CS699
Lecture 6
Performance Evaluation

Model Evaluation and Selection

- Evaluation metrics: How can we measure accuracy? Other metrics to consider?
- Use an **independent test dataset** instead of training dataset when assessing accuracy
- Methods for estimating a classifier's accuracy:
  - Holdout method, random subsampling
  - Cross-validation
  - Bootstrap
- Comparing classifiers:
  - Confidence intervals
  - ROC Curves

Model Evaluation and Selection

- Model testing

Dataset with known class

predicts
(or classifies)

MODEL

Id	A1	A2	A3	A4	Actual Class	Predicted Class	
1					Y	Y	TP
2					Y	N	FN
3					N	N	TN
4					Y	N	FN
5					N	N	TN
6					N	N	TN
7					N	Y	FP
8					Y	Y	TP
9					N	N	TN
10					N	N	TN

attribute values

Assume:
Y is positive
N is negative
(application dependent)

7 out of 10 were correctly classified:

accuracy = 70%

Classifier Evaluation Metrics: Confusion Matrix

Confusion Matrix:

Actual class\Predicted class	Positive	Negative
Positive	**True Positives (TP)**	**False Negatives (FN)**
Negative	**False Positives (FP)**	**True Negatives (TN)**

- Given *m* classes, an entry, $\boldsymbol{CM_{i,j}}$ in a **confusion matrix** indicates # of tuples in class *i* that were labeled by the classifier as class *j*
- May have extra rows/columns to provide totals
- Confusion matrix of the example in the previous slide

Actual class\Predicted class	Y	N	Total
Y	2	2	4
N	1	5	6
Total	3	7	10

Classifier Evaluation Metrics: Confusion Matrix

Another Example:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

C: buys_computer = yes (P)
¬C: buys_computer = no (N)

- **Classifier Accuracy,** or recognition rate: percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All

- **Error rate:** *1 – accuracy*, or

  Error rate = (FP + FN)/All

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

- **Class Imbalance Problem**:
  - One class may be *rare*, e.g. fraud, or HIV-positive
  - Significant majority of the negative class and minority of the positive class
  - Example: 9900 N's (no cancer) and 100 P's (cancer)
  - A model correctly classifies all N's and 20 P's
  - Model accuracy = 9920 / 10000 = 99.2% => very high
  - But, 80% of cancer patients were misdiagnosed as non-cancer patient

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

- If it is important to correctly diagnose cancer patients, we need a different measure.
- In this example, we need TP/P.

- **Sensitivity**: True Positive recognition rate
  - **Sensitivity = TP/P**

- **Specificity**: True Negative recognition rate
  - **Specificity = TN/N**

Classifier Evaluation Metrics: Precision and Recall, and F-measures

- **Precision**: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

- **Recall:** completeness – what % of actual positive tuples did the classifier label as positive?

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

- Recall is the same as sensitivity
- Perfect score is 1.0 (for both)

Classifier Evaluation Metrics: Precision and Recall, and F-measures

- Combine precision and recall into a single measure
- ***F* measure (*F_1* or *F*-score)**: harmonic mean of precision and recall

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- ***F_β*:** weighted measure of precision and recall
  - ß is a parameter
  - assigns ß times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

Classifier Evaluation Metrics: Example

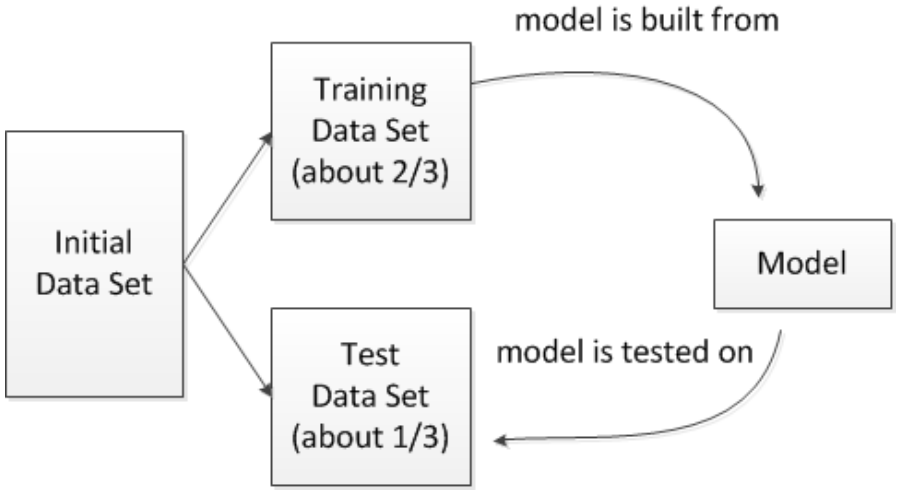
Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*
cancer = no	**140**	**9560**	9700	98.56 (*specificity)*
Total	230	9770	10000	96.50 (*accuracy*)

- *Precision* = 90/230 = 39.13% *Recall* = 90/300 = 30.00%

- *F* (or F_1 or *F*-score) = (2 * 0.39 * 0.3) / (0.39 + 0.3) = 0.339
- F_2 = [(1 + 2^2) * 0.39 * 0.3] / (2^2 * 0.39 + 0.3) = 0.315

Evaluating Classifier Accuracy: Holdout Method

- **Holdout method**
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation



  - Random subsampling: a variation of holdout
    - Repeat holdout *k* times, accuracy = avg. of the *k* accuracies obtained

Evaluating Classifier Accuracy: Cross-Validation Method

- **Cross-validation** (*k*-fold, where *k* = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, D_1, D_2, …, D_k, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set

    Iter. 1: {D_2, …, D_{10}} used for training, D_1 used for testing
    Iter. 2: {D_1, D_3,…, D_{10}} used for training, D_2 used for testing
    …
    Iter. 10: {D_1, …, D_9} used for training, D_{10} used for testing

    Accuracy = (# tuples correctly classified) / (total # tuples)

  - Leave-one-out: *k* = # of tuples, each fold has one sample, for small sized data
  - Stratified cross-validation: folds are stratified so that class distribution in each fold is approx. the same as that in the initial data

Evaluating Classifier Accuracy: Bootstrap

- **Bootstrap**
  - Works well with small data sets
  - Samples the given training tuples uniformly *with replacement*
    - i.e., each time a tuple is sampled it is re-added to the training set
- Several bootstrap methods, and a common one is **.632 boostrap**
  - A data set with *d* tuples is sampled *d* times, with replacement, resulting in a training set of *d* samples.
  - Since samples are replaced, the same tuple can be sampled multiple times.
  - So, the training sample can have duplicates.
  - The data tuples that did not make it into the training set end up forming the test set.

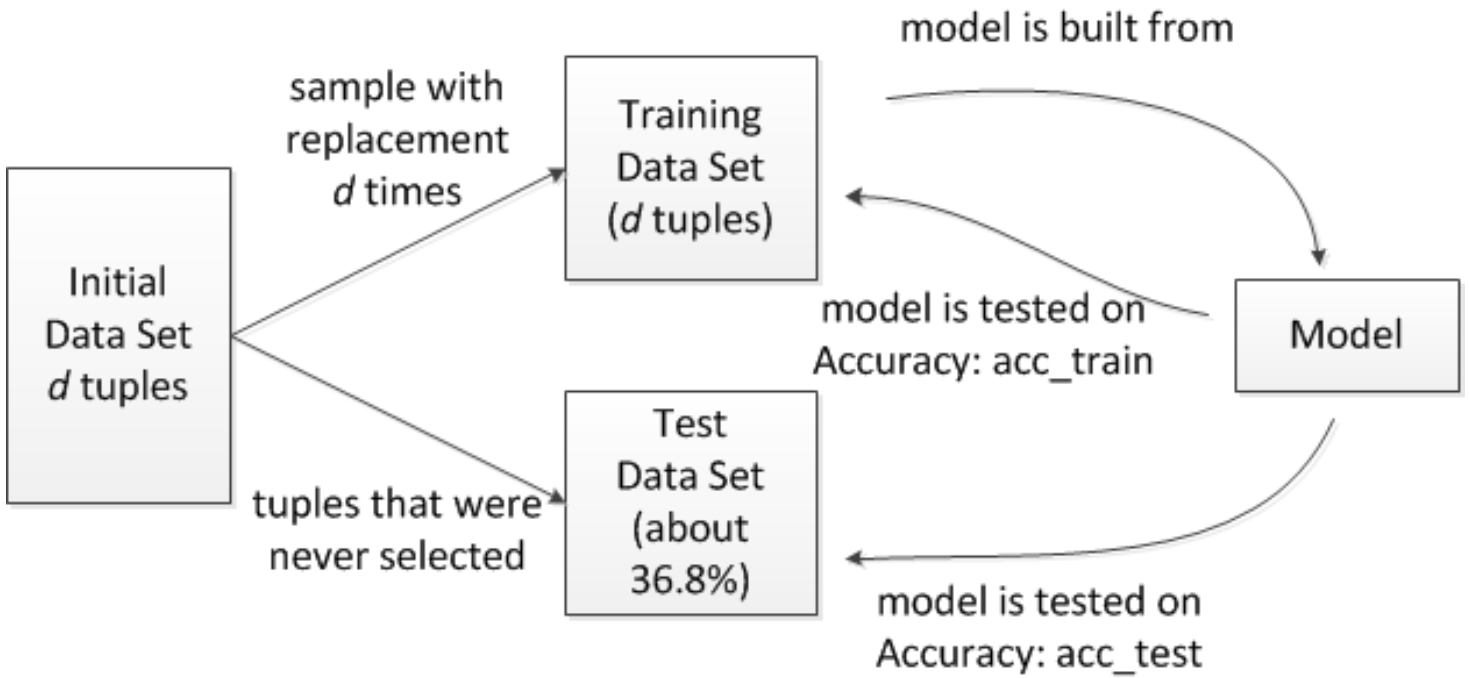
Evaluating Classifier Accuracy: Bootstrap

- **Bootstrap**
  - About 63.2% of the original data end up in the training set (which is also referred to as **bootstrap**), and the remaining 36.8% form the test set (since $(1 - 1/d)^d \approx e^{-1} = 0.368$, for a large d)
  - Model is built from the training dataset and it is tested on both training dataset and test dataset
  - Accuracy of the model is computed by combining two accuracies (see next slide)
  - This is repeated k times, and overall accuracy of the model is computed as:

$$Acc(M) = \frac{1}{k}\sum_{i=1}^{k}(0.632 \times Acc(M_i)_{test\_set} + 0.368 \times Acc(M_i)_{train\_set})$$

Evaluating Classifier Accuracy: Bootstrap

- Bootstrap illustration:



accuracy = 0.632 * acc_test + 0.368 * acc_tain

Comparing Models M_1 vs. M_2: Hypothesis Testing

- Suppose we have 2 classifiers, M_1 and M_2, which one is better?
- Use 10-fold cross-validation to obtain $\overline{err}(M_1)$ and $\overline{err}(M_2)$
- These mean error rates are just *estimates*
- We want to know whether the difference between the 2 error rates is attributed to just a *chance* or *statistically significant*.
  - Use a **test of statistical significance**

Hypothesis Testing

- Hypothesis Testing or Testing of Statistical Significance
  - Will describe two-sided testing
  - State null hypothesis (and alternative hypothesis)
  - Choose the significance level
  - Compute test statistic X
  - Find the critical value C from the distribution table
  - If $X > C$ or $X < -C$, reject the null hypothesis.

    Otherwise, fail to reject the null hypothesis.

Hypothesis Testing

- Perform 10-fold cross-validation
- Use **t-test** (or **Student's t-test**)

  with the degrees of freedom = $k - 1$ ($k = 10$ for our case)
- **Null Hypothesis**: M_1 & M_2 are the same
- If we can **reject** null hypothesis, then
  - we conclude that the difference between M_1 & M_2 is **statistically significant**
  - Choose model with lower error rate
- If we cannot reject null hypothesis, we conclude that the difference is by chance (or not statistically significant).

Hypothesis Testing: t-test

- When only 1 test set available: **pairwise comparison**
  - Perform 10-fold cross-validation for M_1 and M_2
  - For each iteration, the same partitions (i.e., the same training set and the same test set) are used for both M_1 and M_2.
  - For each iteration, error rate of each classifier is computed.
  - For example

Iteration No.	$err(M1)_i$	$err(M2)_i$
1	0.023	0.05
2	0.12	0.067
...	...	...

Hypothesis Testing: t-test

- Average over 10 rounds to get $\overline{err}(M_1)$ and $\overline{err}(M_2)$
- Compute **t-statistic** with *k-1* **degrees of freedom:**

$$t_0 = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{\text{var}(M_1 - M_2)/k}}$$, where

$$var(M_1 - M_2) = \frac{1}{k}\sum_{i=1}^{k}\left[err(M_1)_i - err(M_2)_i - (\overline{err}(M_1) - \overline{err}(M_2))\right]^2$$

- Note: This formula from the textbook calculates a population variance (the denominator is k). However, in the example in a later slide, a sample variance will be used.

Hypothesis Testing: t-test

- We choose a significance level α
- For example, significance level $\alpha = 0.05$ (or 5%)
- From the *t*-distribution table, we find *t* value, $t_{\alpha/2,\, df}$, corresponding to $\alpha/2$ (t-distribution is symmetric; typically upper % points of distribution shown) and degrees of freedom $df = k - 1$. This is the critical value.
- For 10-fold cross-validation, $k = 10$.
- Example: if we choose $\alpha = 0.05$, we find $t_{0.025,\, 9}$ from the *t*-distribution table (see next slide)

Hypothesis Testing: t-test

TABLE B: *t*-DISTRIBUTION CRITICAL VALUES

	Tail probability p									
df	.25	.20	.15	.10	.05	.025	.02	.01	.005	.00
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	12
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.4
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.5
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.7
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.3
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.0
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.8
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.6
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.5
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.4
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.4
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.3
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.3
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.2
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.2
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.2
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.1

$t_{0.025,\ 9} = 2.262$

Hypothesis Testing: t-test

- Test
  - **If $t_0 > t_{\alpha/2,\, df}$ or $t_0 < -t_{\alpha/2,\, df}$,** then t value lies in the rejection region:
    - Reject null hypothesis
    - Conclude the difference between M_1 and M_2 is **statistically significant**
    - We choose the one with a lower error rate
  - **Otherwise**, conclude that any difference is **chance**

Hypothesis Testing: t-test

E1	E2	E1-E2		
0.12	0.03	0.09		mean(E1-E2)
0.18	0.03	0.15		0.057
0.14	0.02	0.12		var(E1-E2), sample variance
0.21	0.25	-0.04		0.005245556
0.21	0.18	0.03		SQRT(var/k)
0.09	0.01	0.08		0.022903178
0.18	0.07	0.11		t=mean(E1-E2)/SQRT(var/k)
0.15	0.05	0.1		2.488737606
0.16	0.17	-0.01		
0.03	0.09	-0.06		
0.147	0.09	0.057		

t = mean(E1-E2) / SQRT(var(E1-E2) / k) = 2.489

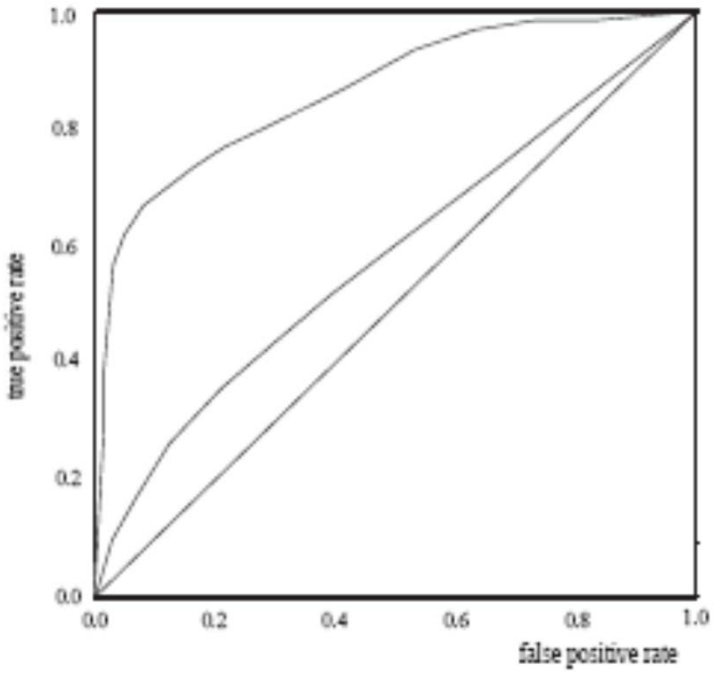
$t_{0.025,9}$ = 2.262 (from t-distribution table)

Since $t > t_{0.025,9}$, we reject the null hypothesis and conclude M_2 is better than M_1.

Note: A sample variance is used for var(E1 – E2)

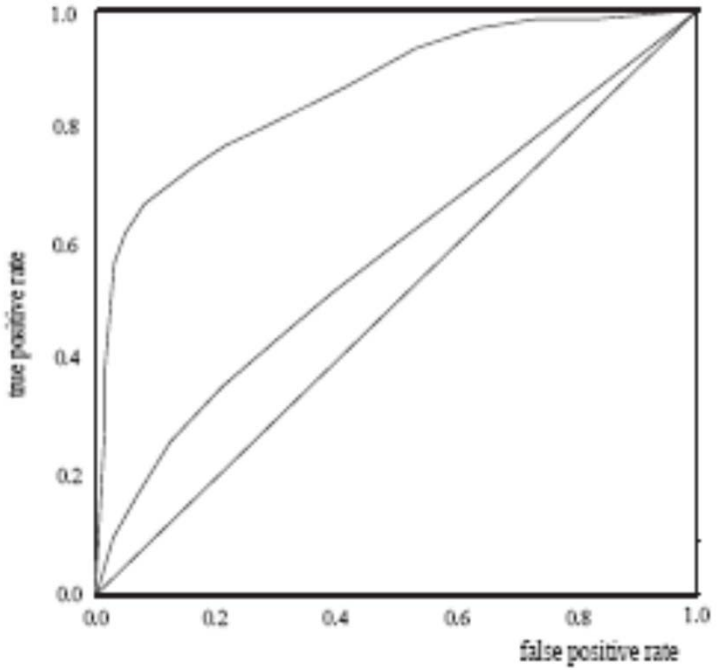
ROC Curves

- **ROC** (Receiver Operating Characteristics) curves: for visual comparison of classification models
- Originated from signal detection theory
- Shows the trade-off between the true positive rate and the false positive rate
- The area under the ROC curve is a measure of the accuracy of the model



ROC Curves

- Vertical axis represents the true positive rate (TPR)
- Horizontal axis represents the false positive rate (FPR)
- The plot also shows a diagonal line
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model
- A model with perfect accuracy will have an area of 1.0



ROC Curves

- Rank the test tuples in decreasing order of predicted probability: the one that is most likely to belong to the positive class appears at the top of the list.

tuple_id	Actual class	Probability
1	P	0.992
2	P	0.964
3	N	0.953
4	P	0.931
5	P	0.893
6	N	0.875
7	P	0.82
8	N	0.793
9	N	0.778
10	P	0.742

ROC Curves

- For each row:
  - Assume all tuples above, including itself, are classified as positive and all tuples below are classified as negative
  - Calculate TP, FP, TN, FN and TPR (= TP/P) and FPR = (FP/N)
  - Then, plot TPR vs. FPR

ROC Curves

P = 6, N = 4

tuple_id	Actual class	Probability	TP	FP	TN	FN	TPR	FPR
1	P	0.992	1	0	4	5	0.17	0
2	P	0.964	2	0	4	4	0.33	0
3	N	0.953	2	1	3	4	0.33	0.25
4	P	0.931	3	1	3	3	0.5	0.25
5	P	0.893	4	1	3	2	0.67	0.25
6	N	0.875	4	2	2	2	0.67	0.5
7	P	0.82	5	2	2	1	0.83	0.5
8	N	0.793	5	3	1	1	0.83	0.75
9	N	0.778	5	4	0	1	0.83	1.0
10	P	0.742	6	4	0	0	1.0	1.0

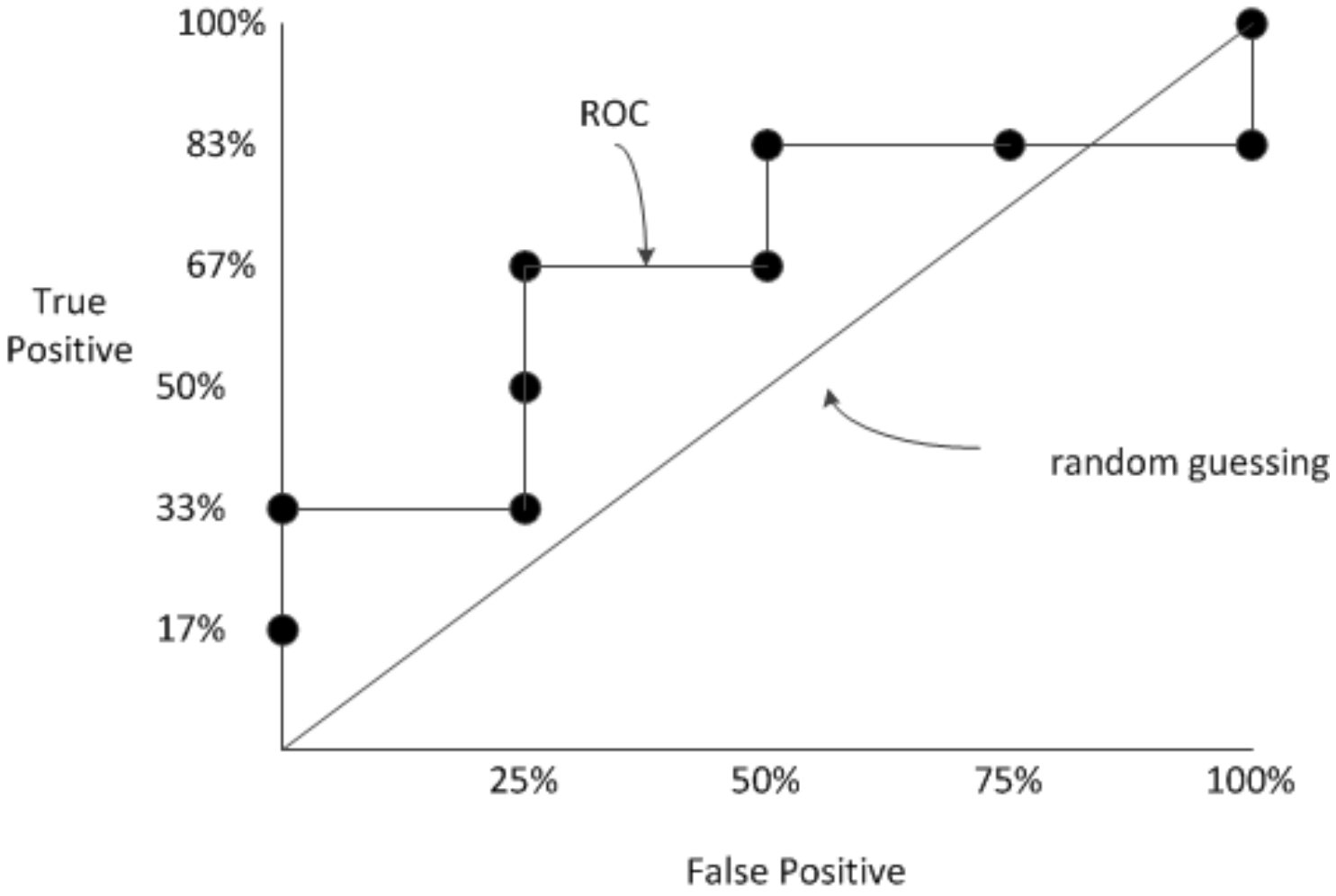
ROC Curves

tuple_id	Actual class	Probability	Classified as	TP	FP	TN	FN	TPR	FPR
1	P	0.992	P	1	0	4	5	0.17	0
2	P	0.964	P	2	0	4	4	0.33	0
3	N	0.953	P	2	1	3	4	0.33	0.25
4	P	0.931	P	3	1	3	3	0.5	0.25
5	P	0.893	N	4	1	3	2	0.67	0.25
6	N	0.875	N	4	2	2	2	0.67	0.5
7	P	0.82	N	5	2	2	1	0.83	0.5
8	N	0.793	N	5	3	1	1	0.83	0.75
9	N	0.778	N	5	4	0	1	0.83	1.0
10	P	0.742	N	6	4	0	0	1.0	1.0

Fourth row:

- Top four are classified as P and all others are classified as N.
- TP is 3 (3 positives are correctly classified as P)
- FP is 1 (1 negative is incorrectly classified as P)
- TN is 3 (3 negatives are correctly classified as N)
- FN is 3 (3 positives are incorrectly classified as N)
- *TPR* = 3 / 6 = 0.5 and *FPR* = 1 / 4 = 0.25

ROC Curves


100%
83%
67%
50%
33%
17%
True
Positive
ROC
random guessing
25%
50%
75%
100%
False Positive

Issues Affecting Model Selection

- **Accuracy**
  - classifier accuracy: predicting class label
- **Speed**
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)
- **Robustness**: handling noise and missing values
- **Scalability**: efficiency in disk-resident databases
- **Interpretability**
  - understanding and insight provided by the model
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

Performance Measures for Numeric Prediction

- Will discuss measures on Weka:
  - Correlation coefficient
  - Mean Absolute error
  - Root mean squared error
  - Relative absolute error
  - Root relative squared error

```
Correlation coefficient                  0.6115
Mean absolute error                     61.8725
Root mean squared error                127.1763
Relative absolute error                 64.2904 %
Root relative squared error             79.0639 %
Total Number of Instances              209
```

Performance Measures for Numeric Prediction

- **Notations:**
  - $a_1, a_2, \ldots, a_n$: Actual values of the dependent variable (also called output attribute or class attribute)
  - $p_1, p_2, \ldots, p_n$: Predicted values (by a prediction/classifier algorithm)

- Performance measures represent how far the predicted values are from the actual attribute values

Performance Measures for Numeric Prediction

- Correlation coefficient
  - Correlation between *a*'s and *p*'s
  - Between -1 and 1

$$\frac{S_{PA}}{\sqrt{S_P S_A}}, \text{ where}$$

$$S_{PA} = \frac{\sum_{i=1}^{n}(p_i - \bar{p})(a_i - \bar{a})}{n-1},$$

$$S_P = \frac{\sum_{i=1}^{n}(p_i - \bar{p})^2}{n-1}, \qquad S_A = \frac{\sum_{i=1}^{n}(a_i - \bar{a})^2}{n-1}$$

Performance Measures for Numeric Prediction

- Mean absolute error
  - Average of the magnitude of individual errors
  - Less affected by outliers

$$\frac{|a_1 - p_1| + |a_2 - p_2| + \ldots + |a_n - p_n|}{n} \text{ or } \frac{\sum_{i=1}^{n} |a_i - p_i|}{n}$$

Performance Measures for Numeric Prediction

- Root mean squared error
  - Square root of the average of the squared individual errors
  - Effect of outliers is exaggerated.

$$\sqrt{\frac{(a_1 - p_1)^2 + (a_2 - p_2)^2 + \ldots + (a_n - p_n)^2}{n}} \quad \text{or} \quad \sqrt{\frac{\sum_{i=1}^{n}(a_i - p_i)^2}{n}}$$

Performance Measures for Numeric Prediction

- Relative absolute error
  - Absolute error is normalized by the error that would have been generated when a simple predictor had been used. The average of actual attribute values of *a*'s is used as the simple predictor.

$$\frac{|a_1 - p_1| + |a_2 - p_2| + \ldots + |a_n - p_n|}{|a_1 - \bar{a}| + |a_2 - \bar{a}| + \ldots + |a_n - \bar{a}|} \quad or \quad \frac{\sum_{i=1}^{n} |a_i - p_i|}{\sum_{i=1}^{n} |a_i - \bar{a}|}$$

Performance Measures for Numeric Prediction

- Root relative squared error
  - Square root of relative squared error

$$\sqrt{\frac{(a_1 - p_1)^2 + (a_2 - p_2)^2 + \ldots + (a_n - p_n)^2}{(a_1 - \bar{a})^2 + (a_2 - \bar{a})^2 + \ldots + (a_n - \bar{a})^2}} \quad or \quad \sqrt{\frac{\sum_{i=1}^{n}(a_i - p_i)^2}{\sum_{i=1}^{n}(a_i - \bar{a})^2}}$$

Performance Measures for Numeric Prediction

- Which measure is best?
- Should be determined by the application where the prediction is used.
- For many practical problems, the best prediction method is still the best regardless of which performance measure is used.

References

- Han, J., Kamber, M., Pei, J., “Data mining: concepts and techniques,” 3rd Ed., Morgan Kaufmann, 2012
- http://www.cs.illinois.edu/~hanj/bk3/
- I.H. Witten and E. Frank, "Data Mining Practical Machine Learning and Techniques," Second Edition, 2005, Morgan Kaufmann, pp. 176 – 179