Classification: Basic Concepts, Decision Trees, and Model Evaluation
Model Evaluation

Nicolás García-Pedrajas

Computational Intelligence and Bioinformatics Research Group

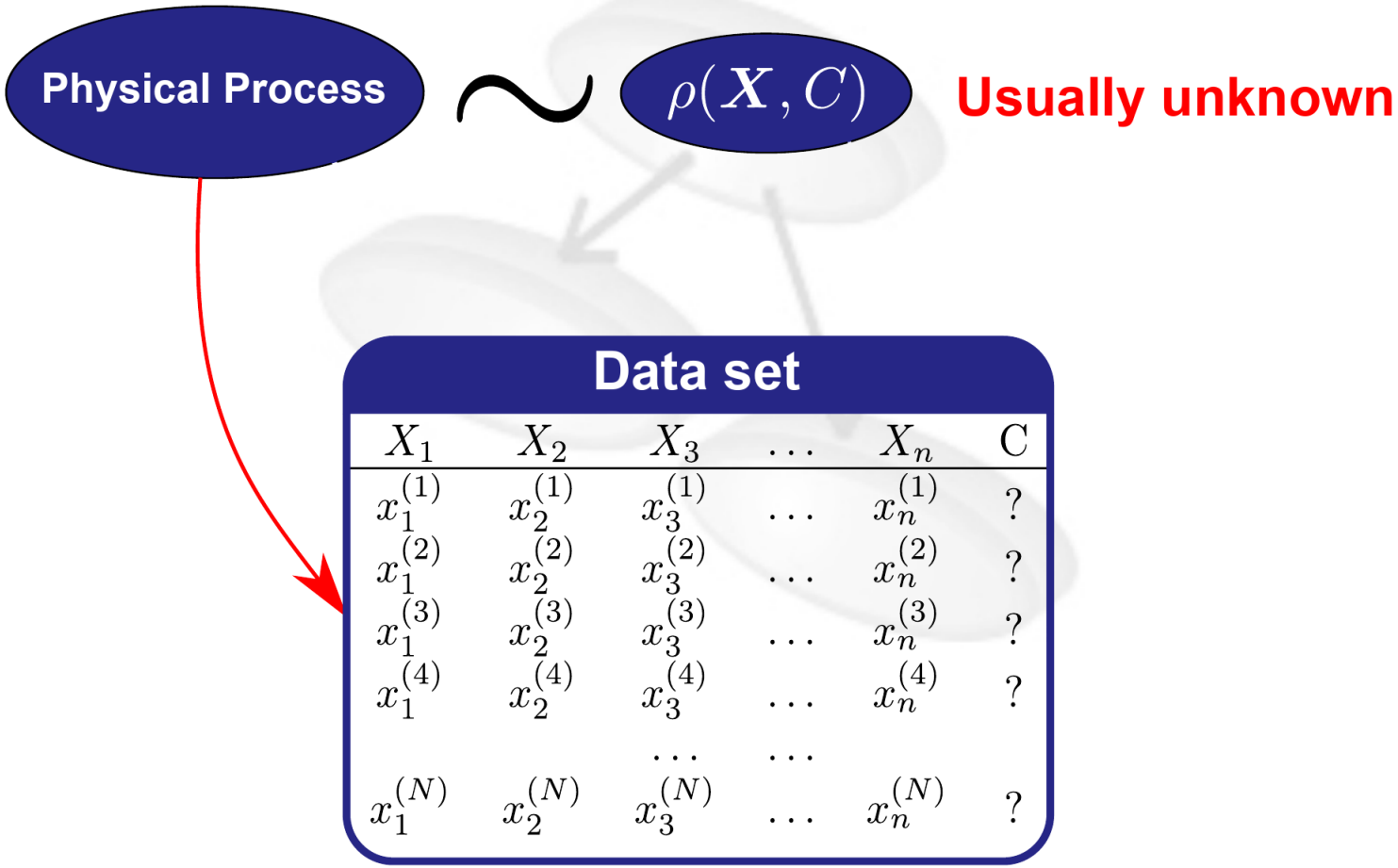
January 18, 2022

Table of contents

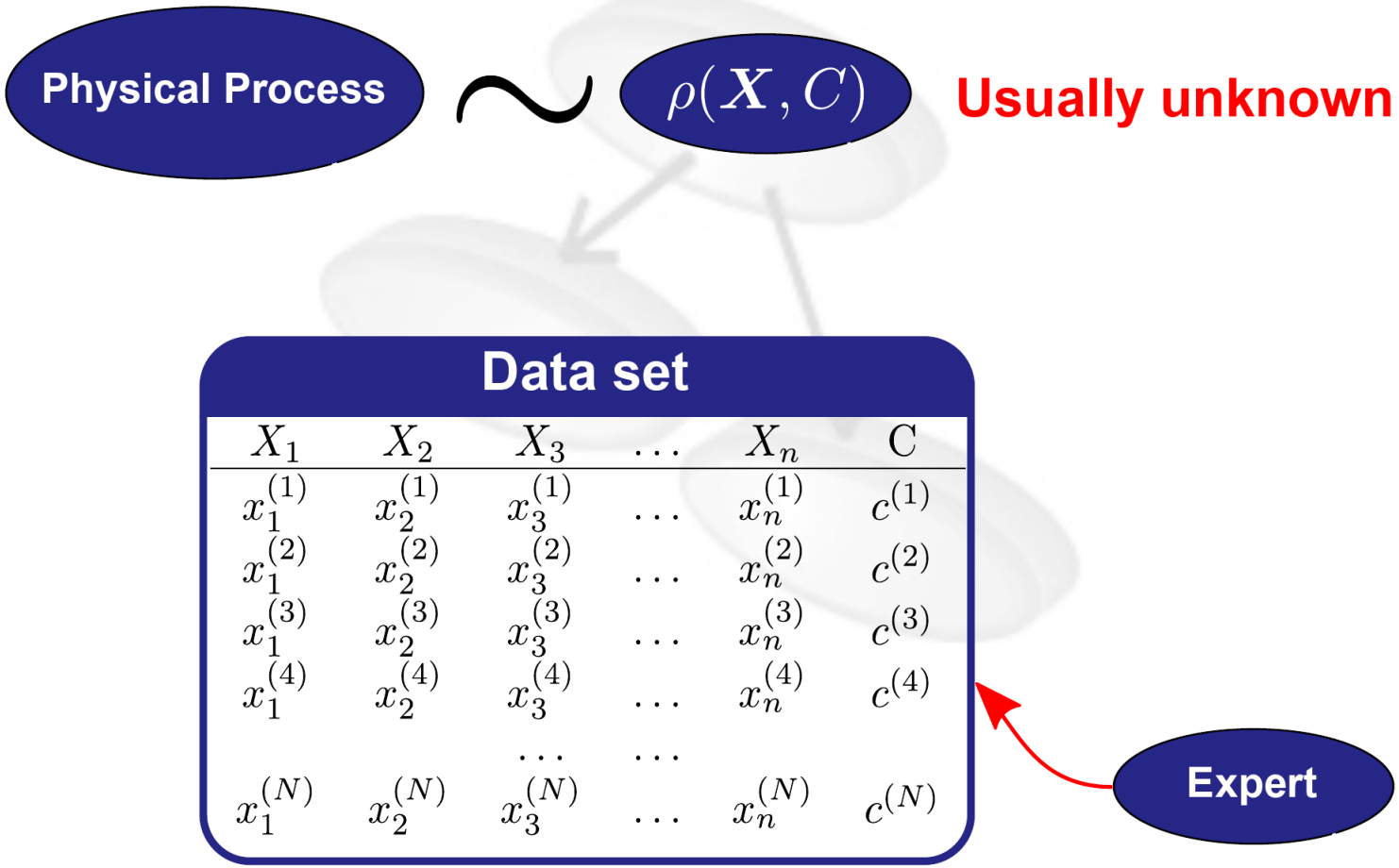
Based on Lozano, Santafé, and Inza (2010).

Basic concepts

Classification problem

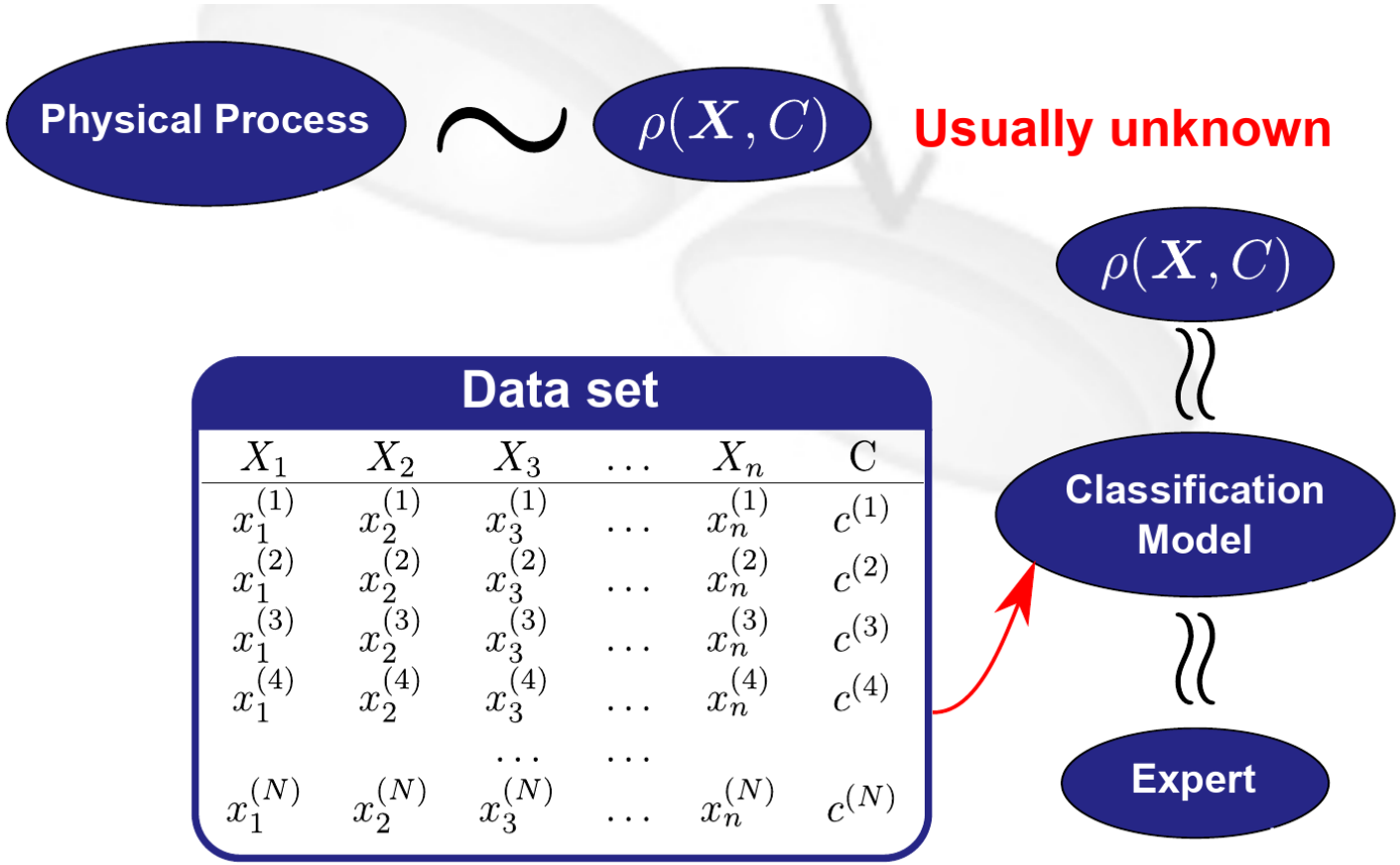


Classification problem



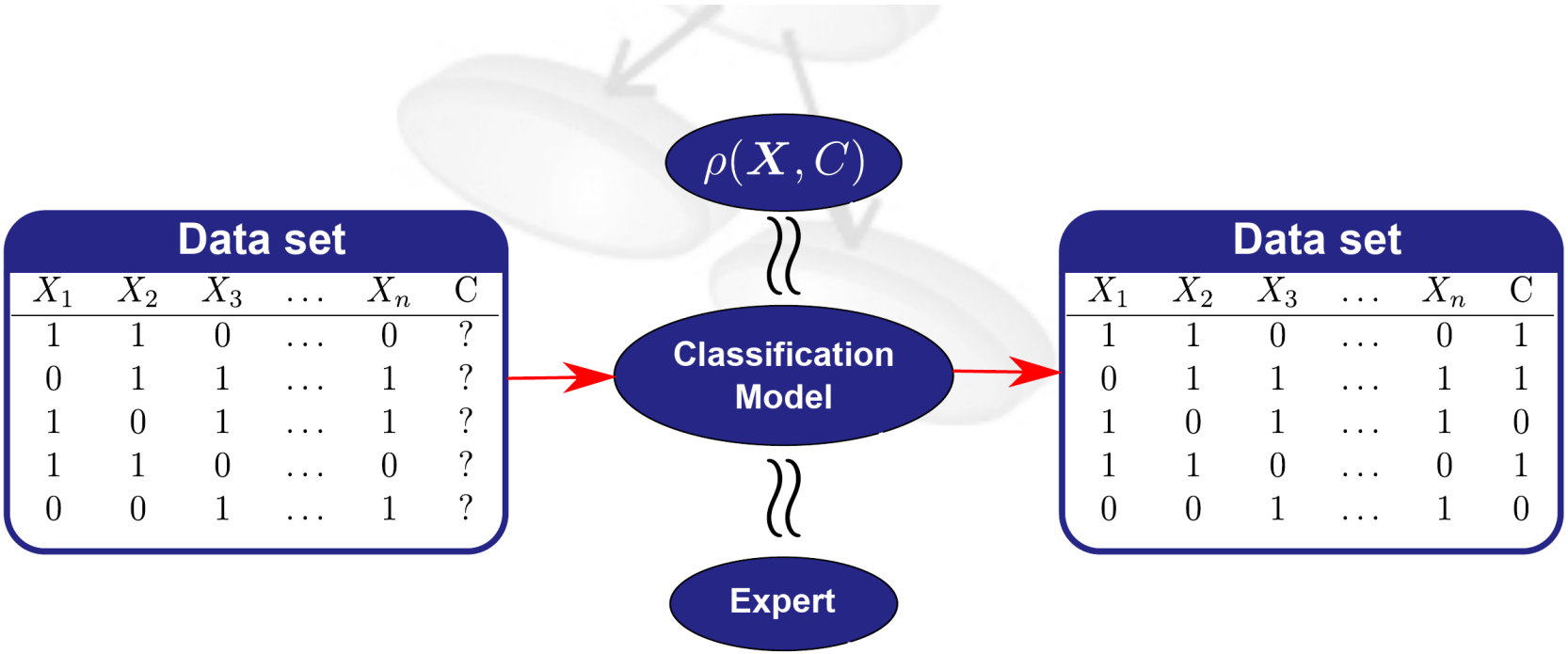
Supervised classification

- Automate the work of the expert
- Tries to model $\rho(\mathbf{x}, C)$



Classification model

Classifier labels new data (unknown class value)

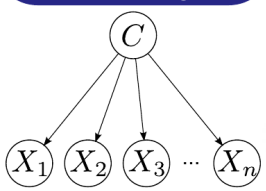


Motivation for evaluation

Motivation #1

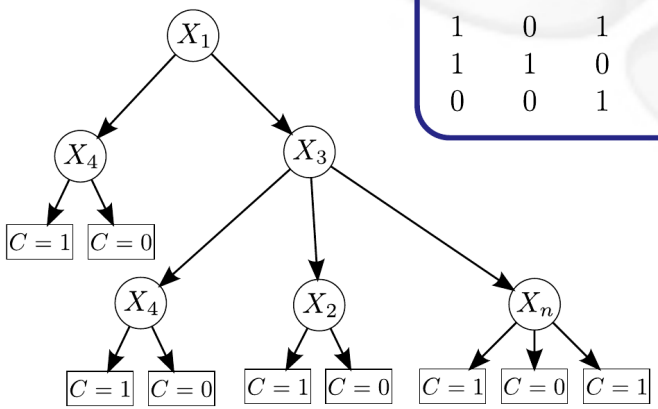
Many classification paradigms

Naive Bayes

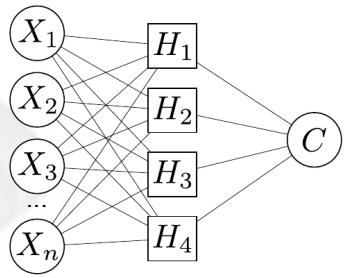


Data set

X_1	X_2	X_3	...	X_n	C
1	1	0	...	0	1
0	1	1	...	1	1
1	0	1	...	1	0
1	1	0	...	0	1
0	0	1	...	1	0



Decision Tree



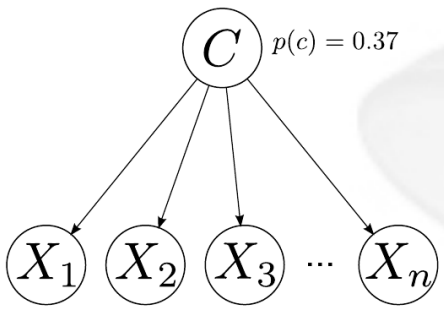
Neural Net

Motivation for evaluation

Motivation #2

Many parameter configurations

Naive Bayes

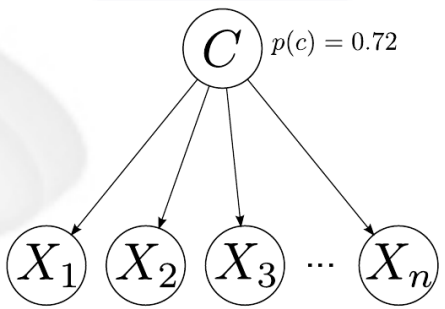


$p(x_1|c) = 0.25 \quad p(x_3|c) = 0.05$
$p(x_1|\bar{c}) = 0.60 \quad p(x_3|\bar{c}) = 0.60$
$p(x_2|c) = 0.20 \quad p(x_n|c) = 0.80$
$p(x_2|\bar{c}) = 0.70 \quad p(x_n|\bar{c}) = 0.21$

Data set

X_1	X_2	X_3	...	X_n	C
1	1	0	...	0	1
0	1	1	...	1	1
1	0	1	...	1	0
1	1	0	...	0	1
0	0	1	...	1	0

Naive Bayes



$p(x_1|c) = 0.11 \quad p(x_3|c) = 0.87$
$p(x_1|\bar{c}) = 0.80 \quad p(x_3|\bar{c}) = 0.65$
$p(x_2|c) = 0.99 \quad p(x_n|c) = 0.43$
$p(x_2|\bar{c}) = 0.50 \quad p(x_n|\bar{c}) = 0.16$

Motivation for evaluation

- Evaluation
  - ✓ Need to know the goodness of a classifier
  - ✓ Methodology to compare classifiers
  - ✓ Assess the validity of evaluation/comparison
- Steps for Evaluation
  - ✓ Scores: quality measures
  - ✓ Estimation methods: estimate value of a score
  - ✓ Statistical tests: comparison among different solutions

Scores

Motivation

- How to compare classification models?
- We need some way to compare the classification performance

Score

Function that provides a quality measure for a classifier when solving a classification problem

Motivation

- What Does Best Quality Mean?
  - ✓ What are we interested in?
  - ✓ What do we want to optimize?
  - ✓ Characteristics of the problem
  - ✓ Characteristics of the data set

Different kind of scores

Scores

- Based on Confusion Matrix
  - ✓ Accuracy/Error
  - ✓ Recall = Sensitivity = True positive rate
  - ✓ Specificity = True negative rate
  - ✓ Precision
  - ✓ F-Score
- Based on Receiver Operating Characteristics (ROC) curve
  - ✓ Area under the ROC curve (auROC)
- Based in Precision/Recall Curve (PRC)
  - ✓ Area under de PRC (auPRC)

Confusion matrix

Two-class problems

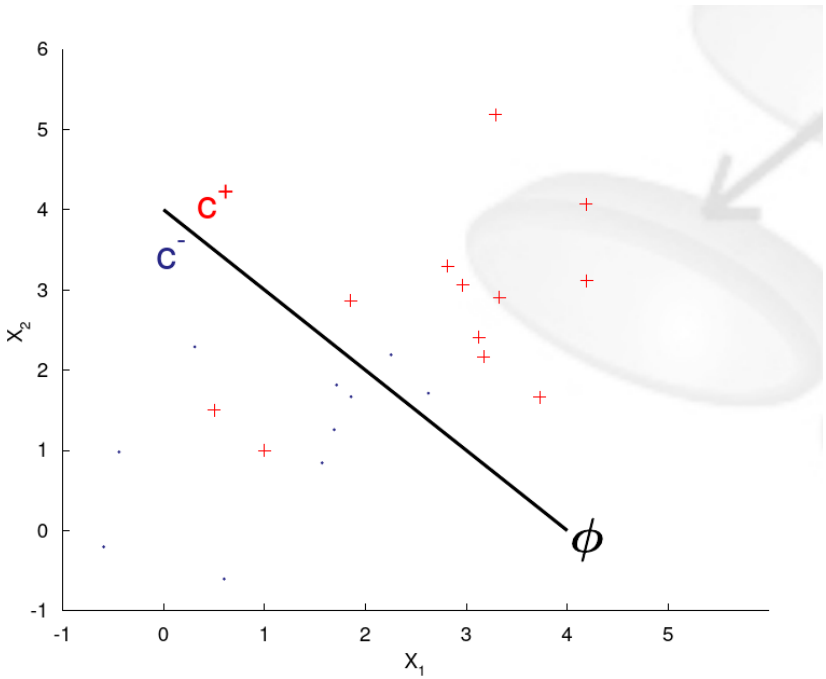
		Prediction		
		c^+	c^-	Total
Actual	c^+	**TP**	**FN**	N^+
	c^-	**FP**	**TN**	N^-
	Total	$\hat{N}^+$	$\hat{N}^-$	N

Confusion matrix

Multi-class problems

		Prediction					
		c_1	c_2	c_3	$\cdots$	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	$\cdots$	FN_{1n}	N_1
	c_2	FN_{21}	TP_2	FN_{23}	$\cdots$	FN_{2n}	N_2
	c_3	FN_{31}	FN_{32}	TP_3	$\cdots$	FN_{3n}	N_3
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	$\cdots$	TP_n	N_n
	Total	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_2$	$\cdots$	$\hat{N}_n$	N

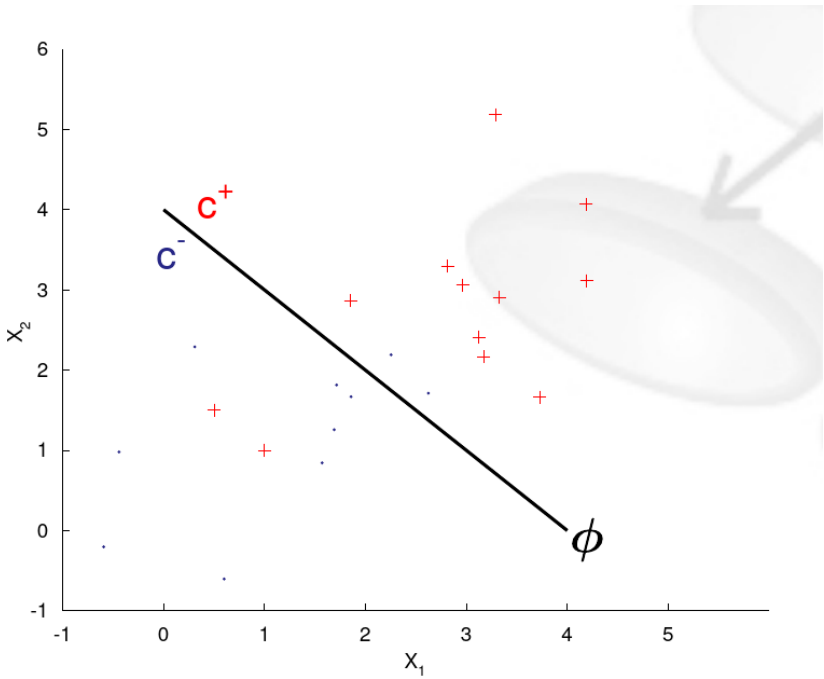
Two-class problem: Example



		Prediction		
		c^+	c^-	Total
Actual	c^+	**10**	**2**	12
	c^-	**2**	**8**	10
	Total	12	10	22

Accuracy/classification error

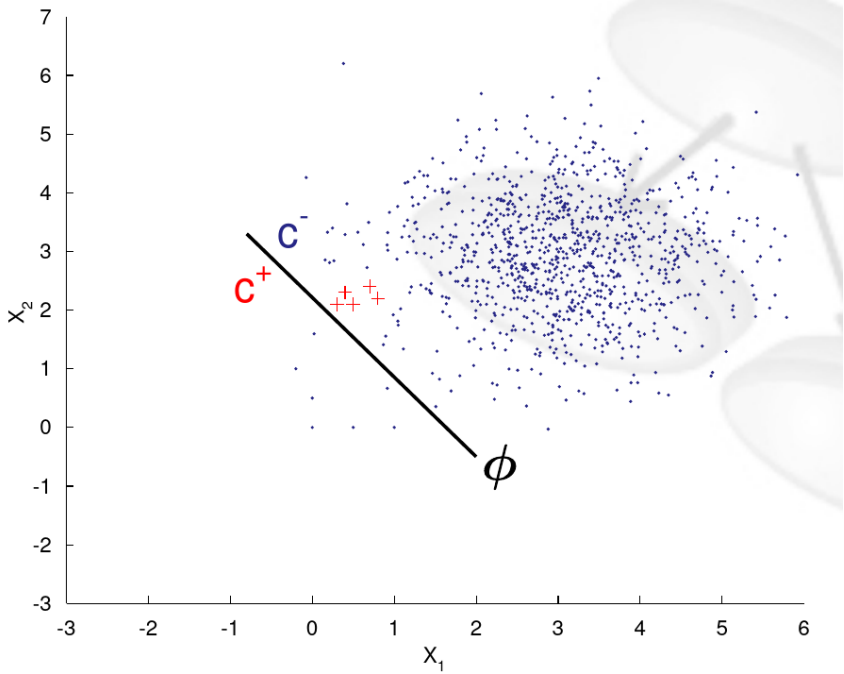
Data samples classifier correctly



		Prediction		
		c^+	c^-	Total
Actual	c^+	**10**	**2**	12
	c^-	**2**	**8**	10
	Total	12	10	22

$$\epsilon = \frac{FP+FN}{N} = \frac{2+2}{22} = 0.182$$

Skew data: Classification error

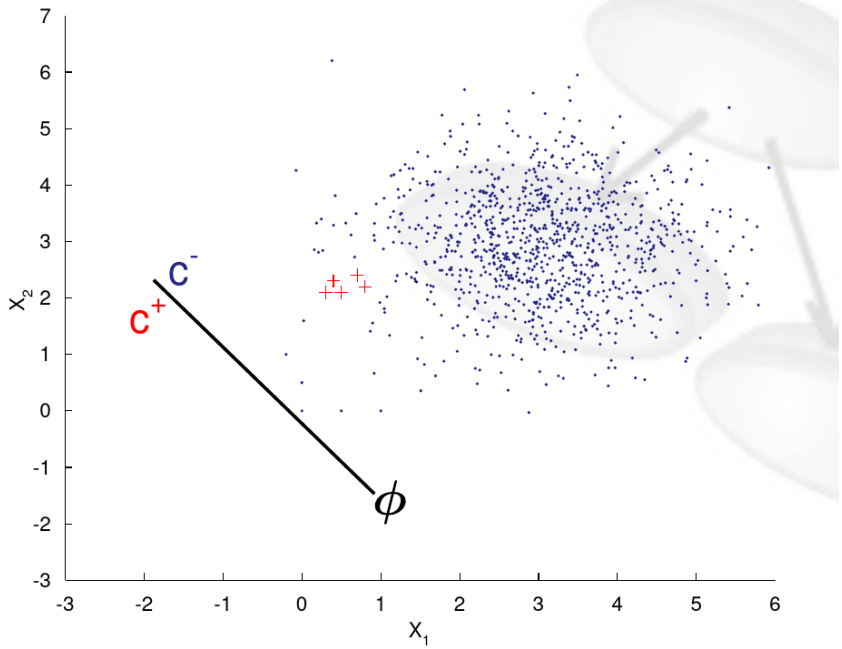


		Prediction		
		c^+	c^-	Total
Actual	c^+	0	5	5
	c^-	7	993	1000
	Total	7	998	1005

$$\epsilon = \frac{FP+FN}{N} = \frac{7+5}{1005} = 0.012$$

VERY LOW!!!

Skew data: Classification error

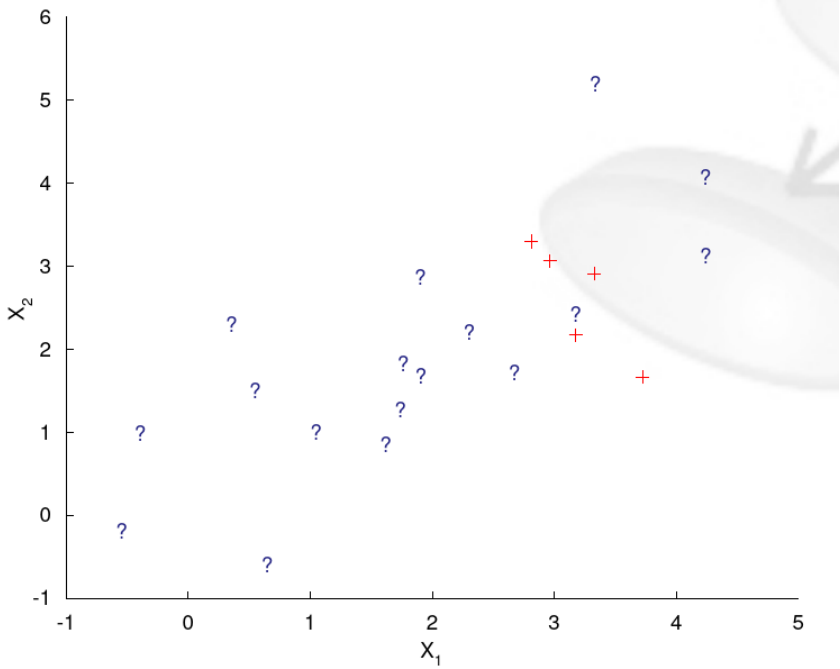


		Prediction		
		c^+	c^-	Total
Actual	c^+	0	5	5
	c^-	0	1000	1000
	Total	0	1005	1005

$$\epsilon = \frac{FP+FN}{N} = \frac{0+5}{1005} = 0.005$$

BETTER!!!

Positive unlabeled learning

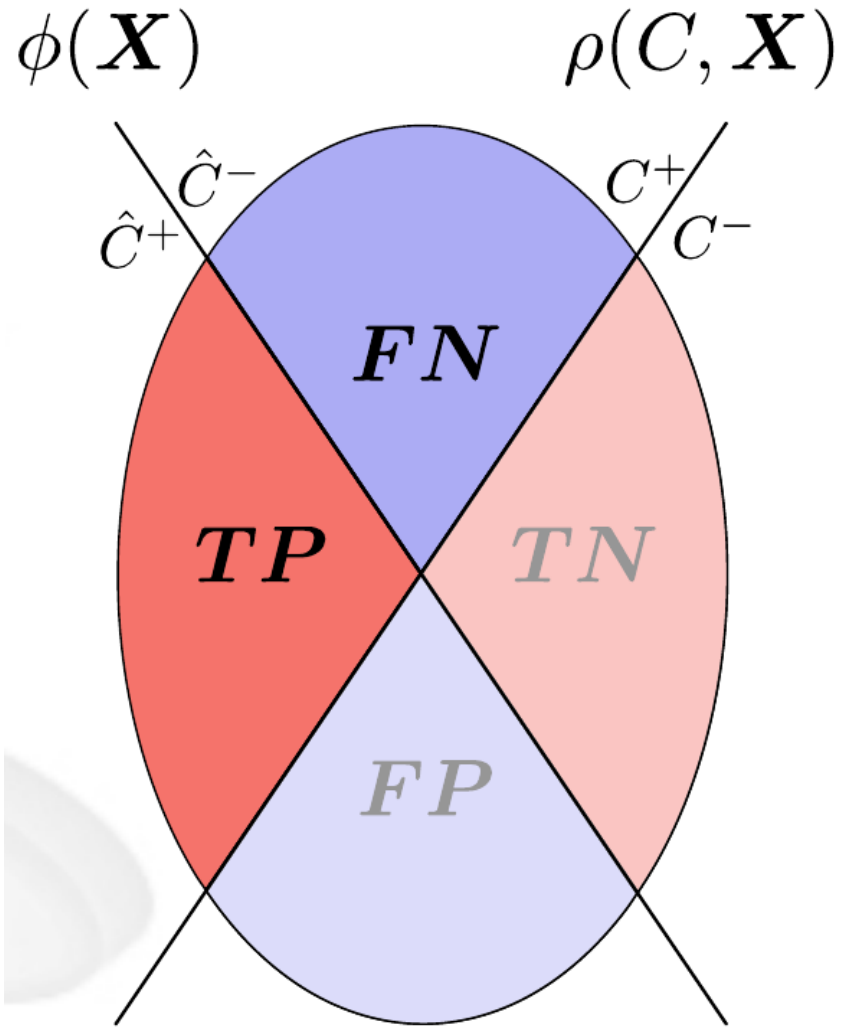


- Positive labeled data
  - ✓ Only positive samples labeled
  - ✓ Many unlabeled samples
    - Positive?
    - Negative?
  - ✓ Classification error is useless

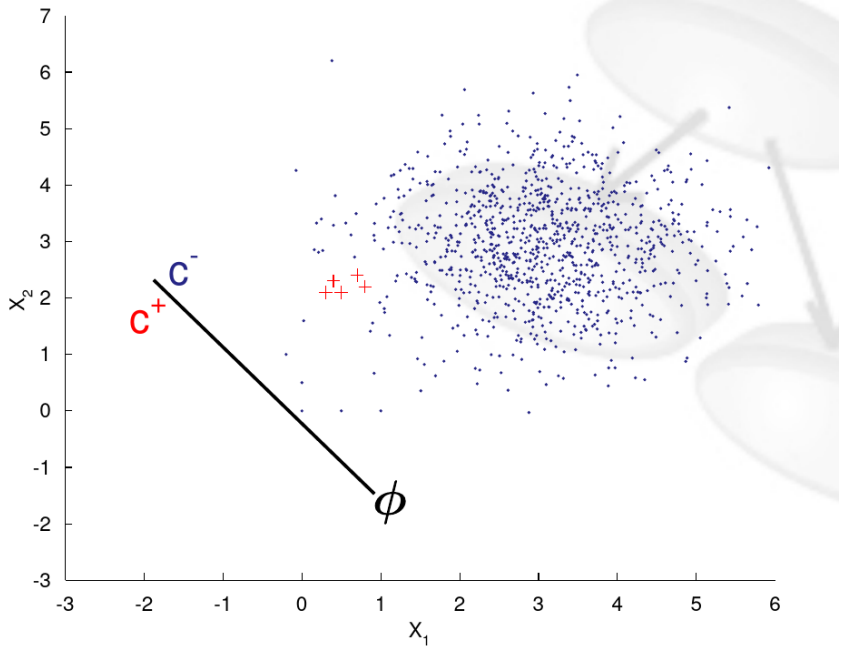
Recall (R), sensitivity (Sn), True positive rate (TP_{rate})

Fraction of positive class samples correctly classified:

$$R = \frac{TP}{TP+FN} = \frac{TP}{P}$$



Skew data: Recall

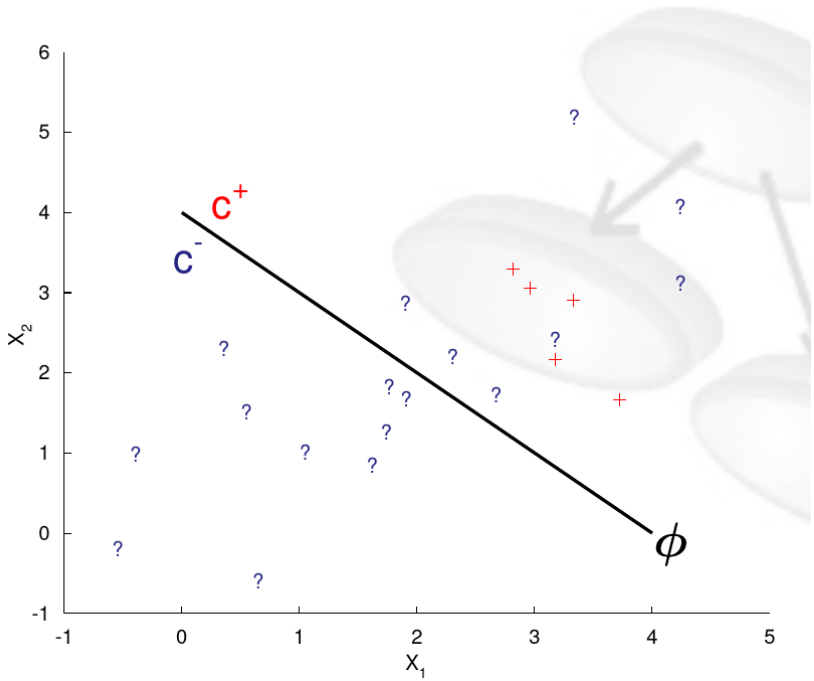


		Prediction		
		c^+	c^-	Total
Actual	c^+	0	5	5
	c^-	0	1000	1000
	Total	0	1005	1005

$$R = \frac{TP}{P} = \frac{0}{0+5} = 0$$

VERY BAD RECALL!!!

Positive Unlabeled Learning - Recall



		Prediction		
		c^+	$c^?$	Total
Actual	c^+	5	0	5
	$c^?$	7	10	17
	Total	12	10	22

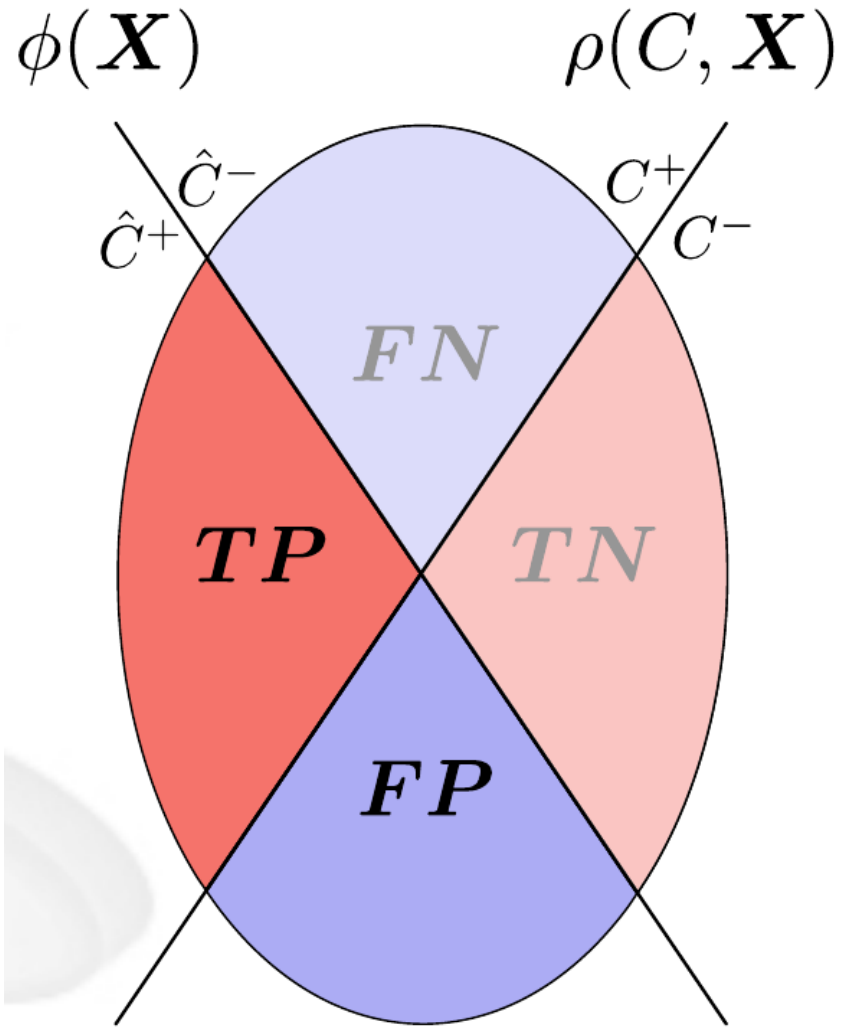
$$R = \frac{TP}{P} = \frac{5}{0+5} = 1$$

It is possible to calculate recall for positive-unlabeled problems

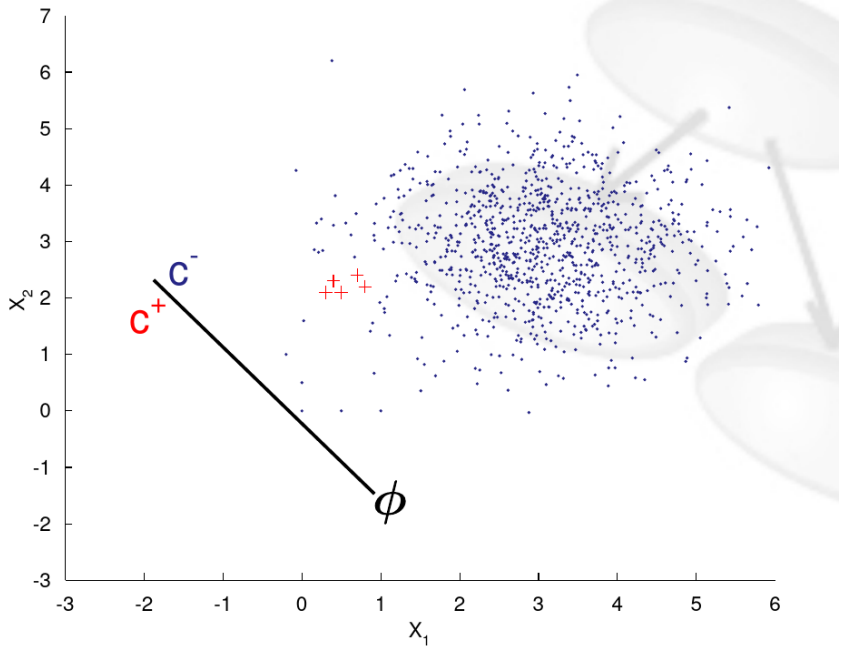
Precision (P), Positive predictive value (PP_{value}), Specificity in Bioinformatics

Fraction of data samples classified as positive that are actually positive:

$$P = \frac{TP}{TP+FP} = \frac{TP}{\hat{P}}$$



Skew data: Precision

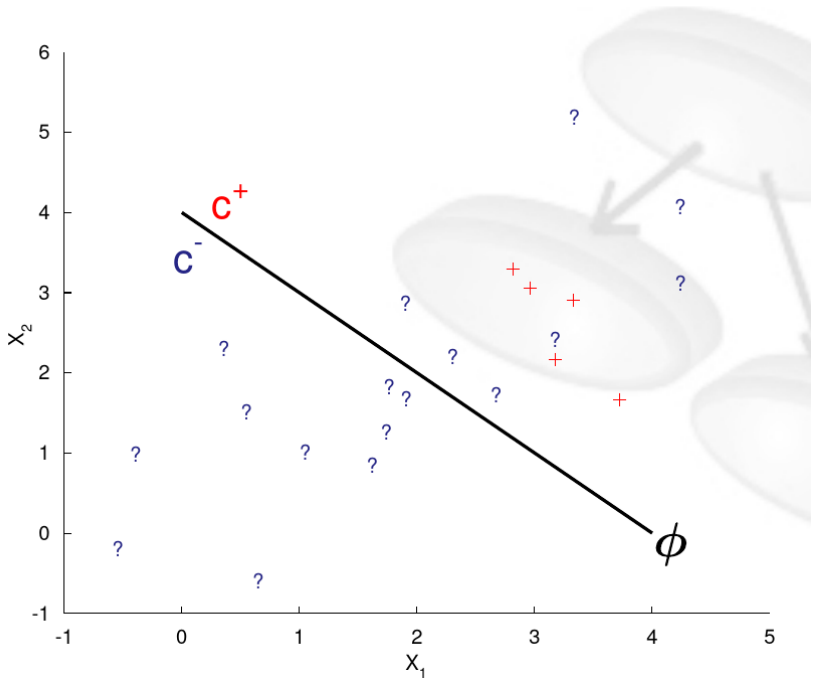


		Prediction		
		c^+	c^-	Total
Actual	c^+	0	5	5
	c^-	7	993	1000
	Total	7	998	1005

$$P = \frac{TP}{\hat{P}} = \frac{0}{0+7} = 0$$

VERY BAD PRECISION!!!

Positive Unlabeled Learning - Precision



- Precision is not a good score for positive-unlabeled data samples
- Not all the positive samples are labeled

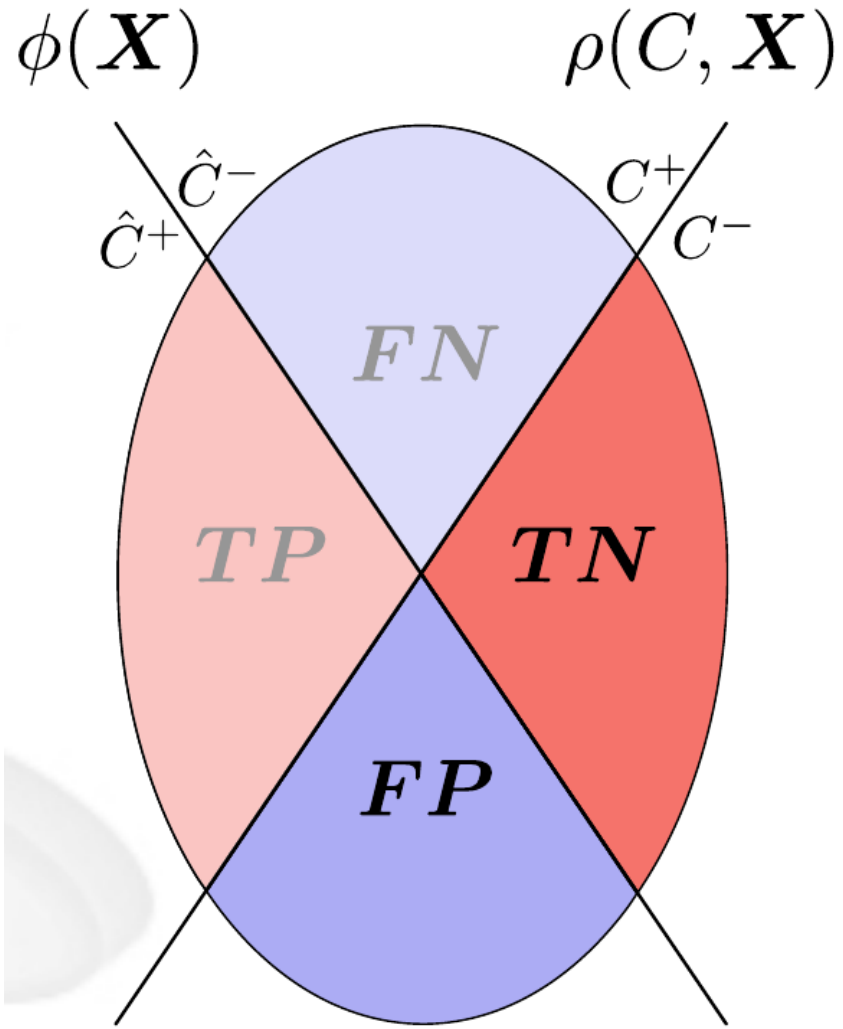
Precision & Recall Application Domains

- Spam Filtering
  - ✓ Decide if an email is spam or not
    - Precision: Proportion of real spam in the spam-box
    - Recall: Proportion of total spam messages identified by the system
- Sentiment Analysis
  - ✓ Classify opinions about specific products given by users in blogs, webs, forums, etc.
    - Precision: Proportion of opinions classified as positive being actually positive
    - Recall: Proportion of positive opinions identified as positive

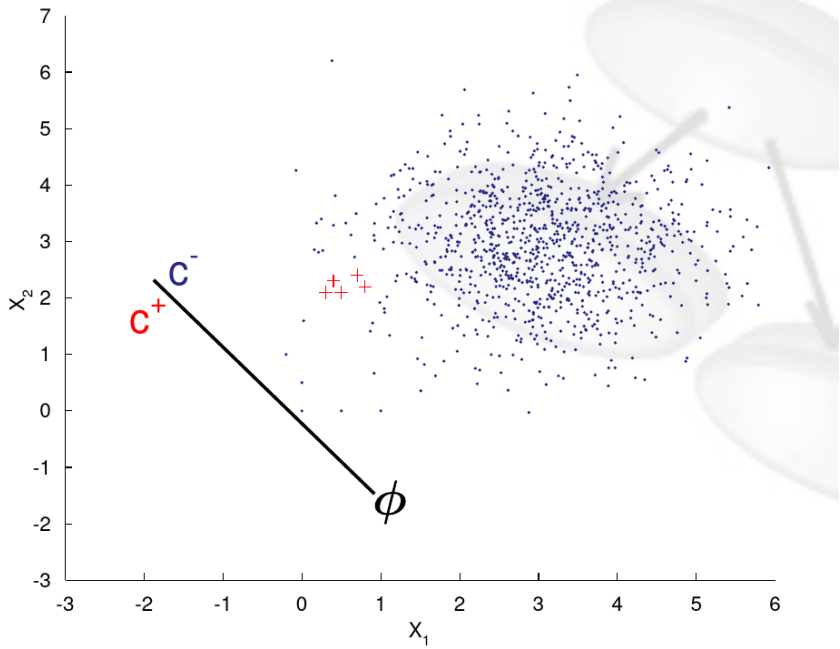
Specificity (Sp), True negative rate (TN_{rate})

- Fraction of negative class samples correctly identified
- Specificity = 1 - False positive rate

$$Sp = \frac{TN}{TN+FP} = \frac{TN}{N}$$



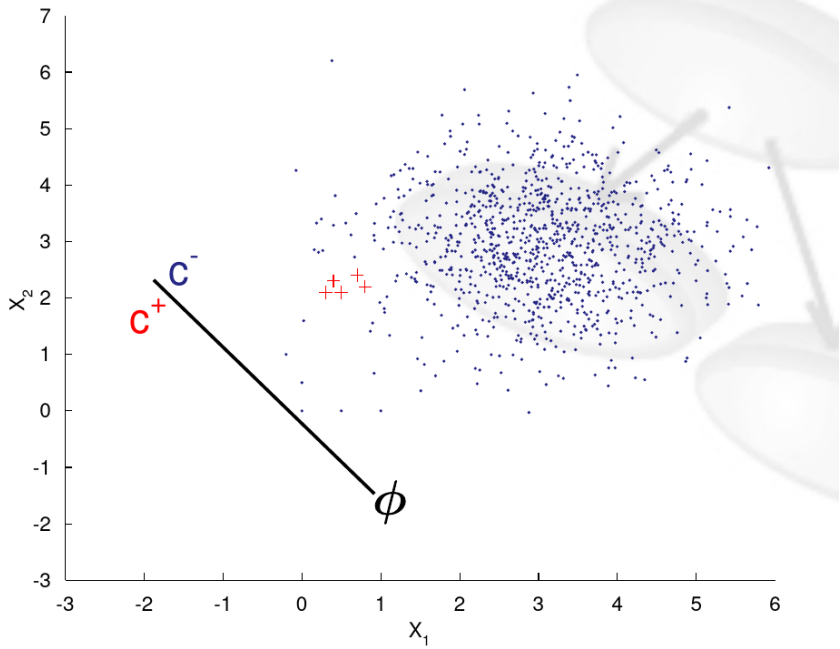
Skew data: Specificity



		Prediction		
		c^+	c^-	Total
Actual	c^+	0	5	5
	c^-	7	993	1000
	Total	7	998	1005

$$Sp = \frac{TN}{N} = \frac{993}{993+7} = 0.99$$

Skew data: Specificity



		Prediction		
		c^+	c^-	Total
Actual	c^+	**0**	5	5
	c^-	**0**	**1000**	1005
	Total	0	1005	1005

$$Sp = \frac{TP}{\hat{P}} = \frac{1000}{1000+0} = 1.00$$

Other measures

- False positive rate (FP_{rate})
  $FP_{rate} = \frac{FP}{TN+FP} = \frac{FP}{N}$
- False negative rate (FN_{rate})
  $FN_{rate} = \frac{FN}{TP+FN} = \frac{FN}{P}$
- Negative predictive value (NP_{value})
  $NP_{value} = \frac{TN}{TN+FN} = \frac{TN}{\hat{N}}$

Balanced scores

Combine different metrics

- Balanced accuracy rate
  $Bal.acc = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right) = \frac{Sn+Sp}{2}$
- Balanced error rate
  $Bal.\epsilon = \frac{1}{2}\left(\frac{FP}{P} + \frac{TN}{N}\right)$

Skew data

		Prediction		
		c^+	c^-	Total
Actual	c^+	0	5	5
	c^-	7	993	1000
	Total	7	998	1005

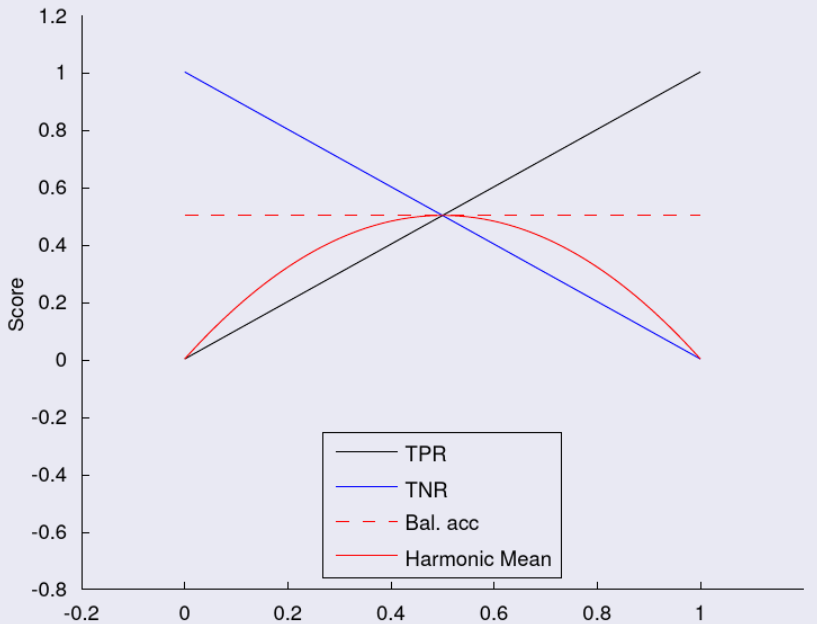
$$Bal.acc = \frac{1}{2}\left(\frac{0}{5} + \frac{993}{1000}\right) \approx 0.5$$
$$Bal.\epsilon = \frac{1}{2}\left(\frac{7}{7} + \frac{5}{1000}\right) \approx 0.5$$

Balanced scores

- $F - score = \frac{(\beta^2+1)Precision \cdot Recall}{\beta^2(Precision+Recall)}$
- $F_1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision+Recall} \longrightarrow$ Harmonic mean
- $G - mean = \sqrt{Specificity \cdot Sensitivity}$

Harmonic mean

- Maximized when balanced components
- Bal. acc $\longrightarrow$ Arithmetic mean



Balanced scores

Correlation coefficient (CC)

$$CC = \frac{(TP \cdot TN)(FP \cdot FN)}{\sqrt{(TP+FP)(TN+FN)(TP+FN)(TN+FP)}}$$

Classification cost

- All misclassifications cannot be equally considered
  - ✓ Medical Diagnosis Problem
    - Does not have the same cost as diagnosing a healthy patient as ill rather than diagnosing an ill patient as healthy
  - ✓ Classification Model
    - May be of interest to minimize the expected cost instead the classification error

Dealing with Classification Cost

Loss function

Associate an economic/utility/etc. cost to each classification
Typical loss function in classification ⟶ 0/1 Loss

We can use a cost matrix to specify the associated cost:

		Prediction c^+	Prediction c^-
Actual	c^+	0	1
	c^-	1	0

Dealing with Classification Cost

The associated cost matrix is defined by the expert

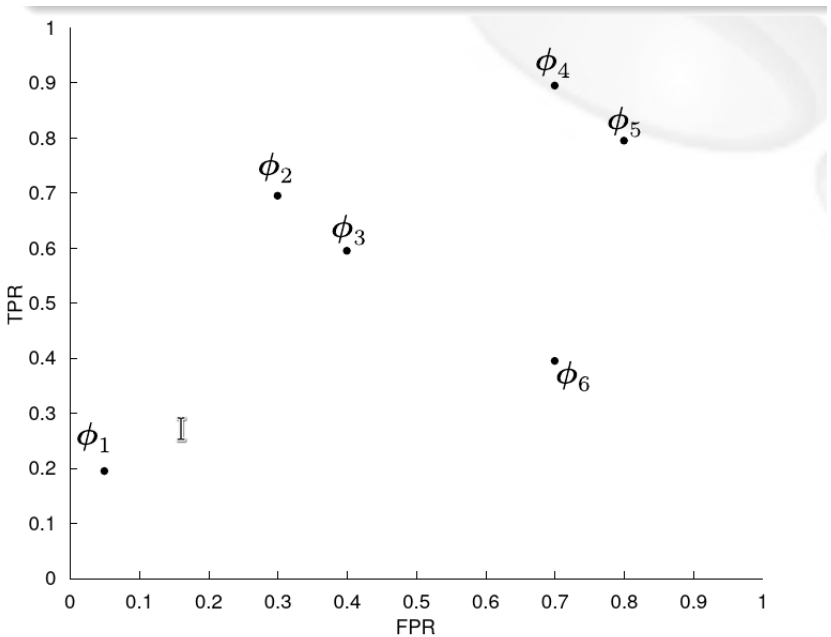
		Prediction	
		c^+	c^-
Actual	c^+	TP_{cost}	FN_{cost}
	c^-	FP_{cost}	TN_{cost}

Usually not easy to give an associated cost

Receiver Operating Characteristics (ROC) curve

ROC space

Coordinate system used for visualizing classifiers performance where TPR is plotted on the Y axis and FPR is plotted on the X axis.

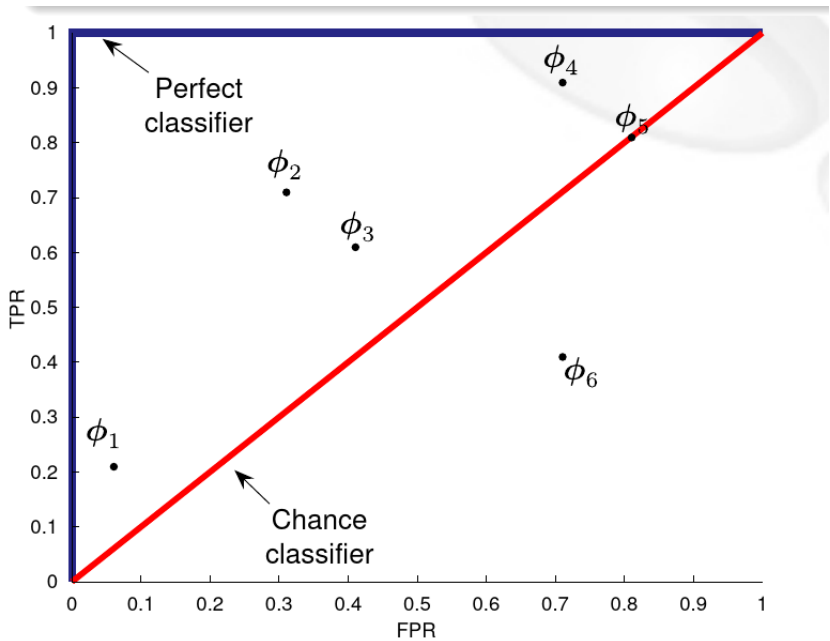


- ϕ_1: k-NN
- ϕ_2: Neural network
- ϕ_3: Naïve Bayes
- ϕ_4: SVM
- ϕ_5: Linear regression
- ϕ_6: Decision tree

Receiver Operating Characteristics (ROC) curve

ROC space

Coordinate system used for visualizing classifiers performance where TPR is plotted on the Y axis and FPR is plotted on the X axis.

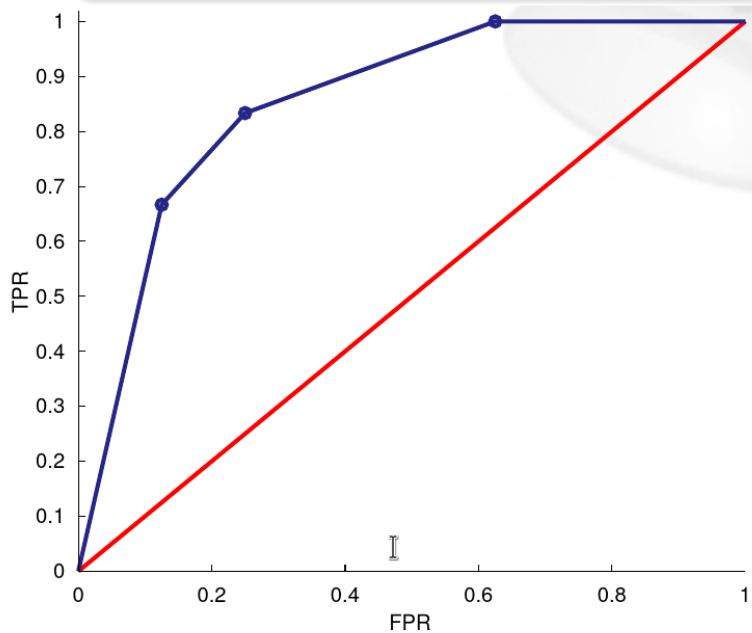


- ϕ_1: k-NN
- ϕ_2: Neural network
- ϕ_3: Naïve Bayes
- ϕ_4: SVM
- ϕ_5: Linear regression
- ϕ_6: Decision tree

Receiver Operating Characteristics (ROC) curve

ROC curve

For a probabilistic/fuzzy classifier, a ROC curve is a plot of the TPR vs. FPR as its discrimination threshold is varied

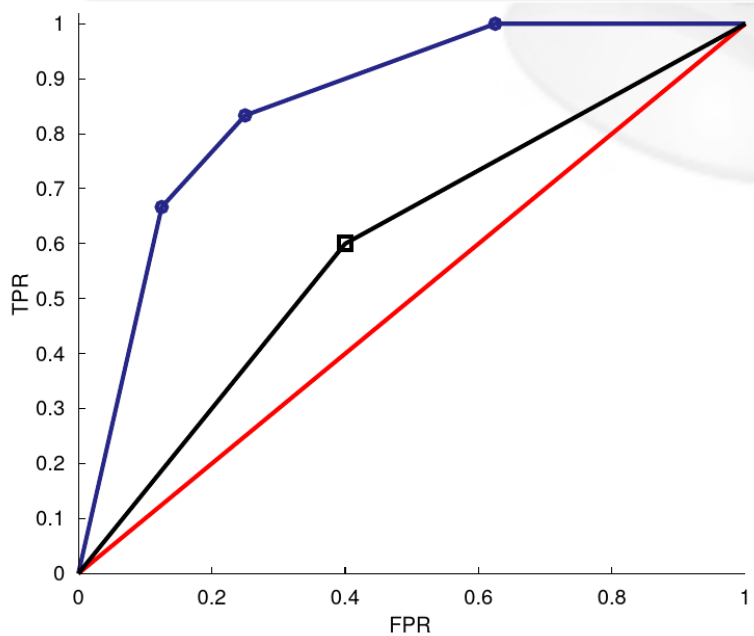


$p(c\|\boldsymbol{x})$	$T = 0{,}2$	$T = 0{,}5$	$T = 0{,}8$	C
0,99	c^+	c^+	c^+	c^+
0,90	c^+	c^+	c^+	c^+
0,85	c^+	c^+	c^+	c^+
0,80	c^+	c^+	c^+	c^-
0,78	c^+	c^+	c^-	c^+
0,70	c^+	c^+	c^-	c^-
0,60	c^+	c^+	c^-	c^+
0,45	c^+	c^-	c^-	c^-
0,40	c^+	c^-	c^-	c^-
0,30	c^+	c^-	c^-	c^-
0,20	c^+	c^-	c^-	c^+
0,15	c^-	c^-	c^-	c^-
0,10	c^-	c^-	c^-	c^-
0,05	c^-	c^-	c^-	c^-

Receiver Operating Characteristics (ROC) curve

ROC curve

For a probabilistic/fuzzy classifier, a ROC curve is a plot of the TPR vs. FPR as its discrimination threshold is varied



$p(c\|\boldsymbol{x})$	$T = 0{,}2$	$T = 0{,}5$	$T = 0{,}8$	C
0,99	c^+	c^+	c^+	c^+
0,90	c^+	c^+	c^+	c^+
0,85	c^+	c^+	c^+	c^+
0,80	c^+	c^+	c^+	c^-
0,78	c^+	c^+	c^-	c^+
0,70	c^+	c^+	c^-	c^-
0,60	c^+	c^+	c^-	c^+
0,45	c^+	c^-	c^-	c^-
0,40	c^+	c^-	c^-	c^-
0,30	c^+	c^-	c^-	c^-
0,20	c^+	c^-	c^-	c^+
0,15	c^-	c^-	c^-	c^-
0,10	c^-	c^-	c^-	c^-
0,05	c^-	c^-	c^-	c^-

Receiver Operating Characteristics (ROC) curve

ROC curve

- Insensitive to skew class distribution
- Insensitive to misclassification cost

ROC construction

- We need a classifier that produces posterior probability for each test instance $P(+|A)$
- Sort the instances according to $P(+|A)$ in decreasing order
- Apply threshold at each unique value of $P(+|A)$
- Count the number of TP, FP, TN, FN at each threshold
- Calculate TPR = TP/(TP+FN)
- Calculate FPR = FP/(FP+TN)
- Plot the results

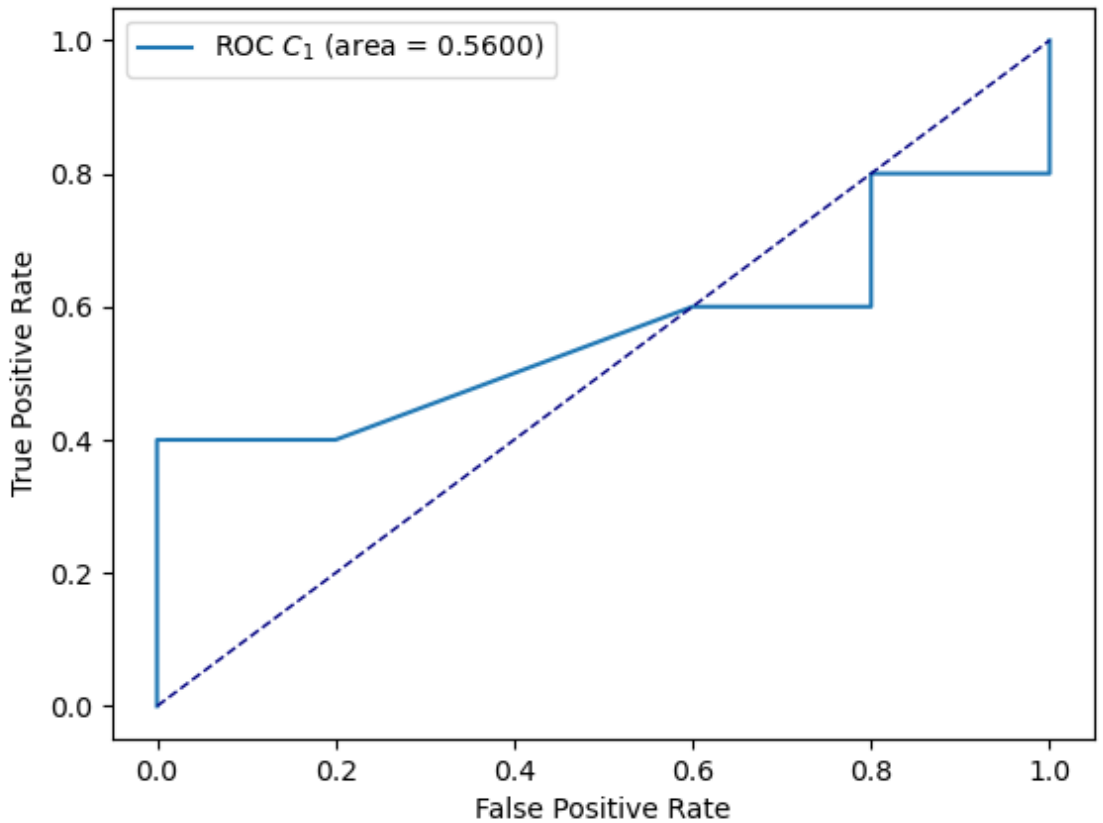
Instance	$P(+\|A)$	Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

ROC construction

Obtaining TPR and FPR

Instance	10	9	8	7	4	5	6	3	2	1	
Class	+	-	+	-	-	-	+	-	+	+	
Threshold ($\geqslant$ is +)	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1.0	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0.0
FPR	1.0	1.0	0.8	0.8	0.6	0.4	0.2	0.2	0.0	0.0	0.0

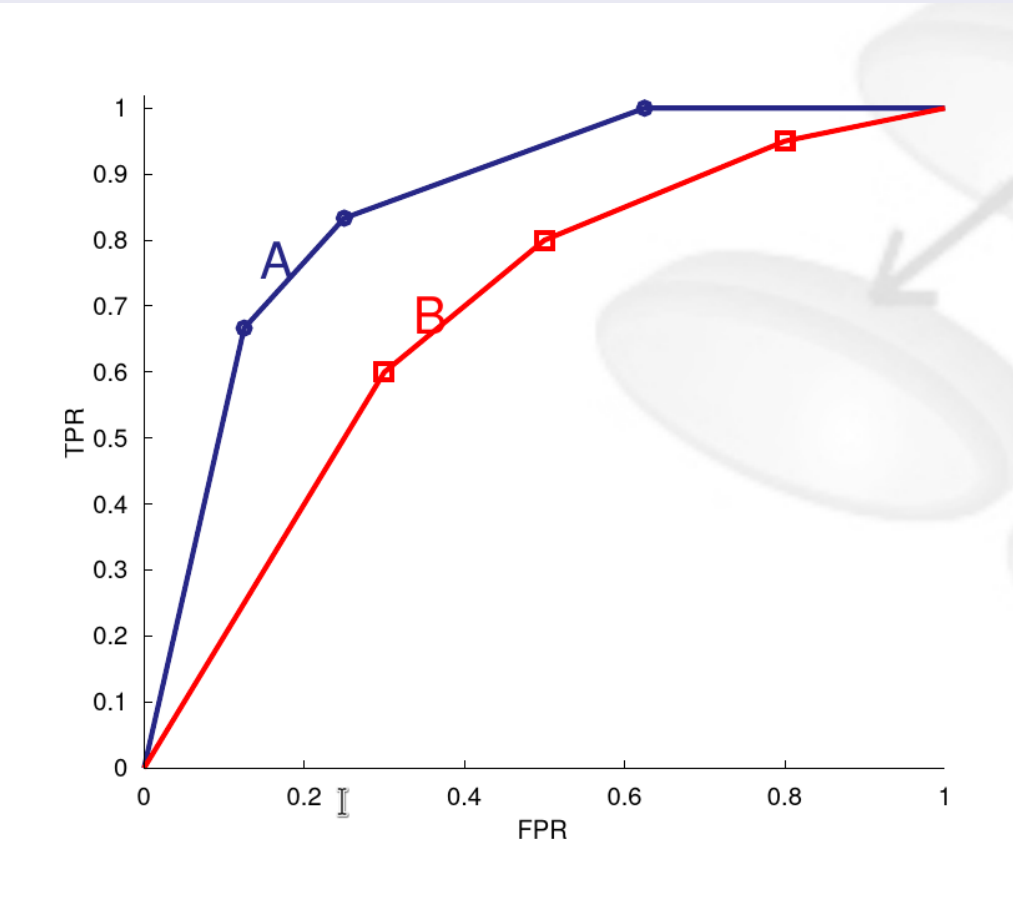
ROC construction



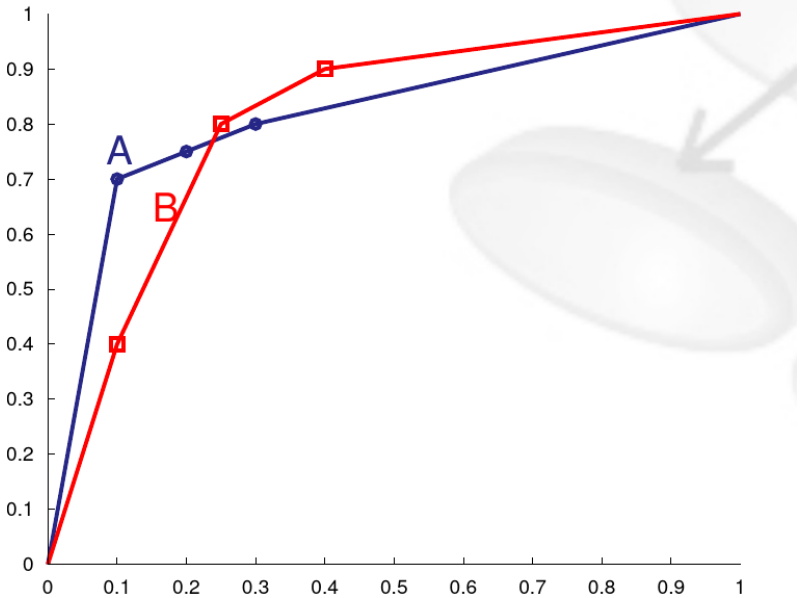
Receiver Operating Characteristics (ROC) curve

Dominance Relationship

A ROC curve A dominates another ROC curve B if A is always above and to the left of B in the plot

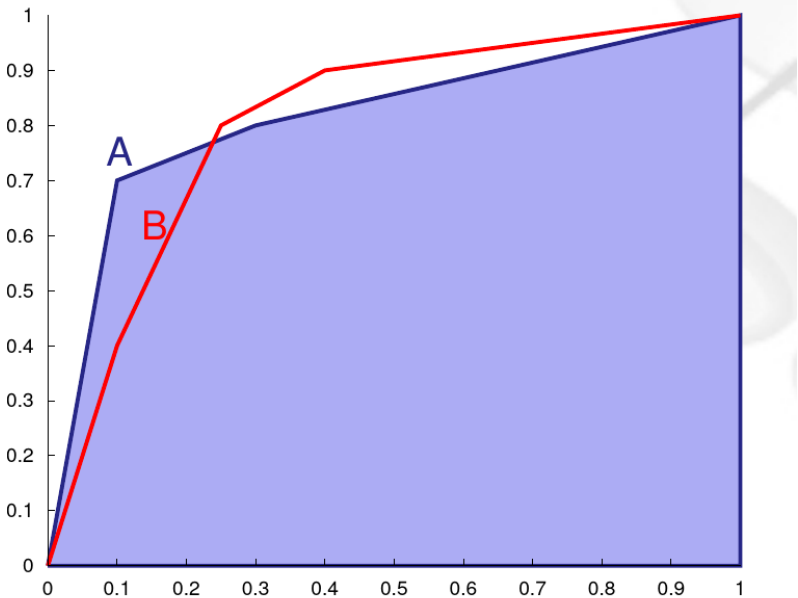


Receiver Operating Characteristics (ROC) curve



- The dominance relationship may not be so clear
- No model is the best under all possible scenarios

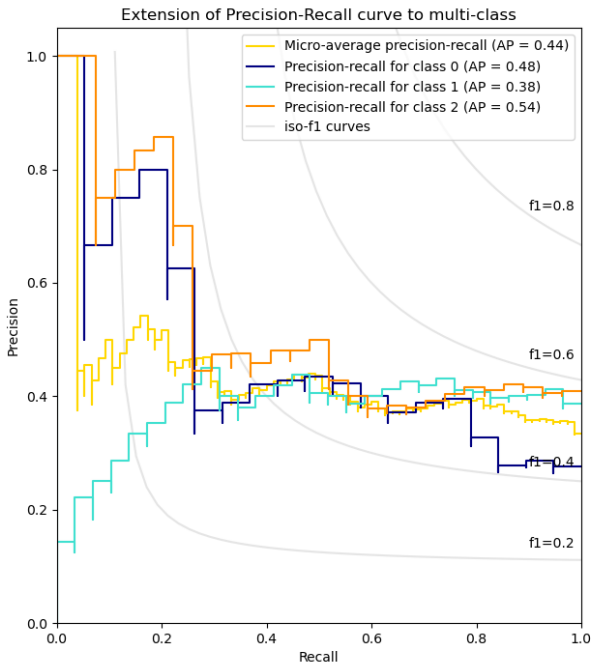
Area under the ROC curve (auROC)



- Equivalent to Wilcoxon test
- If A dominates B: auROC(A) $\geqslant$ auROC(B)
- If A does not dominate B auROC "cannot identify the best classifier"

Precision recall curve (PRC)

X axis: Recall, Y axis: Precision



Area under the precision recall curve (auPRC) can be obtained, but it is more difficult to calculate.

Generalization to Multilabel-Class

- Most of the presented scores are for binary classification
- Generalization to multilabel is possible
  - ✓ E.g. One-vs-All approach

Generalization to Multilabel-Class: One-vs-all

		Prediction					
		c_1	c_2	c_3	$\cdots$	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	$\cdots$	FN_{1n}	N_1
	c_2	FN_{21}	TP_2	FN_{23}	$\cdots$	FN_{2n}	N_2
	c_3	FN_{31}	FN_{32}	TP_3	$\cdots$	FN_{3n}	N_3
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	$\cdots$	TP_n	N_n
	Total	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_2$	$\cdots$	$\hat{N}_n$	N

c_1 vs. all

- TP
- TN
- FN
- FP

Generalization to Multilabel-Class: One-vs-all

		Prediction					
		c_1	c_2	c_3	$\cdots$	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	$\cdots$	FN_{1n}	N_1
	c_2	FN_{21}	TP_2	FN_{23}	$\cdots$	FN_{2n}	N_2
	c_3	FN_{31}	FN_{32}	TP_3	$\cdots$	FN_{3n}	N_3
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	$\cdots$	TP_n	N_n
	Total	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_2$	$\cdots$	$\hat{N}_n$	N

c_1 vs. all

- TP
- TN
- FN
- FP

Generalization to Multilabel-Class: One-vs-all

		Prediction					
		c_1	c_2	c_3	$\cdots$	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	$\cdots$	FN_{1n}	N_1
	c_2	FN_{21}	TP_2	FN_{23}	$\cdots$	FN_{2n}	N_2
	c_3	FN_{31}	FN_{32}	TP_3	$\cdots$	FN_{3n}	N_3
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	$\cdots$	TP_n	N_n
	Total	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_2$	$\cdots$	$\hat{N}_n$	N

c_1 vs. all

- TP
- TN
- FN
- FP

Generalization to Multilabel-Class: One-vs-all

		Prediction					
		c_1	c_2	c_3	$\cdots$	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	$\cdots$	FN_{1n}	N_1
	c_2	FN_{21}	TP_2	FN_{23}	$\cdots$	FN_{2n}	N_2
	c_3	FN_{31}	FN_{32}	TP_3	$\cdots$	FN_{3n}	N_3
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	$\cdots$	TP_n	N_n
	Total	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_2$	$\cdots$	$\hat{N}_n$	N

c_1 vs. all

- TP
- TN
- FN
- FP

Generalization to Multilabel-Class: One-vs-all

		Prediction					Total
		c_1	c_2	c_3	$\cdots$	c_n	
Actual	c_1	TP_1	FN_{12}	FN_{13}	$\cdots$	FN_{1n}	N_1
	c_2	$\mathbf{FN_{21}}$	TP_2	FN_{23}	$\cdots$	FN_{2n}	N_2
	c_3	$\mathbf{FN_{31}}$	FN_{32}	TP_3	$\cdots$	FN_{3n}	N_3
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	c_n	$\mathbf{FN_{n1}}$	FN_{n2}	FN_{n3}	$\cdots$	TP_n	N_n
	Total	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_2$	$\cdots$	$\hat{N}_n$	N

c_1 vs. all

- TP
- TN
- FN
- FP

Combining scores: Micro and macro averaging

- **Micro**-average
  - ✓ Obtain the values of true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs) of the model as the sum for all classes
  - ✓ Then calculate the desired metric
- **Macro**-average
  - ✓ The metric is calculated as the arithmetic mean of individual classes metric

When to use micro-averaging and macro-averaging scores?

- Use micro-averaging score when there is a need to weight each instance or prediction equally
- Use macro-averaging score when all classes need to be treated equally to evaluate the overall performance of the classifier with regard to the most frequent class labels
- Use weighted macro-averaging score in case of class imbalances (different number of instances related to different class labels). The weighted macro-average is calculated by weighting the score of each class label by the number of true instances when calculating the average

Example: Micro-Average & Macro-Average Precision Scores

- For multi-class classification problem, micro-average precision scores can be defined as sum of true positives for all the classes divided by the all positive predictions. The positive prediction is sum of all true positives and false positives:

$$P_{\text{micro}} = \frac{TP}{TP + FP} = \frac{\sum_i^n TP_i}{\sum_i^n TP_i + \sum_i^n FP_i}$$

- Macro-average precision score can be defined as the arithmetic mean of all the precision scores of different classes:

$$P_{\text{macro}} = \frac{\sum_i^n P_i}{n}$$

- Weighted macro-average precision score can be defined as the weighted mean of all the precision scores of different classes:

$$P_{\text{macro}} = \sum_i^n p(c_i) P_i$$

Other multiclass metrics

Cohen's κ

- The value of κ can be computed from the confusion matrix in a classification task as follows:

$$\kappa = \frac{n \sum_{i=1}^{C} x_{ii} - \sum_{i=1}^{C} x_{i\cdot} x_{\cdot i}}{n^2 - \sum_{i=1}^{C} x_{i\cdot} x_{\cdot i}},$$

  where x_{ii} is the cell count on the main diagonal, n is the number of examples, C is the number of classes, and $x_{i\cdot}$ and $x_{\cdot i}$ are the column and row total counts, respectively.

- The value of κ ranges from -1 (total disagreement) to 1 (perfect agreement).

- For multi-class problems, κ is a very useful yet simple metric for measuring the accuracy of the classifier while compensating for random successes.

Other multiclass metrics

Multi-class Matthews Correlation Coefficient (Jurman, Riccadonna, and Furlanello, 2012)

Being X, Y two matrices where X_{sn} is 1 if the sample is predicted to be of class n and o otherwise, and $Y_{sn} = 1$ if sample s belongs to class n and 0 otherwise, Matthews Correlation Coefficient (MCC) is defined as:

$$MCC = \frac{cov(X, Y)}{\sqrt{cov(X, X) \cdot cov(Y, Y)}}$$

MCC $\in [-1, 1]$, with 1 being perfect classification.

Other multiclass metrics

Confusion Entropy (CEN) (Wei et al., 2010)

- Classification evaluation metric based on Shannon's Entropy based on the confusion matrix

- Let C be a $N \times N$ confusion matrix. $C_{i,j}$ denotes the number instances of class i classified as class j. The probability for an element of class i to be classified as class j, subjected to class i, $P^i_{i,j}$, is defined as:

$$P^i_{i,j} = \frac{C_{i,j}}{\sum_{k=1}^{N}(C_{i,k} + C_{k,i})}, \quad i,j = 1,\ldots,N, i \neq j$$

The probability for an element of class i to be classified as class j subjected to class j, $P^j_{i,j}$, is defined as:

$$P^j_{i,j} = \frac{C_{i,j}}{\sum_{k=1}^{N}(C_{j,k} + C_{k,j}}), \quad i,j = 1,\ldots,N, i \neq j$$

Other multiclass metrics

Confusion Entropy (CEN)

- *CEN* is defined as:

$$CEN = \sum_{j=1}^{N} P_j CEN_j$$

$$CEN_j = -\sum_{k=1, k\neq j}^{N} \left(P^j_{j,k} \log_{2(N-1)}(P^j_{j,k}) + P^j_{k,j} \log_{2(N-1)}(P^j_{k,j}) \right)$$

$$P_j = \frac{\sum_{k=1}^{N}(C_{j,k} + C_{k,j})}{2\sum_{k,l=1}^{N} C_{k,l}}$$

- $CEN = 0$ the best classification (all values in the main diagonal of the confusion matrix)

- $CEN = 1$ the worst possible misclassification (all values out of the main diagonal of the confusion matrix and uniformly distributed)

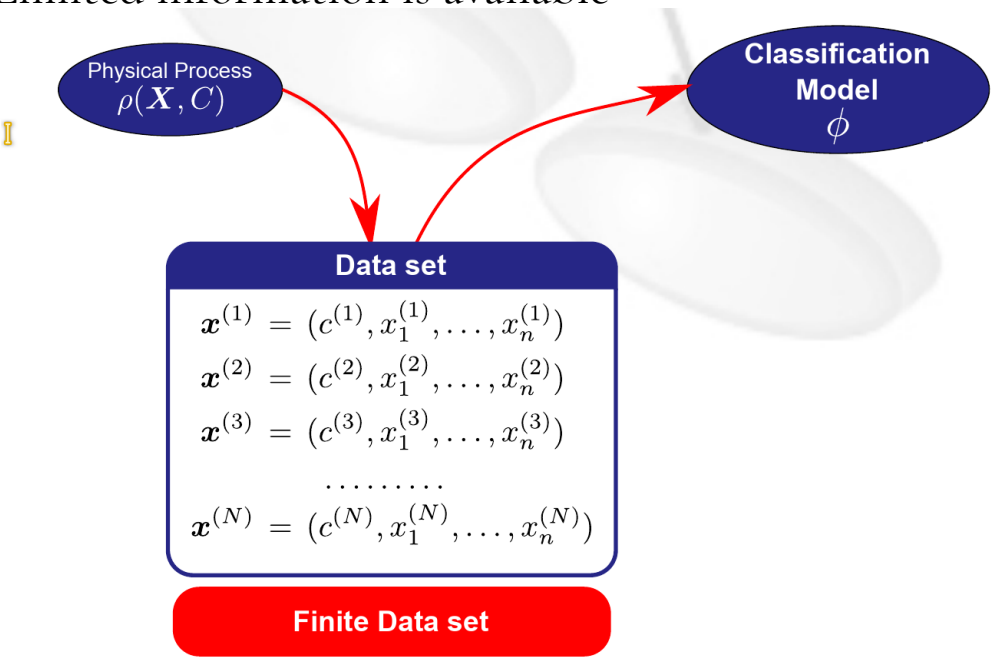
Metric selection

- The Use of a Specific Score Depends on
  - ✓ Application domain
  - ✓ Characteristics of the problem
  - ✓ Characteristics of the data set
  - ✓ Our interest when solving the problem
  - ✓ Other considerations

Estimation methods

Estimation

- Select a score to measure the quality
- Calculate the true value of the score
- Limited information is available

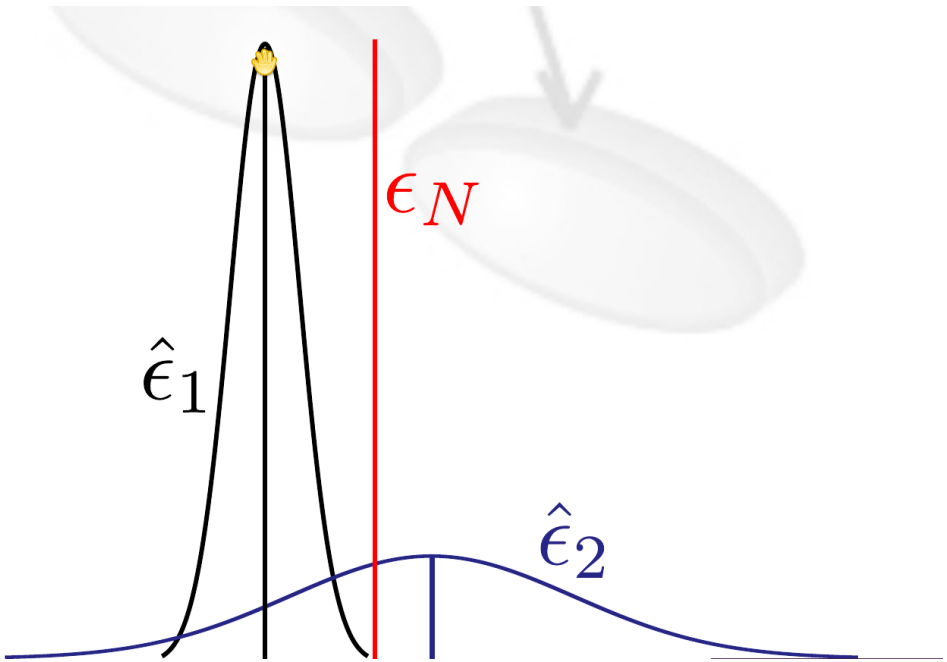


- Quality measures (metrics) are random variables

Estimation

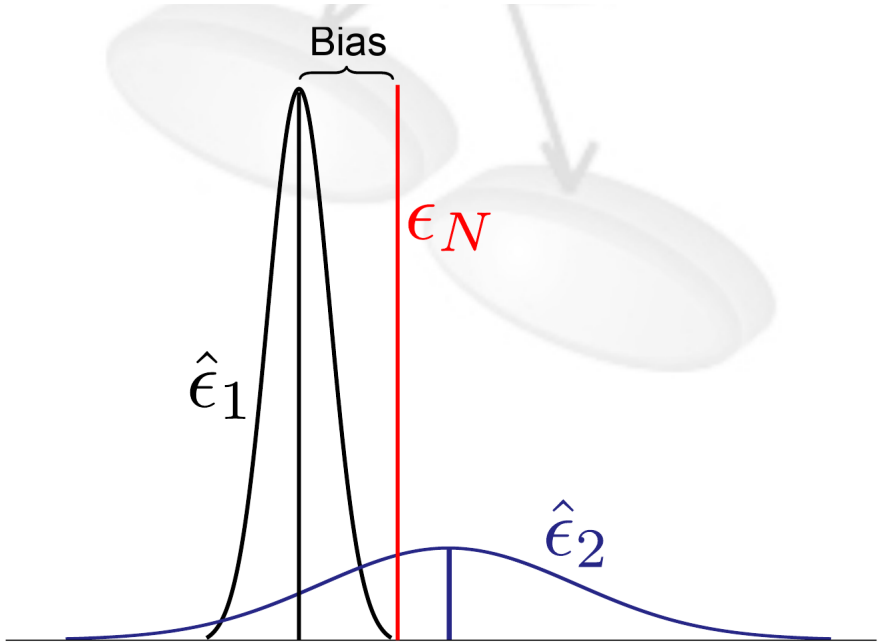
True Value: ϵ_N

- Expected value of the score for a set of N data samples sampled from $\rho(C, X)$
- $\rho(C, X)$ unknown $\longrightarrow$ Point estimation of the score $(\hat{\epsilon})$



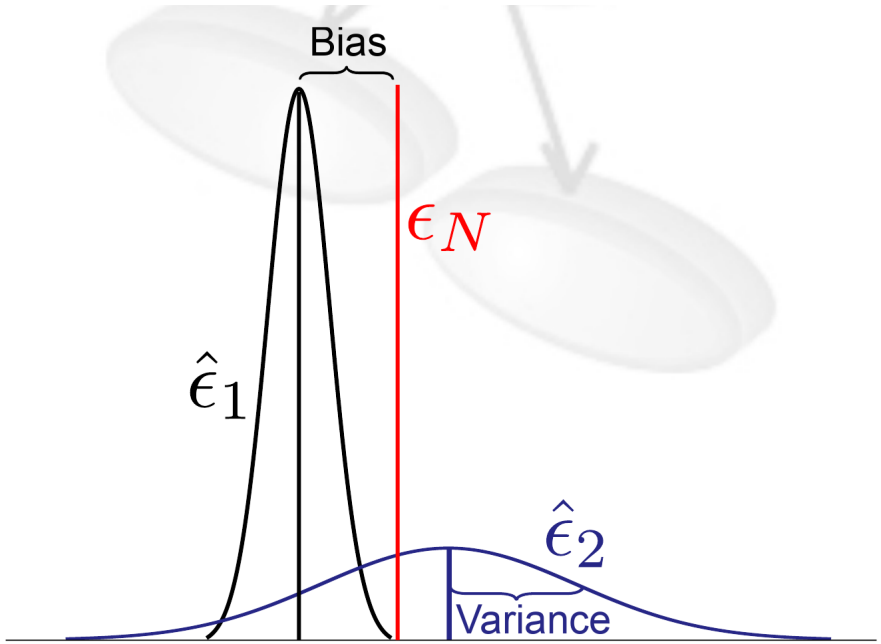
Estimation: Bias

Difference between the estimation of the score and its true value: $E_\rho(\hat{\epsilon}-\epsilon_N)$



Estimation: Variance

Deviation of the estimated value from its expected value: $var(\hat{\epsilon}-\epsilon_N)$



Bias anv variance

- Bias and variance depend on the estimation method
- Trade-off between bias and variance needed

Estimation: Finite dataset

Data set

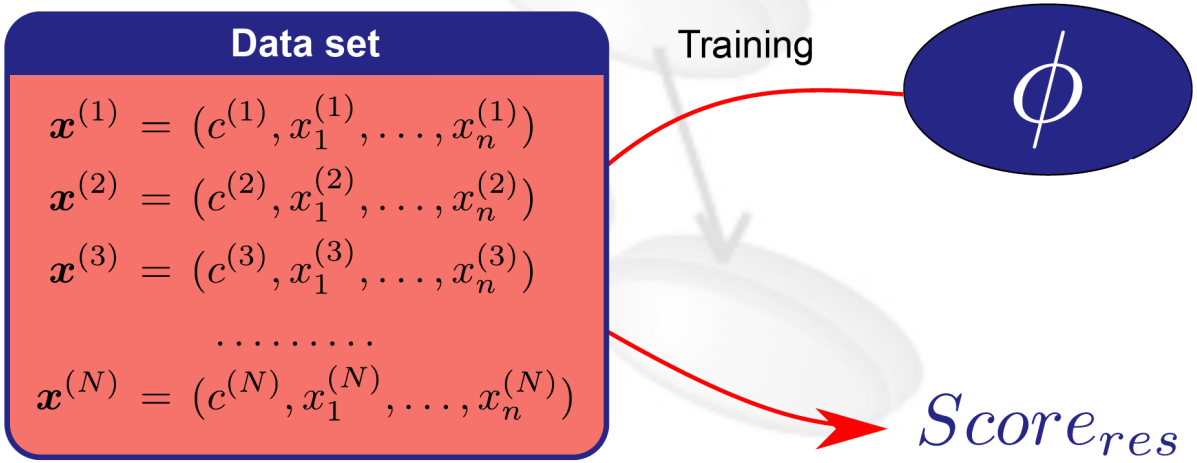
$$\begin{aligned}
\boldsymbol{x}^{(1)} &= (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)}) \\
\boldsymbol{x}^{(2)} &= (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)}) \\
\boldsymbol{x}^{(3)} &= (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)}) \\
&\ldots\ldots\ldots \\
\boldsymbol{x}^{(N)} &= (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})
\end{aligned}$$

- Finite data set to estimate the score
- Several choices depending on how this data set is dealt with

Estimation methods

- Resubstitution
- Hold-out
- k-Fold cross-validation
- Bootstrap
- Leave-one-out bootstrap

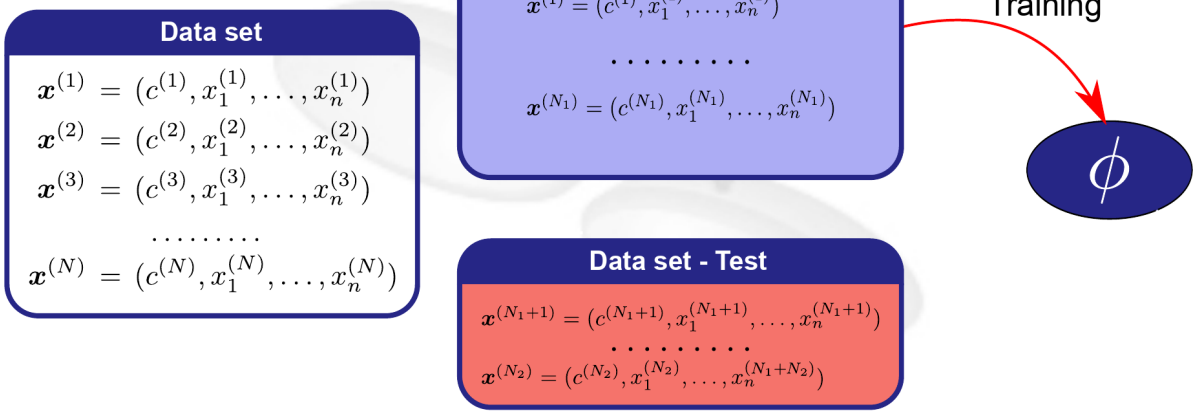
Resubstitution



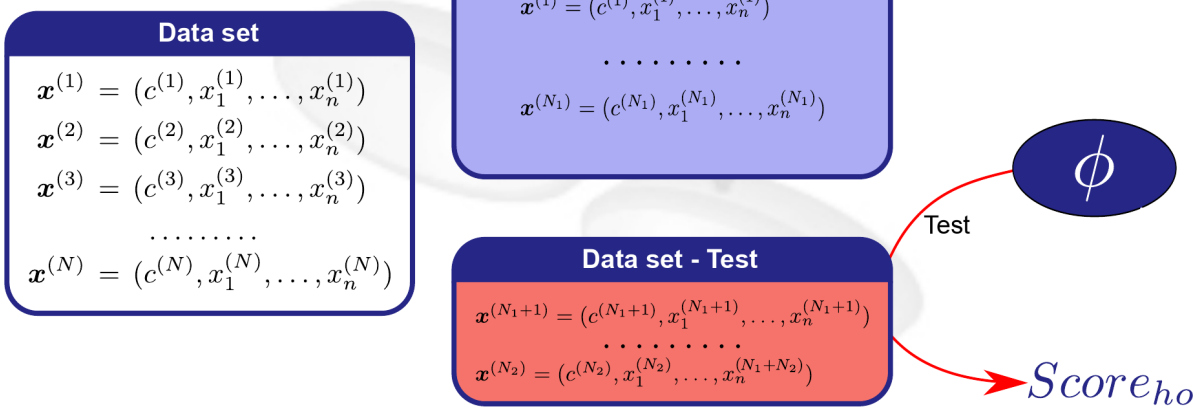
Resubstitution

- The simplest estimation method
- Biased estimation ϵ_N
- Smaller variance
- Too optimistic (overfitting problem)
- Bad estimator of the true classification error

Hold-out


Data set
$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$
$\boldsymbol{x}^{(2)} = (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)})$
$\boldsymbol{x}^{(3)} = (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)})$
. . . . . .
$\boldsymbol{x}^{(N)} = (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})$
Data set - Training
$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$
. . . . . .
$\boldsymbol{x}^{(N_1)} = (c^{(N_1)}, x_1^{(N_1)}, \ldots, x_n^{(N_1)})$
Training
ϕ
Data set - Test
$\boldsymbol{x}^{(N_1+1)} = (c^{(N_1+1)}, x_1^{(N_1+1)}, \ldots, x_n^{(N_1+1)})$
. . . . . .
$\boldsymbol{x}^{(N_2)} = (c^{(N_2)}, x_1^{(N_2)}, \ldots, x_n^{(N_1+N_2)})$

Hold-out

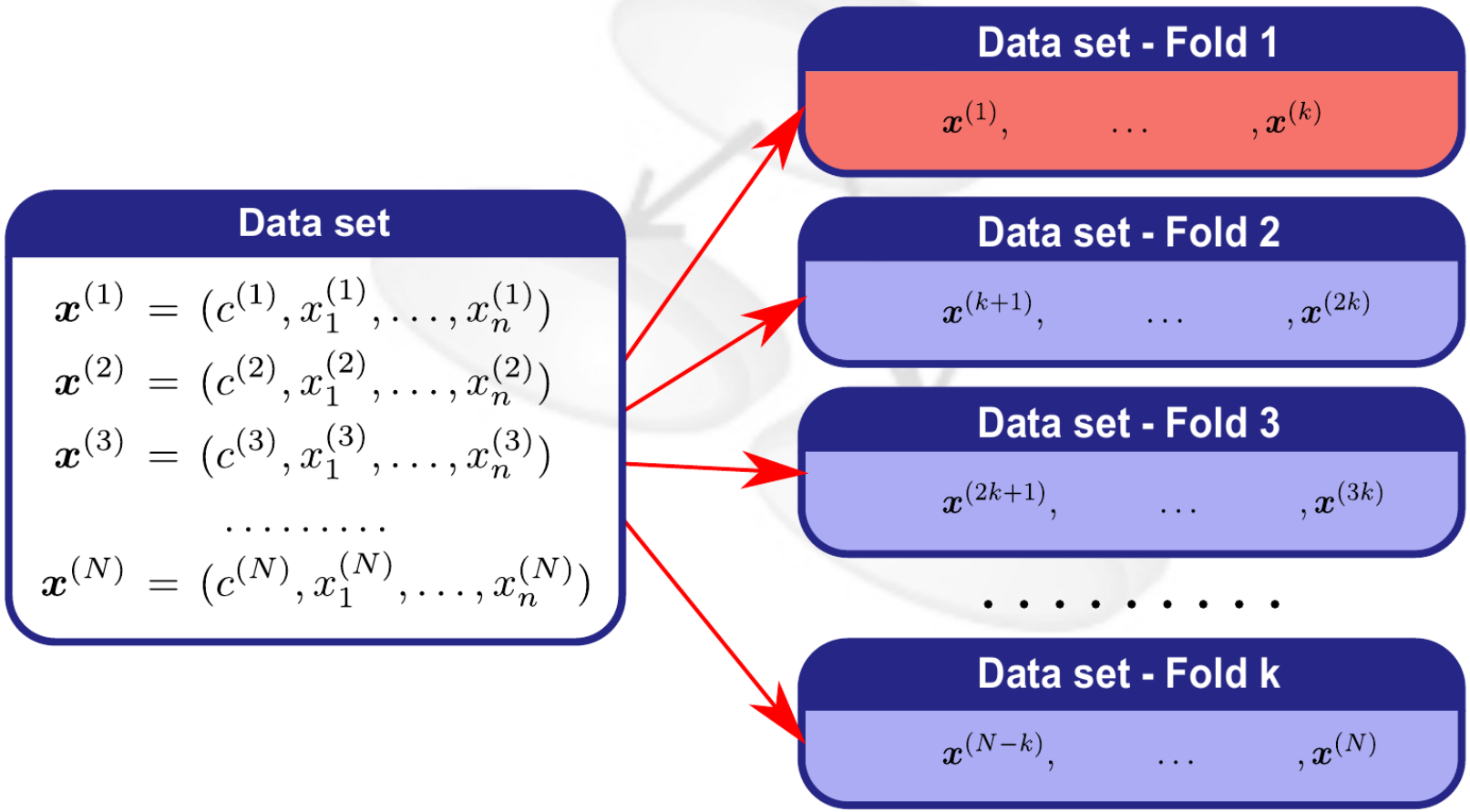

Data set
$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$
$\boldsymbol{x}^{(2)} = (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)})$
$\boldsymbol{x}^{(3)} = (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)})$
$\boldsymbol{x}^{(N)} = (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})$
Data set - Training
$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$
$\boldsymbol{x}^{(N_1)} = (c^{(N_1)}, x_1^{(N_1)}, \ldots, x_n^{(N_1)})$
Data set - Test
$\boldsymbol{x}^{(N_1+1)} = (c^{(N_1+1)}, x_1^{(N_1+1)}, \ldots, x_n^{(N_1+1)})$
$\boldsymbol{x}^{(N_2)} = (c^{(N_2)}, x_1^{(N_2)}, \ldots, x_n^{(N_1+N_2)})$
ϕ
Test
$Score_{ho}$

Hold-out

- Unbiased estimator of ϵ_{N1}
- Biased estimator of ϵ_N
- Large bias (pessimistic estimation of the true classification error)
- Bias related to N_1 and N_2

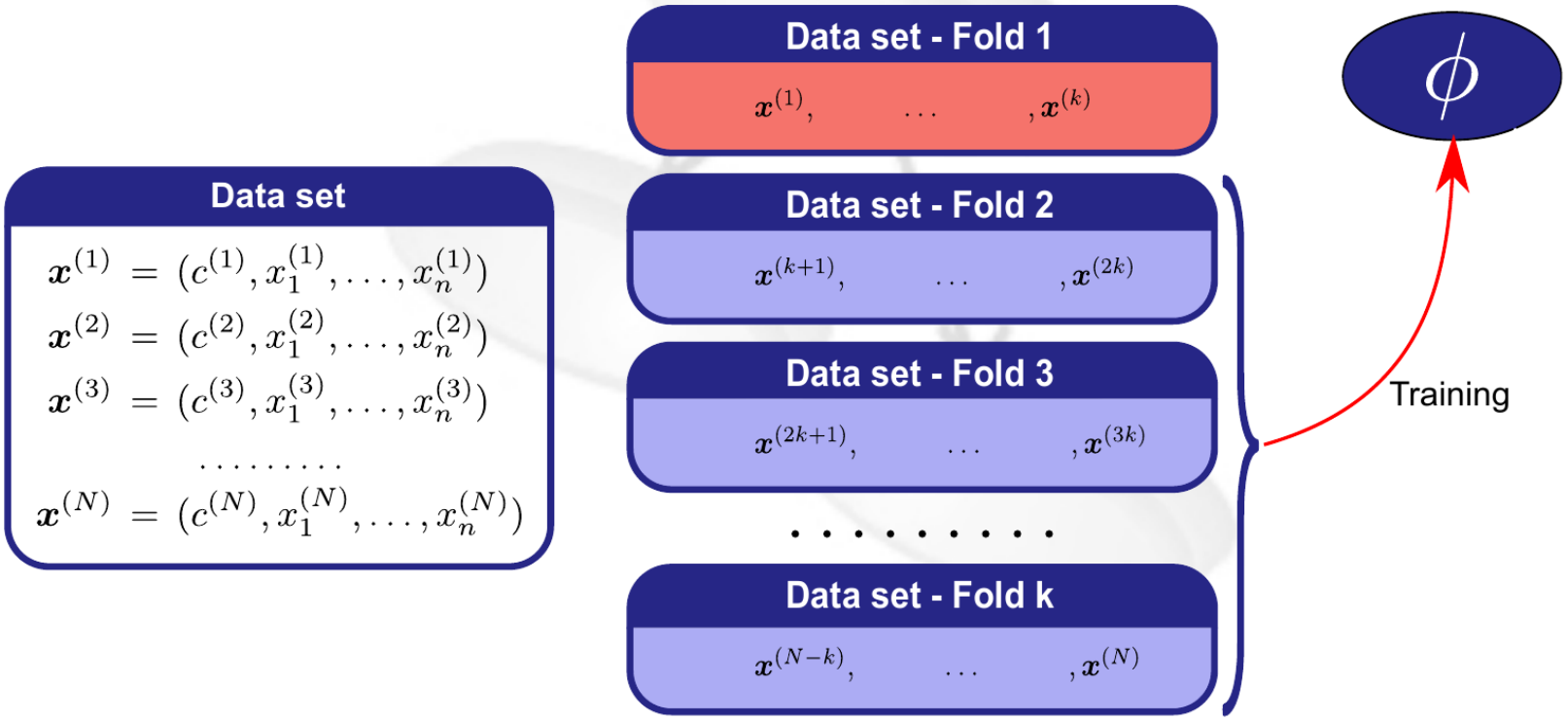
k-Fold Cross-Validation

Dataset is divided into k folds



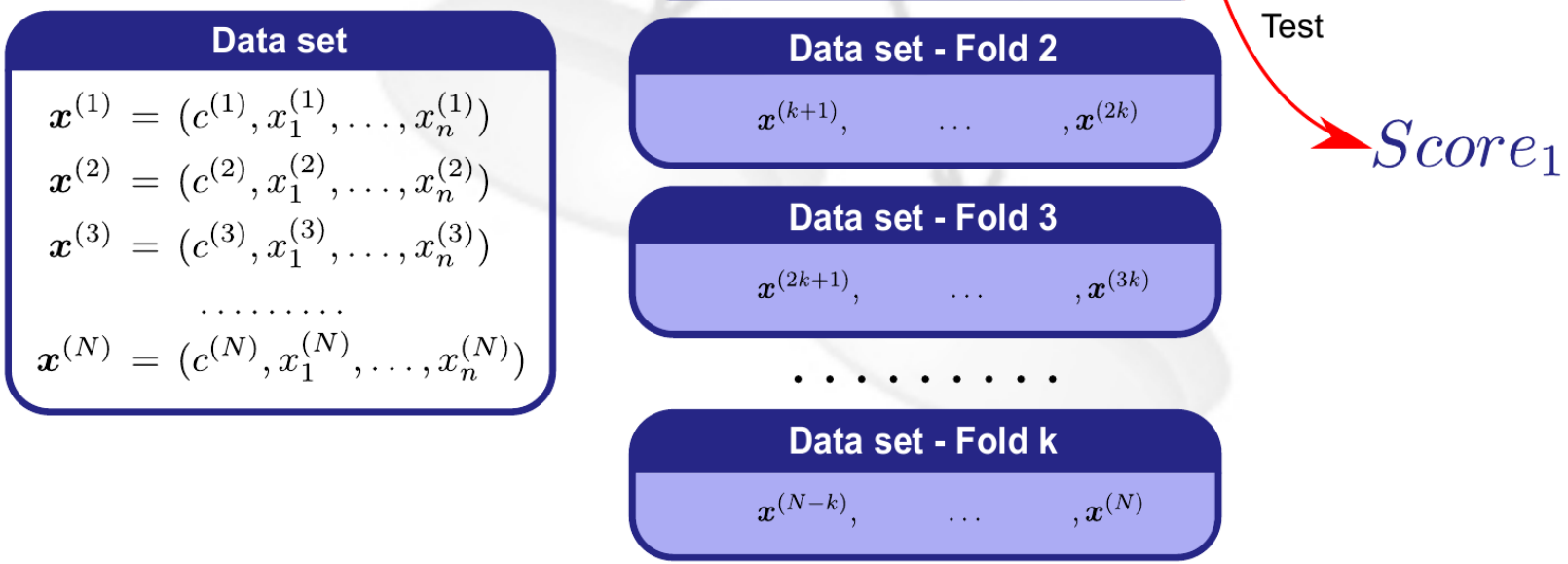
k-Fold Cross-Validation

Train with folds 1 to k



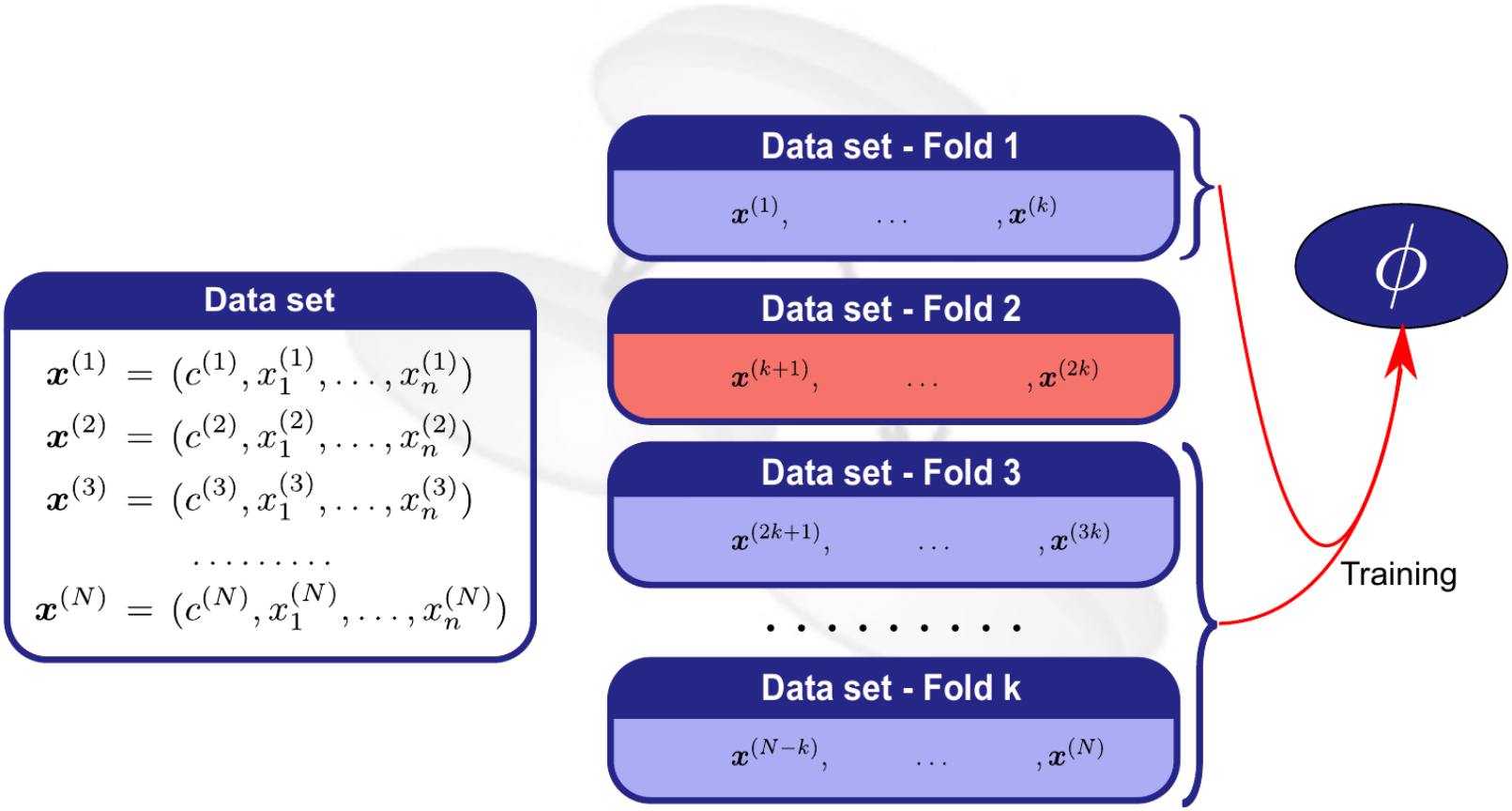
k-Fold Cross-Validation

Test with fold 1



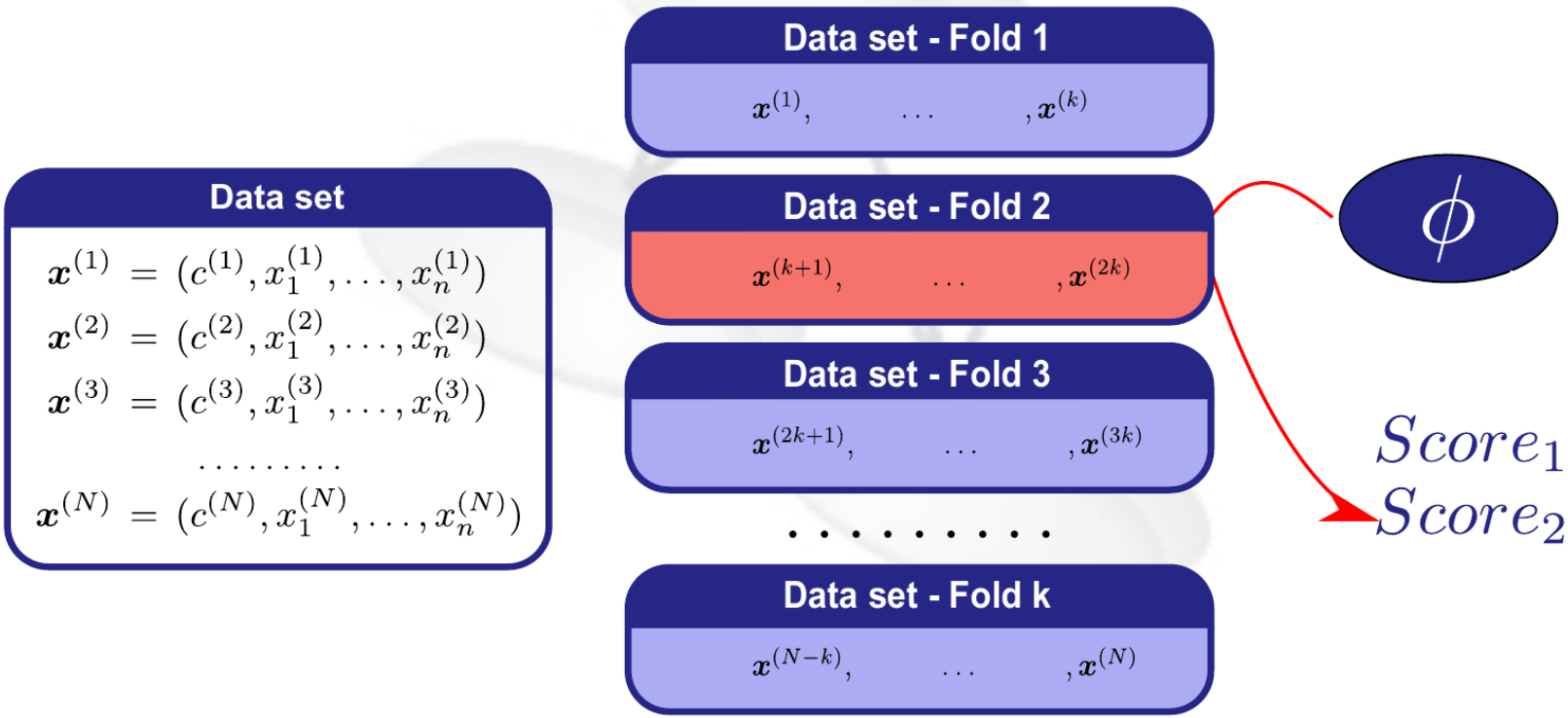
k-Fold Cross-Validation

Train with folds 1, 3 to k



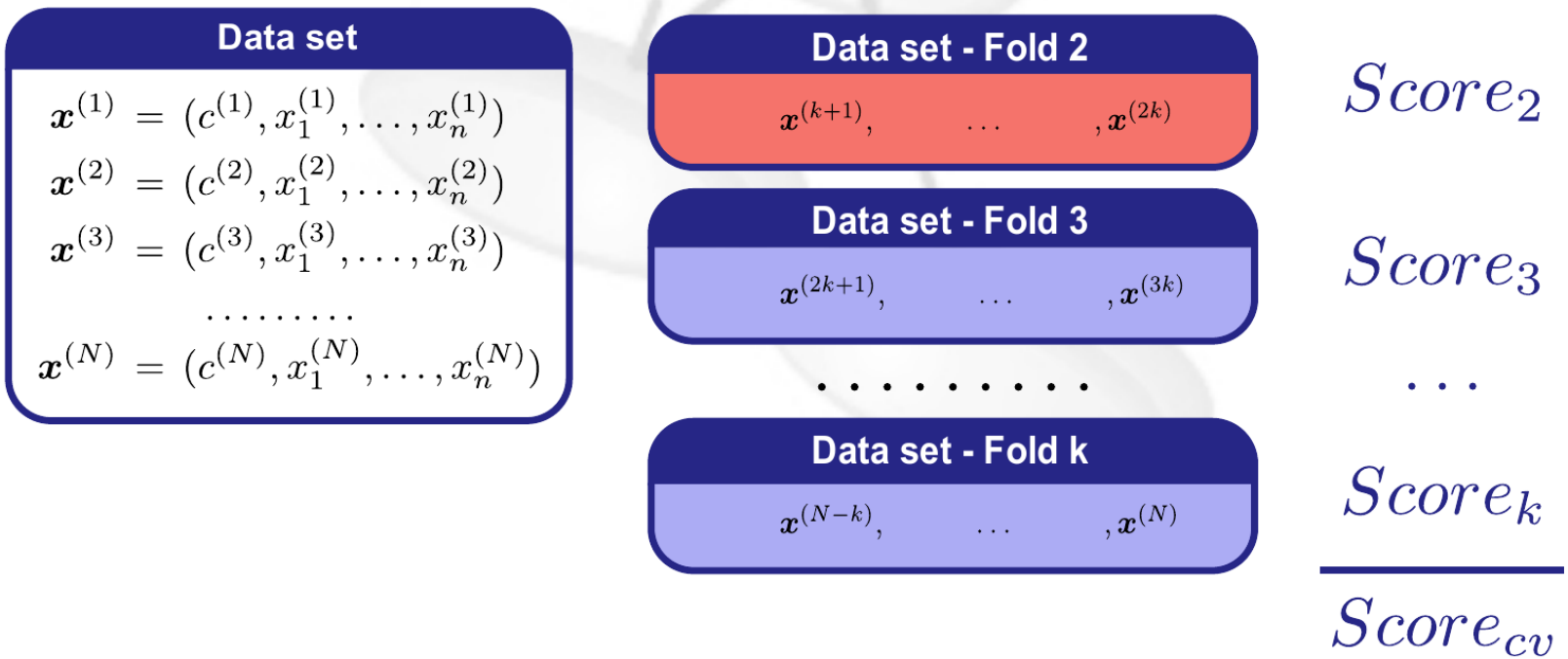
k-Fold Cross-Validation

Test with fold 2



k-Fold Cross-Validation

Score is the average score



k-Fold Cross-Validation

Classification Error Estimation

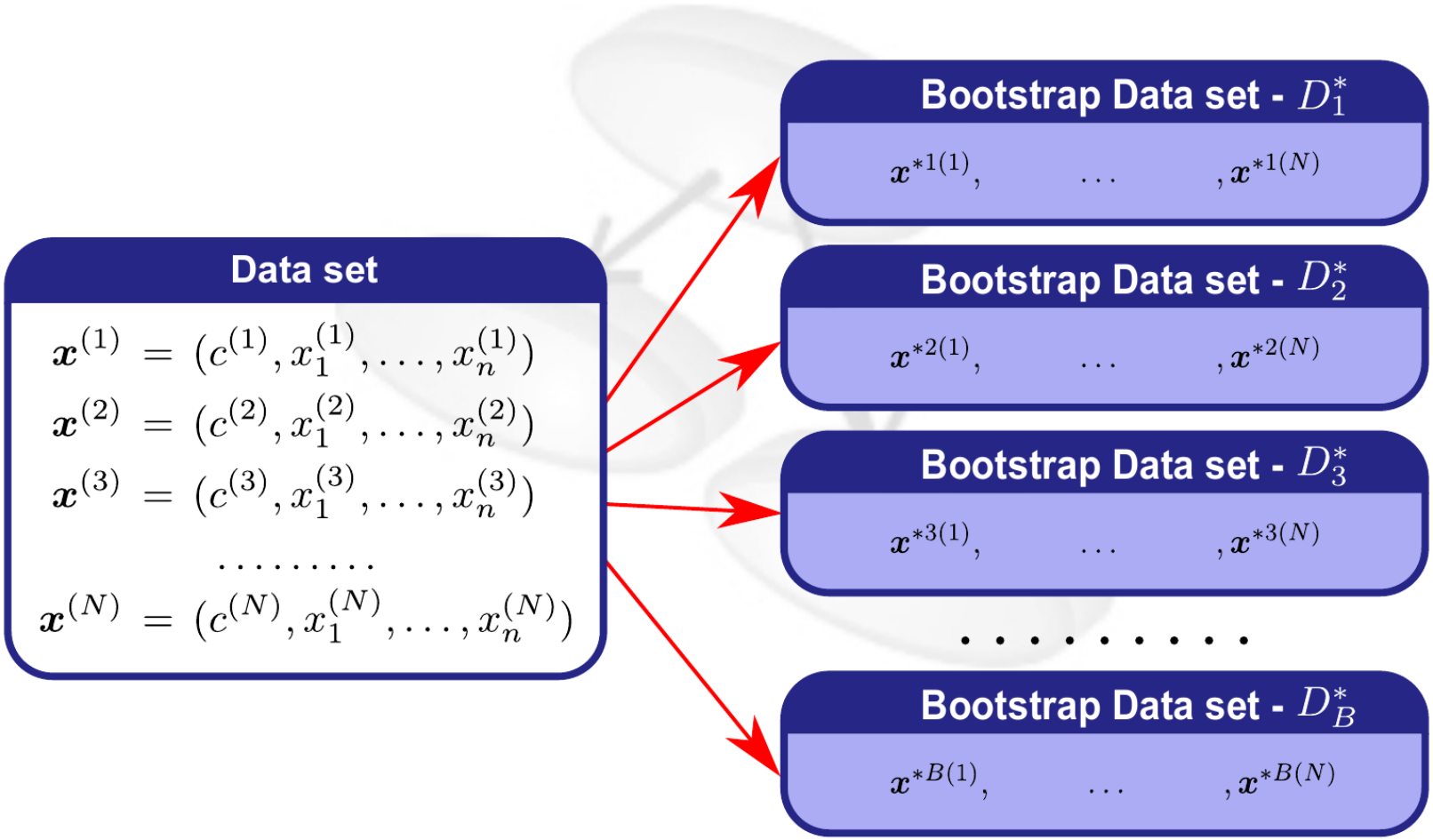
- Unbiased estimator of $\epsilon_{N-\frac{N}{k}}$
- Biased estimation of ϵ_N
- Smaller bias than Hold-Out

Leaving-One-Out

- Special case of k-fold Cross-Validation ($k = N$)
- Quasi unbiased estimation for N
- Improves the bias with respect to CV
- Increases the variance $\longrightarrow$ more unstable
- Higher computational cost

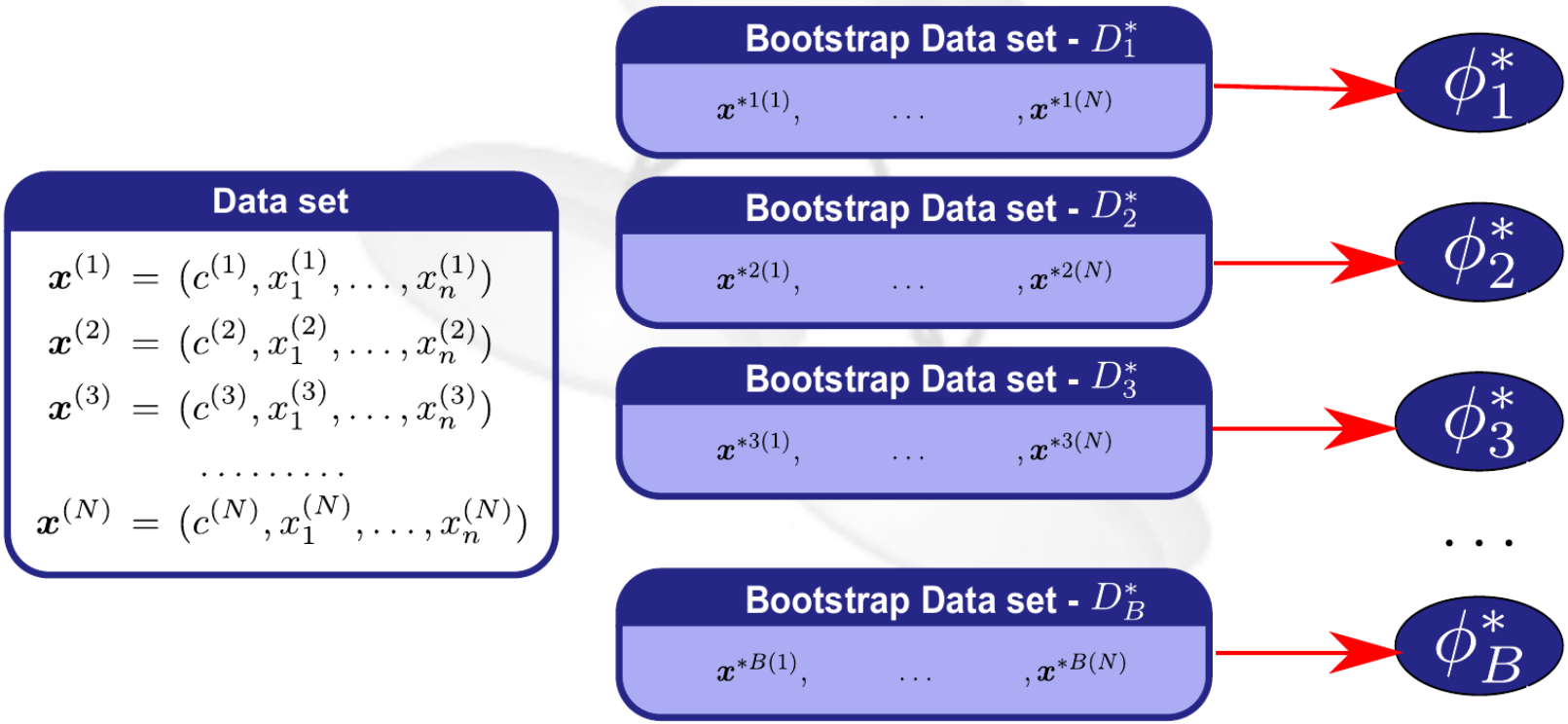
Bootstrap

Bootstrap sample: Sample with replacement



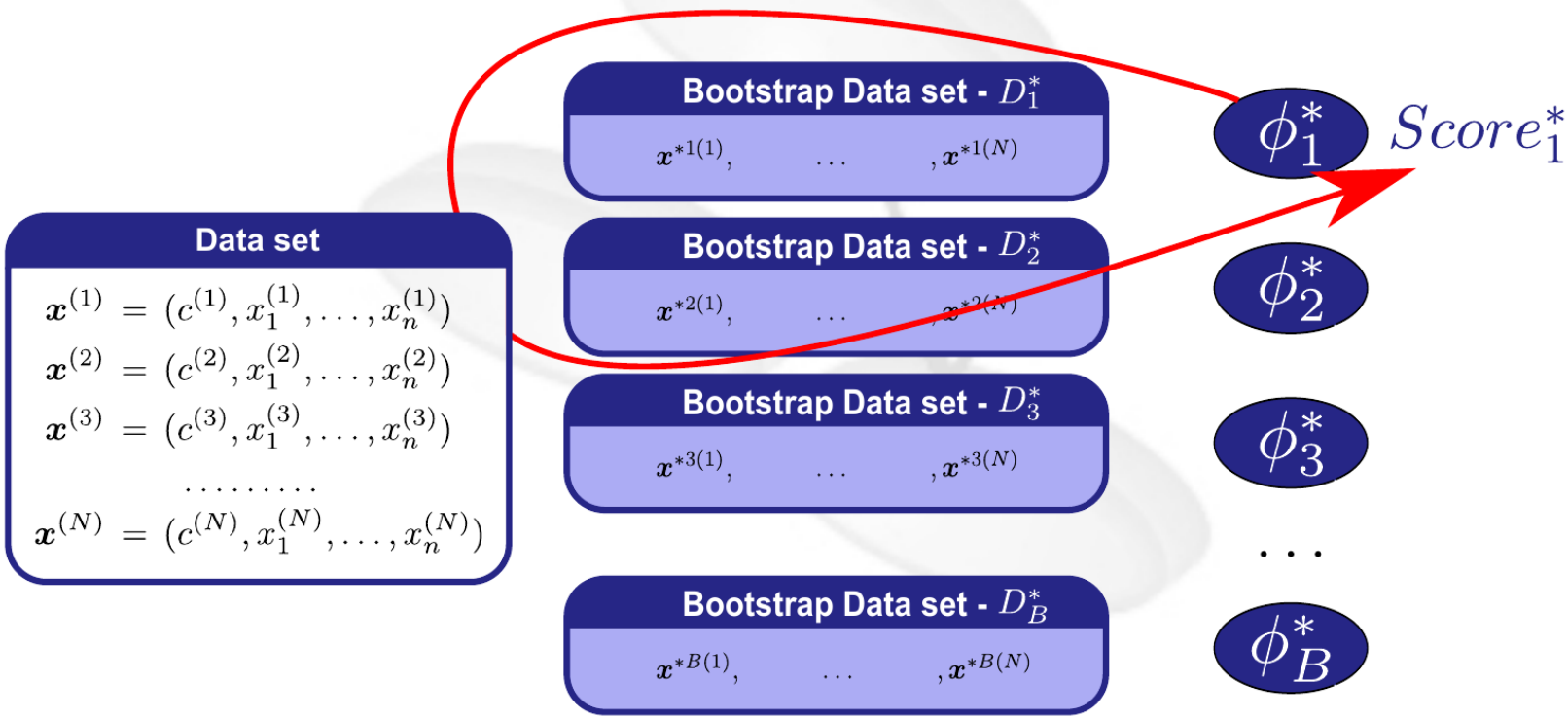
Bootstrap

Train with each bootstrap sample



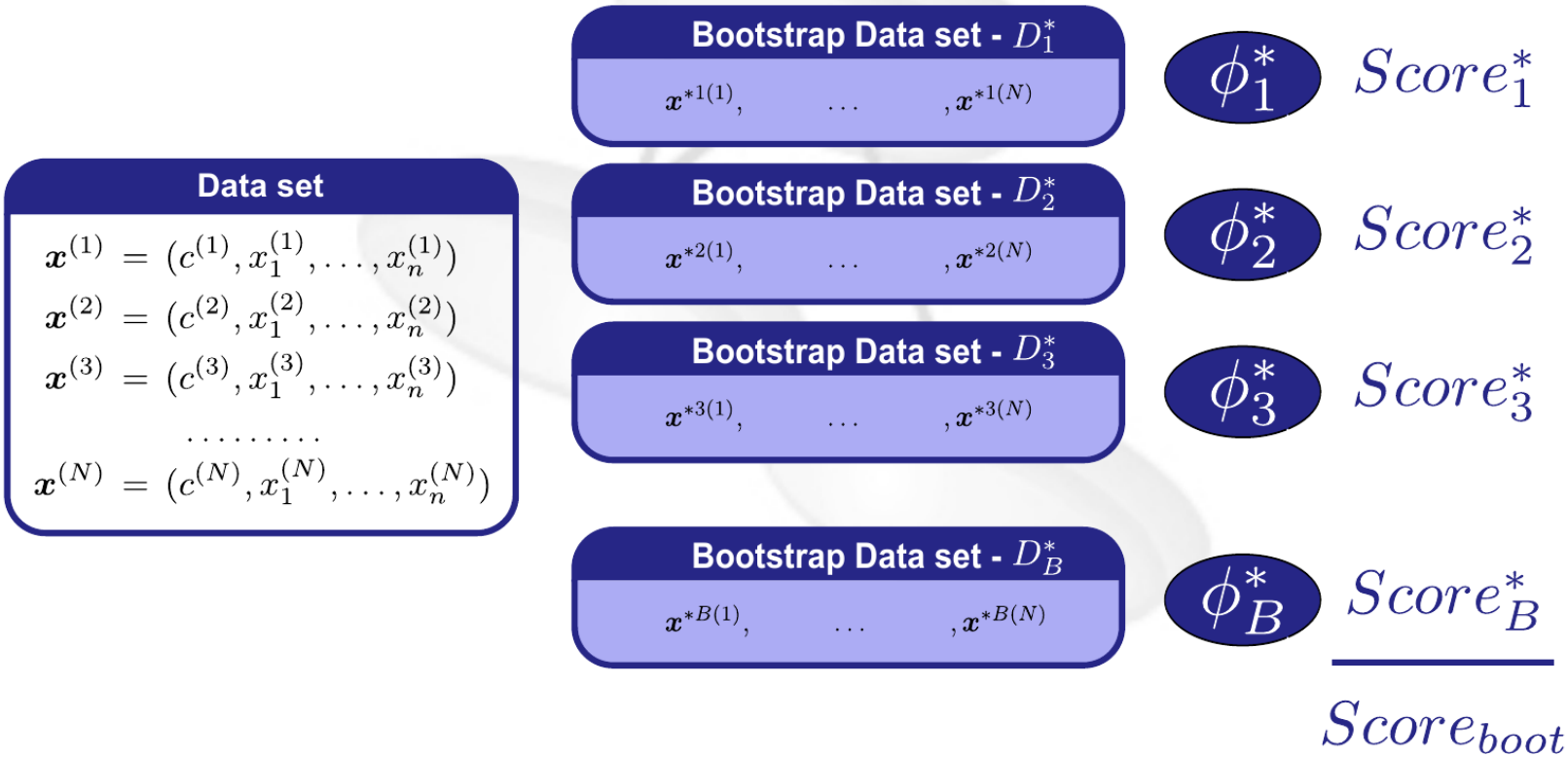
Bootstrap

Test with the original dataset (not resubstitution)



Bootstrap

Score is the average score



Bootstrap classification error estimation

- Biased estimation of the classification error
- Variance improved because of resampling
- Uses for testing part of the data used for learning
- *Similar* to resubstitution
- Problem of overfitting

Leaving-One-Out Bootstrap

- Mimics Cross-Validation
- Each ϕ_i is tested on D/D_i^*

Tries to Avoid the Overfitting Problem

- Expected number of distinct samples on bootstrap data set $\approx 0.632N$
- Similar to repeated Hold-Out
- Biased upwards: Tends to be a pessimistic estimation of the score

Improving the Estimation - Bias

- Bias correction terms can be used for error estimation
- Hold-Out/Cross-Validation
  - ✓ Several proposals
  - ✓ Improves bias estimation
  - ✓ Surprisingly not very extended
- Bootstrap
  - ✓ Improves bias estimation
  - ✓ Well established methods

Improving the Estimation - Bias

Corrected Hold-Out ($\hat{\epsilon}^{+}_{ho}$) (Burman, 1989)

$$\hat{\epsilon}^{+}_{ho} = \hat{\epsilon}_{ho} + \hat{\epsilon}_{res} - \hat{\epsilon}_{ho-N}$$

Where

- $\hat{\epsilon}_{ho}$ = standard Hold-Out estimator
- $\hat{\epsilon}_{res}$ = resubstitution error
- $\hat{\epsilon}_{ho-N}$ = ϕ learned on Hold-Out learning set but tested on D

Corrected Cross-Validation ($\hat{\epsilon}^{+}_{cv}$) (Burman, 1989)

$$\hat{\epsilon}^{+}_{cv} = \hat{\epsilon}_{cv} + \hat{\epsilon}_{res} - \hat{\epsilon}_{cv-N}$$

Improving the Estimation - Bias

0.632 Bootstrap ($\hat{\epsilon}_{boot}^{0.632}$)

$$\hat{\epsilon}_{boot}^{0.632} = 0.368\hat{\epsilon}_{res} + 0.632\hat{\epsilon}_{loo-boot}$$

Improvement

- Tries to balance optimism (resubstitution) and pessimism (loo-bootstrap)
- Works well with "light-fitting" classifiers
- With overfitting classifiers $\hat{\epsilon}_{boot}^{0.632}$ is still too optimistic

Improving the Estimation - Variance

Stratification

- Keeps the proportion of each class in the train/test data
  - ✓ Hold-Out: Stratified splitting
  - ✓ Cross-Validation: Stratified splitting
  - ✓ Bootstrap: Stratified sampling
- May improve the variance of the estimation

Improving the Estimation - Variance

Repeated Methods

- Applicable to Hold-Out and Cross-Validation
- Bootstrap already includes sampling

Repeated Hold-Out/Cross-Validation

- Repeat estimation process t times
- Simple average over results

Repeated methods behavior

- Same bias as standard estimation methods
- Reduces the variance with respect Hold-Out/Cross-Validation

Which estimation method is better?

- May Depend on Many Aspects
  - ✓ The size of the data set
  - ✓ The classification paradigm used
  - ✓ The stability of the learning algorithm
  - ✓ The characteristics of the classification problem
  - ✓ The bias/variance/computational cost trade-off

Which estimation method is better?

- Large Data Sets
  - ✓ Hold-out may be a good choice
    - Computationally not so expensive
    - Larger bias but depends on the data set size
- Smaller Data Sets
  - ✓ Repeated Cross-Validation
  - ✓ Bootstrap 0.632

k-Fold Cross-Validations has become ("arguably") the *de facto* standard

Hypothesis testing

Basic concepts

- Hypothesis testing forms the basis of scientific reasoning in experimental sciences
- Tests are used to set scientific statements
- A hypothesis H_0 called **null hypothesis** is tested against another hypothesis H_1 called **alternative**
- The two hypotheses are not at the same level: reject H_0 does not mean acceptance of H_1
- The objective is to know when the differences in H_0 are due to randomness or not

Basic concepts

Possible Outcomes of a Test

- Given a sample, a decision is taken about the null hypothesis H_0
- The decision is taken under uncertainty

		H_0 True	H_0 False
Decision	Accept	✓	Type II error (β)
	Reject	Type I error (α)	✓

A simple hypothesis test example

- A natural process is given in nature that follows a Gaussian distribution $N(\mu, \sigma^2)$
- We have a sample of this process $\{x_1, \ldots, x_n\}$ and a decision must be taken about the following hypotheses: $\begin{cases} H_0: & \mu = 60 \\ H_1: & \mu = 50 \end{cases}$
- A statistic (function) of the sample is used to take the decision. In our example $\bar{X} = \frac{1}{n} \sum_{i=1}^{n} x_i$

A simple hypothesis test example

Accept and Reject Regions

- The possible values of the statistic are divided in accept and reject regions

$$A.R. = \{(x_1, \ldots, x_n) | X > 55\}$$
$$R.R. = \{(x_1, \ldots, x_n) | X \leqslant 55\}$$

- Assuming a probability distribution on the statistic $\bar{X}$ (it depends on the distribution of $\{x_1, \ldots, x_n\}$) the probability of each error type can be calculated

$$\alpha = P_{H_0}(\bar{X} \in R.R.) = P_{H_0}(\bar{X} \leqslant 55)$$
$$\beta = P_{H_1}(\bar{X} \in A.R.) = P_{H_1}(\bar{X} > 55)$$

A simple hypothesis test example

Accept and Reject Regions

- The A.R. and R.R. can be modified in order to have a particular value of α

$$0.1 = \alpha = P_{H_0}(\bar{X} \in R.R.) = P_{H_0}(\bar{X} \leqslant 51)$$
$$0.05 = \alpha = P_{H_0}(\bar{X} \in R.R.) = P_{H_0}(\bar{X} \leqslant 50.3)$$

- p-value. Given a sample and the specific value of the test statistic $\bar{x}$ for the sample

$$p - value = P_{H_0}(\bar{X} \leqslant \bar{x})$$

- The p value is the probability of obtaining an effect at least as extreme as the one in your sample data, assuming the truth of the null hypothesis

Interpretation of p-value

Probability of accepting H_1 when H_0 is true

Hypothesis Testing: Remarks

Power: $(1 - \beta)$

The statistical power of a binary hypothesis test is the probability that the test correctly rejects the null hypothesis, H_0, when a specific alternative hypothesis, H_1, is true.

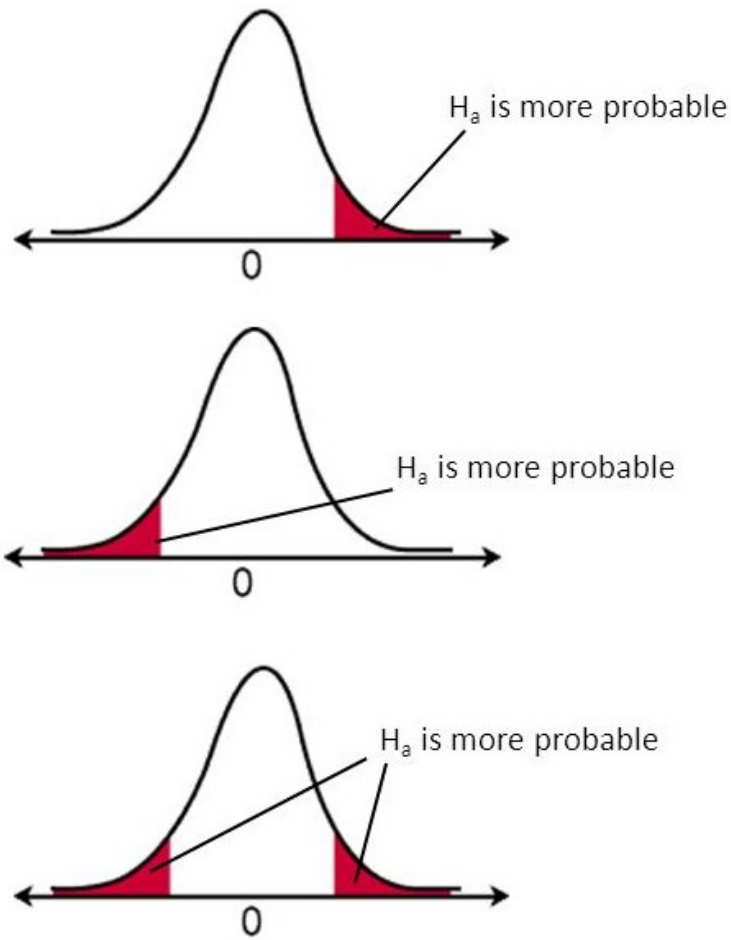
- Depending on the hypotheses the type II error (β) can not be calculated
$$\begin{cases} H_0: & \mu = 60 \\ H_1: & \mu \neq 60 \end{cases}$$
- In this case we do not know the value of μ for H_1 so we can not calculate the power (1-β)
- A good hypothesis test: given an α the test maximizes the power (1-β)

Two-tailed test vs. one-tailed test

- Two-tailed tests
  - ✓ It is associated to an alternative hypotheses for which the sign of the potential difference is unknown. For example, suppose we wish to compare the averages of two samples A and B and we do not know whether A would be higher than B or the opposite. This drives us to choose a two-tailed test, associated to the following alternative hypothesis: H_1: average(A) $\neq$ average(B)

- One-tailed tests
  - ✓ A One-tailed test is associated to an alternative hypothesis for which the sign of the potential difference is known before running the experiment and the test. In the example described above, the alternative hypothesis related to a one-tailed test could be written as follows: average(A) $<$ average(B) or average(A) $>$ average(B)

Two-tailed test vs. one-tailed test



Right-tail test

$H_a: \mu > \text{value}$

Left-tail test

$H_a: \mu < \text{value}$

Two-tail test

$H_a: \mu \neq \text{value}$

Hypothesis Testing: Remarks

Parametric test vs non-parametric test

- **Parametric tests** assume underlying statistical distributions in the data.
  - ✓ Parametric tests can provide trustworthy results with distributions that are skewed and nonnormal
  - ✓ Parametric tests can provide trustworthy results when the groups have different amounts of variability
  - ✓ Parametric tests have greater statistical power
- **Nonparametric tests** do not rely on any distribution
  - ✓ Nonparametric tests assess the median which can be better for some study areas
  - ✓ Nonparametric tests are valid when our sample size is small and your data are potentially nonnormal
  - ✓ Nonparametric tests can analyze ordinal data, ranked data, and outliers

Hypothesis Testing in Supervised Classification

Scenarios

- Testing two algorithms in a Dataset
- Testing two algorithms in several Datasets
- Testing several algorithms in several Datasets

Testing Two Algorithms in a Dataset

The general approach

$$\begin{cases} H_0: & \text{classifier } \phi \text{ has the same score value as classifier } \phi' \text{ in } p(\boldsymbol{x}, c) \\ H_1: & \text{they have different values} \end{cases}$$

$$\begin{cases} H_0: & \textbf{algorithm } \phi \text{ has the same } \textbf{average} \text{ score value as } \textbf{algorithm } \phi' \text{ in } p(\boldsymbol{x}, c) \\ H_1: & \text{they have different values} \end{cases}$$

An Ideal Context: We Can Sample $p(\mathbf{x}, c)$

1. Sample i.i.d. $2n$ datasets from $p(\mathbf{x}, c)$
2. Learn $2n$ classifiers $\phi_i^1, \phi_i^2, i = 1, \ldots, n$
3. For each classifier obtain enough i.i.d. samples $\{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \ldots (\mathbf{x}_N, c_N)\}$ from $p(\mathbf{x}, c)$
4. For each data set calculate the error of each algorithm in the test set $\epsilon_i^1, \epsilon_i^2$
5. Calculate the average values over the n training datasets

$$\bar{\epsilon}^1 = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i^1, \quad \bar{\epsilon}^2 = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i^2 \tag{1}$$

An Ideal Context: We Can Sample $p(x, c)$

- Our test rejects the null hypothesis if $|\bar{\epsilon}^1 - \bar{\epsilon}^2|$ (the statistic) is big
- Fortunately, by the central limit theorem

$$\bar{\epsilon}^1 \rightsquigarrow \mathcal{N}(score(\phi^i), s_i), \quad i = 1, 2$$

- Therefore, under the null hypothesis

$$\hat{Z} = \frac{\bar{\epsilon}^1 - \bar{\epsilon}^2}{\sqrt{\frac{s_1^2 + s_2^2}{n}}} \rightsquigarrow \mathcal{N}(0, 1)$$

- Then, finally we reject H_0 when $|\hat{Z}| > z_{1-\alpha/2}$

The not so ideal context

Properties of Our Ideal Framework

- Training datasets are independent
- Testing datasets are independent

The Sad Reality

- We can not get i.i.d. training samples from $p(\boldsymbol{x}, c)$
- We can not get i.i.d. testing samples from $p(\boldsymbol{x}, c)$
- We have only one sample from $p(\boldsymbol{x}, c)$, the training set D

McNemar test (non-parametric)

- Compare two classifiers in a dataset after a Hold-Out process
- It is a paired non-parametric test

		ϕ^2	
		Error	Ok
ϕ^1	Error	n_{00}	n_{01}
	Ok	n_{10}	n_{11}

- Under H_0 we have $n_{10} \approx n_{01}$ and the statistic

$$\frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}$$

follows a χ^2 distribution with 1 degree of freedom

- When $n_{01} + n_{10}$ is small (<25) the binomial distribution can be used

Tests Based on Resampling: Resampled t-test (parametric)

- The dataset is randomly divided n times in training and test
- Let $\hat{p}_i$ be the difference between the performance of both algorithms in run i and the average performance $\bar{p}$. When it is assumed that $\hat{p}_i$ are Gaussian and independent, under the null hypothesis

$$t = \frac{\bar{p}\sqrt{n}}{\sqrt{\frac{\sum_{i=1}^{n}(\hat{p}_i - \bar{p})^2}{n-1}}} \tag{2}$$

  follows a t student distribution with n–1 degrees of freedom

- Caution:
  - ✓ $\hat{p}_i$ are not Gaussian as $\hat{p}_i^1$ and $\hat{p}_i^2$ are not independent
  - ✓ $\hat{p}_i$ are not independent (overlap in training and testing)

Resampled t-test Improved (Nadeau and Bengio, 2003)

- The variance in this case is too optimistic
- Two alternatives
  - ✓ Corrected resampled t

$$\left(\frac{1}{n} + \frac{n_2}{n_1}\right)\sigma^2$$

  - ✓ Conservative Z (overestimation of the variance)

t-test for k-fold Cross-validation

- It is similar to *t*-test for resampling
- In this case the testing datasets are independent
- The training datasets are still dependent

5x2 fold Cross-Validation (Dietterich, 1998; Alpaydin, 1999)

- Each Cross-Validation process has independent training and testing datasets
- A 2-Fold Cross-validation is repeated 5 times
- The following statistic

$$\frac{\sum_{i=1}^{5}\sum_{j=1}^{2}(p_i^j)^2}{2\sum_{i=1}^{5}s_i^2}$$

follows a F distribution with 10 and 5 degrees of freedom under the null hypothesis

Testing Two Algorithms in Several Datasets

Initial approach

- Averaging over datasets
- Paired t-test
  - ✓ $c^i = c_1^i - c_2^i$ and $\bar{d} = \frac{1}{N}\sum_{i=1}^{N} c^i$ then $\bar{d}/\sigma_{\bar{d}}$ follows a t distribution with N-1 degrees of freedom

Problems

- Commensurability (sharing of a common measure)
- Outlier susceptibility
- (t-test) Gaussian assumption

Wilcoxon Signed-Ranks Test (Demšar, 2006)

- It is a non-parametric test that works as follows:
  1. Rank the module of the performance differences between both algorithms
  2. Calculate the sum of the ranks R^+ and R^- where the first (resp. the second) algorithm outperforms the other
  3. Calculate $T = \min(R^+, R^-)$
- For $N \leqslant 25$ there are tables with critical values
- For N > 25

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} \quad \rightsquigarrow \quad \mathcal{N}(0,1) \tag{3}$$

Wilcoxon Signed-Ranks Test

	ψ^1	ψ^2	diff	rank
Dataset1	0.763	0.598	-0.165	10
Dataset2	0.599	0.591	-0.008	6
Dataset3	0.954	0.971	+0.017	7
Dataset4	0.628	0.661	+0.033	8
Dataset5	0.882	0.888	+0.006	4
Dataset6	0.936	0.931	-0.005	3
Dataset7	0.661	0.668	+0.007	5
Dataset8	0.583	0.583	0.000	1.5
Dataset9	0.775	0.838	+0.063	9
Dataset10	1.000	1.000	0.000	1.5

$$R^+ = 7 + 8 + 4 + 5 + 9 + 1/2(1{,}5 + 1{,}5)$$

Wilcoxon Signed-Ranks Test

	ψ^1	ψ^2	diff	rank
Dataset1	0.763	0.598	-0.165	10
Dataset2	0.599	0.591	-0.008	6
Dataset3	0.954	0.971	+0.017	7
Dataset4	0.628	0.661	+0.033	8
Dataset5	0.882	0.888	+0.006	4
Dataset6	0.936	0.931	-0.005	3
Dataset7	0.661	0.668	+0.007	5
Dataset8	0.583	0.583	0.000	1.5
Dataset9	0.775	0.838	+0.063	9
Dataset10	1.000	1.000	0.000	1.5

$R^+ = 34.5 \qquad R^- = 10 + 6 + 3 + 1/2(1{,}5 + 1{,}5)$

Wilcoxon Signed-Ranks Test

	ψ^1	ψ^2	diff	rank
Dataset1	0.763	0.598	-0.165	10
Dataset2	0.599	0.591	-0.008	6
Dataset3	0.954	0.971	+0.017	7
Dataset4	0.628	0.661	+0.033	8
Dataset5	0.882	0.888	+0.006	4
Dataset6	0.936	0.931	-0.005	3
Dataset7	0.661	0.668	+0.007	5
Dataset8	0.583	0.583	0.000	1.5
Dataset9	0.775	0.838	+0.063	9
Dataset10	1.000	1.000	0.000	1.5

$R^+ = 34.5 \qquad R^- = 20.5 \qquad T = min(R^+, R^-) = 20.5$

Wilcoxon Signed-Ranks Test

- It also suffers from commensurability but only qualitatively
- When the assumptions of the t test are met, Wilcoxon is less powerful than t test

Sign test

- It is a non-parametric test that counts the number of losses, ties and wins
- Under the null the number of wins follows a binomial distribution $B(1/2, N)$
- For large values of N the number of wins follows $\mathcal{N}(N/2, \sqrt{N/2})$ under the null hypothesis
- This test does not make any assumptions
- It is weaker than Wilcoxon

Sign test

Critical values

#datasets	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$w_{0.05}$	5	6	7	7	8	9	9	10	10	11	12	12	13	13	14	15	15	16	17	18	18
$w_{0.10}$	5	6	6	7	7	8	9	9	10	10	11	12	12	13	13	14	14	15	16	16	17

Critical values for the two-tailed sign test at $\alpha = 0.05$ and $\alpha = 0.10$. A classifier is significantly better than another if it performs better on at least w_α data sets.

- It is much weaker than the Wilcoxon signed-ranks test
- As shown in table, the sign test will not reject the null-hypothesis unless one algorithm almost always outperforms the other

Testing Several Algorithms in Several Datasets

Several methods, ϕ^1, ϕ^2, ϕ^3 and ϕ^4, and several datasets, D_1, D_2, D_3, D_4, D_5 and D_6

	ϕ^1	ϕ^2	ϕ^3	ϕ^4
D_1	0.84	0.79	0.89	0.43
D_2	0.57	0.78	0.78	0.93
D_3	0.62	0.87	0.88	0.71
D_4	0.95	0.55	0.49	0.72
D_5	0.84	0.67	0.89	0.89
D_6	0.51	0.63	0.98	0.55

Multiple Hypothesis Testing

- Testing all possible pairs of hypotheses $\mu_{\phi^i} = \mu_{\phi^j}, \forall i, j$. Multiple hypothesis testing
- Testing the hypothesis $\mu_{\phi^1} = \mu_{\phi^2} = \ldots = \mu_{\phi^k}$
  - ✓ ANOVA vs Friedman
    - Repeated measures ANOVA: Assumes Gaussianity and sphericity
    - Friedman: Non-parametric test

Friedman test

1. Rank the algorithms for each dataset separately (1-best). In case of ties assigned average ranks
2. Calculate the average rank R_j of each algorithm ϕ^j
3. The following statistic:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right]$$

follows a χ^2 with k-1 degrees of freedom ($N > 10, k > 5$)

Friedman test: Example

	ψ^1	ψ^2	ψ^3	ψ^4
D_1	0.84 (2)	0.79 (3)	0.89 (1)	0.43 (4)
D_2	0.57 (4)	0.78 (2.5)	0.78 (2.5)	0.93 (1)
D_3	0.62 (4)	0.87 (2)	0.88 (1)	0.71 (3)
D_4	0.95 (1)	0.55 (3)	0.49 (4)	0.72 (2)
D_5	0.84 (3)	0.67 (4)	0.89 (1.5)	0.89 (1.5)
D_6	0.51 (4)	0.63 (2)	0.98 (1)	0.55 (3)
avr. rank	3	2.75	1.83	2.41

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] = 2{,}5902$$

Iman-Davenport test (Iman and Davenport, 1980)

- An improved Friedman test

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}$$

follows a F distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom

```
#!      /usr/bin/pytyhon
from scipy.stats import f
p_value = 1.0 - f.cdf(F_F, k-1, (k-1)*(N-1))
```

Post-hoc tests

- A *post-hoc* test is used only after we find a statistically significant result and need to determine where our differences truly came from
- The term *post hoc* comes from the Latin for "after the event"
- Decision on the null hypothesis
- In case of rejection use of *post-hoc* tests to
  - ✓ Compare all pairs
  - ✓ Compare all classifiers with a control

Multiple Hypothesis Testing

- Several related hypothesis simultaneously $H_1, H_2, \ldots, H_n$

		H_0 True	H_0 False
Decision	Accept	✓	Type II error (β)
	Reject	Type I error (α)	✓

- Family-wise error (FWE): Probability of rejecting at least one NULL hypothesis assuming that ALL ARE TRUE
- False discovery rate (FDR): proportion of "discoveries" (rejected null hypotheses) that are false (incorrect rejections of the null)

Designing Multiple Hypothesis Test

- Controlling family-wise error
- If each test H_i has a type I error α then the family-wise error (FWE) in n tests is:

$$\begin{aligned} & P(\text{accept } H_1 \cap \text{ accept } H_2 \cap \ldots \cap \text{accept } H_n) \\ = & P(\text{accept } H_1) \times P(\text{accept } H_2) \times \ldots \times P(\text{accept } H_n) \\ & = (1\text{-}\alpha)^n \end{aligned} \tag{4}$$

and therefore

$$FWE = 1\text{-}(1\text{-}\alpha)^n \approx 1\text{-}(1\text{-}\alpha n) = \alpha n \tag{5}$$

In order to have FWE α we need to modify the threshold at each test

Comparing with a control or pairwise

- The statistic for comparing ϕ^i and ϕ^j is

$$z = \frac{(R_i - R_j)}{\sqrt{\frac{k(k+1)}{6N}}} \rightsquigarrow \mathcal{N}(0,1) \tag{6}$$

where k is the number of methods and N the number of datasets

Bonferroni-Dunn test

- It is a one-step method
- Modify α by taking into account the number of comparisons

$$\frac{\alpha}{K}$$

- K is the number of comparisons performed, $K = k - 1$ for a comparison against a control or $K = k(k-1)/2$ for an all pairs comparison

Comparing with a control or pairwise

- Methods based on ordered p-values
- The p-values are ordered $p_1 \leqslant p_2 \leqslant \ldots \leqslant p_K$
- K is the number of comparisons performed, $K = k - 1$ for a comparison against a control or $K = k(k-1)/2$ for an all pairs comparison

Holm method

- It is a step-down procedure
- Starting from p_1 check the first $i = 1, \ldots, k$-1 such that $p_i > \alpha/(K + 1\text{-}i)$
- For the comparison against a control: $\alpha/(k - i)$
- The hypothesis $H_1, \ldots, H_{i\text{-}1}$ are rejected. The rest of hypotheses are kept

Bonferroni-Dunn and Holm tests example: Pairwise comparison

	ψ^1	ψ^2	ψ^3	ψ^4
D_1	0.84 (2)	0.79 (3)	0.89 (1)	0.43 (4)
D_2	0.57 (4)	0.78 (2.5)	0.78 (2.5)	0.93 (1)
D_3	0.62 (4)	0.87 (2)	0.88 (1)	0.71 (3)
D_4	0.95 (1)	0.55 (3)	0.49 (4)	0.72 (2)
D_5	0.84 (3)	0.67 (4)	0.89 (1.5)	0.89 (1.5)
D_6	0.51 (4)	0.63 (2)	0.98 (1)	0.55 (3)
avr. rank	3	2.75	1.83	2.41

Bonferroni-Dunn and Holm tests example: Pairwise comparison

$$z = \frac{(R_i - R_j)}{\sqrt{\frac{k(k+1)}{6N}}}$$

	z
z_{12}	0.3354
z_{13}	1.5697
z_{14}	0.7915
z_{23}	1.2343
z_{24}	0.4561
z_{34}	-0.7781

Bonferroni-Dunn and Holm tests example: Pairwise comparison

	z	p-value
z_{12}	0.3354	0.7373
z_{13}	1.5697	0.1165
z_{14}	0.7915	0.4286
z_{23}	1.2343	0.2171
z_{24}	0.4561	0.6483
z_{34}	-0.7781	0.4365

```
#!      /usr/bin/pytyhon
from scipy.stats import norm
p_value = 1.0-norm.cdf(z)
```

Bonferroni-Dunn and Holm tests example: Pairwise comparison

Consider $\alpha = 0.05$, $k = 6$ (number of comparisons)

	z	p-value	Bonferroni ($\alpha/6$)	Holm ($\alpha/(7-i)$)
z_{12}	0.3354	0.7373	0.008 ✓	0.050 ✗
z_{13}	1.5697	0.1165	0.008 ✓	0.008 ✓
z_{14}	0.7915	0.4286	0.008 ✓	0.012 ✗
z_{23}	1.2343	0.2171	0.008 ✓	0.010 ✗
z_{24}	0.4561	0.6483	0.008 ✓	0.025 ✗
z_{34}	-0.7781	0.4365	0.008 ✓	0.017 ✗

Reject NULL hypothesis

✓: Accept NULL hypothesis
✗: Not even considered

Holm tests example: Compare with a control

- First option: Best against all
- Best Friedman rank: $\phi_3, R_3 = 1.83$

	z	p-value	Holm $(\alpha/(4-i))$
z_{43}	0.7781	0.4365	0.0500 ✗
z_{23}	1.2343	0.2171	0.0250 ✗
z_{13}	1.5697	0.1165	0.0167 ✓

Reject NULL hypothesis
✓: Accept NULL hypothesis
✗: Not even considered

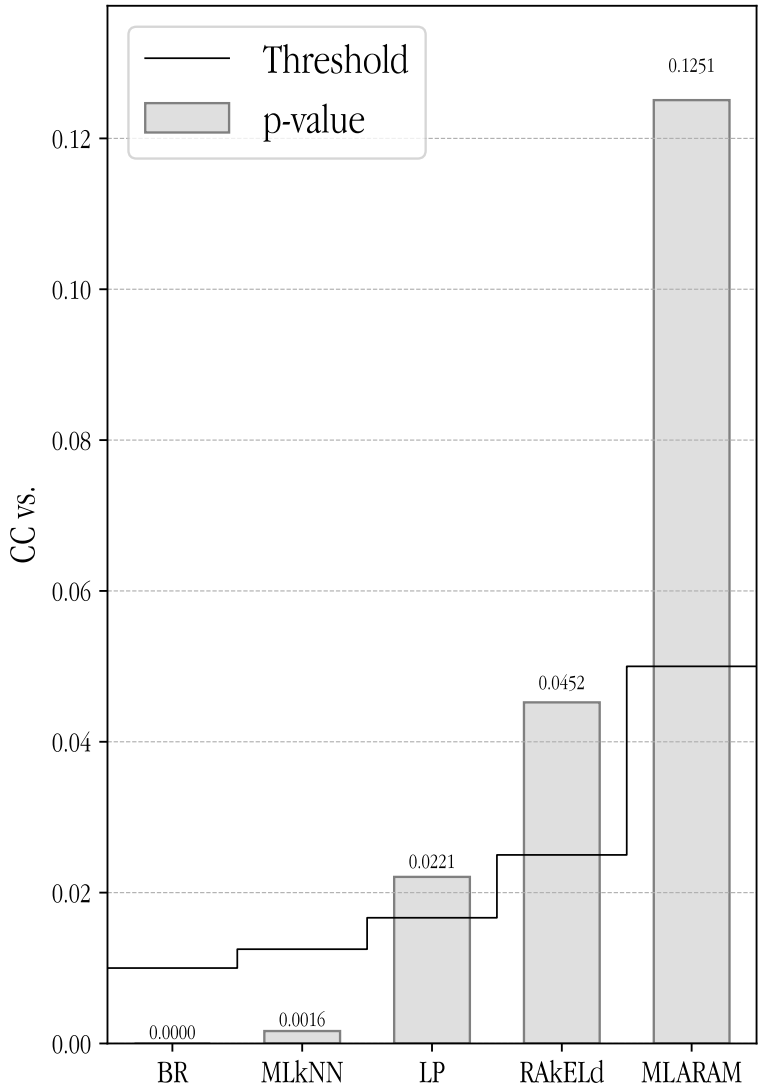
Holm tests example: Compare with a control

- First option: Worst against all
- Worst Friedman rank: $\phi_1, R_1 = 3.00$

	z	p-value	Holm $(\alpha/(4-i))$
z_{13}	1.5697	0.1165	0.0167 ✓
z_{14}	0.7916	0.4286	0.0250 ✗
z_{12}	0.3354	0.7373	0.0500 ✗

Reject NULL hypothesis
✓: Accept NULL hypothesis
✗: Not even considered

Holm tests example: Graphical representation



Testing Several Algorithms in Several Datasets

Hochberg Method

- It is a step-up procedure
- Starting with p_{k-1} check the first $i = k-1, \ldots, 1$ such that $p_i < \alpha/(k-i)$
- The hypothesis $H_1, \ldots, H_{i-1}$ are rejected. The rest of hypotheses are kept

Hommel Method

- Find the largest j such that $p_{n-j+k} > k\alpha/j$ for all $k = 1, \ldots, j$, where n is the number of hypothesis
- Reject all hypotheses i such that $p_i \leqslant \alpha/j$

Comments on the tests

- Holm, Hochberg and Hommel tests are more powerful than Bonferroni
- Hochberg and Hommel are based on Simes conjecture and can have a higher than α FWE
- In practice Holm obtains very similar results to the other

Simes conjeture

Let $X_{1:n} \leqslant \ldots \leqslant X_{n:n}$ denote the n ordered values of a set of random variables $X_1, \ldots, X_n$. Assuming that the X_i's are continuous with a common cdf F, Simes considered the following inequality involving the ordered values $U_{1:n} \leqslant \ldots \leqslant U_{n:n}$ of $U(0,1)$ random variables $U_i = F(X_i), i = 1, \ldots, n$,

$$P\{U_{i:n} \geqslant i\alpha/n, i = 1, \ldots, n\} \geqslant 1\text{-}\alpha$$

All Pairwise Comparisons

- Differences with Comparing with a Control
- The all pairwise hypotheses are logically related: not all combinations of true and false hypotheses are possible
  - ✓ C1 better than C2 and
  - ✓ C2 better than C3 and
  - ✓ C1 equal to C3

Shaffer Static Procedure

- It is a modification of Holm's procedure
- Starting from p_1 check the first $i = 1, \ldots, k(k-1)/2$ such that $p_i > \alpha/t_i$
- The hypothesis $H_1, \ldots, H_{i-1}$ are rejected. The rest of hypotheses are kept
- t_i is the maximum number of hypotheses that can be true given that $(i-1)$ are false
- It is a static procedure: t_i is determined given the hypotheses independently of the p-values

Shaffer Dynamic Procedure

- It is similar to the previous procedure but t_i is changed by t_i^*
- t_i^* considers the maximum number of hypotheses that can be true given that the previous $(i-1)$ hypotheses are false
- It is a dynamic procedure as t_i^* depends on the hypotheses already rejected
- It is more powerful than the Shaffer Static Procedure

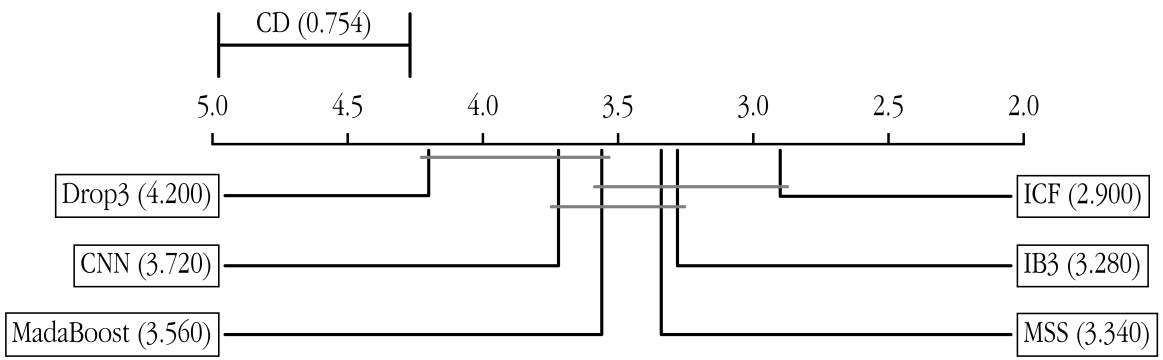
Nemenyi test

- Based on Friedman ranks
- Methods significantly different if rank different above critical value

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{7}$$

where k is the number of methods, N is the number of datasets and q_α is the critical value (Student's t)

Graphical representation



Hypothesis testing conclusions

Two Classifiers in a Dataset

The complexity of the estimation of the scores makes it difficult to carry out good statistical testing

Two Classifiers in Several Datasets

- Wilcoxon Signed-Ranks Test is a good choice
- In case of many datasets and to avoid the commensurability problem the Sign test could be used

Several Classifiers in Several Datasets

- Friedman or Iman & Davenport are required
- Post-hoc test more powerful than Bonferroni:
  - ✓ Comparison with a control: Holm method
  - ✓ All-to-all comparison: Nemenyi test

References 1


Alpaydin, E. (1999). 'Combined 5 × 2 cv F test for comparing supervised classification learning algorithms'. In: *Neural Computation* 11, pp. 1885–1892.


Burman, P. (1989). 'A Comparative Study of Ordinary Cross-Validation, v-Fold Cross-Validation and the Repeated Learning-Testing Methods'. In: *Biometrika* 76, pp. 503–514.


Demšar, J. (2006). 'Statistical Comparisons of Classifiers over Multiple Data Sets'. In: *Journal of Machine Learning Research* 7, pp. 1–30.


Dietterich, T. G. (1998). 'Approximate statistical tests for comparing supervised classification learning algorithms'. In: *Neural Computation* 10.7, pp. 1895–1923.


Iman, R. L. and J. M. Davenport (1980). 'Approximations of the critical regions of the Friedman statistic'. In: *Communications in Statistics* 6, pp. 571–595.


Jurman, G., S. Riccadonna, and C. Furlanello (2012). 'A Comparison of MCC and CEN Error Measures in Multi-Class Prediction'. In: *PLoS ONE* 7 (8), e41882.

References II

Lozano, J. A., G. Santafé, and I. Inza (2010). 'Classifier performance evaluation and comparison'. In: *International Conference on Machine Learning and Applications (ICMLA 2010)*.

Nadeau, C. and Y. Bengio (2003). 'Inference for the generalization error'. In: *Machine Learning* 52, pp. 239–281.

Wei, J.-M. et al. (2010). 'A novel measure for evaluating classifiers'. In: *Expert Systems with Applications* 37 (5), pp. 3799–3809.