NLP 201: Bayesian Networks and Bayesian Learning

Jeffrey Flanigan

University of California Santa Cruz

jmflanig@ucsc.edu

Fall 2023

Outline

- Bayesian Networks (BNs)
- BN Examples
- Parameter Estimation for BN
- Sample from BNs
- Bayesian Learning

Graphical Models and Bayesian Networks

Defining joint probability distributions

- By the **chain rule** of probability,

$$\begin{aligned} p(x_{1:V}) &= p(x_1)p(x_2|x_1)p(x_3|x_2,x_1) \\ & \quad p(x_4|x_1,x_2,x_3)\ldots p(x_V|x_{1:V-1}) \end{aligned}$$

- $p(x_t|\mathbf{x}_{1:t-1})$ needs $O(K^t)$ parameters if K states per variable.
- We will make **conditional independence** assumptions to simplify things.

Conditional independence

- Definition

$$X \perp Y|Z \iff p(X, Y|Z) = p(X|Z)p(Y|Z)$$

- Example: first order Markov chain: "the future is independent of the past given the present":

$$y_{t+1} \perp \mathbf{y}_{1:t-1}|y_t$$

- Joint distribution

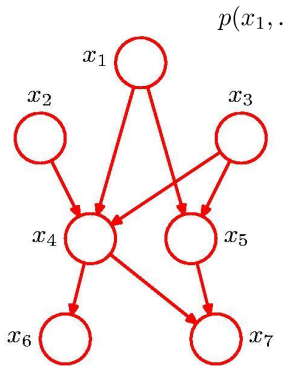
$$p(y_{1:T}) = p(y_1)\prod_{t=2}^{T} p(y_t|y_{1:t-1}) \stackrel{*}{=} p(y_1)\prod_{t=2}^{T} p(y_t|y_{t-1})$$

This is characterized by an initial distribution over states, $p(y_1 = i)$, plus a **state transition matrix** $p(y_t = j|y_{t-1} = i)$.

Graphical models

- Nodes represent random variables
- Edges represent conditional independence.
- Details of CI depends on whether the graph is directed ("Bayes net") or undirected ("Markov random field")
- Structure of graph brings statistical and computational efficiencies (less data, less time).

Bayesian Networks



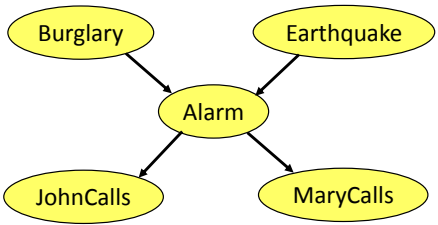
General Factorization

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)$$

Slide from Bishop

Bayesian Networks

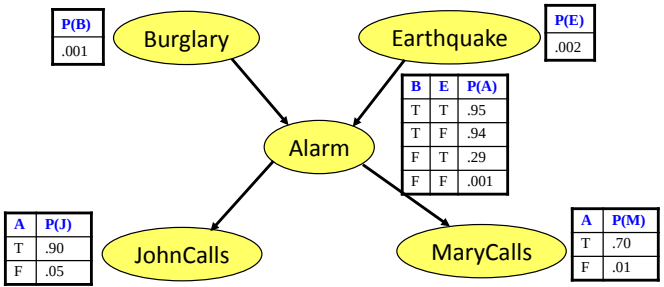
- Directed Acyclic Graph (DAG)
  - Nodes are random variables
  - Edges indicate causal influences



Slide credit: Ray Mooney

Conditional Probability Tables

- Each node has a **conditional probability table** (**CPT**) that gives the probability of each of its values given every possible combination of values for its parents (conditioning case).
  - Roots (sources) of the DAG that have no parents are given prior probabilities.



Slide credit: Ray Mooney

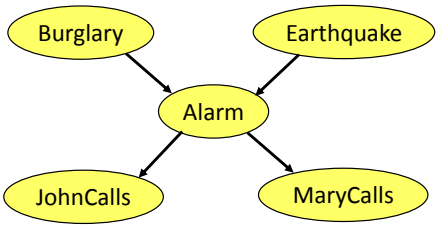
CPT Comments

- Probability of false not given since rows must add to 1.
- Example requires 10 parameters rather than $2^5-1=31$ for specifying the full joint distribution.
- Number of parameters in the CPT for a node is exponential in the number of parents.

Slide credit: Ray Mooney

Bayesian Networks

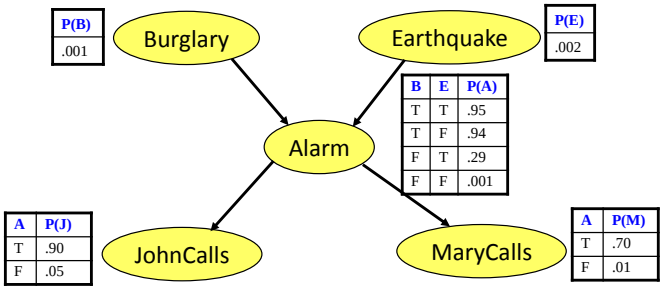
- Directed Acyclic Graph (DAG)
  - Nodes are random variables
  - Edges indicate causal influences



Slide credit: Ray Mooney

Conditional Probability Tables

- Each node has a **conditional probability table** (**CPT**) that gives the probability of each of its values given every possible combination of values for its parents (conditioning case).
  - Roots (sources) of the DAG that have no parents are given prior probabilities.



Slide credit: Ray Mooney

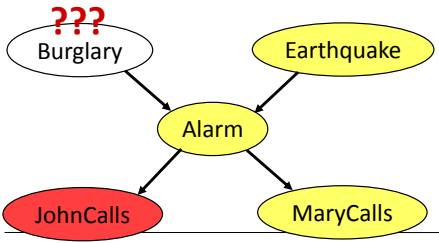
CPT Comments

- Probability of false not given since rows must add to 1.
- Example requires 10 parameters rather than $2^5-1=31$ for specifying the full joint distribution.
- Number of parameters in the CPT for a node is exponential in the number of parents.

Slide credit: Ray Mooney

Bayes Net Inference

- Given known values for some **evidence variables**, determine the posterior probability of some **query variables**.
- Example: Given that John calls, what is the probability that there is a Burglary?



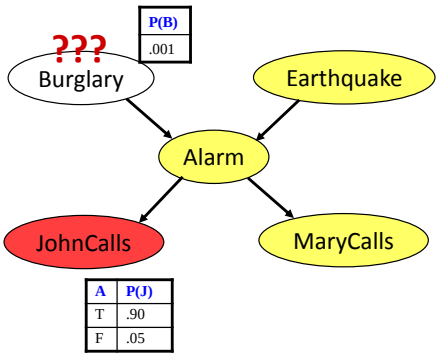
John calls 90% of the time there is an Alarm and the Alarm detects 94% of Burglaries so people generally think it should be fairly high.

However, this ignores the prior probability of John calling.

Slide credit: Ray Mooney

Bayes Net Inference

- Example: Given that John calls, what is the probability that there is a Burglary?



John also calls 5% of the time when there is no Alarm. So over 1,000 days we expect 1 Burglary and John will probably call. However, he will also call with a false report 50 times on average. So the call is about 50 times more likely a false report: P(Burglary | JohnCalls) ≈ 0.02

Slide credit: Ray Mooney

General DAGs

- If the graph is a DAG (directed acyclic graph), we can order nodes such that parents come before children. This is called a topological ordering.
- **Ordered Markov property** is the assumption that a node only depends on its immediate parents, not on all predecessors in the ordering, i.e.,

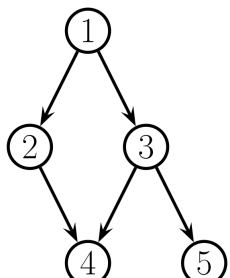
$$x_s \perp \mathbf{x}_{\mathrm{pred}(s) \setminus \mathrm{pa}(s)} | \mathbf{x}_{\mathrm{pa}(s)}$$

  where $\mathrm{pa}(s)$ are the parents of node s, and $\mathrm{pred}(s)$ are the predecessors of node s in the ordering. This is a natural generalization of the first-order Markov property from chains to general DAGs.
- Each node has a CPD $p(x_t|\mathbf{x}_{\mathrm{pa}(t)})$

$$p(\mathbf{x}_{1:V}|G) = \prod_{t=1}^{V} p(x_t|\mathbf{x}_{\mathrm{pa}(t)})$$

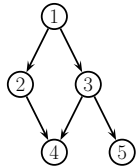
Board work: Bayes Net Example

(3 min) Write the joint probability distribution for the following Bayesian Network. Use $x_1, \ldots, x_5$ as the random variable names.



(3 min) Discuss with a partner

Example



$$
\begin{aligned}
p(\mathbf{x}_{1:5}) &= p(x_1)p(x_2|x_1)p(x_3|x_1,\cancel{x_2})p(x_4|\cancel{x_1},x_2,x_3)p(x_5|\cancel{x_1},\cancel{x_2},x_3,\cancel{x_4}) \\
&= p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2,x_3)p(x_5|x_3)
\end{aligned}
$$

Bayesian Network Examples

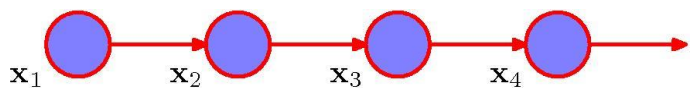
Markov Chains

In general:

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{n=1}^{N} p(\mathbf{x}_n | \mathbf{x}_1, \ldots, \mathbf{x}_{n-1})$$

First-order Markov chain:

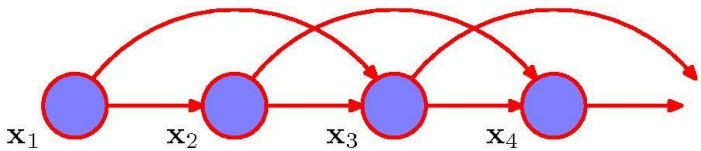
$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{x}_n | \mathbf{x}_{n-1})$$



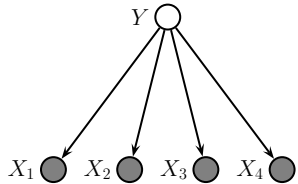
Markov Chains:

Second-order Markov chain:

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1) \prod_{n=3}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{x}_{n-2})$$

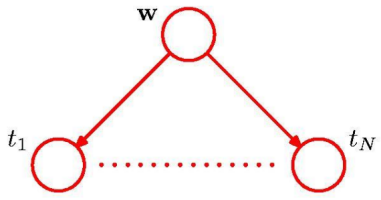


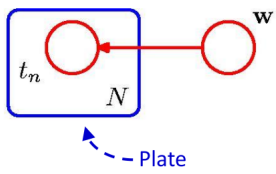
Naive Bayes classifiers



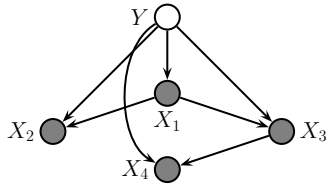
$$p(y, \mathbf{x}) = p(y) \prod_{j=1}^{D} p(x_j|y)$$

Plate notation





Tree augmented Naive Bayes classifiers



$$p(y, \mathbf{x}) = p(y) \prod_{j=1}^{D} p(x_j | x_{\mathrm{pa}(j)}, y)$$

Parameter Estimation in Bayesian Networks

Parameter estimation: Bayes Nets

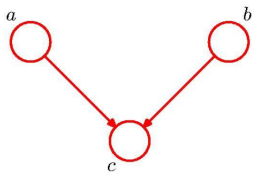
For Bayes Nets with discrete RVs, CPTs are usually a collection of categorical distributions.

Categorical distribution: k possible outcomes, each with probability $\theta_1, \ldots, \theta_k$, where $\theta_1 + \ldots + \theta_k = 1$

For a maximum-likelihood estimate (MLE), just do a relative-frequency estimate given the various values for the parent RVs. This is like in the heads and tails example, $\theta = k/n$, except now we condition on the parent RVs and have k outcomes.

Board work: Parameter Estimation in a Bayes Net

(5 min) For the following BN and data on the next slide, estimate the parameters for the BN using MLE



(3 min) Discuss with a partner

Board work: Parameter Estimation in a Bayes Net

(5 min) Estimate the parameters for the BN using MLE

A	B	C
F	F	F
F	F	T
T	F	F
T	T	T

(3 min) Discuss with a partner

Maximum Likelihood Estimation

- For N-gram language models
  - $p(w_i|w_{i-1}, \dots, w_{i-n+1}) = \frac{c(w_i, w_{i-1}, \dots, w_{i-n+1})}{c(w_{i-1}, \dots, w_{i-n+1})}$

Parameter estimation: Bayes Nets

- This technique works for learning CPTs in Bayes nets if all R.V.s are discrete and observed (fully observed)
- If there are **latent variables** (hidden variables), need to use **Expectation-Maximization** (EM, more on this next week)

Sampling from a Bayesian Network

Drawing samples from a Bayes net: **Ancestral sampling**

To draw samples from the joint distribution:

- Order R.V.s so each node has a higher number than its parent
- Draw samples from the CPT of each R.V. in this order **This ensures parents will always be sampled first**

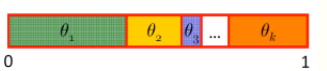
(The reason this procedure works is that each R.V. will be distributed according to the CPT you sample from, and thus the entire sample will be a sample from the joint probability.)

To draw samples from a marginal distribution of a subset of R.V.s, sample the whole graph, and return only the R.V.s of interest.

A model which models the joint distribution (and thus you can generate samples from) is called a generative model.

Sampling from a categorical distribution

- For discrete R.V.s, the CPTs are usually a collection of categorical distributions
- **Categorical distribution**: k possible outcomes, each with probability $\theta_1, \ldots, \theta_k$, where $\theta_1 + \ldots + \theta_k = 1$
- (special case: $k = 2$ **Bernoulli distribution**)
- Subdivide $[0, 1]$ into k regions with region i having size θ_i.



- To sample: sample uniformly from $[0, 1]$ and return the value for the region in which the sample falls.
- **Note: if k is large, there are faster methods**

Multiple draws: Multinomial distribution

If you have n draws from categorical distribution, the distribution of counts follow a **multinomial distribution**.

$$f(x_1, \ldots, x_k; n, p_1, \ldots, p_k) = \Pr(X_1 = x_1 \text{ and } \ldots \text{ and } X_k = x_k)$$
$$= \begin{cases} \dfrac{n!}{x_1! \cdots x_k!} p_1^{x_1} \times \cdots \times p_k^{x_k}, & \text{when } \sum_{i=1}^{k} x_i = n \\ 0 & \text{otherwise,} \end{cases}$$

Special case: $k = 2$ **Binomial distribution**

Multinomial coefficients:

$$\binom{n}{k_1, k_2, \ldots, k_r} = \frac{n!}{k_1! k_2! \cdots k_r!}$$

Bayesian Learning

Frequentist vs Bayesian statistical thinking

- **Frequentist philosophy**: parameters are fixed, not a R.V.. May not know the true value, but it's fixed, given to us from nature. Makes no sense to ask about probabilities of parameters.
- **Bayesian philosophy**: parameters are an unknown R.V. Can ask about probabilities of parameters (called **degree of belief**). Parameters can have distributions of their own.

Frequentist vs Bayesian statistical thinking

- **Frequentist philosophy**: parameters are fixed, not a R.V.. May not know the true value, but it's fixed, given to us from nature. Makes no sense to ask about probabilities of parameters.
- **Bayesian philosophy**: parameters are an unknown R.V. Can ask about probabilities of parameters (called **degree of belief**). Parameters can have distributions of their own.
- In machine learning, and in NLP, the distinction doesn't matter so much. We don't care about the parameter values, only predictions on new data. We'll use both ways of thinking, whichever gives better performance.

Bayesian Learning

In Bayesian learning, we'd like to estimate the probability of the parameters θ, given the data $\mathcal{D}$.

Using Bayes' rule:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int_\theta p(\mathcal{D})|\theta)p(\theta)} \qquad \text{Bayes' rule}$$

$$\propto p(\mathcal{D}|\theta)p(\theta) \qquad \text{Constant over } \theta$$

Bayesian Learning: Terminology

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$$

- $p(\theta)$ is called the **prior**
- $p(\mathcal{D}|\theta)$ is called the **likelihood**
- $p(\theta|\mathcal{D})$ is called the **posterior**

If we estimate θ by maximizing $p(\theta|\mathcal{D})$, this is called **maximum a posteriori (MAP)** estimation.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\theta|\mathcal{D})$$

MLE vs MAP estimation

Maximum likelihood estimate (MLE):

$$\hat{\theta} = \operatorname*{argmax}_{\theta} p(\mathcal{D}|\theta)$$

Maximum a posteriori (MAP) estimate:

$$\begin{aligned}\hat{\theta} &= \operatorname*{argmax}_{\theta} p(\theta|\mathcal{D}) \\ &= \operatorname*{argmax}_{\theta} p(\mathcal{D}|\theta)p(\theta)\end{aligned}$$

Frequentist vs Bayesian Learning

In Bayesian Learning, θ is a random variable that we marginalize over.
Frequentist prediction:

$$\hat{y} = \operatorname*{argmax}_y p(y|x;\hat{\theta})$$

Bayesian prediction:

$$\begin{aligned}\hat{y} &= \operatorname*{argmax}_y \int_\theta p(y|x;\theta)p(\theta|\mathcal{D}) \\ &= \operatorname*{argmax}_y \int_\theta p(y|x;\theta)p(\mathcal{D}|\theta)p(\theta)\end{aligned}$$

Advanced Stuff:

Add-α Smoothing as Bayesian Learning

MLE vs MAP estimation

Maximum likelihood estimate (MLE):

$$\hat{\theta} = \operatorname*{argmax}_{\theta} p(\mathcal{D}|\theta)$$

Maximum a posteriori (MAP) estimate:

$$\begin{aligned}\hat{\theta} &= \operatorname*{argmax}_{\theta} p(\theta|\mathcal{D}) \\ &= \operatorname*{argmax}_{\theta} p(\mathcal{D}|\theta)p(\theta)\end{aligned}$$

Categorical distribution

Categorical distribution: k possible outcomes, each with probability $\theta_1, \ldots, \theta_k$, where $\theta_1 + \ldots + \theta_k = 1$.

The distribution of counts from a categorical distribution follows a multinomial distribution:

$$p(\mathcal{D}|\theta) = \frac{n!}{count(x=i)! \ldots count(x=k)!} \prod_{i=1}^{k} \theta_i^{count(x=i)}$$
$$\propto \prod_{i=1}^{k} \theta_i^{count(x=i)}$$

The Prior $p(\theta)$

We need a distribution over the parameters θ. (A distribution over distributions).

The Prior $p(\theta)$

- Notice that $p(\mathcal{D}|\theta) = \prod_{i=1}^{k} \theta_i^{\beta_i}$ for some vector β.
- If we choose the prior $p(\theta) \propto \prod_{i=1}^{k} \theta_i^{\alpha_i}$ for some vector α, then

$$\begin{aligned} p(\theta|\mathcal{D}) &\propto p(\mathcal{D}|\theta)p(\theta) \\ &\propto (\prod_{i=1}^{k} \theta_i^{\beta_i})(\prod_{i=1}^{k} \theta_i^{\alpha_i}) \\ &\propto \prod_{i=1}^{k} \theta_i^{\beta_i+\alpha_i} = \prod_{i=1}^{k} \theta_i^{\gamma_i} \end{aligned}$$

for $\gamma = \alpha + \beta$.

Posterior is of the same form as the prior and the likelihood!
Priors for which this occurs are called **conjugate priors.**

Dirichlet distribution

$$p(\theta) \propto \prod_{i=1}^{k} \theta_i^{\alpha_i} \implies p(\theta) = \frac{1}{B(\alpha)} \prod_{i=1}^{k} \theta_i^{\alpha_i}$$

where $B(\alpha)$ is a normalizing constant so it integrates to 1 over all θ such that $\sum_i \theta_i = 1$.

This distribution is called the **Dirichlet distribution**

$$\text{Dirichlet}(\alpha + 1)$$

The Dirichlet distribution is the conjugate prior for the categorical distribution.

Notation: Parameters

If we have a distribution, say $p(x, y)$ with parameters θ, statisticians will often write

$$p_\theta(x, y) \quad \text{parameters as a subscript}$$
$$\text{or}$$
$$p(x, y; \theta) \quad \text{parameters after a semicolon}$$

(Remember $\sum_{x,y} p(x, y; \theta) = 1$ for any values of θ (or integral if continuous). **That's why they put a semicolon - to show that you don't include it in the sum that equals 1**)

For our Dirichlet distribution prior $p(\theta)$, θ is a R.V. and α are hyperparameters, so we write

$$p(\theta; \alpha) \propto \prod_{i=1}^{k} \theta_i^{\alpha_i}$$

Sampling Notation

If a random variable X follows a distribution $p(x)$, we sometimes write

$$X \sim p$$

If the distribution p has parameters θ, we write

$$X \sim p(\theta)$$

This is called **sampling notation**.

Bayesian interpretation of add-α smoothing

$$\theta \sim \text{Dirichlet}(\alpha + 1)$$
$$x \sim \text{Categorical}(\theta)$$

Assuming this model, what is the most probable value of θ, having observed the training data $\mathcal{D} = \{x_1, \ldots x_N\}$?

$$\hat{\theta} = \underset{\theta}{\text{argmax}}\, p(\mathcal{D}|\theta)p(\theta) = \underset{\theta}{\text{argmax}} \prod_{i=1}^{N} p(x_i|\theta)p(\theta)$$

(You prove on your next homework):

$$\hat{\theta}_i = \frac{count(x = i) + \alpha_i}{N + \sum_j \alpha_j}$$ **This is additive smoothing!**

Smoothing in NB

- One solution: add a little probability mass to every element.

maximum likelihood estimate

$$P(x_i \mid y) = \frac{n_{i,y}}{n_y}$$

$n_{i,y}$ = count of word i in class y
n_y = number of words in y
V = size of vocabulary

smoothed estimates

$$P(x_i \mid y) = \frac{n_{i,y} + \alpha}{n_y + V\alpha}$$

same α for all x_i

$$P(x_i \mid y) = \frac{n_{i,y} + \alpha_i}{n_y + \sum_{j=1}^{V} \alpha_j}$$

possibly different α for each x_i