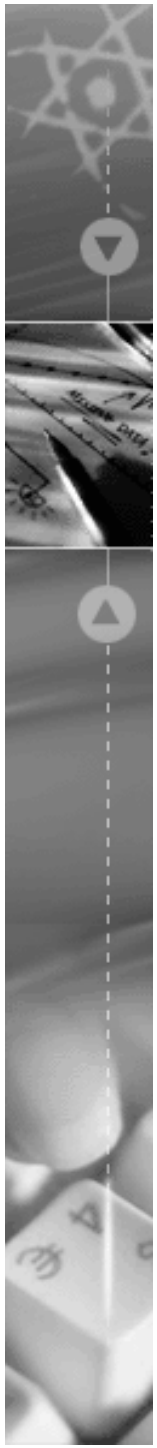
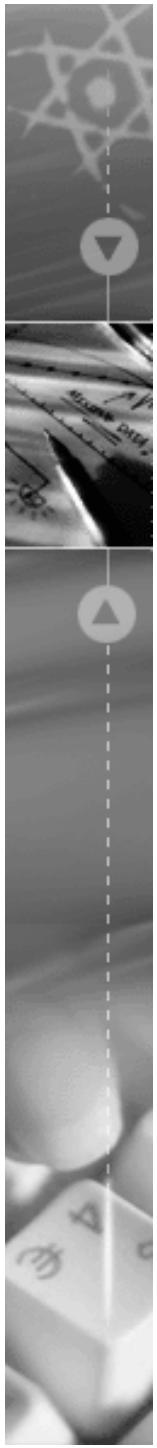


Text Categorization

Yen, Yung-Tai
Graduate Institute of Medical Informatics
Taipei Medical University

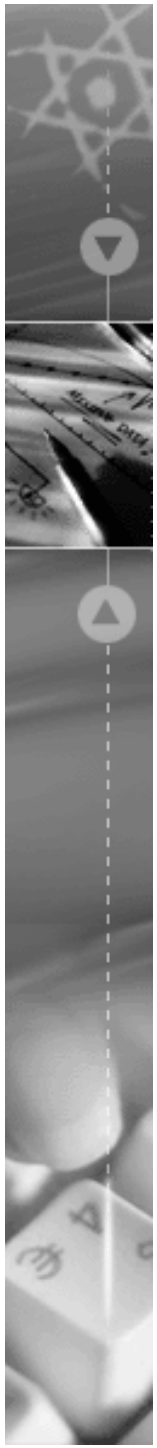
C. Manning and H. Schutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.



Tasks of Categorization

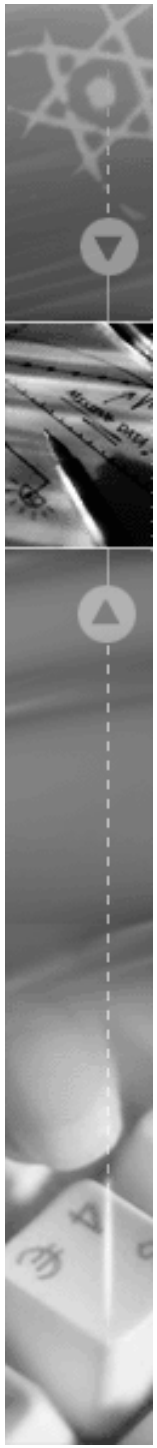
- Assign objects from a universe to two or more classes.
- Example:

Problem	Object	Categories
Tagging	Conext of a word	Word's tags
Author ID.	Document	Authors
Text Cat.	Document	Topics



Problems

- Data representation model.
  - Training set of objects with labels.
- Model class.
  - A parameterized family of classifiers
- Training procedure.
  - Select one classifier from the family.

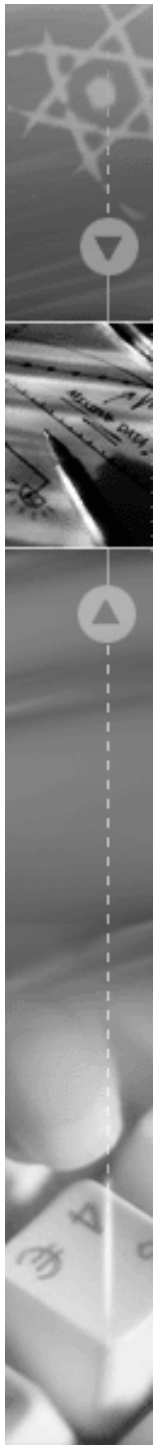


Data Representation Model

- General form:

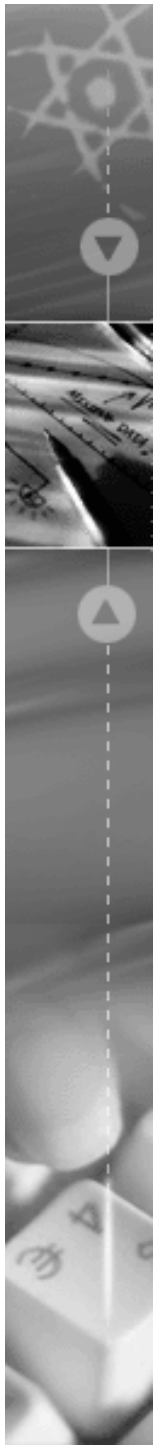
$$(\vec{x}, c)$$

- $\vec{x}$ is a vector measurements.
- c is the class label.
- Vector space model is frequently used.



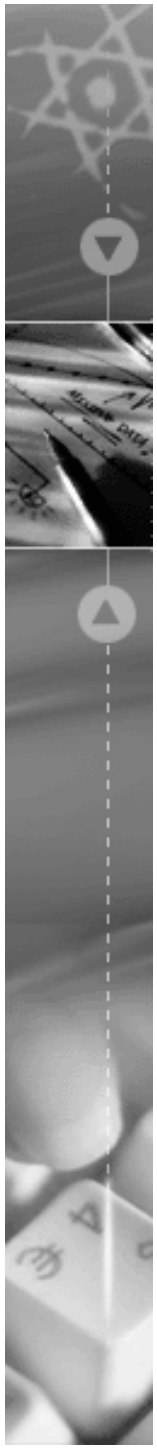
Model Class

- A parameterized family of classifiers.
- Example of linear classifiers for binary classification:
- $g(\vec{x}) = \vec{w} \cdot \vec{x} + w_0$
- $g(\vec{x}) > 0 \Rightarrow c_1, \quad g(\vec{x}) \leq 0 \Rightarrow c_2$



Training Procedure

- Function fitting algorithm.
- Search for a good set of parameter.
  - Optimization criteria:
    - Misclassification rate
    - Entropy



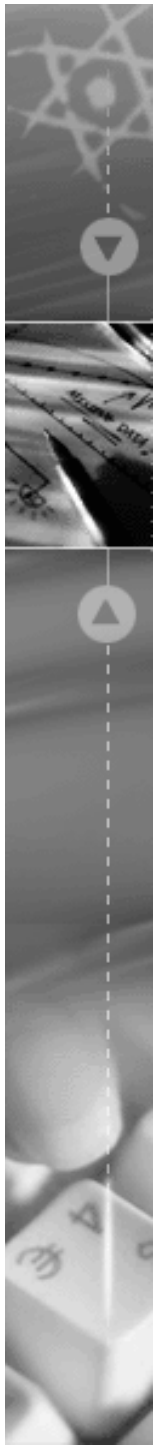
Evaluation

- Test set for measuring performance.
- Contingency table for evaluating a binary classifier.

Evaluation (*cont.*)

	YES is correct	NO is correct
YES was assigned	*a*	*b*
NO was assigned	*c*	*d*

- *Accuracy:* $\frac{a+d}{a+b+c+d}$
- *Precision:* $\frac{a}{a+b}$ *Recall:* $\frac{a}{a+c}$
- *Fallout:* $\frac{b}{b+d}$

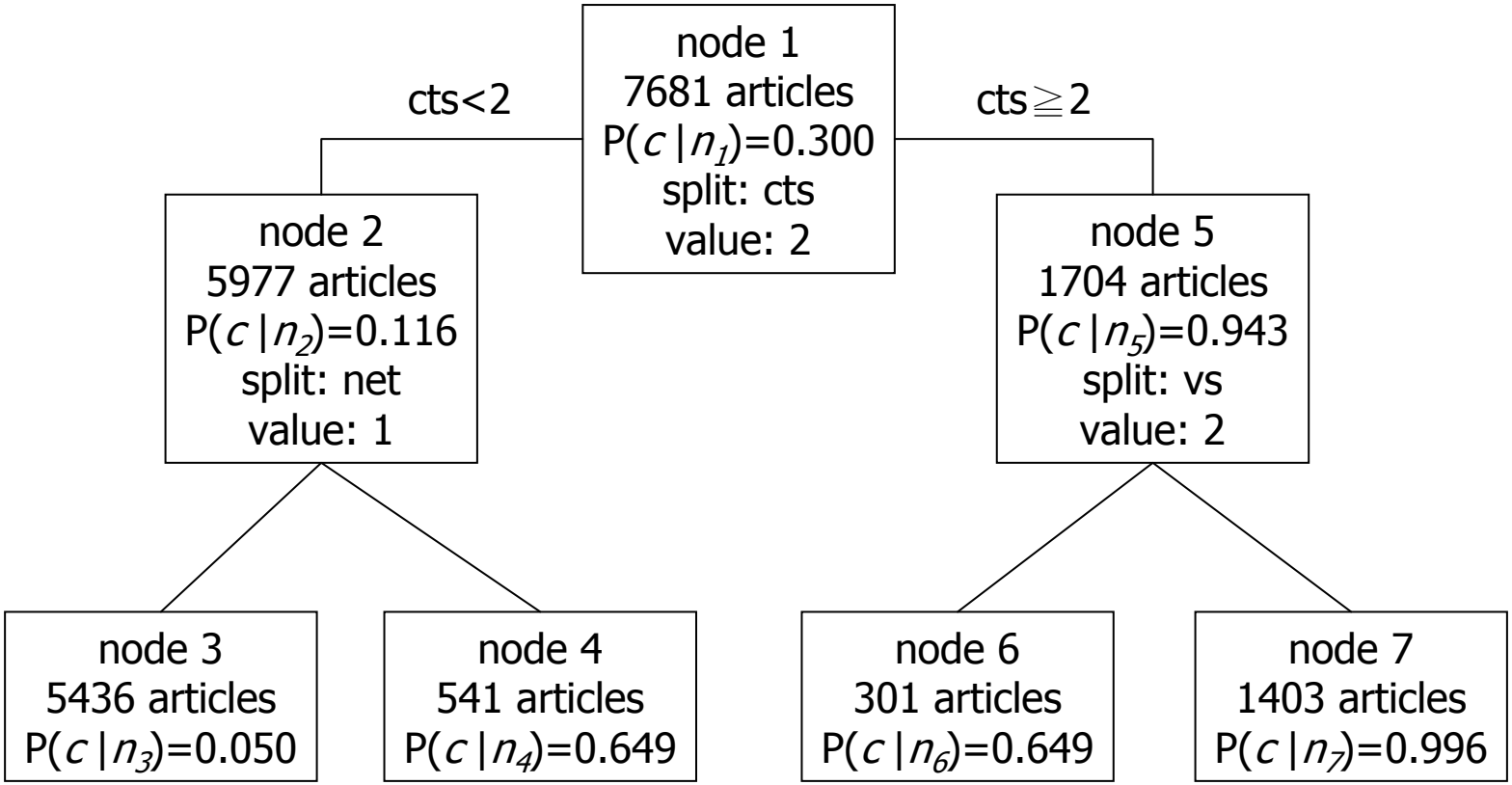


Evaluation for More Than Two Categories

- 2x2 contingency table for each category c_i (c_i vs. not c_i).
- Macro-averaging.
  - Average measure over categories.
- Micro-averaging.
  - One table for all data by summing the scores in each cell for all categories.

Decision Trees

- Assign docs to category "earnings".



Data Representation Model

- Choose 20 words whose X^2 score are highest in the training set.
- Each document is assigned as a vector
  - $\vec{x} = (s_{1j}, ..., s_{kj})$
  - $s_{ij} = round(10 \times \frac{1 + \log(tf_{ij})}{1 + \log(l_j)})$

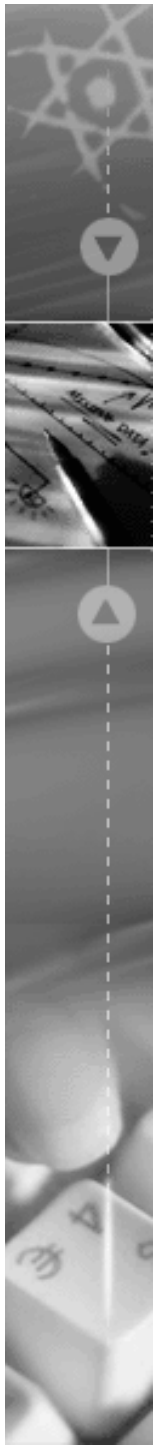
Term Frequency

Document length

Example of Representation

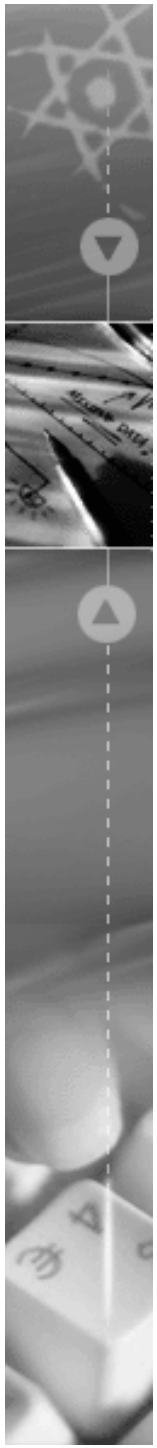
Word w^j	Term weight s_{ij}	Classification
vs	5	
mln	5	
cts	3	
;	3	
&	3	
000	4	
loss	0	
'	0	
"	0	
3	4	
profit	0	
dlrs	3	
1	2	
pct	0	
is	0	
s	0	
that	0	
net	3	
lt	2	
at	0	

$\vec{x} =$ (the term weight column) $\qquad c = 1$



Training Procedure

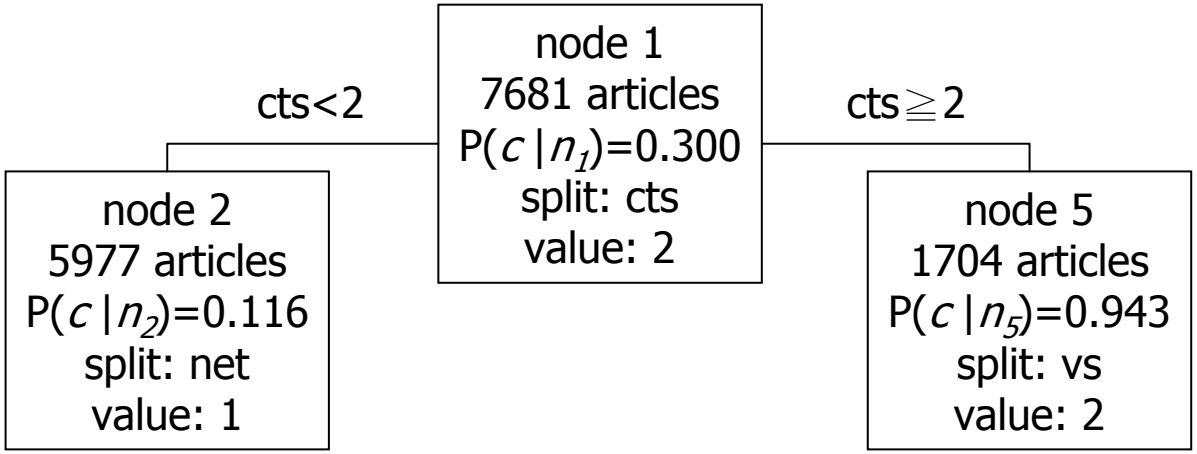
- Splitting criterion
  - finding the feature and its value.
- Stopping criterion
  - determine when to stop splitting
  - leaf node.
- Decision trees are built
  - by growing a large tree.
  - pruning back to a reasonable size.



Splitting Criterion- Maximum Information Gain

- Difference of the entropy of the mother node and weighted sum of child nodes.
- $G(a, y) = H(t) - H(t \mid a) = H(t) - (p_L H(t_L) + p_R H(t_R))$
- (a, y, t)=split (attribute, value, distribution).
- p_L, p_R: proportion elements of left and right nodes.
- t_L, t_R: distribution of the left and right nodes.

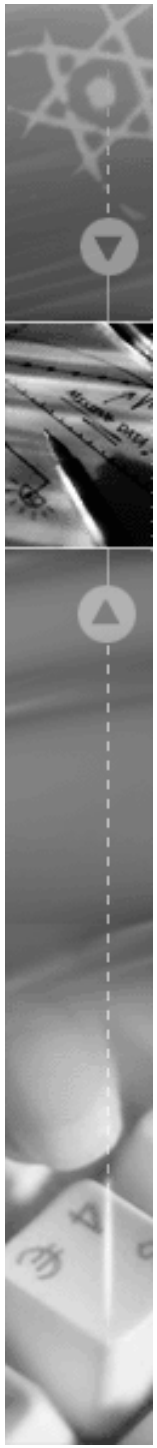
Example of Maximum Information Gain



$$G(a, y) = H(t) - H(t \mid a) = H(t) - (p_L H(t_L) + p_R H(t_R))$$

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

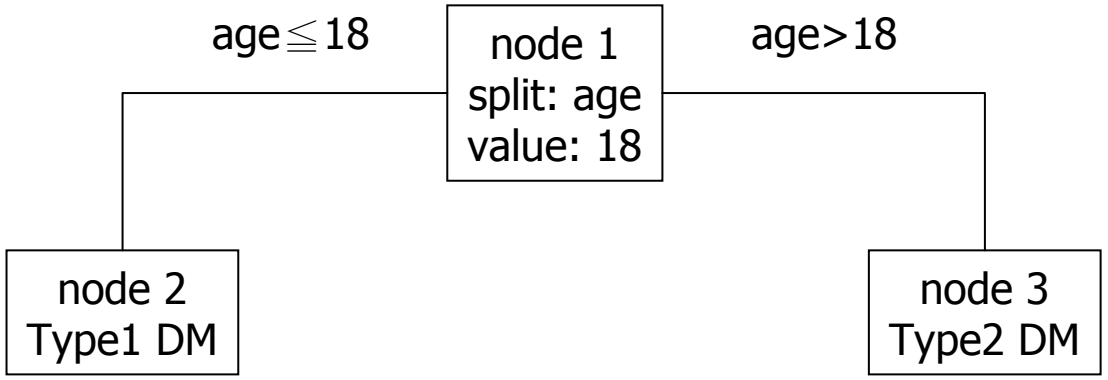
- entropy at node1 = - (0.3 * $\log_2(0.3)$ + 0.7 * $\log_2(0.7)$) = 0.611
- entropy at node2 = 0.359
- entropy at node5 = 0.219
- weighted sum of 2 and 5 $\frac{5977}{7681} \times 0.359 + \frac{1704}{7681} \times 0.219 = 0.328$
- information gain 0.611-0.328=0.238



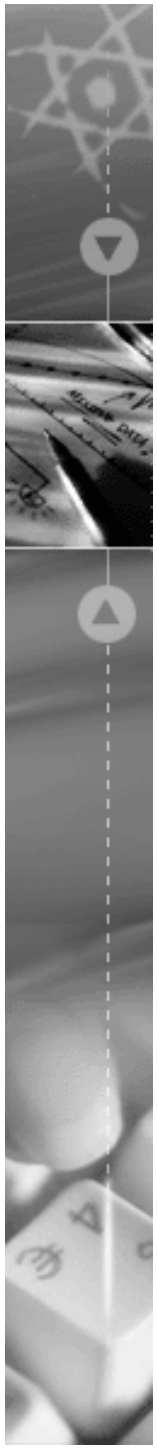
The Goals of Pruning

- Avoid overfitting.
  - too specific for the training data.
  - can not generalize to other data.
- Optimize performance.

Overfitting

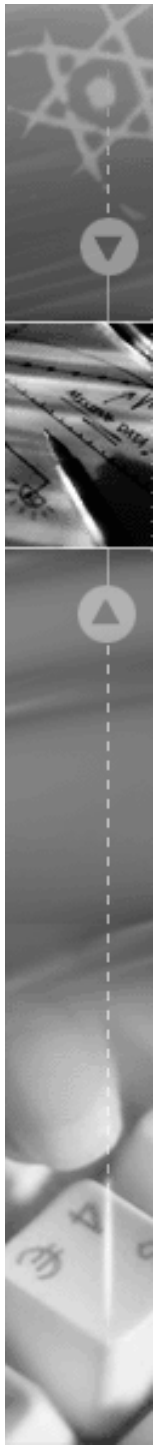


- Type1 diabetes mellitus usually occurs at young age in early year.
- Obesity is the main factor of young people with Type2 diabetes recently.



Validation

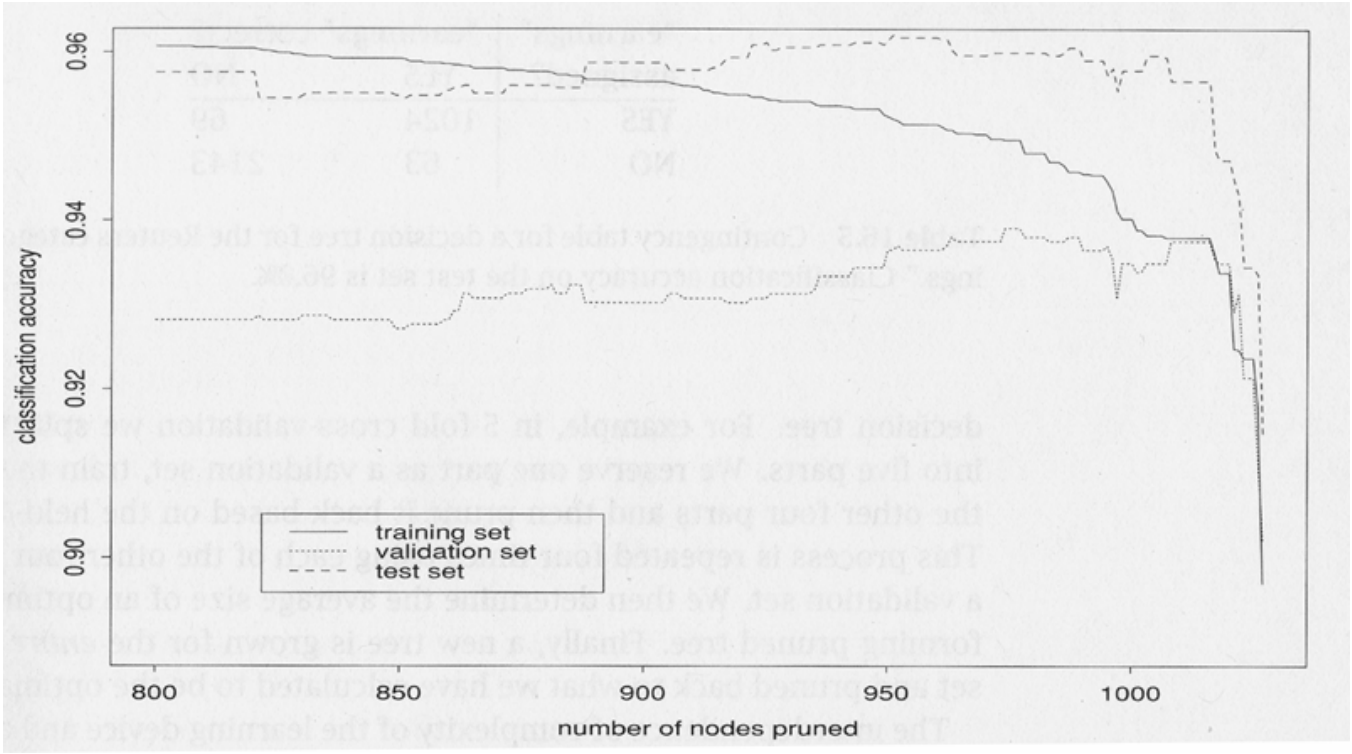
- Evaluate a classifier on a held-out or validation data set.
- Held-out data is a subset of the training data.
- Smooth counts from training data. or estimate other parameters based on held-out data.



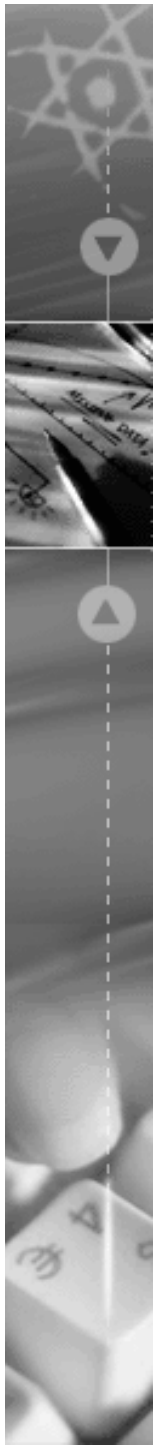
Alternative Pruning

- Keep the whole tree.
- Get more reliable probability distributions of higher nodes.

Pruning a Decision Tree

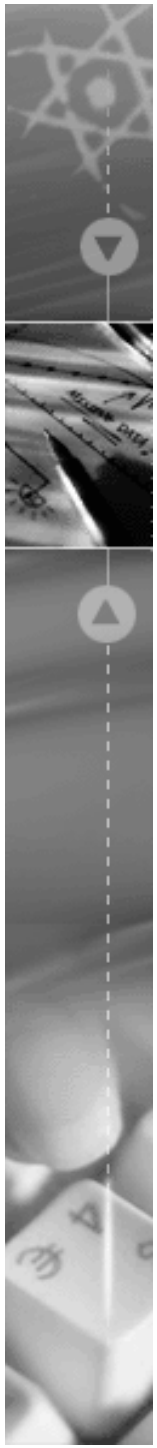


- Optimal performance:
  - 93.91% accuracy on validation set
  - 96.21% accuracy on test set



N-fold Cross-Validation

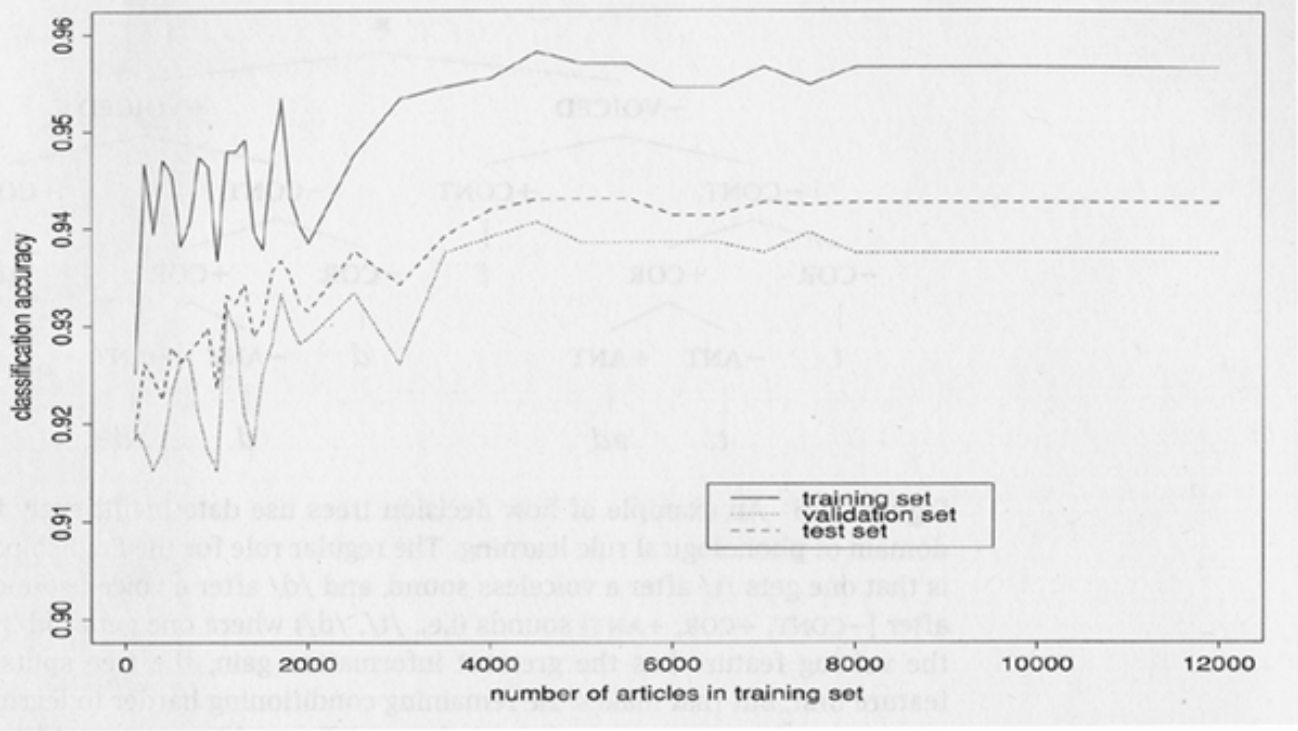
- Estimate a good validation set size.
- 5-fold cross-validation for example:
  - split the data into 5 parts
    - reserve 1 part as validation set and train the tree on other 4 parts.
    - prune the tree based on validation set.
    - repeat this process 4 times using each of the other 4 parts as a validation set.
  - Average size for optimal performance.



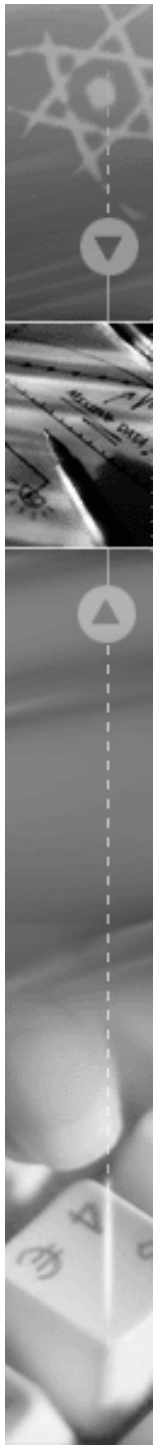
Learning Curve

- Training procedure is usually computationally expensive.
- Insufficient training data will result in suboptimal accuracy.
- Determine an appropriate training set.

Learning Curve (*cont.*)

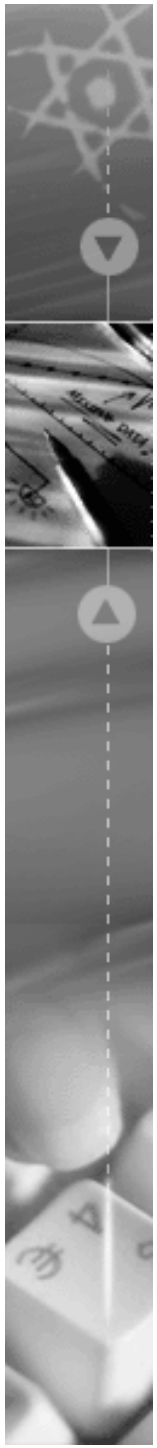


- Accuracy is highly variable for small training set.



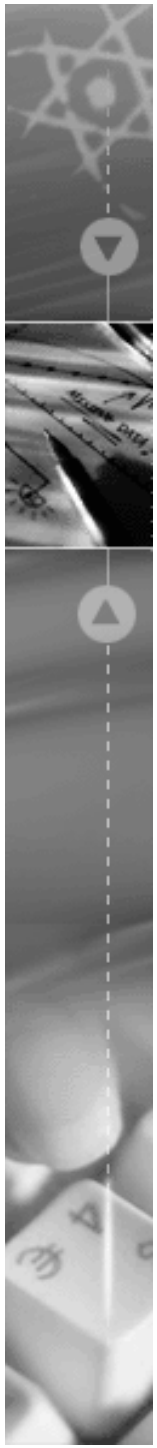
Advantages of Decision Trees

- Easily interpret and explain.
- Easily trace the path from root to leaf node.



K Nearest Neighbor (KNN)

- Classify a new object
  - Find the most similar objects in the training set.
  - Assign the category of the nearest neighbor.
- The complexity of KNN is finding a good similarity metrics.



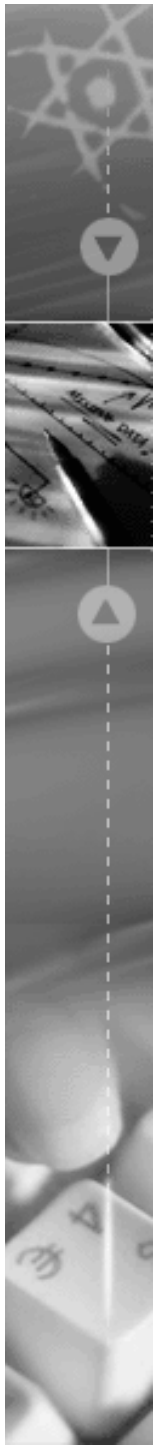
1NN Algorithm

- Categorize $\vec{y}$ based on training set *X*.
- Use cosine measure to estimate the similarity between $\vec{y}$ and any $\vec{x}$

$$sim_{\max}(\vec{y}) = \max_{\vec{x} \in X} sim(\vec{x}, \vec{y})$$

- Collect the subset of X that has highest similarity with $\vec{y}$

$$A = \{\vec{x} \in X \mid sim(\vec{x}, \vec{y}) = sim_{\max}(\vec{y})\}$$



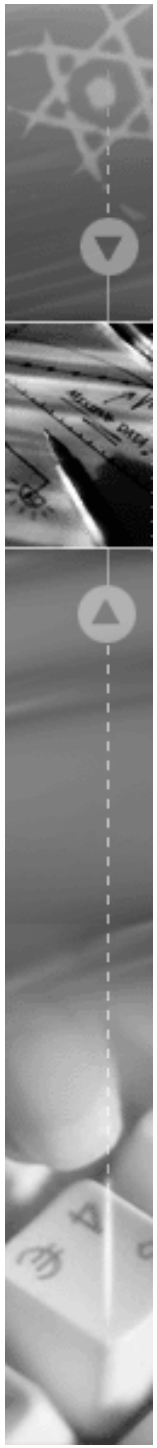
1NN Algorithm (*cont.*)

- Let n_1, n_2 be the number of elements that belong to class c_1, c_2

$$p(c_1 \mid \vec{y}) = \frac{n_1}{n_1 + n_2} \qquad p(c_2 \mid \vec{y}) = \frac{n_2}{n_1 + n_2}$$

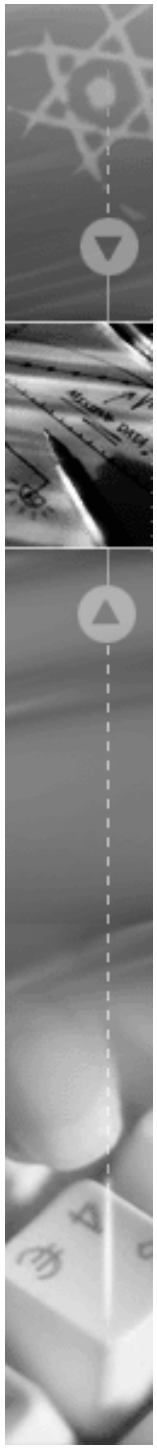
- Decide c_1 if $p(c_1 \mid \vec{y}) > p(c_2 \mid \vec{y})$

  c_2 otherwise



Generalization

- Choose *k* nearest neighbors.
- Give desirable weight to neighbors according to their similarity.



Difficulty of KNN

- Performance is very dependent on the right similarity metric.
- Computing similarity with all training exemplars takes time.