Classifier's Performance Metrics

Chenghua Lin
Department of Computing Science
University of Aberdeen

Outline

- Accuracy
- Recall, precision and F-measure
- ROC curve
- Cross validation

Confusion Matrix

- The four possible outcomes of a binary classifier are usually shown in a confusion matrix
- A number of performance metrics defined using these counts

		Predicted Class	
		Positive'	negative'
Actual Class	positive	TP	FN
	negative	FP	TN

Confusion Matrix

		Predicted Class	
		Positive'	negative'
Actual Class	positive	TP	FN
	negative	FP	TN

- True Positives (TP)
  - # of correct predictions that an instance is positive
- True Negatives (TN)
  - # of correct predictions that an instance is negative
- False Positives (FP)
  - # of incorrect predictions that an instance is positive
- False Negatives (FN)
  - # of incorrect of predictions that an instance negative

Accuracy

		Positive'	negative'	Predicted Class
Actual Class	positive	TP	FN	
	negative	FP	TN	

- **Accuracy of positive class**: the proportions of positive class instances have been correctly predicted
  - Acc_pos = TP / (TP + FN)
- **Accuracy of negative class**: the proportions of negative class instances have been correctly predicted
  - Acc_pos = TN / (FP + TN)
- **Overall accuracy**: the proportion of the total number of predictions that were correct
  - Acc = (TP + TN) / (TP + FP + FN + TN)

Confusion Matrix: example1

	Spam (Predicted)	Non-Spam (Predicted)	Accuracy
Spam (Actual)	27	6	81.81
Non-Spam (Actual)	10	57	85.07
Overall Accuracy			84

The spam dataset:

- Contains 100 instances
- 33 instances are spam
- 67 instances are non-spam

Accuracy:

- Acc(spam) = 27/(27 + 6) = 81.81%
- Acc(non-spam) =57/(10 + 57) = 85.07%
- Overall_acc = (27 + 57) / (27+6+10+57) = 84%

Confusion Matrix: example2

	Spam (Predicted)	Non-Spam (Predicted)	Accuracy
Spam (Actual)	0	10	??
Non-Spam (Actual)	0	990	??
Overall Accuracy			??

The spam dataset:

- 10 patterns are spam
- 990 pattern are non-spam

Accuracy:

- Acc(spam) = 0/10 = 0%
- Acc(non-spam) =990/990 = 100%
- Overall_acc = (0+990)/(0+10+0+990) = 99%

Issues with accuracy

- The confusion matrix tells us how the classifier is behaving for individual classes.
- Accuracy
  - Work well for (more or less) balanced dataset (e.g., 100 positive and 100 negative data instances)
  - Cannot capture true classifier performance when dataset is highly unbalanced.

Beyond accuracy…

- Recall
  - Aka True Positive rate (TP), or sensitivity
  - Recall = TP/(TP+FN)
- Precision
  - the proportion of the predicted positive instance that were correct (positive predictive value)
  - Precision = TP/(TP + FP)
- F-measure
  - Aka F_1-score, is the harmonic mean of precision and recall
  - Suitable for cases where one of the classes is rare
  - F_1=2x(recall x precision) / (recall + precision)

Confusion Matrix: example3

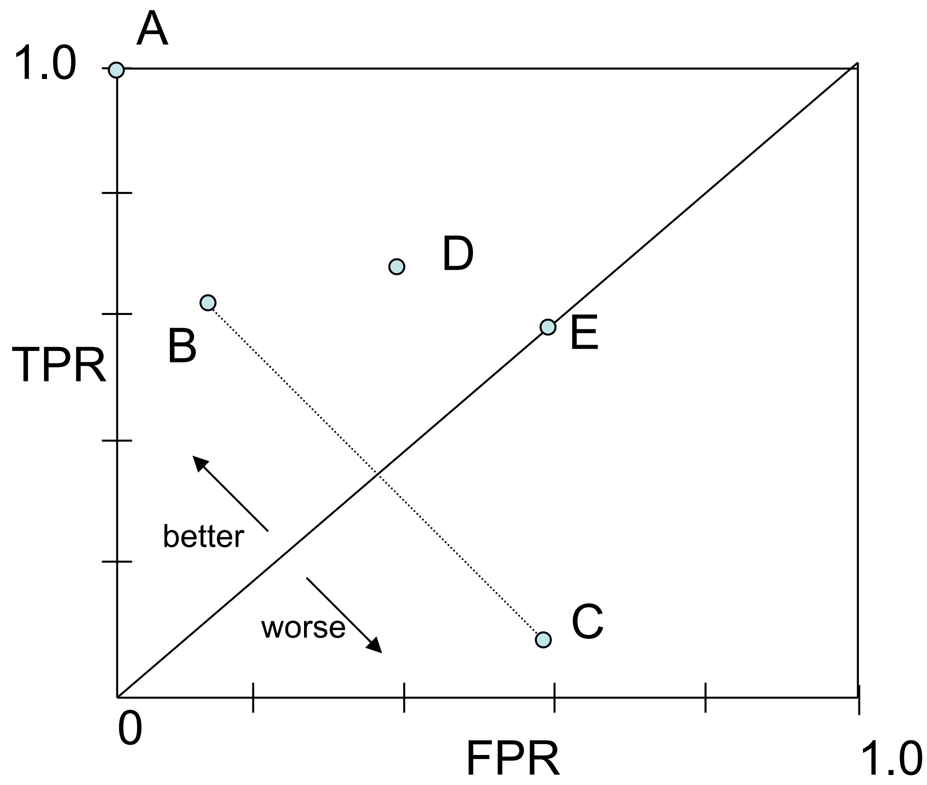
	Positive (Predicted)	Negative (Predicted)
Positive (Actual)	100	50
Negative (Actual)	150	9700

The dataset:
- 150 positive class instances
- 9850 negative class instances

Accuracy:
- Overall_acc = (100+9700)/(100+50+150+9700) = 0.98
- Recall = 100/(100+50) = 0.667
- Precision = 100/(100+150) = 0.4
- F1 = 2 x (0.667x0.4)/(0.667+0.4) = 0.5

ROC - Receiver Operating Characteristic

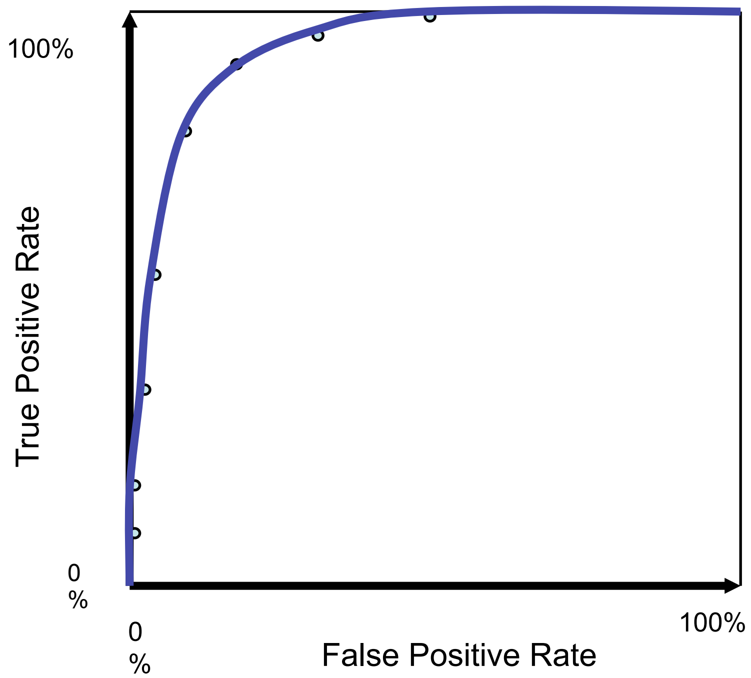
- Particularly a plot of TPR on y-axis against FPR on x axis is known as ROC
- A, B, C, D and E are five classifiers with different TPR and FPR values
- A is the ideal classifier because it has TPR = 1.0 and FPR = 0
- E is on the diagonal which stands for random guess
- C performs worse than random guess
  - But inverse of C which is B is better than D
- Classifiers should aim to be in the northwest



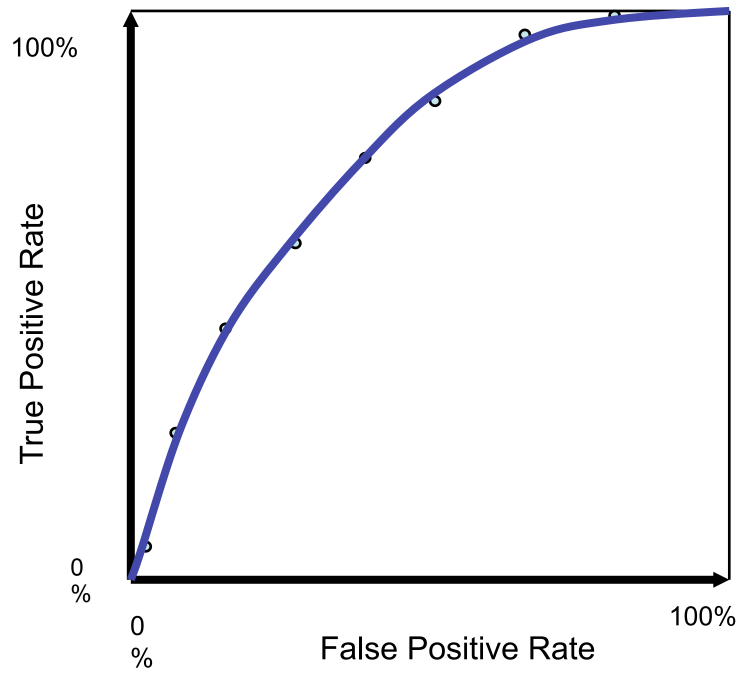
TPR = TP/(TP+FN)
FPR= FP/(FP+TN)

ROC curve comparison

A good test:



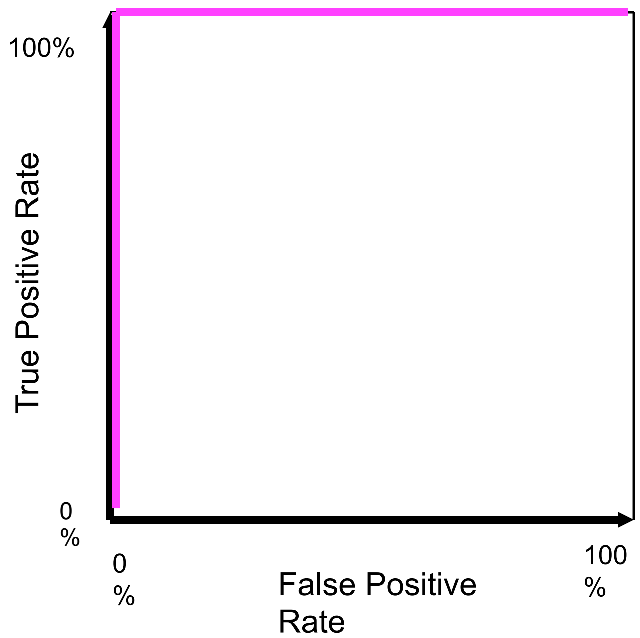
A poor test:



AUC: area under the curve

Summary

Best Test:



The distributions don’t overlap at all

Testing Classifier

- Testing the classifier on training data is not useful
  - Performance figures from such testing will be optimistic
  - Because the classifier is trained from the very data
- Ideally, a new data set called 'test set' needs to be used for testing
  - If test set is large performance figures will be more realistic
  - Creating test set needs experts' time and therefore creating large test sets is expensive
  - After testing, test set is combined with training data to produce a new classifier
  - Sometimes, a third data set called 'validation data' used for fine tuning a classifier or to select a classifier among many
- In practice several strategies used to make up for lack of test data
  - Holdout procedure – a certain proportion of training data is held as test data and remaining used for training
  - Cross-validation
  - Leave-one-out cross-validation

Testing Classifier 2

- Cross Validation
  - Partition the data into a fixed number of folds
  - Use data from each of the partitions for testing while using the remaining for training
  - Every instance is used for testing once
  - 10-fold cross-validation is standard, particularly repeating it 10 times
- Leave-one-out
  - Is n-fold cross-validation, where n is the data size
  - One instance is held for testing while using the remaining for training
  - Results from single instance tests are averaged to obtain the final test result
  - Maximum utilization of data for training
  - No sampling of data for testing, each instance is systematically used for testing
  - High costs involved because classifier is trained n times
  - Hard to ensure representative data for training

Summary

What you should know

- Accuracy
- Precision, recall, F-measure
- ROC curve
- when to use accuracy or F-measure
- Why you need test data
- Cross validation