ECE521 Lecture 15

Graphical Models

Bayesian Network

UNIVERSITY OF TORONTO

Outline

- **Generative models**
    - Naive Bayes: connection between MoG and logistic regression
- Conditional Independence
- Bayesian network

Generative model

- A generative model is a probability model of a set of **hidden / latent / unobserved** variables and **visible** variables.
  - The visible variables “match” with the training dataset
- E.g. a simple 1-D Mixture of Gaussian model: the joint distribution of the **latent variable z** and **visible variable x** is defined as:

$$p(z, x) = p(z)p(x|z)$$

$$p(z = k) = \pi_k, \quad z \in \{1, \ldots, K\}$$ ← K clusters

$$p(x|z = k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$ ← Gaussian

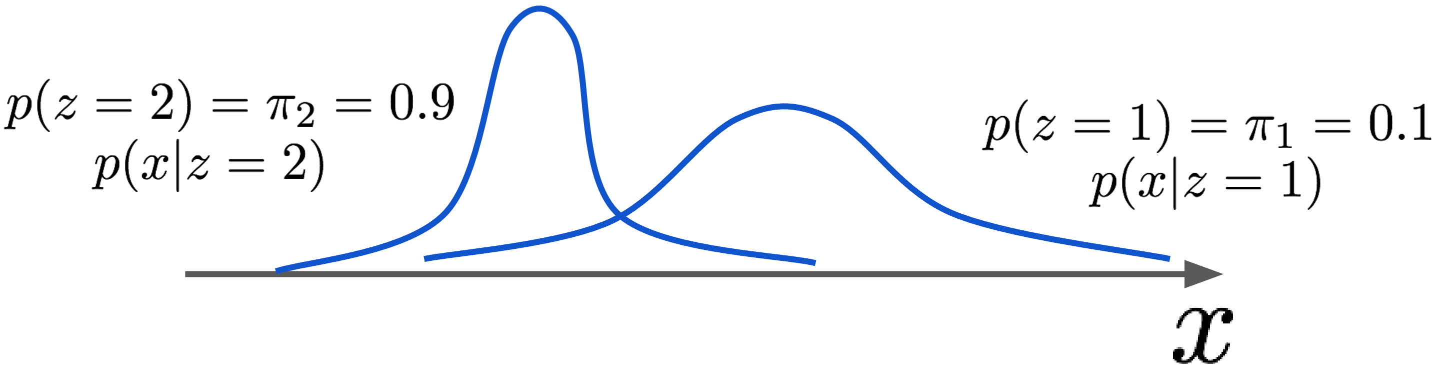
$$p(z, x) = p(z)p(x|z)$$

$$p(z = k) = \pi_k, \quad z \in \{1, \dots, K\}$$

$$p(x|z = k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

Generative model

- Assume there are two clusters / mixture components, i.e. K = 2 and z in {1, 2}
- We can **generating data** from a Mixture of Gaussian model**:** sample z from p(z), then sample x from p(x | z)
    - This gives a joint sample of (x, z) from the joint probability distribution p(x, z) in two steps



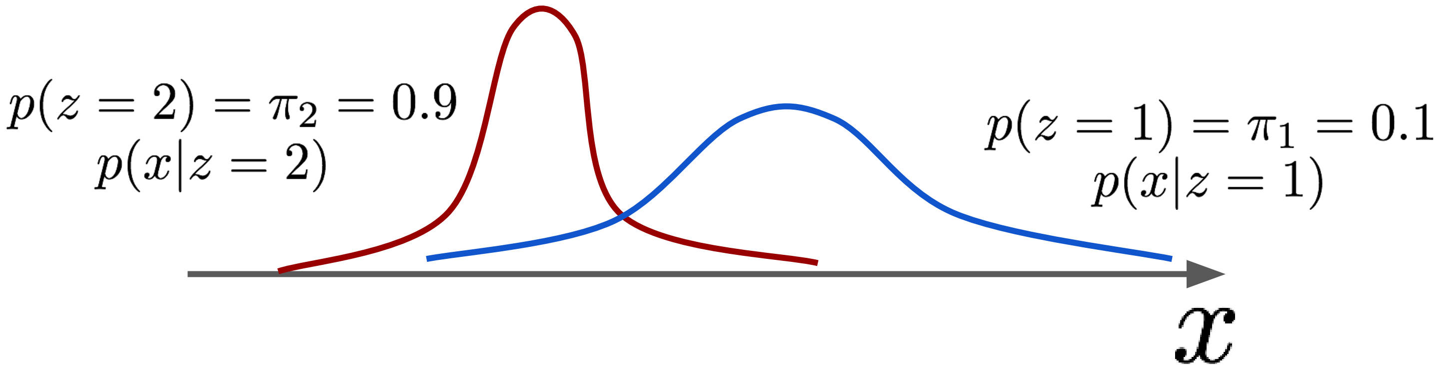
Generative model

$$p(z, x) = p(z)p(x|z)$$

$$p(z = k) = \pi_k, \quad z \in \{1, \ldots, K\}$$

$$p(x|z = k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

- Assume there are two clusters / mixture components, i.e. K = 2 and z in {1, 2}
- We can **generating data** from a Mixture of Gaussian model**:** sample z from p(z), then sample x from p(x | z)
  - This gives a joint sample of (x, z) from the joint probability distribution p(x, z) in two steps



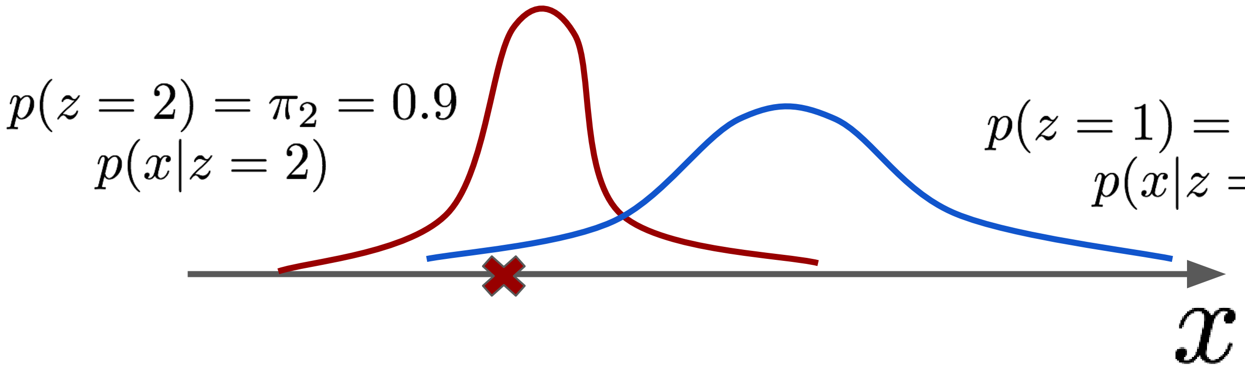
Generative model

$$p(z, x) = p(z)p(x|z)$$

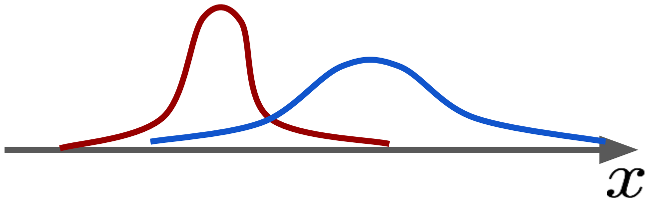
$$p(z = k) = \pi_k, \quad z \in \{1, \ldots, K\}$$

$$p(x|z = k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

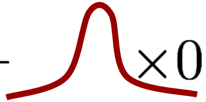
- Assume there are two clusters / mixture components, i.e. K = 2 and z in {1, 2}
- We can **generating data** from a Mixture of Gaussian model**:** sample z from p(z), then sample x from p(x | z)
    - This gives a joint sample of (x, z) from the joint probability distribution p(x, z) in two steps



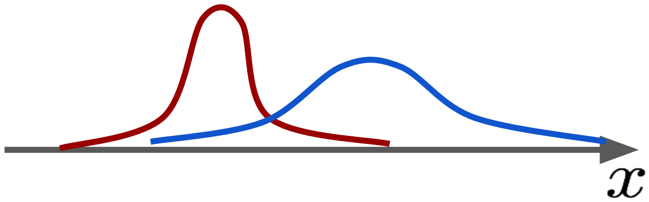
Generative model



- Assume there are two clusters / mixture components, i.e. K = 2 and z in {1, 2}
- We can **generating data** from a Mixture of Gaussian model**:** sample z from p(z), then sample x from p(x | z)
  - This gives a joint sample of (x, z) from the joint probability distribution p(x, z) in two steps
  - Marginal distribution is a weighted sum of the two distributions

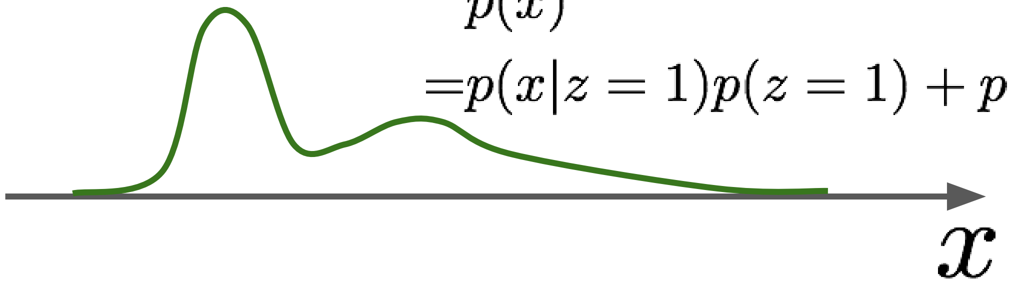
$$p(x) = \sum_{z=1}^{2} p(x, z) = \sum_{z=1}^{2} p(x|z)p(z) =$$  $\times 0.1 +$  $\times 0.9$

Generative model

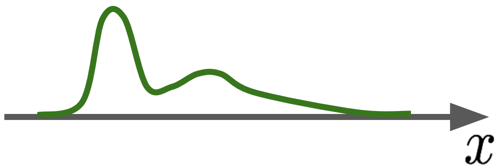


- Assume there are two clusters / mixture components, i.e. K = 2 and z in {1, 2}
- We can **generating data** from a Mixture of Gaussian model**:** sample z from p(z), then sample x from p(x | z)
  - This gives a joint sample of (x, z) from the joint probability distribution p(x, z) in two steps
  - Marginal distribution is a weighted sum of the two distributions

$$p(x)$$
$$= p(x|z=1)p(z=1) + p(x|z=2)p(z=2)$$



Generative model



- Question: Can any probability density function be modeled as a Mixture of Gaussians?

- Answer: Yes! A weighted sum of infinite number of Gaussians can approximate any continuous PDF to arbitrary desired degree of accuracy.
    - By introducing latent variables, we can now model any non-trivial PDFs from summing a few simple Gaussian distributions.

Naive Bayes

- Instead of using generative model to fit the input feature distribution p(x), here we will build a classifier based on a simple generative model and Bayes rules.
    - Let latent variable z be the class label
    - Suppose we have a generative model of x given z: p(x | z), E.g. p(x|z) is a Gaussian
        - We can easily learn these Gaussians by take all the x labeled from one class and fit a Gaussian on them.
- How do we classify a test case?

Naive Bayes

- Instead of using generative model to fit the input feature distribution p(x), here we will build a classifier based on a simple generative model and Bayes rules.
  - Let latent variable z be the class label
  - Suppose we have a generative model of x given z: p(x | z), E.g. p(x|z) is a Gaussian
    - We can easily learn these Gaussians by take all the x labeled from one class and fit a Gaussian on them.
- How do we classify a test case?
  - Perform Bayesian inference (Bayes' rule) to get the posterior distribution over the label z

$$p(z=k|x) = \frac{p(x|z=k)p(z=k)}{\sum_j p(x|z=j)p(z=j)}$$

Naive Bayes

$$p(z=k|x) = \frac{p(x|z=k)p(z=k)}{\sum_j p(x|z=j)p(z=j)}$$

- Suppose the input features are D-dimensional vectors $\mathbf{x} = (x_1, \dots, x_D)$.

- **Naive Bayes** assumption: all dimensions of x are **independent given the label z.**
  - Conditional independence is very strong (naive) assumption on generative process of the data. Intuitively, the data are generated by first pick a label z ∈ {1,...,K} then generate all the input feature in parallel conditioned on the label.

$$p(\mathbf{x}|z=k) = p(x_1, \dots, x_D|z=k)$$
$$= \prod_{d=1}^{D} p(x_d|z=k)$$

Naive Bayes

$$p(z=k|x) = \frac{p(x|z=k)p(z=k)}{\sum_j p(x|z=j)p(z=j)}$$

- E.g. consider a Naive Bayes classifier where the input features $\mathbf{x} = (x_1, \ldots, x_D)$ are conditional Gaussian given the label z and there are two classes z ∈ {1,2}
- Let us first derive the expression of the posterior distribution of the label p(z|x) for 1-dimensional input features:

$$p(x|z=1) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}$$

$$p(z=1|x) = \frac{\pi_1 p(x|z=1)}{\pi_1 p(x|z=1) + \pi_2 p(x|z=2)}$$

$$= \frac{\frac{\pi_1}{\sqrt{2\pi\sigma_1^2}} \exp\{-\frac{(x-\mu_1)^2}{2\sigma_1^2}\}}{\frac{\pi_1}{\sqrt{2\pi\sigma_1^2}} \exp\{-\frac{(x-\mu_1)^2}{2\sigma_1^2}\} + \frac{\pi_2}{\sqrt{2\pi\sigma_2^2}} \exp\{-\frac{(x-\mu_2)^2}{2\sigma_2^2}\}}$$

Naive Bayes

$$p(z=k|x) = \frac{p(x|z=k)p(z=k)}{\sum_j p(x|z=j)p(z=j)}$$

- Let us further simplify the expression by assuming $\sigma_1^2 = \sigma_2^2 = \sigma^2$
  - The coefficients $\frac{1}{\sqrt{2\pi\sigma^2}}$ terms cancels out

$$p(z=1|x) = \frac{1}{1+\exp\{-\frac{\mu_1-\mu_2}{\sigma^2}x - \log\frac{\pi_1}{\pi_2} - \frac{\mu_2^2-\mu_1^2}{2\sigma^2}\}}$$

$$= \frac{1}{1+\exp\{-w_1 x - w_0\}}$$

  - Let w1 and w0 denote the coefficient in front of the 1st order and zero order terms

$$w_1 = \frac{\mu_1-\mu_2}{\sigma^2} \quad w_0 = \log\frac{\pi_1}{\pi_2} + \frac{\mu_2^2-\mu_1^2}{2\sigma^2}$$

  - The posterior distribution is a **sigmoid / logistic function**!

Naive Bayes

$$p(z=k|x) = \frac{p(x|z=k)p(z=k)}{\sum_j p(x|z=j)p(z=j)}$$

- Let us further simplify the expression by assuming $\sigma_1^2 = \sigma_2^2 = \sigma^2$
  - The coefficients $\frac{1}{\sqrt{2\pi\sigma^2}}$ terms cancels out

$$p(z=1|x) = \frac{1}{1+\exp\{-\frac{\mu_1-\mu_2}{\sigma^2}x - \log\frac{\pi_1}{\pi_2} - \frac{\mu_2^2-\mu_1^2}{2\sigma^2}\}}$$

$$= \frac{1}{1+\exp\{-w_1 x - w_0\}}$$

  - Let w1 and w0 denote the coefficient in front of the 1st order and zero order terms

$$w_1 = \frac{\mu_1-\mu_2}{\sigma^2} \quad w_0 = \log\frac{\pi_1}{\pi_2} + \frac{\mu_2^2-\mu_1^2}{2\sigma^2}$$

  - The posterior distribution is a **sigmoid / logistic function**!

Naive Bayes

$$p(z=k|x) = \frac{p(x|z=k)p(z=k)}{\sum_j p(x|z=j)p(z=j)}$$

- Let us further simplify the expression by assuming $\sigma_1^2 = \sigma_2^2 = \sigma^2$
  - The coefficients $\frac{1}{\sqrt{2\pi\sigma^2}}$ terms cancels out

$$p(z=1|x) = \frac{1}{1+\exp\{-\frac{\mu_1-\mu_2}{\sigma^2}x - \log\frac{\pi_1}{\pi_2} - \frac{\mu_2^2-\mu_1^2}{2\sigma^2}\}}$$

$$= \frac{1}{1+\exp\{-w_1 x - w_0\}}$$

  - Let w1 and w0 denote the coefficient in front of the 1st order and zero order terms

$$w_1 = \frac{\mu_1 - \mu_2}{\sigma^2} \quad w_0 = \log\frac{\pi_1}{\pi_2} + \frac{\mu_2^2 - \mu_1^2}{2\sigma^2}$$

  - The posterior distribution is a **sigmoid / logistic function**!

Naive Bayes

$$p(z=k|x) = \frac{p(x|z=k)p(z=k)}{\sum_j p(x|z=j)p(z=j)}$$

- Let us further simplify the expression by assuming $\sigma_1^2 = \sigma_2^2 = \sigma^2$
  - The coefficients $\frac{1}{\sqrt{2\pi\sigma^2}}$ terms cancels out

$$p(z=1|x) = \frac{1}{1+\exp\{-\frac{\mu_1-\mu_2}{\sigma^2}x - \log\frac{\pi_1}{\pi_2} - \frac{\mu_2^2-\mu_1^2}{2\sigma^2}\}}$$

$$= \frac{1}{1+\exp\{-w_1 x - w_0\}}$$

  - Let w1 and w0 denote the coefficient in front of the 1st order and zero order terms

$$w_1 = \frac{\mu_1 - \mu_2}{\sigma^2} \quad w_0 = \log\frac{\pi_1}{\pi_2} + \frac{\mu_2^2 - \mu_1^2}{2\sigma^2}$$

  - The posterior distribution is a **sigmoid / logistic function**!

Naive Bayes

$$p(z=k|x) = \frac{p(x|z=k)p(z=k)}{\sum_j p(x|z=j)p(z=j)}$$

- Let us extend the expression to D-dimensional data $\mathbf{x} = (x_1, \ldots, x_D)$

$$\begin{aligned} p(z=1|\mathbf{x}) &= p(z=1|x_1, \ldots, x_d) \\ &= \frac{1}{1+\exp\{-\sum_{d=1}^{D} \frac{\mu_{d,1}-\mu_{d,2}}{\sigma^2} x_d - \log\frac{\pi_1}{\pi_2} - \sum_{d=1}^{D} \frac{\mu_{d,2}^2-\mu_{d,1}^2}{2\sigma^2}\}} \\ &= \frac{1}{1+\exp\{-\sum_{d=1}^{D} w_d x_d - w_0\}} \end{aligned}$$

  - It suggest that naive Bayes is a linear classifier similar to logistic regression

High Dimensional Probability Models

- Suppose we have a D-dimensional input features $\mathbf{x} = (x_1, \ldots, x_D)$
- D can be in the order of thousands or millions.

$$p(\mathbf{x}) = p(x_1, \ldots, x_D)$$

- E.g. each dimension can either be discrete or continuous

$$x_i \in \mathcal{S}_i, \quad \text{e.g. } \mathcal{S}_i = \{0, 1\}$$

$$\mathcal{S}_i = \mathbb{R}$$

$$\mathcal{S}_i = \mathbb{I}^+$$

High Dimensional Probability Models

$$p(\mathbf{x}) = p(x_1, \ldots, x_D)$$

- Now, recall that we can write any joint distributions using the product rule in terms of the conditionals:

$$p(x, y) = p(x|y)p(y)$$

- Define y = (x2, x3, …, xD), then we can recursively write the joint as a product of the conditionals:

$$\begin{aligned} p(x_1, \ldots, x_D) &= p(x_1|x_2, \ldots, x_D)p(x_2, \ldots, x_D) \\ &= \prod_{d=1}^{D} p(x_d|x_{d+1}, \ldots, x_D) \\ &= \prod_{d=1}^{D} p(x_d|x_1, \ldots, x_{d-1}) \end{aligned}$$

order does not matter

High Dimensional Probability Models

- Suppose we take the following specific order of the conditionals:

$$p(x_1, \ldots, x_D) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \ldots p(x_D|x_1, \ldots, x_{D-1})$$

- E.g. $x_d, x_{d+1}, x_{d+2}, \ldots$ can be the stock price on the d*th* day which depends on the price in the previous d-1 days
    - The problem is D can be very long, a year?, 10 years?
    - The later conditional distributions are huge, intractable as the number of the state space is exponentially growing $\prod_{j=1}^{D-1} |\mathcal{S}_j|$, e.g. binary variables will have 2^(D-1) states

High Dimensional Probability Models

- Suppose we take the following specific order of the conditionals:

$$p(x_1, \ldots, x_D) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \ldots p(x_D|x_1, \ldots, x_{D-1})$$

- How can we compute the conditional probability efficiently?
- **Solution**: let us assume that x_d only depends on x_{d-1}

$$p(x_d|x_1, \ldots, x_{d-1}) = p(x_d|x_{d-1})$$

  - E.g. the stock price of today only depends on yesterday's price and not the prices before yesterday

High Dimensional Probability Models

- Suppose we take the following specific order of the conditionals:

$$p(x_1, \ldots, x_D) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \ldots p(x_D|x_1, \ldots, x_{D-1})$$

- How can we compute the conditional probability efficiently?
- **Solution**: let us assume that x_d only depends on x_{d-1}

$$p(x_d|x_1, \ldots, x_{d-1}) = p(x_d|x_{d-1})$$ ← conditional independence assumption

  - E.g. the stock price of today only depends on yesterday's price and not the prices before yesterday

Conditional Independence

- Formally, we say that xi is independent of xj given xk if:

$$p(x_i, x_j | x_k) = p(x_i | x_k) p(x_j | x_k)$$

Conditional Independence

- Formally, we say that xi is independent of xj given xk if:

$$p(x_i, x_j | x_k) = p(x_i | x_k) p(x_j | x_k)$$

- E.g. x_d is independent of $x_1, x_2, \ldots, x_{d-2}$ given x_{d-1} express the following factorization:

$$p(x_1, \ldots, x_{d-2}, x_d | x_{d-1}) = p(x_d | x_{d-1}) p(x_1, \ldots, x_{d-2} | x_{d-1})$$

Conditional Independence

- Formally, we say that xi is independent of xj given xk if:

$$p(x_i, x_j | x_k) = p(x_i | x_k) p(x_j | x_k)$$

- E.g. the stock price at day d is independent of day 1,..., d-2, given the stock price at day d-1:

$$p(x_1, \ldots, x_D) = p(x_1) \prod_{i=2}^{D} p(x_d | x_{d-1})$$

  - This is also known as the "**Markov assumption**", i.e. "future is independent of the past given present"
  - Huge computational gain by assuming the conditional independence

Conditional Independence

- Formally, we say that xi is independent of xj given xk if:

$$p(x_i, x_j | x_k) = p(x_i | x_k) p(x_j | x_k)$$

- Notation for conditional independence:

$$x_i \perp\!\!\!\perp x_j | x_k$$

Bayesian Network

- It is often clumsy to write down the probability to express our model and we may also need to separately specify conditional independence assumptions.

- **Graphical models** are a set of tools for us to express our modelling assumption **visually** that can be easily read off from graph.
    - There are many classes of graphical models: Bayesian networks, Markov random fields, factor graphs and some hybrid graphs.

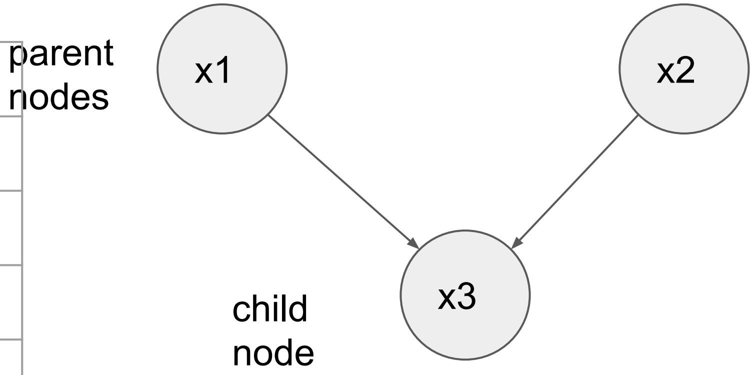
Bayesian Network

- A Bayesian network is a directed acyclic graph on **a set of nodes** corresponding to x1, …, xd, plus a **conditional probability model** for each child given its parents.
  - Directed acyclic graph (DAG) has no cycles when following the arrows of the graph.
- E.g. $x_1, x_2, x_3 \in \{0, 1\}$

x1	p(x1)
0	0.9
1	0.1

x2	p(x2)
0	0.2
1	0.8

x1	x2	x3	p(x3|x1,x2)
0	0	0	0.7
0	0	1	0.3
0	1	0	0.5
...	...	...	...



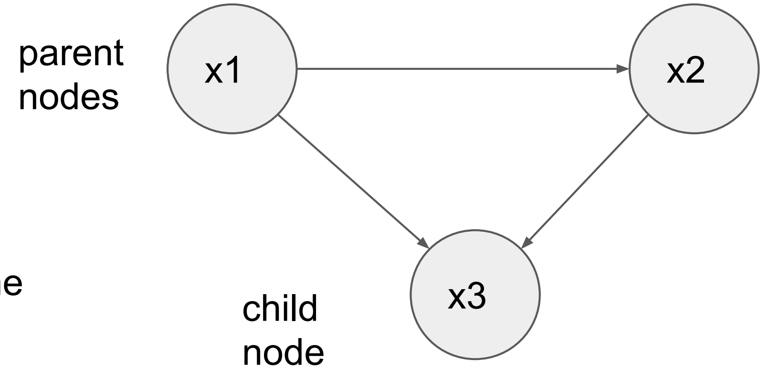
Bayesian Network

- A Bayesian network is a directed acyclic graph on **a set of nodes** corresponding to x1, …, xd, plus a **conditional probability model** for each child given its parents.
  - Directed acyclic graph (DAG) has no cycles when following the arrows of the graph.
- E.g. $x_1, x_2, x_3 \in \{0, 1\}$

x1	p(x1)
0	0.9
1	0.1

x1	x2	p(x2|x1)
0	0	0.2
0	1	0.8
1	0	0.8
1	1	0.2

p(x3|x1,x2) same as before



Bayesian Network

- If only the DAG is provided, then the DAG refers to **all** probability distributions corresponding to different choice for p(x | parents)

- A Bayes net implies that: $$p(x_1, \ldots x_D) = \prod_{d=1}^{D} p(x_d | X_{\mathcal{A}_d})$$
    - where $\mathcal{A}_d$ are the indices of the parents of xd and $X_{\mathcal{A}_d}$ are their values.

Bayesian Network

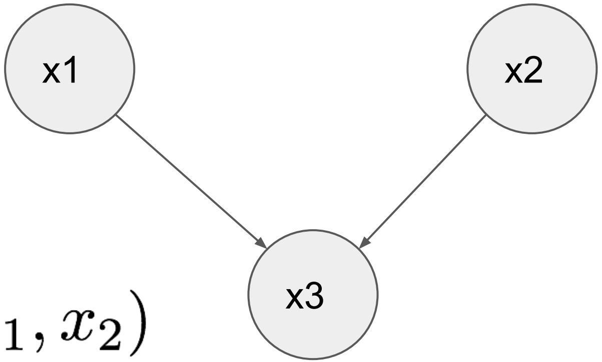
- A Bayes net implies that: $p(x_1, \ldots x_D) = \prod_{d=1}^{D} p(x_d | X_{\mathcal{A}_d})$
  - where $\mathcal{A}_d$ are the indices of the parents of xd and $X_{\mathcal{A}_d}$ are their values.

- E.g.

$$\mathcal{A}_1 = \emptyset, \quad \mathcal{A}_2 = \emptyset$$

$$\mathcal{A}_3 = \{1, 2\}$$

$$p(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3 | x_1, x_2)$$



Bayesian Network

- A Bayes net implies that: $p(x_1, \ldots x_D) = \prod_{d=1}^{D} p(x_d | X_{\mathcal{A}_d})$
  - where $\mathcal{A}_d$ are the indices of the parents of xd and $X_{\mathcal{A}_d}$ are their values.

- E.g. recall Markov assumption:

$$p(x_1, \ldots, x_D) = p(x_1) \prod_{i=2}^{D} p(x_d | x_{d-1})$$