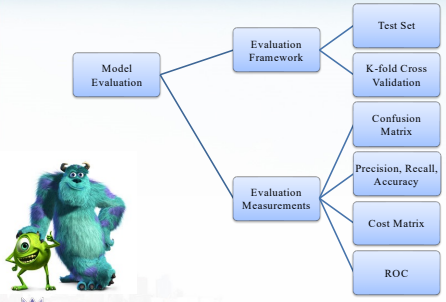
Lecture Notes on
Pattern Recognition

Session 08B: Model Evaluation

Gang Li
School of Information Technology
Deakin University, VIC 3125, Australia

0

Road map



1

Model Evaluation

- What do we care?
- The Test Set method
- The Leave One Out method
- The K-fold Cross-Validation

2

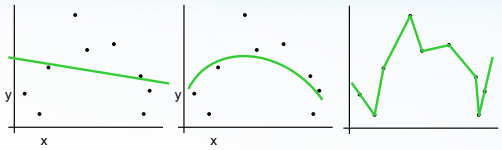
What do we care?

- Regular Performance of Algorithms
  - *Accuracy*
  - *Scope*
  - *Efficiency*
- Which ones we care?
  1. *Accuracy*
  2. *Efficiency*
  3. *Scope*

? Fitting Accuracy, or
? Prediction Accuracy

3

What kind of accuracy we want?

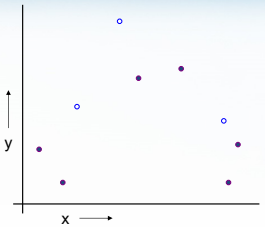


Why not choose the method with the best fit to the data?

"How well are you going to *predict* future data drawn from the same distribution?"

4

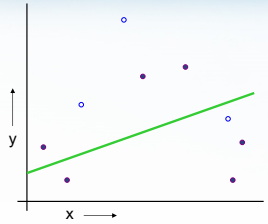
The test set method



1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set

5

The test set method

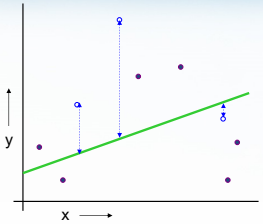


1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set

(Linear regression example)

6

The test set method



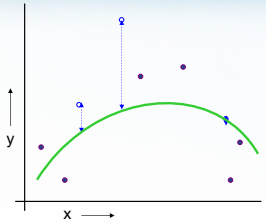
1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set
4. Estimate your future performance with the test set

(Linear regression example)

Mean Squared Error = 2.4

7

The test set method



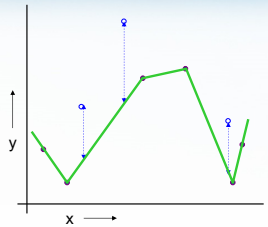
1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set
4. Estimate your future performance with the test set

(Quadratic regression example)

Mean Squared Error = 0.9

8

The test set method



1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set
4. Estimate your future performance with the test set

(Join the dots example)

Mean Squared Error = 2.2

9

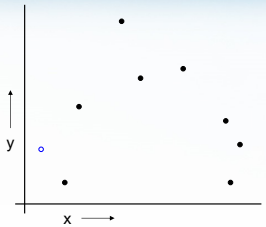
The test set method

Pros	Cons
• Very very simple • Can then simply choose the method with the best test-set score	• Waste data: we get an estimate of the best method to apply to 30% less data • If we don't have much data, our test-set might just be lucky or unlucky

We say the "test-set estimator of performance has high variance"

10

Leave-One-Out Method

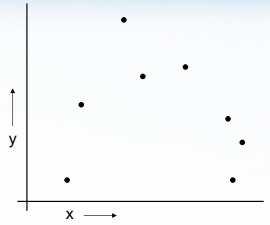


For k=1 to R

1. Let (x_k, y_k) be the k^{th} record

11

Leave-One-Out Method

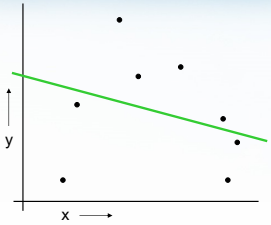


For k=1 to R

1. Let (x_k,y_k) be the k^{th} record
2. Temporarily remove (x_k,y_k) from the dataset

12

Leave-One-Out Method

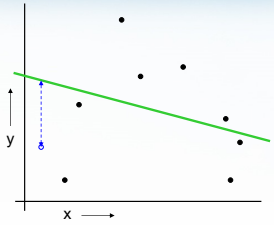


For k=1 to R

1. Let (x_k,y_k) be the k^{th} record
2. Temporarily remove (x_k,y_k) from the dataset
3. Train on the remaining R-1 datapoints

13

Leave-One-Out Method

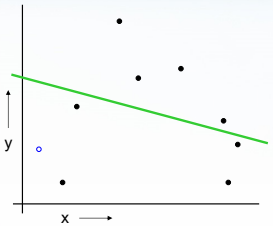


For k=1 to R

1. Let (x_k,y_k) be the k^{th} record
2. Temporarily remove (x_k,y_k) from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error (x_k,y_k)

14

Leave-One-Out Method



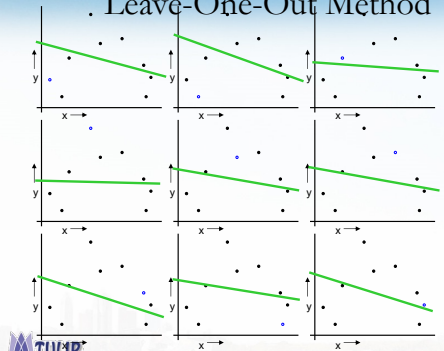
For k=1 to R

1. Let (x_k,y_k) be the k^{th} record
2. Temporarily remove (x_k,y_k) from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error (x_k,y_k)

When you've done all points, report the mean error.

15

Leave-One-Out Method



For k=1 to R

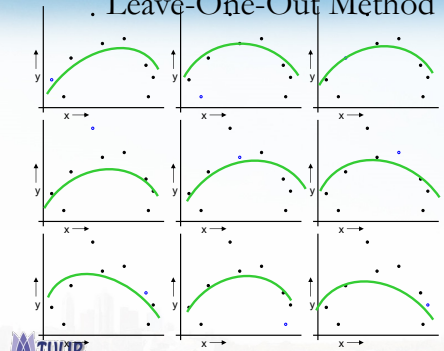
1. Let (x_k,y_k) be the k^{th} record
2. Temporarily remove (x_k,y_k) from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error (x_k,y_k)

When you've done all points, report the mean error.

$MSE_{LOOCV} = 2.12$

16

Leave-One-Out Method



For k=1 to R

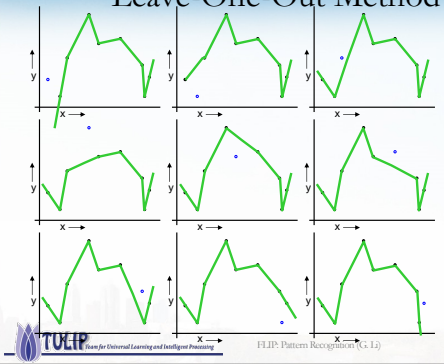
1. Let (x_k,y_k) be the k^{th} record
2. Temporarily remove (x_k,y_k) from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error (x_k,y_k)

When you've done all points, report the mean error.

$MSE_{LOOCV}=0.962$

17

Leave-One-Out Method



For k=1 to R

1. Let (x_k, y_k) be the k^{th} record
2. Temporarily remove (x_k, y_k) from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error (x_k, y_k)

When you've done all points, report the mean error.

$MSE_{LOOCV}=3.33$

18

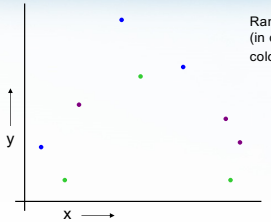
Which kind of Cross Validation?

	Cons	Pros
Test-set	Variance: unreliable estimate of future performance	Cheap
Leave-one-out	Expensive. Has some weird behavior	Doesn't waste data

..can we get the best of both worlds?

19

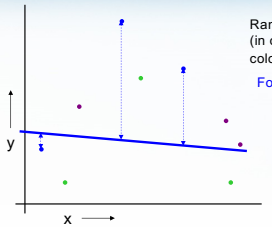
k-fold Cross Validation



Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored blue Green and yellow)

20

k-fold Cross Validation

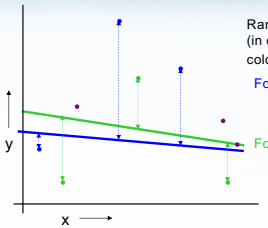


Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored blue Green and yellow)

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

21

k-fold Cross Validation



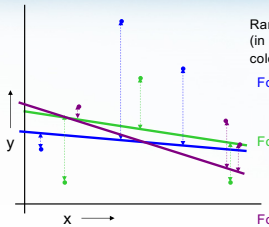
Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored blue Green and yellow)

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

For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

22

k-fold Cross Validation



Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored blue Green and yellow)

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

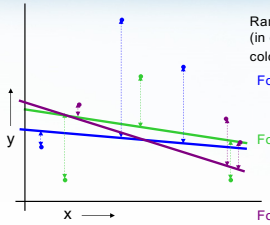
For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

For the yellow partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.

Then report the mean error

23

k-fold Cross Validation



Linear Regression $MSE_{3FOLD}=2.05$

Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored blue Green and yellow)

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

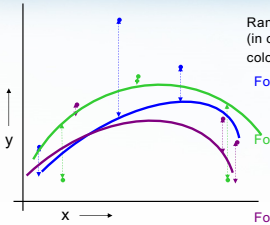
For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

For the yellow partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.

Then report the mean error

24

k-fold Cross Validation



Quadratic Regression $MSE_{3FOLD}=1.11$

Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored blue Green and yellow)

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

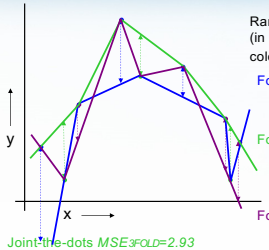
For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

For the yellow partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.

Then report the mean error

25

k-fold Cross Validation



Joint-the-dots $MSE_{3FOLD}=2.93$

Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored blue Green and yellow)

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

For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

For the yellow partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.

Then report the mean error

26

Which kind of Cross Validation?

	Downside	Upside
Test-set	Variance: unreliable estimate of future performance	Cheap
Leave-one-out	Expensive. Has some weird behavior	Doesn't waste data
10-fold	Wastes 10% of the data. 10 times more expensive than test set	Only wastes 10%. Only 10 times more expensive instead of R times.
3-fold	Wastier than 10-fold. Expensivier than test set	Slightly better than test-set
R-fold	Identical to Leave-one-out	

27

CV-based Model Selection

- We're trying to decide which algorithm to use.
- We train each model and make a table…

i	fi	TRAIN-ERR	10-FOLD-CV-ERR	Choice
1	f1			
2	f2			
3	f3			
4	f4			
5	f5			
6	f6			

28

Other Model Selection Methods

- Model selection methods:
  1. *Cross-validation*
  2. *Occam's Razor Type: The smaller, the better*
     - *Minimum Description Length (MDL)*
     - *Minimum Messaging Length (MML)*
     - *AIC (Akaike Information Criterion)*
     - *BIC (Bayesian Information Criterion)*
     - *VC-dimension (Vapnik-Chervonenkis Dimension)*

Only directly applicable to choosing classifiers

29

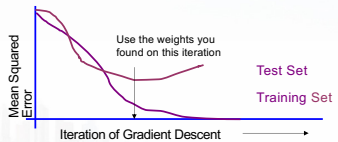
Cross-Validation for regression

- Choosing the number of hidden units in a neural net
- Feature selection
- Choosing a polynomial degree
- Choosing which regressor to use

30

Supervising Gradient Descent

- This is a weird but common use of ***Test-set validation***
  - Suppose you have a neural net with too many hidden units. It will overfit.
  - As gradient descent progresses, maintain a graph of `MSE-testset-error` vs. `Iteration`



31

Cross-validation for classification

- Instead of computing the **mean squared errors (MSE)** on a test set, you should compute various measurements …
  - *Error rate* (or its dual part *Accuracy*):
    - The total number of misclassifications on a test-set

32

Evaluation and performance

- Confusion Matrix
- Accuracy
- Model Comparison

33

Evaluation and performance

- How well is your classification algorithm?
  - Focus on the predictive capability of a model, rather than how fast it takes to classify or build models, scalability, etc.
- Confusion matrix
  - Detail classification result for each class.
- Accuracy
  - How well we can predict for each class

34

Evaluation

- Take 'Yes' as the positive class (class of interest)

True Class	Algorithm returns
Yes	Yes
Yes	Yes
Yes	No
Yes	Yes
No	No
No	No
No	Yes
No	No
No	Yes
Yes	Yes
Yes	Yes
No	No
No	No
Yes	Yes

Classified As →	A=Yes	B=No
A=Yes	6	1
B=No	2	5

		Predicted class	
		+	-
Actual Class	+	a	b
	-	c	d

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

`sklearn.metrics.confusion_matrix`

35

Evaluation

- Most widely-used metric:

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

- Consider a binary classification problem
  - # of Class "Positive" instances = 9990
  - # of Class "Negative" instances = 10
- A naïve model that predicts everything positive with accuracy 99.9%

		Predicted class	
		+	-
Actual Class	+	a	b
	-	c	d

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

36

Evaluation

- Alternative Measures

  Precision = a/(a+c)

  Recall = a/(a+b)

  F-measure or F1 = 2a/(2a+b+c)
- Other measures
  - ROC
  - AUC

		Predicted class	
		+	-
Actual Class	+	a	b
	-	c	d

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

37

Cost Matrix

- `C(i,j)`
  - Cost of misclassifying class `i` instance as class `j`

Cost matrix		Predicted class	
	C(i,j)	+	-
Actual Class	+	…	
	-	…	

		Predicted class	
		+	-
Actual Class	+	a	b
	-	c	d

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

38

Cost Matrix: example

Cost matrix		Predicted class	
	C(i,j)	+	-
Actual Class	+	-1	100
	-	1	0

Model1		Predicted class	
		+	-
Actual Class	+	180	20
	-	60	240

accuracy = 82%
Cost = 1880

Model2		Predicted class	
		+	-
Actual Class	+	140	60
	-	10	290

accuracy = 86%
Cost = 5870

39

Cost-Sensitive Classification

- Traditional Classification
  - Given a query `t`, we will classify it as class `i` if

    $$i = \arg\max_j p(j \mid t)$$

  - For Binary Classification, classify it as "positive" if

    `p(+|t) > p(-|t)`
- Cost-Sensitive Classification
  - Classify `t` as j

    $$i = \arg\min_j \sum_k p(k \mid t) \times C(k, j)$$

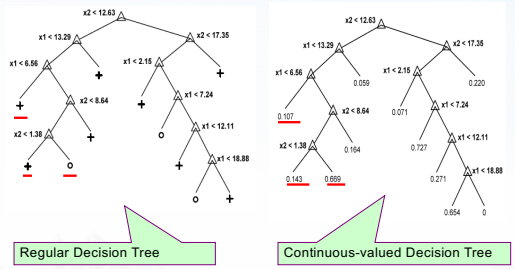
40

ROC: Receiver Operating Characteristic

- A graphical approach for displaying *trade-off* between *detection rate* and *false alarm rate*
  - Developed in 1950s for signal detection theory to analyze noisy signals
- To draw ROC curve, classifier must produce *continuous-valued output*
  - Outputs are used to ~~rank~~ test records, *from* the most likely record to be classified as a positive class *to* the most likely record to be classified as a negative class
  - ~~*Performance of each classifier*~~ is represented as ~~*a point*~~ on the ROC curve

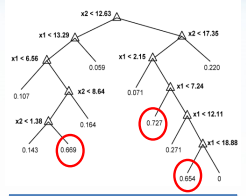
41

Continuous-valued outputs from a decision tree



42

ROC Curve Example



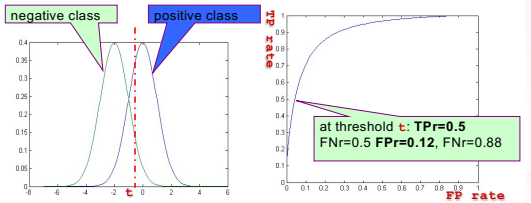
a=0.3		Predicted class	
		o	+
Actual Class	o	645	209
	+	298	948

a=0.7		Predicted class	
		o	+
Actual Class	o	181	673
	+	78	1168

43

ROC Curve Example

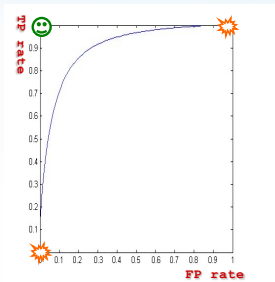
- 1-dimensional data set containing 2 classes
  - *Positive* vs. *negative*
  - Any points located at `x>t` is classified as positive



44

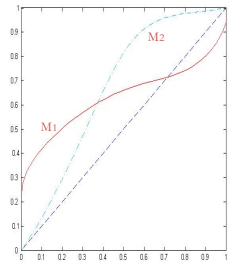
ROC Curve

- `(TPr, FPr)`
  - `TPR = TP/(TP+FN)` = RECALL
  - `FPR = FP/(FP+TN)`
  - `(0,0)`
    - Declare everything to be negative
  - `(1,1)`
    - Declare everything to be positive class
  - `(1,0)`
    - Ideal
- Diagonal line:
  - Random guessing
- Below diagonal line
  - Prediction is opposite of the true class



45

Using ROC for Model Comparison



- No Model consistently outperforms the other
  - `M1` is better for small FP Rate
  - `M2` is better for large FP Rate
- Area under the ROC curve
  - Ideal:
    - `Area = 1`
  - Random guess
    - `Area = 0.5`

46

How to Construct an ROC Curve

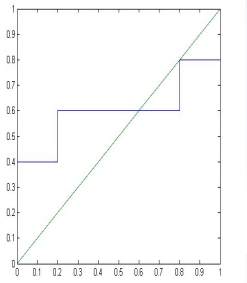
Instance	P(+|A)	True
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

- Use classifier that produces continuous valued output for each test instance `P(+|A)`
- Sort the instances according to `P(+|A)` in decreasing order
- Apply threshold at each unique value of `P(+|A)`
- Count the number of `TP, FP, TN, FN` at each threshold
- `TPR = TP/(TP+FN)`
- `FPR = FP/(FP+TN)`

47

How to Construct an ROC Curve

Class	Thrd	TP	FP	TN	FN	TPR	FPR
+	0.25	5	5	0	0	1	1
-	0.43	4	5	0	1	0.8	1
+	0.53	4	4	1	1	0.8	0.8
-	0.76	3	4	1	2	0.6	0.8
-	0.85	3	3	2	2	0.6	0.6
-	0.85	3	2	3	2	0.6	0.4
+	0.85	3	1	4	2	0.6	0.2
-	0.87	2	1	4	3	0.4	0.2
+	0.93	2	0	5	3	0.4	0
+	0.95	1	0	5	4	0.2	0
	1.00	0	0	5	5	0	0



48

Model Comparison

- Task 1
  - If we have two models M_1 and M_2 tested on ***different data sets***, how can we tell the relative performance of these two models?
- Task 2
  - If we have two models M_1 and M_2 tested on ***same data sets***, how can we tell the relative performance of these two models?

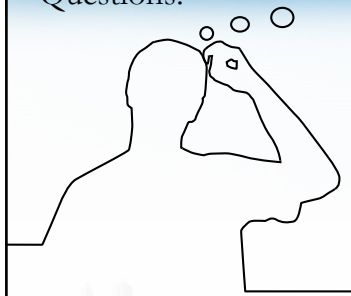
49

Model Comparison

- Given two models:
  - M_1: accuracy = 85%, tested on 30 instances
  - M_2: accuracy = 75%, tested on 5000 instances
- Can we say M_1 is better than M_2?
  - How much ***confidence*** can we place on accuracy of M_1 and M_2?
  - Can the ***difference*** in performance measure be explained as a result of ***random fluctuations*** in the test set?

50

Questions?



51