



University of Colorado
Boulder

CSCI 4502/5502

Data Mining

Fall 2020

Lecture 13 (Oct 13)

Reminder/Announcement

- ◆ In-person learning survey
 - ◆ due at 9:30am, Tuesday, Oct 13
- ◆ Office hour today
 - ◆ changed from 11am-12pm to 3pm-4pm
- ◆ Homework 4
 - ◆ posted in Canvas, due at 9:30am, Thursday, Oct 15



Review: Chapter 8: Classification

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



Naïve Bayesian Classifier (I)

- ◆ $X = (x_1, x_2, \dots, x_n)$ (i.e., n attributes)
- ◆ m classes: C_1, C_2, \dots, C_m
- ◆ Classification: maximal $P(C_i|X)$
- ◆ Based on Bayes' Theorem
$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$
- ◆ 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

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) \quad g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Naïve Bayesian Classifier Example

- ◆ 2 classes: `buys_computer`

- ◆ C_1 : yes

- ◆ C_2 : no

- ◆ $X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$

| 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 |



Naïve Bayesian Classifier Example

- ◆ buys_computer: $C_1 = \text{yes}$, $C_2 = \text{no}$
- ◆ $X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$
- ◆ $P(C_i|X) = P(X|C_i) P(C_i) / P(X)$
 - $P(X|C_i) = P(\text{age} \leq 30 | C_i)$
 - * $P(\text{income} = \text{medium} | C_i)$
 - * $P(\text{student} = \text{yes} | C_i)$
 - * $P(\text{credit_rating} = \text{fair} | C_i)$

| CID | age | income | student | credit rating | buys comput |
|-----|-----------|--------|---------|---------------|-------------|
| 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 |



Naïve Bayesian Classifier Example

- ◆ $P(\text{buys_computer} = \text{"yes"}) = 9/14 = 0.643$
 $P(\text{buys_computer} = \text{"no"}) = 5/14 = 0.357$
- ◆ $P(\text{age} = \text{"<=30"} | \text{buys_computer} = \text{"yes"}) = 2/9 = 0.222$
 $P(\text{age} = \text{"<= 30"} | \text{buys_computer} = \text{"no"}) = 3/5 = 0.6$
 $P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"yes"}) = 4/9 = 0.444$
 $P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$
 $P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$
 $P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"no"}) = 1/5 = 0.2$
 $P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$
 $P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$
- ◆ $P(X|\text{buys_computer} = \text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$
 $P(X|\text{buys_computer} = \text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$
- ◆ $P(X|\text{buys_computer} = \text{"yes"}) * P(\text{buys_computer} = \text{"yes"}) = 0.028$
 $P(X|\text{buys_computer} = \text{"no"}) * P(\text{buys_computer} = \text{"no"}) = 0.007$

| CID | age | income | stude | credit_rati | buys_compu |
|-----|-------|--------|-------|-------------|------------|
| 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 |



Avoid 0-Probability

- ◆ The 0-probability problem

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

- ◆ e.g., income: low(0), medium(990), high(10)
- ◆ Laplacian correction (or Laplace estimator)
 - ◆ add 1 to each case: non-zero, close to original probabilities
 - ◆ e.g., income: low(1), medium(991), high(11)



Naïve Bayesian Classifier

- ◆ **Advantage**

- ◆ easy to compute, good results in most cases

- ◆ **Disadvantage**

- ◆ assumption: class conditional independence

- ◆ dependencies exist in practice

- ◆ e.g., hospital patients: age, family history, fever, cough, lunch cancer, diabetes, etc.



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
- ◆ $\text{coverage}(R) = n_{\text{covers}} / |D|$
- ◆ $\text{accuracy}(R) = n_{\text{correct}} / n_{\text{covers}}$



Rule Assessment Example

◆ R: IF age ≤ 30 AND student = yes THEN buys_computer = yes

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

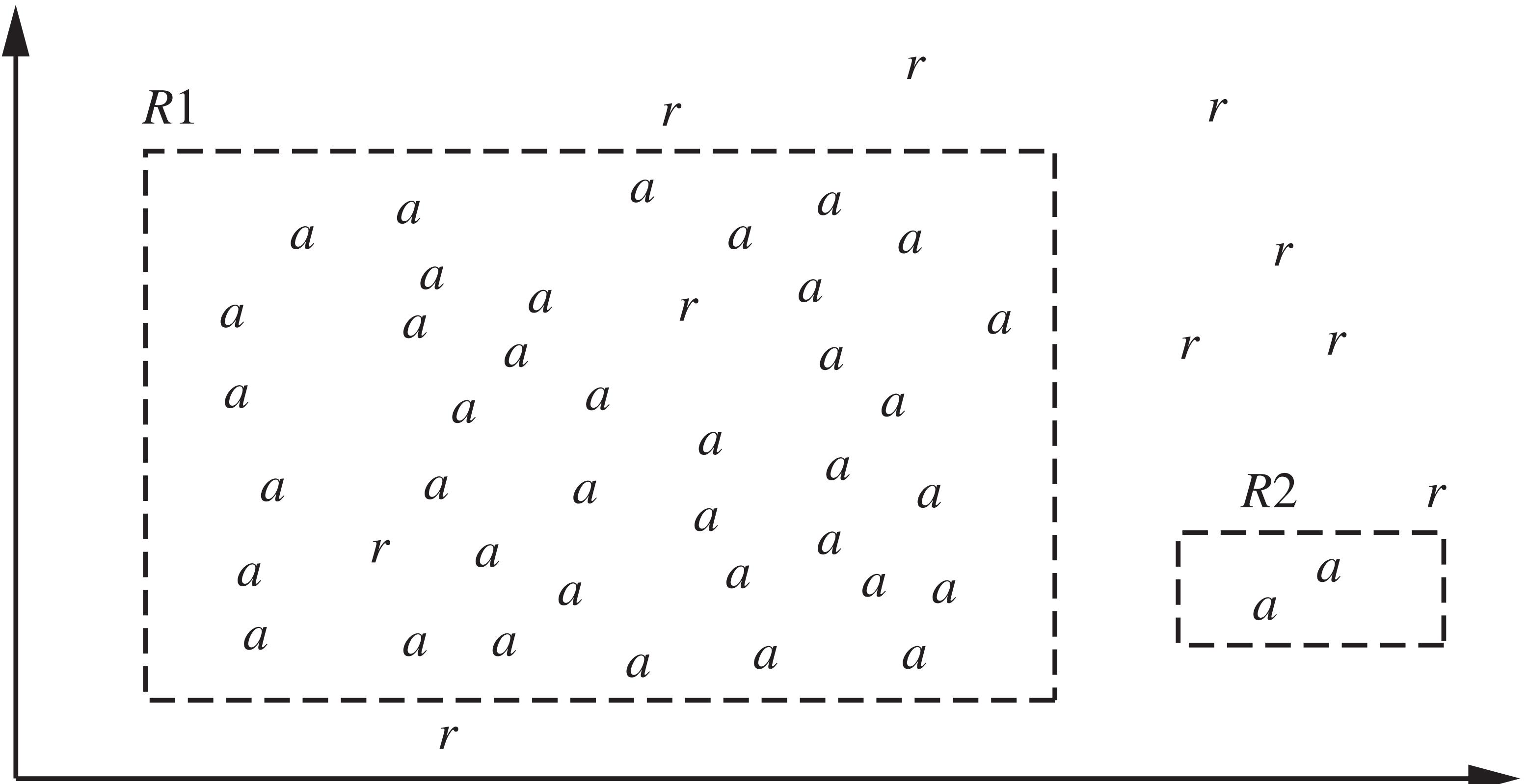
◆ accuracy (R) = $n_{correct} / n_{covers}$ = 2/2

| 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 Quality Measures

- ◆ Coverage?
- ◆ Accuracy?



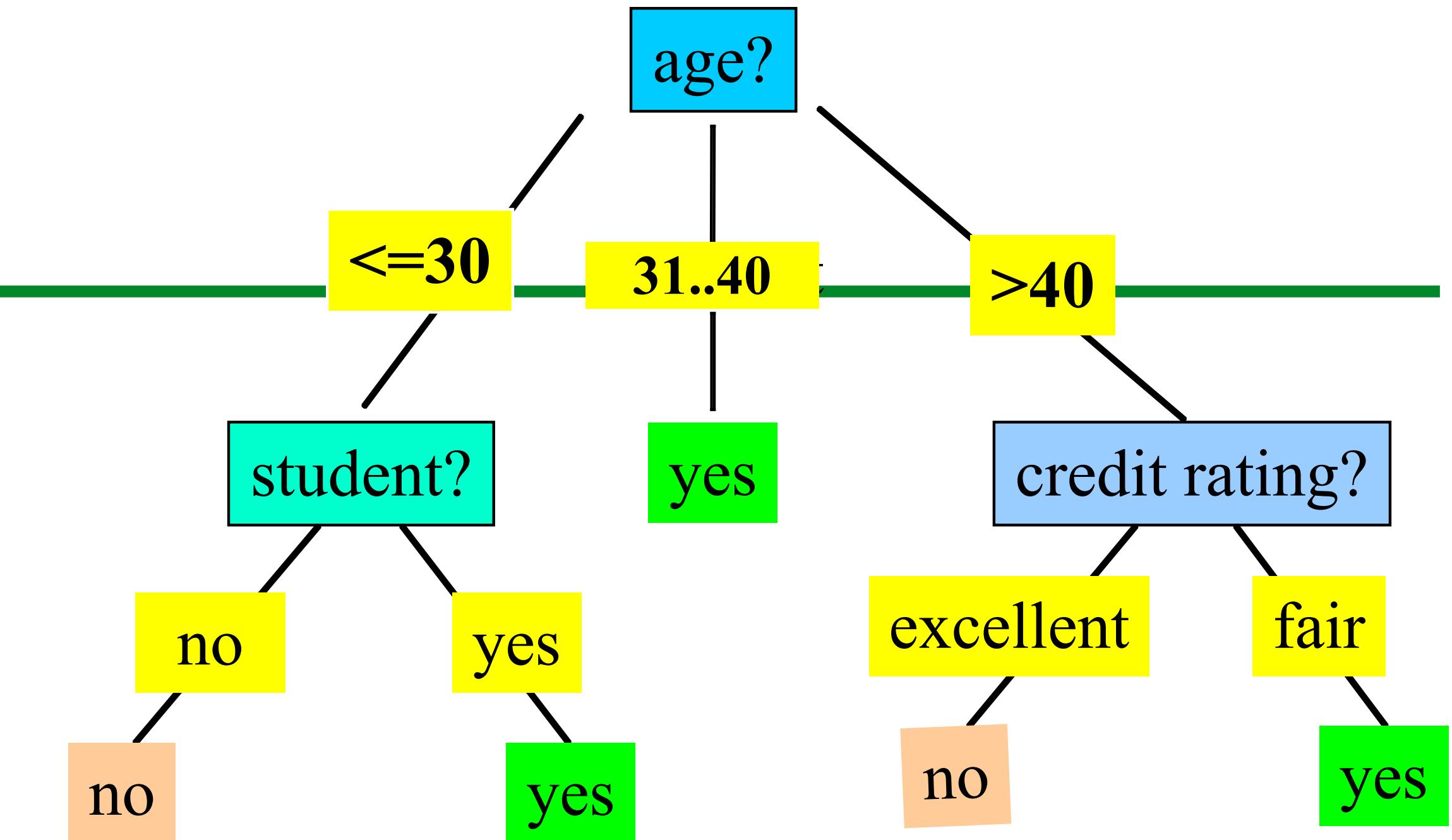
Rule-Based Classification

- ◆ 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

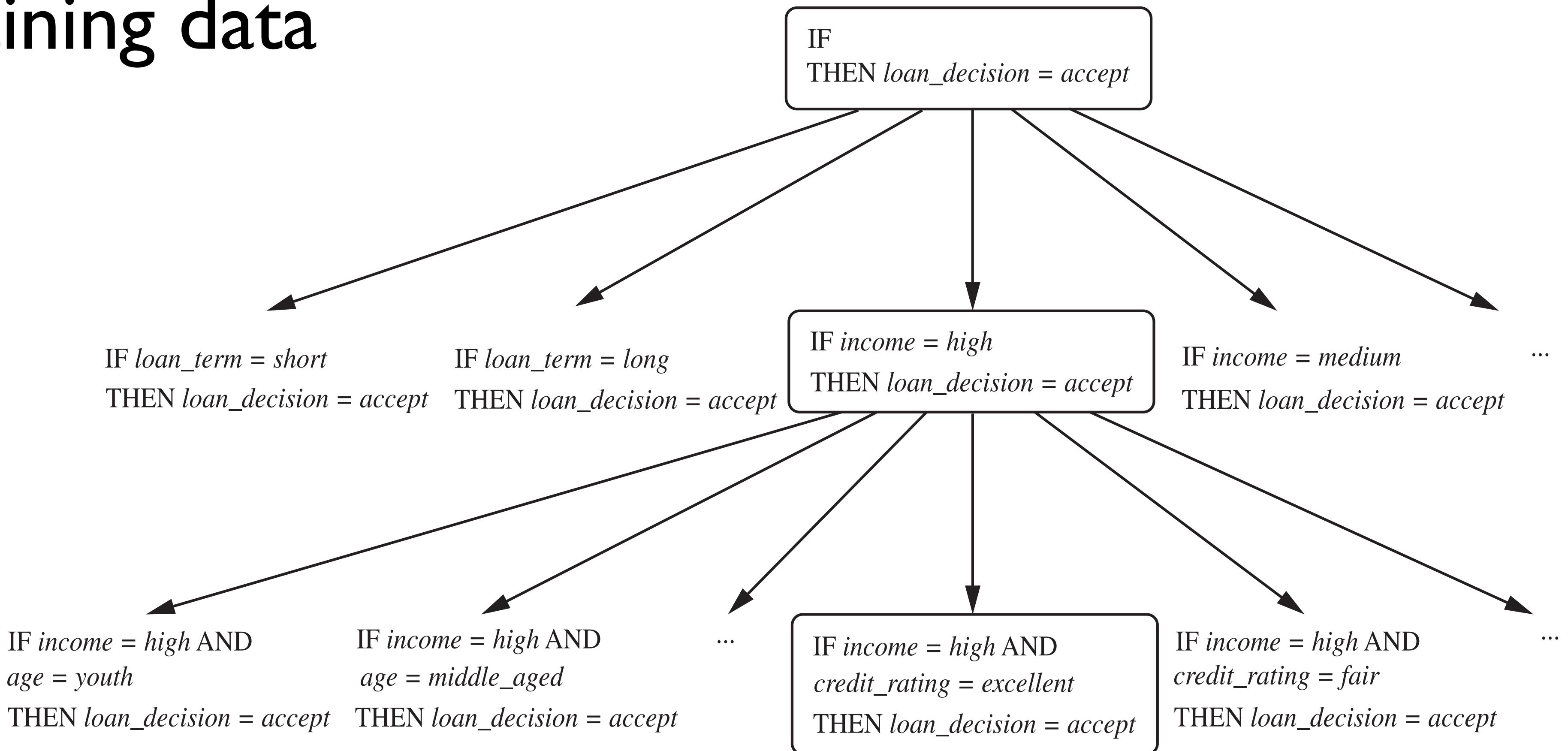


- ◆ Rules are exhaustive and mutually exclusive
- ◆ IF age=31...40 THEN buys_computer=yes
- ◆ IF age ≤ 30 AND student=no THEN buys_computer=no
- ◆ IF age > 40 AND credit_rating=fair THEN buys_computer = yes



Example

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



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

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

- ◆ Confusion matrix

| Classes | <i>buys_computer = yes</i> | <i>buys_computer = no</i> | Total | Recognition (%) |
|----------------------------|----------------------------|---------------------------|--------|-----------------|
| <i>buys_computer = yes</i> | 6,954 | 46 | 7,000 | 99.34 |
| <i>buys_computer = no</i> | 412 | 2,588 | 3,000 | 86.27 |
| Total | 7,366 | 2,634 | 10,000 | 95.52 |



Classifier Accuracy Measures

- ◆ Sensitivity: $t_{\text{pos}} / \text{pos}$
- ◆ Specificity: $t_{\text{neg}} / \text{neg}$
- ◆ Precision: $t_{\text{pos}} / (t_{\text{pos}} + f_{\text{pos}})$

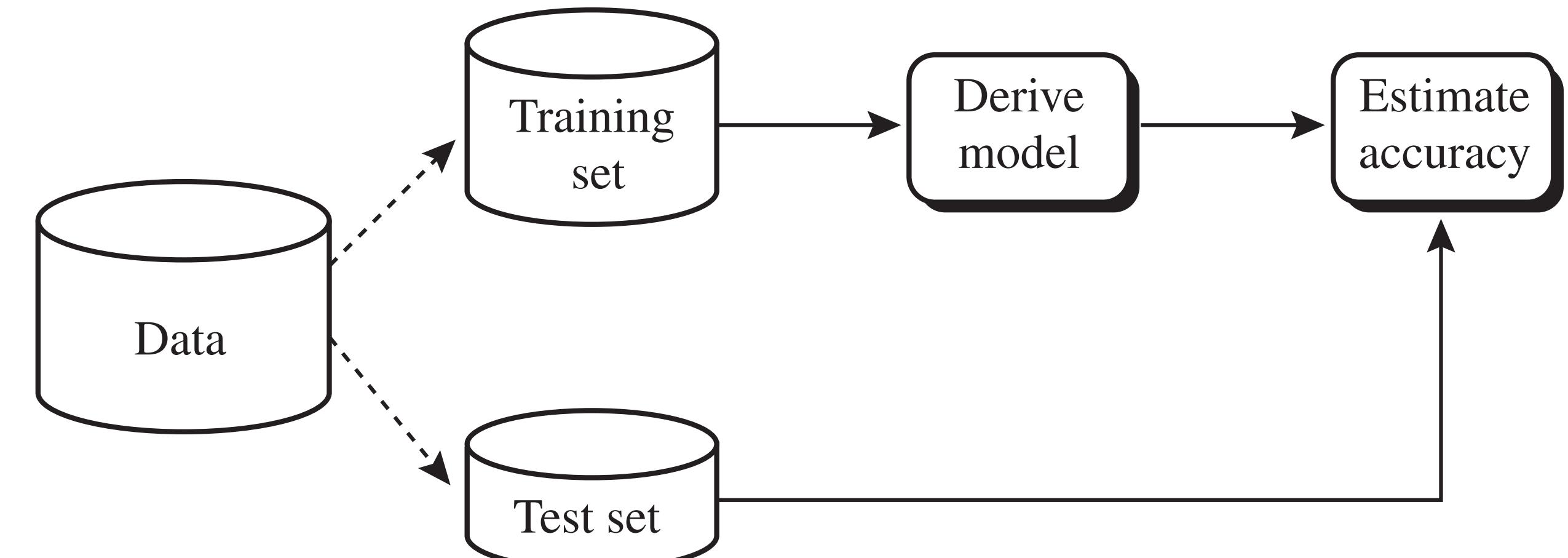
$$\text{accuracy} = \text{sensitivity} \frac{\text{pos}}{\text{pos} + \text{neg}} + \text{specificity} \frac{\text{neg}}{\text{pos} + \text{neg}}$$

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

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



Model Selection: ROC Curves

- ◆ Visual comparison of models

- ◆ X: false positive rate

- ◆ f_pos / neg

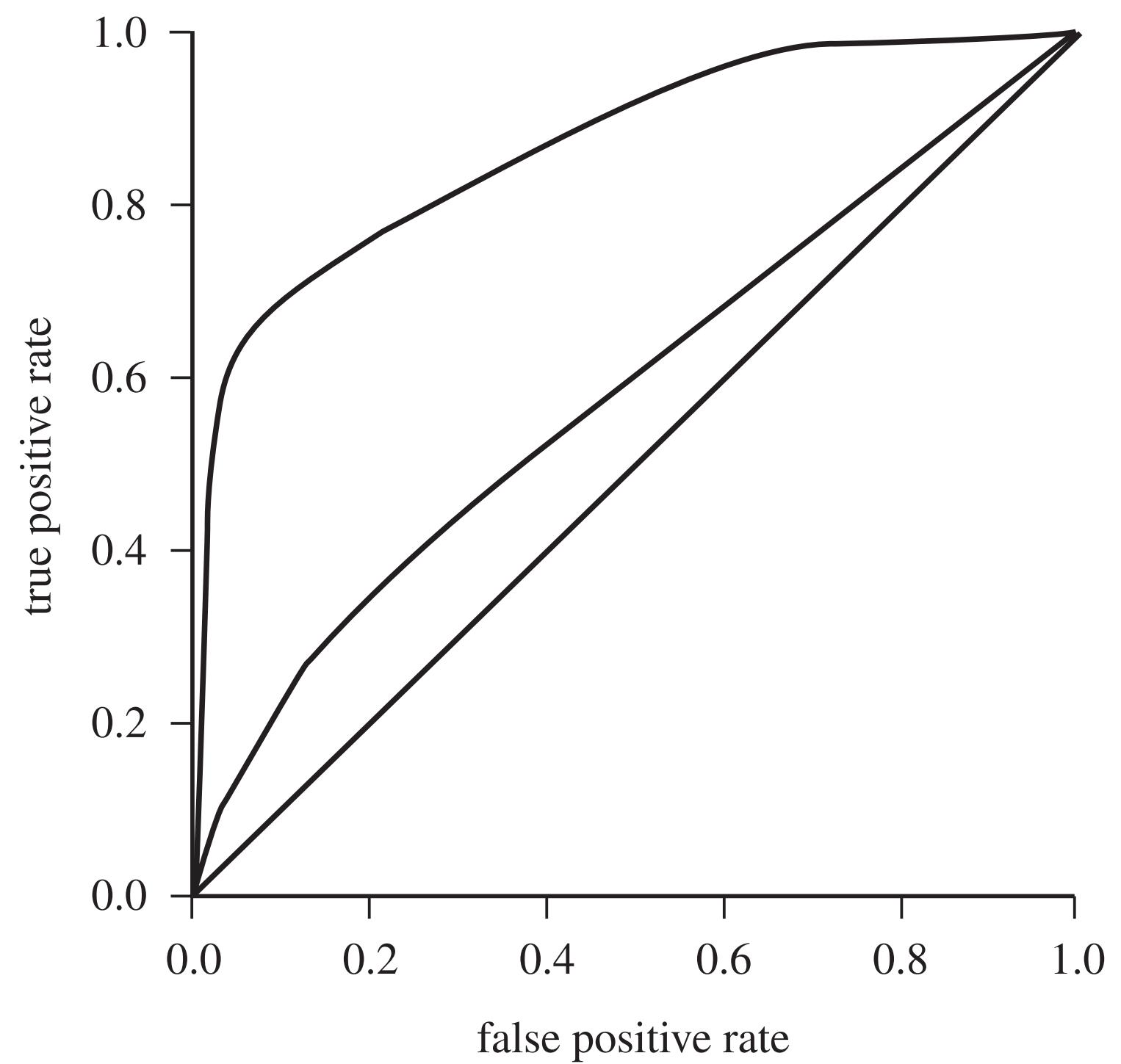
- ◆ Y: true positive rate

- ◆ t_pos / pos

- ◆ Area below curve:

- ◆ accuracy, diagonal line: 0.5 accuracy

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



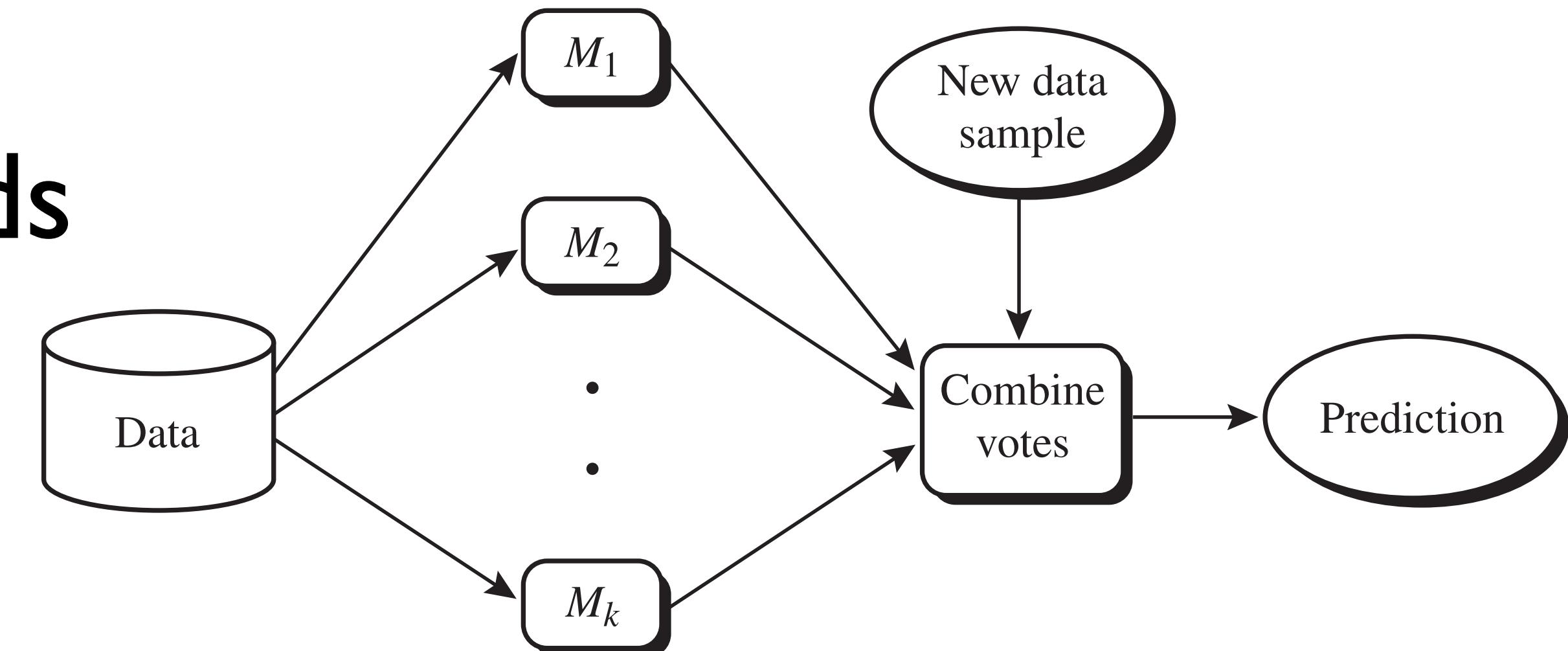
Chapter 8: Classification

- ◆ 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)
- ◆ 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 (I)

- ◆ $D: (X_1, y_1), (X_2, y_2), \dots, (X_d, y_d)$
- ◆ Initial weight of each tuple: $1/d$
- ◆ Round i ($i=1, \dots, k$)
$$error(M_i) = \sum_{j=1}^d w_j \times err(X_j)$$
- ◆ D_i : sample d tuples w/ replacement from D
- ◆ $\Pr(\text{choose tuple}_j)$ based on tuple $_j$'s weight
- ◆ Learn M_i from D_i , compute its error rate



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., $(0.1, 0.3, 0.1) \Rightarrow (0.2, 0.6, 0.2)$ $weight(M_i) = \log \frac{1 - error(M_i)}{error(M_i)}$
 - ◆ Classification: weighted votes of k classifiers



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

