Honest Evaluation of Classification Models

Jose A. Lozano, Guzmán Santafé, Iñaki Inza

Intelligent Systems Group
The University of the Basque Country

Asian Conference on Machine Learning (ACML'10)
November 8-10, 2010

Outline of the Tutorial

1. Introduction
2. Scores
3. Estimation Methods
4. Hypothesis Testing

Outline of the Tutorial

Classification Problem

Physical Process $\sim$ $\rho(\boldsymbol{X}, C)$ **Usually unknown**

Data set

X_1	X_2	X_3	$\dots$	X_n	C
$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$\dots$	$x_n^{(1)}$	?
$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$	$\dots$	$x_n^{(2)}$	?
$x_1^{(3)}$	$x_2^{(3)}$	$x_3^{(3)}$	$\dots$	$x_n^{(3)}$	?
$x_1^{(4)}$	$x_2^{(4)}$	$x_3^{(4)}$	$\dots$	$x_n^{(4)}$	?
		$\dots$	$\dots$		
$x_1^{(N)}$	$x_2^{(N)}$	$x_3^{(N)}$	$\dots$	$x_n^{(N)}$	?

Classification Problem

Physical Process $\sim$ $\rho(X, C)$ **Usually unknown**

Data set

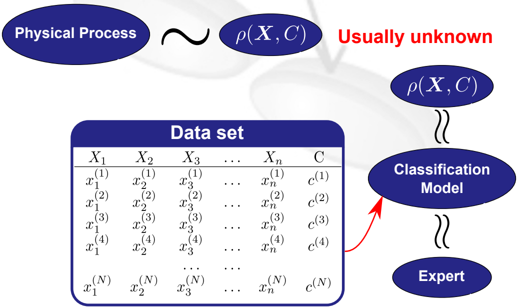
X_1	X_2	X_3	$\ldots$	X_n	C
$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$\ldots$	$x_n^{(1)}$	$c^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$	$\ldots$	$x_n^{(2)}$	$c^{(2)}$
$x_1^{(3)}$	$x_2^{(3)}$	$x_3^{(3)}$	$\ldots$	$x_n^{(3)}$	$c^{(3)}$
$x_1^{(4)}$	$x_2^{(4)}$	$x_3^{(4)}$	$\ldots$	$x_n^{(4)}$	$c^{(4)}$
		$\ldots$	$\ldots$		
$x_1^{(N)}$	$x_2^{(N)}$	$x_3^{(N)}$	$\ldots$	$x_n^{(N)}$	$c^{(N)}$

Expert

Supervised Classification

Learning from Experience

- “Automate the work of the expert”
- Tries to model $\rho(C, \boldsymbol{X})$



X_1	X_2	X_3	$\dots$	X_n	C
$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$\dots$	$x_n^{(1)}$	$c^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$	$\dots$	$x_n^{(2)}$	$c^{(2)}$
$x_1^{(3)}$	$x_2^{(3)}$	$x_3^{(3)}$	$\dots$	$x_n^{(3)}$	$c^{(3)}$
$x_1^{(4)}$	$x_2^{(4)}$	$x_3^{(4)}$	$\dots$	$x_n^{(4)}$	$c^{(4)}$
		$\dots$	$\dots$		
$x_1^{(N)}$	$x_2^{(N)}$	$x_3^{(N)}$	$\dots$	$x_n^{(N)}$	$c^{(N)}$

Supervised Classification

Classification Model

- Classifier labels new data (unknown class value)

$\rho(\boldsymbol{X}, C)$

$\wr\wr$

Data set

X_1	X_2	X_3	$\ldots$	X_n	C
1	1	0	$\ldots$	0	?
0	1	1	$\ldots$	1	?
1	0	1	$\ldots$	1	?
1	1	0	$\ldots$	0	?
0	0	1	$\ldots$	1	?

→ **Classification Model** →

Data set

X_1	X_2	X_3	$\ldots$	X_n	C
1	1	0	$\ldots$	0	1
0	1	1	$\ldots$	1	1
1	0	1	$\ldots$	1	0
1	1	0	$\ldots$	0	1
0	0	1	$\ldots$	1	0

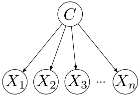
$\wr\wr$

Expert

Motivation for Honest Evaluation

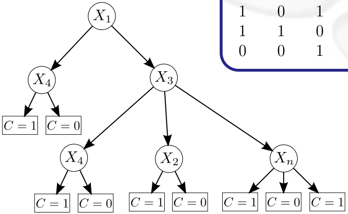
- Many classification paradigms

Naive Bayes

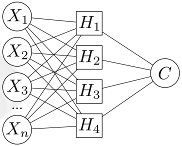


Data set

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



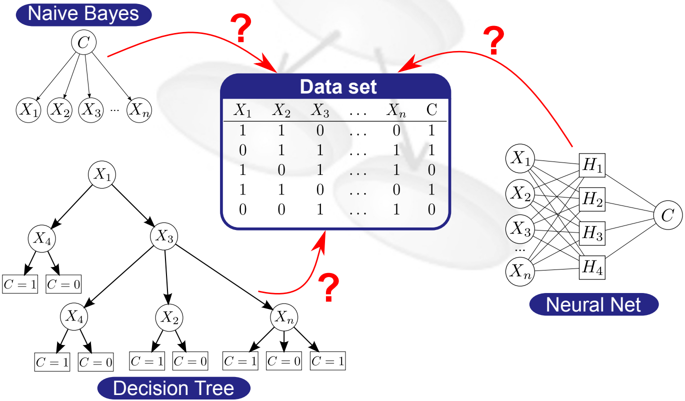
Decision Tree



Neural Net

Motivation for Honest Evaluation

- Which is the best paradigm for a classification problem?

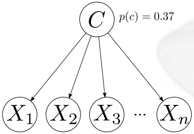


X_1	X_2	X_3	...	X_n	C
1	1	0	...	0	1
0	1	1	...	1	1
1	0	1	...	1	0
1	1	0	...	0	1
0	0	1	...	1	0

Motivation for Honest Evaluation

- Many parameter configurations

Naive Bayes

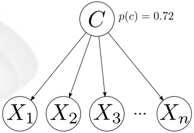


$$
\begin{aligned}
p(x_1|c) &= 0.25 & p(x_3|c) &= 0.05\\
p(x_1|\bar{c}) &= 0.60 & p(x_3|\bar{c}) &= 0.60\\
p(x_2|c) &= 0.20 & p(x_n|c) &= 0.80\\
p(x_2|\bar{c}) &= 0.70 & p(x_n|\bar{c}) &= 0.21
\end{aligned}
$$

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

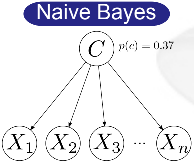
Naive Bayes



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

Motivation for Honest Evaluation

- Which is the best parameter configuration for a classification problem?



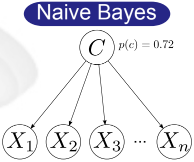
$$
\begin{aligned}
p(x_1|c) &= 0.25 & p(x_3|c) &= 0.05\\
p(x_1|\bar{c}) &= 0.60 & p(x_3|\bar{c}) &= 0.60\\
p(x_2|c) &= 0.20 & p(x_n|c) &= 0.80\\
p(x_2|\bar{c}) &= 0.70 & p(x_n|\bar{c}) &= 0.21
\end{aligned}
$$

?

Data set

X_1	X_2	X_3	$\ldots$	X_n	C
1	1	0	$\ldots$	0	1
0	1	1	$\ldots$	1	1
1	0	1	$\ldots$	1	0
1	1	0	$\ldots$	0	1
0	0	1	$\ldots$	1	0

?



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

Motivation for Honest Evaluation

Honest Evaluation

- Need to know the goodness of a classifier
- Methodology to compare classifiers
- Assess the validity of evaluation/comparison

Steps for Honest Evaluation

- Scores: quality measures
- Estimation methods: estimate value of a score
- Statistical tests: comparison among different solutions

Outline of the Tutorial

Motivation

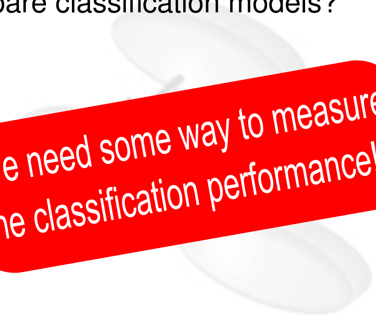
- How to compare classification models?

Score

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

Motivation

- How to compare classification models?



We need some way to measure the classification performance!!!

Score

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

Motivation

- How to compare classification models?

We need some way to measure 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/Classification error

- Recall
- Specificity
- Precision
- F-Score

Based on Receiver Operating Characteristics (ROC)

- Area under the ROC curve (AUC)

Scores

Based on Confusion Matrix

- Accuracy/Classification error $\longrightarrow$ Classification
- Recall
- Specificity
- Precision
- F-Score

Based on Receiver Operating Characteristics (ROC)

- Area under the ROC curve (AUC)

Scores

Based on Confusion Matrix

- Accuracy/Classification error $\longrightarrow$ Classification
- Recall
- Specificity $\longrightarrow$ Information Retrieval
- Precision
- F-Score

Based on Receiver Operating Characteristics (ROC)

- Area under the ROC curve (AUC)

Scores

Based on Confusion Matrix

- Accuracy/Classification error $\longrightarrow$ Classification
- Recall
- Specificity $\longrightarrow$ Information Retrieval
- Precision
- F-Score

Based on Receiver Operating Characteristics (ROC)

- Area under the ROC curve (AUC) $\longrightarrow$ Medical Domains

Confusion Matrix

Two-Class Problem

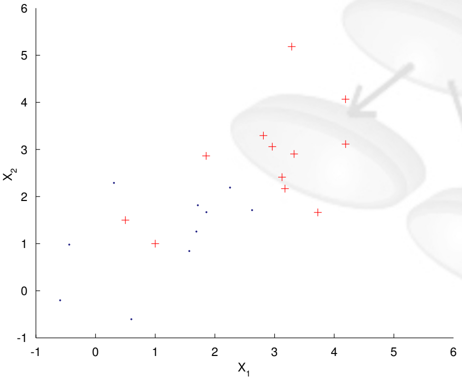
		Prediction		
		c^+	c^-	Total
Actual	c^+	TP	FP	N^+
	c^-	FN	TN	N^-
	Total	$\hat{N}^+$	$\hat{N}^-$	N

Confusion Matrix

Several-Class Problem

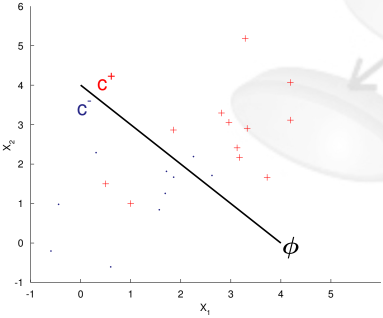
		Prediction					
		c_1	c_2	c_3	$\dots$	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	$\dots$	FN_{1n}	N_1
	c_2	FN_{21}	TP_2	FN_{23}	$\dots$	FN_{2n}	N_2
	c_3	FN_{31}	FN_{32}	TP_3	$\dots$	FN_{3n}	N_3
	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	$\dots$	TP_n	N_n
	Total	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_3$	$\dots$	$\hat{N}_n$	N

Two-Class Problem - Example



X_1	X_2	C
3,1	2,4	c^+
1,7	1,8	c^-
3,3	5,2	c^+
2,6	1,7	c^-
1,8	2,9	c^+
0,3	2,3	c^-
...	...	...

Two-Class Problem - Example

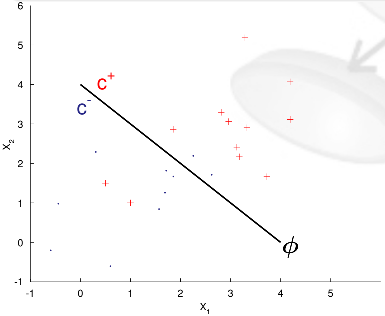


		Prediction c^+	Prediction c^-	Total
Actual	c^+	10	2	12
Actual	c^-	2	8	10
	Total	12	10	**22**

Accuracy/Classification Error

Definition

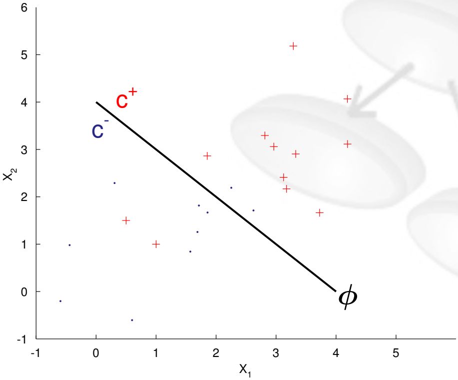
- Data samples classified correctly/incorrectly



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

$$\epsilon(\phi) = p(\phi(\boldsymbol{X}) \neq C) = E_{\rho(\boldsymbol{x},c)}[1 - \delta(c, \phi(\boldsymbol{x}))]$$

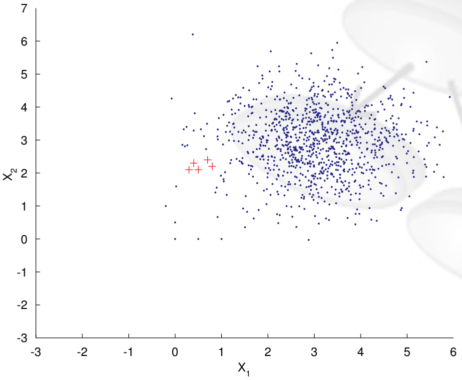
Accuracy/Classification Error



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

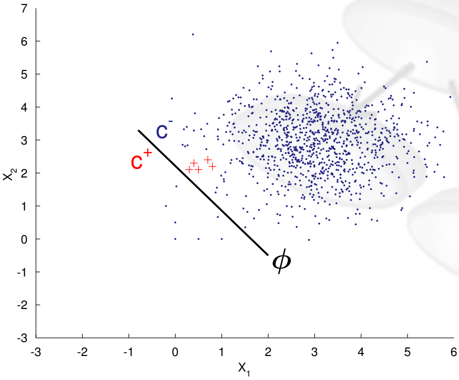
$$\epsilon = \frac{FP + FN}{N} = \frac{2+2}{22} = 0{,}182$$

Skew Data



X_1	X_2	C
0,8	2,2	c^+
0,47	2,3	c^+
0,5	2,1	c^+
2,4	2,9	c^-
3,1	1,2	c^-
2,5	3,1	c^-
...	...	...

Skew Data - Classification Error

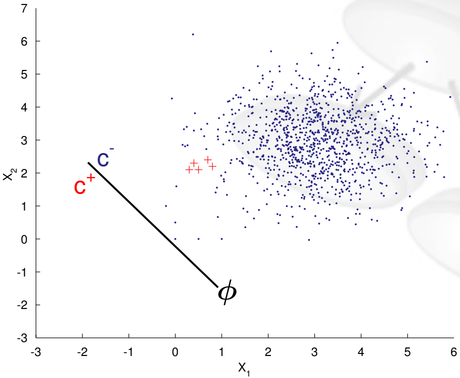


		Prediction c^+	c^-	Total
Actual	c^+	0	5	5
	c^-	7	993	1000
	Total	7	998	**1005**

$$\epsilon = \frac{7+5}{1005} = 0{,}012$$

Very low ϵ!!

Skew Data - Classification Error

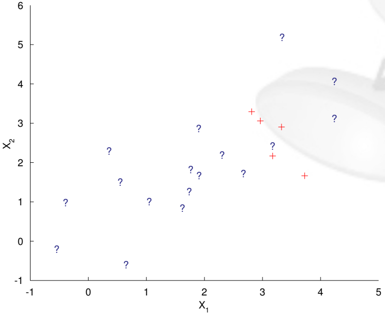


		Prediction c^+	Prediction c^-	Total
Actual	c^+	0	5	5
Actual	c^-	0	1000	1000
	Total	0	1005	**1005**

$$\epsilon = \frac{0+5}{1005} = 0{,}005$$

Better??

Positive Unlabeled Learning



Positive Labeled Data

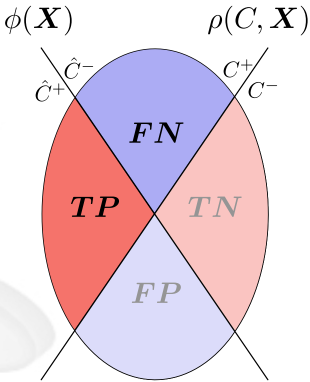
- Only positive samples labeled
- Many unlabeled samples:
  - Positive?
  - Negative?
- Classification error is useless

Recall

Definition

- Fraction of positive class samples correctly classified
- Other names $\begin{cases} \text{True positive rate} \\ \text{Sensitivity} \end{cases}$

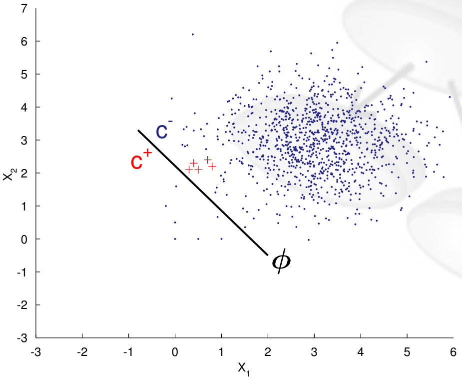
$$r(\phi) = \frac{TP}{TP + FN} = \frac{TP}{P}$$



Definition Based on Probabilities

$$r(\phi) = p(\phi(\boldsymbol{x}) = c^+ | C = c^+) = E_{\rho(\boldsymbol{x}|C=c^+)}[\delta(\phi(\boldsymbol{x}), c^+)]$$

Skew Data - Recall

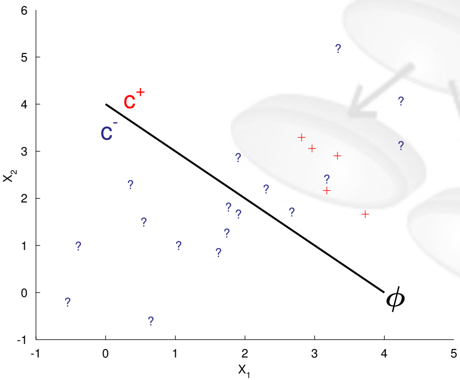


		Prediction c^+	Prediction c^-	Total
Actual	c^+	0	5	5
Actual	c^-	7	993	1000
	Total	7	998	**1005**

$$r(\phi) = \frac{0}{0+5} = 0$$

Very bad recall!!

Positive Unlabeled Learning - Recall

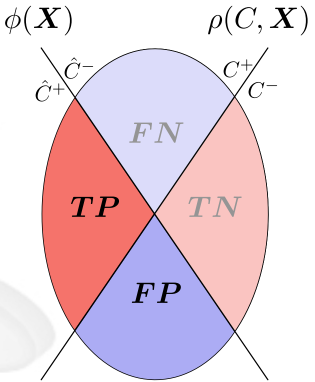


		Prediction c^+	Prediction $c^?$	Total
Actual	c^+	0	5	5
Actual	$c^?$	7	10	1
	Total	12	10	**22**

$$r(\phi) = \frac{5}{0+5} = 1$$

It is possible to calculate recall in positive-unlabeled problems

Precision



Definition

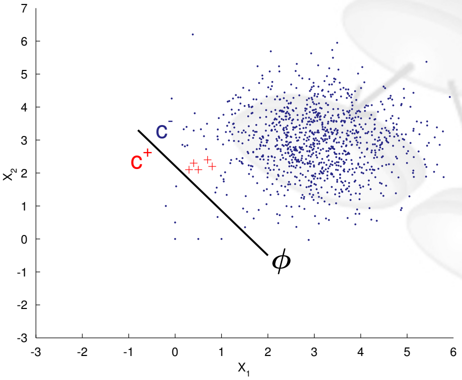
- Fraction of data samples classified as c^+ which are actually c^+

$$pr(\phi) = \frac{TP}{TP + FP} = \frac{TP}{\hat{P}}$$

Definition Based on Probabilities

$$pr(\phi) = p(C = c^+ | \phi(\boldsymbol{x}) = c^+) = E_{\rho(\boldsymbol{x}|\phi(\boldsymbol{x})=c^+)}[\delta(\phi(\boldsymbol{x}), c^+)]$$

Skew Data - Precision

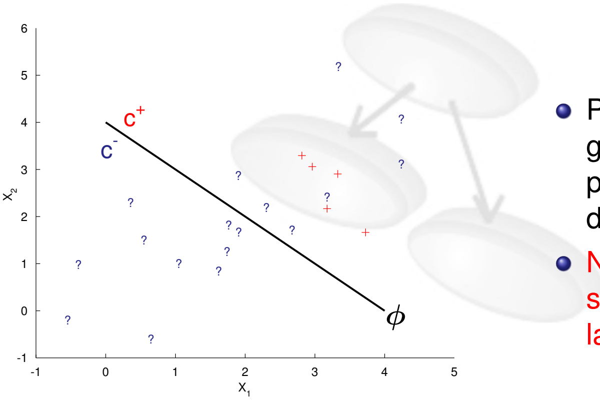


		Prediction c^+	Prediction c^-	Total
Actual	c^+	0	5	5
Actual	c^-	7	993	1000
	Total	7	998	**1005**

$$pr(\phi) = \frac{0}{0+7} = 0$$

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

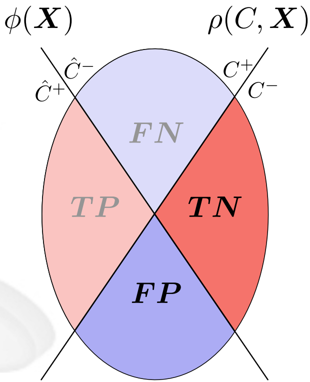
Spam Filtering

- 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, forum, etc.
  - Precision: Proportion of opinions classified as positive being actually positive
  - Recall: Proportion of positive opinions identified as positive

Specificity



Definition

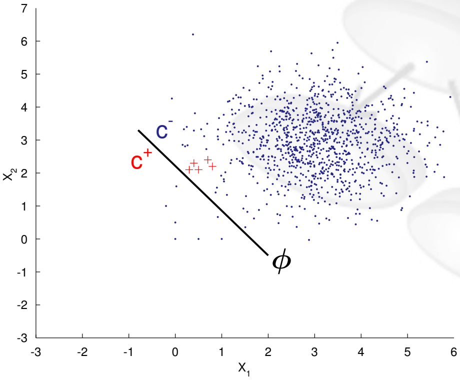
- Fraction of negative class samples correctly identified
- $Specificity = 1 - FalsePositiveRate$

$$sp(\phi) = \frac{TN}{TN + FP} = \frac{TN}{N}$$

Definition Based on Probabilities

$$sp(\phi) = p(\phi(\boldsymbol{x}) = c^- | C = c^-) = E_{\rho(\boldsymbol{x}|C=c^-)}[1 - \delta(\phi(\boldsymbol{x}), c^-)]$$

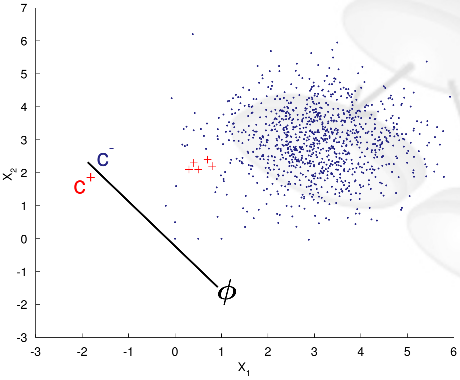
Skew Data - Specificity



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

$$sp(\phi) = \frac{993}{993 + 7} = 0{,}99$$

Skew Data - Specificity



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

$$sp(\phi) = \frac{1000}{1000 + 0} = 1{,}00$$

Balanced Scores

- Balanced accuracy rate

$$Bal.\ acc = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right) = \frac{recall + specificity}{2}$$

- Balanced error rate

$$Bal.\ \epsilon = \frac{1}{2}\left(\frac{FP}{P} + \frac{FN}{N}\right)$$

Skew Data

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

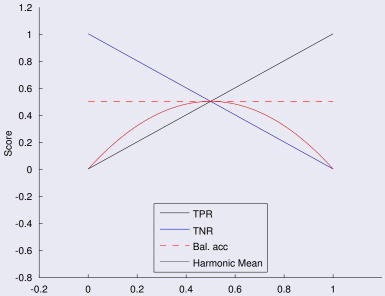
- $Bal.\ acc = \frac{1}{2}\left(\frac{0}{5} + \frac{993}{1000}\right) \approx 0{,}5$
- $Bal.\ \epsilon = \frac{1}{2}\left(\frac{7}{7} + \frac{5}{1000}\right) \approx 0{,}5$

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*

Harmonic Mean

- Maximized with balanced components
- Bal. acc $\rightarrow$ arithmetic mean



Classification Cost

- All misclassifications cannot be equally considered

E.g. 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 $\rightarrow$ 0/1 Loss

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

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

Dealing with Classification Cost

Loss Function

Associate an economic/utility/etc. cost to each classification.

- Typical loss function in classification $\rightarrow$ 0/1 Loss

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

		Prediction	
		c^+	c^-
Actual	c^+	$Cost_{TP}$	$Cost_{FN}$
Actual	c^-	$Cost_{FP}$	$Cost_{TN}$

Dealing with Classification Cost

Loss Function

Associate an economic/utility/etc. cost to each classification.

- Typical loss function in classification $\rightarrow$ 0/1 Loss

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

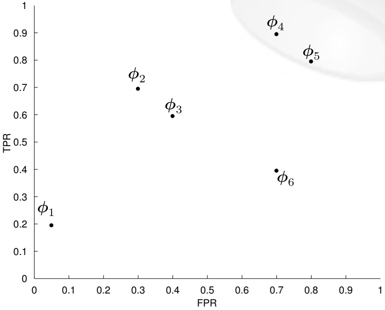
		Prediction	
		c^+	c^-
Actual	c^+	$Cost_{TP}$	$Cost_{FN}$
	c^-	$Cost_{FP}$	$Cost_{TN}$

Usually not easy to give an associated cost

Receiver Operating Characteristics (ROC)

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.

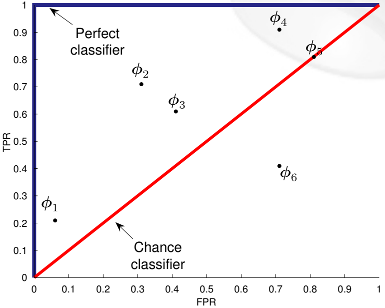


- ϕ_1: *k*NN
- ϕ_2: Neural network
- ϕ_3: Naive Bayes
- ϕ_4: SVM
- ϕ_5: Linear regression
- ϕ_6: Decision tree

Receiver Operating Characteristics (ROC)

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.

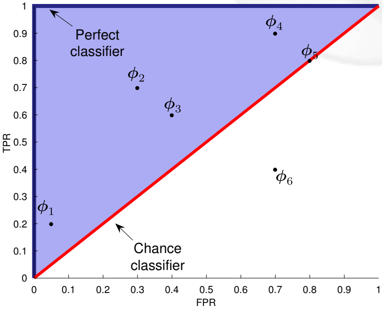


- ϕ_1: *k*NN
- ϕ_2: Neural network
- ϕ_3: Naive Bayes
- ϕ_4: SVM
- ϕ_5: Linear regression
- ϕ_6: Decision tree

Receiver Operating Characteristics (ROC)

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.

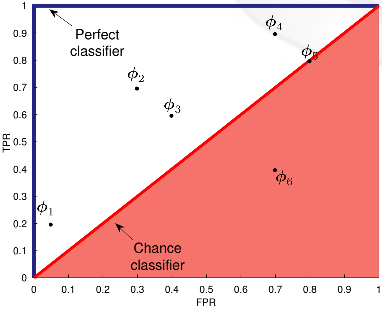


- ϕ_1: *k*NN
- ϕ_2: Neural network
- ϕ_3: Naive Bayes
- ϕ_4: SVM
- ϕ_5: Linear regression
- ϕ_6: Decision tree

Receiver Operating Characteristics (ROC)

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.

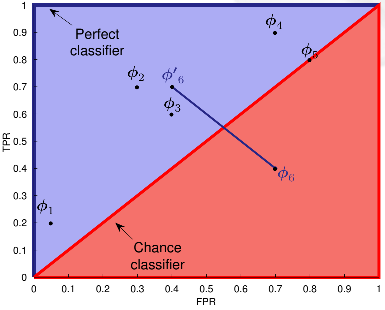


- ϕ_1: *k*NN
- ϕ_2: Neural network
- ϕ_3: Naive Bayes
- ϕ_4: SVM
- ϕ_5: Linear regression
- ϕ_6: Decision tree

Receiver Operating Characteristics (ROC)

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.



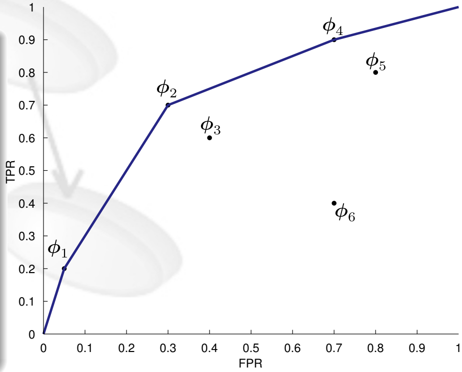
If we invertrevert the class assignation in ϕ_6 a classifier better than chance (ϕ_6') is obtained

Receiver Operating Characteristics (ROC)

ROC Convex Hull (ROCCH)

Minimal set of points for a given data set in the ROC space that meets:

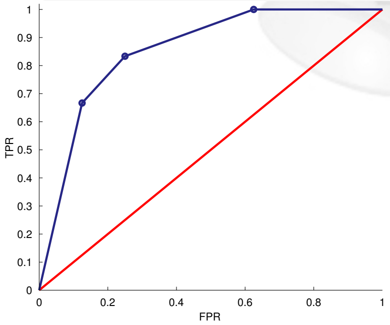
- Linear interpolation used between adjacent points
- No point lies above the final curve
- Segment connecting any point in the original set is equal or below the curve



Receiver Operating Characteristics (ROC)

ROC Curve

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

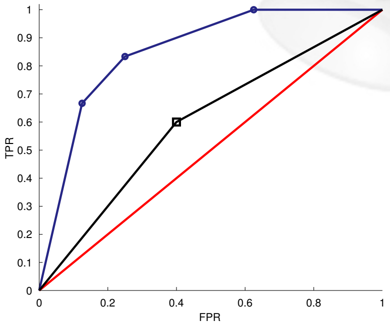


$p(c\|\mathbf{x})$	$T = 0{,}2$	$T = 0{,}5$	$T = 0{,}8$	C
0,99	c^+	c^+	c^+	c^+
0,90	c^+	c^+	c^+	c^+
0,85	c^+	c^+	c^+	c^+
0,80	c^+	c^+	c^+	c^-
0,78	c^+	c^+	c^-	c^+
0,70	c^+	c^+	c^-	c^-
0,60	c^+	c^+	c^-	c^+
0,45	c^+	c^-	c^-	c^-
0,40	c^+	c^-	c^-	c^-
0,30	c^+	c^-	c^-	c^-
0,20	c^+	c^-	c^-	c^+
0,15	c^-	c^-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

Receiver Operating Characteristics (ROC)

ROC Curve

For a crisp classifier a ROC curve can be obtained by interpolation from a single point



$p(c\|\mathbf{x})$	$T = 0{,}2$	$T = 0{,}5$	$T = 0{,}8$	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)

ROC Curve

- Insensitive to skew class distribution
- Insensitive to misclassification cost

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)

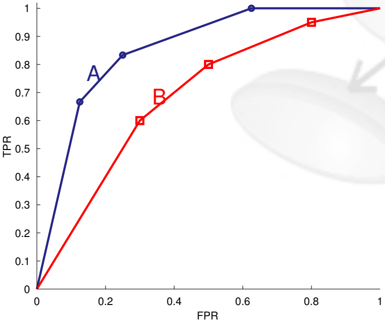
ROC Curve

- Insensitive to skew class distribution
- Insensitive to misclassification cost

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

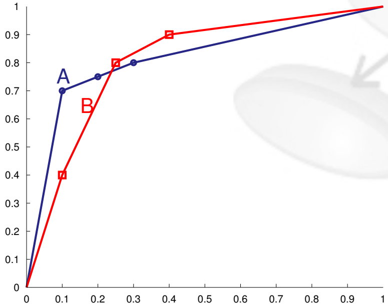
Receiver Operating Characteristics (ROC)



Dominance

- *A* dominates *B* throughout all the range of *T*
- *A* has a better predictive performance over any condition of cost and class distribution

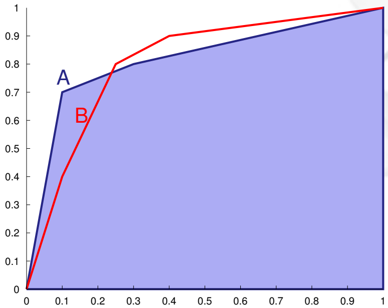
Receiver Operating Characteristics (ROC)



No-Dominance

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

Receiver Operating Characteristics (ROC)



Area Under ROC Curve

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

Generalization to Multilabel-Class

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

		Prediction					
		c_1	c_2	c_3	...	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	...	FN_{1n}	P_1
	c_2	FN_{21}	TP_2	FN_{23}	...	FN_{2n}	P_2
	c_3	FN_{31}	FN_{32}	TP_3	...	FN_{3n}	P_3
	...	...	...	...	...	...	...
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	...	TP_n	P_n
	Total	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	...	$\hat{P}_n$	

c_1 vs. All ($score_1$)

- TP
- TN
- FN
- FP

Generalization to Multilabel-Class

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

		Prediction					
		c_1	c_2	c_3	...	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	...	FN_{1n}	P_1
	c_2	FN_{21}	TP_2	FN_{23}	...	FN_{2n}	P_2
	c_3	FN_{31}	FN_{32}	TP_3	...	FN_{3n}	P_3
	...	...	...	...	...	...	...
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	...	TP_n	P_n
	Total	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	...	$\hat{P}_n$	

c_1 vs. All ($score_1$)

- *TP*
- *TN*
- *FN*
- *FP*

Generalization to Multilabel-Class

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

		Prediction					
		c_1	c_2	c_3	...	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	...	FN_{1n}	P_1
	c_2	FN_{21}	TP_2	FN_{23}	...	FN_{2n}	P_2
	c_3	FN_{31}	FN_{32}	TP_3	...	FN_{3n}	P_3
	...	...	...	...	...	...	...
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	...	TP_n	P_n
	Total	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	...	$\hat{P}_n$	

c_1 vs. All ($score_1$)

- TP
- TN
- FN
- FP

Generalization to Multilabel-Class

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

		Prediction					
		c_1	c_2	c_3	...	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	...	FN_{1n}	P_1
	c_2	FN_{21}	TP_2	FN_{23}	...	FN_{2n}	P_2
	c_3	FN_{31}	FN_{32}	TP_3	...	FN_{3n}	P_3
	...	...	...	...	...	...	...
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	...	TP_n	P_n
	Total	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	...	$\hat{P}_n$	

c_1 vs. All ($score_1$)

- TP
- TN
- FN
- FP

Generalization to Multilabel-Class

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

		Prediction					
		c_1	c_2	c_3	$\ldots$	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	$\ldots$	FN_{1n}	P_1
	c_2	FN_{21}	TP_2	FN_{23}	$\ldots$	FN_{2n}	P_2
	c_3	FN_{31}	FN_{32}	TP_3	$\ldots$	FN_{3n}	P_3
	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	$\ldots$	TP_n	P_n
	Total	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	$\ldots$	$\hat{P}_n$	

c_1 vs. All ($score_1$)

- *TP*
- *TN*
- *FN*
- *FP*

Generalization to Multilabel-Class

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

		Prediction					
		c_1	c_2	c_3	$\dots$	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	$\dots$	FN_{1n}	P_1
	c_2	FN_{21}	TP_2	FN_{23}	$\dots$	FN_{2n}	P_2
	c_3	FN_{31}	FN_{32}	TP_3	$\dots$	FN_{3n}	P_3
	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	$\dots$	TP_n	P_n
	Total	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	$\dots$	$\hat{P}_n$	

c_1 vs. All ($score_1$)

- *TP*
- *TN*
- *FN*
- *FP*

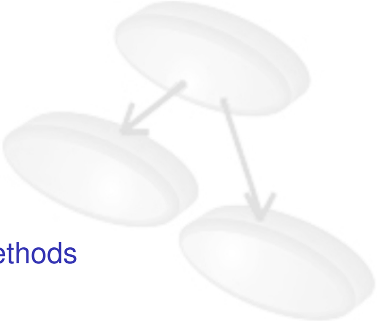
$$score_{TOT} = \sum_{i=1}^{n} score_i \cdot p(c_i)$$

Scores

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
- etc.

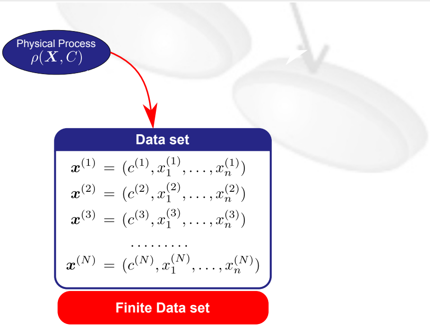
Outline of the Tutorial



Introduction

Estimation

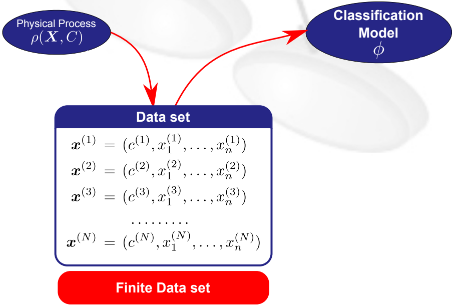
- Select a score to measure the quality
- Calculate the true value of the score
- Limited information is available



Introduction

Estimation

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



Introduction

Estimation

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

Physical Process
$\rho(\boldsymbol{X}, C)$

Classification Model
ϕ

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

Quality Measures

Error
Recall
Precision
....

Introduction

Estimation

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

Physical Process
$\rho(\boldsymbol{X}, C)$

Classification Model
ϕ

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

Quality Measures

Error
Recall
Precision
....

Random Variables

Introduction

True Value - ϵ_N

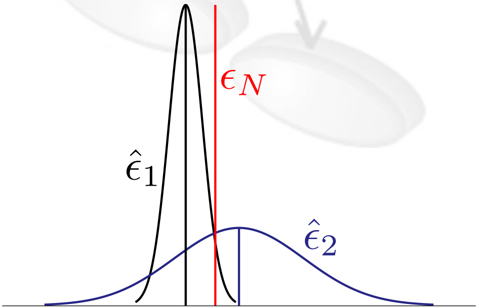
Expected value of the score for a set of N data samples sampled from $\rho(C, \boldsymbol{X})$

Introduction

True Value - ϵ_N

Expected value of the score for a set of N data samples sampled from $\rho(C, \boldsymbol{X})$

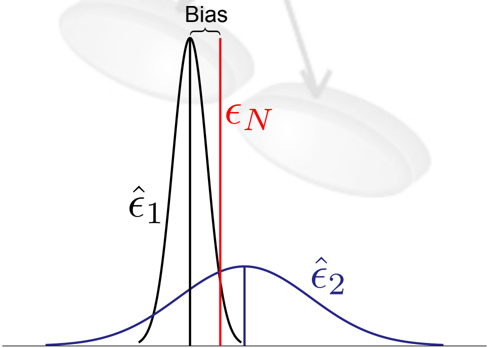
$\rho(C, \boldsymbol{X})$ unknown $\rightarrow$ Point estimation of the score ($\hat{\epsilon}$)



Introduction

Bias

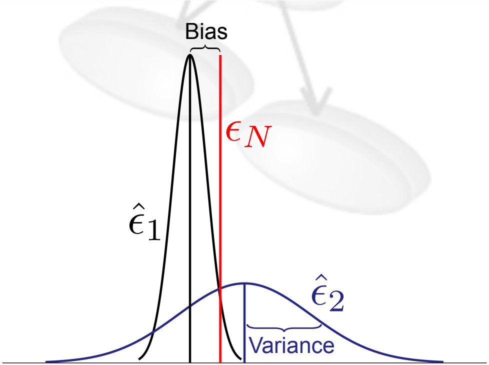
Difference between the estimation of the score and its true value: $E_\rho(\hat{\epsilon} - \epsilon_N)$



Introduction

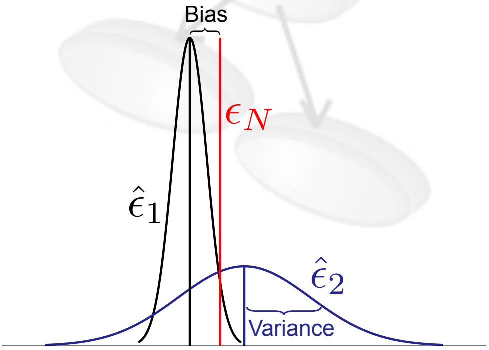
Variance

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



Introduction

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



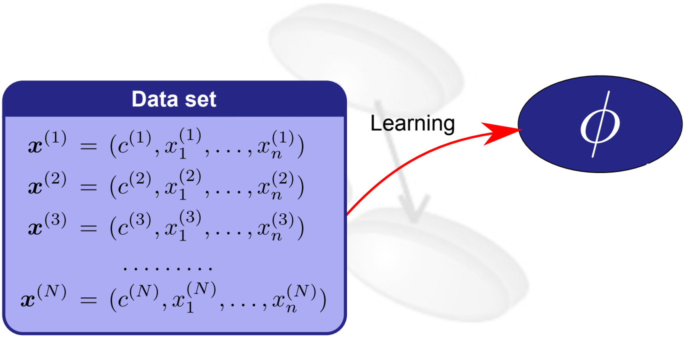
Introduction

Data set

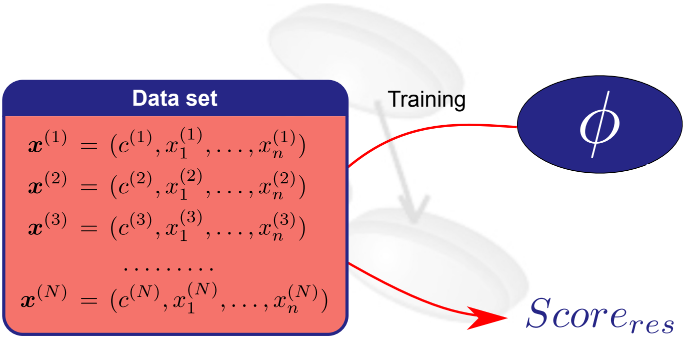
$$\begin{aligned}
\boldsymbol{x}^{(1)} &= (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)}) \\
\boldsymbol{x}^{(2)} &= (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)}) \\
\boldsymbol{x}^{(3)} &= (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)}) \\
& \cdots\cdots\cdots \\
\boldsymbol{x}^{(N)} &= (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})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

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

Resubstitution



Resubstitution



Resubstitution

Classification Error Estimation

- 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

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

Data set - Training

$$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$$
$$\ldots\ldots\ldots$$
$$\boldsymbol{x}^{(N_1)} = (c^{(N_1)}, x_1^{(N_1)}, \ldots, x_n^{(N_1)})$$

Data set - Test

$$\boldsymbol{x}^{(N_1+1)} = (c^{(N_1+1)}, x_1^{(N_1+1)}, \ldots, x_n^{(N_1+1)})$$
$$\ldots\ldots\ldots$$
$$\boldsymbol{x}^{(N_2)} = (c^{(N_2)}, x_1^{(N_2)}, \ldots, x_n^{(N_2)})$$

Hold-Out

Data set

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

Data set - Training

$$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$$
$$\ldots\ldots\ldots$$
$$\boldsymbol{x}^{(N_1)} = (c^{(N_1)}, x_1^{(N_1)}, \ldots, x_n^{(N_1)})$$

$$\xrightarrow{\text{Training}} \phi$$

Data set - Test

$$\boldsymbol{x}^{(N_1+1)} = (c^{(N_1+1)}, x_1^{(N_1+1)}, \ldots, x_n^{(N_1+1)})$$
$$\ldots\ldots\ldots$$
$$\boldsymbol{x}^{(N_2)} = (c^{(N_2)}, x_1^{(N_2)}, \ldots, x_n^{(N_2)})$$

Hold-Out

Data set

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

Data set - Training

$$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$$

$$\ldots\ldots\ldots$$

$$\boldsymbol{x}^{(N_1)} = (c^{(N_1)}, x_1^{(N_1)}, \ldots, x_n^{(N_1)})$$

Data set - Test

$$\boldsymbol{x}^{(N_1+1)} = (c^{(N_1+1)}, x_1^{(N_1+1)}, \ldots, x_n^{(N_1+1)})$$

$$\ldots\ldots\ldots$$

$$\boldsymbol{x}^{(N_2)} = (c^{(N_2)}, x_1^{(N_2)}, \ldots, x_n^{(N_2)})$$

ϕ

Test

$Score_{ho}$

Hold-Out

Classification Error Estimation

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

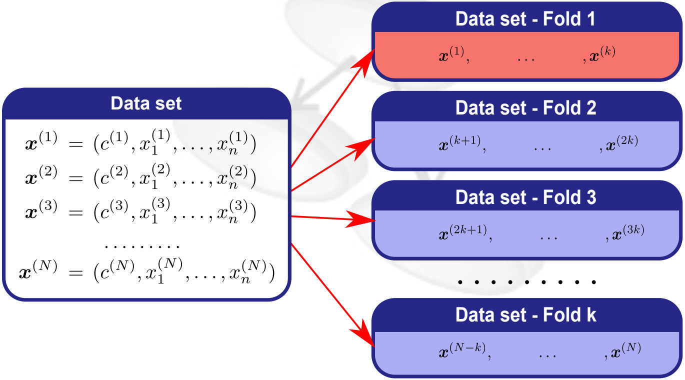
Repeated Hold-Out

- Repeat the hold-out t-times
- Simple average over results

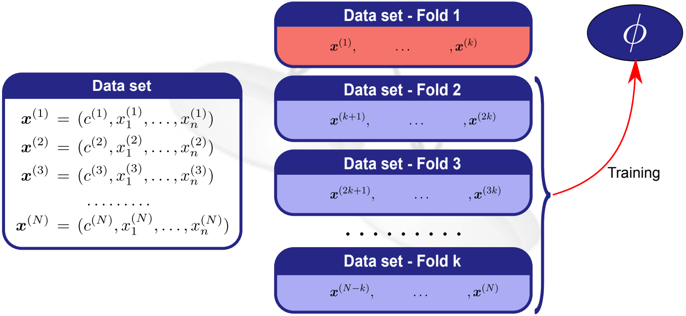
Classification Error Estimation

- Same bias as standard hold-out
- Reduces the variance with respect to the hold-out

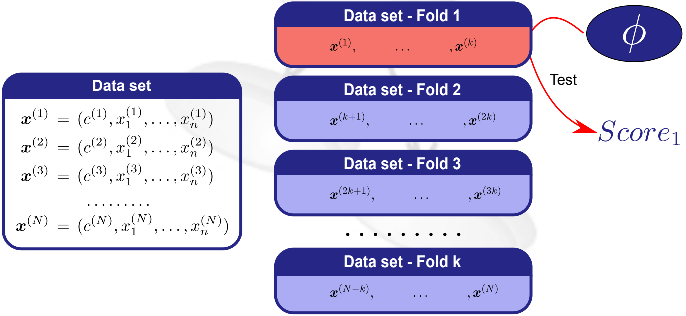
k-Fold Cross-Validation



k-Fold Cross-Validation



k-Fold Cross-Validation



k-Fold Cross-Validation

Data set

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

Data set - Fold 1

$\boldsymbol{x}^{(1)}, \quad \ldots \quad , \boldsymbol{x}^{(k)}$

Data set - Fold 2

$\boldsymbol{x}^{(k+1)}, \quad \ldots \quad , \boldsymbol{x}^{(2k)}$

Data set - Fold 3

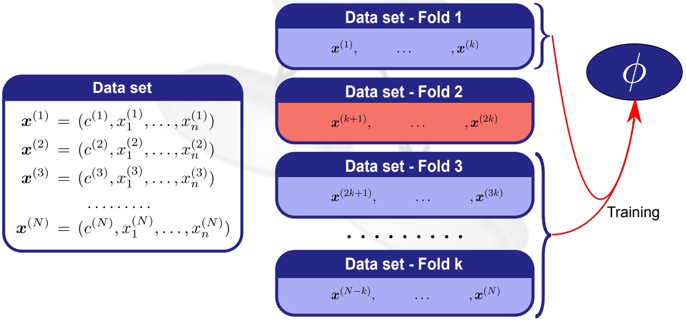
$\boldsymbol{x}^{(2k+1)}, \quad \ldots \quad , \boldsymbol{x}^{(3k)}$

. . . .

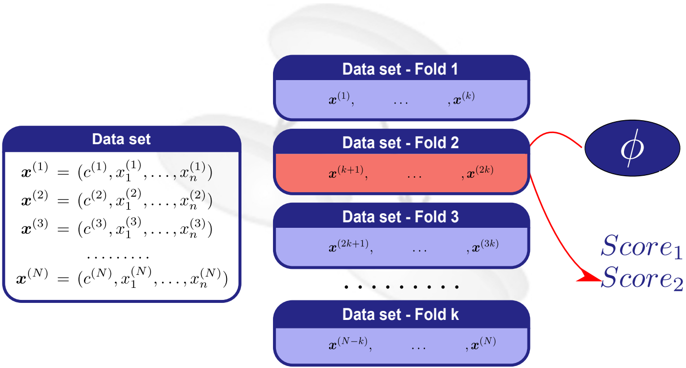
Data set - Fold k

$\boldsymbol{x}^{(N-k)}, \quad \ldots \quad , \boldsymbol{x}^{(N)}$

k-Fold Cross-Validation



k-Fold Cross-Validation



k-Fold Cross-Validation

Data set

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

Data set - Fold 1

$\boldsymbol{x}^{(1)}, \quad \ldots \quad , \boldsymbol{x}^{(k)}$ — $Score_1$

Data set - Fold 2

$\boldsymbol{x}^{(k+1)}, \quad \ldots \quad , \boldsymbol{x}^{(2k)}$ — $Score_2$

Data set - Fold 3

$\boldsymbol{x}^{(2k+1)}, \quad \ldots \quad , \boldsymbol{x}^{(3k)}$ — $Score_3$

$\ldots\ldots\ldots \qquad \ldots$

Data set - Fold k

$\boldsymbol{x}^{(N-k)}, \quad \ldots \quad , \boldsymbol{x}^{(N)}$ — $Score_k$

$Score_{cv}$

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 $\rightarrow$ more unstable
- Higher computational cost

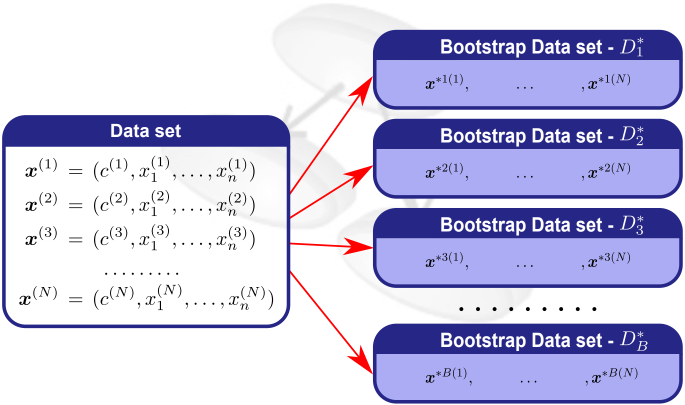
Repeated k-Fold cross-validation

- Similar to repeated hold-out:
  - Repeat cross-validation t-times
  - Simple average over results

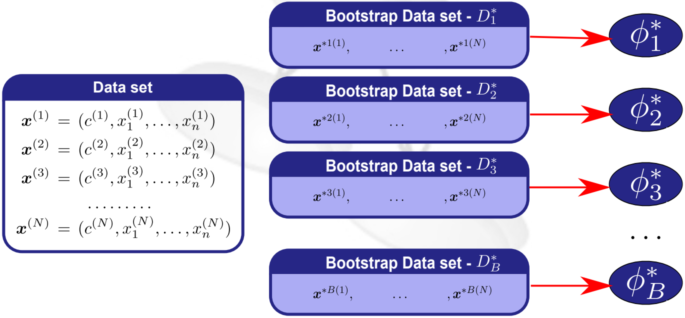
Classification Error Estimation

- Same bias as standard k-fold cross-validation
- Reduces the variance with respect k-fold cross-validation

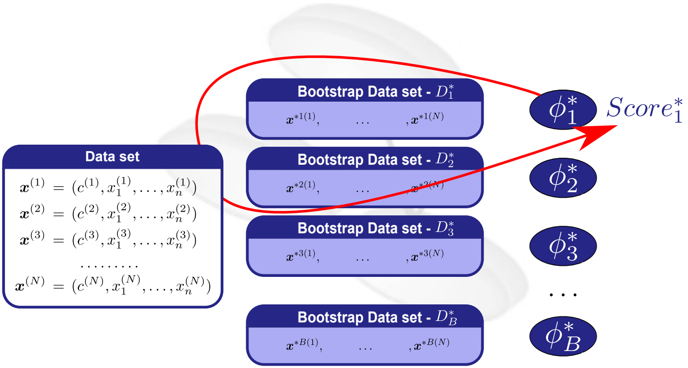
Bootstrap


Data set
$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$
$\boldsymbol{x}^{(2)} = (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)})$
$\boldsymbol{x}^{(3)} = (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)})$
....
$\boldsymbol{x}^{(N)} = (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})$
Bootstrap Data set - D_1^*
$\boldsymbol{x}^{*1(1)}, \ldots, \boldsymbol{x}^{*1(N)}$
Bootstrap Data set - D_2^*
$\boldsymbol{x}^{*2(1)}, \ldots, \boldsymbol{x}^{*2(N)}$
Bootstrap Data set - D_3^*
$\boldsymbol{x}^{*3(1)}, \ldots, \boldsymbol{x}^{*3(N)}$
....
Bootstrap Data set - D_B^*
$\boldsymbol{x}^{*B(1)}, \ldots, \boldsymbol{x}^{*B(N)}$

Bootstrap


Data set
$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$
$\boldsymbol{x}^{(2)} = (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)})$
$\boldsymbol{x}^{(3)} = (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)})$
....
$\boldsymbol{x}^{(N)} = (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})$
Bootstrap Data set - D_1^*
$\boldsymbol{x}^{*1(1)}, \quad \ldots \quad , \boldsymbol{x}^{*1(N)}$
Bootstrap Data set - D_2^*
$\boldsymbol{x}^{*2(1)}, \quad \ldots \quad , \boldsymbol{x}^{*2(N)}$
Bootstrap Data set - D_3^*
$\boldsymbol{x}^{*3(1)}, \quad \ldots \quad , \boldsymbol{x}^{*3(N)}$
Bootstrap Data set - D_B^*
$\boldsymbol{x}^{*B(1)}, \quad \ldots \quad , \boldsymbol{x}^{*B(N)}$
ϕ_1^*
ϕ_2^*
ϕ_3^*
...
ϕ_B^*

Bootstrap


Data set
$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$
$\boldsymbol{x}^{(2)} = (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)})$
$\boldsymbol{x}^{(3)} = (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)})$
....
$\boldsymbol{x}^{(N)} = (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})$
Bootstrap Data set - D_1^*
$\boldsymbol{x}^{*1(1)}, \ldots, \boldsymbol{x}^{*1(N)}$
Bootstrap Data set - D_2^*
$\boldsymbol{x}^{*2(1)}, \ldots, \boldsymbol{x}^{*2(N)}$
Bootstrap Data set - D_3^*
$\boldsymbol{x}^{*3(1)}, \ldots, \boldsymbol{x}^{*3(N)}$
Bootstrap Data set - D_B^*
$\boldsymbol{x}^{*B(1)}, \ldots, \boldsymbol{x}^{*B(N)}$
ϕ_1^*
$Score_1^*$
ϕ_2^*
ϕ_3^*
$\ldots$
ϕ_B^*

Bootstrap

Data set

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

Bootstrap Data set - D_1^*

$\boldsymbol{x}^{*1(1)}, \quad \ldots \quad, \boldsymbol{x}^{*1(N)}$

ϕ_1^* $Score_1^*$

Bootstrap Data set - D_2^*

$\boldsymbol{x}^{*2(1)}, \quad \ldots \quad, \boldsymbol{x}^{*2(N)}$

ϕ_2^* $Score_2^*$

Bootstrap Data set - D_3^*

$\boldsymbol{x}^{*3(1)}, \quad \ldots \quad, \boldsymbol{x}^{*3(N)}$

ϕ_3^* $Score_3^*$

Bootstrap Data set - D_B^*

$\boldsymbol{x}^{*B(1)}, \quad \ldots \quad, \boldsymbol{x}^{*B(N)}$

ϕ_B^* $Score_B^*$

$Score_{boot}$

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

Improvement: Leaving-one-out bootstrap

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 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

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
- $\hat{\epsilon}_{ho-N} = \phi$ learned on hold-out learning set but tested on D.

Improving the Estimation

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

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

Improvement

- $Bias_{\hat{\epsilon}_{ho}} \approx Cons_0 \frac{N_2}{N_1 \cdot N}$
- $Bias_{\hat{\epsilon}^{+}_{ho}} \approx Cons_1 \frac{N_2}{N_1 \cdot N^2}$

Improving the Estimation

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

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

Improvement

- $Bias_{\hat{\epsilon}_{cv}} \approx Cons_0 \frac{1}{(k-1)\cdot N}$
- $Bias_{\hat{\epsilon}_{cv}^{+}} \approx Cons_1 \frac{1}{(k-1)\cdot N^2}$

Improving the Estimation

0.632 Bootstrap ($\hat{\epsilon}^{.632}_{boot}$)

$$\hat{\epsilon}^{.632}_{boot} = 0.368\hat{\epsilon}_{res} + 0.632\hat{\epsilon}_{loo-boot}$$

Improvement

- Tries to balance optimism (resubstitution) and pessimism (loo-bootstrap)
- Works well with “light-fitting” classifiers
- With overfitting classifiers $\hat{\epsilon}^{.632}_{boot}$ is still too optimistic

Improving the Estimation

0.632+ Bootstrap ($\hat{\epsilon}_{boot}^{.632+}$) - *(Efron & Tibshirani, 1997)*

- Correct bias when there is great amount of overfitting
- Based on the non-information error rate (γ):

$$\hat{\gamma} = \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(c_i, \phi_{\mathbf{x}}(\mathbf{x}_j))/N^2$$

- Uses the relative overfitting to correct the bias:

$$\hat{R} = \frac{\hat{\epsilon}_{loo-boot} - \hat{\epsilon}_{res}}{\hat{\gamma} - \hat{\epsilon}_{res}}$$

Improving the Estimation

0.632+ Bootstrap ($\hat{\epsilon}_{boot}^{.632+}$) - *(Efron & Tibshirani, 1997)*

$$\hat{\epsilon}_{boot}^{.632} = (1 - \hat{w})\hat{\epsilon}_{res} + \hat{w}\hat{\epsilon}_{loo-boot}$$

- $\hat{w} = \frac{0.632}{1-0.638\hat{R}}$
- $\hat{\gamma} = \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(c_i, \phi_{\mathbf{x}}(\mathbf{x}_j))/N^2$
- $\hat{R} = \frac{\hat{\epsilon}_{loo-boot} - \hat{\epsilon}_{res}}{\hat{\gamma} - \hat{\epsilon}_{res}}$

Estimation Methods

- 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
- . . .

Estimation Methods

- 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

Estimation Methods

- Which estimation method is better?

Small Data Sets

- Bootstrap and repeated cross-validation may not be informative
- Permutation test *(Ojala & Garriga, 2010)*:
  - Can be used to ensure the validity of the estimation
- Confidence intervals *(Isaksson et al., 2008)*:
  - May provide more reliable information about the estimation

Outline of the Tutorial

Motivation

Basic Concepts

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

Hypothesis Testing

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	$\surd$	Type II error (β)
Decision: REJECT	Type I error (α)	$\surd$

Hypothesis Testing: An Example

A Simple Hypothesis Test

- A natural process is given in nature that follows a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$
- 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 $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Hypothesis Testing: An Example

Accept and Reject Regions

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

$$A.R. = \{(x_1, \dots, x_n) | \overline{X} > 55\}$$
$$R.R. = \{(x_1, \dots, x_n) | \overline{X} \leq 55\}$$

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

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

Hypothesis Testing: An Example

Accept and Reject Regions

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

$$0{,}1 = \alpha = P_{H_0}(\overline{X} \in R.R.) = P_{H_0}(\overline{X} \leq 51)$$

$$0{,}05 = \alpha = P_{H_0}(\overline{X} \in R.R.) = P_{H_0}(\overline{X} \leq 50{,}3)$$

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

$$p\text{-value} = P_{H_0}(\overline{X} \leq \overline{x})$$

Hypothesis Testing: Remarks

Power: $(1-\beta)$

- 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 $(1-\beta)$
- A good hypothesis test: given an α the test maximises the power $(1-\beta)$

Parametric test vs non-parametric test

Hypothesis Testing in Supervised Classification

Scenarios

- Two classifiers (algorithms) vs More than two
- One dataset vs More than one dataset
- Score
- Score estimation method known vs unknown
- The classifiers are trained and tested in the same datasets
- .....

Testing Two Algorithms in a Dataset

The General Approach

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

Testing Two Algorithms in a Dataset

The General Approach

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

$$\begin{cases} H_0 : \text{algorithm } \psi \text{ has the same average score value as} \\ \qquad\quad \text{algorithm } \psi' \text{ in } p(\mathbf{x}, c) \\ \\ H_1 : \text{they have different values} \end{cases}$$

Testing Two Algorithms in a Dataset

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 ψ_i^1, ψ_i^2 for $i = 1, \ldots, n$
3. For each classifier obtain enough i.i.d. samples $\{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_N, c_N)\}$ from $p(\mathbf{x}, c)$
4. For each data set calculate the error of each algorithm in the test set

$$\epsilon_i^1 = \frac{1}{N}\sum_{j=1}^{N} error_i^1(\mathbf{x}_j) \qquad \epsilon_i^2 = \frac{1}{N}\sum_{j=1}^{N} error_i^2(\mathbf{x}_j)$$

5. Calculate the average values over the n training datasets:

$$\bar{\epsilon}^1 = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i^1 \qquad \bar{\epsilon}^2 = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i^2$$

Testing Two Algorithms in a Dataset

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

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

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

- Therefore, under the null hypothesis:

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

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

Testing Two Algorithms in a Dataset

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

Testing Two Algorithms in a Dataset

McNemar Test (non-parametric)

- Compare two classifiers in a dataset after a hold-out process
- It is a paired non-parametric test

	ψ^2 error	ψ^2 ok
ψ^1 error	n_{00}	n_{01}
ψ^1 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 dist. can be used

Testing Two Algorithms in a Dataset

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 $\overline{p}$ the average. When it is assumed that $\hat{p}_i$ are Gaussian and independent, under the null

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

  follows a t student distribution with $n-1$ degree 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)

Testing Two Algorithms in a Dataset

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

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

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

  - Conservative Z

Testing Two Algorithms in a Dataset

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

Testing Two Algorithms in a Dataset

5x2 fold cross-validation (Dietterich 1998, Alpaydin 1999)

- Each cross-validation process has independent training and testing datasets
- 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 Approaches

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

Problems

- Commensurability
- Outlier susceptibility
- (t-test) Gaussian assumption

Testing Two Algorithms in Several Datasets

Wilcoxon Signed-Ranks Test

- 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 $N \leq 25$ there are tables with critical values
- For $N > 25$

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

Wilcoxon Signed-Ranks Test: Example

	ψ^1	ψ^2	diff	rank
Dataset1	0.763	0.598		
Dataset2	0.599	0.591		
Dataset3	0.954	0.971		
Dataset4	0.628	0.661		
Dataset5	0.882	0.888		
Dataset6	0.936	0.931		
Dataset7	0.661	0.668		
Dataset8	0.583	0.583		
Dataset9	0.775	0.838		
Dataset10	1.000	1.000		

Wilcoxon Signed-Ranks Test: Example

	ψ^1	ψ^2	diff	rank
Dataset1	0.763	0.598	-0.165	
Dataset2	0.599	0.591		
Dataset3	0.954	0.971		
Dataset4	0.628	0.661		
Dataset5	0.882	0.888		
Dataset6	0.936	0.931		
Dataset7	0.661	0.668		
Dataset8	0.583	0.583		
Dataset9	0.775	0.838		
Dataset10	1.000	1.000		

Wilcoxon Signed-Ranks Test: Example

	ψ^1	ψ^2	diff	rank
Dataset1	0.763	0.598	-0.165	
Dataset2	0.599	0.591	-0.008	
Dataset3	0.954	0.971		
Dataset4	0.628	0.661		
Dataset5	0.882	0.888		
Dataset6	0.936	0.931		
Dataset7	0.661	0.668		
Dataset8	0.583	0.583		
Dataset9	0.775	0.838		
Dataset10	1.000	1.000		

Wilcoxon Signed-Ranks Test: Example

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

Wilcoxon Signed-Ranks Test: Example

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

Wilcoxon Signed-Ranks Test: Example

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

Wilcoxon Signed-Ranks Test: Example

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

Wilcoxon Signed-Ranks Test: Example

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

Wilcoxon Signed-Ranks Test: Example

	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

Wilcoxon Signed-Ranks Test: Example

	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

$$R^+ =$$

Wilcoxon Signed-Ranks Test: Example

	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: Example

	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

$$R^+ = 34.5$$

Wilcoxon Signed-Ranks Test: Example

	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: Example

	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

$$R^+ = 34.5 \qquad R^- = 20.5$$

Wilcoxon Signed-Ranks Test: Example

	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

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

Wilcoxon Signed-Ranks Test: Example

	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

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

Testing Two Algorithms in Several Datasets

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

Testing Two Algorithms in Several Datasets

Signed 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
- This test does not make any assumptions
- It is weaker than Wilcoxon

Testing Several Algorithms in Several Datasets

Dataset (Demšar, 2006)

	ψ^1	ψ^2	ψ^3	ψ^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

Testing Several Algorithms in Several Datasets

Multiple Hypothesis Testing

- Testing all possible pairs of hypotheses $\mu_{\psi^i} = \mu_{\psi^j} \quad \forall \quad i, \; j$.
  Multiple hypothesis testing
- Testing the hypothesis $\mu_{\psi^1} = \mu_{\psi^2} = \ldots = \mu_{\psi^k}$

Testing Several Algorithms in Several Datasets

Multiple Hypothesis Testing

- Testing all possible pairs of hypotheses $\mu_{\psi^i} = \mu_{\psi^j} \ \forall \ i, \ j$.
  Multiple hypothesis testing
- Testing the hypothesis $\mu_{\psi^1} = \mu_{\psi^2} = \ldots = \mu_{\psi^k}$

Testing Several Algorithms in Several Datasets

Multiple Hypothesis Testing

- Testing all possible pairs of hypotheses $\mu_{\psi^i} = \mu_{\psi^j} \;\; \forall \;\; i, \; j.$ Multiple hypothesis testing
- Testing the hypothesis $\mu_{\psi^1} = \mu_{\psi^2} = \ldots = \mu_{\psi^k}$

ANOVA vs Friedman

- *Repeated measures* ANOVA: Assumes Gaussianity and sphericity
- Friedman: Non-parametric test

Testing Several Algorithms in Several Datasets

Freidman 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 ψ^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)

Testing Several Algorithms in Several Datasets

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

Testing Several Algorithms in Several Datasets

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

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

Testing Several Algorithms in Several Datasets

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

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

Testing Several Algorithms in Several Datasets

Iman & Davenport, 1980

- An improvement of 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

Testing Several Algorithms in Several Datasets

Post-hoc Tests

- Decision on the null hypothesis
- In case of rejection use of post-hoc tests to:
    1. Compare all pairs
    2. Compare all classifiers with a control

Testing Several Algorithms in Several Datasets

Multiple Hypothesis Testing

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

	H_0 TRUE	H_0 FALSE
Decision: ACCEPT	$\surd$	Type II error (β)
Decision: REJECT	Type I error (α)	$\surd$

Testing Several Algorithms in Several Datasets

Multiple Hypothesis Testing

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

	H_0 TRUE	H_0 FALSE
Decision: ACCEPT	$\surd$	Type II error (β)
Decision: REJECT	Type I error (α)	$\surd$

Testing Several Algorithms in Several Datasets

Multiple Hypothesis Testing

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

	H_0 TRUE	H_0 FALSE
Decision: ACCEPT	$\surd$	Type II error (β)
Decision: REJECT	Type I error (α)	$\surd$

- Family-wise error: Probability of rejecting at least one hypothesis assuming that ALL ARE TRUE

Testing Several Algorithms in Several Datasets

Multiple Hypothesis Testing

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

	H_0 TRUE	H_0 FALSE
Decision: ACCEPT	$\surd$	Type II error (β)
Decision: REJECT	Type I error (α)	$\surd$

- Family-wise error: Probability of rejecting at least one hypothesis assuming that ALL ARE TRUE
- False discovery rate

Testing Several Algorithms in Several Datasets

Multiple Hypothesis Testing

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

	H_0 TRUE	H_0 FALSE
Decision: ACCEPT	$\surd$	Type II error (β)
Decision: REJECT	Type I error (α)	$\surd$

- Family-wise error: Probability of rejecting at least one hypothesis assuming that ALL ARE TRUE
- False discovery rate

Testing Several Algorithms in Several Datasets

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:

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

and therefore

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

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

Testing Several Algorithms in Several Datasets

Comparing with a Control

- The statistic for comparing ψ^i and ψ^j is:

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

Bonferroni-Dunn Test

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

$$\frac{\alpha}{k-1}$$

Testing Several Algorithms in Several Datasets

Comparing with a Control

- Methods based on ordered *p*-values
- The p-values are ordered $p_1 \leq p_2 \leq \ldots \leq p_{k-1}$

Holm Method

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

Testing Several Algorithms in Several Datasets

Friedman Test: Example ($\alpha = 0.05$)

	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

Testing Several Algorithms in Several Datasets

Friedman Test: Example ($\alpha = 0.05$)

	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

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

Testing Several Algorithms in Several Datasets

Friedman Test: Example ($\alpha = 0.05$)

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

	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

Testing Several Algorithms in Several Datasets

Friedman Test: Example ($\alpha = 0.05$)

	z	p-value
z_{12}	0.3354	0.259
z_{13}	2.1569	0.031
z_{14}	0.7915	0.125
z_{23}	1.9843	0.042
z_{24}	0.4561	0.221
z_{34}	-2.7781	0.009

Testing Several Algorithms in Several Datasets

Friedman Test: Example ($\alpha = 0.05$)

	z	p-value	Bonferroni ($\alpha/6$)
z_{12}	0.3354	0.259	0.008
z_{13}	2.1569	0.031	0.008
z_{14}	0.7915	0.125	0.008
z_{23}	1.9843	0.042	0.008
z_{24}	0.4561	0.221	0.008
z_{34}	-2.7781	0.007	0.008

Testing Several Algorithms in Several Datasets

Friedman Test: Example ($\alpha = 0.05$)

	z	p-value	Bonferroni ($\alpha/6$)
z_{12}	0.3354	0.259	0.008
z_{13}	2.1569	0.031	0.008
z_{14}	0.7915	0.125	0.008
z_{23}	1.9843	0.042	0.008
z_{24}	0.4561	0.221	0.008
z_{34}	-2.7781	0.007	0.008

Testing Several Algorithms in Several Datasets

Friedman Test: Example ($\alpha = 0.05$)

	z	p-value	Bonferroni ($\alpha/6$)	Holm ($\alpha/(7-i)$)
z_{12}	0.3354	0.259	0.008	
z_{13}	2.1569	0.031	0.008	
z_{14}	0.7915	0.125	0.008	
z_{23}	1.9843	0.009	0.008	
z_{24}	0.4561	0.221	0.008	
z_{34}	-2.7781	0.007	0.008	

Testing Several Algorithms in Several Datasets

Friedman Test: Example ($\alpha = 0.05$)

	z	p-value	Bonferroni ($\alpha/6$)	Holm ($\alpha/(7-i)$)
z_{12}	0.3354	0.259	0.008	
z_{13}	2.1569	0.031	0.008	
z_{14}	0.7915	0.125	0.008	
z_{23}	1.9843	0.009	0.008	
z_{24}	0.4561	0.221	0.008	
z_{34}	-2.7781	0.007	0.008	0.008

Testing Several Algorithms in Several Datasets

Friedman Test: Example ($\alpha = 0.05$)

	z	p-value	Bonferroni ($\alpha/6$)	Holm ($\alpha/(7-i)$)
z_{12}	0.3354	0.259	0.008	
z_{13}	2.1569	0.031	0.008	
z_{14}	0.7915	0.125	0.008	
z_{23}	1.9843	0.009	0.008	0.010
z_{24}	0.4561	0.221	0.008	
z_{34}	-2.7781	0.007	0.008	0.008

Testing Several Algorithms in Several Datasets

Friedman Test: Example ($\alpha = 0.05$)

	z	p-value	Bonferroni ($\alpha/6$)	Holm ($\alpha/(7-i)$)
z_{12}	0.3354	0.259	0.008	
z_{13}	2.1569	0.031	0.008	0.012
z_{14}	0.7915	0.125	0.008	
z_{23}	1.9843	0.009	0.008	0.010
z_{24}	0.4561	0.221	0.008	
z_{34}	-2.7781	0.007	0.008	0.008

Testing Several Algorithms in Several Datasets

Friedman Test: Example ($\alpha = 0.05$)

	z	p-value	Bonferroni ($\alpha/6$)	Holm ($\alpha/(7-i)$)
z_{12}	0.3354	0.259	0.008	
z_{13}	2.1569	0.031	0.008	0.012
z_{14}	0.7915	0.125	0.008	
z_{23}	1.9843	0.009	0.008	0.010
z_{24}	0.4561	0.221	0.008	
z_{34}	-2.7781	0.007	0.008	0.008

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 $k = 1, \ldots, j$
- Reject all hypotheses i such that $p_i \leq \alpha/j$

Testing Several Algorithms in Several Datasets

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

Testing Several Algorithms in Several Datasets

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

$$C_1 \text{ better than } C_2 \quad \text{and} \quad C_2 \text{ better than } C_3$$

$$\text{and} \quad C_1 \text{ equal to } C_3$$

Testing Several Algorithms in Several Datasets

Shaffer Static Procedure

- It is a modification of Homl'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

Testing Several Algorithms in Several Datasets

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

Testing Several Algorithms in Several Datasets

Bregmann & Hommel

- More powerful alternative than Shaffer Dynamic Procedure
- Difficult implementation

Remarks

- Adjusted p-values

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 Signed test could be used

Conclusions

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: Shaffer Static method

An Idea for Future Work

- To consider the variability of the score in each classifier and dataset

Honest Evaluation of Classification Models

Jose A. Lozano, Guzmán Santafé, Iñaki Inza

Intelligent Systems Group
The University of the Basque Country

Asian Conference on Machine Learning (ACML'10)
November 8-10, 2010