Model Evaluation

Boston University CS 506 - Lance Galletti

evaluate how our model is !

Confusion Matrix

we prefer TP + TN

	Predicted Class		
Actual Class		Class = Yes	Class = No
	Class = Yes	a (TP) prefer	b (FN) not prefer
	Class = No	c (FP) not prefer	d (TN) prefer.

Accuracy = (a + d) / (a + b + c + d)

Accuracy can be misleading

Binary classification problem where:

Number of Class 0 examples: 9990

Number of Class 1 examples: 10

A model that predicts everything to be class 0 will have an accuracy of 99.9%

not always true

Cost Matrix

<table>
<tr><td></td><td colspan="3">Predicted Class</td></tr>
<tr><td rowspan="3">Actual Class</td><td></td><td>Class = Yes</td><td>Class = No</td></tr>
<tr><td>Class = Yes</td><td>C(Yes|Yes)</td><td>C(No|Yes)</td></tr>
<tr><td>Class = No</td><td>C(Yes|No)</td><td>C(No|No)</td></tr>
</table>

Cost of Classification

COST	Predicted Class	
Actual Class	Yes	No
Yes	-1	100
No	1	0

$-250 + 4500$

$+5+0$

4250

4255

by num just multiply corresponding num here

Model 1	Predicted Class	
Actual Class	Yes	No
Yes	150	40
No	60	250

Accuracy = 80%
Cost = 3910

Model 2	Predicted Class	
Actual Class	Yes	No
Yes	250	45
No	5	200

Accuracy = 90%
Cost = 4255

FORM

Other metrics

COST	Predicted Class	
Actual Class	Yes	No
Yes	a	b
No	c	d

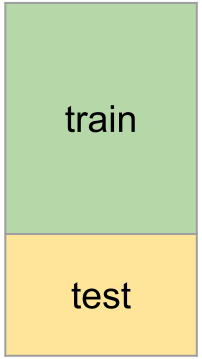
- Precision = a / (a + c)
- Recall = a / (a + b)
- F-measure = 2RP / (R + P)

Methods of Estimation

Goal: get a reliable estimate of the performance of the model on unseen data

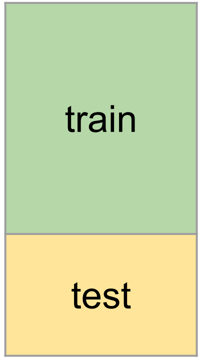
Methods of Estimation

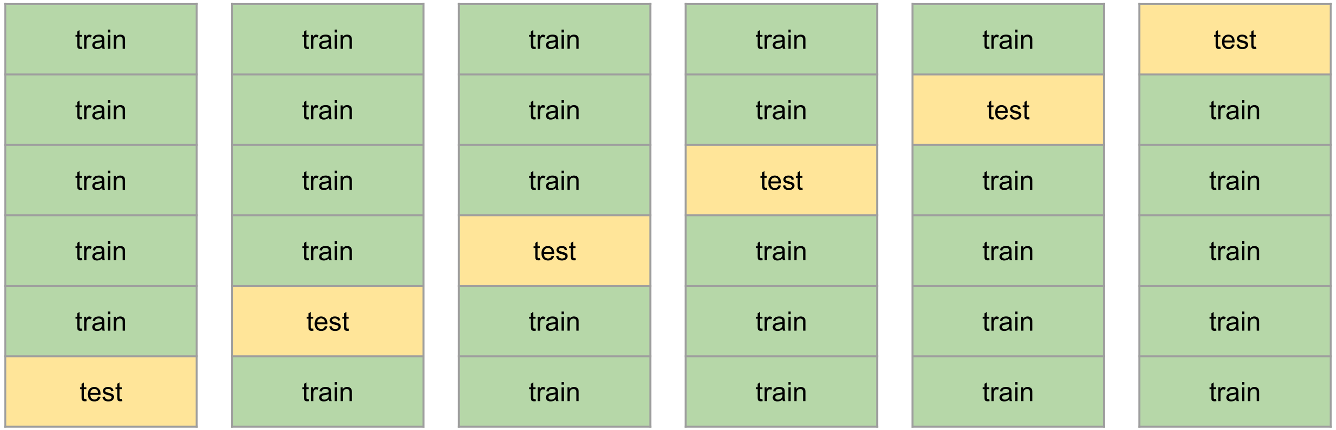
- Holdout:
  - Ex: reserve ¼ of the dataset for testing and use ¾ for training



Methods of Estimation

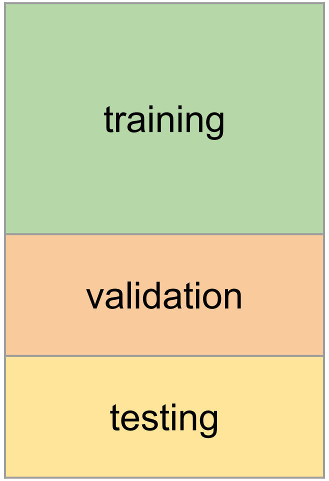
- Holdout:
  - Ex: reserve ¼ of the dataset for testing and use ¾ for training
- Cross Validation:
  - Partition into K disjoint subsets
  - K-fold: train on K-1 partitions, test on the remaining on
  - K = n : leave-one-out





Validation Set

For tuning parameters



together they're better than individually

Ensemble Methods

Boston University CS 506 - Lance Galletti

The idea

Suppose you have trained 17 different classifiers on a dataset.

The idea

Suppose you have trained 17 different classifiers on a dataset.

- Every classifier has error rate **ε** = .20
- Assume all classifiers are independent

The idea

Suppose you have trained 17 different classifiers on a dataset.

- Every classifier has error rate **ε** = .20
- Assume all classifiers are independent

In order to classify a new record we poll all 17 classifiers and take the class that the majority agrees on.

What is the probability that this ensemble classifier makes a wrong prediction?

The idea

The majority needs to make a mistake (i.e. at least 9 out of 17 make mistakes)

$$P(X \geq 9) = \sum_{k=9}^{17} \binom{17}{k} (.2)^k (1 - .2)^{17-k} = 0.002581463$$

* from error rate of 20 to like 0.02584 percent!

How to generate independent classifiers?

By generating samples of the data to train on

- Bagging → replacement
- Boosting →

Bagging

Original data	1	2	3	4	5	6	7	8	9	10
Bootstrap sample 1	7	8	10	10	3	6	1	1	4	5
Bootstrap sample 2	6	2	7	9	3	5	7	7	1	8
Bootstrap sample 3	2	5	6	1	4	1	8	9	4	3

Build a classifier on each bootstrapped sample

Boosting

at each step adjust sample weight

so they explored more next time

An adaptive sampling process to change the sampling distribution based on difficult-to-classify examples.

Start with all samples having equal probability of being selected. Next boosting round, increase the weights of those samples that were misclassified, decrease the weights of those samples that were correctly classified.

depend on how

well it does, modify the weight.

Boosting

Original data	1	2	3	4	5	6	7	8	9	10
Bootstrap sample 1	7	8	10	10	3	6	1	1	4	5
Bootstrap sample 2	6	4	7	9	3	5	7	4	1	8
Bootstrap sample 3	2	5	4	1	4	1	4	4	2	3

Here, sample 4 is hard to classify.

Boosting

have weight.

Original data	1	2	3	4	5	6	7	8	9	10
Bootstrap sample 1	7	8	10	10	3	6	1	1	4	5
Bootstrap sample 2	6	4	7	9	3	5	7	4	1	8
Bootstrap sample 3	2	5	4	1	4	1	4	4	2	3

Classifiers trained on each sample and are given a weight that is a function of their error rate.

~~Boosting~~ Bagging

Example: 5 classifiers independent?

C_1 predicts 1
C_2 predicts 0
C_3 predicts 0
C_4 predicts 1
C_5 predicts 0

Boosting

Example: 5 classifiers

Associate w/ weight.

C_1 predicts 1 has a weight of .25
C_2 predicts 0 has a weight of .1
C_3 predicts 0 has a weight of .5
C_4 predicts 1 has a weight of .05
C_5 predicts 0 has a weight of .1