

Data Mining

CS57300

Purdue University

September 23, 2010

Project proposal

- **Due:** Tuesday Oct 5
- **Length:** 1/2 page
- **Content:**
 - The project's goals, including primary task and possible hypotheses
 - A description of the data that you will use
 - A list of the algorithms that you will develop and/or analyze

Decision trees (cont)

When to stop growing

- Full growth methods
 - All samples for at a node belong to the same class
 - There are no attributes left for further splits
 - There are no samples left
- What impact does this have on the quality of the learned trees?

Overfitting

- Overfitting the training data
 - Given a model space M , a model $m \in M$ is overfitting the training data if $\exists m' \in M$, such that m has smaller error than m' on the training data, but m' has smaller error on the entire distribution of instances
- Approaches for avoiding overfitting
 - Prepruning
 - Postpruning

Pruning

- Postpruning
 - Use a separate set of examples to evaluate the utility of pruning nodes from the tree (after tree is fully grown)
- Prepruning
 - Apply a statistical test to decide whether to expand a node
 - Use an explicit measure of complexity to penalize large trees (e.g., Minimum Description Length)

Algorithm comparison

- CART

- Evaluation criterion:
Gini index
- Search algorithm:
Simple to complex,
hill-climbing search
- Stopping criterion:
When leaves are pure
- Pruning mechanism:
**Cross-validation to select
gini threshold**

- C4.5

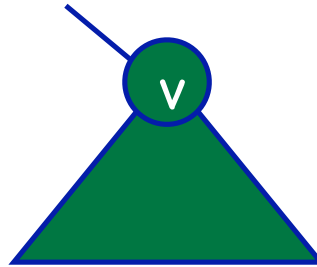
- Evaluation criterion:
Information gain
- Search algorithm:
Simple to complex,
hill-climbing search
- Stopping criterion:
When leaves are pure
- Pruning mechanism:
Reduce error pruning

Post-pruning methods

- Reduced error pruning
 - Quinlan 87, Mingers 87, Esposito et al. 96
- Minimal error pruning
 - Niblett & Bratko 86, Cestnik & Bratko 91
- Pessimistic error pruning
 - Quinlan 87
- Error-Based pruning
 - Quinlan 87
- Cost-Complexity pruning
 - Brieman et al. 84

Example: reduced error pruning

- Use **pruning set** to estimate accuracy in sub-trees and for individual nodes
- Let T be a sub-tree rooted at node v



- Define:
$$\text{Gain from pruning at } v = \# \text{misclassification in } T - \# \text{misclassification at } v$$
- Repeat: Prune at node with largest gain until only negative gain nodes remain
- “Bottom-up restriction”: T can only be pruned if it does not contain a sub-tree with lower error than T

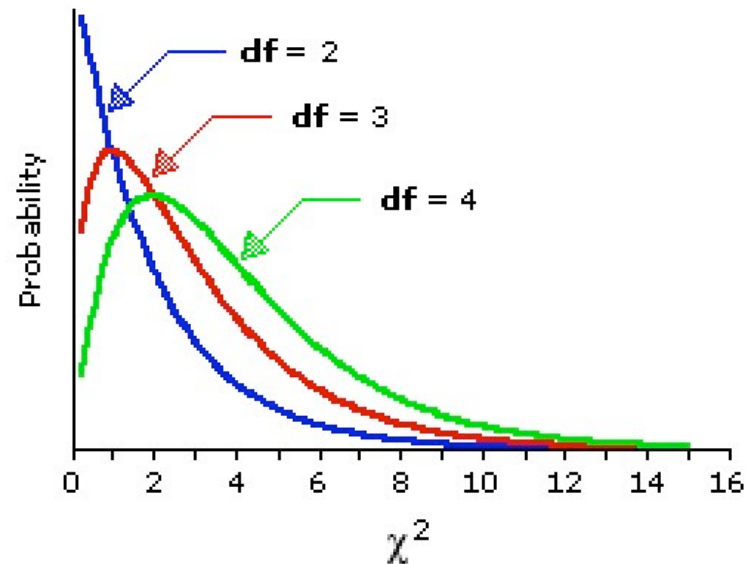
Pre-pruning methods

- Stop growing tree at some point during top-down construction when there is no longer sufficient data to make reliable decisions
- Approach:
 - Choose threshold on feature score
 - Stop splitting if the best feature score is below threshold

Determine threshold analytically

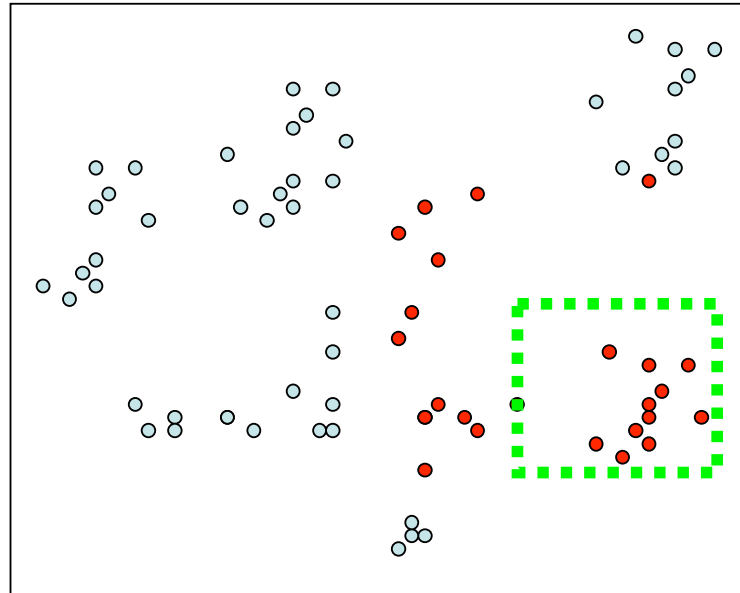
- Stop growing when chi-square feature score is not **statistically significant**
- Chi-square has known sampling distribution, can look up significance threshold
 - Degrees of freedom = $(\text{\#rows}-1)(\text{\#cols}-1)$
 - 2X2 table:
3.84 is 95% critical value

$$\chi^2 = \sum_{i=1}^k \frac{(o_i - e_i)^2}{e_i}$$



Movement in the table

		Actual	
		+	-
Predicted	+	a	b
	-	c	d



Can be used to make search more efficient

Naive Bayes classifiers

Classification as probability estimation

- Instead of learning a function h that assigns labels
- Learn a conditional probability distribution over the output of function f
- $P(F(\mathbf{x}) | \mathbf{x}) = P(f(\mathbf{x}) = Y | x_1, x_2, \dots, x_m)$
- Can use probabilities for the other two tasks
 - Classification
 - Ranking

Knowledge representation

- Goal: a system that takes an input a set of attributes, \mathbf{x} , and returns a probability distribution over labels: $P[F(\mathbf{x})|\mathbf{x}]$
- Bayesian networks are a tool for representing uncertain knowledge...
- Can we apply them for classification?
 - Yes! Label and each attribute is a random variable

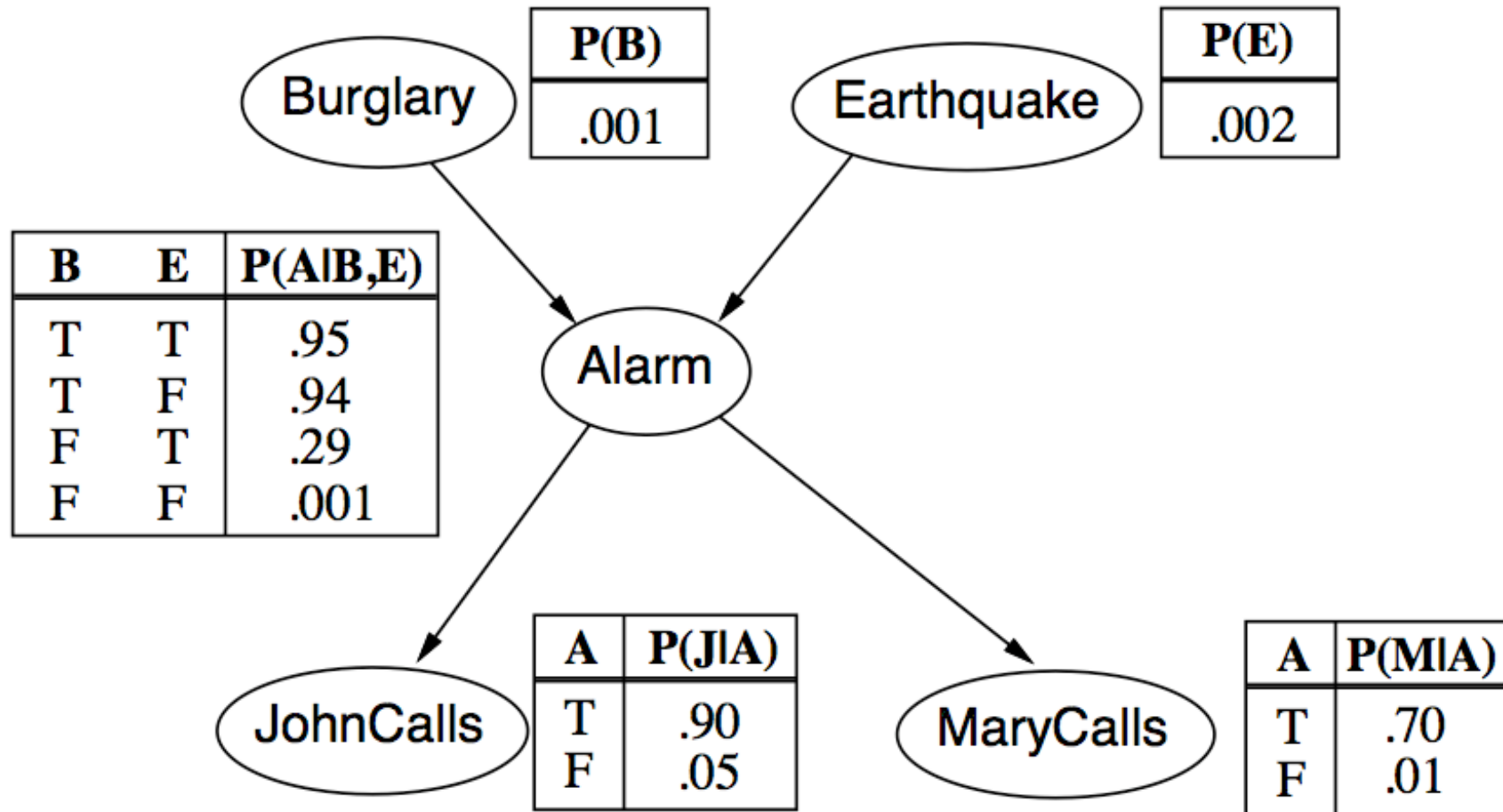
Bayesian networks

- Simple, graphical notation for conditional independence assertions
 - Compact representation for full joint distributions
- Syntax:
 - A set of nodes, one per variable
 - A directed, acyclic graph (link \approx "directly influences")
 - A conditional distribution for each node given its parents:
 $\mathbf{P}(X_i \mid \text{Parents}(X_i))$
 - In the simplest case, the conditional distribution is represented as a **conditional probability table** (CPT) giving the distribution over X_i for each combination of parent values

Example: Home security

- I'm at work, and my neighbor John calls to say my alarm is ringing, but my neighbor Mary doesn't call. We live in California, and sometimes the alarm is set off by minor earthquakes.
- Is there a burglar?
- Variables: *Burglary, Earthquake, Alarm, JohnCalls, MaryCalls*
- We have some probabilistic "causal" knowledge:
 - A burglar can set the alarm off
 - An earthquake can set the alarm off
 - The alarm can cause Mary to call
 - The alarm can cause John to call

Example: Home security



Semantics

- The full joint distribution is defined as the product of the local conditional distributions:

- $P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{Parents}(X_i))$

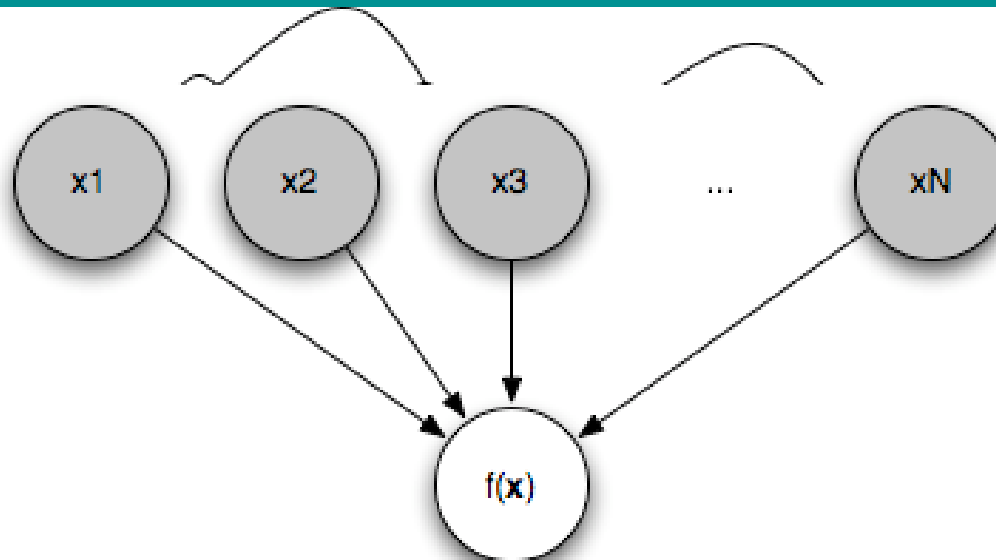
- Example:

- $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) = P(j|a) P(m|a) P(a|\neg b, \neg e) P(\neg b) P(\neg e)$

Challenge 1: choosing Bayes net structure

- How do we determine conditional independence relations?

Naïve Bayes assumption: attributes are conditionally independent given class label



Naive Bayes classifier

$$P(C|\mathbf{X}) = \frac{P(\mathbf{X}|C)P(C)}{P(\mathbf{X})}$$
$$\propto \prod_{i=1}^m P(X_i|C) P(C)$$

NBC learning

- Estimate prior $P(C)$ and conditional probability distributions $P(X_i | C)$ independently w/MLE
- $P(C)=9/14$
 $P(I=high|C=yes)=2/9$
 $P(I=med|C=yes)=4/9$
 $P(I=low|C=yes)=3/9$
etc.

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

Learning CPTs from examples

		X_1		
		Low	Medium	High
$f(\mathbf{x})$	True	10	13	17
	False	2	13	0

$$P[X_1 = \text{Low} \mid f(\mathbf{x}) = \text{True}] = \frac{10}{(10 + 13 + 17)}$$

$$P[f(\mathbf{x}) = \text{False}] = \frac{(2 + 13)}{(2 + 13 + 10 + 13 + 17)}$$

Zero counts are a problem

- If an attribute value does not occur in training example, we assign **zero** probability to that value
- How does that affect the conditional probability $P[F(x) | \mathbf{x}]$?
- It equals 0!!!
- Why is this a problem?
- Adjust for zero counts by “smoothing” probability estimates

Smoothing: Laplace correction

		X_1		
		Low	Medium	High
$f(\mathbf{x})$	True	10	13	17
	False	2	13	0

$$P[X_1 = \text{High} \mid f(\mathbf{x}) = \text{False}] = \frac{0 + 1}{(2 + 13 + 0) + 3}$$

Add uniform prior

Is assuming independence a problem?

- What is the effect on probability estimates?
 - Over-counting evidence, leads to overly confident probability estimate
- What is the effect on classification?
 - Less clear...
 - For a given input \mathbf{x} , suppose $f(\mathbf{x}) = \text{True}$
 - Naïve Bayes will correctly classify if

$$P[F(\mathbf{x}) = \text{True} \mid \mathbf{x}] > 0.5$$

Naive Bayes classifier

- Simplifying (naive) assumption: attributes are conditionally independent given the class
- Strengths:
 - Easy to implement
 - Often performs well even when assumption is violated
 - Can be learned incrementally
- Weaknesses:
 - Class conditional assumption produces skewed probability estimates
 - Dependencies among variables cannot be modeled

NBC learning

- Model space?
- Search algorithm?
- Scoring function?

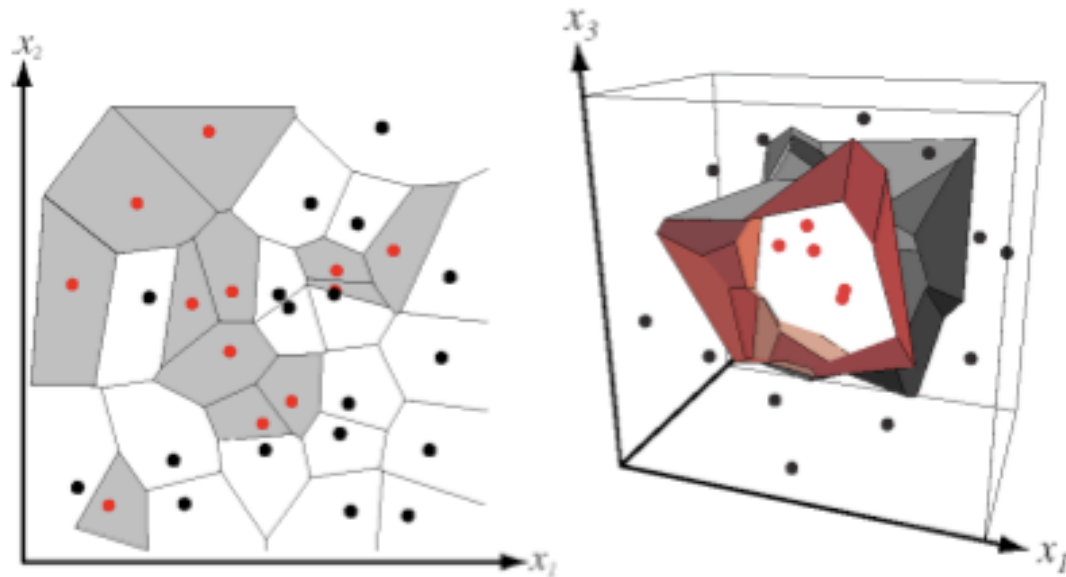
Other predictive models

Nearest neighbor

- Instance-based method
 - Store instance and delay processing until a new instance must be classified
 - All point represented in m -dimensional space
 - Nearest neighbors are calculated using Euclidean distance
- k -NN returns the most common value among k closest training examples
 - How to choose k ?

Nearest neighbor decision boundary

- All points in such a cell are labeled by the class of the training point, forming a Voronoi tessellation of the feature space.



Nearest neighbor

- Strengths:
 - Simple model, easy to implement
 - Very efficient learning: $O(1)$
- Weaknesses:
 - Inefficient inference: time and space $O(n)$
 - Curse of dimensionality

k-NN learning

- Model space?
- Search algorithm?
- Scoring function?