Feature Selection from Means and Variances

- Principle
  - Compute the means of a feature for each class, normalized by the variances;
  - If the means are far apart, interest in a feature increases (the feature has potential in terms of distinguishing between classes);
  - If the means are indistinguishable, interest wanes in that feature.
- Two intuitive methods
  - Independent feature analysis (√)
    - Assuming the features are independent. Features are examined individually.
  - Distance-based feature selection
    - Features are examined collectively.
- Limitation: only applied to continuous features.

Independent Feature Analysis

- For a problem with two classes: C_1 and C_2 :
  - Compute $mean_1(f)$ and $mean_2(f)$: the means of feature f measured for C_1 and C_2
  - Compute $var_1(f)$ and $var_2(f)$: the variances of feature f measured for C_1 and C_2
  - Significance test (t-test):

$$|mean_1(f) - mean_2(f)| > sig \times \sqrt{\frac{\mathrm{var}_1(f)}{n_1} + \frac{\mathrm{var}_2(f)}{n_2}}$$

    - n_1 and n_2 are the numbers of cases in C_1 and C_2
    - sig = 2 for the 95% confidence level.
  - If the comparison fails the test, the feature can be deleted.
- For k classes, k pairwise comparisons are conducted for f.
  - Each pairwise comparison compares feature means for class C_i and $\neg C_i$ ($i=1, \ldots, k$).
  - A feature is retained if it is significant for at least one of the pairwise comparisons.
- Limitation: Treat each feature independently

Feature Selection by Mutual Information

- Objective: Select features according to the mutual information between a feature and the class variable.
- The mutual information (also called information gain) between the class variable y and a discrete feature x :

$$MI_x = \sum_v \sum_c [P(y=c, x=v) \times \log_2 \frac{P(y=c, x=v)}{P(y=c)P(x=v)}]$$

  - $P(y=c)$ is the probability of cases in class c.
  - $P(x=v)$ is the probability that feature x takes on value v.
- *MI* measures the degree to which x and y are not independent. The bigger the value, the more dependent y is on x.
- *MI* is used to select or weight features.
  - You can select the top k features with the highest weights. Or some mining algorithms can take the feature weights and select features in the mining process.
- Suitable for nominal or discrete attributes. For continuous features, a discretization algorithm can be applied first to convert a real-valued feature to a discrete-valued feature.
- Limitation: Treat each feature independently.

32

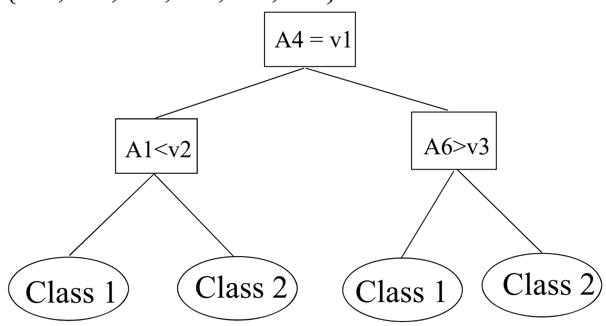
Feature Selection by Decision Trees

- Objective
  - Decision tree learning methods integrates feature selection to their algorithms and decision tree is a fast learning method.
  - Make use of the decision tree learning technique to select features from a data set for other learning methods, such as neural networks, that take substantially more time to search their solution space.
    - Decision tree learning is a relatively fast learning method.
- Method
  - Apply a decision tree learning algorithm to the data set to generate a decision tree.
  - Select features that appear in the tree.
- Advantage
  - *Context sensitive*. Tree methods evaluate candidate features in the context of related features that have already been selected.

33

Example of Feature Selection Using Decision Tree

Initial attribute set:
{A1, A2, A3, A4, A5, A6}



------> Reduced attribute set: {A1, A4, A6}

34

Data Reduction Outline

- Feature Selection
- <u>Case Reduction</u>
- Value Reduction

35

Case Reduction

- Objective: reduce the number of cases, the largest dimension in the data set
- How many cases are enough?
  - Application dependent - depends on the complexity of the patterns to be extracted from the data.
    - If the pattern is simple, the results are unlikely to change even with additional cases. For example, $x>1$ completely separates two classes.
    - For complex patterns, large volumes of data can supply more evidence for the correctness of the induced patterns.
- Some types of problems requiring more data than others:
  - Multiclass classification
  - Regression
  - Imbalanced data sets: almost all cases belong to the larger class, and far fewer cases to the smaller, usually more interesting class.

Case Reduction Methods

- Simple Random Sampling
  - A single sample
  - Incremental samples
  - Average samples
- Sampling by Adjusting Prevalence
- Stratified Sampling

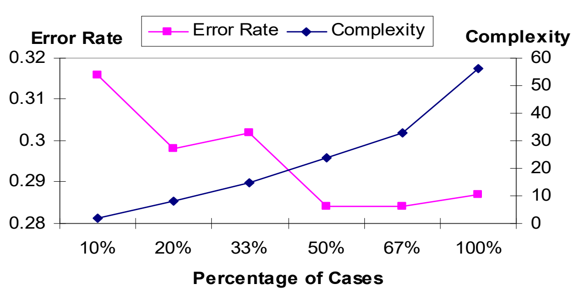
Single Simple Random Sample

- Choose n objects randomly from a set D of N objects ($n<N$) so that each object has the same probability of being chosen.
- Two methods
  - Simple random sampling without replacement (SRSWOR)
    - Each object cannot be chosen more than once
  - Simple random sampling with replacement (SRSWR)
    - Each time an object is drawn, it is recorded and placed back to D so that it can be drawn again.

38

Problem with Single Sampling

- We don't know the suitable sample size.
- Too small, the model may not be accurate enough; too big, the model may be more complex

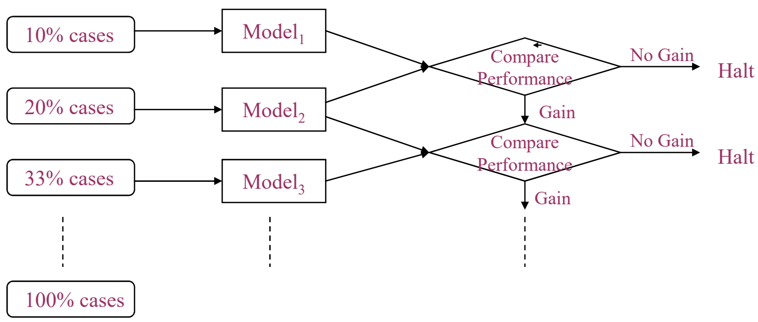


39

Incremental Sampling

- Objective: Spot trends in error and complexity by *learning with incrementally larger random subsets of the data* to help produce a single solution.
- A typical pattern of incremental subsets is:

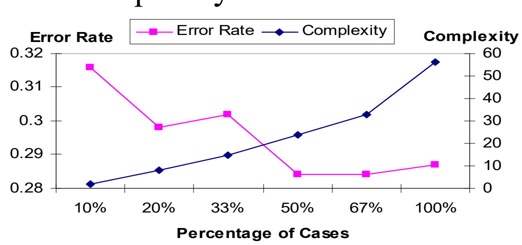
  10%, 20%, 33%, 50%, 67%, 100%

- 

Incremental Sampling (Cont.)

- Performance measures:
  - Error rate (test error, that is error on a test data set)
  - Complexity of the solution (e.g. number of nodes in a tree)
- Plot error and complexity relative to increasing sample size.

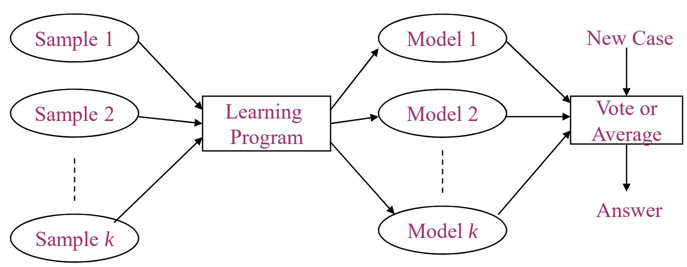
Example:



- Make decision on whether to do further sampling
  - Net changes in error and complexity are examined:
    - Is the error smaller?
    - Is the complexity acceptable?
    - Is complexity increasing much more than error is decreasing?

Average Sampling

- For a dataset containing a huge number of cases that exceed the maximum capacity of a learning program.
- Average sampling:
  - Select k random samples of n cases
  - Solutions from different samples are combined in the prediction phase.



- Averaged or voted solutions usually have less error than the single solution found on all cases in the database.

42

Sampling by Adjusting Prevalence

- Directly adjust the prevalence of cases over the classes.
- Suitable for classification problems with a very imbalance data set
  - In a bio-chemistry data set for predicting biological potency of chemical compounds,
    - Only 0.16% of the compounds belong to the class of highly active compounds, which is the most interesting class that can lead to discovery of new drugs.
    - Remaining 99.84% of compounds are inactive.
  - Low-prevalence class, usually the most interesting class.

43

Sampling by Adjusting Prevalence

- Two ways for boosting prevalence:
  - *Up-sampling*: repeat (or give higher weights to) the cases in the low-prevalence class in the sample - increase the sample size.
  - *Down-sampling*: keep the low-prevalence cases intact or randomly sample a high percentage of them, while including a low percentage random subset of a larger class in the training sample.
- Result: the predictive performance on the most interesting new cases may increase, while the overall predictive performance on new data of all classes may decrease.

44

Stratified Sampling

- The data set *D* is partitioned into mutually disjoint subsets, called *strata*.
- Then randomly sample data from each stratum
- Objective: ensure a representative sample, especially when the data are skewed.

45

Data Reduction Outline

- Feature Selection
- Case Reduction
- Value Reduction

46

Reducing and Smoothing Values

- Objective
  - Reduce the number of distinct values of a feature so that the size of the search space for patterns is reduced.
  - Smooth out noise
- Methods for reducing values
  - Nominal attributes
    - Generalization.

Toronto → Ontario → Central Canada → Canada

47

Reducing and Smoothing Values (Cont’d)

- Integer or real-valued attributes
  - Rounding
    - e.g. 462.4 can be rounded to 462, 460, or 500 according to requirements
  - Binning
    - Partition the value range of an attribute into bins

      $\underbrace{1, 1, 2,}_{\text{bin1}} \underbrace{3, 3, 3,}_{\text{bin2}} \underbrace{4, 5, 5, 7}_{\text{bin3}}$

    - Smooth values by bin medians, means or boundaries

      $\underbrace{1, 1, 1,}_{\text{bin1}} \underbrace{3, 3, 3,}_{\text{bin2}} \underbrace{5, 5, 5, 5}_{\text{bin3}}$ $\qquad$ $\underbrace{1, 1, 2,}_{\text{bin1}} \underbrace{3, 3, 3,}_{\text{bin2}} \underbrace{4, 4, 4, 7}_{\text{bin3}}$

  - Discretization: label each bin by discrete values