Evaluating classification algorithms

CS434

Evaluation methods

- **Test set**: The available data set *D* is divided into two disjoint subsets,
  - the *training set* D_{train} (for learning a model)
  - the *test set* D_{test} (for testing the model)
- **Important:** training set should not be used in testing and the test set should not be used in learning in any way (including parameter tuning).
  - Unseen test set provides an unbiased estimate of accuracy.
- The test set is also called the holdout set
- This method is mainly used when the data set *D* is large

Evaluation methods (cont…)

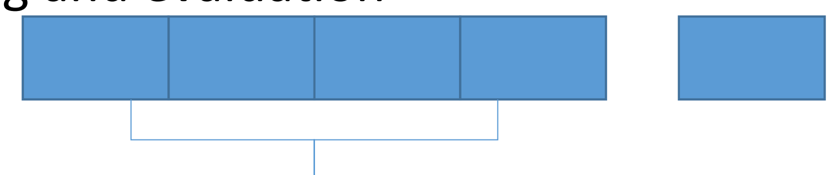
- **n-fold cross-validation for evaluation**: The available data is partitioned into *n* equal-size disjoint subsets
- Use each subset as the test set and combine the rest *n*-1 subsets as the training set to learn a classifier.
- The procedure is run *n* times, which give *n* accuracies.
- The final estimated accuracy of learning is the average of the *n* accuracies.
- 10-fold and 5-fold cross-validations are commonly used.
- This method is used when the available data is not large and we want to get a robust estimate of the performance

Evaluation methods (cont…)

- **Leave-one-out cross-validation**: This method is used when the data set is very small.
- It is a special case of cross-validation
- Each fold of the cross validation has only a single test example and all the rest of the data is used in training.

Evaluation methods (cont…)

- **Validation set for tuning parameters**: the available data is divided into three subsets,
  - a training set, a validation set and a test set.
- Validation set is used often to tune hyper-parameters (e.g., regularization parameter, c for SVM)
- In such cases, the values that give the best accuracy on the validation set are used as the final parameter values to estimate test data performance
- Nested cross-validation (see example below) can be used to do both parameter tuning and evaluation



Cross-validation within these 4 folds to decide the parameter (e.g. c for SVM), then apply the selected c to the 4 folds together to learn a model and predict and evaluate accuracy on fold 5. This process is repeated for five times for a nested 5-fold cross-validation

Classification performance measure

- Accuracy is only one commonly used measure (error = 1-accuracy).
- **Accuracy is not suitable in some applications**.
- In text mining, we may only be interested in the documents of a particular topic, which are only a small portion of a big document collection.
- In classification involving skewed or highly imbalanced data, e.g., network intrusion and financial fraud detections, we are interested only in the minority class.
  - High accuracy does not mean any intrusion is detected.
  - E.g., 1% intrusion. Achieve 99% accuracy by doing nothing.
- The class of interest is commonly called the **positive class**, and the rest **negative classes.**

Precision and **recall** measures

- Used in information retrieval and text classification.
- We use a confusion matrix to introduce them.

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

where

TP: the number of correct classifications of the positive examples (**true positive**),

FN: the number of incorrect classifications of positive examples (**false negative**),

FP: the number of incorrect classifications of negative examples (**false positive**), and

TN: the number of correct classifications of negative examples (**true negative**).

Precision and recall measures (cont…)

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

$$p = \frac{TP}{TP + FP}. \qquad r = \frac{TP}{TP + FN}.$$

- Precision *p* is the number of correctly classified positive examples divided by the total number of examples that are classified as positive.
- Recall *r* is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

An example

	Classified Positive	Classified Negative
Actual Positive	1	99
Actual Negative	0	1000

- This confusion matrix gives
  - precision p = 100% and
  - recall r = 1%

  because we only classified one positive example correctly and no negative examples wrongly.
- Note: precision and recall only measure classification on the positive class.

F_1-value (also called F_1-score)

- It is hard to compare two classifiers using two measures. F_1 score combines precision and recall into one measure

$$F_1 = \frac{2pr}{p + r}$$

F_1-score is the harmonic mean of precision and recall.

$$F_1 = \frac{2}{\frac{1}{p} + \frac{1}{r}}$$

- The harmonic mean of two numbers tends to be closer to the smaller of the two.
- For F_1-value to be large, both *p* and *r* much be large.

Receive operating characteristics curve

- It is commonly called the ROC curve.
- It is a plot of the true positive rate (TPR) against the false positive rate (FPR).
- True positive rate:

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

Total number of ground-truth positives

- False positive rate:

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

Total number of ground-truth negatives

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Example ROC curves

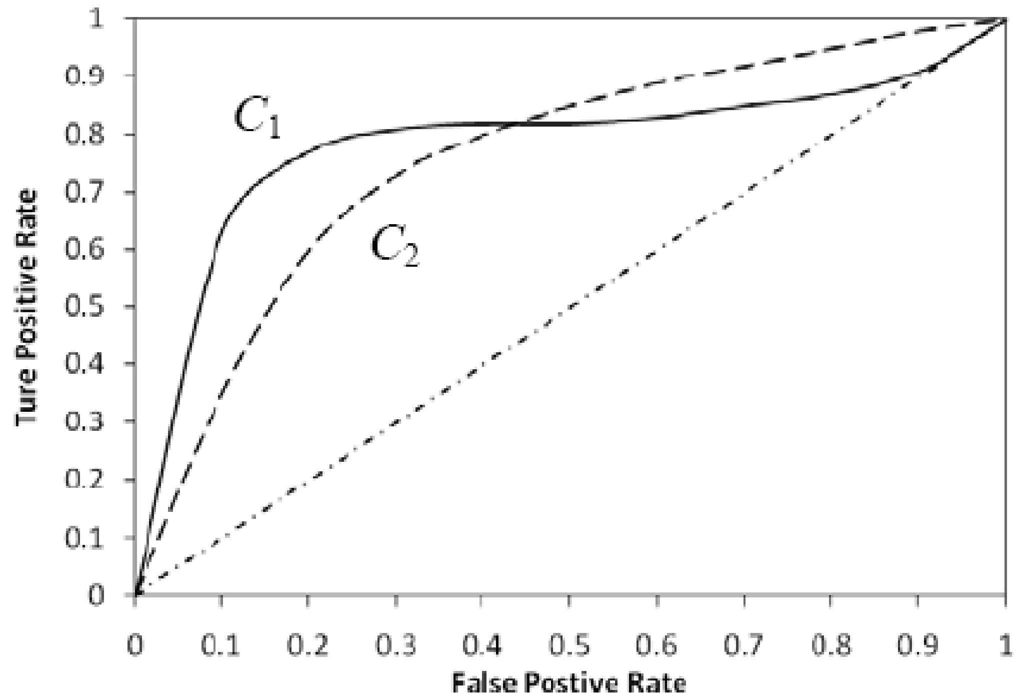


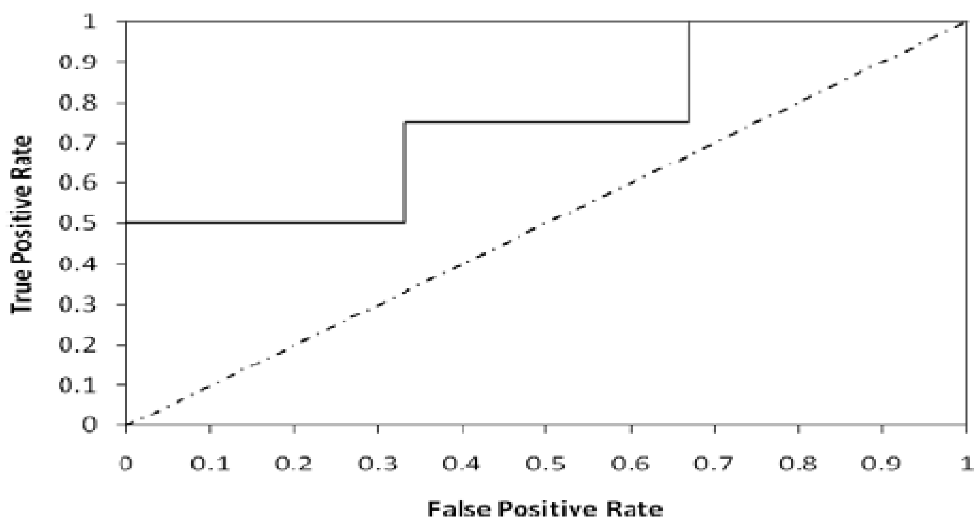
Fig. 3.8. ROC curves for two classifiers (C_1 and C_2) on the same data

Area under the curve (AUC)

- Which classifier is better, C_1 or C_2?
  - It depends on which region your classifier will be operating in
- Can we have one measure?
  - Yes, we compute the area under the curve (AUC)
- If AUC for C_i is greater than that of C_j, it is said that C_i is better than C_j.
  - If a classifier is perfect, its AUC value is 1
  - If a classifier makes all random guesses, its AUC value is 0.5.

Drawing an ROC curve

Rank		1	2	3	4	5	6	7	8	9	10
Actual class		+	+	–	–	+	–	–	+	–	–
TP	0	1	2	2	2	3	3	3	4	4	4
FP	0	0	0	1	2	2	3	4	4	5	6
TN	6	6	6	5	4	4	3	2	2	1	0
FN	4	3	2	2	2	1	1	1	0	0	0
TPR	0	0.25	0.5	0.5	0.5	0.75	0.75	0.75	1	1	1
FPR	0	0	0	0.17	0.33	0.33	0.50	0.67	0.67	0.83	1



Key points for appropriate evaluation

- Different applications may require different evaluation criteria
  - Log-likelihood
  - Accuracy
  - Precision, recall, F1
  - ROC curve
  - Area under ROC curve
- To claim good performance for your algorithm
  - Need proper set up for evaluation – do not ever train or tune on test data
  - Unless you have large amounts of data, random repetitions are required to show that your performance is not due to random chance
    - Repeat Cross-validation multiple times
    - Randomly split the data into training and testing multiple times
    - Report not only the mean but also the variance around it