# Deep Generative Models

## Structured density

Hamid Beigy

Sharif University of Technology

February 19, 2024

# Table of contents

# Introduction

# Density estimation

1. Density estimation is the problem of reconstructing the probability density function using a set of given data points.
2. In density estimation, we observe $X_1, \ldots, X_n \sim P$ and we want to recover the underlying probability density function generating this dataset.
3. $P$ is the underlying population CDF and $p$ as it pdf .
4. We assume that $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are identically independently distributed random variables. Hence,

$$\mathrm{p}(x_1, \ldots, x_n) = \prod_{k=1}^{n} \mathrm{p}(x_k)$$

5. Density can be estimated using two approaches:
   - Parametric approach
   - Non-parametric approach

# Parametric density estimation approach

# Parametric density estimation approach

1. Assume that we can approximate the probability density function $p$ using density function $\mathrm{p}_\theta(x)$.
2. Here, $\theta$ is parameters of density function $\mathrm{p}_\theta(x)$.
3. There are many approaches for estimating $\theta$ such as
   - maximum likelihood method (ML)
   - maximum a posteriori probability (MAP)
   - method of moments
   - Bayesian estimation method
4. Let $\mathbf{X}_i$ be a one-dimensional real valued random variable.
5. Let be the target pdf, where $\theta = \{\mu, \sigma^2\}$ is its parameters.

1. Let $\mathrm{p}_\theta(x) = \mathcal{N}(\mu, \sigma^2)$. Then $\theta = \{\mu, \sigma^2\}$ .

2. The likelihood equals

$$L(\theta) = \mathrm{p}(x_1, \ldots, x_n; \theta) = \prod_{k=1}^{n} \mathrm{p}(x_k; \theta)$$

$$\mathrm{LL}(\theta) = \ln L(\theta) = \sum_{k=1}^{n} \mathrm{p}(x_k; \theta)$$

3. By differentiating $\mathrm{LL}(\theta)$ with respect to $\theta$ and setting to zero, we obtain

$$\widehat{\mu}_n = \frac{1}{n} \sum_{k=1}^{n} x_k$$

$$\widehat{\sigma}_n^2 = \frac{1}{n} \sum_{k=1}^{n} (x_k - \widehat{\mu}_n)^2$$

4. Then the resulting density function is

$$\widehat{\mathrm{p}}_n(x) = \frac{1}{\sqrt{2\pi\widehat{\sigma}^2}} e^{-\frac{1}{2}\left(\frac{x-\widehat{\mu}_n}{\widehat{\sigma}_n}\right)^2}$$

**Definition (Bias of an estimator)**

Let $\widehat{\theta}$ be a point estimator for $\theta$. The bias of point estimator $\widehat{\theta}$ is defined by

$$\text{Bias } (\widehat{\theta}) = \mathbb{E}\left[\widehat{\theta}\right] - \theta.$$

**Definition (Unbiased estimator)**

Let $\widehat{\theta}$ be a point estimator for $\theta$. We say that the point estimator $\widehat{\theta}$ is an unbiased estimator of $\theta$ if for all values of $\theta$, we have

$$\text{Bias } (\widehat{\theta}) = 0.$$

**Example (Unbiased estimator)**

Let $\widehat{\mu}_n = \frac{1}{n}\sum_{k=1}^{n} X_k$, then $\widehat{\mu}_n$ is an unbiased estimator.

$$\begin{aligned}\text{Bias } (\widehat{\mu}_n) &= \mathbb{E}[\widehat{\mu}_n] - \mu = \mathbb{E}\left[\frac{1}{n}\sum_{k=1}^{n} X_k\right] - \mu \\ &= \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}[X_k] - \mu = \mu - \mu = 0.\end{aligned}$$

**Example (Biased estimator)**

Let $\widehat{\sigma}_n^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \widehat{\mu}_n)^2$, then $\widehat{\sigma^2}_n$ is a biased estimator.

$$
\begin{aligned}
\text{Bias}\,(\widehat{\sigma}_n^2) &= \mathbb{E}\left[\frac{1}{n}\sum_{k=1}^{n}(x_k - \widehat{\mu}_n)^2\right] - \sigma^2 \\
&= \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[(x_k - \frac{1}{n}\sum_{j=1}^{n}x_j)^2\right] - \sigma^2 \\
&= \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[x_k^2 - \frac{2}{n}x_k\sum_{j=1}^{n}x_j + \frac{1}{n^2}\sum_{k=1}^{n}x_k\sum_{j=1}^{n}x_j\right] - \sigma^2 \\
&= \frac{1}{n}\sum_{k=1}^{n}\left[\frac{n-2}{n}\mathbb{E}[x_k^2] - \frac{2}{n}\sum_{j\neq k}^{n}\mathbb{E}[x_k x_j] + \frac{1}{n^2}\sum_{j=1}^{n}\sum_{k\neq j}\mathbb{E}[x_k x_j] + \frac{1}{n^2}\sum_{j=1}^{n}\mathbb{E}[x_j^2]\right] - \sigma^2 \\
&= \frac{1}{n}\sum_{k=1}^{n}\left[\frac{n-2}{n}(\mu^2+\sigma^2) - \frac{2(n-1)}{n}\mu^2 + \frac{n(n-1)}{n^2}\mu^2 + \frac{1}{n}(\mu^2+\sigma^2)\right] - \sigma^2 \\
&= \frac{1}{n}\sum_{k=1}^{n}\left[\left(\frac{n-1}{n}\right)\sigma^2\right] - \sigma^2 = \left(\frac{n-1}{n}\right)\sigma^2 - \sigma^2 \neq 0.
\end{aligned}
$$

**Definition (Mean squared error of an estimator)**

The mean squared error (MSE) of a point estimator $\widehat{\theta}$, shown by MSE $(\widehat{\theta})$, is defined as

$$\text{MSE}\ (\widehat{\theta}) = \mathbb{E}\left[\left(\widehat{\theta} - \theta\right)^2\right].$$

**Example**

Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be a random sample from a distribution with mean $\mathbb{E}[\mathbf{X}_i] = \theta$ and variance $\text{var}[\mathbf{X}_i] = \sigma^2$. Consider two estimators for $\theta$

$$\widehat{\theta}_1 = \mathbf{X}_1 \qquad\qquad \widehat{\theta}_2 = \frac{1}{n}\sum_{k=1}^{n} \mathbf{X}_k.$$

These two estimators are both unbiased. Hence, we study their MSE:

$$\text{MSE}\ (\widehat{\theta}_1) = \mathbb{E}\left[\left(\widehat{\theta}_1 - \theta\right)^2\right] = \mathbb{E}\left[\left(\mathbf{X}_1 - \mathbb{E}[\mathbf{X}_1]\right)^2\right] = \text{var}[\mathbf{X}_1] = \sigma^2.$$

$$\text{MSE}\ (\widehat{\theta}_2) = \mathbb{E}\left[\left(\widehat{\theta}_2 - \theta\right)^2\right] = \mathbb{E}\left[\left(\frac{1}{n}\sum_{k=1}^{n} \mathbf{X}_k - \theta\right)^2\right] = \frac{\sigma^2}{n}.$$

Thus MSE $(\widehat{\theta}_1) >$ MSE $(\widehat{\theta}_2)$. Hence, $\widehat{\theta}_2$ is better.

**Definition (Consistency of an estimator)**

Let $\widehat{\theta}_1, \widehat{\theta}_2, \ldots, \widehat{\theta}_n, \ldots$ be a sequence of point estimators of $\theta$. We say $\widehat{\theta}_n$ is a **consistent estimator** of $\theta$, if

$$\lim_{n\to\infty} \mathrm{p}(|\widehat{\theta}_n - \theta| \geq \epsilon) = 0, \text{ for all } \epsilon > 0.$$

**Example (Consistency of sample average)**

Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be a random sample from a distribution with mean $\mathbb{E}[\mathbf{X}_i] = \theta$ and variance $\mathrm{var}[\mathbf{X}_i] = \sigma^2$. Consider the following estimator for $\theta$

$$\widehat{\theta}_n = \frac{1}{n}\sum_{k=1}^{n} \mathbf{X}_k.$$

We have found that $\mathrm{MSE}\,(\widehat{\theta}_n) = \frac{\sigma^2}{n}$. Thus,

$$\lim_{n\to\infty} \mathrm{MSE}\,(\widehat{\theta}_n) \to 0.$$

**Hence, this estimator is consistent.**

**Theorem**

*Let* $\widehat{\theta}$ *is a point estimator for* $\theta$*. Then* $\text{MSE}\ (\widehat{\theta}) = \text{var}[\widehat{\theta}] + \text{Bias}\ (\widehat{\theta})^2$

**Theorem (Consistency of an estimator)**

*Let* $\widehat{\theta}_1, \widehat{\theta}_2, \ldots$ *be a sequence of point estimators of* $\theta$*. If* $\lim_{n\to\infty} \text{MSE}\ (\widehat{\theta}_n) = 0$*, then* $\widehat{\theta}_n$ *is a* **consistent estimator** *of* $\theta$*.*

**Proof.**

We can write

$$\begin{aligned} \text{p}(|\widehat{\theta}_n - \theta| \geq \epsilon) &= \text{p}(|\widehat{\theta}_n - \theta|^2 \geq \epsilon^2) \\ &\leq \frac{\mathbb{E}\left[\left(\widehat{\theta}_n - \theta\right)^2\right]}{\epsilon^2} \qquad \textit{byMarkov's inequality} \\ &= \frac{\text{MSE}\ (\widehat{\theta}_n)}{\epsilon^2}, \end{aligned}$$

which goes to 0 as $n \to \infty$ by the assumption. □

**Definition (Convergence in Probability)**

A sequence of random variables $\mathbf{Z}_1, \mathbf{Z}_2, \ldots$ **converges in probability** to a random variable $\mathbf{Z}$, shown by $\mathbf{Z}_n \xrightarrow{p} \mathbf{Z}$, if

$$\lim_{n\to\infty} \mathrm{p}(|\mathbf{Z}_n - \mathbf{Z}| \geq \epsilon) = 0, \qquad \text{for all } \epsilon > 0.$$

This implies that the distribution is concentrating at the targeting point.

**Lemma**

*Let $\widehat{\theta}$ be an estimator of $\theta$. If* Bias $(\widehat{\theta}) \to 0$ *and* $\mathrm{var}[\widehat{\theta}] \to 0$, *then $\widehat{\theta} \xrightarrow{p} \theta$, i.e. $\widehat{\theta}$ is a* **consistent** *estimator of $\theta$.*

**Definition (Convergence in Distribution)**

Let $F_1, F_2, \ldots$ be the corresponding CDFs of $\mathbf{Z}_1, \mathbf{Z}_2, \ldots$. For a random variable $\mathbf{Z}$ with CDF $F$, we say $\mathbf{Z}_n$ **converges in distribution** to a random variable $\mathbf{Z}$, shown by $\mathbf{Z}_n \xrightarrow{d} \mathbf{Z}$, if

$$\lim_{n\to\infty} F_n(x) = F(x),$$

This implies that $F_n$ converge to the CDF of a fixed random variable.

# Stochastic order notations

**Definition**

For a sequence of numbers $a_n$ (indexed by $n$), we write

1. $a_n = o(1)$ if $\lim_{n\to\infty} a_n \to 0$. For another sequence $b_n$, we write $a_n = o(b_n)$ if $\frac{a_n}{b_n} = o(1)$.
2. $a_n = O(1)$ if for all large $n$, there exists a constant $C$ such that $|a_n| < C$. For another sequence $b_n$, we write $a_n = O(b_n)$ if $\frac{a_n}{b_n} = O(1)$.

The $O_p$ and $o_p$ are similar notations to $O$ and $o$ but are designed for random numbers.

**Definition**

For a sequence of random variables $\mathbf{X}_n$, we write

1. $\mathbf{X}_n = o_p(1)$ if for any $\epsilon > 0$,

$$\lim_{n\to\infty} \mathrm{p}(|\mathbf{X}_n| > \epsilon) \to 0$$

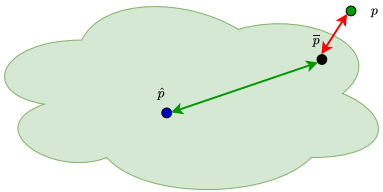
   Namely, $\mathrm{p}(|\mathbf{X}_n| > \epsilon) = o_p(1)$ for any $\epsilon > 0$. Let $a_n$ be a nonrandom sequence, we write $\mathbf{X}_n = o_p(a_n)$ if $\frac{\mathbf{X}_n}{a_n} = o_p(1)$.
2. $\mathbf{X}_n = O_p(1)$ if for any $\epsilon > 0$, there exists a constant $C$ such that

$$\mathrm{p}(|\mathbf{X}_n| > C) < \epsilon.$$

   We write $\mathbf{X}_n = O_p(a_n)$ if $\frac{\mathbf{X}_n}{a_n} = O_p(1)$.

**Is the parametric approach a good one?** We analyze the quality of estimation in the parametric approach for Gaussian distribution.

1. We quantify $\widehat{\mathrm{p}}_n(x) - \mathrm{p}(x)$.




2. Since the sample mean $\widehat{\mu} \xrightarrow{P} \overline{\mu} = \mathbb{E}[\mathbf{X}]$ and the sample variance $\widehat{\sigma}^2 \xrightarrow{P} \overline{\sigma}^2 = \mathrm{var}[\mathbf{X}]$, we define another density function

$$\overline{\mathrm{p}}(x) = \frac{1}{\sqrt{2\pi\overline{\sigma}^2}} e^{-\frac{1}{2}\left(\frac{x-\overline{\mu}_n}{\overline{\sigma}_n}\right)^2}$$

3. The estimated density function is

$$\widehat{\mathrm{p}}_n(x) = \frac{1}{\sqrt{2\pi\widehat{\sigma}^2}} e^{-\frac{1}{2}\left(\frac{x-\widehat{\mu}_n}{\widehat{\sigma}_n}\right)^2}$$

# Analysis of parametric density estimation approach

1. Using $\overline{\mathrm{p}}(x)$, we have

$$\widehat{\mathrm{p}}_n(x) - \mathrm{p}(x) = \widehat{\mathrm{p}}_n(x) - \overline{\mathrm{p}}(x) + \overline{\mathrm{p}}(x) - \mathrm{p}(x)$$

2. The first difference $\widehat{\mathrm{p}}_n(x) - \overline{\mathrm{p}}(x)$ is something that converges to $0$ because the sample mean and variance converges to their population counterparts. Namely, we have $\widehat{\mathrm{p}}_n(x) \xrightarrow{p} \overline{\mathrm{p}}(x)$.

3. However, the second difference $\overline{\mathrm{p}}_n(x) - \mathrm{p}(x)$ never goes to $0$ unless the true pdf is Gaussian.

$$p_n(x) = \frac{1}{\sqrt{2\pi\widehat{\sigma}^2}} e^{-\frac{1}{2}\left(\frac{x-\widehat{\mu}_n}{\widehat{\sigma}_n}\right)^2}$$

4. We study the convergence rate of $\widehat{\mathrm{p}}_n(x) - \overline{\mathrm{p}}(x)$ as

$$\widehat{\mathrm{p}}_n(x) - \overline{\mathrm{p}}(x) = O_p\left(\frac{1}{\sqrt{n}}\right).$$

5. This will help us understand when a parametric approach may be better than a nonparametric approach.

1. Let the parametric model be

$$\widehat{\mathrm{p}}_n(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \sigma_k^2)$$

$$\sum_{k=1}^{K} \pi_k = 1$$

2. Then, we compute parameters $\theta = \{\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots, \sigma_K^2, \pi_1, \ldots, \pi_K\}$ based on training data.

3. We use EM algorithm to estimate the parameters.

4. The convergence rate of $\widehat{\mathrm{p}}_n(x) - \overline{\mathrm{p}}(x)$ equals to

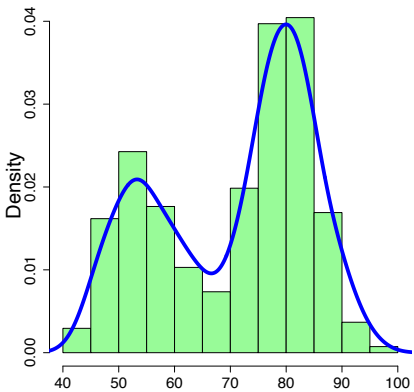
$$\widehat{\mathrm{p}}_n(x) - \overline{\mathrm{p}}(x) = O_p\left(\frac{1}{\sqrt{n}}\right).$$

# Nonparametric density estimation approach

1. For simplicity, we assume that $\mathbf{X}_i \in [0,1]$. So $\mathrm{p}(x) > 0$ in interval $[0,1]$.
2. We also assume that $\mathrm{p}(x) > 0$ is smooth and $|\mathrm{p}(x)'| \leq L$ for all $x$.
3. In histogram we partition interval $[0,1]$ into $M$ bins $(B_k)$ of equal lengths as
$$B_k = \left[\frac{k-1}{M}, \frac{k}{M}\right]$$
4. Then, we count the number of samples in a bin as density estimate. Hence, then for a given point $x \in B_l$, the density estimator from the histogram will be
$$\widehat{\mathrm{p}}_n(x) = \frac{|B_l|}{n} \times \frac{1}{\text{len}\,(B_l)} = \frac{M}{n} \sum_{i=1}^{n} \mathbb{I}[\mathbf{X}_i \in B_l]$$
5. Now we study the bias of the histogram density estimator (Drive the following bounds.)
$$\text{Bias}\,(\widehat{\mathrm{p}}_n(x)) \leq \frac{L}{M}$$
$$\text{var}[\widehat{\mathrm{p}}_n(x)] = M\frac{\mathrm{p}(x^*)}{n} + \frac{(\mathrm{p}(x^*))^2}{n}$$
$$\text{MSE}\,(\widehat{\mathrm{p}}_n(x)) \leq \frac{L}{M} + M\frac{\mathrm{p}(x^*)}{n} + \frac{(\mathrm{p}(x^*))^2}{n}$$

1. To balance the bias and variance, we choose $M$ that minimizes the MSE, which leads to

$$M_{opt} = \left( \frac{n \times L^2}{\mathrm{p}(x^*)} \right)$$



# Kernel density estimator

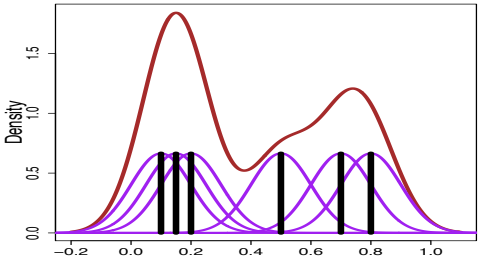
1. The KDE is a function of

$$\widehat{\mathrm{p}}_n(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{\mathbf{X}_i - x}{h}\right)$$

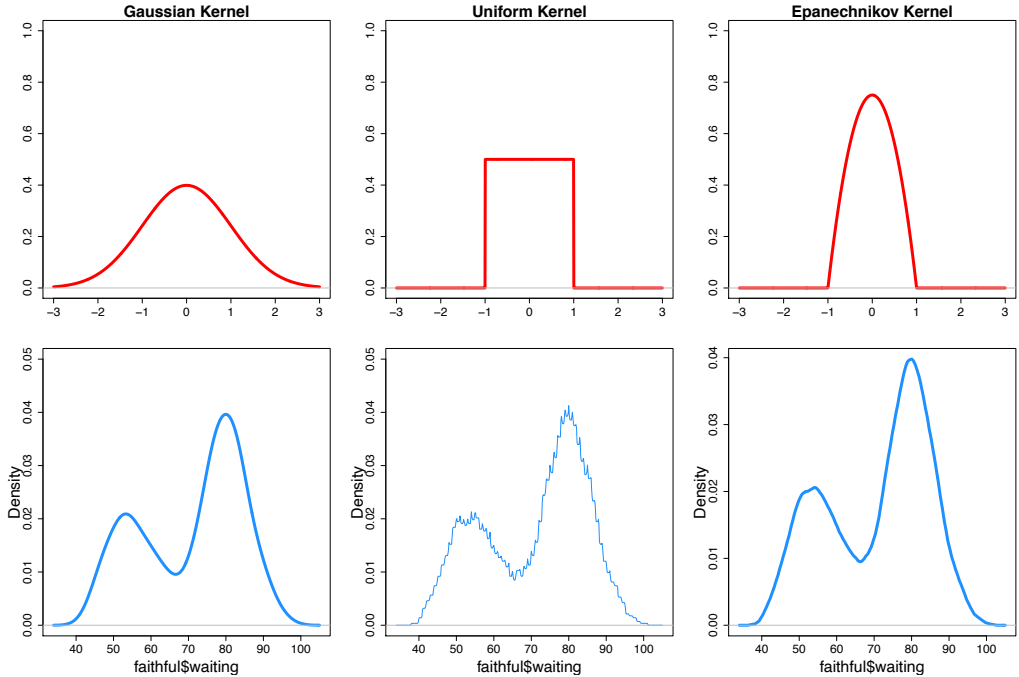
2. where $K(x)$ is called the kernel function that is generally a smooth, symmetric function such as a Gaussian where

   - $K(x)$ is symmetric.
   - $\int K(x)dx = 1$
   - $\lim_{|x|\to\infty} K(x) = 0$

3. $h > 0$ is called the smoothing bandwidth that controls the amount of smoothing.



# Kernel density estimator



1. We first analyze the bias. The bias of KDE is

$$\text{Bias}\left(\widehat{\mathrm{p}}_n(x_0)\right) = \frac{1}{2}h^2\,\mathrm{p}''(x_0)\mu_K + o(h^2) \qquad \mu_K = \int y^2 K(y)dy$$

2. This means that when we allow $h \to 0$, the bias is shrinking at a rate $O(h^2)$.

3. The upper bound of variance of KDE is

$$\text{var}[\widehat{\mathrm{p}}_n(x_0)] = \frac{1}{nh}\,\mathrm{p}(x_0)\sigma_K^2 + o\left(\frac{1}{nh}\right) \qquad \sigma_K^2 = \int K^2(y)dy$$

4. Now putting both bias and variance together, we obtain the MSE of the KDE:

$$\text{MSE}\left(\widehat{\mathrm{p}}_n(x_0)\right) = O(h^4) + O\left(\frac{1}{nh}\right)$$

5. The optimal bandwidth equals to

$$h_{opt} = C_1 n^{-\frac{1}{5}}$$

6. And this choice of smoothing bandwidth leads to a MSE at rate

$$\text{MSE}\left(\widehat{\mathrm{p}}_n(x_0)\right) = O\left(n^{-\frac{1}{5}}\right)$$

# Structured density estimation approach

# The number of parameters of density estimators

1. Let $\mathbf{X} = \{x_1, \ldots, x_d\}$ be an $d-$dimensional random variable where $x_i \in \{0, 1\}$.
2. How many parameters do we need to estimate the density function?

| Sample | $x_d$ | $x_{d-1}$ | $\ldots$ | $x_2$ | $x_1$ |
|---|---|---|---|---|---|
| 1 | 0 | 0 | $\ldots$ | 0 | 0 |
| 2 | 0 | 0 | $\ldots$ | 0 | 1 |
| 3 | 0 | 0 | $\ldots$ | 1 | 0 |
| 4 | 0 | 0 | $\ldots$ | 1 | 1 |
| | | | $\vdots$ | | |
| $2^d$ | 1 | 1 | $\ldots$ | 1 | 1 |

3. How can we decrease the number of parameters?

# How to find the structure of density functions?

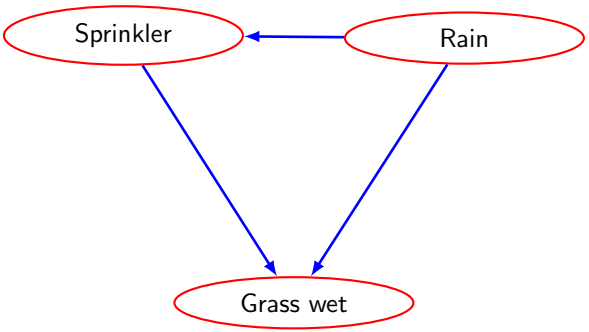
1. One way is to use probabilistic graphical models. A (probabilistic) graphical model defines a family of probability distributions over a set of random variables, by means of a graph.
2. These models offer several useful properties:
   - They provide a simple way to visualize the structure of a probabilistic model and can be used to design and motivate new models.
   - Insights into the properties of the model, including conditional independence properties, can be obtained by inspection of the graph.
   - Complex computations, required to perform inference and learning in sophisticated models, can be expressed in terms of graphical manipulations, in which underlying mathematical expressions are carried along implicitly.
3. A graph $G = (V, E)$ comprises nodes (vertices) $V$ connected by links (edges or arcs) $E$.
   - Each node represents a random variable (or group of random variables).
   - Each link express probabilistic relationships between these variables.
   - The graph captures joint distribution over random variables and can be decomposed into a product of factors each depending only on a subset of the variables.

# Probabilistic graphical model

1. Some types of probabilistic graphical models:
   - Bayesian networks,
   - Markov random fields,
   - Factor graphs
2. Important problems probabilistic graphical models:
   - Structure learning,
     - Constraint-based approach
     - Score-based approach
     - Hybrid-approach
   - Parameter learning
   - Probabilistic inference : Compute marginal probabilities $\mathrm{p}(x \mid E)$

# Structured density function

| Rain | Sprinkler T | Sprinkler F |
|---|---|---|
| F | 0.4 | 0.6 |
| T | 0.01 | 0.99 |




| Sprinkler T | Sprinkler F |
|---|---|
| 0.2 | 0.8 |

| Sprinkler rain | | Grass wet T | Grass wet F |
|---|---|---|---|
| F | F | 0.4 | 0.6 |
| F | T | 0.01 | 0.99 |
| T | F | 0.01 | 0.99 |
| T | T | 0.01 | 0.99 |

# Structured density estimation approach

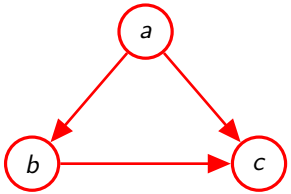
## Bayesian networks

1. Let $\mathrm{p}(a, b, c)$ be joint distribution over three variables $a$, $b$, and $c$.

2. By application of the product rule of probability, we can write the joint distribution as

$$\mathrm{p}(a, b, c) = \mathrm{p}(c \mid a, b)\, \mathrm{p}(a, b)$$
$$\mathrm{p}(a, b, c) = \mathrm{p}(c \mid a, b)\, \mathrm{p}(b \mid a)\, \mathrm{p}(a)$$

3. This decomposition holds for any choice of the joint distribution.

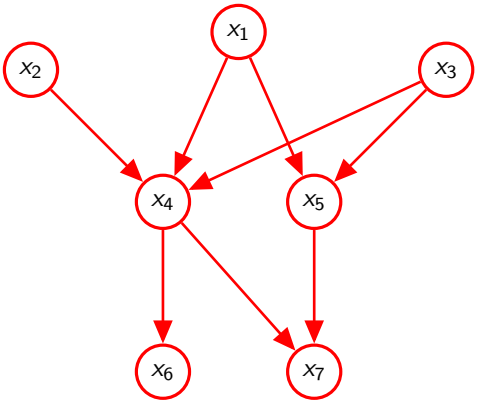



4. An interesting point: $\mathrm{p}(a, b, c)$ symmetrical with respect to $a$, $b$, and $c$, whereas $\mathrm{p}(c \mid a, b)\, \mathrm{p}(b \mid a)\, \mathrm{p}(a)$ is not.

5. Generalization to $K$ variables:

$$\mathrm{p}(x_1, \ldots, x_K) = \mathrm{p}(x_K \mid x_1, \ldots, x_{K-1}) \ldots \mathrm{p}(x_2 \mid x_1)$$

1. Consider the following Bayesian networks




2. The joint distribution of all $x_1, \ldots, x_7$ variables is

$$\mathrm{p}(x_1, \ldots, x_7) = \mathrm{p}(x_1)\,\mathrm{p}(x_2)\,\mathrm{p}(x_3)\,\mathrm{p}(x_4 \mid x_1, x_2, x_3)\,\mathrm{p}(x_5 \mid x_1, x_3)\,\mathrm{p}(x_6 \mid x_4)\,\mathrm{p}(x_7 \mid x_4, x_5).$$

3. For a graph with $K$ nodes, the joint distribution is

$$\mathrm{p}(x_1, \ldots, x_K) = \prod_{k=1}^{K} \mathrm{p}(x_k \mid \mathrm{pa}_k).$$

# Conditional independence

1. An important concept for probability distributions over multiple variables is **conditional independence**.

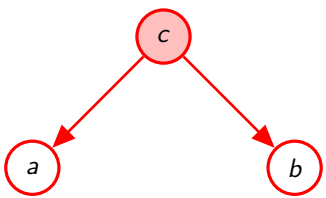
2. For three variables $a, b, c$, and suppose $\mathrm{p}(a \mid b, c)$ does not depend on the value of $b$.

$$\mathrm{p}(a \mid b, c) = \mathrm{p}(a \mid c)$$

3. We say that $a$ is conditionally independent of $b$ given $c$.

$$\begin{aligned} \mathrm{p}(a, b \mid c) &= \mathrm{p}(a \mid b, c)\,\mathrm{p}(b \mid c) \\ &= \mathrm{p}(a \mid c)\,\mathrm{p}(b \mid c). \end{aligned}$$

4. We sometimes use a shorthand notation for conditional independence $a \perp\!\!\!\perp b \mid c$ as.



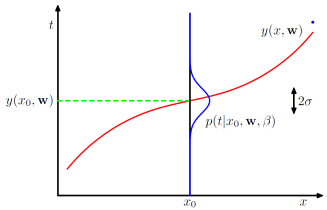

$$\begin{aligned} \mathrm{p}(a, b, c) &= \mathrm{p}(a \mid c)\,\mathrm{p}(b \mid c)\,\mathrm{p}(c) \\ \mathrm{p}(a, b \mid c) &= \frac{\mathrm{p}(a, b, c)}{\mathrm{p}(c)} \\ &= \mathrm{p}(a \mid c)\,\mathrm{p}(b \mid c). \end{aligned}$$

So we obtain the conditional independence property $a \perp\!\!\!\perp b \mid c$.

# Polynomial regression

1. Consider the regression model in which
   - set of $n$ iid observations $\mathbf{x} = (x_1, \ldots, x_n)$ and
   - with corresponding target values $\mathbf{t} = (t_1, \ldots, t_n)$,
   - where $t_k$ is actual value plus a Gaussian noise value with precision $\beta$.
2. Let $y(x, \mathbf{w})$ be the predicted function and the goal is to make predictions of target variable $t$ for new input $x$.

$$\mathrm{p}(t \mid x, \mathbf{w}, \beta) = \mathcal{N}\big(t \mid y(x, \mathbf{w}), \beta^{-1}\big)$$




3. Using training data $\{\mathbf{x}, \mathbf{t}\}$, we can determine $\mathbf{w}$ and $\beta$ by MLE.

$$\mathrm{p}(\mathbf{t} \mid \mathbf{x}, \mathbf{w}, \beta) = \prod_{k=1}^{K} \mathcal{N}\big(t_k \mid y(x_k, \mathbf{w}), \beta^{-1}\big)$$

1. Let introduce a prior distribution over parameters $\mathbf{w}$ as

$$\mathrm{p}(\mathbf{w} \mid \alpha) = \mathcal{N}\left(\mathbf{w} \mid 0, \alpha^{-1}\mathbf{I}\right)$$

   where $\alpha$ is the precision of the distribution.

2. The posterior distribution for $\mathbf{w}$ can be estimated using MAP as

$$\mathrm{p}(\mathbf{w} \mid \mathbf{x}, \mathbf{t}, \alpha, \beta) \propto \mathrm{p}(\mathbf{t} \mid \mathbf{x}, \mathbf{w}, \alpha, \beta)\, \mathrm{p}(\mathbf{w} \mid \alpha).$$

3. In Bayesian regression model, for a new point $x$, we need to predict value $t$ as

$$\mathrm{p}(t \mid x, \mathbf{x}, \mathbf{t}) = \int \mathrm{p}(t \mid x, \mathbf{w})\, \mathrm{p}(\mathbf{w} \mid \mathbf{x}, \mathbf{t}) dw.$$

   where we assume that parameters $\alpha$ and $\beta$ are fixed and known in advance.

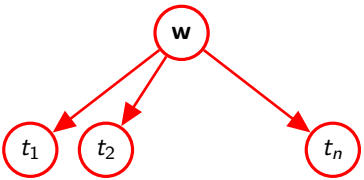
4. The random variables are parameters $\mathbf{w}$ and observed data $\mathbf{t} = (t_1, \ldots, t_n)$.

5. In addition, this model contains input data $\mathbf{x} = (x_1, \ldots, x_n)$ and parameters $\alpha$ and $\beta$.

1. By focusing only on random variables, the joint distribution is

$$\mathrm{p}(\mathbf{t}, \mathbf{w}) = \mathrm{p}(\mathbf{w}) \prod_{k=1}^{n} \mathrm{p}(t_n \mid \mathbf{w}).$$

2. The conditional distributions $\mathrm{p}(t_n \mid \mathbf{w})$ (for $n = 1, \ldots, n$) is




3. The random variables in this model are t
   - the vector of coefficients $\mathbf{w}$
   - the observed data $\mathbf{t} = (t_1, \ldots, t_n)$.
4. Other parameters are not random variables
   - the input data $\mathbf{x} = (x_1, \ldots, x_n)$
   - the noise precision $\beta$ and the hyper-parameter $\alpha$.

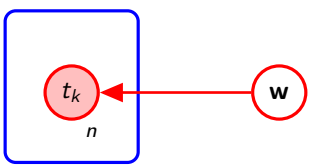
1. The joint distribution $\mathrm{p}(\mathbf{t}, \mathbf{w})$ is

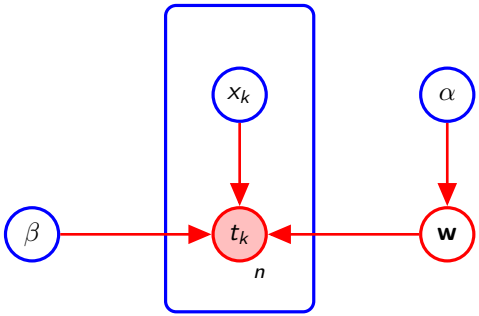
$$\mathrm{p}(\mathbf{t}, \mathbf{w}) = \mathrm{p}(\mathbf{w}) \prod_{k=1}^{n} \mathrm{p}(t_n \mid \mathbf{w}).$$

2. Sometimes it is helpful to make the parameters of a model, as well as its random variables, explicit.

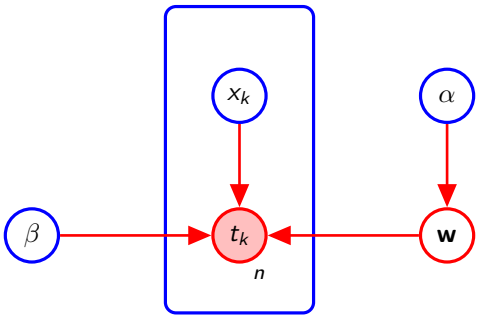
$$\mathrm{p}(\mathbf{t}, \mathbf{w} \mid \mathbf{x}, \alpha, \beta) = \mathrm{p}(\mathbf{w} \mid \alpha) \prod_{k=1}^{n} \mathrm{p}(t_n \mid \mathbf{w}, x_n, \beta).$$

3. We can represent it in graphical notations.



1. Having observed values $\{t_k\}$ we can evaluate the posterior distribution of **w**




$$\mathrm{p}(\mathbf{w} \mid \mathbf{t}) \propto \mathrm{p}(\mathbf{w}) \prod_{k=1}^{n} \mathrm{p}(t_k \mid \mathbf{w})$$

2. Let new input $\widehat{x}$ is given and we wish to find the corresponding probability distribution for $\widehat{t}$ conditioned on the observed data.

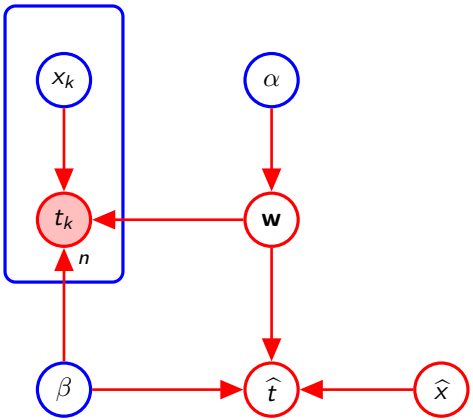
3. The joint distribution of **all random variables** conditioned on **deterministic parameters** is

$$\mathrm{p}(\widehat{t}, \mathbf{t}, \mathbf{w} \mid \widehat{x}, \mathbf{x}, \alpha, \beta) = \left[ \prod_{k=1}^{n} \mathrm{p}(t_k \mid x_k, \mathbf{w}, \beta) \right] \mathrm{p}(\mathbf{w} \mid \alpha)\, \mathrm{p}(t_k \mid \widehat{x}, \mathbf{w}, \beta)$$

1. The joint distribution of all random variables conditioned on deterministic parameters is

$$\mathrm{p}(\widehat{t}, \mathbf{t}, \mathbf{w} \mid \widehat{x}, \mathbf{x}, \alpha, \beta) = \left[\prod_{k=1}^{n} \mathrm{p}(t_k \mid x_k, \mathbf{w}, \beta)\right] \mathrm{p}(\mathbf{w} \mid \alpha)\, \mathrm{p}(t_k \mid \widehat{x}, \mathbf{w}, \beta)$$

2. The corresponding graphical model is




$$\mathrm{p}(\widehat{t} \mid \widehat{x}, \mathbf{x}, \alpha, \beta) = \int \mathrm{p}(\widehat{t}, \mathbf{t}, \mathbf{w} \mid \widehat{x}, \mathbf{x}, \alpha, \beta) dw$$

3. We are implicitly setting the random variables in **t** to the specific values observed in the data set.

# Generative models

1. There are many situations in which we wish to draw samples from a given probability distribution.

2. Let $\mathrm{p}(x_1, \ldots, x_d)$ be the joint distribution over $d$ variables.

3. The goal is to draw a sample $(x_1, \ldots . x_d)$ from the joint distribution.

4. To do this (suppose that the variables have been ordered such that there are no links from any node to any lower numbered node),

   4.1 Start with the lowest-numbered node and draw a sample from the distribution $\mathrm{p}(x_1)$, which we call $\widehat{x}_1$.

   4.2 For a node $x_n$, draw a sample from the conditional distribution $\mathrm{p}(x_n \mid \mathrm{pa}_n)$

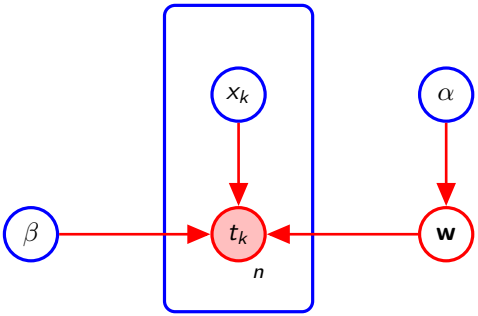
   4.3 Continue until the last variable is being sampled.

5. To obtain a sample from some marginal distribution corresponding to a subset of the variables:

   5.1 we simply take the sampled values for the required nodes and

   5.2 ignore the sampled values for the remaining nodes.

1. Consider the following graphical model: **Is it generative?**



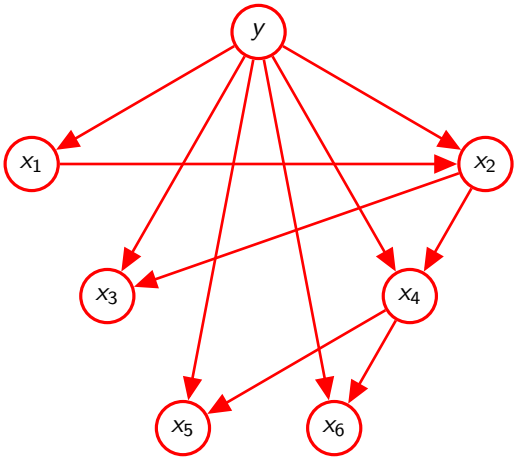

2. This model is not generative because **there is no probability distribution associated with the input variable** $x$.

3. So it is not possible to generate synthetic data points from this model.

4. Can we make the above model generative?

5. We could make it generative by introducing a suitable prior distribution $\mathrm{p}(x)$, at the expense of a more complex model.

1. Consider the following graphical model.




2. How do you compute $\mathrm{p}(y \mid x_5)$?

3. The joint distribution $\mathrm{p}(y, x_1, x_2, x_3, x_4, x_5, x_6)$ equals to

$$\mathrm{p}(y, x_1, x_2, x_3, x_4, x_5, x_6) = \mathrm{p}(y)\,\mathrm{p}(x_1 \mid y)\,\mathrm{p}(x_2 \mid x_1, y)\,\mathrm{p}(x_3 \mid x_2, y)$$
$$\mathrm{p}(x_4 \mid x_2, y)\,\mathrm{p}(x_5 \mid x_4, y)\,\mathrm{p}(x_6 \mid x_4, y)$$

$$
\begin{aligned}
\mathrm{p}(y \mid x_5) &\propto \sum_{x_1}\sum_{x_2}\sum_{x_3}\sum_{x_4}\sum_{x_6} \mathrm{p}(y)\,\mathrm{p}(x_1 \mid y)\,\mathrm{p}(x_2 \mid x_1, y)\,\mathrm{p}(x_3 \mid x_2, y)\,\mathrm{p}(x_4 \mid x_2, y)\,\mathrm{p}(x_5 \mid x_4, y)\,\mathrm{p}(x_6 \mid x_4, y) \\
&= \sum_{x_1}\sum_{x_2}\sum_{x_4} \mathrm{p}(y)\,\mathrm{p}(x_1 \mid y)\,\mathrm{p}(x_2 \mid x_1, y)\,\mathrm{p}(x_4 \mid x_2, y)\,\mathrm{p}(x_5 \mid x_4, y) \underbrace{\sum_{x_3} \mathrm{p}(x_3 \mid x_2, y) \underbrace{\sum_{x_6} \mathrm{p}(x_6 \mid x_4, y)}_{=1}}_{=1} \\
&= \mathrm{p}(y) \sum_{x_1} \mathrm{p}(x_1 \mid y) \sum_{x_2} \mathrm{p}(x_2 \mid x_1, y) \underbrace{\sum_{x_4} \mathrm{p}(x_4 \mid x_2, y)\,\mathrm{p}(x_5 \mid x_4, y)}_{m_4(x_2)} \\
&= \mathrm{p}(y) \sum_{x_1} \mathrm{p}(x_1 \mid y) \underbrace{\sum_{x_2} \mathrm{p}(x_2 \mid x_1, y) m_4(x_2)}_{m_2(x_1)} \\
&= \mathrm{p}(y) \underbrace{\sum_{x_1} \mathrm{p}(x_1 \mid y) m_2(x_1)}_{m_1} = \mathrm{p}(y) m_1.
\end{aligned}
$$

**The order of summations is important.**

**Consider ordering $\mathbf{X}_4, \mathbf{X}_1, \mathbf{X}_2, \mathbf{Y}, \mathbf{X}_3$.**

$$\begin{aligned}
\mathrm{p}(x_3 \mid x_5) &\propto \sum_y \mathrm{p}(y) \sum_{x_2} \mathrm{p}(x_3 \mid x_2, y) \sum_{x_1} \mathrm{p}(x_2 \mid x_1, y)\, \mathrm{p}(x_1 \mid y) \underbrace{\sum_{x_4} \mathrm{p}(x_4 \mid x_2, y)\, \mathrm{p}(x_5 \mid x_4, y)}_{m_4(x_2, y)} \\
&= \sum_y \mathrm{p}(y) \sum_{x_2} \mathrm{p}(x_3 \mid x_2, y) \underbrace{\sum_{x_1} \mathrm{p}(x_2 \mid x_1, y)\, \mathrm{p}(x_1 \mid y) m_4(x_2, y)}_{m_1(x_2, y)} \\
&= \sum_y \mathrm{p}(y) \underbrace{\sum_{x_2} \mathrm{p}(x_3 \mid x_2, y) m_1(x_2, y)}_{m_2(y)} \\
&= \underbrace{\sum_y \mathrm{p}(y) m_2(y)}_{m_y}.
\end{aligned}$$

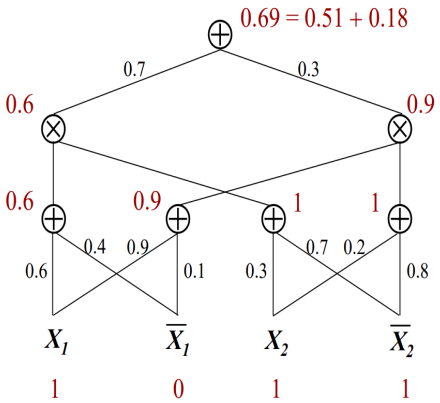
# Structured density estimation approach

## Sum Product Networks

# Sum Product Networks (SPN)

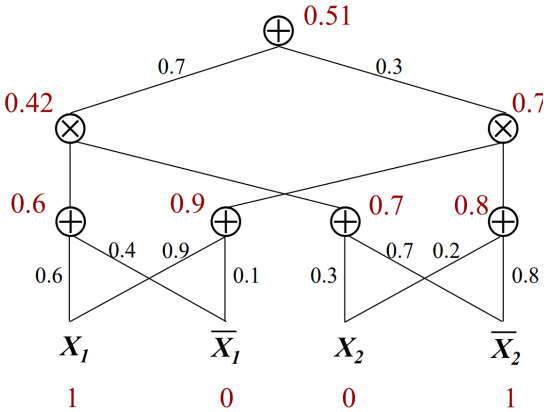
An SPN is a rooted directed acyclic graph that efficiently computes the marginals and modes of a probabilistic graphical model (PGM) (Poon and Domingos 2011).

1. A SPN has two types of internal nodes: sum nodes and product nodes.
2. The leaves of A SPN are $x_1, \ldots, x_n$ and $\bar{x}_1, \ldots, \bar{x}_n$.
3. Each edge $(i, j)$ emanating from sum node $i$ has a weight $w_{ij} \geq 0$.
4. The value of a product node is the product of the value of its children.
5. The value of a sum node $i$ is $\sum_{j \in Ch(i)} w_{ij} v_j$, where $Ch(j)$ are the children of node $i$ and $v_j$ is its value.
6. The value of a SPN is the value of the root after a bottom up evaluation.
7. Layers of sum and product nodes usually alternate.

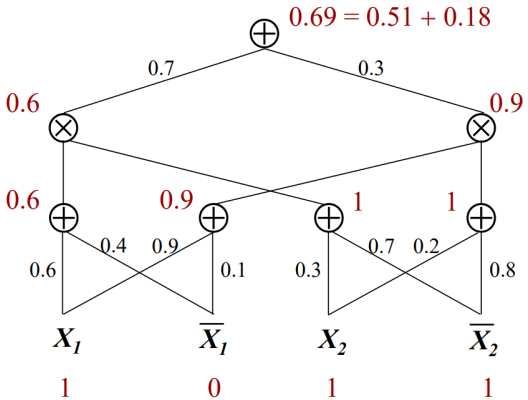



**What is the output of the above network?**

SPN represents a joint distribution over a set of random variables. What is value of $p(x_1 = 1, x_2 = 0)$?



SPN represents a joint distribution over a set of random variables. What is value of $p(x_1 = 1)$?

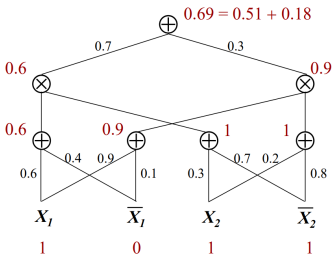


SPN represents a joint distribution over a set of random variables.

1. As an example consider

$$\mathrm{p}(x_1 = \mathit{true} \mid x_2 = \mathit{false}) = \frac{\mathrm{p}(x_1 = \mathit{true}, x_2 = \mathit{false})}{\mathrm{p}(x_2 = \mathit{false})}$$

2. Hence any inference query can be answered in two bottom-up passes of the network.



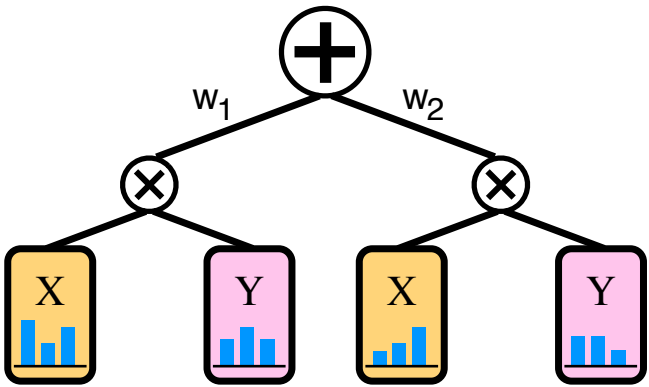

The inference algorithm has linear time complexity with respect to the number of nodes in the network.

A valid SPN encodes a hierarchical mixture distribution.

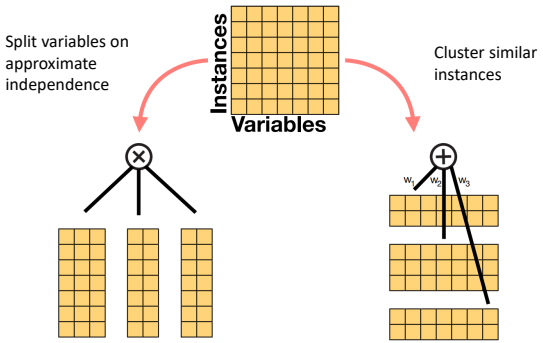
1. Sum nodes: hidden variables (mixture).
2. Product nodes: factorization (independence).

**We must specify the structure of SPN (structure Estimation or structure learning).**

1. What is SPN for univariate distribution? → A univariate distribution is an SPN.
2. What is SPN for product of disjoint random variables? → A product of SPNs over disjoint variables is an SPN.
3. What is SPN for a mixture model? → A weighted sum of SPNs over the same variables is an SPN.



1. In a structure learning, one alternates between
   - Data Clustering: sum nodes
   - Variable partitioning: product nodes




2. Some others use SVD decomposition (Adel, Balduzzi, and Ghodsi 2015).

# SPN parameter learning

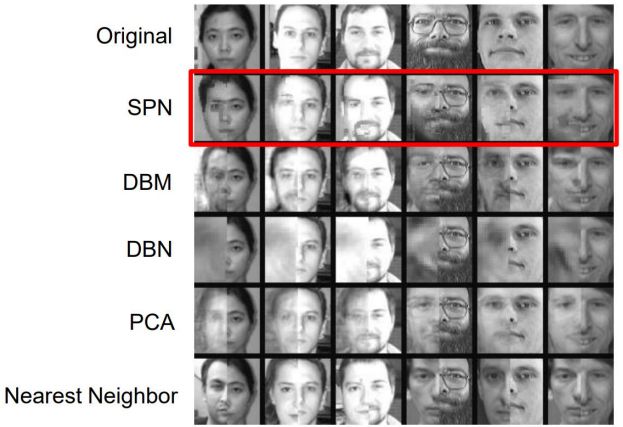
1. Initialize the SPN using a dense valid SPN.
2. Learn the SPN weights using gradient descent or EM.
3. Add some penalty to the weights so that they tend to be zero.
4. Prune edges with zero weights at convergence.

**Algorithm 1** LearnSPN

**Input:** Set $D$ of instances over variables $X$.
**Output:** An SPN with learned structure and parameters.
$S \leftarrow \text{GenerateDenseSPN}(X)$
$\text{InitializeWeights}(S)$
**repeat**
  **for all** $d \in D$ **do**
    $\text{UpdateWeights}(S, \text{Inference}(S, d))$
  **end for**
**until** convergence
$S \leftarrow \text{PruneZeroWeights}(S)$
**return** $S$

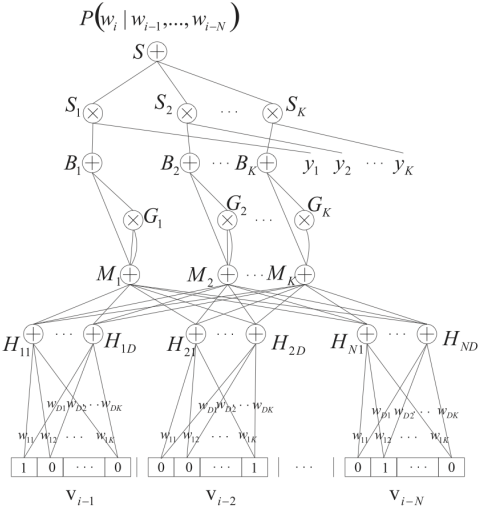
# SPN applications: image completion

1. Main evaluation: Caltech-101
   - 101 categories: such as faces, cars, elephants
   - Each category: 30 – 800 images
2. Each category: last third for test
3. Test images: unseen objects



# SPN applications: language modeling

1. Fixed structure SPN encoding the conditional probability $p(w_i|w_{i-1}\ldots,w_{i-N})$ as an $N$th order language model (Cheng et al. 2014).



1. Perplexity scores (PPL) of different language models

| Model | Individual $PPL$ | +KN5 |
|---|---|---|
| TrainingSetFrequency | 528.4 | |
| KN5 [3] | 141.2 | |
| Log-bilinear model [4] | 144.5 | 115.2 |
| Feedforward neural network [5] | 140.2 | 116.7 |
| Syntactical neural network [8] | 131.3 | 110.0 |
| RNN [6] | 124.7 | 105.7 |
| LDA-augmented RNN [9] | 113.7 | 98.3 |
| **SPN-3** | **104.2** | **82.0** |
| **SPN-4** | **107.6** | **82.4** |
| **SPN-4'** | **100.0** | **80.6** |

# SPN applications: other applications

1. Image completion
2. Image classification
3. Activity recognition
4. Click-through logs
5. Nucleic acid sequences
6. Collaborative filtering

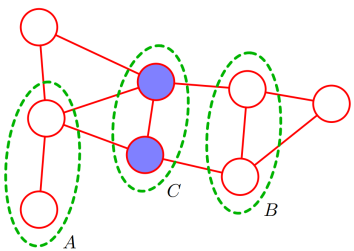
For more information, please read (Paris, Sanchez-Cauce, and Diez 2020).

# Structured density estimation approach

## Markov Random Fields

# Markov random field

1. A Markov random field, also known as a Markov network or an undirected graphical model, has
   - a set of nodes each of which corresponds to a variable or group of variables and
   - a set of links each of which connects a pair of nodes.
2. The links are undirected, that is they do not carry arrows.




3. In above undirected graph every path from any node in set $A$ to any node in set $B$ passes through at least one node in set $C$. Hence,

$$A \perp\!\!\!\perp B \mid C$$

# Factorization properties

1. We need a factorization rule for undirected graphs that correspond to the conditional independence test.

2. Consider two nodes $x_i$ and $x_j$ that are not connected by a link, then these variables must be conditionally independent given all other nodes in the graph.

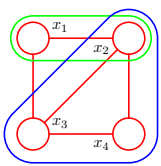
3. This conditional independence property can be expressed as

$$\mathrm{p}(x_i, x_j \mid \mathbf{x}_{\setminus\{i,j\}}) = \mathrm{p}(x_i \mid \mathbf{x}_{\setminus\{i,j\}})\, \mathrm{p}(x_j \mid \mathbf{x}_{\setminus\{i,j\}})$$

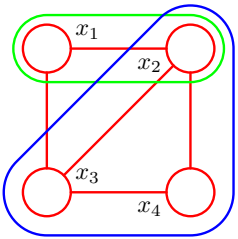
4. The factorization of the joint distribution must be such that $x_i$ and $x_j$ do not appear in the same factor in order for the conditional independence property to hold for all possible distributions belonging to the graph.

5. This leads us to consider a graphical concept called a **clique**.

6. A **maximal clique** is a clique such that it is not possible to include any other nodes from the graph in the set without it ceasing to be a clique.



1. Consider the following graph




Two-nodes cliques

- $\{x_1, x_2\}$
- $\{x_2, x_3\}$
- $\{x_3, x_4\}$
- $\{x_4, x_2\}$
- $\{x_1, x_3\}$

Two maximal cliques

- $\{x_1, x_2, x_3\}$
- $\{x_2, x_3, x_4\}$

# Factorization properties

1. We can define the factors in the decomposition of the joint distribution to be functions of the variables in the cliques.

2. We can consider functions of the maximal cliques, because other cliques must be subsets of maximal cliques.

3. If $\{x_1, x_2, x_3\}$ is a maximal clique and we define an arbitrary function over this clique, then including another factor defined over a subset of these variables would be redundant.

4. Let us denote a clique by $C$ and the set of variables in that clique by $\mathbf{x}_C$.

5. The joint distribution is written as a product of **potential functions** $\psi(\mathbf{x}_C) > 0$ over the maximal cliques of the graph.

$$\mathrm{p}(\mathbf{x}) = \frac{1}{Z} \prod_C \psi(\mathbf{x}_C)$$

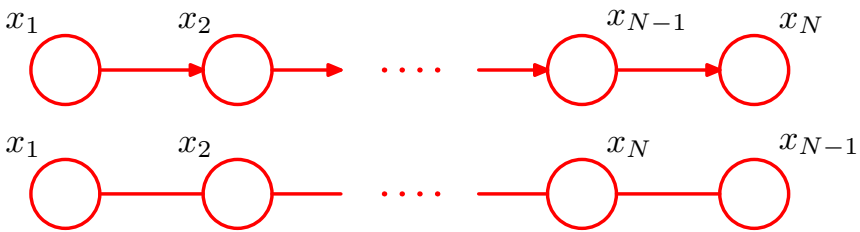
6. The quantity $Z$, called the **partition function**, is a normalization constant given by (for discrete variables)

$$Z = \sum_{\mathbf{x}} \prod_C \psi(\mathbf{x}_C)$$

to ensure the distribution $\mathrm{p}(\mathbf{x})$ is correctly normalized.

# Relation between directed and undirected graphs

1. Consider the following graphs




2. For the directed graph, we have

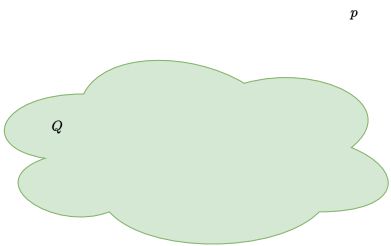
$$\mathrm{p}(\mathbf{x}) = \mathrm{p}(x_1)\,\mathrm{p}(x_2 \mid x_1)\,\mathrm{p}(x_3 \mid x_2)\ldots\,\mathrm{p}(x_N \mid x_{N-1})$$

3. For the undirected graph, we have

$$\mathrm{p}(\mathbf{x}) = \frac{1}{Z}\psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\ldots\psi_{N-1,N}(x_{N-1}, x_N)$$

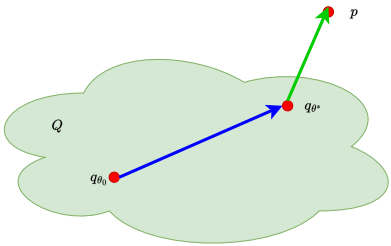
# Variational inference

1. Let $D$ be the data set.
2. Let $\mathrm{p}(\mathbf{x}) \triangleq \mathrm{p}(\mathbf{x} \mid D)$ be the true but intractable distribution.
3. Let $\mathrm{q}(\mathbf{x})$ be some approximation chosen from some tractable family $Q$ such as multi-variate Gaussian.
4. We assume $\mathrm{q}(\mathbf{x})$ has some free parameters which we want to optimize so as to make q "similar to" p.




5. An obvious cost function is to try minimize the difference between $\mathrm{q}_\theta$ and $p$.

1. An obvious cost function is to try minimize the KL divergence between $q_\theta$ and $p$.




$$\begin{aligned} D_{KL}(p \parallel q) &= \sum_x p(x) \log \frac{p(x)}{q(x)} \\ &= \mathbb{E}_p\left[\log \frac{p(x)}{q(x)}\right] \end{aligned}$$

2. This is hard to compute, since $\mathbb{E}_p[.]$ is assumed to be intractable.

1. A natural alternative is the reverse KL divergence.

$$\begin{aligned} D_{KL}(q \mid\mid p) &= \sum_{x} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} \\ &= \mathbb{E}_q \left[ \log \frac{q(\mathbf{x})}{p(\mathbf{x})} \right] \end{aligned}$$

2. The main advantage of the objective function is that computing $\mathbb{E}_q[.]$ is tractable.

3. Equation $\mathbb{E}_q \left[ \log \frac{q(\mathbf{x})}{p(\mathbf{x})} \right]$ **is not tractable** because evaluating $p(\mathbf{x})$ point-wise is hard since it requires $Z = \int_{\mathbf{x}} p(\mathbf{x})$.

4. By using un-normalized distribution $\widetilde{p}(\mathbf{x}) \triangleq p(\mathbf{x} \mid D) = p(\mathbf{x})Z$, it becomes tractable to compute.

5. Then, we define the objective function as

$$J(q) = D_{KL}(q \mid\mid \widetilde{p})$$

1. Then, we define the objective function as

$$J(\mathrm{q}) = \mathrm{D}_{KL}(\mathrm{q} \mid\mid \widetilde{\mathrm{p}})$$

2. The above KL was abused because $\widetilde{\mathrm{p}}$ is not a valid distribution.

$$\begin{aligned} J(\mathrm{q}) &= \sum_{\mathbf{x}} \mathrm{q}(\mathbf{x}) \log \frac{\mathrm{q}(\mathbf{x})}{\widetilde{\mathrm{p}}(\mathbf{x})} \\ &= \sum_{\mathbf{x}} \mathrm{q}(\mathbf{x}) \log \frac{\mathrm{q}(\mathbf{x})}{Z\,\mathrm{p}(\mathbf{x})} \\ &= \sum_{\mathbf{x}} \mathrm{q}(\mathbf{x}) \log \frac{\mathrm{q}(\mathbf{x})}{\mathrm{p}(\mathbf{x})} - \mathit{logZ} \\ &= \mathrm{D}_{KL}(\mathrm{q} \mid\mid \mathrm{p}) - \log Z \end{aligned}$$

3. Since $Z$ is a constant, by minimizing $J(\mathrm{q})$, we will force q to become close to p.

1. Since KL divergence is always non-negative, $J(\mathrm{q})$ is an upper bound on $\log Z$.

$$\begin{aligned} J(\mathrm{q}) &= \mathrm{D}_{KL}(\mathrm{q} \mid\mid \mathrm{p}) - logZ \\ &= -logZ \end{aligned}$$

2. The value of $\log Z$ is called evidence lower bound (ELBO).

3. Alternatively, we can try to maximize the following quantity, called energy functional.

$$\begin{aligned} L(\mathrm{q}) &= -J(\mathrm{q}) \\ &= -\mathrm{D}_{KL}(\mathrm{q} \mid\mid \mathrm{p}) + logZ \\ &\leq \log Z. \end{aligned}$$

4. This is a lower bound on the log likelihood of the data.

# Alternative interpretation of $J(\mathrm{q})$

1. The objective function $J(\mathrm{q})$ can be written as

$$\begin{aligned} J(\mathrm{q}) &= \mathbb{E}_{\mathrm{q}}[\log \mathrm{q}(\mathbf{x})] + \mathbb{E}_{\mathrm{q}}[\log \widetilde{\mathrm{p}}(\mathbf{x})] \\ &= H(\mathrm{q}) + \mathbb{E}_{\mathrm{q}}[E(\mathbf{x})] \end{aligned}$$

   where $E(\mathbf{x}) = -\log \widetilde{\mathrm{p}}(\mathbf{x})$ is **energy**.

2. Thus, $J(\mathrm{q})$ is expected energy minus Entropy of the system.

3. In statistical physics, $J(\mathrm{q})$ is called the variational free energy or the Helmholtz free energy.

# Computing KL divergence

1. Let p and q be two $k$-dimensional Gaussian distribution.

$$\mathrm{p}(\mathbf{x}) = \frac{1}{(2\pi)^{k/2}|\Sigma_p|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_p)^T\Sigma_p^{-1}(\mathbf{x}-\mu_p)\right)$$
$$\mathrm{q}(\mathbf{x}) = \frac{1}{(2\pi)^{k/2}|\Sigma_q|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_q)^T\Sigma_q^{-1}(\mathbf{x}-\mu_q)\right)$$

2. Then, KL divergence can be written as

$$\begin{aligned}
D_{KL}(\mathrm{p} \mid\mid \mathrm{q}) &= \mathbb{E}_{\mathrm{p}}[\log \mathrm{p} - \log \mathrm{q}] \\
&= \mathbb{E}_{\mathrm{p}}\left[\frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_p|} - \frac{1}{2}(\mathbf{x}-\mu_p)^T\Sigma_p^{-1}(\mathbf{x}-\mu_p) + \frac{1}{2}(\mathbf{x}-\mu_q)^T\Sigma_q^{-1}(\mathbf{x}-\mu_q)\right] \\
&= \frac{1}{2}\mathbb{E}_{\mathrm{p}}\left[\log\frac{|\Sigma_q|}{|\Sigma_p|}\right] - \frac{1}{2}\mathbb{E}_{\mathrm{p}}\left[(\mathbf{x}-\mu_p)^T\Sigma_p^{-1}(\mathbf{x}-\mu_p)\right] + \frac{1}{2}\mathbb{E}_{\mathrm{p}}\left[(\mathbf{x}-\mu_q)^T\Sigma_q^{-1}(\mathbf{x}-\mu_q)\right] \\
&= \frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma_p|} - \frac{1}{2}\mathbb{E}_{\mathrm{p}}\left[(\mathbf{x}-\mu_p)^T\Sigma_p^{-1}(\mathbf{x}-\mu_p)\right] + \frac{1}{2}\mathbb{E}_{\mathrm{p}}\left[(\mathbf{x}-\mu_q)^T\Sigma_q^{-1}(\mathbf{x}-\mu_q)\right]
\end{aligned}$$

3. Since, $(\mathbf{x}-\mu_p)^T\Sigma_p^{-1}(\mathbf{x}-\mu_p)$ is a scaler, we can write it as $\mathrm{tr}((\mathbf{x}-\mu_p)^T\Sigma_p^{-1}(\mathbf{x}-\mu_p)) = \mathrm{tr}((\mathbf{x}-\mu_p)(\mathbf{x}-\mu_p)^T\Sigma_p^{-1})$.

# Computing KL divergence

1. The expectation and trace can be interchanged to get,

$$= \frac{1}{2}\,\mathrm{tr}\big(\,\mathbb{E}_{\mathrm{p}}\big[(\mathbf{x}-\mu_p)(\mathbf{x}-\mu_p)^T\Sigma_p^{-1}\big]\big)$$
$$= \frac{1}{2}\,\mathrm{tr}\big(\,\mathbb{E}_{\mathrm{p}}\big[(\mathbf{x}-\mu_p)(\mathbf{x}-\mu_p)^T\big]\Sigma_p^{-1}\big)$$

2. We know $\Sigma_p = \mathbb{E}_{\mathrm{p}}\big[(\mathbf{x}-\mu_p)(\mathbf{x}-\mu_p)^T\big]$. Simplifying it to

$$\frac{1}{2}\,\mathrm{tr}\big(\,\mathbb{E}_{\mathrm{p}}\big[(\mathbf{x}-\mu_p)(\mathbf{x}-\mu_p)^T\big]\Sigma_p^{-1}\big) = \frac{1}{2}\,\mathrm{tr}(\Sigma_p\Sigma_p^{-1})$$
$$= \frac{1}{2}\,\mathrm{tr}(I_k) = \frac{k}{2}$$

3. By using matrix cookbook, the third term is also equals to

$$\mathbb{E}_{\mathrm{p}}\big[(\mathbf{x}-\mu_q)^T\Sigma_q^{-1}(\mathbf{x}-\mu_q)\big] = (\mu_p-\mu_q)^T\Sigma_q^{-1}(\mu_p-\mu_q) + \mathrm{tr}(\Sigma_q^{-1}\Sigma_p)$$

4. Combining all this we get,

$$\mathrm{D}_{KL}(\mathrm{p}\;||\;\mathrm{q}) = \frac{1}{2}\left\{\log\frac{|\Sigma_q|}{|\Sigma_p|} - k + (\mu_p-\mu_q)^T\Sigma_q^{-1}(\mu_p-\mu_q) + \mathrm{tr}(\Sigma_q^{-1}\Sigma_p)\right\}$$

5. **What happens if we have not distributions explicitly?**

# Mean field variational inference

1. In mean field variational inference, we assume that the variational family factorizes,

$$\mathrm{q}(x_1, \ldots, x_d) = \prod_{j=1}^{d} \mathrm{p}(x_j)$$

2. The goal is to solve this optimization problem:

$$\min_{q_1, \ldots, q_d} \mathrm{D}_{KL}(q \mid\mid p)$$

3. We optimize over the parameters of each marginal distribution $q_i$.

4. The standard way of performing this optimization problem is via coordinate descent over the $q_j$.

5. Interestingly, the optimization problem for one coordinate has a simple closed form solution.

# Advantages of SPNs

1. We can estimate distribution functions using parametric or non-parametric methods.
2. In probability graphical models, we can find the structure of distribution function.
3. Unlike graphical models, SPNs are tractable over high treewidth models.
4. SPNs are deep architectures with full probabilistic semantics
5. SPNs can incorporate features into an expressive model without requiring approximate inference.

# References

# Reading

1. Chapter 21 of Machine Learning: A Probabilistic Perspective (Murphy 2012).
2. Chapter 10 of Probabilistic Machine Learning: Advanced Topics (Murphy 2023).
3. Chapter 8 of Pattern Recognition and Machine Learning (Bishop 2006).
4. Chapter 11 of Deep Learning: Foundations and Concepts (**bishop2024**).
5. Chapter 7 of All of Statistics: A Concise Course in Statistical Inference (Wasserman 2010).

# References


- Adel, Tameem, David Balduzzi, and Ali Ghodsi (2015). "Learning the Structure of Sum-Product Networks via an SVD-based Algorithm". In: *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 32–41.
- Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag.
- Cheng, Wei-Chen et al. (2014). "Language modeling with sum-product networks". In: *Proceedings of the 15th Annual Conference of the International Speech Communication Association*, pp. 2098–2102.
- Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.
- — (2023). *Probabilistic Machine Learning: Advanced Topics*. The MIT Press.
- Paris, Iago, Raquel Sanchez-Cauce, and Francisco Javier Diez (2020). "Sum-product networks: A survey". In: *CoRR* abs/2004.01167. arXiv: 2004.01167.
- Poon, Hoifung and Pedro M. Domingos (2011). "Sum-Product Networks: A New Deep Architecture". In: *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 337–346.
- Wasserman, Larry (2010). *All of Statistics: A Concise Course in Statistical Inference*. Springer.

# Questions?