Bayesian Belief Networks

By Bartosz Bajer

Overview

- Short review in Elementary Probabilities
- Bayes Theorem
- Down falls of Naive Bayes Classifier
- Bayesian belief networks (BBN)
- BBN and Variable Dependence
- BBN Classic Example
- BBN Representation
- BBN Learning
- BBN Creation
- BBN and NLP

Introduction

- Bayesian Belief Networks are directed acyclic graphs that combine prior knowledge with observed data.
- They allow for probabilistic dependencies and probabilistic conditional independence
- This makes them more powerful then most previous models such as Naive Bayes Model
- These characteristics make it useful for NLP

Probabilities Axioms

- A and B are Boolean variables that represent the occurrence of an event.
- If an event is certain to occur then the probability is 1
- If an event is certain to not occur then the probability is 0.
- If the probability of the event is uncertain then the probability is between 0 and 1.

$$0 \leq P(A) \leq 1$$
$$P(True) = 1$$
$$P(False) = 0$$
$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$
$$P(\neg A) = 1 - P(A)$$

Probabilistic Inference

- Suppose you are one of the 1/10 people that have a headache (H)
- Suppose 1/40 of people have the flu (F).
- Suppose that half the people that have the flu also have a headache.
- Given the fact that you have a headache what are the chances that you have the flu?

$$P(H) = 1/10$$
$$P(F) = 1/40$$
$$P(H|F) = 1/2$$

$$P(F|H) = ?$$

Bayes Theorem

$$P(h|x)=\frac{(P(x|h)P(h))}{(P(x))}$$

$$P(F|H)=\frac{(P(H|F)P(F))}{(P(H))}=?$$

- P(h) = prior probability of hypothesis h
- P(x) = prior probability that examples is observed
- P(h|x) = posterior probability of h given x
- P(x|h)=conditional probability of x given h (often called likelihood of x given h)

Bayes Theorem

$$P(h|x) = \frac{(P(x|h)P(h))}{(P(x))}$$

$$P(F|H) = \frac{(P(H|F)P(F))}{(P(H))}$$

$$P(F|H) = \frac{(0.5 * 0.025)}{0.1}$$

$$P(F|H) = 0.125$$

- P(h) = prior probability of hypothesis h
- P(x) = prior probability that examples is observed
- P(h|x) = posterior probability of h given x
- P(x|h)=conditional probability of x given h (often called likelihood of x given h)

Conditional Probability and The Chain Rule

- The probability that A and B occur.

$$P(A|B)=\frac{(P(A\wedge B))}{(P(B))}$$

$$P(A\wedge B)=P(A|B)P(B)$$

- What is the probability that a person has a head ache and the Flu?

$$P(H\wedge F)=?$$

Conditional Probability and The Chain Rule

- The probability that A and B occur.

$$P(A|B) = \frac{(P(A \wedge B))}{(P(B))}$$

$$P(A \wedge B) = P(A|B)\,P(B)$$

- What is the probability that a person has a head ache and the Flu?

$$P(H \wedge F) = P(H|F)\,P(F)$$

$$P(H \wedge F) = 0.5 * 0.025$$

$$P(H \wedge F) = 0.0125$$

Joint Probability Distribution

Headache	Flu	Probability
0	0	?
0	1	?
1	0	?
1	1	?

- How can we find these probabilities?

What we have so far:

$P(F)=0.025$ $P(\neg F)=0.975$

$P(H)=0.1$ $P(\neg H)=0.9$

$P(H|F)=0.5$ $P(\neg H|F)=0.5$

$P(F|H)=0.125$ $P(\neg F|H)=0.8875$

$P(H \wedge F)=0.0125$

Joint Probability Distribution

Headache	Flu	Probability
0	0	0.888
0	1	0.125
1	0	0.088
1	1	0.125

- We can use Bayes Theorem and Chain Rule to generate the joint probability distribution table for headache

What we have so far:

$$P(F)=0.025 \qquad P(\neg F)=0.975$$
$$P(H)=0.1 \qquad P(\neg H)=0.9$$
$$P(H|F)=0.5 \qquad P(\neg H|F)=0.5$$
$$P(F|H)=0.125 \qquad P(\neg F|H)=0.8875$$
$$P(H\wedge F)=0.0125$$

We can now find:

$$P(\neg H\wedge\neg F)=P(\neg H|\neg F)\,P(\neg F)=0.909*0.975=0.886$$

$$P(\neg H\wedge F)=P(\neg H|F)\,P(F)=0.5*0.025=0.0125$$

$$P(H\wedge\neg F)=P(H|\neg F)\,P(\neg F)=\frac{(P(\neg F|H)\,P(H))}{(P(\neg F))}\,P(\neg F)$$
$$P(H\wedge\neg F)=(P(\neg F|H)\,P(H))=0.8875*0.1=0.088$$

Maximum a Posteriori Hypothesis

$$P(h|x) = \frac{(P(x|h)\,P(h))}{(P(x))}$$

- Imagine we need to find the most probable hypothesis h from a set of examples.
- We can find it using a method called Maximum a Posteriori Hypothesis.

$$h_{MAP}(X) = \underset{h \in H}{argmax}\, P(h|x) = \underset{h \in H}{argmax} \frac{(P(x|h)\,P(h))}{(P(x))} = \underset{h \in H}{argmax}\, P(x|h)\,P(h)$$

Naive Bayes Classifier

- Naive Bayes classifier is naive because it assumes that values of the attributes are conditionally independent given a hypothesis

$$P(x_1, x_2, \ldots x_n | c_j) = \prod_i P(x_i | c_j)$$

$$c_{nb} = \underset{c_j \in C}{argmax}\, P(c_j) P(x_1, x_2, \ldots x_n | c_j) = \underset{c_j \in C}{argmax}\, P(c_j) \prod_i P(x_i | c_j)$$

Probability Estimation

- We can estimate the unknown values to cj and a xi given cj as follows:

$$P(c_j) = \frac{(\text{\# of training examples of class } c_j)}{(\text{\# of training examples})}$$

$$P(x_i | c_j) = \frac{(\text{\# of training examples of class } c_j \text{ with } x_i \text{ for } A_i)}{(\text{\# number of training examples of class } c_j)}$$

Naive Bayes Algorithm (Learning from examples)

For each class c_j

$$P(c_j) <- estimate\, P(C_j)$$

For each attribute for which x_i is a value

$$P(x_i|c_j) <- estimate\, P(x_i|c_j)$$

Classify new instance x

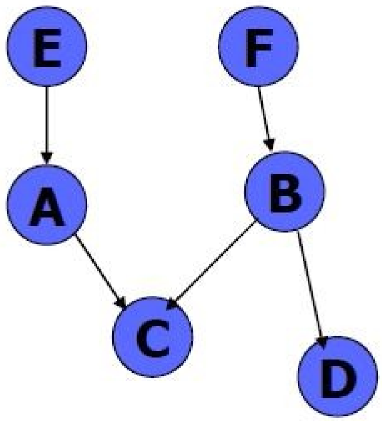
$$c_{nb} = \underset{c_j \in C}{argmax}\, P(c_j) \prod_i P(x_i|c_j)$$

A Problem with Naive Bayes Classification

- The assumption that all class attributes are independent results in a loss of accuracy
  - Recall the example about headaches and flu shown before. Clearly there is a dependencies between attributes which a naive classifier would not be able to model.
- The solution?
  - Bayesian Belief Networks

Bayesian Belief Networks (BBN)

- A *directed acyclic graph*: represents dependency among variables (attributes)



- ***Nodes***: variables (including class attribute)
- ***Links***: dependencies (e.g., A dependes on E)
- ***Parents***: immediate predecessors. E.g., A,B are the parents of C. B is the parent of D
- ***Descendent***: X is a descendent of Y if there is a direct path from Y to X.
- ***Conditional Independency***:
  - Assume: each variable is conditionally independent of its nondescendants given its parents.
  - Definition: X is *conditionally independent* of Y given Z iff P(X|Y,Z)=P(X|Z)
  - E.g.: C is conditional independent of D given A and B. Thus, P(C|A, B, D)=P(C|A, B)
- ***Acyclic***: has no loops or cycles
- A *conditional probability table* (CPT) for each variable X: specifies the conditional probability distribution P(X|Parents(X)).

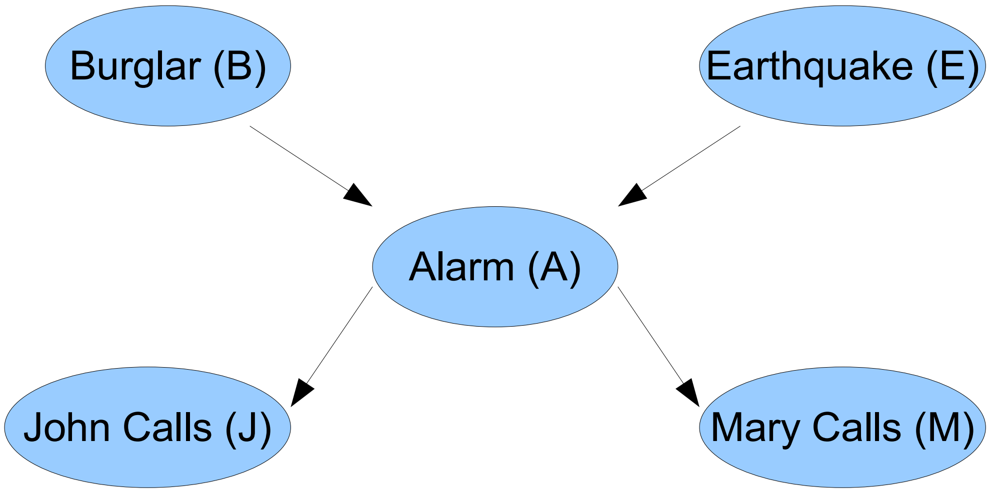
Image taken from: Data Mining (CSE 6412) Bayesian Classification Slide by Aijun Ann

Dependence and Independence of Bayesian Belief Networks

- In other words BBN allow for dependency among variables but allow independence among subsets of variables
- Each variable is conditionally independent of all its non descendant in the graph given the value of all its parents.

$$P(x_1 .. x_n) = \prod_{x_i \in X} P(x_i | parents(x_i))$$

The Classic Example



- You go on vacation. You have a new burglar alarm setup that detects burglary well but has a chance of responding to earthquakes.
- In case the alarm goes your two neighbors John and Mary can call you to inform you of the situation. Unfortunately,
- John has a tendency to confuse the alarm with the phone ringing
- Mary is slightly deaf.

Classic Example: Chain Rule

$$P(B,E,A,J,M)=P(B)P(E|B)P(A|B,E)P(J|A,B,E)P(M|J,A,B,E)$$

- Recall benefits of Bayesian Networks.

Classic Example: Chain Rule

$$P(B,E,A,J,M)=P(B)P(E|B)P(A|B,E)P(J|A,B,E)P(M|J,A,B,E)$$

- Recall benefits of Bayesian Networks.

$$P(B,E,A,J,M)=P(B)P(E)P(A|B,E)P(J|A)P(M|A)$$

Classic Example: Conditional Probability Tables (CPT)

B	P(B)
T	0.001
F	0.999

E	P(E)
T	0.002
F	0.998

B	E	A	P(A|B,E)
T	T	T	0.95
T	T	F	0.05
T	F	T	0.94
T	F	F	0.06
F	T	T	0.29
F	T	F	0.71
F	F	T	0.001
F	F	F	0.999

A	J	P(J|A)
T	T	0.90
T	F	0.10
F	T	0.05
F	F	0.95

A	M	P(M|A)
T	T	0.70
T	F	0.30
F	T	0.01
F	F	0.99

- Recall benefits of Bayesian Networks.

Classic Example: Inference

- Lets infer the probability that the burglar is not in the house given that John heard the alarm

Calculate $P(B=F, J=T)$:

$$P(B=F, J=T) = \sum_{E,A,M} P(B=F, E, A, J=T, M)$$

$$P(B=F, J=T) = \sum_{E,A,M} P(B=F) P(E) P(A \vee B=F, E) P(J=T \vee A) P(M \vee A)$$

$$\begin{aligned}
P(B=F, J=T) = {} & P(B=F) P(E=T) P(A=T \vee B=F, E=T) P(J=T \vee A=T) P(M=T \vee A=T) \\
& + P(B=F) P(E=T) P(A=T \vee B=F, E=T) P(J=T \vee A=T) P(M=F \vee A=T) \\
& + P(B=F) P(E=T) P(A=F \vee B=F, E=T) P(J=T \vee A=F) P(M=T \vee A=F) \\
& + P(B=F) P(E=T) P(A=F \vee B=F, E=T) P(J=T \vee A=F) P(M=F \vee A=F) \\
& + P(B=F) P(E=F) P(A=T \vee B=F, E=F) P(J=T \vee A=T) P(M=T \vee A=T) \\
& + P(B=F) P(E=F) P(A=T \vee B=F, E=F) P(J=T \vee A=T) P(M=F \vee A=T) \\
& + P(B=F) P(E=F) P(A=F \vee B=F, E=F) P(J=T \vee A=F) P(M=T \vee A=F) \\
& + P(B=F) P(E=F) P(A=F \vee B=F, E=F) P(J=T \vee A=F) P(M=F \vee A=F)
\end{aligned}$$

$$\begin{aligned}
P(B=F, J=T) = {} & 0.999 \cdot 0.002 \cdot 0.29 \cdot 0.9 \cdot 0.7 \\
& + 0.999 \cdot 0.002 \cdot 0.29 \cdot 0.9 \cdot 0.3 \\
& + 0.999 \cdot 0.002 \cdot 0.71 \cdot 0.05 \cdot 0.01 \\
& + 0.999 \cdot 0.002 \cdot 0.71 \cdot 0.05 \cdot 0.99 \\
& + 0.999 \cdot 0.998 \cdot 0.001 \cdot 0.9 \cdot 0.7 \\
& + 0.999 \cdot 0.998 \cdot 0.001 \cdot 0.9 \cdot 0.3 \\
& + 0.999 \cdot 0.998 \cdot 0.999 \cdot 0.05 \cdot 0.01 \\
& + 0.999 \cdot 0.998 \cdot 0.999 \cdot 0.05 \cdot 0.99
\end{aligned}$$

$$P(B=F, J=T) = 5.12899587 \cdot 10^{-2}$$

Representational Power of BBN

- BBN can represent other models such as fully joint distribution, fully independent model, naive bayes model, and HMM model.

Learning Bayesian Networks

- If the structure of the Bayesian Network is known the simply just learn the CPTs for each variable in the network by estimating the conditional probabilities from a training set. (Similar to naive Bayes classifier)
- What if the structure is unknown?

Building Bayesian Networks

- Problem: Find the most probable Bayes network structure given a database
- Bayesian Networks can be built using the K2 algorithm
- The algorithm heuristically searches for the most probable belief network structure given a dataset of cases.
- Input: n number of nodes, an ordering of the nodes, and upper bound u on the number of parents a node may have, and a data set D containing m cases.
- Output: The set of root parent nodes.

Building Bayesian Networks

- Structures are ranked by their posterior probabilities using the following:

$$g(i,\pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$$

- For more details see: "A Bayesian Method for the Induction of Probabilistic Networks from Data", Gregory F. Cooper and Edward Herskovits, Machine Learning 9, 1992

BBN and NLP

- So how does BBN relate to NLP?
  - Word recognition for the English Language (kinda like the monkey problem)
  - We need a data set (English: books, articles, etc)
  - We feed the data set in to the BBN structure creator (the structure is already present for us in the words it self)
  - We the generate the conditional probabilities based on the data set.
- But BBN can be even more powerful

BBN and NLP continued

- Consider the following paper:
  - X. Jin, A. Xu, R. Bie, X. Shen, M. Yin. Spam Email Filtering with Bayesian Belief Network: using Relevant Words, *IEEE International Conference on Granular Computing, 2006*
    - In the paper the authors attempt to classify whether an email is spam or non spam.
    - Classification was based on the contents of the email itself

BBN and NLP continued

- The authors used 3 different criteria for relevant word selection
  - Information Gain

$$InfoGain = [\sum_{k=1}^{m} -(\frac{N_{C_k}}{N})\log(\frac{N_{C_k}}{N})] - [\sum_{v=1}^{V}(\frac{N^{(v)}}{N})\sum_{k=1}^{m} -(\frac{N_{C_k}^{(v)}}{N^{(v)}})\log(\frac{N_{C_k}^{(v)}}{N^{(v)}})]$$

  - Gain Ratio

$$GainRatio = InfoGain / [\sum_{k=1}^{m} -(\frac{N_{C_k}}{N})\log(\frac{N_{C_k}}{N})]$$

  - Chi Squared

$$ChiSqured = \chi^2 = \sum_{k=1}^{m}\sum_{v=1}^{V}\frac{(N_{C_k}^{(v)} - \tilde{N}_{C_k}^{(v)})^2}{\tilde{N}_{C_k}^{(v)}}$$

BBN and NLP continued

- Using the word selection algorithms the authors found a “good” subset of words to use as a learning data set
- The authors used BBN classifiers/model among others (such as Naive Bayes Classifier) to filter emails as spam and none spam.
- They found that BBN out perform all other models for email filtering with a 97.6% accuracy.
- The authors attribute this outcome due to BBN ability to learn dependencies.

Conclusion

- Bayesian Belief Networks combine prior knowledge with observed data
- They allow for both dependencies and conditional independencies
- They have a flexible structure and can represent other probabilistic models
- These features make them powerful for modeling probabilities

Questions?