CS 730/730W/830: Intro AI

1 handout: slides
730W blog entries were due

Unsupervised Learning

Overview

modeling = predicting = understanding

clustering

k-Means Clustering

Naive Bayes model: choose class, generate attributes independently

mixture model: choose class, generate data

$$P(x|\theta) = \sum_k P(C = k|\theta_k)P(x|C = k, \theta_k)$$

eg, for mixture of Gaussians,

$$P(x|C = k, \mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\sigma_k^2\pi}} \exp\left(-\frac{(x - \mu_k)^2}{2\sigma_k^2}\right)$$

An Algorithm

Means represent the center of a cluster/class
Values for the means are the model
Model changes based on the classes assigned to the data

init the k means somehow
repeat until cluster assignments do not change:
  Assign each data point to the mean nearest to it
  Calculate new means for the data assigned to each cluster

An Algorithm

Means represent the center of a cluster/class
Values for the means are the model
Model changes based on the classes assigned to the data

init the k means somehow
repeat until cluster assignments do not change:
  Assign each data point to the mean nearest to it
  Calculate new means for the data assigned to each cluster

Example

An Algorithm

Means represent the center of a cluster/class
Values for the means are the model
Model changes based on the classes assigned to the data

init the k means somehow
repeat until cluster assignments do not change:
  Assign each data point to the mean nearest to it
  Calculate new means for the data assigned to each cluster

Example

Is the classification optimal?
What is it optimizing?

Expectation-Maximization

model parameters θ (eg, $\mu, \sigma^2, P(C = k)$)
observed variables x_j
hidden variables C_j

init the θ_k somehow
repeat until done:
- E: compute expected values of hidden vars: $P(C_j = k|x_j, \theta_k)$
  - eg by $\alpha P(C = k)P(x_j|C = k, \theta_k)$
- M: maximize data likelihood using current estimates:
  - θ_k, with each x_j weighted by $P(C_j = k|x_j)$, eg by

Expectation-Maximization

model parameters θ (eg, $\mu, \sigma^2, P(C = k)$)
observed variables x_j
hidden variables C_j

init the θ_k somehow
repeat until done:
  E: compute expected values of hidden vars: $P(C_j = k|x_j, \theta_k)$
    eg by $\alpha P(C = k)P(x_j|C = k, \theta_k)$
  M: maximize data likelihood using current estimates:
    θ_k, with each x_j weighted by $P(C_j = k|x_j)$, eg by

$$\theta \leftarrow \operatorname*{argmax}_{\theta} \sum_{z} P(Z = z|x, \theta)P(x, Z = z|\theta)$$

greedy increase of data likelihood

Expectation-Maximization

Features

- Probabilistic clustering
- Explicit model
- Locally optimal

Issues

- Number of classes (means, Gaussians, etc.)
- Local maxima

Agglomerative Clustering

dendrogram
$O(n^2)$ vs $O(kn)$
AutoClass

Break

- asst 5
- exam 2
- projects

Bayesian Networks

The Alarm Domain

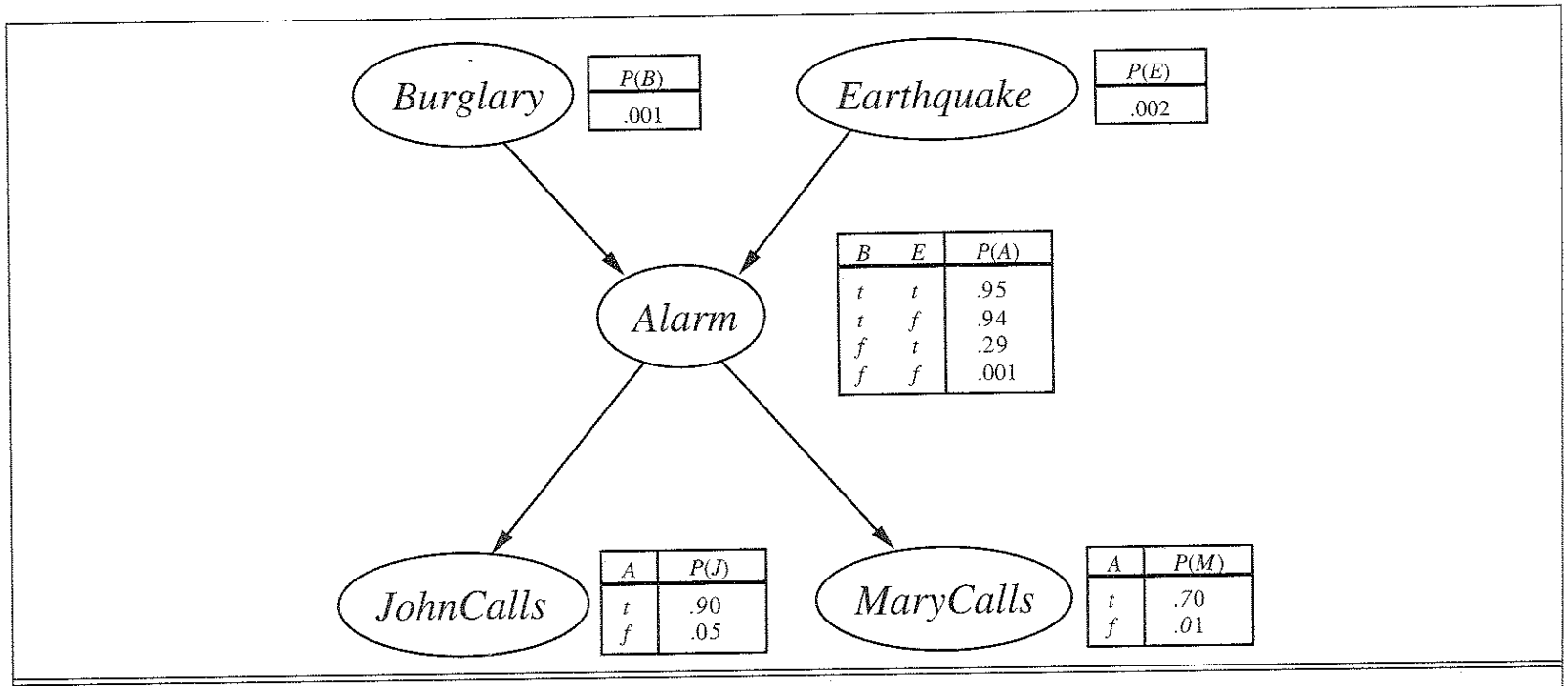

Figure 14.2 A typical Bayesian network, showing both the topology and the conditional probability tables (CPTs). In the CPTs, the letters B, E, A, J, and M stand for *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, and *MaryCalls*, respectively.

Probabilistic Models

MDPs:
Naive Bayes:
k-Means:

Representation: variables, connectives
Inference: approximate, exact

The Full Joint Distribution

ultimate power: knowing the probability of every possible atomic event (combination of values)

The Full Joint Distribution

ultimate power: knowing the probability of every possible atomic event (combination of values)

simple inference via enumeration over the joint:

what is distribution of X given evidence e and unobserved Y

$$P(X|e) = \frac{P(e|X)P(X)}{P(e)} = \alpha P(X, e) = \alpha \sum_y P(X, e, y)$$

Bayes Net = joint probability distribution

The Magic of Independence

In general:

$$P(x_1, \ldots, x_n) = P(x_n | x_{n-1}, \ldots, x_1) P(x_{n-1}, \ldots, x_1)$$

The Magic of Independence

In general:

$$\begin{aligned} P(x_1, \ldots, x_n) &= P(x_n | x_{n-1}, \ldots, x_1) P(x_{n-1}, \ldots, x_1) \\ &= \prod_{i=1}^{n} P(x_i | x_{i-1}, \ldots, x_1) \end{aligned}$$

The Magic of Independence

In general:

$$\begin{aligned} P(x_1, \ldots, x_n) &= P(x_n | x_{n-1}, \ldots, x_1) P(x_{n-1}, \ldots, x_1) \\ &= \prod_{i=1}^{n} P(x_i | x_{i-1}, \ldots, x_1) \end{aligned}$$

A Bayesian net specifies independence:

$$P(X_i | X_{i-1}, \ldots, X_1) = P(X_i | parents(X_i))$$

The Magic of Independence

In general:

$$\begin{aligned} P(x_1, \ldots, x_n) &= P(x_n | x_{n-1}, \ldots, x_1) P(x_{n-1}, \ldots, x_1) \\ &= \prod_{i=1}^{n} P(x_i | x_{i-1}, \ldots, x_1) \end{aligned}$$

A Bayesian net specifies independence:

$$P(X_i | X_{i-1}, \ldots, X_1) = P(X_i | parents(X_i))$$

So we get:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

The Magic of Independence

In general:

$$\begin{aligned} P(x_1, \ldots, x_n) &= P(x_n | x_{n-1}, \ldots, x_1) P(x_{n-1}, \ldots, x_1) \\ &= \prod_{i=1}^{n} P(x_i | x_{i-1}, \ldots, x_1) \end{aligned}$$

A Bayesian net specifies independence:

$$P(X_i | X_{i-1}, \ldots, X_1) = P(X_i | parents(X_i))$$

So we get:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

For n b-ary variables with p parents, that's nb^p instead of b^n!

EOLQs

- What question didn't you get to ask today?
- What's still confusing?
- What would you like to hear more about?

Please write down your most pressing question about AI and put it in the box on your way out.
Thanks!