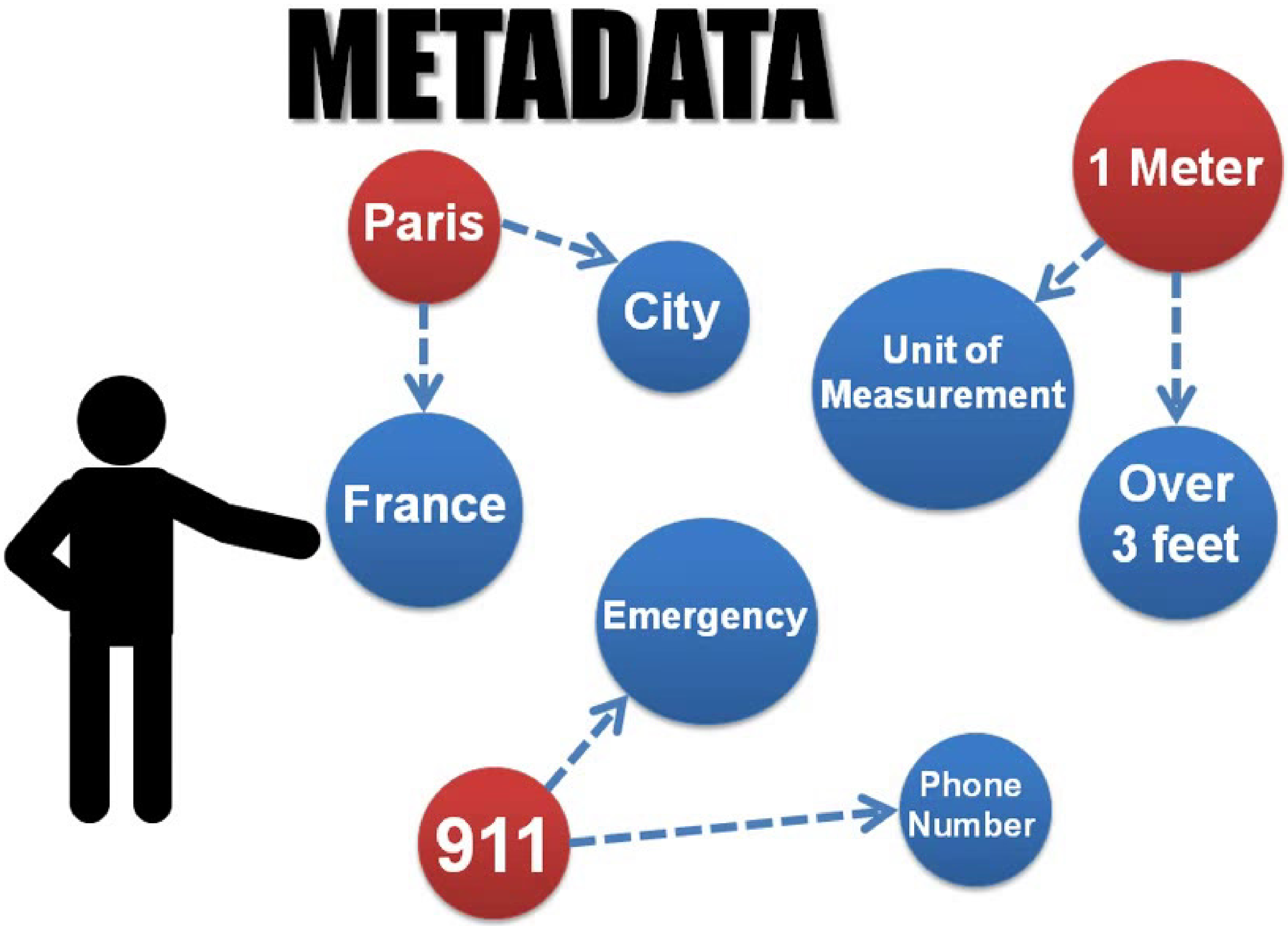
Online Transaction Processing (OLTP) vs. Online Analytical Processing (OLAP)

	OLTP	OLAP
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc, strategic
access	read/write	mostly read
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response time

Data Warehouse: a multi-tiered Architecture



Data warehouse for companies…



http://www.youtube.com/watch?v=KGHbY_Sales

Summary

- Learning from data
- Statistical learning theory
- Using data: cross validation
- Confusion matrix & evaluation metrics
- Data mining tasks
- Frequent pattern mining
- Data warehousing

Outlook: Research at WTM