University of Colorado Boulder

CSCI 4502/5502

Data Mining

Fall 2019
Lecture 12 (Oct 3)

Reminders

- Due at 9:30am, Thursday, Oct 3
  - project announcement
  - project summary slides
  - team availability poll
- Due at 9:30am, Thursday, Oct 10
  - project proposal

Course Project Schedule

- Project proposal (Week 7: 10/8, 10/10)
  - Data, subtasks, evaluation
  - No regular lecture on Tu or Th
- Project checkpoint (Week 12: 11/12, 11/14)
- Fall Break (Week 14: 11/26, 11/28)
- Project final report (Week 16: 12/10, 12/12)

Announcements

- Homework 4
  - will be posted on Thursday, Oct 10
  - due at 9:30am, Thursday, Oct 17
- Midterm schedule
  - Week 8 & 9: classification, clustering, outlier
  - Week 10: midterm review, sample exam questions
  - Thursday, Oct 31: midterm exam

Review

- Chapter 8: Classification: Basic Concepts
  - basic concepts
    - classification vs. prediction
    - supervised vs. unsupervised learning
  - decision tree induction
    - attribute selection, attribute split
    - information gain, gain ratio, gini index

Naïve Bayesian Classifier (1)

- X = (x_1, x_2, ..., x_n) (i.e., n attributes)
- m classes: C_1, C_2, ..., C_m
- Classification: maximal $P(C_i|X)$
- Based on Bayes' Theorm

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

- Since P(X) is constant for all classes, only need to maximize $P(X|C_i)P(C_i)$

Naïve Bayesian Classifier (2)

- Naïve assumption: class conditional independence (no dependence between attributes)

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i)$$

- If A_k is categorical, $P(x_k|C_i)$
- If A_k is continuous-valued, assume Gaussian distribution, $P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2)}{2\sigma^2}}$$

Chap 8: Classification

- Basic concepts
- Decision tree induction
- Bayesian classification
- Rule-based classification
- Model evaluation and selection
- Improve classification accuracy
- Summary

IF-THEN Rules

- IF condition THEN conclusion
  - R: IF age = youth AND student = yes
    THEN buys_computer = yes
  - rule antecedent/precondition (IF)
  - rule consequent (THEN)
- Assessment of a rule
  - coverage(R) = n_{covers} / |D|
  - accuracy(R) = $n_{correct}$ / n_{covers}

Rule Assessment Example

- R: IF age <= 30 AND student = yes THEN buys_computer = yes

- coverage (R) = n_{covers} / |D| =2/14

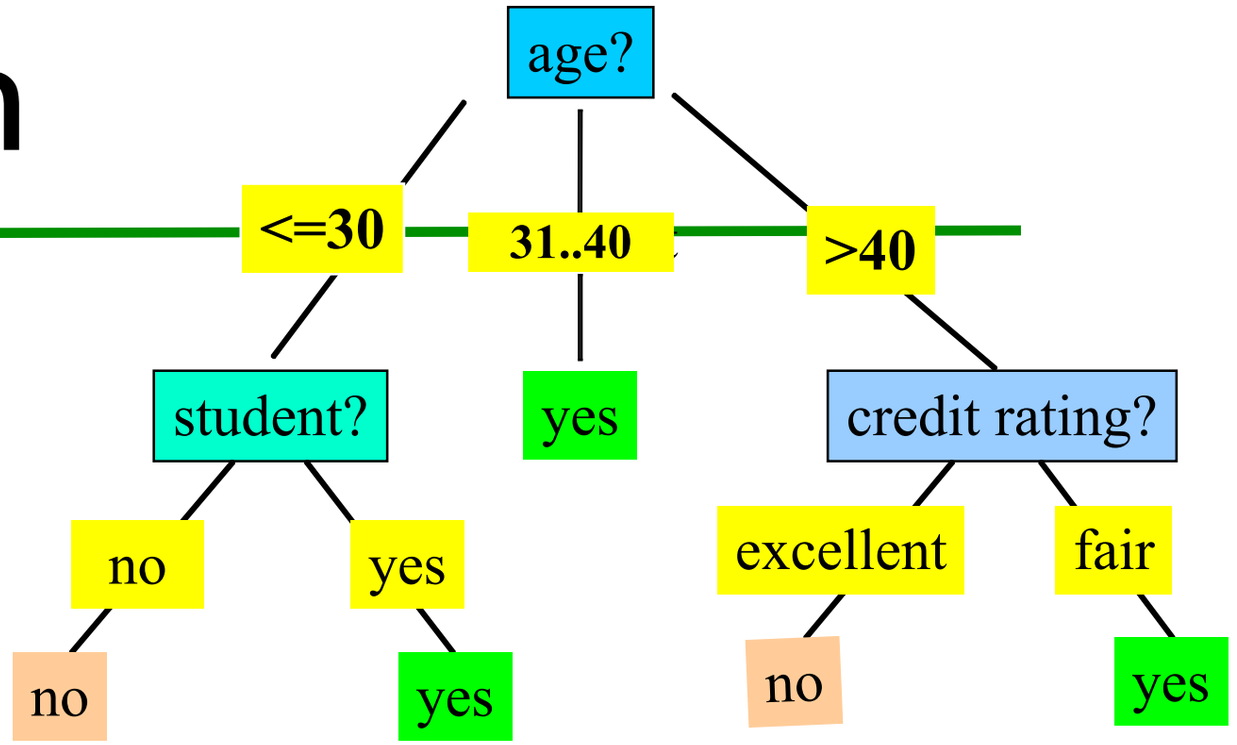
- accuracy (R) = $n_{correct}$ / n_{covers} =2/2

CID	age	income	student	credit_rating	buys_computer
1	<= 30	high	no	fair	no
2	<= 30	high	no	excellent	no
3	31-40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31-40	low	yes	excellent	yes
8	<= 30	medium	no	fair	no
9	<= 30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<= 30	medium	yes	excellent	yes
12	31-40	medium	no	excellent	yes
13	31-40	high	yes	fair	yes
14	>40	medium	no	excellent	no

Rule-Based Classification

- Rule R is triggered (precondition satisfied)
- R is the only rule triggered
- No rule is triggered
- More than one rules are triggered
  - size ordering: most attribute tests
  - class-based ordering: importance (e.g., prevalence, misclassification cost)
  - rule-based ordering: (decision list) priority list ordered by rule quality or by experts

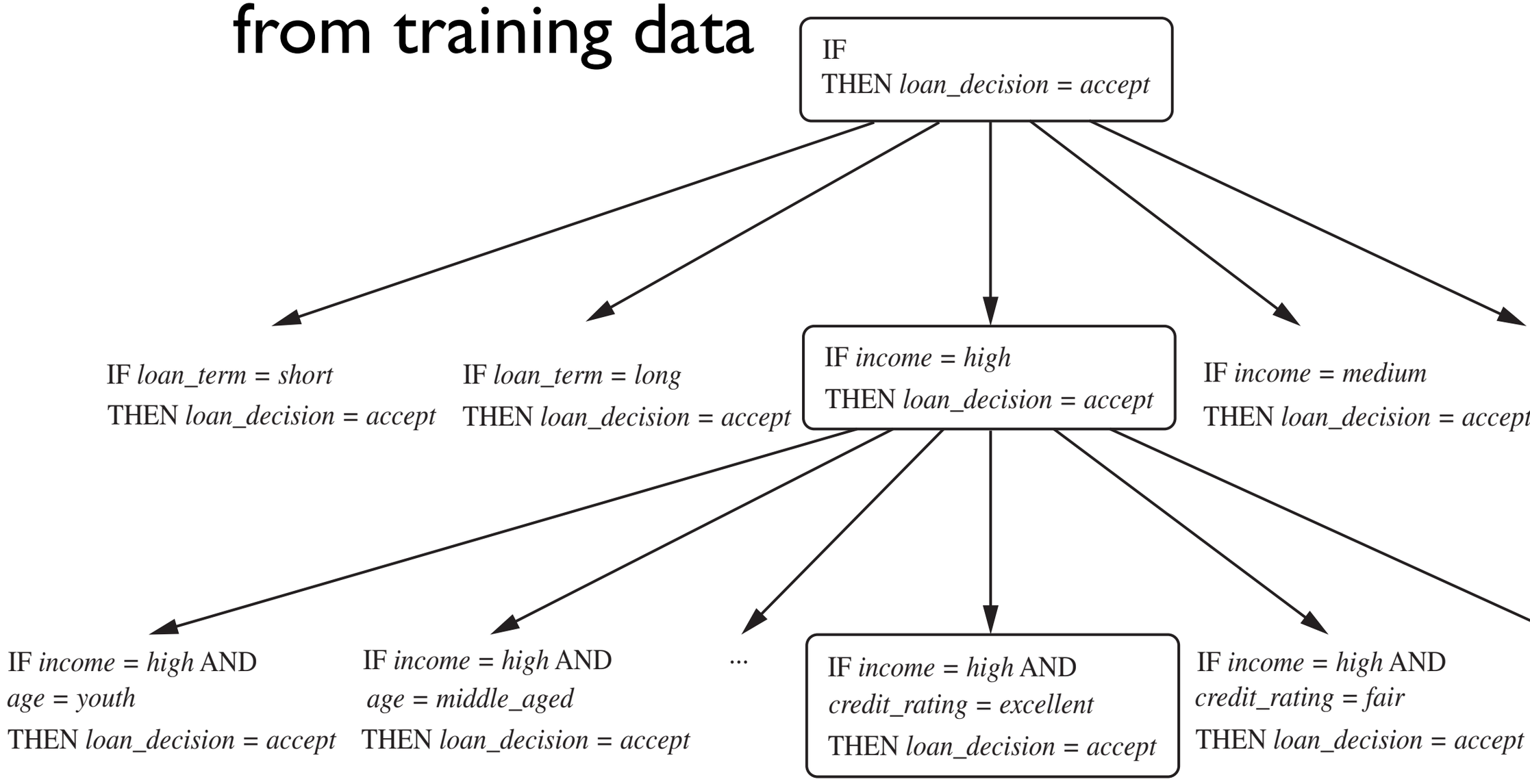
Rule Extraction



- From a decision tree
- Each root to leaf path
- Leaf: class prediction
- Rules are exhaustive and mutually exclusive
  - IF age=31...40 THEN buys_computer=yes
  - IF age<=30 AND student=no THEN buys_computer=no
  - IF age>40 AND credit_rating=fair THEN buys_computer = yes

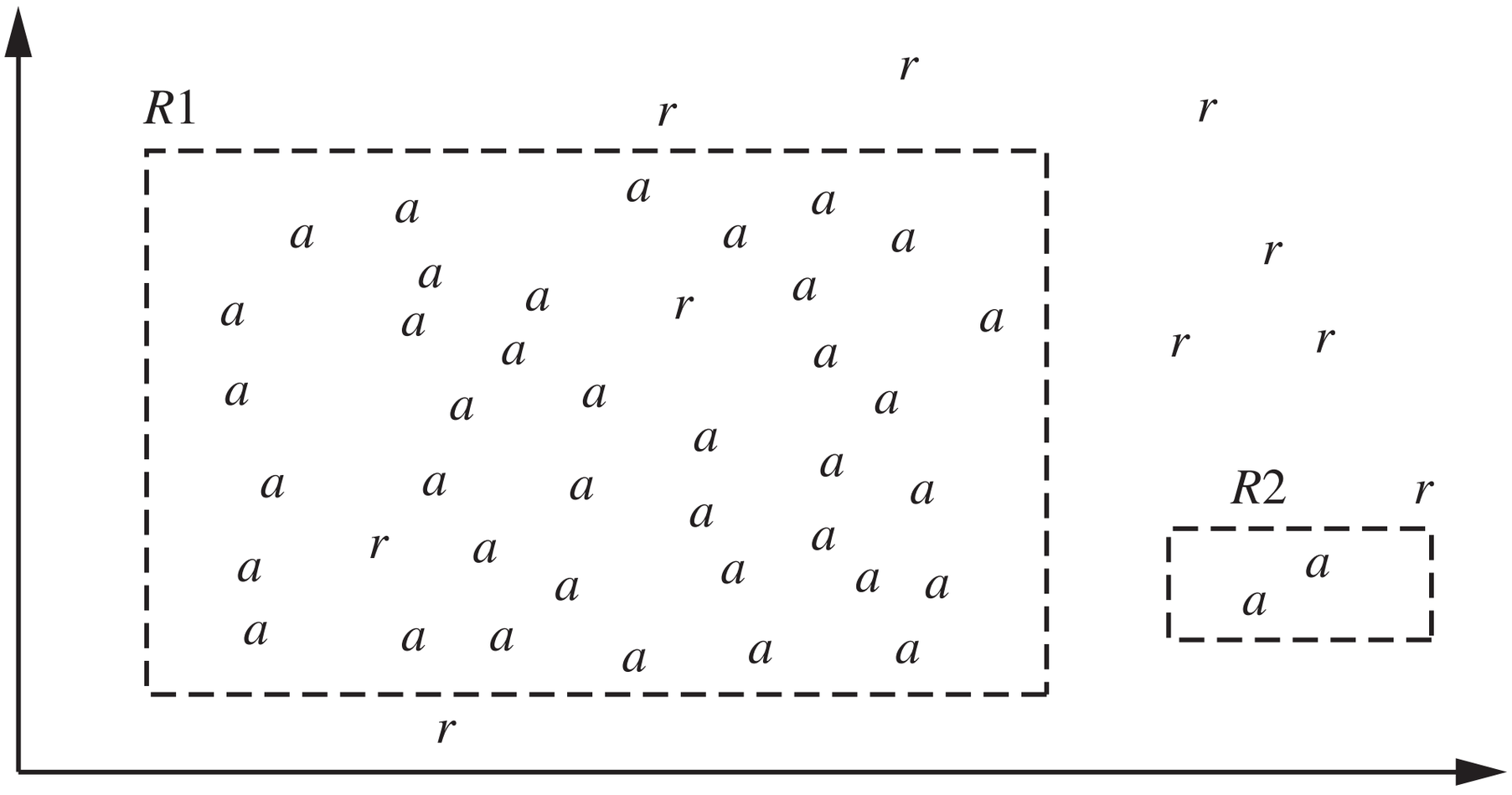
Example

- Greedy depth-first strategy: Learn directly from training data



Rule Quality Measures (1)

- Coverage?
- Accuracy?



Rule Quality Measures (2)

- Consider both coverage and accuracy
- # positive (negative) tuples covered by a rule

$$FOIL\_Gain = pos' \times \left(\log_2 \frac{pos'}{pos' + neg'} - \log_2 \frac{pos}{pos + neg} \right)$$

- Observed frequency vs. expected frequency

$$Likelihood\_Ratio = 2 \sum_{i=1}^{m} f_i \log \left(\frac{f_i}{e_i} \right)$$

- Rule pruning

$$FOIL\_Prune(R) = \frac{pos - neg}{pos + neg}$$

Chapter 8: Classification

- Basic concepts
- Decision tree induction
- Bayesian classification
- Rule-based classification
- Model evaluation and selection
- Improve classification accuracy
- Summary

Classifier Accuracy Measures

- Partition: training data and testing data
- Accuracy, recognition rate
- Error rate, misclassification rate
- Confusion matrix

		Predicted class	
		C_1	C_2
Actual class	C_1	true positives	false negatives
	C_2	false positives	true negatives

Classes	*buys_computer = yes*	*buys_computer = no*	*Total*	*Recognition (%)*
buys_computer = yes	6,954	46	7,000	99.34
buys_computer = no	412	2,588	3,000	86.27
Total	7,366	2,634	10,000	95.52

Classifier Accuracy Measures

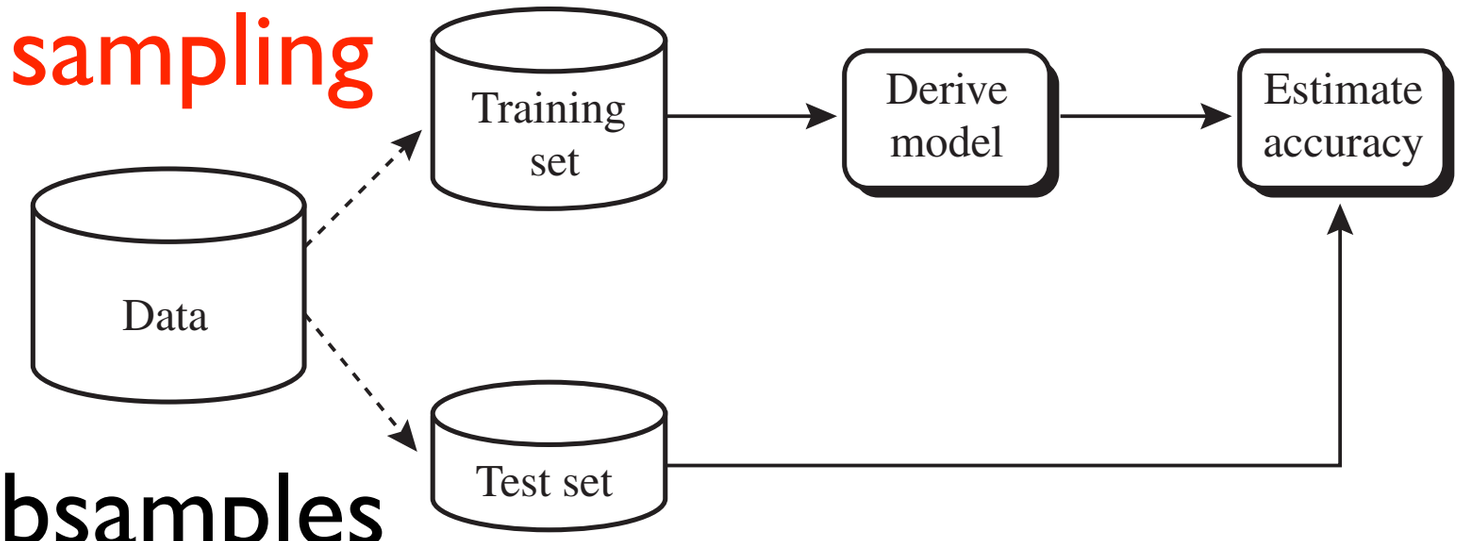
- Sensitivity: t_pos / pos
- Specificity: t_neg / neg
- Precision: t_pos / (t_pos + f_pos)

$$accuracy = sensitivity \frac{pos}{pos + neg} + specificity \frac{neg}{pos + neg}$$

- Costs and benefits of TP, TN, FP, FN

Classifier/Predictor Evaluation

- Holdout, random sampling



- Cross-validation
  - divide into k subsamples
    - use k-1 subsamples for training, one for testing --- k-fold cross validation
- Bootstrapping (e.g., .632 bootstrapping)
  - sample with replacement => training data

Predictor Error Measures

$$Absolute\ error : |y_i - y_i'|$$

$$Square\ error : (y_i - y_i')^2$$

$$Mean\ absolute\ error : \frac{\sum_{i=1}^{d} |y_i - y_i'|}{d}$$

$$Mean\ square\ error : \frac{\sum_{i=1}^{d} (y_i - y_i')^2}{d}$$

$$Relative\ absolute\ error : \frac{\sum_{i=1}^{d} |y_i - y_i'|}{\sum_{i=1}^{d} |y_i - \bar{y}|}$$

$$Relative\ square\ error : \frac{\sum_{i=1}^{d} (y_i - y_i')^2}{\sum_{i=1}^{d} (y_i - \bar{y})^2}$$

Model Selection

- Choose between two models M_1 and M_2
- Mean error rate? (k-fold cross validation)
  - estimated error on future data
- Difference between error rates of M_1 & M_2
  - statistically significant? or by chance?
- t-test

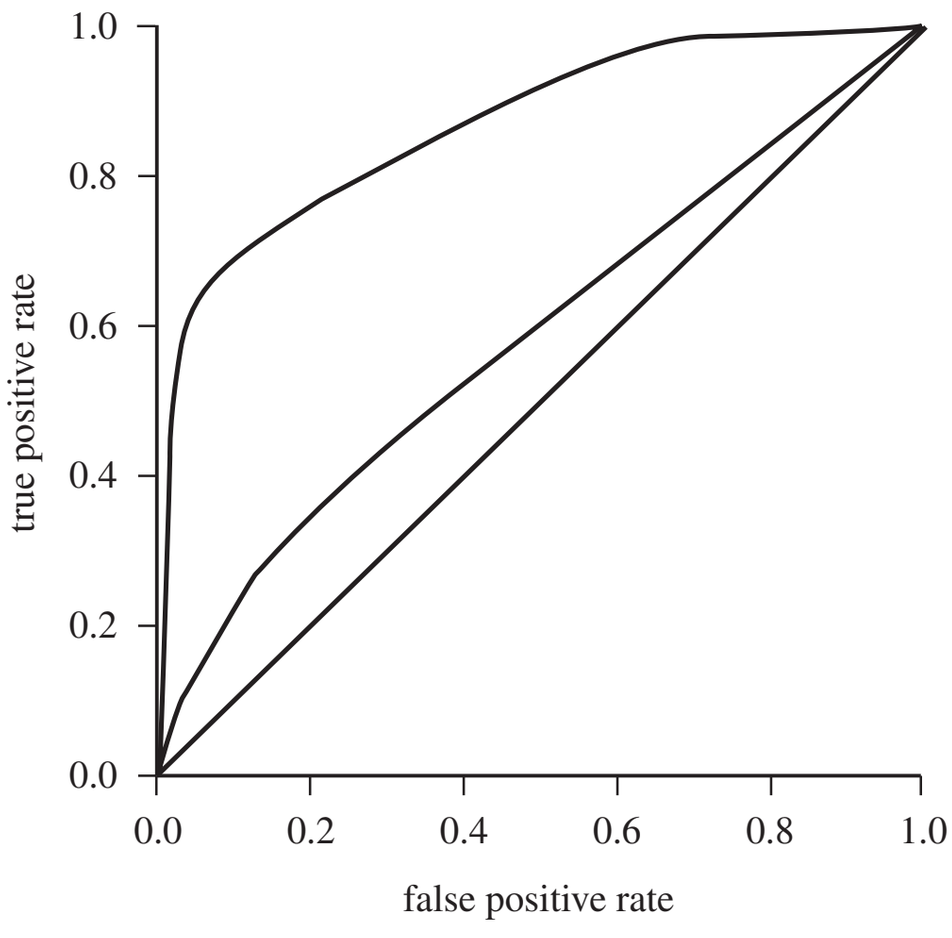
$$t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{var(M_1 - M_2)/k}}$$

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

Model Selection: ROC Curves

- Visual comparison of models
- X: false positive rate
  - f_pos / neg
- Y: true positive rate
  - t_pos / pos
- Area below curve:
  - accuracy
  - diagonal line: 0.5 accuracy

		Predicted class: C_1	Predicted class: C_2
Actual class	C_1	true positives	false negatives
	C_2	false positives	true negatives

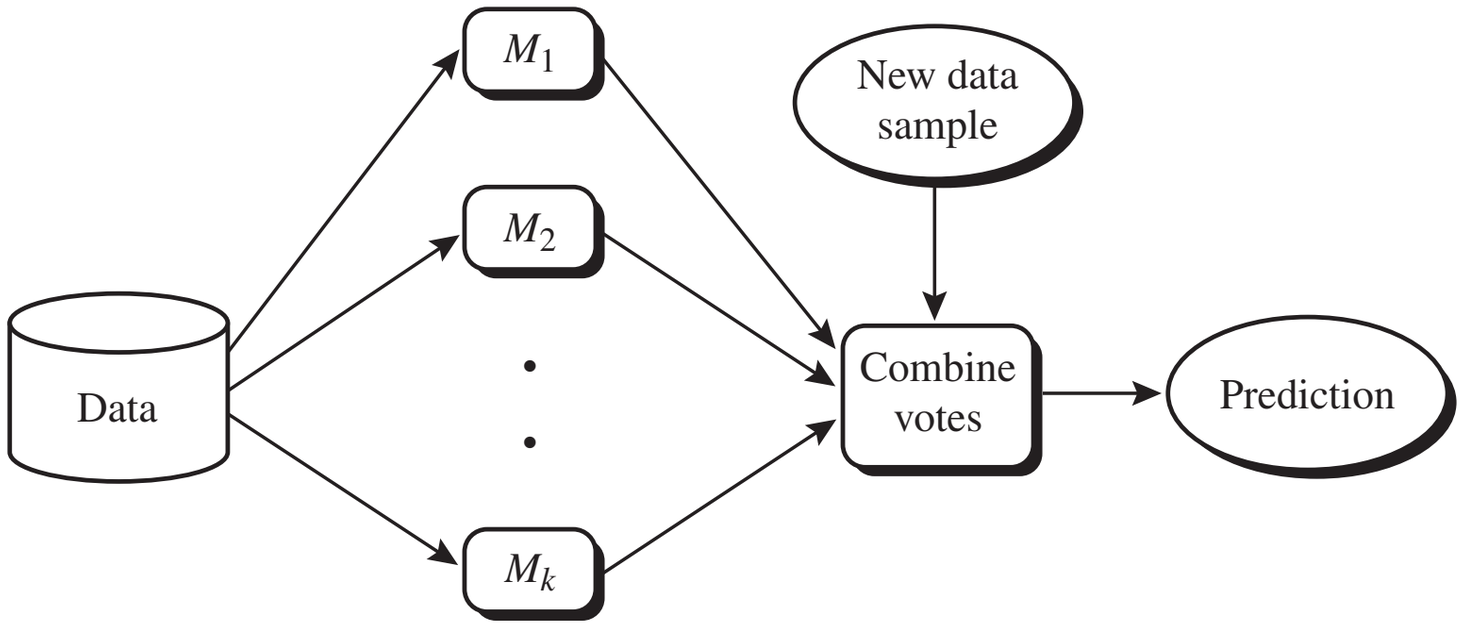


Chapter 8: Classification

- Basic concepts
- Decision tree induction
- Bayesian classification
- Rule-based classification
- Model evaluation and selection
- Improve classification accuracy
- Summary

Ensemble Methods

- Use a combination of multiple models to increase accuracy
- Popular ensemble methods
  - bagging: equal-weight votes
  - boosting: weighted votes



Bagging: Bootstrap Aggregation

- Analogy: diagnosis by multiple doctors
  - multiple doctors' majority vote
- Training: given a data set D of d tuples
  - training set D_i: d tuples sampled randomly with replacement (i.e., bootstrap)
  - a classifier M_i is learned for D_i
- Classification: majority vote
- Prediction: average of multiple predictions

Boosting

- Analogy: diagnosis by multiple doctors
  - weighted by previous diagnosis accuracy
- Weights are assigned to each training tuple
- A series of k classifiers is iteratively learned
- After classifier M_i is learned, adjust weights so M_{i+1} pays more attention to tuples that were misclassified by M_i
- M* combines the votes of all k classifiers, weighted by individual accuracy

Adaboost (1)

- D: (X_1, y_1), (X_2, y_2), ..., (X_d, y_d)
- Initial weight of each tuple: 1/d
- Round i (i=1, ..., k)
  - D_i: sample d tuples w/ replacement from D
    - Pr(choose $tuple_j$) based on $tuple_j$'s weight
    - Learn M_i from D_i, compute its error rate

$$error(M_i) = \sum_{j=1}^{d} w_j \times err(X_j)$$

Adaboost (2)

- Round i (continued)
  - Reduce weights of correctly classified tuples $$w_j = w_j \times \frac{error(M_i)}{1 - error(M_i)}$$
    - Normalize tuple weights so sum is 1.0
      - e.g., (0.1, 0.3, 0.1) => (0.2, 0.6, 0.2)
- Classification: weighted votes of k classifiers

$$weight(M_i) = log \frac{1 - error(M_i)}{error(M_i)}$$

Ensemble Methods: Accuracy

- Bagging
  - often significantly better than single classifier
  - noise: not considerably worse, more robust
  - prediction: proved improved accuracy
- Boosting
  - generally better than bagging
  - may overfit the model to misclassified data

Chapter 8: Classification

- Basic concepts
- Decision tree induction
- Bayesian classification
- Rule-based classification
- Model evaluation and selection
- Improve classification accuracy
- Summary