Estimating classification performance

Guzmán Santafé

Spatial Statistics Group
Public University of Navarre

DATAI-UNAV, November 2022

Spatial Statistics
Public University of Navarre

upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

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

Data set

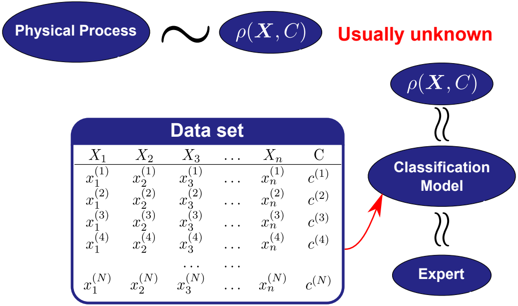
X_1	X_2	X_3	$\dots$	X_n	C
$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$\dots$	$x_n^{(1)}$	$c^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$	$\dots$	$x_n^{(2)}$	$c^{(2)}$
$x_1^{(3)}$	$x_2^{(3)}$	$x_3^{(3)}$	$\dots$	$x_n^{(3)}$	$c^{(3)}$
$x_1^{(4)}$	$x_2^{(4)}$	$x_3^{(4)}$	$\dots$	$x_n^{(4)}$	$c^{(4)}$
		$\dots$	$\dots$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

Expert

Supervised Classification

Learning from Experience

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

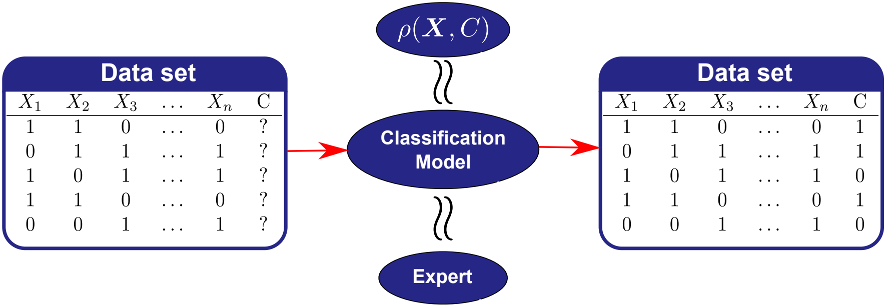


X_1	X_2	X_3	...	X_n	C
$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	...	$x_n^{(1)}$	$c^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$	...	$x_n^{(2)}$	$c^{(2)}$
$x_1^{(3)}$	$x_2^{(3)}$	$x_3^{(3)}$	...	$x_n^{(3)}$	$c^{(3)}$
$x_1^{(4)}$	$x_2^{(4)}$	$x_3^{(4)}$	...	$x_n^{(4)}$	$c^{(4)}$
		...	...		
$x_1^{(N)}$	$x_2^{(N)}$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

Supervised Classification

Classification Model

- Classifier labels new data (unknown class value)



Data set

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

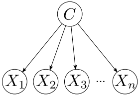
Data set

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

Motivation for Honest Evaluation

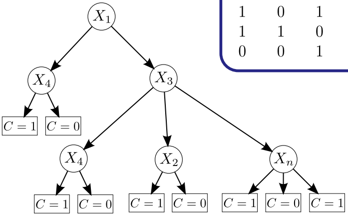
- Many classification paradigms

Naive Bayes

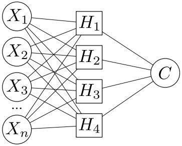


Data set

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



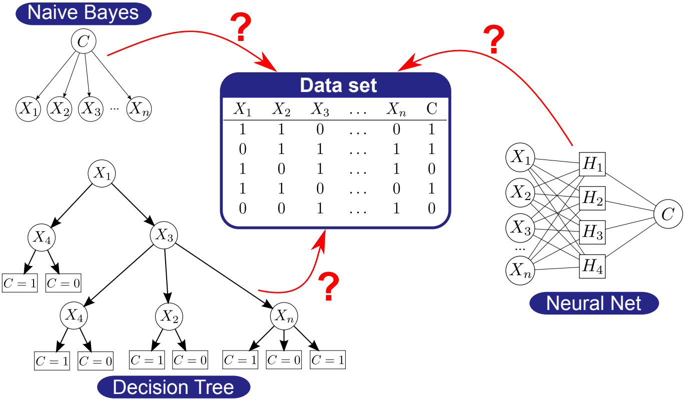
Decision Tree



Neural Net

Motivation for Honest Evaluation

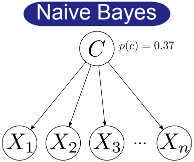
- Which is the best paradigm for a classification problem?



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

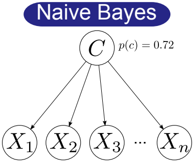
- Many parameter configurations



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

Data set

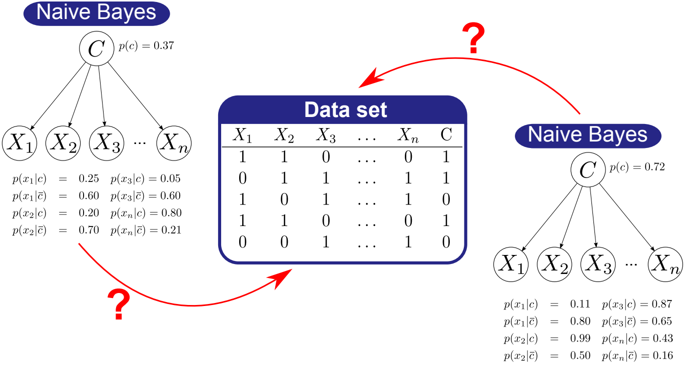
X_1	X_2	X_3	...	X_n	C
1	1	0	...	0	1
0	1	1	...	1	1
1	0	1	...	1	0
1	1	0	...	0	1
0	0	1	...	1	0



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

Motivation for Honest Evaluation

- Which is the best parameter configuration 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

Honest Evaluation

- Need to know the goodness of a classifier
- Methodology to evaluate classifiers

Evaluating classification performance

- Quality measures (Scores)
- Estimate value of a score (Estimation methods)
- Comparing different solutions (Statistical tests?)

Scores

Score

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

But ... 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 kinds of scores

Scores

Binary classification problem

- Non-balanced scores:
  - Accuracy/Classification error
  - Recall
  - Specificity
  - Precision
- Balanced scores:
  - Balanced accuracy
  - F-Score
  - *“ROC curve / AUC”*
  - Kappa coefficient

Scores

Multiclass classification problem

- Non-balanced scores:
  - Accuracy/Classification error
- Balanced scores:
  - Kappa coefficient
- It is possible to addapt scores from binary classification using O.vs.A approach

Confusion Matrix

Binary classification problem

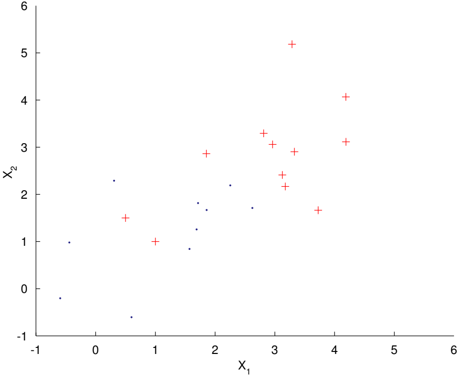
		Prediction		
		c^+	c^-	Total
Actual	c^+	TP	FP	N^+
	c^-	FN	TN	N^-
	Total	$\hat{N}^+$	$\hat{N}^-$	N

Confusion Matrix

Multiclass classification problem

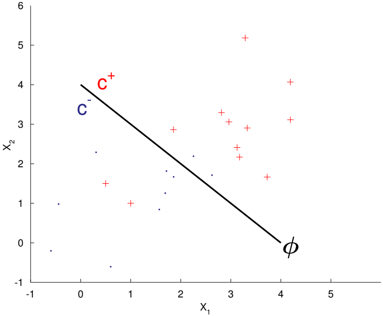
		Prediction					
		c_1	c_2	c_3	$\dots$	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	$\dots$	FN_{1n}	N_1
	c_2	FN_{21}	TP_2	FN_{23}	$\dots$	FN_{2n}	N_2
	c_3	FN_{31}	FN_{32}	TP_3	$\dots$	FN_{3n}	N_3
	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	$\dots$	TP_n	N_n
	Total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

Binary classification 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^-
...	...	...

Binary classification Problem - Example

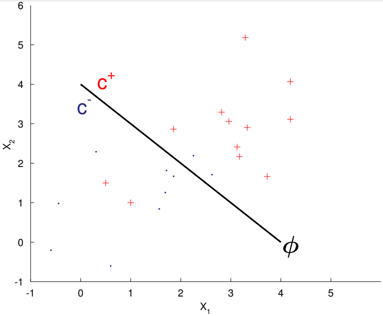


		Prediction c^+	Prediction c^-	Total
Actual	c^+	10	2	12
Actual	c^-	2	8	10
	Total	12	10	**22**

Accuracy/Classification Error

Definition

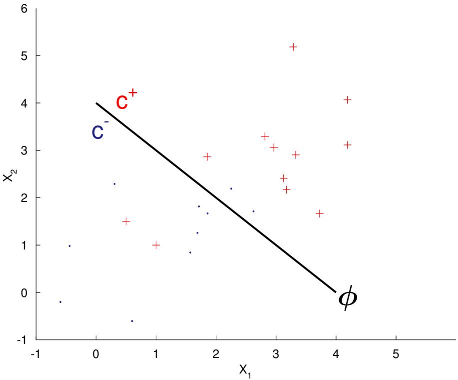
- Data samples classified correctly/incorrectly



		Prediction		
		c^+	c^-	Total
Actual	c^+	10	2	12
	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

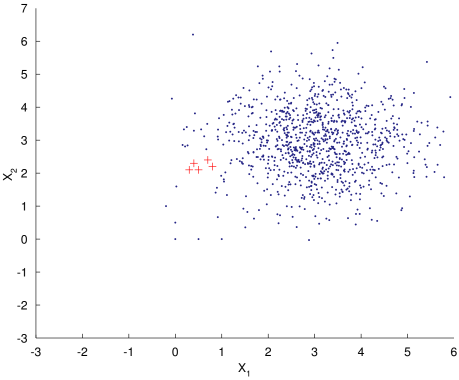


		Prediction c^+	Prediction c^-	Total
Actual	c^+	10	2	12
	c^-	2	8	10
	Total	12	10	**22**

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

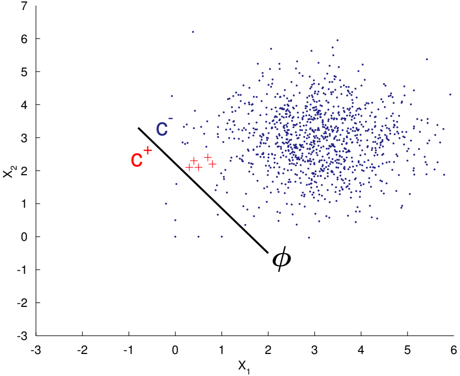
Low ϵ!!

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

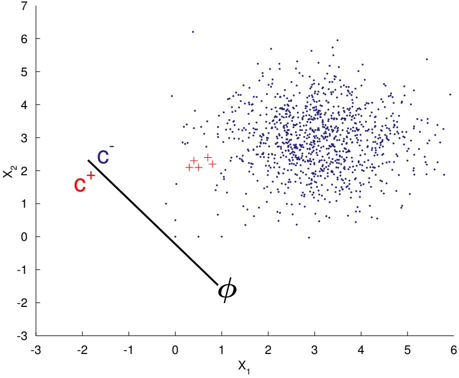


		Prediction c^+	Prediction c^-	Total
Actual	c^+	0	5	5
	c^-	7	993	1000
	Total	7	998	**1005**

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

Very low ϵ!!

Skew Data - Classification Error

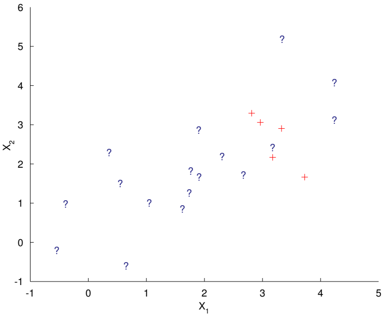


		Prediction		
		c^+	c^-	Total
Actual	c^+	0	5	5
	c^-	0	1000	1000
	Total	0	1005	**1005**

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

Better??

Positive Unlabeled Learning



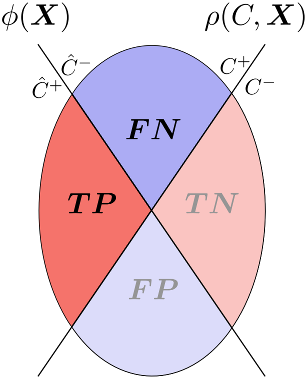
Positive Labeled Data

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

Recall

Definition

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

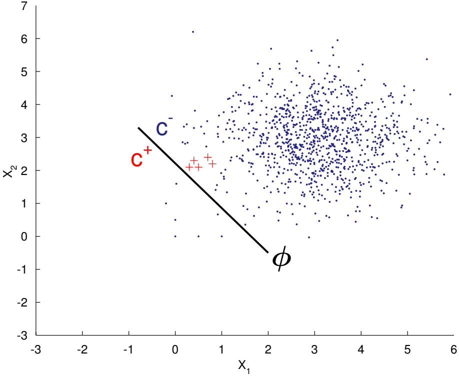


$$r(\phi) = \frac{TP}{TP + FN} = \frac{TP}{P}$$

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

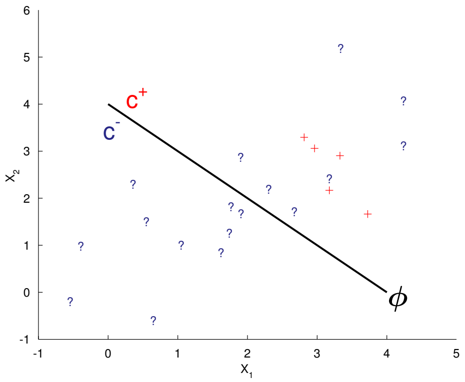


		Prediction c^+	Prediction c^-	Total
Actual	c^+	0	5	5
Actual	c^-	7	993	1000
	Total	7	998	**1005**

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

Very bad recall!!

Positive Unlabeled Learning - Recall



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

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

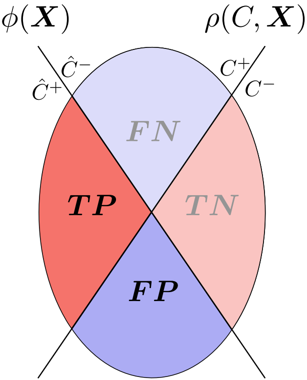
It is possible to calculate recall in positive-unlabeled problems

Precision

Definition

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

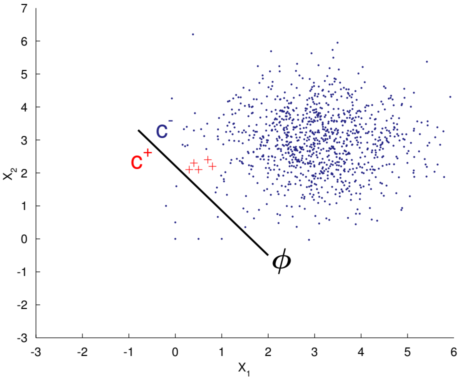
$$pr(\phi) = \frac{TP}{TP + FP} = \frac{TP}{\hat{P}}$$



Definition Based on Probabilities

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

Skew Data - Precision

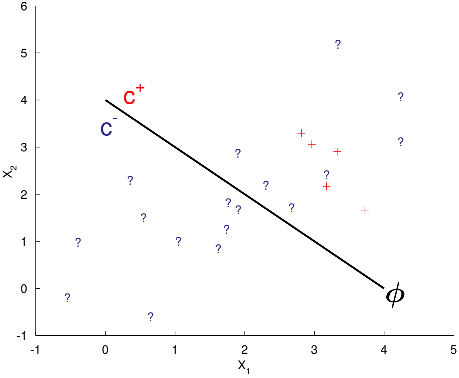


		Prediction c^+	Prediction c^-	Total
Actual	c^+	0	5	5
Actual	c^-	7	993	1000
	Total	7	998	**1005**

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

Very bad precision!!

Positive Unlabeled Learning - Precision

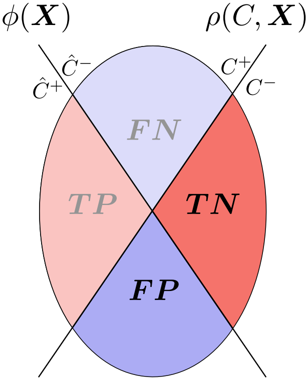


- Precision is not a good score for positive-unlabeled data samples
- Not all the positive samples are labeled

Specificity

Definition

- Fraction of negative class samples correctly identified
- $\mathit{Specificity} = 1 - \mathit{FalsePositiveRate}$

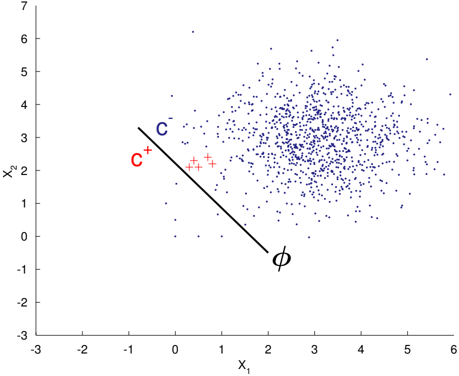


$$sp(\phi) = \frac{TN}{TN + FP} = \frac{TN}{N}$$

Definition Based on Probabilities

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

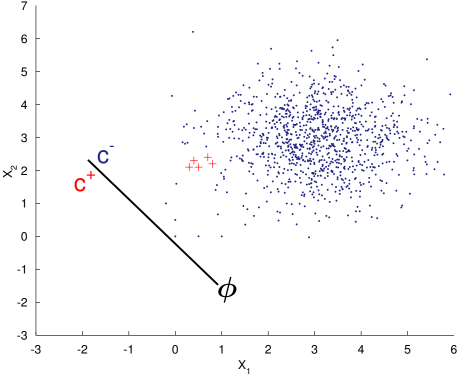
Skew Data - Specificity



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

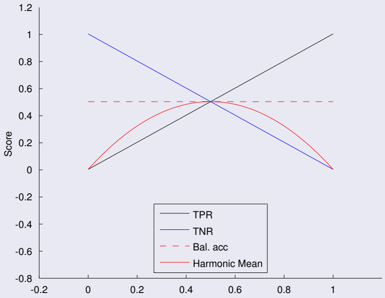
- $Bal.\ acc = \frac{1}{2}\left(\frac{0}{5} + \frac{993}{1000}\right) \approx 0.5$
- $Bal.\ \epsilon = \frac{1}{2}\left(\frac{7}{7} + \frac{5}{1000}\right) \approx 0.5$

Balanced Scores

- $F - Score = \frac{(\beta^2+1)\ Precision \cdot Recall}{\beta^2(Precision + Recall)}$
- $F_1 - Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \longrightarrow$ Harmonic Mean

Harmonic Mean

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



Balanced Scores

- Kappa coefficient: chance corrected proportion of correct classifications.

$$\kappa = \frac{Acc. - P_e}{1 - P_e}, \qquad \text{with } P_e = \frac{N^+}{N} \cdot \frac{\hat{N}^+}{N} + \frac{N^-}{N} \cdot \frac{\hat{N}^-}{N}$$

Skew Data

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

- $Acc = \frac{993}{1005} \approx 0.988$
- $P_e \frac{5}{1005} \cdot \frac{7}{1005} + \frac{1000}{1005} \cdot \frac{998}{1005} \approx 0.988$
- $k \approx 0$

Classification Cost

- All misclassifications cannot be equally considered

E.g. Medical Diagnosis Problem

It does not have the same cost 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 cost matrix to specify the associated cost:

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

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 cost matrix to specify the associated cost:

Prediction

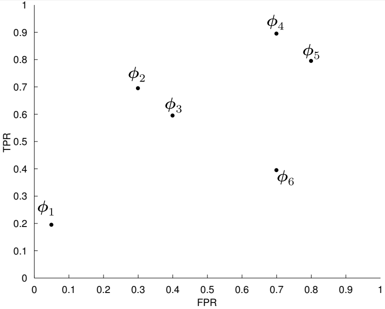
Actual		c^+	c^-
	c^+	$Cost_{TP}$	$Cost_{FN}$
	c^-	$Cost_{FP}$	$Cost_{TN}$

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* (1 − *especificity*) is plotted on the *X* axis.

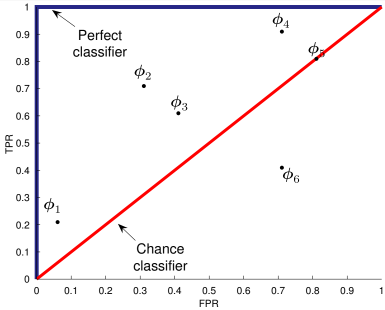


- ϕ_1: *k*NN
- ϕ_2: Neural network
- ϕ_3: Naive Bayes
- ϕ_4: SVM
- ϕ_5: Linear regression
- ϕ_6: Decision tree

Receiver Operating Characteristics (ROC)

ROC Space

Coordinate system used for visualizing classifiers performance where *TPR* is plotted on the *Y* axis and *FPR* (1 − *especificity*) is plotted on the *X* axis.

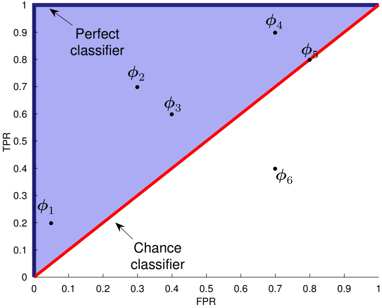


- ϕ_1: *k*NN
- ϕ_2: Neural network
- ϕ_3: Naive Bayes
- ϕ_4: SVM
- ϕ_5: Linear regression
- ϕ_6: Decision tree

Receiver Operating Characteristics (ROC)

ROC Space

Coordinate system used for visualizing classifiers performance where *TPR* is plotted on the *Y* axis and *FPR* ($1 - especificity$) is plotted on the *X* axis.

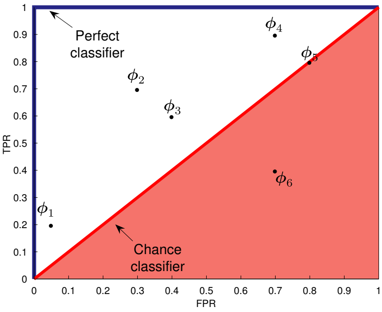


- ϕ_1: *k*NN
- ϕ_2: Neural network
- ϕ_3: Naive Bayes
- ϕ_4: SVM
- ϕ_5: Linear regression
- ϕ_6: Decision tree

Receiver Operating Characteristics (ROC)

ROC Space

Coordinate system used for visualizing classifiers performance where *TPR* is plotted on the *Y* axis and *FPR* (1 − *especificity*) is plotted on the *X* axis.

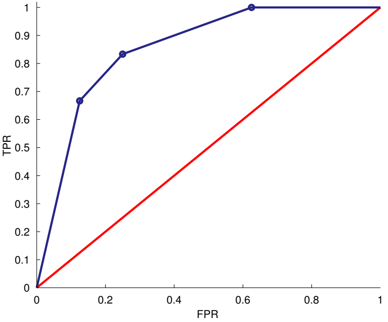


- ϕ_1: *k*NN
- ϕ_2: Neural network
- ϕ_3: Naive Bayes
- ϕ_4: SVM
- ϕ_5: Linear regression
- ϕ_6: Decision tree

Receiver Operating Characteristics (ROC)

ROC Curve

For a probabilistic/fuzzy classifier, a ROC curve is a plot of the TPR *vs.* FPR ($1-$ especificity) as its discrimination threshold is varied

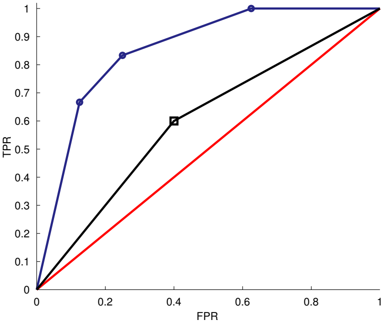


$p(c^+\|\boldsymbol{x})$	$T=0.2$	$T=0.5$	$T=0.8$	C
0.99	c^+	c^+	c^+	c^+
0.90	c^+	c^+	c^+	c^+
0.85	c^+	c^+	c^+	c^+
0.80	c^+	c^+	c^+	c^-
0.78	c^+	c^+	c^-	c^+
0.70	c^+	c^+	c^-	c^-
0.60	c^+	c^+	c^-	c^+
0.45	c^+	c^-	c^-	c^-
0.40	c^+	c^-	c^-	c^-
0.30	c^+	c^-	c^-	c^-
0.20	c^+	c^-	c^-	c^+
0.15	c^-	c^-	c^-	c^-
0.10	c^-	c^-	c^-	c^-
0.05	c^-	c^-	c^-	c^-

Receiver Operating Characteristics (ROC)

ROC Curve

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



$p(c^+\|\boldsymbol{x})$	$T = 0.2$	$T = 0.5$	$T = 0.8$	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)

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)

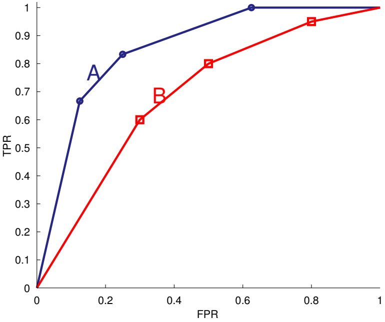
ROC Curve

- 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

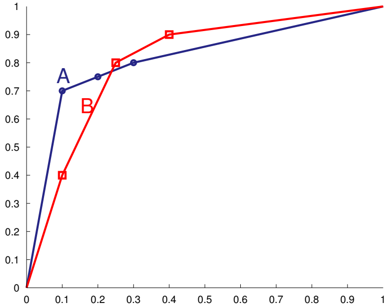
Receiver Operating Characteristics (ROC)



Dominance

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

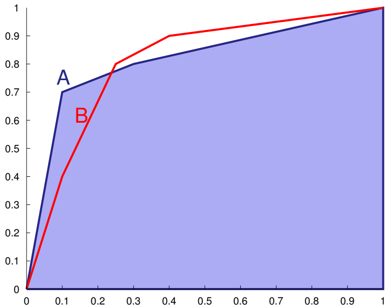
Receiver Operating Characteristics (ROC)



No-Dominance

- The dominance relationship may not be so clear
- No model is the best one in any possible scenario

Receiver Operating Characteristics (ROC)



Caution!! AUC may treats misclassification cost differently for each classification algorithm (Hand 2009, 2010, Hand & Anagnostopoulos 2013)

Area Under ROC Curve

- If A dominates B: $AUC(A) \geq AUC(B)$
- If A does not dominate B AUC “cannot identify the best classifier”
- Less sensitive to skew class distribution than Acc.
- Less sensitive to misclassification cost than Acc.

Generalization to Multilabel-Class

- Most of the presented scores are for binary classification
- A 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*

Generalization to Multilabel-Class

- Most of the presented scores are for binary classification
- A 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
- A 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
- A 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
- A 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
- A 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

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

Scores

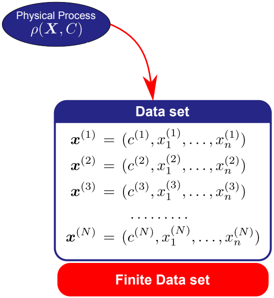
The Use of a Specific Score Depends on:

- Application domain
- Characteristics of the problem
- Characteristics of the data set
- Our interest when solving the problem
- etc.

Introduction

Estimation

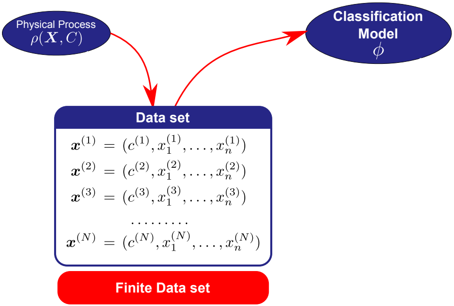
- Select a score to measure the quality
- We would like to calculate the true value of the score
- Limited information is available: only estimations are possible



Introduction

Estimation

- Select a score to measure the quality
- We would like to calculate the true value of the score
- Limited information is available: only estimations are possible



Introduction

Estimation

- Select a score to measure the quality
- We would like to calculate the true value of the score
- Limited information is available: only estimations are possible

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

Classification Model
ϕ

Data set

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

Finite Data set

Quality Measures

Error
Recall
Precision
....

Introduction

Estimation

- Select a score to measure the quality
- We would like to calculate the true value of the score
- Limited information is available: only estimations are possible

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

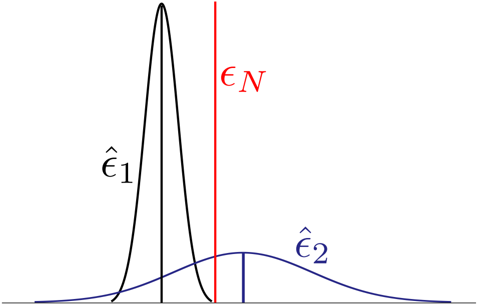
Expected value of the score for a classifier trained on a set of N data samples sampled from $\rho(C, \boldsymbol{X})$

Introduction

True Value - ϵ_N

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

$\rho(C, \boldsymbol{X})$ unknown $\rightarrow$ score estimator ($\hat{\epsilon}_N$)



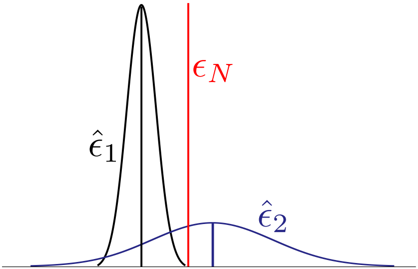
Introduction

True Value

Expected value of the score given $\rho(C, \boldsymbol{X})$

Apparent Value - point estimate

A value of the score obtained from a set of instances sampled from $\rho(C, \boldsymbol{X})$

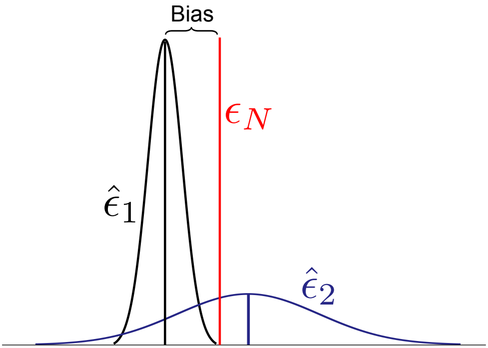


Introduction

Bias

Average difference between the estimate and its true value:

$E_\rho[\hat{\epsilon}_N] - \epsilon_N$

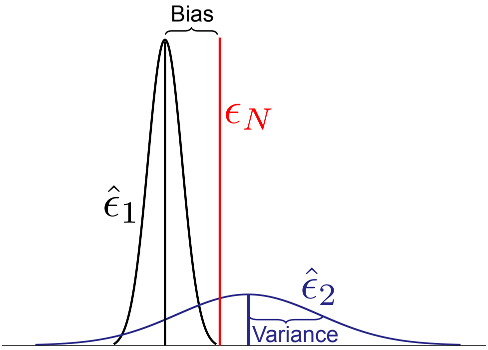


Introduction

Variance

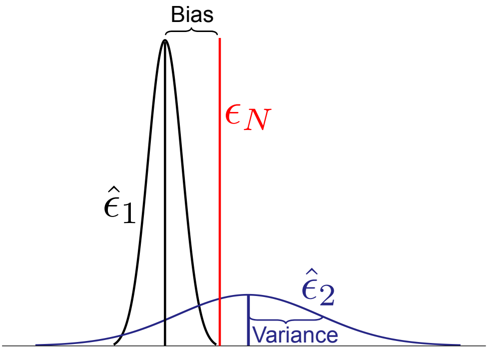
Deviation of the estimated value from its expected value:

$$var(\hat{\epsilon}_N) = E[(\hat{\epsilon}_N - E_\rho[\hat{\epsilon}_N])^2]$$



Introduction

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



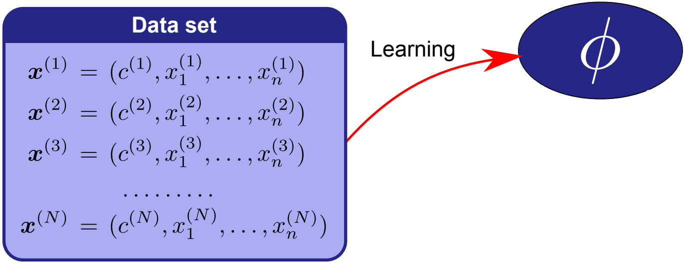
Introduction

Data set

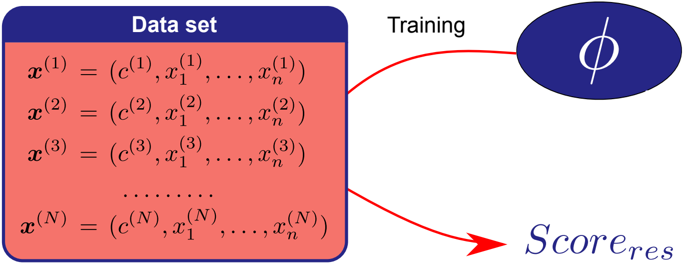
$$\begin{aligned}
\boldsymbol{x}^{(1)} &= (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)}) \\
\boldsymbol{x}^{(2)} &= (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)}) \\
\boldsymbol{x}^{(3)} &= (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)}) \\
& \ldots\ldots\ldots \\
\boldsymbol{x}^{(N)} &= (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})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

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

Resubstitution



Resubstitution

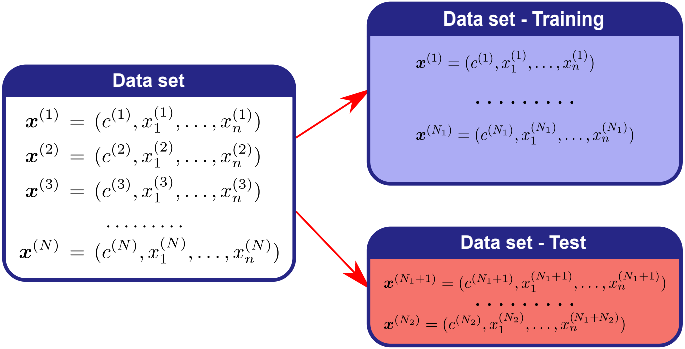


Resubstitution

Classification Error Estimation

- The simplest estimation method
- Biased estimation ϵ_N
- Smaller variance
- Too optimistic (overfitting problem)
- Bad estimator

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

Hold-Out

Data set

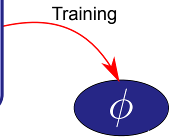
$$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$$
$$\boldsymbol{x}^{(2)} = (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)})$$
$$\boldsymbol{x}^{(3)} = (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)})$$
$$\cdots\cdots\cdots$$
$$\boldsymbol{x}^{(N)} = (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})$$

Data set - Training

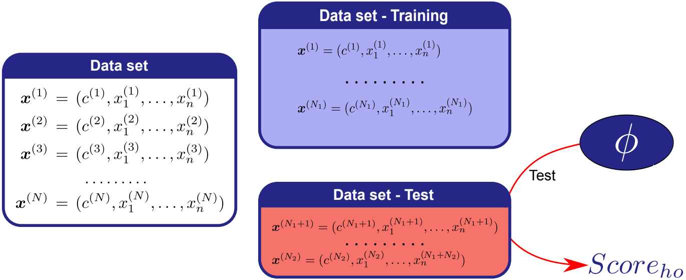
$$\boldsymbol{x}^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$$
$$\cdots\cdots\cdots$$
$$\boldsymbol{x}^{(N_1)} = (c^{(N_1)}, x_1^{(N_1)}, \ldots, x_n^{(N_1)})$$

Data set - Test

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



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

Classification Error Estimation

- Biased estimator of ϵ_N
- Large bias (pessimistic estimation of the true classification error)
- Bias and variance are related to N_1 and N_2

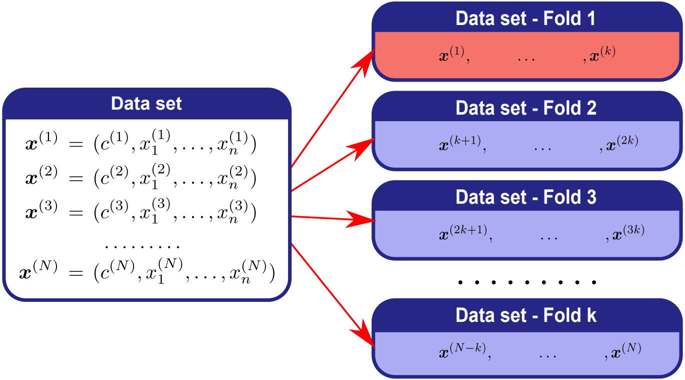
Repeated Hold-Out

- Repeat the Hold-Out t-times
- Simple average over results

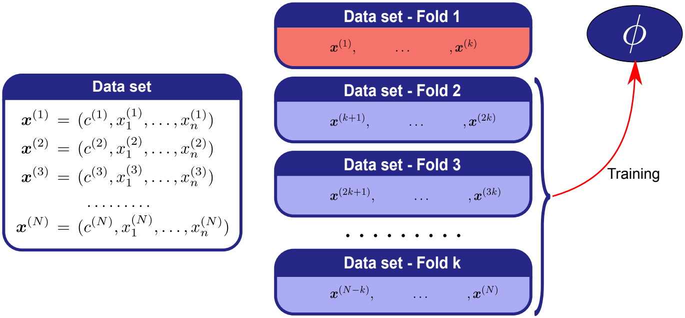
Classification Error Estimation

- Same bias as standard Hold-Out
- Reduces the variance with respect to the Hold-Out

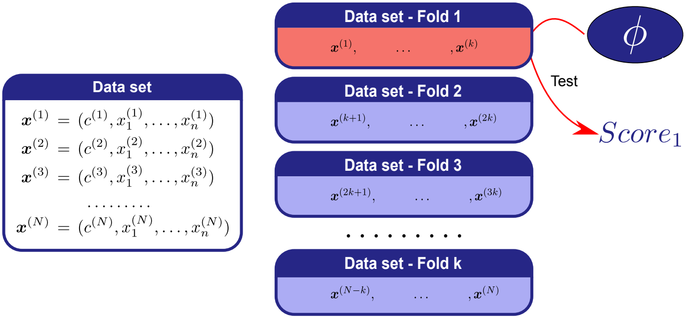
k-Fold Cross-Validation



k-Fold Cross-Validation



k-Fold Cross-Validation



k-Fold Cross-Validation

Data set

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

Data set - 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

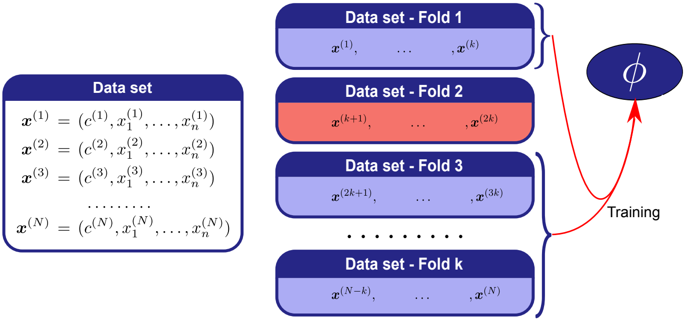
$\boldsymbol{x}^{(2k+1)}, \quad \ldots \quad , \boldsymbol{x}^{(3k)}$

. . . .

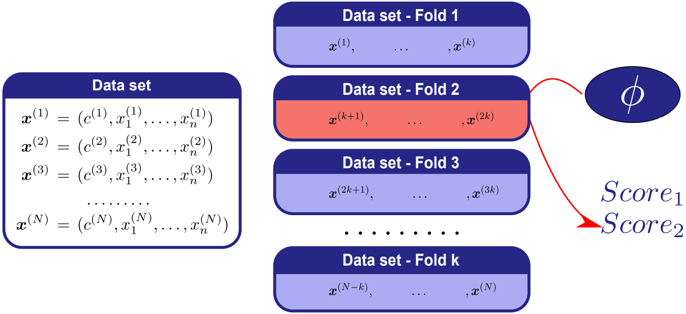
Data set - Fold k

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

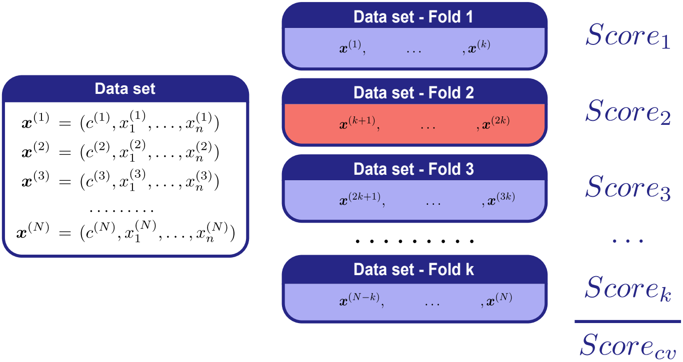
k-Fold Cross-Validation



k-Fold Cross-Validation



k-Fold Cross-Validation



k-Fold Cross-Validation

Classification Error Estimation

- Biased estimation of ϵ_N
- Smaller bias than Hold-Out

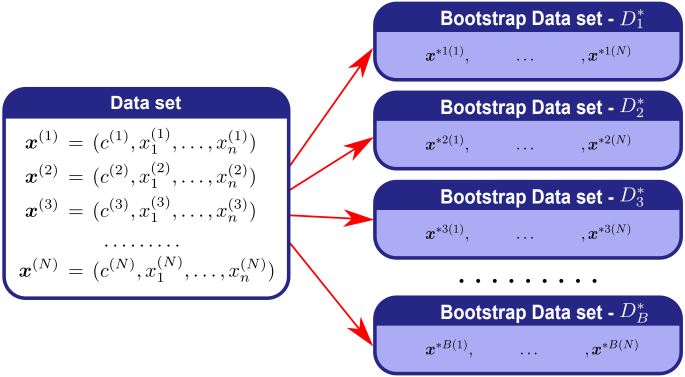
Repeated *k*-Fold Cross-Validation

- Similar to repeated Hold-Out:
  - Repeat Cross-Validation *t*-times
  - Simple average over results

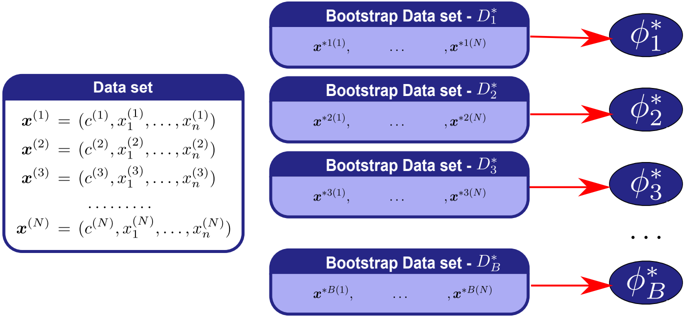
Classification Error Estimation

- Same bias as standard *k*-fold Cross-Validation
- Reduces the variance with respect *k*-fold Cross-Validation

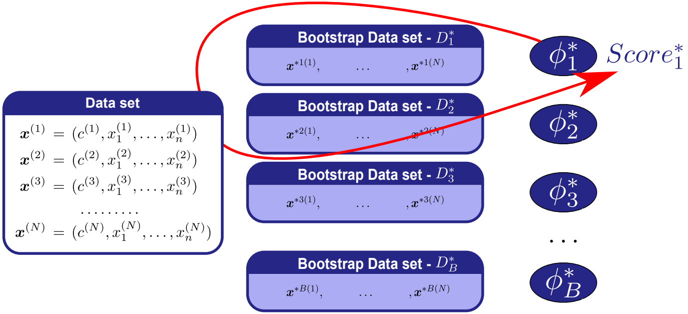
Bootstrap


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

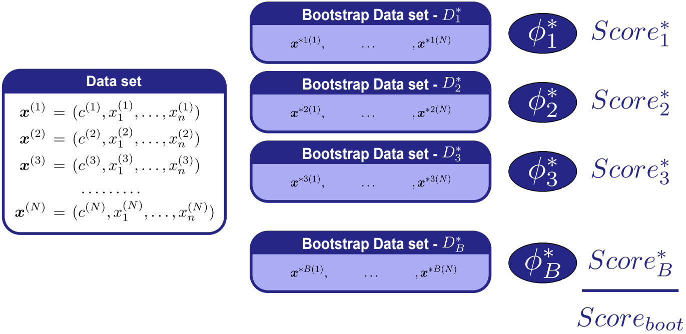
Bootstrap


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^*
$\boldsymbol{x}^{*1(1)}, \ldots, \boldsymbol{x}^{*1(N)}$
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^*
ϕ_2^*
ϕ_3^*
...
ϕ_B^*

Bootstrap


Data set
$x^{(1)} = (c^{(1)}, x_1^{(1)}, \ldots, x_n^{(1)})$
$x^{(2)} = (c^{(2)}, x_1^{(2)}, \ldots, x_n^{(2)})$
$x^{(3)} = (c^{(3)}, x_1^{(3)}, \ldots, x_n^{(3)})$
....
$x^{(N)} = (c^{(N)}, x_1^{(N)}, \ldots, x_n^{(N)})$
Bootstrap Data set - D_1^*
$x^{*1(1)}, \ldots, x^{*1(N)}$
Bootstrap Data set - D_2^*
$x^{*2(1)}, \ldots, x^{*2(N)}$
Bootstrap Data set - D_3^*
$x^{*3(1)}, \ldots, x^{*3(N)}$
Bootstrap Data set - D_B^*
$x^{*B(1)}, \ldots, x^{*B(N)}$
ϕ_1^*
$Score_1^*$
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^*
$Score_2^*$
ϕ_3^*
$Score_3^*$
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

- Bias correction terms can be used for error estimation

Bootstrap

- Improves bias estimation
- Well established methods

Hold-Out/Cross-Validation

- Several proposals
- Improves bias estimation
- Not very extended in practice

Improving the Estimation - Bias

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

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

Improvement

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

Improving the Estimation - Bias

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 - Bias

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

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

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

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

Improving the Estimation - Bias

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

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

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 - Bias

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 - 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

Classification Error Estimation

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

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 very 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

- Statistical test?
- A a controversial statistical tool
- Often criticized due to a misuse of it
- It is not perfect, but can be useful
- Important undertand methodology and limitations
- More information: see Santafé et al. 2015

- A. Urkullu, A. Pèrez, and B. Calvo (2019). On the evaluation and selection of classifier learning algorithms with crowdsourced data. *Applied Soft Computing Journal, 80:832-844*
- B. Calvo and G. Santafé (2016). SCMAMP: Statistical comparison of multiple algorithms in multiple problems. *R Journal, 8(1):248-256*
- G. Santafé, J. A. Lozano, and I. Inza (2015). Dealing with the evaluation of supervised classification algorithms. *Artificial Intelligence Review, 44:467-508*
- Japkowicz, N. and Shah, M. (2011). Evaluating Learning Algorithms: A Classification Perspective. *Cambridge: Cambridge University Press*

Estimating classification performance

Guzmán Santafé

Spatial Statistics Group
Public University of Navarre

DATAI-UNAV, November 2022

Spatial Statistics
Public University of Navarre

upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa