Advanced Statistical Inference
Classification - Performance Evaluation

Maurizio Filippone
Maurizio.Filippone@eurecom.fr

Department of Data Science
EURECOM

Performance evaluation

- How do we choose a classifier?
  - Which algorithm?
  - Which parameters?
- Need performance indicators.
- We'll cover:
  - 0/1 loss.
  - ROC analysis (sensitivity and specificity)
  - Confusion matrices

0/1 loss

- 0/1 loss: proportion of times classifier is wrong.
- Consider a set of predictions $t_1, \ldots, t_N$ and a set of true labels $t_1^*, \ldots, t_N^*$.
- Mean loss is defined as:

$$\frac{1}{N}\sum_{n=1}^{N}\delta(t_n \neq t_n^*)$$

- ($\delta(a)$ is 1 if a is true and 0 otherwise)
- Advantages:
  - Can do binary or multiclass classification.
  - Simple to compute.
  - Single value.

0/1 loss

Disadvantage: Doesn't take into account class imbalance:

- We're building a classifier to detect a rare disease.
- Assume only 1% of population is diseased.
- Diseased: $t = 1$
- Healthy: $t = 0$
- What if we always predict healthy? ($t = 0$)
- Accuracy 99%
- But classifier is rubbish!

Sensitivity and specificity

- We'll stick with our disease example.
- Need to define 4 quantities. The numbers of:
- **True positives (TP)** – the number of objects with $t_n^* = 1$ that are classified as $t_n = 1$ (diseased people diagnosed as diseased).
- **True negatives (TN)** – the number of objects with $t_n^* = 0$ that are classified as $t_n = 0$ (healthy people diagnosed as healthy).
- **False positives (FP)** – the number of objects with $t_n^* = 0$ that are classified as $t_n = 1$ (healthy people diagnosed as diseased).
- **False negatives (FN)** – the number of objects with $t_n^* = 1$ that are classified as $t_n = 0$ (diseased people diagnosed as healthy).

Sensitivity

$$S_e = \frac{TP}{TP + FN}$$

- The proportion of diseased people that we classify as diseased.
- The higher the better.
- In our example, $S_e = 0$.

Introduction
M. Filippone
Assessing classifier performance
0/1 loss
ROC analysis
Confusion matrices
Summary

Specificity

$$S_p = \frac{TN}{TN + FP}$$

- The proportion of healthy people that we classify as healthy.
- The higher the better.
- In our example, $S_p = 1$.

Introduction
M. Filippone
Assessing classifier performance
0/1 loss
ROC analysis
Confusion matrices
Summary

Optimising sensitivity and specificity

- We would like both to be as high as possible.
- Often increasing one will decrease the other.
- Balance will depend on application:
- e.g. diagnosis:
  - We can probably tolerate a decrease in specificity (healthy people diagnosed as diseased)....
  - ...if it gives us an increase in sensitivity (getting diseased people right).

Introduction
M. Filippone
Assessing classifier performance
0/1 loss
ROC analysis
Confusion matrices
Summary

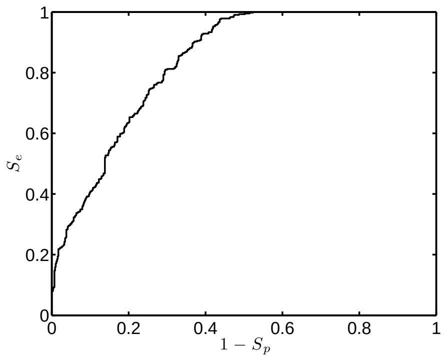
ROC analysis

- Many classification algorithms involve setting a threshold.
- e.g. SVM:

$$t_{\text{new}} = \text{sign}\left(\sum_{n=1}^{N} t_n \alpha_n k(\mathbf{x}_n, \mathbf{x}_{\text{new}}) + b\right)$$

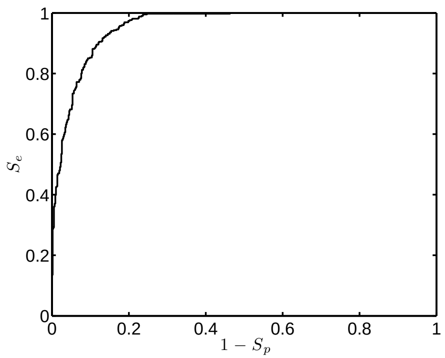
- Implies a threshold of zero (sign function)
- However, we could use any threshold we like....
- The Receiver Operating Characteristic (ROC) curve shows how S_e and $1 - S_p$ vary as the threshold changes.

ROC curve



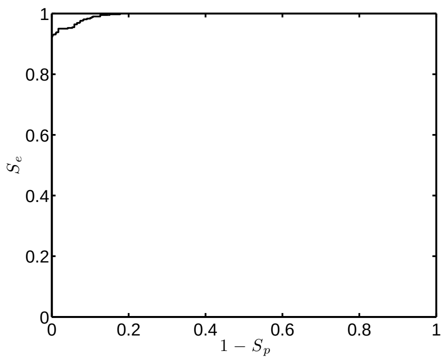
- SVM for nonlinear data (in SVM lecture) with $\beta = 50$.
- Each point is a threshold value.
  - Bottom left – everything classified as 0 (-1 in SVM)
  - Top right – everything classified as 1.
- Goal: get the curve to the top left corner – perfect classification ($S_e = 1, S_p = 1$).

ROC curve



- SVM for nonlinear data (in SVM lecture) with $\beta = 0.01$.
- Better than $\beta = 50$
  - Closer to top left corner.

ROC curve



- SVM for nonlinear data (in SVM lecture) with $\beta = 1$.
- Better still.

AUC

- We can quantify performance by computing the Area Under the ROC Curve (AUC)
- The higher this value, the better.
  - $\beta = 50$: AUC=0.8348
  - $\beta = 0.01$: AUC= 0.9551
  - $\beta = 1$: AUC=0.9936
- AUC is generally a safer measure than 0/1 loss.

Confusion matrices

The quantities we used to compute S_e and S_p can be neatly summarised in a table:

		True class	
		1	0
Predicted class	1	TP	FP
	0	FN	TN

- This is known as a confusion matrix
- It is particularly useful for multi-class classification.
- Tells us where the mistakes are being made.
- Note that normalising columns gives us S_e and S_p

Confusion matrices – example

- 20 newsgroups data.
- Thousands of documents from 20 classes (newsgroups)
- Use a Naive Bayes classifier ($\approx$ 50000 dimensions (words)!)
  - Details in book Chapter.
- $\approx$ 7000 independent test documents.
- Summarise results in 20×20 confusion matrix:

True class

Predicted class	...	10	11	12	13	14	15	16	17	18	19	20
1	...	4	2	0	2	10	4	7	1	12	7	47
2	...	0	0	4	18	7	8	2	0	1	1	3
3	...	0	0	1	0	1	0	1	0	0	0	0
4	...	1	0	1	28	3	0	0	0	0	0	0
⋮												
16	...	3	2	2	5	17	4	376	3	7	2	**68**
17	...	1	0	9	0	3	1	3	325	3	**95**	19
18	...	2	1	0	2	6	2	1	2	325	4	5
19	...	8	4	8	0	10	21	1	16	19	**185**	7
20	...	0	0	1	0	1	1	2	4	0	1	**92**

- Algorithm is getting 'confused' between classes 20 and 16, and 19 and 17.
  - 17: talk.politics.guns
  - 19: talk.politics.misc
  - 16: talk.religion.misc
  - 20: soc.religion.christian
- Maybe these should be just one class?
- Maybe we need more data in these classes?
- Confusion matrix helps us direct our efforts to improving the classifier.

Summary

- Introduced two different performance measures:
  - 0/1 loss
  - ROC/AUC
- Introduced confusion matrices – a way of assessing the performance of a multi-class classifier.