Probabilistic Methods (PM - 330725)

TOPIC 3: Probabilistic Models

Lecture 7

May 2025

Roadmap

Learning Outcomes

By the end of this topic, you will be able to:

- update probabilities and make predictions using Bayesian methods and prior distributions.
- model and analyze processes where future states depend on current states, using Markov chains and their properties.
- construct and apply Bayesian networks and Hidden Markov Models for complex system representation.
- use probabilistic models to handle and predict outcomes in sequential data and time series.

Recap: Bayesian Networks and Graphical Models

- **Bayes nets** in a nutshell:
  - Structured probability (density/mass) functions $f_X(x;\theta)$
  - Provides a graph-based language/grammar to express conditional independence
  - Allows formalizing the problem of inferring a subset of components of $\mathbf{X}$ from another subset thereof
  - Allows formalizing the problem of learning θ from observed i.i.d. realizations of $\mathbf{X}$: $x_1, \ldots, x_n$
- Bayes nets are one type of graphical models, based on directed graphs.
  Other types (more on them later):
  - Markov random fields (MRF), based on undirected graphs
  - Factor graphs

Recap: Bayesian Networks

- For $\mathbf{X} \in R^n$, the pdf/pmf $p(\mathbf{x})$ can be factored by Bayes law:

$$p(\mathbf{x}) = p(x_n \mid x_{n-1}, \ldots, x_1) \cdots p(x_2 \mid x_1) p(x_1)$$

- In general, this is not more compact than $p(\mathbf{x})$
  - Example: if $x_i \in \{1, \ldots, K\}$, a general $p(\mathbf{x})$ has $K^n - 1 \simeq K^n$ parameters
  - But $p(x_n \mid x_{n-1}, \ldots, x_1)$ alone has $(K-1)K^{n-1} \simeq K^n$ parameters!
- ...unless, there are some conditional independencies
  - Example: $\mathbf{X}$ is a Markov chain:

$$p(x_i \mid x_{i-1}, \ldots, x_1) = p(x_i \mid x_{i-1})$$

    - in this case,

$$p(\mathbf{x}) = p(x_n \mid x_{n-1}) p(x_{n-1} \mid x_{n-2}) \cdots p(x_2 \mid x_1) p(x_1)$$

      has

$$nK(K-1) + K \simeq nK^2 \text{ parameters}$$

  - linear in n, rather than exponential!

Recap: Goal of Inference in Directed Graphical Models

- Visible and hidden variables: $\mathbf{x} = (\mathbf{x}_v, \mathbf{x}_h)$
- Joint pdf/pmf $p(\mathbf{x}_v, \mathbf{x}_h \mid \theta)$
- Inferring the hidden variables from the visible ones:

$$p(\mathbf{x}_h \mid \mathbf{x}_v, \theta) = \frac{p(\mathbf{x}_h, \mathbf{x}_v \mid \theta)}{p(\mathbf{x}_v \mid \theta)} = \frac{p(\mathbf{x}_h, \mathbf{x}_v \mid \theta)}{\sum_{\mathbf{x}'_h} p(\mathbf{x}'_h, \mathbf{x}_v \mid \theta)}$$

- ...with integral instead of sum, if $\mathbf{x}_h$ has real components
- Sometimes, only a subset $\mathbf{x}_q$ of $\mathbf{x}_h$ is of interest, $\mathbf{x}_h = (\mathbf{x}_q, \mathbf{x}_n)$:

$$p(\mathbf{x}_q \mid \mathbf{x}_v, \theta) = \sum_{\mathbf{x}_n} p(\mathbf{x}_q, \mathbf{x}_n \mid \mathbf{x}_v, \theta)$$

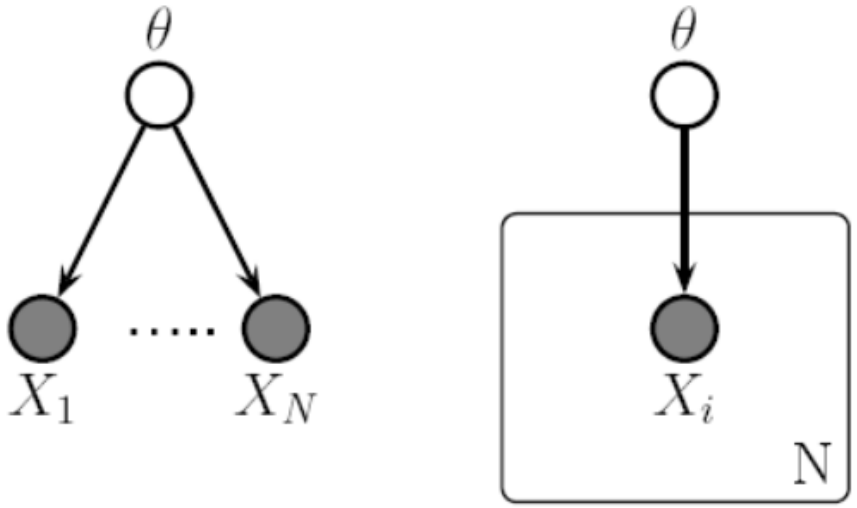
- ... $\mathbf{x}_n$ are sometimes called nuisance variables

Recap: Learning in Directed Graphical Models

- Observe samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ of N i.i.d. copies of $X \sim p(\mathbf{x} \mid \theta)$
- MAP estimate of θ:

$$\hat{\theta} = \arg\max_{\theta} \left(\log p(\theta) + \sum_{i=1}^{N} \log p(\mathbf{x}_i \mid \theta) \right)$$

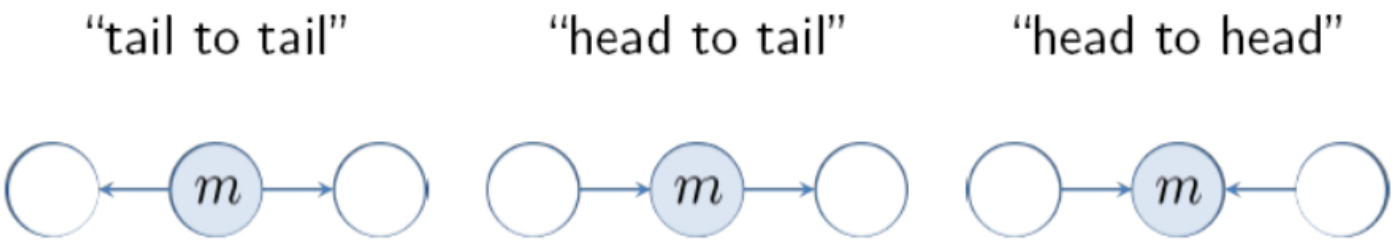
- Plate notation: for a collection of i.i.d. copies of some variable



- If there are hidden variables, $\mathbf{x}$ should be understood as denoting only the visible ones

Recap: Conditional Independence Properties

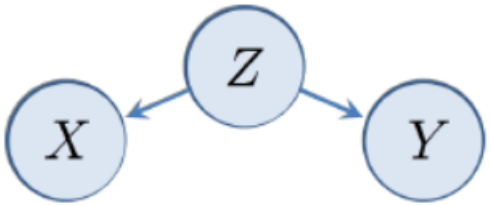
- What type conditional independence (CI) statements are expressed by some graph G?
- In an path through some node m, there are three possible orientation structures:



- Before proceeding to the general statement, we next exemplify which type of CI corresponds to each of these structures.

Recap: Conditional Independence Structures

- **Tail to tail**



$$X \perp Y \mid Z$$

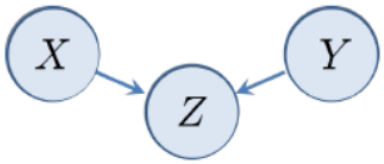
$$\begin{aligned} p(x, y \mid z) &= \frac{p(x, y, z)}{p(z)} \\ &= \frac{p(x \mid z)p(y \mid z)p(z)}{p(z)} \\ &= p(x \mid z)p(y \mid z) \end{aligned}$$

- **Head to tail**

$$X \perp Y \mid Z$$

$$\begin{aligned} p(x, y \mid z) &= \frac{p(y \mid z)p(z \mid x)p(x)}{p(z)} \\ &= p(y \mid z)p(x \mid z) \end{aligned}$$

Recap: example on the noisy fuel gauge



$$p(x, y \mid z) \neq p(x \mid z)\, p(y \mid z)$$

$$p(x, y) = \sum_z p(z \mid x, y)\, p(x, y) = p(x)\, p(y)$$

$X \perp Y \mid \varnothing$

- Binary variables: X = battery OK, Y = full tank, Z = fuel gauge on
- $\mathbb{P}(X = 1) = \mathbb{P}(Y = 1) = 0.9$

x	y	$\mathbb{P}(Z = 1 \mid x, y)$
1	1	0.8
1	0	0.2
0	1	0.2
0	0	0.1

- Gauge is off ($Z = 0$); is the tank empty?
  $\mathbb{P}(Y = 0 \mid Z = 0) = 0.257$
- ...but, the battery is also dead,
  $\mathbb{P}(Y = 0 \mid Z = 0, X = 0) = 0.11$
- Dead battery explains away the empty tank

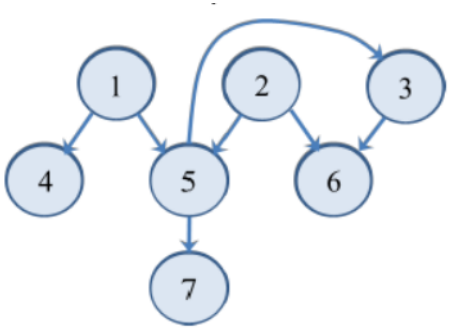
Recap: D-Separation

- Graph $G = (\mathcal{V}, \mathcal{E})$ and three disjoint subsets of $\mathcal{V}$: A, B, and C
- An (undirected) path from a node in A to a node in B is blocked by C if it includes a node m such that:
  - $m \in C$ and the arrows meet "head to tail" or "tail to tail"
  - the arrows meet "head to head", $m \notin C$, and $\text{desc}(m) \cap C = \text{Å}$
- C D-separates A from B and

$$\mathbf{x}_A \perp_G \mathbf{x}_B \mid \mathbf{x}_C$$

if any path from a node in A to a node in B is blocked by C

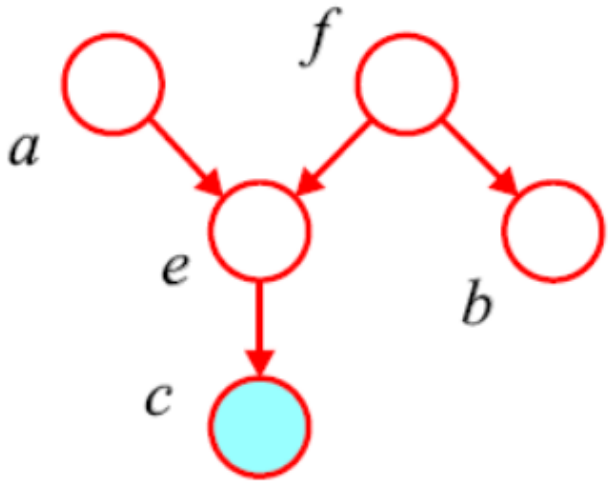
- Example:



- $x_4 \perp_G x_5 \mid x_1$ (tail to tail)
- $x_1 \perp_G x_2 \mid x_4$ (head to head in 5, $5 \notin \{4\}$)
- $x_5 \perp_G x_6 \mid \mathbf{x}_{\{2,3\}}$ (tail to tail in 2 and head to tail in 3)
- $x_1 \perp_G x_2 \mid x_7$ is false (head to head in 5, but $7 \in \text{desc}(5)$)

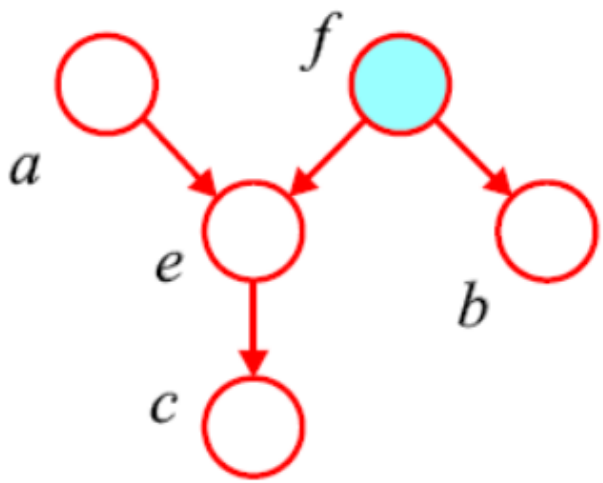
Recap: D-Separation

- wrapping up, Conditional independence if, and only if, all possible paths are blocked.
- more examples:



(i)

$a \not\perp\!\!\!\perp b \mid c$



(ii)

$a \perp\!\!\!\perp b \mid f$

Learning probabilistic models from data

Model:

- a random variable X
- a probability density/mass function f associated with X
- a parameter θ of theoretical distribution f

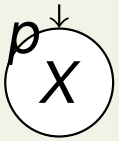
Learning:

- assume f is known except for parameter θ, and denote this fact as $f(x;\theta)$ or $f(x \mid \theta)$
- for a sample $X_1, \ldots, X_n$ drawn from $f(x \mid \theta)$, regard the joint density/mass $f(x_1, \ldots, x_n \mid \theta)$ as a function of parameter θ, called the likelihood function:

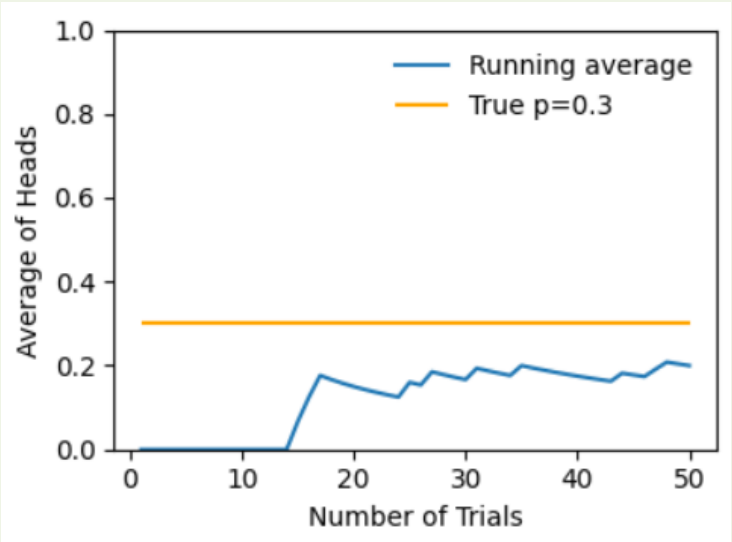
$$\ell(\theta \mid x_1, \ldots, x_n) \stackrel{\text{def}}{=} f(x_1, \ldots, x_n \mid \theta) = \prod_{i=1}^{n} f(x_i \mid \theta).$$

Learning parameters from data: frequentist approach

- Parameter θ has a fixed but unknown value



p is a probability:
relative frequency "in the long run"



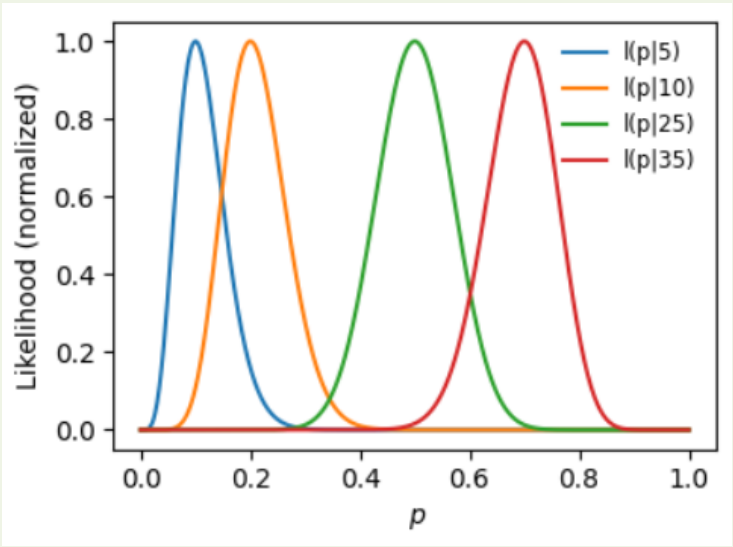
$$HHHTTH \ldots TH$$

$$\hat{p} = \frac{15}{50}$$

Assume a sample $X \sim \mathrm{B}(50, p)$.

The likelihood function is

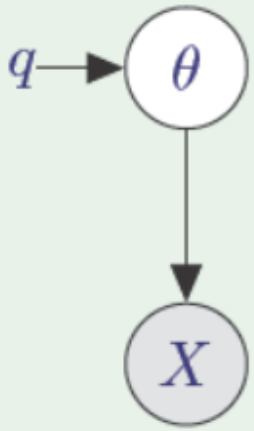
$$\ell(p \mid x) = \binom{50}{x} p^x (1-p)^{50-x}.$$



Learning parameters from data: Bayesian approach

- Parameters are modelled as random variables and information about them can be included prior to observing data
- By Bayes' rule, the prior information is combined with the likelihood, yielding a posterior distribution
- As such, inferences about the parameter allow for its updating

Bayesian networks for Bayesian learning



- Random variables (and parameters) inside circles
- Grey if observable; white if hidden
- Fixed quantities without circle

Distributions in a Bayesian model

- The prior distribution of θ, $\boxed{\pi(\theta)}$
- The joint distribution of (X, θ), $\boxed{\psi(x, \theta) = f(x \mid \theta)\, \pi(\theta)}$
- The posterior distribution of θ given x,

$$\boxed{\pi(\theta \mid x) = \frac{f(x \mid \theta)\, \pi(\theta)}{\int_\theta f(x \mid \theta)\, \pi(\theta)\, d\theta}}$$

- The prior predictive distribution of X,

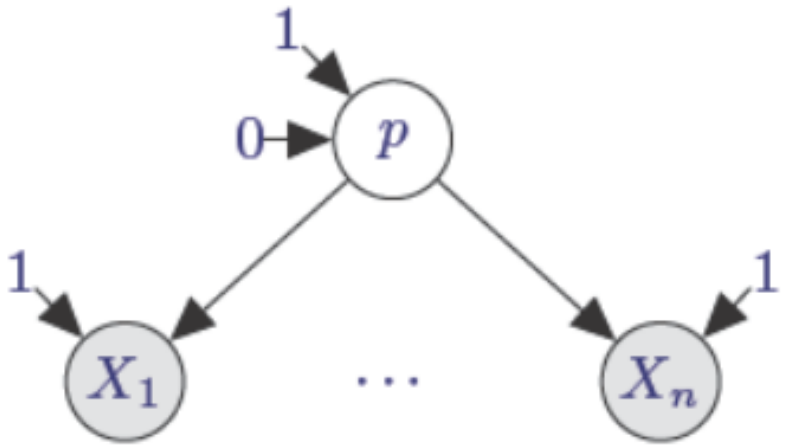
$$\boxed{m(x) = \int_\theta f(x \mid \theta)\, \pi(\theta)\, d\theta}$$

- The (posterior) predictive distribution given $x = \{x_1, \ldots, x_n\}$:

$$\boxed{f(x_{n+1} \mid x) = \int_\theta f(x_{n+1} \mid \theta, x)\, \pi(\theta \mid x)\, d\theta = \int_\theta f(x_{n+1} \mid \theta)\, \pi(\theta \mid x)\, d\theta}$$

Learning from data: Example of Bayesian approach

- Assume a sample $X_1, X_2, \ldots, X_n \sim \mathrm{B}(1, p)$ and $p \sim \mathcal{U}(0, 1)$



- Then the likelihood and the prior are,

$$f(x_1, \ldots, x_n \mid p) = p^{\sum_i x_i} (1-p)^{n-\sum_i x_i}, \quad \text{with } x_i \in \{0, 1\},\ p \in (0, 1),$$

$$\pi(p) = \frac{1}{1-0} = 1, \quad p \in (0, 1).$$

Learning from data: Example of Bayesian approach

Assume a sample

$$X_1, X_2, \ldots, X_n \sim \mathrm{B}(1, p) \quad \text{and} \quad p \sim \mathcal{U}(0, 1).$$

- Then the likelihood and the prior are

$$f(x_1, \ldots, x_n \mid p) = p^{\sum_i x_i} (1-p)^{n - \sum_i x_i}, \quad \text{with } x_i \in \{0, 1\},\ p \in (0, 1),$$

$$\pi(p) = 1, \quad p \in (0, 1).$$

- The posterior distribution is

$$\pi\left(p \mid x_1, \ldots, x_n\right) = \frac{f(x_1, \ldots, x_n \mid p)\, \pi(p)}{\displaystyle\int_0^1 f(x_1, \ldots, x_n \mid p)\, \pi(p)\, dp}$$

$$= \frac{p^{\sum_i x_i} (1-p)^{n - \sum_i x_i}}{\displaystyle\int_0^1 p^{\sum_i x_i} (1-p)^{n - \sum_i x_i}\, dp}.$$

Learning from data: Example of Bayesian approach

Pattern matching: the Beta distribution $\mathrm{Be}(\alpha, \beta)$

$$f(p) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\,\Gamma(\beta)}\; p^{\alpha-1}\,(1-p)^{\beta-1}$$

Now consider the integral:

$$\int_0^1 p^{\sum x_i}(1-p)^{n-\sum x_i}\,dp$$

$$= \int_0^1 \frac{\Gamma(\sum x_i+1)\Gamma(n-\sum x_i+1)}{\Gamma(n+2)} \cdot \frac{\Gamma(n+2)}{\Gamma(\sum x_i+1)\Gamma(n-\sum x_i+1)} \cdot p^{\sum x_i}(1-p)^{n-\sum x_i}\,dp$$

$$= \frac{\Gamma(\sum x_i+1)\Gamma(n-\sum x_i+1)}{\Gamma(n+2)} \cdot \int_0^1 \frac{\Gamma(n+2)}{\Gamma(\sum x_i+1)\Gamma(n-\sum x_i+1)} \cdot p^{\sum x_i}(1-p)^{n-\sum x_i}\,dp$$

$$= \frac{\Gamma(\sum x_i+1)\Gamma(n-\sum x_i+1)}{\Gamma(n+2)} \cdot 1$$

Learning from data: Example of Bayesian approach

Assume a sample

$$X_1, X_2, \ldots, X_n \sim \mathrm{B}(1, p) \quad \text{and} \quad p \sim \mathcal{U}(0,1) = \mathrm{Be}(1,1).$$

- Then the **likelihood** and the **prior** are

$$f(x_1, \ldots, x_n \mid p) = p^{\sum_i x_i} (1-p)^{n - \sum_i x_i}, \quad x_i \in \{0,1\},\ p \in (0,1),$$

$$\pi(p) = 1, \quad p \in (0,1).$$

- The **posterior** distribution is

$$\pi(p \mid x_1, \ldots, x_n) = \frac{f(x_1, \ldots, x_n \mid p)\, \pi(p)}{\int_0^1 f(x_1, \ldots, x_n \mid p)\, \pi(p)\, dp} = \frac{p^{\sum_i x_i} (1-p)^{n - \sum_i x_i}}{\int_0^1 p^{\sum_i x_i} (1-p)^{n - \sum_i x_i}\, dp}$$

$$= \frac{\Gamma(n+2)}{\Gamma\left(\sum_i x_i + 1\right) \Gamma\left(n - \sum_i x_i + 1\right)} p^{\sum_i x_i} (1-p)^{n - \sum_i x_i} = \mathrm{Be}\left(\sum_i x_i + 1,\ n - \sum_i x_i + 1\right).$$

Very easy to compute for some models

Common Conjugate Priors

Likelihood	**Prior**	**Posterior**
$\mathrm{B}(1,\theta)$	$\mathrm{Be}(\alpha,\beta)$	$\mathrm{Be}\big(\alpha+\sum_i x_i,\ \beta+n-\sum_i x_i\big)$
$\mathrm{NB}(r,\theta)$	$\mathrm{Be}(\alpha,\beta)$	$\mathrm{Be}\big(\alpha+rn,\ \beta-rn+\sum_i x_i\big)$
$\mathrm{G}(\theta)$	$\mathrm{Be}(\alpha,\beta)$	$\mathrm{Be}\big(\alpha+n,\ \beta+\sum_i x_i\big)$
$\mathrm{MN}\big(n,\theta_1,\ldots,\theta_k\big)$	$\mathrm{Dir}(\alpha_1,\ldots,\alpha_k)$	$\mathrm{Dir}(\alpha_1+x_1,\ldots,\alpha_k+x_k)$
$\mathrm{P}(\theta)$	$\Gamma(\alpha,\beta)$	$\Gamma\big(\alpha+\sum_i x_i,\ \beta+n\big)$
$\mathrm{Exp}(\theta)$	$\Gamma(\alpha,\beta)$	$\Gamma\big(\alpha+n,\ \beta+\sum_i x_i\big)$
$\mathcal{N}(\mu,\tau)$	$\mathcal{N}(\mu_0,\tau_0)$	$\mathcal{N}\Big(\frac{\tau_0\mu_0+n\tau\,\bar{x}}{\tau_0+n\tau},\ \tau_0+n\tau\Big)$
$\mathcal{N}(\mu,\tau)$	$\Gamma(\alpha_0,\beta_0)$	$\Gamma\Big(\alpha_0+\frac{n}{2},\ \beta_0+\frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^2\Big)$

Learning from data: Bayesian approach

- The method above is known as *fully Bayesian* approach
- Sometimes, we don't need to compute the denominator of the posterior distribution, in which case θ can be estimated as

$$\hat{\theta} = \arg\max_{\theta} f(x_1, \ldots, x_n, \theta) = \arg\max_{\theta} \left[f(x_1, \ldots, x_n \mid \theta)\, \pi(\theta) \right]$$

$$= \arg\max_{\theta} \left\{ \log f(x_1, \ldots, x_n \mid \theta) + \log \pi(\theta) \right\}$$

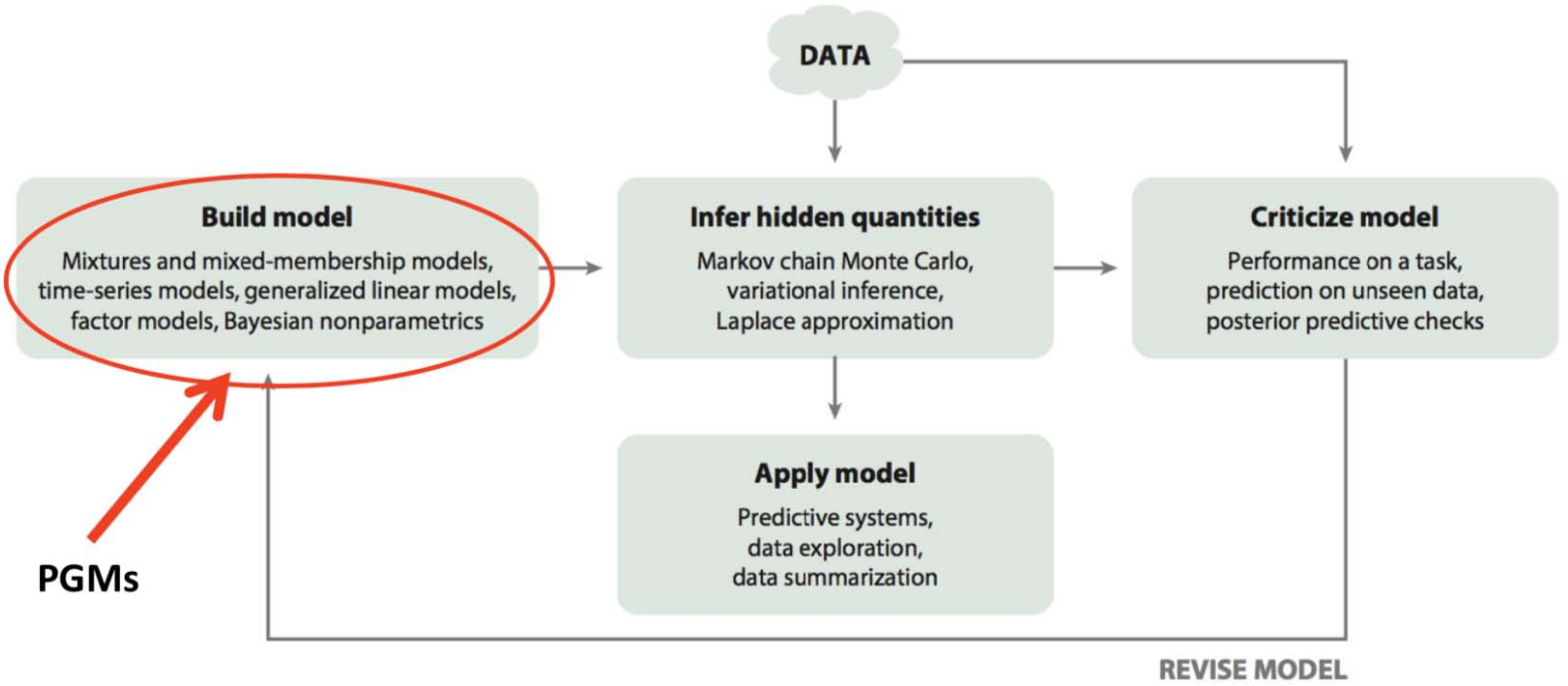
  known as the MAP (Maximum A Posteriori) estimator

- Note that we could also choose

$$\hat{\theta} = \arg\max_{\theta} \log f(x_1, \ldots, x_n \mid \theta)$$

  which is actually the MLE (Maximum Likelihood Estimator)

Learning from data: Bayesian approach

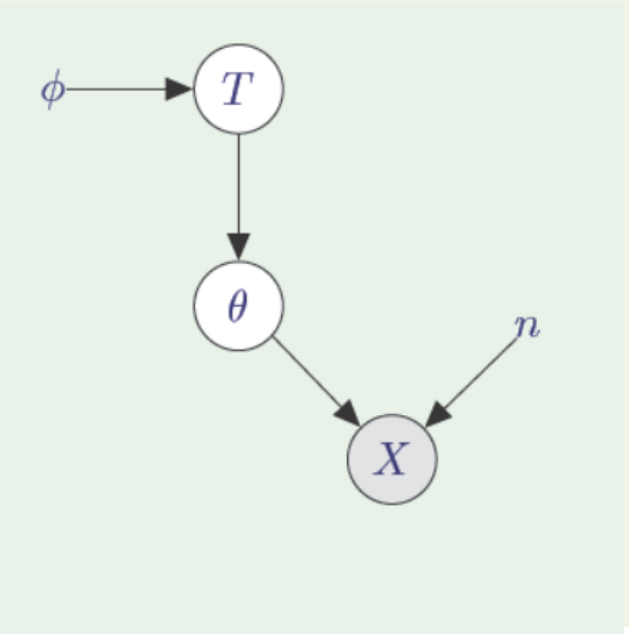


Blei, David M. "Build, compute, critique, repeat: Data analysis with latent variable models." *Annual Review of Statistics and Its Application* 1 (2014): 203-232.

Some simple examples

Tossing a biased(?) coin

- X : result of n coin tosses with some $p(\text{heads})$
- random variables?
- fixed quantities?
- hidden variables?
- coin is possibly biased towards tails



Learning from data: Bayesian approach

Two coins

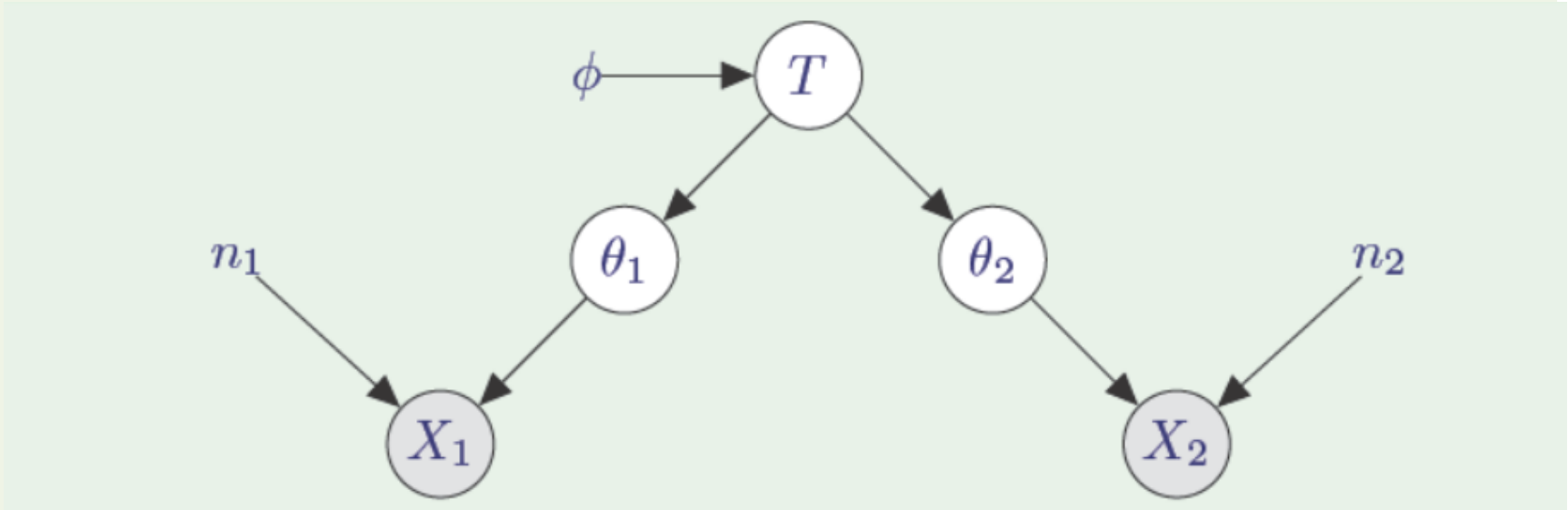


Plate notation

The idea is to avoid repeated substructures

Example: independent data points

- Assume the elements in a sample $X_1, X_2, \ldots, X_N$ are independent if the parameter θ is known

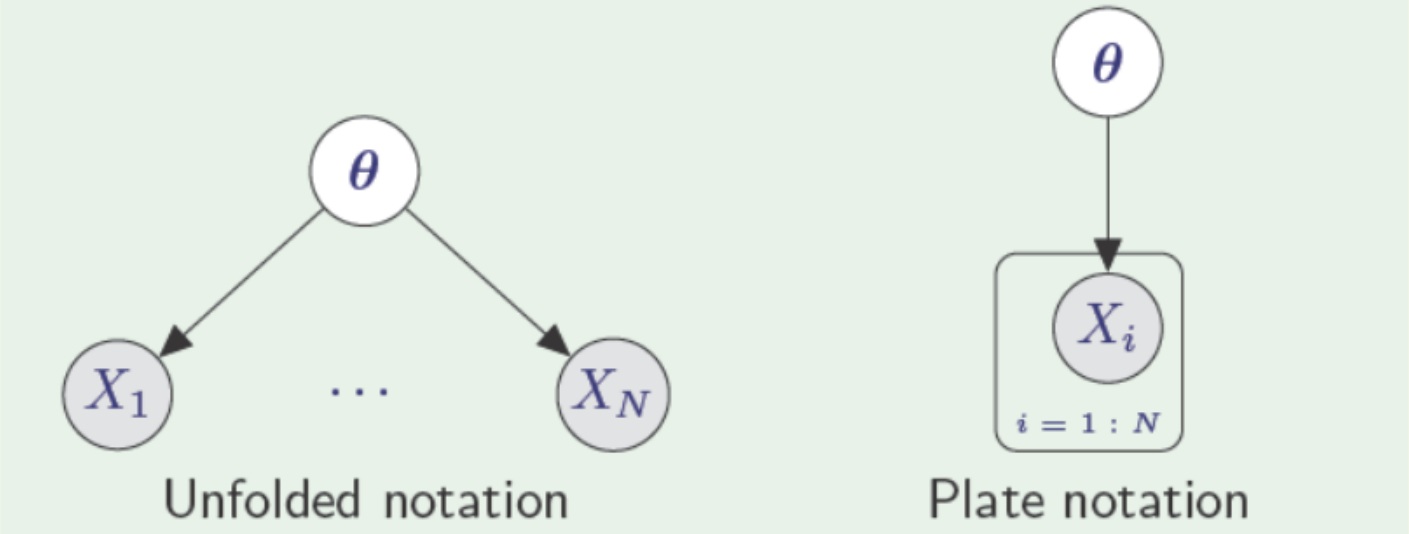


Plate notation: linear regression revisited

Example: linear regression (fully probabilistic)

- $Y_i \mid \{w, x_i\} = w^\top x_i + \epsilon_i$ with $x_i = [1,\ x_i]^\top$
- $\epsilon_i \sim \mathcal{N}(0,\ 1/\gamma)$ with precision parameter γ known
- $w \sim \mathcal{N}(\mu_0 = 0,\ \Sigma_0 = I_{2\times 2})$

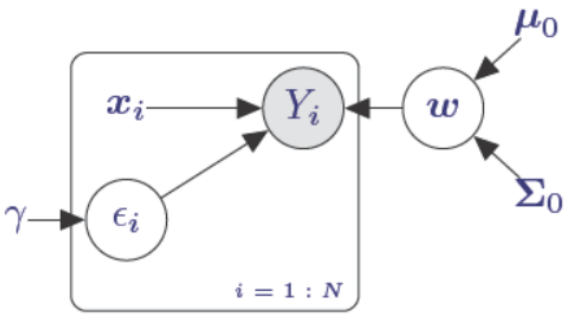
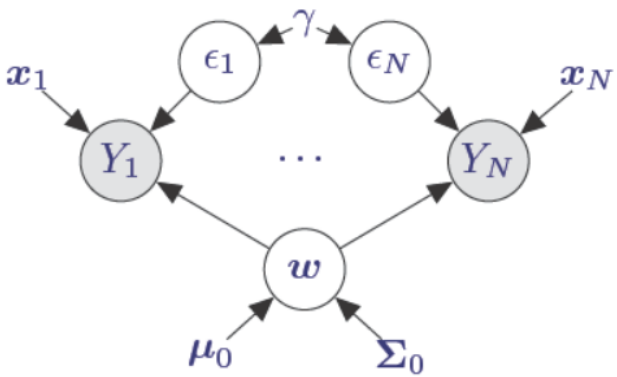


Plate notation



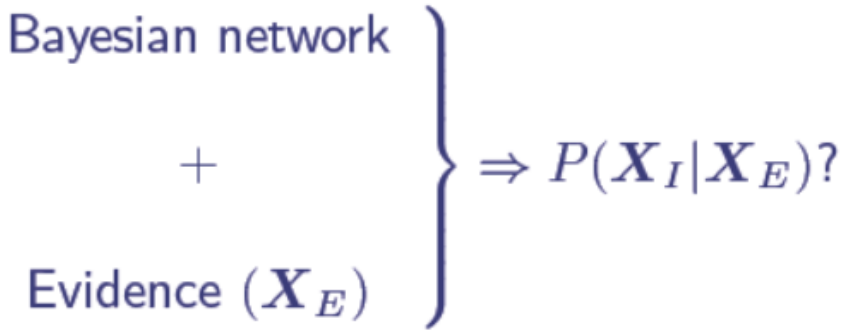
Unfolded notation

- Underlying model:

$$y_i = w_0 + w_1\, x_i + \epsilon_i$$

Inference in Bayesian networks so far...

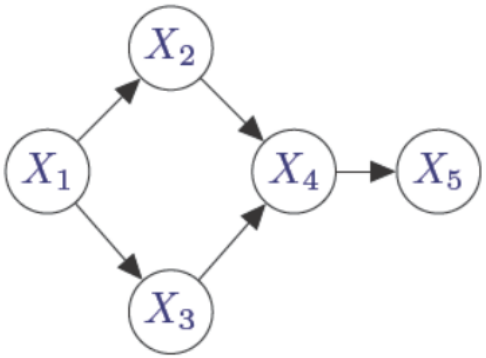
Assume a Bayesian network over variables $\mathbf{X} = \{X_1, \ldots, X_n\}$



Inference methods

- **Exact**
  - Brute force: compute $P(X, X_E)$ and marginalize out $X \setminus X_I$
  - Take advantage of the network structure
- **Approximate**
  - Sampling
  - Deterministic

Exact inference: Variable elimination



- We are interested in X_5
- All variables are discrete
- $E = \emptyset$

$$
\begin{aligned}
p(x_5) &= \sum_{x_1,\ldots,x_4} p(x_1, x_2, x_3, x_4, x_5) \\
&= \sum_{x_1,\ldots,x_4} p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2, x_3)p(x_5 \mid x_4) \\
&= \sum_{x_2,\ldots,x_4} \sum_{x_1} p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2, x_3)p(x_5 \mid x_4) \\
&= \sum_{x_2,\ldots,x_4} p(x_4 \mid x_2, x_3)p(x_5 \mid x_4) \boxed{\sum_{x_1} p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)} \\
&= \sum_{x_2,\ldots,x_4} p(x_4 \mid x_2, x_3)p(x_5 \mid x_4)h(x_2, x_3)
\end{aligned}
$$

We have reached a similar problem as initially, but with one variable less.

Variable elimination algorithm

Input

- $\mathcal{P}$: Conditional distributions in the network
- $\mathbf{X}$: Variables in the network
- X_I: Target variable
- X_E: Observed variables
- Y: All variables in $\mathbf{X}$ except X_I ($Y = X \setminus \{X_I\}$)

Algorithm

1. Restrict all distributions in $\mathcal{P}$ to the evidence $X_E = x_E$.
2. For each $Y \in Y$:
    1. Let $\mathcal{P}_Y$ be the subset of $\mathcal{P}$ whose factors mention Y.
    2. Compute $q \leftarrow \prod_{p \in \mathcal{P}_Y} p$.
    3. Compute $r \leftarrow \sum_Y q$.
    4. Update $\mathcal{P} \leftarrow (\mathcal{P} \setminus \mathcal{P}_Y) \cup \{r\}$.
3. Form $p(x_I) = \prod_{p \in \mathcal{P}} p$.
4. Normalize p.

Considerations about exact inference

- Product of functions raises complexity
  - Exponentially in the case of discrete variables
- Complexity also depends on the elimination order
- Representation of densities turns out to be relevant
  - Closed-form solutions to product and marginalization are preferable

Bayesian Network Parameter Learning

Once the structure of the model is known, the problem of estimating the parameters of the global distribution can be solved by estimating the parameters of the local distributions, one at a time.

Three common choices are:

- maximum likelihood estimators: just the usual estimators from classic linear models and contingency-table literature.
- Bayesian posterior estimators: posterior estimators, based on conjugate priors to keep computations fast, simple, and in closed form.
- regularised estimators: regularised estimators based either on James–Stein or Bayesian shrinkage results.

Events, Evidence and Queries

A BN represents a working model of the world that a computer can understand; but how does a computer system use it to help and perform its assigned task?

We ask questions, and the computer system performs probabilistic inference to answer them and decide what to do in the process.

Questions that can be asked are called queries and are typically about an event of interest given some evidence. The evidence is the input to the computer system and the event is the output. This is often called belief update: we observe some evidence and we update our beliefs before taking action.

Events, Evidence and Queries

The two most common queries are:

- conditional probability queries ("What is the probability of a particular event after we observe some specific evidence?")
- most probable explanation queries ("What is the most probable value of this subset of variables given that we observe some specific values for one or more other variables?")

In both cases the evidence is hard evidence: we set some variables to particular values. Then the computer system checks how the probabilities of other variables change and provides an answer to the query.

No more manual probability computations...

Exact and Approximate Inference

There are two approaches to answering queries using BNs.

Exact algorithms use the DAG of a BN to schedule and perform repeated applications of Bayes' theorem and the probability axioms on the model's probabilities. In other words, the computer system uses the DAG to perform all the math we would do by hand.

The two best known are:

- variable elimination
- belief updates based on junction trees

PROS: they return exact values for the probabilities of interest. CONS: they do not scale well when BNs have many nodes and many arcs.

Exact and Approximate Inference

Approximate algorithms use the BN as a model of the world in a very literal sense. In the real world, to answer a question scientifically, we would perform an experiment and observe the outcome; approximate algorithms imitate this by generating random observations from the BN—running a simulated experiment that approximates reality.

The two best known are:

- logic sampling
- likelihood weighting

PROS: they scale really well when BNs have many nodes and many arcs.

CONS: they return approximate, estimated values for the probabilities of interest.

Approximate inference

- sampling: Monte Carlo techniques, e.g. importance sampling, MCMC
  - accurate with enough samples
  - sampling can be computationally demanding
- deterministic, e.g. variational approaches
  - uses analytical approximations to the posterior
  - some techniques scale well

Monte Carlo inference algorithms

- A Bayesian network is a representation of a joint probability distribution over $\mathbf{X}$ it describes some population consisting of all possible configurations of $\mathbf{X}$
- If the entire population were available, the inference problem could be solved exactly, basically by counting cases
- **Problem:** Population size can be huge or even infinite
- Monte Carlo methods operate by drawing an artificial sample from the population using some random mechanism
- The sample (much smaller than the population) is used to estimate the distribution of each variable of interest

Key issues in a Monte Carlo inference algorithm:

- The sampling mechanism
- The functions (estimators) which compute the probabilities from the sample

Wrapping up

Today we discussed:

1. Learning probabilistic models from data
2. Variable elimination algorithm
3. Events, Evidence and Queries
4. Exact and Approximate Inference
5. MonteCarlo inference algorithms

Some References

- Pratt J., Raiffa H., Schlaifer R., Introduction to Statistical Decision Theory, The MIT Press, 2008
- Hoffmann M. D., Gelman A., The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo, arXiv:1111.4246, 2011
- A. Gelman, J. B. Carlin, H. S. Stern, Bayesian Data Analysis, CRC Press, 2013
- Walsh B., Markov Chain Monte Carlo and Gibbs Sampling, Lecture Notes for EEB 596z, 2002
- R. A. Howard, Dynamic Programming and Markov Process, The MIT Press, 1960
- Rabiner L. R., A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77.2, 1989
- W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrik, 57/1, 04/1970
- Kevin B. Korb, Ann E. Nicholson, Bayesian Artificial Intelligence, CRC Press, 2010 Pearl J., Causality, Cambridge University Press, 2009
- L. E. Baum, T. Petrie, Statistical Inference for Probabilistic Functions of Finite State Markov Chains, The Annals of Mathematical Statistics, 37, 1966

Thank you very much!

ANY QUESTIONS OR COMMENTS?