# CS699
# Lecture 5
# Classification 2

# Using IF-THEN Rules for Classification

- Represent the knowledge in the form of IF-THEN rules

  R: IF *age* = youth AND *student* = yes THEN *buys_computer* = yes

  - IF antecedent/precondition/condition

    THEN consequent/conclusion
- Assessment of a rule: *coverage* and *accuracy*
  - A tuple is covered by R if it satisfies the antecedent of R
  - $n_{covers}$ = # of tuples covered by R
  - $n_{correct}$ = # of tuples correctly classified by R
  - coverage(R) = $n_{covers}$ /|D| /* D: training data set */
  - accuracy(R) = $n_{correct}$ / $n_{covers}$
  - Refer to Example 8.6, page 356

# Using IF-THEN Rules for Classification

- Another example
- R2: IF *income* = high AND *student* = no

  THEN *buys_computer* = no
- R2 covers three tuples

  coverage(R2) = 3/14 = 21.43%

  Among these three tuples, two tuples are correctly classified by R2.

  accuracy(R2) = 2/3 = 66.67%

| age | income | student | credit_rating | buys_computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

# Using IF-THEN Rules for Classification

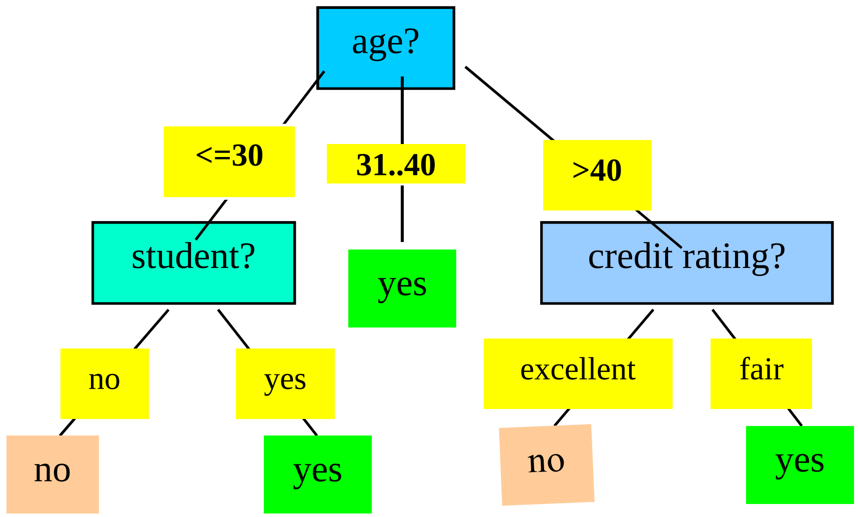
- If more than one rule are triggered, need **conflict resolution**
  - Size ordering: assign the highest priority to the triggering rules that has the "toughest" requirement (i.e., with the *most attribute tests*)
  - Class-based ordering: decreasing order of *prevalence or misclassification cost per class*
  - Rule-based ordering (**decision list**): rules are organized into one long priority list, according to some measure of rule quality or by experts
  - When a tuple is not covered by any rule, a default rule is used. E.g., a majority class is assigned.

# Rule Extraction from a Decision Tree

- One rule is created *for each path* from the root to a leaf
- Each attribute-value pair along a path forms a conjunction (AND): the leaf holds the class prediction

# Rule Extraction from a Decision Tree

- Example: Rule extraction from our *buys_computer* decision-tree




| | |
|---|---|
| IF *age* <= 30 AND *student* = *no* | THEN *buys_computer* = *no* |
| IF *age* <= 30 AND *student* = *yes* | THEN *buys_computer* = *yes* |
| IF *age* in 31..40 | THEN *buys_computer* = *yes* |
| IF *age* > 40 AND *credit_rating* = *excellent* | THEN *buys_computer* = *no* |
| IF *age* > 40 AND *credit_rating* = *fair* | THEN *buys_computer* = *yes* |

# Rule Extraction from the Training Data

- Sequential covering algorithm: Extracts rules directly from training data
- Rules are learned *sequentially*
- Each rule for a given class will ideally cover many tuples of the class but none (or few) of the tuples of other classes.
- There are many sequential covering algorithms.
- General strategy:
  - Rules are learned one at a time
  - Each time a rule is learned, the tuples covered by the rules are removed
  - The process repeats on the remaining tuples unless *termination condition*, e.g., when no more training examples or when the quality of a rule returned is below a user-specified threshold
- Comp. w. decision-tree induction: learning a set of rules *simultaneously*

# Simple Covering Algorithm (Example)

- Example
- First, *yes* rules.
- IF ? THEN *buys_computer* = *yes*

| | |
|---|---|
| *age* <= 30 | 2/5 |
| *age* 31..40 | 4/4 |
| *age* > 40 | 3/5 |
| *income* = low | 3/4 |
| *income* = medium | 4/5 |
| *income* = high | 2/4 |
| *student* = no | 3/7 |
| *student* = yes | 6/7 |
| *CR* = fair | 6/8 |
| *CR* = excellent | 3/6 |

| age | income | student | credit_rating | s_comp |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

# Simple Covering Algorithm (Example)

- A new rule is created
- IF *age* = 31..40
  THEN *buys_computer* = *yes*
- Four tuples covered by this rule are removed and the same is repeated (for *yes* tuples)
- Once all *yes* tuples are covered, the same is done for *no* tuples

| age | income | student | credit_rating | s_comp |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| >40 | medium | no | excellent | no |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Simple Covering Algorithm (Example)

- IF *age* = 31..40

  THEN *buys_computer* = *yes*

| | |
|---|---|
| *age* <= 30 | 2/5 |
| *age* > 40 | 3/5 |
| *income* = low | 2/3 |
| *income* = medium | 3/5 |
| *income* = high | 0/2 |
| *student* = no | 1/5 |
| *student* = yes | 4/5 |
| *CR* = fair | 4/6 |
| *CR* = excellent | ¼ |

- IF *age* = 31..40 AND

  *student = yes*

  THEN *buys_computer* = *yes*

| age | income | student | credit_rating | s_comp |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| >40 | medium | no | excellent | no |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Numeric Prediction

- (Numerical) prediction is similar to classification
  - construct a model
  - use model to predict continuous values
- Prediction is different from classification
  - Classification refers to prediction of categorical class label
  - Prediction models continuous-valued functions
- Major method for prediction: regression
  - model the relationship between one or more *independent* or **predictor** variables and a *dependent* or **response** variable
- Regression analysis
  - Linear and multiple regression
  - Non-linear regression
  - Other regression methods: generalized linear model, Poisson regression, log-linear models, regression trees

# Linear Regression

- Linear regression: involves a response variable y and a single predictor variable x

  $$y = w_0 + w_1 x$$

  where $w_0$ (y-intercept) and $w_1$ (slope) are regression coefficients

  Method of least squares: estimates the best-fitting straight line

- Multiple linear regression: involves more than one predictor variable
  - Training data is of the form $(\mathbf{X_1}, y_1), (\mathbf{X_2}, y_2), \ldots, (\mathbf{X_{|D|}}, y_{|D|})$
  - Ex. For 2-D data, we may have: $y = w_0 + w_1 x_1 + w_2 x_2$
  - Many nonlinear functions can be transformed into the above

# Other Regression

- Nonlinear regression: some nonlinear models can be modeled by a math function

Example: a polynomial regression model

$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

- Other Regression-Based Models
  - Logistic regression
  - Regression trees

# Classification Example

- Adult
- Car evaluation
- Vote
- Credit screening
- CPU (numeric prediction)
- All from UCI Machine Learning Data Repository

# Adult Dataset

- This dataset was extracted from census bureau database.
- 32561 tuples, 15 attributes
- Class attribute: whether earns > 50K or <= 50K

- Attributes: age, workclass, fnlwgt, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native-country, class

# Car Evaluation Dataset

- The model evaluates cars according to the following concept structure:
- CAR — car acceptability
  - PRICE — overall price
    - buying — buying price
    - maint — price of the maintenance
  - TECH — technical characteristics
    - COMFORT — comfort
      - doors — number of doors
      - persons — capacity in terms of persons to carry
      - lug_boot — the size of luggage boot
    - safety — estimated safety of the car

# Car Evaluation Dataset

- 1728 tuples, 7 attributes
- Class attribute:
  - car acceptability
  - values: unacc, acc, good, vgood
- Attributes: buying, maint, doors, persons, lug_boot, safety

# Vote Dataset

- This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes
- 435 tuples, 7 attributes
- Class attribute: Democrat or Republican
- Attributes: handicapped-infants, water-project-cost-sharing, adoption-of-the-budget-resolution, physician-fee-freeze, el-salvador-aid, religious-groups-in-schools, anti-satellite-test-ban, aid-to-nicaraguan-contras, mx-missile, immigration, synfuels-corporation-cutback, education-spending, superfund-right-to-sue, crime, duty-free-exports, export-administration-act-south-africa

# Credit Screening Dataset

- This file concerns credit card applications. All attribute names and values have been changed to protect confidentiality of the data.
- 690 tuples, 16 attributes
- Class attribute: approved or denied

# CPU Dataset

- CPU performance
- 209 tuples, 7 attributes
- Class attribute: relative performance (numeric)
- Attributes:
  - MYCT: cycle times
  - MMIN: minimum main memory (KB)
  - MMAX: maximum main memory (KB)
  - CACH: cache (KB)
  - CHMIN: minimum channels
  - CHMAX: maximum channels

# References


- Han, J., Kamber, M., Pei, J., “Data mining: concepts and techniques,” 3rd Ed., Morgan Kaufmann, 2012
- http://www.cs.illinois.edu/~hanj/bk3/
- Ian H. Witten, E. Frank, and M.A. Hall, "Data Mining Practical Machine Learning Tools and Techniques," Third Ed., 2011, Morgan Kaufmann