Learning Bayesian Networks (part 3)

Mark Craven and David Page
Computer Sciences 760
Spring 2018

www.biostat.wisc.edu/~craven/cs760/

Some of the slides in these lectures have been adapted/borrowed from materials developed by Tom Dietterich, Pedro Domingos, Tom Mitchell, David Page, and Jude Shavlik

Goals for the lecture

you should understand the following concepts

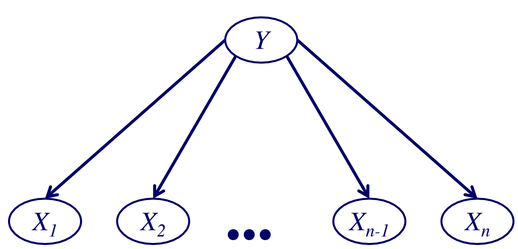
- the naïve Bayes classifier
- the Tree Augmented Network (TAN) algorithm

Bayes nets for classification

- the learning methods for BNs we've discussed so far can be thought of as being unsupervised
  - the learned models are not constructed to predict the value of a special class variable
  - instead, they can predict values for arbitrarily selected query variables
- now let's consider BN learning for a standard supervised task (learn a model to predict Y given $X_1 \ldots X_n$)

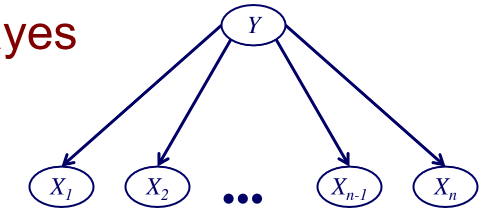
Naïve Bayes

- one very simple BN approach for supervised tasks is *naïve Bayes*
- in naïve Bayes, we assume that all features X_i are conditionally independent given the class Y



$$P(X_1, \ \ldots, X_n, \ Y) = P(Y)\prod_{i=1}^{n} P(X_i \mid Y)$$

Naïve Bayes



Learning

- estimate $P(Y = y)$ for each value of the class variable Y
- *estimate* $P(X_i = x \mid Y = y)$ *for each* X_i

Classification: use Bayes' Rule

$$P(Y = y \mid \boldsymbol{x}) = \frac{P(y)P(\boldsymbol{x} \mid y)}{\sum_{y' \in \text{values}(Y)} P(y')P(\boldsymbol{x} \mid y')} = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{\sum_{y' \in \text{values}(Y)} \left(P(y')\prod_{i=1}^{n} P(x_i \mid y') \right)}$$

Naïve Bayes vs. BNs learned with an unsupervised structure search

test-set error on 25 classification data sets from the UC-Irvine Repository

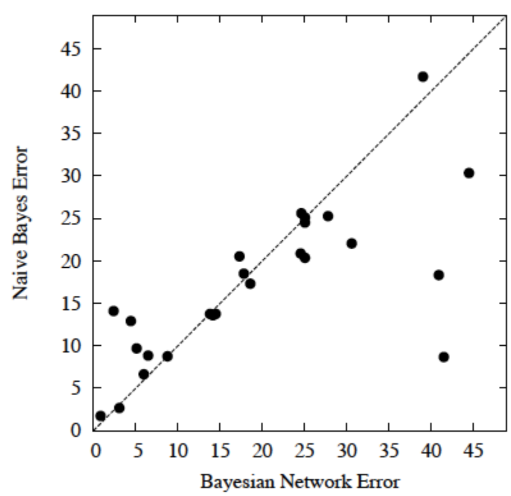


Figure from Friedman et al., *Machine Learning* 1997

The Tree Augmented Network (TAN) algorithm

[Friedman et al., *Machine Learning* 1997]

- learns a tree structure to augment the edges of a naïve Bayes network

- algorithm
  1. compute weight $I(X_i, X_j \mid Y)$ for each possible edge (X_i, X_j) between features
  2. find maximum weight spanning tree (MST) for graph over $X_1 \ldots X_n$
  3. assign edge directions in MST
  4. construct a TAN model by adding node for Y and an edge from Y to each X_i

Conditional mutual information in the TAN algorithm

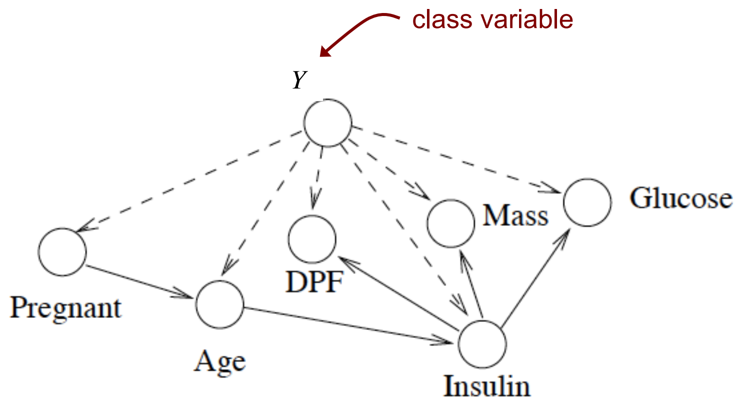
conditional mutual information is used to calculate edge weights

$$I(X_i, X_j \mid Y) =$$

$$\sum_{x_i \in \text{values}(X_i)} \sum_{x_j \in \text{values}(X_j)} \sum_{y \in \text{values}(Y)} P(x_i, x_j, y) \log_2 \frac{P(x_i, x_j \mid y)}{P(x_i \mid y) P(x_j \mid y)}$$

"how much information X_i provides about X_j when the value of Y is known"

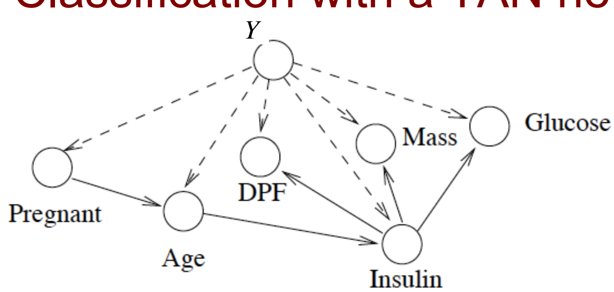
Example TAN network



naïve Bayes edges

edges determined by MST

Classification with a TAN network



As before use Bayes' Rule:

$$P(Y = y|\boldsymbol{x}) = \frac{P(y)P(\boldsymbol{x}|y)}{\sum_{y'} P(y')P(\boldsymbol{x}|y')}$$

In the example network, we calculate $P(\boldsymbol{x}|y)$ as:

$$P(\boldsymbol{x}|y) = P(pregnant\ |y)P(age|y, pregnant)P(insulin|y, age)P(dpf|y, insulin)$$
$$P(mass|y, insulin)P(glucose|y, insulin)$$

TAN vs. Chow-Liu

- TAN is mostly* focused on learning a Bayes net specifically for classification problems
- the MST includes only the feature variables (the class variable is used only for calculating edge weights)
- conditional mutual information is used instead of mutual information in determining edge weights in the undirected graph
- the directed graph determined from the MST is added to the $Y \rightarrow X_i$ edges that are in a naïve Bayes network

* although parameters are still set to maximize $P(y, \boldsymbol{x})$ instead of $P(y \mid \boldsymbol{x})$

TAN vs. Naïve Bayes

test-set error on 25 data sets from the UC-Irvine Repository

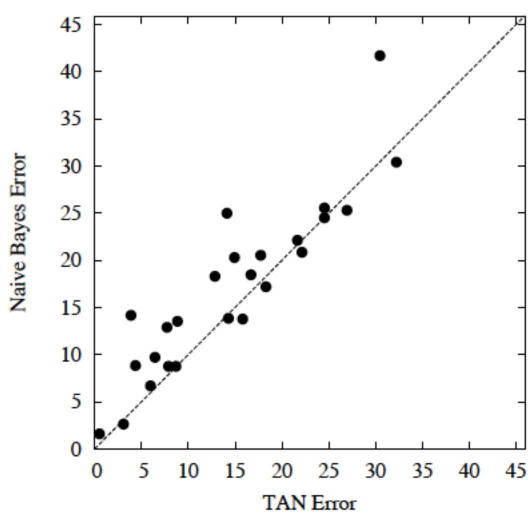


Figure from Friedman et al., *Machine Learning* 1997

Comments on Bayesian networks

- the BN representation has many advantages
  - easy to encode domain knowledge (direct dependencies, causality)
  - can represent uncertainty
  - principled methods for dealing with missing values
  - can answer arbitrary queries (in theory; in practice may be intractable)
- for supervised tasks, it may be advantageous to use a learning approach (e.g. TAN) that focuses on the dependencies that are most important

Comments on Bayesian networks (continued)

- although very simplistic, naïve Bayes often learns highly accurate models
- we focused on learning Bayes nets with only discrete variables; can also have numeric variables (although not as parents)
- BNs are one instance of a more general class of *probabilistic graphical models*