Machine Learning Principles and Methods

Lecturer: Joris Mooij
Scribe: Thomas Jongstra & Richard Rozeboom
Lecture #0
December 11, 2013
Updated: February 3, 2015

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

The Big Picture

Probabilistic modeling: specify **likelihood function**: $p(\boldsymbol{X}|\vec{\boldsymbol{\Theta}})$

- $\boldsymbol{X}$: observed data (typically, $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, with N number of samples/observations and D the number of features)
- $\vec{\boldsymbol{\Theta}}$: vector of model parameters (typically, $\vec{\boldsymbol{\Theta}} \in \mathbb{R}^p$, with p the number of parameters / dimensionality of the model)

There are different approaches to learning:

(1) Maximum Likelihood Estimation (frequentist approach)

$$\hat{\boldsymbol{\Theta}}_{ML} = \arg\max_{\vec{\boldsymbol{\Theta}}} p(\boldsymbol{X}|\vec{\boldsymbol{\Theta}})$$

(2a) Maximum a posteriori (MAP) estimation ("penalized ML")

Specify **prior** $p(\vec{\boldsymbol{\Theta}})$ on parameters

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\vec{\boldsymbol{\Theta}}} p(\boldsymbol{X}, \vec{\boldsymbol{\Theta}}) = \arg\max_{\vec{\boldsymbol{\Theta}}} \overbrace{p(\boldsymbol{X}|\vec{\boldsymbol{\Theta}})}^{\text{Likelihood}} \overbrace{p(\vec{\boldsymbol{\Theta}})}^{\text{Prior}}$$

$$= \arg\max_{\vec{\boldsymbol{\Theta}}} \frac{p(\boldsymbol{X}|\vec{\boldsymbol{\Theta}})p(\vec{\boldsymbol{\Theta}})}{p(\boldsymbol{X})} = \arg\max_{\vec{\boldsymbol{\Theta}}} \underbrace{p(\vec{\boldsymbol{\Theta}}|\boldsymbol{X})}_{\text{Posterior}}$$

(2b) Bayesian Approach

Specify **prior** $p(\vec{\boldsymbol{\Theta}})$ on parameters
Calculate $\underbrace{p(\vec{\boldsymbol{\Theta}}|\boldsymbol{X})}_{\text{Posterior}}$ instead of maximizing with respect to $\vec{\boldsymbol{\Theta}}$

Supervised Learning

Inputs: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$

Outputs: $\boldsymbol{Y} \in \mathbb{R}^N$ (regression), $\boldsymbol{Y} \in \{C_1, \ldots, C_K\}^N$ (classification)

Conditional Likelihood

$p(\boldsymbol{Y}|\boldsymbol{X}, \vec{\boldsymbol{\Theta}}) = \prod_{i=1}^{N} p(\boldsymbol{Y_n}|\boldsymbol{X_n}, \vec{\boldsymbol{\Theta}})$ (for iid data)

After learning, we can do prediction:

(1)**Maximum likelihood**

$\hat{\boldsymbol{\Theta}} = \arg\max_{\vec{\boldsymbol{\Theta}}} p(\boldsymbol{Y}|\boldsymbol{X}, \vec{\boldsymbol{\Theta}})$
Then predictive distribution is $p(\boldsymbol{y^*}|\boldsymbol{x^*}, \hat{\boldsymbol{\Theta}})$ where $\boldsymbol{y^*}$ = new output and $\boldsymbol{x^*}$ = new input
If asked for a single best prediction $\hat{\boldsymbol{Y}} : \hat{\boldsymbol{Y}} = \arg\max_{\boldsymbol{y^*}} p(\boldsymbol{y^*}|\boldsymbol{x^*}, \hat{\boldsymbol{\Theta}})$
If given a loss function $L(\boldsymbol{y'}, \boldsymbol{y}^*)$ that quantifies the loss if the true output is $\boldsymbol{y'}$ but we predict $\boldsymbol{y}^*$, we minimize expected loss: $\hat{\boldsymbol{y}}^* = \arg\min_{\boldsymbol{y}^*} \sum_{\boldsymbol{y'}} p(\boldsymbol{y'}|\boldsymbol{x^*}, \hat{\boldsymbol{\Theta}}) L(\boldsymbol{y'}, \boldsymbol{y}^*)$
Look in 1.5 in Bishop for more info

(2a)**Similarly for MAP estimation**
(but include the prior)

(2b)**Bayesian prediction ("Bayesian model averaging")**
Use complete posterior $p(\vec{\boldsymbol{\Theta}}|\boldsymbol{X}, \boldsymbol{Y})$
Predictive distribution $p(\boldsymbol{y^*}|\boldsymbol{x^*}, \boldsymbol{X}, \boldsymbol{Y}) =$
$\int p(\boldsymbol{y^*}|\boldsymbol{x^*}, \vec{\boldsymbol{\Theta}}) p(\vec{\boldsymbol{\Theta}}|\boldsymbol{X}, \boldsymbol{Y}) d\vec{\boldsymbol{\Theta}}$

Then proceed as before when a single best prediction is needed.
If model contains latent variables $\boldsymbol{Z}$, i.e. different latent variables (typically one for each datapoint, but could also be shared parameters): $\boldsymbol{Z} = (Z_1 ... Z_n)$, the likelihood looks like $p(\boldsymbol{X}|\vec{\boldsymbol{\Theta}}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\Theta}) = \sum_{Z_1} ... \sum_{Z_n} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\Theta})$ or $p(\boldsymbol{X}|\vec{\boldsymbol{\Theta}}) = \int p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\Theta}) d\boldsymbol{Z}$
for continuous latent variables. Similarly for the conditional likelihood in the supervised learning case. In this case use EM algorithm for max-likelihood (or MAP) estimation or Variational Bayes or sampling methods for approximate Bayesian learning.

Machine Learning 2

Lecturer: Max Welling
Scribe: Steven Laan & Michael Cabot
Updated: February 6, 2015; March 31, 2016

Lecture #1
October 28, 2013

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

1 Exponential Family Distributions

The probability density functions that belong to the exponential family are characterised by the following formula:

$$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^\mathsf{T}\mathbf{u}(\mathbf{x})\right\}$$

Where $\boldsymbol{\eta}$ are called the *natural parameters* of the distribution, and $\mathbf{u}(\mathbf{x})$ is some function of $\mathbf{x}$. The function $g(\boldsymbol{\eta})$ can be seen as a normalization term.

$$z(\boldsymbol{\eta}) = \frac{1}{g(\boldsymbol{\eta})} = \int \exp\left\{\boldsymbol{\eta}^\mathsf{T}\mathbf{u}(\mathbf{x})\right\} h(\mathbf{x})d\mathbf{x}$$

$$\frac{\partial}{\partial\boldsymbol{\eta}}\log z(\boldsymbol{\eta}) = \frac{\int \exp\left\{\boldsymbol{\eta}^\mathsf{T}\mathbf{u}(\mathbf{x})\right\}\mathbf{u}(\mathbf{x})h(\mathbf{x})d\mathbf{x}}{z(\boldsymbol{\eta})} = \mathbb{E}\left[\mathbf{u}(\mathbf{x})\,|\,\boldsymbol{\eta}\right]$$

Examples of the exponential family: Bernoulli, categorical ("multinomial" in Bishop), and most distributions in chapter 2 (except for mixtures of Gaussians). Now we show that the normal (Gaussian) distribution is a member of this family:

$$\begin{aligned}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) &= (2\pi)^{-\frac{D}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \\ &= \underbrace{(2\pi)^{-\frac{D}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right)}_{g(\boldsymbol{\eta})}\exp\left(-\frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\mathbf{x}\mathbf{x}^\mathsf{T}) + (\boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{x})\right)\end{aligned} \tag{1}$$

where:

$$\begin{aligned}\begin{pmatrix}\boldsymbol{\eta}_1\\ \boldsymbol{\eta}_2\end{pmatrix} &= \begin{pmatrix}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\\ -\frac{1}{2}\boldsymbol{\Sigma}^{-1}\end{pmatrix} \\ \begin{pmatrix}\mathbf{u}_1(\mathbf{x})\\ \mathbf{u}_2(\mathbf{x})\end{pmatrix} &= \begin{pmatrix}\mathbf{x}\\ \mathbf{x}\mathbf{x}^\mathsf{T}\end{pmatrix} \\ h(\mathbf{x}) &= 1\end{aligned}$$

Note that in (1) we use the special notation $|\mathbf{A}| = |\det(\mathbf{A})|$.

Now it is shown that the expected value of the normal distribution is equal to $\boldsymbol{\mu}$:

$$\text{if } \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma}) \text{ then } \mathbb{E}\left[\mathbf{x}\right] \stackrel{?}{=} \boldsymbol{\mu}$$

Note that, Gaussian distribution belongs to exponential distribution family, thus

$$\begin{aligned}\mathbb{E}(\mathbf{x}\,|\,\boldsymbol{\eta}) = \mathbb{E}(\mathbf{u}_1(\mathbf{x})\,|\,\boldsymbol{\eta}) &= \frac{\partial}{\partial \boldsymbol{\eta}_1}\log z(\boldsymbol{\eta}) = \frac{\partial}{\partial(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})}\left(\frac{1}{2}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right)\\ &= \frac{\partial}{\partial \boldsymbol{\eta}_1}(\frac{1}{2}\boldsymbol{\eta}_1^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{\eta}_1) = \boldsymbol{\Sigma}\boldsymbol{\eta}_1 = \boldsymbol{\Sigma}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}) = \boldsymbol{\mu}\end{aligned}$$

The following derivation is left as an exercise for the readers:

$$\mathbb{E}[\mathbf{x}\mathbf{x}^{\mathsf{T}}] = \frac{\partial}{\partial \boldsymbol{\eta}_2}\log z = \cdots = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^{\mathsf{T}}$$

2 Maximum Likelihood

Given a dataset $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the log-likelihood is given by:

$$\mathcal{L}(\boldsymbol{\eta}, \mathcal{D}) = \sum_{n=1}^{N} \ln(p(\mathbf{x}_n|\boldsymbol{\eta}))$$

Taking the gradient of the log-likelihood with respect to $\boldsymbol{\eta}$, we have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\eta}} &= \sum_n \frac{\partial}{\partial \boldsymbol{\eta}} \ln\left(h((x)_n)g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^{\mathsf{T}}\mathbf{u}(\mathbf{x}_n)\right\}\right)\\ &= N\frac{\partial}{\partial \boldsymbol{\eta}}\ln\left(g(\boldsymbol{\eta})\right) + \sum_n \mathbf{u}(\mathbf{x}_n)\end{aligned}$$

Setting the gradient of $\mathcal{L}(\boldsymbol{\eta}, \mathcal{D})$ with respect to $\boldsymbol{\eta}$ to zeros, we get

$$-\frac{\partial}{\partial \boldsymbol{\eta}}\ln\left(g(\boldsymbol{\eta})\right) = \frac{1}{N}\sum_n \mathbf{u}(\mathbf{x}_n) = \overline{\mathbf{u}(\mathbf{x})}$$

Which yields that the Maximum Likelihood estimate $\hat{\boldsymbol{\eta}}$ is a function of only $\overline{u(\mathbf{x})}$:

$$\hat{\boldsymbol{\eta}} = F(\overline{\mathbf{u}})$$

The Maximum Likelihood Estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are:

$$\hat{\boldsymbol{\mu}} = \mathbb{E}[\mathbf{x}] = \overline{u(\mathbf{x})} = \overline{\mathbf{x}} = \frac{1}{N}\sum_n \mathbf{x}_n \tag{2}$$

$$\hat{\boldsymbol{\Sigma}} = \mathbb{E}[\mathbf{x}\mathbf{x}^{\mathsf{T}}] + \hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^{\mathsf{T}} = \frac{1}{N}\sum_n \mathbf{x}_n\mathbf{x}_n^{\mathsf{T}} + \hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^{\mathsf{T}} \tag{3}$$

3 Gaussians

Some properties of Gaussian distributions. Given distribution $\mathcal{N}(\boldsymbol{x}, \mu, \Sigma)$, then

$$\left.\begin{array}{l}\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)\\ \boldsymbol{\mu} = (\boldsymbol{\mu}_a, \boldsymbol{\mu}_b)\\ \boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma}_{a,a} & \boldsymbol{\Sigma}_{a,b}\\ \boldsymbol{\Sigma}_{b,a} & \boldsymbol{\Sigma}_{b,b}\end{bmatrix}\end{array}\right\} \begin{array}{rll} p(\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{a,a}) & & \text{(marginal distribution)}\\ p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b}) & & \text{(conditional distribution)}\\ \boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Sigma}_{a,a} - \boldsymbol{\Sigma}_{a,b}\boldsymbol{\Sigma}_{b,b}^{-1}\boldsymbol{\Sigma}_{b,a} & & \\ \boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{a,b}\boldsymbol{\Sigma}_{b,b}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b) & & \end{array}$$

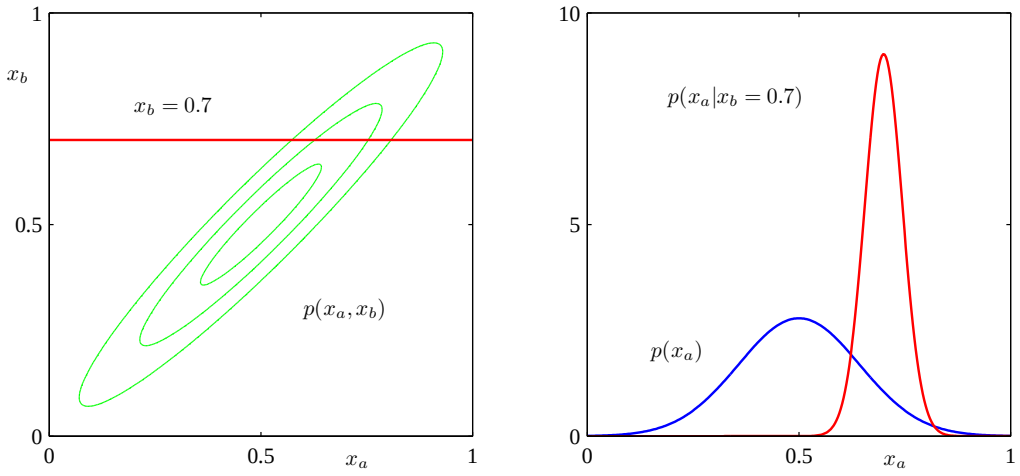


Figure 1: Left shows the contours of a Gaussian $p(\boldsymbol{x}_a, \boldsymbol{x}_b)$. Right shows the marginal distribution $p(\boldsymbol{x}_a)$ (blue) and conditional distribution $p(\boldsymbol{x}_a|\boldsymbol{x}_b = 0.7)$ (red).

This is visualised in figure 1. The figure shows the marginal distribution $p(\mathbf{x}_a)$ and conditional distribution $p(\mathbf{x}_a|\mathbf{x}_b)$ of the Gaussian distribution $p(\mathbf{x}_a, \mathbf{x}_b)$ over two variables.

Another nice identity (that is not in Bishop, but still useful) concerns products of Gaussians. Certain products of Gaussians can be rewritten as a different product of Gaussians:

$$\mathcal{N}(\mathbf{x}\,|\,\mathbf{a}, \mathbf{A})\mathcal{N}(\mathbf{x}\,|\,\mathbf{b}, \mathbf{B}) = \mathcal{N}(\mathbf{a}\,|\,\mathbf{b}, \mathbf{A} + \mathbf{B})\mathcal{N}(\mathbf{x}\,|\,\mathbf{c}, \mathbf{C})$$

where

$$\begin{aligned}\mathbf{C} &= (\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1},\\ \mathbf{c} &= \mathbf{C}(\mathbf{A}^{-1}\mathbf{a} + \mathbf{B}^{-1}\mathbf{b}).\end{aligned}$$

This is e.g. also useful to derive that the convolution of two Gaussians is again a Gaussian.

4 Student-t distribution

The Student-t distribution is a heavy-tailed distribution:

$$\begin{aligned}x \to \pm\infty \quad & \mathcal{N} \propto e^{-\frac{1}{\sigma^2}x^2}\\ & \mathrm{St}(x) \propto |x|^{-\alpha}\end{aligned}$$

There is a powerlaw instead of an exponential one.

A Student-t distribution emerges if for example an Infinite Mixture of Gaussians is used:

1. Draw precision $\tau \sim \mathrm{Gamma}(a, b)$
2. Draw $x \sim \mathcal{N}(\mu, \tau^{-1})$

The resulting x will be distributed according to the Student-t distribution:

$$p(x) \sim \mathrm{St}(x \mid \mu, \lambda = a/b, \nu = 2a)$$

Where the Gamma distribution is defined as:

$$\mathrm{Gamma}(\tau \mid a, b) = \begin{cases} -\frac{b^a}{\Gamma(a)} \tau^{a-1} e^{-b\tau} & \tau > 00\tau \leq 0 \end{cases}$$

where Γ is the gamma function, defined as follows where the gamma function is defined as follows and has the given properties:

$$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u}\, du$$

and with the properties:

$$\Gamma(x+1) = \Gamma(x)x \quad \forall x \in \mathbb{R}; \qquad \Gamma(n+1) = n! \quad \forall n \in \mathbb{N}$$

Therefore, x is proportional to:

$$x \sim c \cdot e^{-\frac{\tau}{2}(x-\mu)^2}$$

The Student-t distribution can now be calculated using this knowledge:

$$\begin{aligned} p(x|\mu, a, b) &= \int_0^\infty p(\tau \mid a, b) p(x|\tau \mid \mu, \tau) d\tau \\ &= \int_0^\infty \frac{b^a}{\Gamma(a)} e^{-b\tau} \tau^{a-1} \left(\frac{\tau}{2\pi}\right)^{\frac{1}{2}} \exp(-\frac{\tau}{2}(x-\mu)^2) d\tau \\ &= \frac{b^a}{\Gamma(a)\sqrt{2\pi}} \int_0^\infty \tau^{a-\frac{1}{2}} \exp(-(b + \frac{1}{2}(x-\mu)^2)\tau) d\tau \end{aligned}$$

Now note that the integrand is proportional to:

$$\mathrm{Gamma}(\tau \mid a + \frac{1}{2}, b + \frac{1}{2}(x-\mu)^2)$$

We use the fact that the Gamma distribution is properly normalized:

$$\frac{\Gamma(a)}{b^a} = \int_0^\infty \tau^{a-1} e^{-b\tau}\, d\tau$$

to obtain

$$\begin{aligned} p(x|\mu, a, b) &= \frac{b^a}{\Gamma(a)\sqrt{2\pi}} \int_0^\infty \tau^{a-\frac{1}{2}} \exp(-(b + \frac{1}{2}(x-\mu)^2)\tau) d\tau \\ &= \frac{b^a}{\Gamma(a)\sqrt{2\pi}} \left(b + \frac{1}{2}(x-\mu)^2\right)^{-a-\frac{1}{2}} \Gamma(a + \frac{1}{2}) \\ &= \mathrm{St}(x \mid \mu, \lambda, \nu) \end{aligned}$$

with $\lambda = a/b$ and $\nu = 2a$.

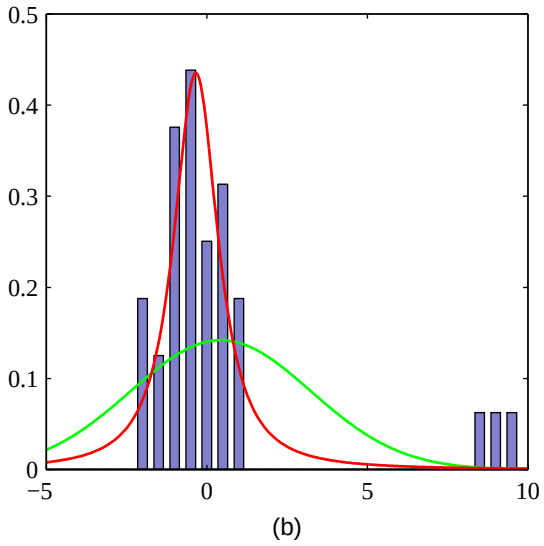


Figure 2: Histogram distribution fitted by a Student-t distribution (red) and a Gaussian (Green).

The d-dimensional Student distribution is given by:

$$\mathrm{St}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma},v) = \frac{\Gamma(\frac{d}{2}+\frac{v}{2})}{\Gamma(\frac{v}{2})}\frac{1}{(\pi v)^{\frac{d}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}(1+\nu^{-1}\underbrace{(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}_{\text{Mahalanobis distance}})^{-\frac{d}{2}-\frac{v}{2}}$$

The heavy tail of the student-t distribution makes it more robust against outliers as is shown in figure 2. The figure shows a histogram distribution of 30 data points drawn from a Gaussian distribution with three additional outlying data points to the right. The red and green line are the maximum likelihood fit obtained from a t-distribution and a Gaussian distribution, respectively.

5 Independent Component Analysis

Given is a signal S, consisting of two components:

$$S(t) = \begin{bmatrix} S_1(t) \\ S_2(t) \end{bmatrix}$$

You can view this as two sources of sound.

Next we have two microphones, that capture a mixture of the sounds.

$$\left.\begin{aligned} \mathbf{X}_1(t) &= \alpha_1 S_1(t) + \beta_1 S_2(t) \\ \mathbf{X}_2(t) &= \alpha_2 S_2(t) + \beta_2 S_2(t) \end{aligned}\right\} \mathbf{X}(t) = \begin{bmatrix} \boldsymbol{X}_1(t) & \mathbf{X}_2(t) \end{bmatrix}$$

Where α and β are the parameters specifying the exact mixture. We can define the parameter matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \end{bmatrix}$$

And using that:

$$\boldsymbol{X}_t = \mathbf{A}\mathbf{S}_t$$

Now the joint probability can be expressed as:

$$\begin{aligned} p(S_1, S_2) &= p_1(S_1)p_2(S_2) && \text{(independence assumption)} \\ &= \mathrm{St}(S_1|\mu_1 = 0, v_1, \lambda_1)\mathrm{St}(S_2|\mu_2 = 0, v_2, \lambda_2) \end{aligned}$$

We assume that $\mathbf{A}$ is invertible and denote $\mathbf{A}^{-1} = \mathbf{W}$. Using general transformation rule for densities of random variables for the case when $\mathbf{x}$ is a deterministic function of $\mathbf{S}$:

$$\begin{aligned} p(\mathbf{x}) &= p(\mathbf{S}) \left| \frac{\partial \mathbf{S}}{\partial \mathbf{x}} \right| \\ &= p_1(\mathbf{W}_1^\mathsf{T}\mathbf{x}) p_2(\mathbf{W}_2^\mathsf{T}\mathbf{x}) |\mathbf{W}| \end{aligned}$$

Where we use:

$$\mathbf{X} = \mathbf{A}\mathbf{S}$$

and

$$\mathbf{S} = \mathbf{A}^{-1}\mathbf{X} = \mathbf{W}\mathbf{X}$$

Note that $\mathbf{W}$ coincides with the Jacobian matrix:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W_1} \\ \mathbf{W_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{S}_1}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{S}_1}{\partial \mathbf{x}_2} \\ \frac{\partial \mathbf{S}_2}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{S}_2}{\partial \mathbf{x}_2} \end{bmatrix}$$

The likelihood is given by:

$$\mathcal{L}(D, \mathbf{W}) = \sum_n \Big(\log(p_1(\mathbf{W}_1^\top \mathbf{x}_n)) + \log(p_2(\mathbf{W}_2^\top \mathbf{x}_n)) + \log(|\mathbf{W}|) \Big)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial W_{ij}} &= \sum_n \left(\frac{\partial}{\partial W_{ij}} \log(p_i(\underbrace{\mathbf{W}_i^\top \mathbf{x}_n}_{S_{in}})) + \frac{\partial}{\partial W_{ij}} \log(|\mathbf{W}|) \right) \qquad (4) \\ &= \left(\sum_n \frac{\partial}{\partial S_i} \log(p_i(S_i)) \Big|_{S_i = S_{in}} x_{jn} \right) + N(\mathbf{W}^{-\top})_{ij} \end{aligned}$$

Where in (4) this special rule is used:

$$\frac{\partial}{\partial \mathbf{A}} \log(|\mathbf{A}|) = \mathbf{A}^{-\top}$$

In vector notation:

$$\boldsymbol{\nabla}_\mathbf{W} \mathcal{L} = \sum_n \left(\boldsymbol{\nabla}_\mathbf{S} \log p(\mathbf{S}) \Big|_{\mathbf{S}=\mathbf{S}_n} \mathbf{x}_n^\mathsf{T} + \mathbf{W}^{-\mathsf{T}} \right)$$

Update rule for an iterative algorithm:

$$\mathbf{W}^{t+1} = \mathbf{W}^t + \frac{1}{N} (\boldsymbol{\nabla}_\mathbf{W} \mathcal{L}) \mathbf{W}^\mathsf{T} \mathbf{W}$$

(note that it has a fixed point at $\boldsymbol{\nabla}_\mathbf{W} \mathcal{L} = \mathbf{0}$) Because $\mathbf{X}^\mathsf{T}\mathbf{W}^\mathsf{T} = \mathbf{S}^\mathsf{T}$, this simplifies to:

$$\mathbf{W}^{t+1} = \mathbf{W}^t + \frac{1}{N} \sum_n \left(\boldsymbol{\nabla}_\mathbf{S} \log p(\mathbf{S}) \Big|_{\mathbf{S}=\mathbf{S}_n} \mathbf{S}_n^\mathsf{T} + \mathbf{I} \right) \mathbf{W}$$

After convergence, one can reconstruct the latent signals by:

$$\mathbf{S}_n = \mathbf{W}\mathbf{X}_n$$

Iterative algorithm:

1. Pick data-case $\mathbf{X}_n$.
2. Compute $\mathbf{S}_n = \mathbf{W}^\mathsf{T}\mathbf{X}_n$.
3. Update $\mathbf{W}^{t+1}$ using the formula given above.

Machine Learning 2

Lecturer: dr. Joris Mooij
Scribe: Norbert Heijne & Adam Sasiadek
Updated: February 23, 2015; April 4, 2016

Lecture #2
October 30, 2013

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

2.1 Binary variables

Bernoulli Distribution

The Bernoulli distribution is a discrete probability distribution which takes the value 1 with a success probability μ and value 0 with a failure probability $(1-p)$. It is given by:

$$\text{Bern}(x|\mu) = \mu^x(1-\mu)^{1-x} = \begin{cases} \mu & x = 1 \\ 1-\mu & x = 0 \end{cases}$$

Expectation, Variance and Max Likelihood

Mean

$$\mathbb{E}[x] = \mu$$

Variance

$$\mathbb{V}\text{ar}[x] := \mathbb{E}[(x - \mathbb{E}[x])^2] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \mu - \mu^2$$

Max Likelihood

$$\mu_{ML} = 1/N(\sum_{n=1}^{N} x_n) = \underbrace{\frac{m}{N}}_{\text{where m} = \#\ \{x_n = 1\ \}}$$

The Bernoulli distribution is part of the exponential family:

$$p(x|\eta) = \sigma(-\eta)\exp(\eta \cdot x) \qquad \text{where } \sigma(\eta) = \frac{1}{1+\exp(-\eta)}, \text{ i.e., } \eta = \ln\frac{\mu}{1-\mu}$$

Binomial Distribution

The binomial distribution is the discrete probability distribution of the number of successes in a sequence of N independent Bernoulli experiments, each of which yields success with probability μ.

$$m = \sum_{i=1}^{N} x_i, \qquad x_i \overset{i.i.d}{\sim} \text{Bern}(\mu) \qquad \text{Bin}(m|N,\mu) = \binom{N}{m}\mu^m(1-\mu)^{N-m}$$

Here, $\binom{N}{m} := \frac{N!}{(N-m)!m!}$, the "binomial coefficient".

Expectation,Variance and Maximum Likelihood

Mean	Variance
$\mathbb{E}[m] = \mu N$	$\mathbb{E}[(m - \mathbb{E}[m])^2] = N\mu(1-\mu)$

Max Likelihood

$$\mu_{ML} = \frac{m}{N}$$

Bayesian approach

Maximum Likelihood leads to overfitting. Example: coin flip, observing twice heads and zero tails. The Bayesian approach can avoid this phenomenon. In order to use the Bayesian approach we need a prior.

Likelihood

$$p(X|\mu) = \prod_{i=1}^{N} p(x_i|\mu) = \prod_{i=1}^{N} \mu^{x_i}(1-\mu)^{1-x_i} = \mu^{\sum_i x_i}(1-\mu)^{\sum_i (1-x_i)} = \mu^m (1-\mu)^{N-m}$$

Prior

A convenient choice for a prior is a *conjugate prior*, that is, it should have a similar functional form as the likelihood in order to make it easier to handle analytically. In this case, this means the prior should depend on μ as follows:

$$p(\mu) \propto \mu^{a-1}(1-\mu)^{b-1} \text{ for some } a, b$$

Beta distribution

This is called the *Beta* distribution with parameters $a > 0, b > 0$:

$$p(\mu) = \text{Beta}(\mu|a,b) := \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \mu^{a-1}(1-\mu)^{b-1} \tag{1}$$

where the gamma function is defined as follows and has the given properties:

$$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u}\, du$$

$$\Gamma(x+1) = \Gamma(x)x \quad \forall x \in \mathbb{R}; \qquad \Gamma(n+1) = n! \quad \forall n \in \mathbb{N}$$

The posterior can then be written as:

$$p(\mu|X) \propto p(X|\mu)p(\mu) = \mu^{a-1+\sum x_i}(1-\mu)^{b-1+\sum(1-x_i)} = \text{Beta}(\mu|\underbrace{a + \sum x_i}_{\text{positive cases}}, \underbrace{b + N - \sum x_i}_{\text{negative cases}})$$

Note that we do not explicitly need the normalization constant for the posterior, as it is a Beta distribution (for which we know the normalization constant, see (1)). Note that observing the data increases the hyperparameters a by $m = \sum_{i=1}^{N} x_i$ and b by $N - m$. So a and b can be interpreted as "pseudo-counts" from virtual data that we can interpret to give us our prior belief on μ. Also, if we observe more data, the posterior after the first batch can act as the prior for the second batch of data.

Mean and Variance

If $\mu \sim \text{Beta}(a, b)$ then:

Mean	**Variance**
$\mathbb{E}[\mu] = \frac{a}{a+b}$	$Var[\mu] = \frac{ab}{(a+b)^2(a+b+1)}$

Predictive Distribution

$$p(x^* = 1|X) = \int_0^1 p(x^* = 1|\mu)p(\mu|X)d\mu = \int_0^1 \mu p(\mu|X)d\mu = \mathbb{E}[\mu|X] = \frac{m+a}{m+a+(N-m)+b}$$

Note: if $N \to \infty$, then $p(x^* = 1|X) \to \frac{m}{N}$, the maximum likelihood estimate of μ. Coin flip example: Bayesian prediction gives more sensible predictions than maximum likelihood.

2.2 Discrete variables

We have K possible values. A value is represented as $\vec{x} \in \mathbb{R}^K$ e.g:

$$\begin{bmatrix}1\\0\\0\end{bmatrix} = C_1, \qquad \begin{bmatrix}0\\1\\0\end{bmatrix} = C_2, \qquad \begin{bmatrix}0\\0\\1\end{bmatrix} = C_3$$

This is in order to obtain a simple form for the *categorical distribution* with parameter $\vec{\mu}$:

$$p(\vec{x}|\vec{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k} \qquad \vec{\mu} \in [0,1]^K \text{ and } \sum_k \mu_k = 1$$

The likelihood for N i.i.d. data points is:

$$p(X|\vec{\mu}) = p(\vec{x}_1, \ldots, \vec{x}_N|\vec{\mu}) = \prod_{n=1}^{N}\prod_{k=1}^{K} \mu_k^{x_{nk}} = \prod_{k=1}^{K}\prod_{n=1}^{N} \mu_k^{x_{nk}} = \prod_{k=1}^{K} \mu_k^{\sum_n x_{nk}} = \prod_{k=1}^{K} \mu_k^{m_k}$$

where $m_k = \sum_{n=1}^{N} x_{nk}$ counts the number of values in each category/class. The maximum likelihood estimate for $\vec{\mu}$ is then:

$$\vec{\mu}_{ML} = \frac{\vec{m}}{N}$$

where $\vec{m}$ is the number of observed values for each class.
The categorical distribution is part of the exponential family:

$$p(\vec{x}|\vec{\eta}) = \frac{\exp(\vec{\eta}^T\vec{x})}{1 + \sum_{k=1}^{K-1}\exp(\eta_k)} \qquad \text{where } \eta_k = \ln\frac{\mu_k}{1 - \sum_{j=1}^{K-1}\mu_j}$$

Multinomial distribution

The multinomial distribution is the probability distribution on the "counts" that results from performing N independent draws from a categorical distribution with parameter $\vec{\mu}$:

$$\text{Mult}(m_1, \ldots, m_K | N, \vec{\mu}) = \binom{N}{m_1, \ldots, m_K} \prod_{k=1}^{K} \mu_k^{m_k}$$

Here,

$$\binom{N}{m_1, \ldots, m_K} := \frac{N!}{m_1! \ldots m_K!}$$

Dirichlet distribution

The conjugate prior for the multinomial distribution is the Dirichlet distribution:

$$\text{Dir}(\vec{\mu}|\vec{\alpha}) = \frac{\Gamma(\sum_k \alpha_k)}{\Gamma(\alpha_1) \ldots \Gamma(\alpha_k)} \prod_{k=1}^{K} \mu_k^{(\alpha_k - 1)}$$

Like in the binary case we could calculate the posterior etc.

Binary Variables		Discrete Variables
Bernoulli	$\rightarrow$	Categorical Distribution
Binomial	$\rightarrow$	Multinomial
Beta	$\rightarrow$	Dirichlet

Be aware: the categorical distribution is often also called multinomial distribution (including in Bishop's book).

2.3 Gaussian distribution

See also lecture 1.

2.3.3 Bayes theorem for Gaussian Variables

If

$$\begin{aligned} \vec{x} \in \mathbb{R}^M, \vec{y} \in \mathbb{R}^D \\ p(\vec{x}) = \mathcal{N}(\vec{x}|\vec{\mu}, \mathbf{\Lambda^{-1}}) \\ p(\vec{y}|\vec{x}) = \mathcal{N}(\vec{y}|\mathbf{A}\vec{x} + \vec{b}, \mathbf{L}^{-1}) \end{aligned}$$

then the marginal distribution for $\vec{y}$ is again Gaussian:

$$p(\vec{y}) = \mathcal{N}(\vec{y}|\mathbf{A}\vec{\mu} + \vec{b}, \mathbf{L}^{-1} + \mathbf{A}\mathbf{\Lambda^{-1}}\mathbf{A^T})$$

and applying Bayes theorem shows that the conditional distribution for $\vec{x}$ given $\vec{y}$ is also Gaussian:

$$\begin{aligned} p(\vec{x}|\vec{y}) = \mathcal{N}(\vec{x}|\Sigma(\mathbf{A^T L}(\vec{y} - \vec{b}) + \mathbf{\Lambda}\vec{\mu}), \Sigma) \\ \Sigma = (\mathbf{\Lambda} + \mathbf{A^T L A})^{-1} \end{aligned}$$

2.3.4 Maximum Likelihood for Gaussian $\vec{x}_n \in \mathbb{R}^D$

Data $\mathbf{X} = (\vec{x}_1, \ldots, \vec{x}_N)^T$
Log Likelihood= $\ln p(\mathbf{X}|\vec{\mu}, \mathbf{\Sigma}) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln|\mathbf{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N}(\vec{x}_n - \vec{\mu})^T\mathbf{\Sigma}^{-1}(\vec{x}_n - \vec{\mu})$

The log-likelihood depends on $\mathbf{X}$ only via the sufficient statistics: $\sum_n \vec{x}_n$, $\sum_n \vec{x}_n\vec{x}_n^T$:

$$\vec{\mu}_{ML} = \frac{1}{N}\sum_{n=1}^{N}\vec{x}_n$$
$$\mathbf{\Sigma}_{ML} = \frac{1}{N}\sum_{n=1}^{N}(\vec{x}_n - \vec{\mu}_{ML})(\vec{x}_n - \vec{\mu}_{ML})^T$$

Online learning $\leftrightarrow$ Sequential Learning
Stochastic gradient Descent $\leftrightarrow$ Special case of Robbins-Monro Algorithm for ML estimation

$$\max_{\vec{\theta}} p(\vec{x}_1, \ldots, \vec{x}_N|\vec{\theta})$$
$$\vec{\theta}^{(N)} = \vec{\theta}^{(N-1)} + a_{N-1}\frac{\partial}{\partial\vec{\theta}^{(N-1)}}\ln p(\vec{x}_n|\vec{\theta}^{(N-1)})$$

where:

$$\lim_{N\to\infty}(a_N) = 0$$
$$\sum_{n=1}^{\infty} a_n = \infty$$
$$\sum_{n=1}^{\infty} a_n^2 < \infty$$

Example $\vec{\mu}_{ML}$

$\vec{\mu}^{(N)} = \vec{\mu}^{(N-1)} + \frac{1}{N}(\vec{x}_n - \vec{\mu}^{(N-1)})$

2.3.6 Bayesian Inference for Gaussian

dimensions = 1
Data = $\{x_1, \ldots, x_N\}$

Variance known, mean estimated

σ^2 known
μ estimated
conjugate prior for μ: $p(\mu) = \mathcal{N}(\mu|\mu_0, \sigma_0^2)$

$$p(\mu|D) \propto p(D|\mu)p(\mu) = \mathcal{N}(\mu|\mu_N, \sigma_N^2)$$

where

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2 + \sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2}\mu_{ML}$$
$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}$$

This can be derived by using Bayes' theorem for Gaussian variables on page 93.

Variance unknown, mean known

σ^2 unknown
μ known
conjugate prior for $\lambda = \frac{1}{\sigma^2}$

$$\begin{aligned}
\text{likelihood:} \quad p(D|\lambda) &= \prod_n \mathcal{N}(x_n|\mu, \lambda^{-1}) \propto \lambda^{N/2} \exp\left[-\frac{\lambda}{2}\sum_n (x_n - \mu)^2\right] \\
&\propto \text{Gamma}\left(\lambda \middle| \frac{N}{2} + 1, \frac{1}{2}\sum_n (x_n - \mu)^2\right) \\
\text{prior:} \quad p(\lambda) &= \text{Gamma}(\lambda|a_0, b_0) \\
\text{posterior:} \quad p(\lambda|D) &= \text{Gamma}(\lambda|a_N, b_N)
\end{aligned}$$

where $a_N = a_0 + \frac{N}{2}$ and $b_N = b_0 + \frac{1}{2}\sum_n (x_n - \mu)^2$. Here, a_0 and b_0 can be interpreted as if they were generated from virtual data points.

Variance unknown, mean unknown

σ^2 unknown
μ unknown
"normal-Gamma" distribution
$\mathcal{N}(\mu|\mu_0, (\beta\lambda)^{-1})\text{Gamma}(\lambda|a, b)$

See Bishop for more details.

2.3.7 Student's t distribution

See lecture 1.

2.3.8 Periodic variables: Von Mises distribution

See Bishop.

2.3.9 Mixtures of Gaussians

Treated in Machine Learning: Pattern Recognition.

2.4.2 Conjugate priors for exponential family distributions

Read 2.4.2 in Bishop for the general theory on conjugate priors for the exponential family. All exponential family distributions have corresponding conjugate priors.

1.6 Information theory

"Information of event A":

$$\begin{aligned} h(A) &= -\log_2 p(A), \text{ in bits} \\ &= -\ln p(A) \text{ in naturals} \end{aligned}$$

Shannon Entropy:

$$H(x) := -\sum_x p(x) \cdot \log_a(p(x))$$

Example fair coin

$\frac{1}{2}$ vs $\frac{1}{2} = 1$ bit
K independent coins = K bits

Differential entropy:

$$H(\vec{x}) = -\int p(\vec{x}) \ln p(\vec{x}) d\vec{x}$$

Kullback-Leibler divergence or Relative entropy or the "distance" between distributions:

$$\begin{aligned} KL(p||q) = -\int p(\vec{x}) \ln \left(\frac{q(\vec{x})}{p(\vec{x})} \right) d\vec{x} \\ KL(p||q) \geq 0 \\ KL(p||q) = 0 \iff p = q \end{aligned}$$

Conditional entropy:

$$H(\vec{y} \,|\, \vec{x}) = -\int p(\vec{x}) \int p(\vec{y} \,|\, \vec{x}) \ln p(\vec{y} \,|\, \vec{x}) \, d\vec{y} d\vec{x}$$

$$H(\vec{x}, \vec{y}) = H(\vec{x}) + H(\vec{y} \,|\, \vec{x}) = H(\vec{y} + H(\vec{x} \,|\, \vec{y})$$

Mutual information:

$$I(\vec{x} : \vec{y}) = KL\big(p(\vec{x}, \vec{y}) \,||\, p(\vec{x})p(\vec{y})\big) = H(\vec{x}) - H(\vec{x} \,|\, \vec{y}) = H(\vec{y} - H(\vec{y} \,|\, \vec{x})$$

Information-theoretic interpretation of Maximum likelihood:

$$\min_{\vec{\theta}} KL\big(p(X)||q(X \,|\, \vec{\theta})\big)$$

where p is the empirical distribution and q the model distribution (parameterized by $\vec{\theta}$) results in the ML estimation of $\vec{\theta}$.

Machine Learning 2

Lecturer: Max Welling
Scribe: Agnes van Belle & Nikos Voskarides
Updated: April 17, 2016

Lecture #3
November 4, 2013

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

(Probabilistic) Graphical Models

GMs:

- Capture noise
- Are useful for reasoning
- Are useful when you have many variables, enormous data sets
- Are useful for causality

In the latter case you can not just make a table:

$$\overbrace{M \text{ sensors, } K \text{ states}}^{K^M \text{ states}}$$

Imagine for every state having a separate parameter. This leads to $K^M - 1$ parameters (the -1 is added because they have to sum to 1, $\sum_Z p(Z) = 1$). We will be overfitting this way. We want to build *structure* in these models: reduce number of parameters or degrees of freedom.

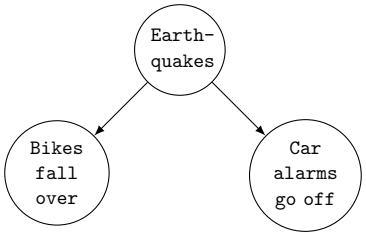


Figure 1: Example. Variables can have values from $\{0, 1\}$

In Figure 1, for example, if `Earthquakes` is 1, then `Bikes fall over` can be 0 and `Car alarms go off` is 0.001. We can model *conditional independence*: `Bikes fall over` and `Car alarms go off` are likely to not be independent. But if an earthquake occurs, i.e. `Earthquakes` is 1, they will likely be independent.

An example GM that is a BN is given in Figure 2. You can see it as nature generating node 1 first, then nodes $2 \ldots 6$, etc. It is a generative process, that captures causality.

GMs are good for:

- Designing models
- Communicating models (you don't need (long) equations to summarize your model)

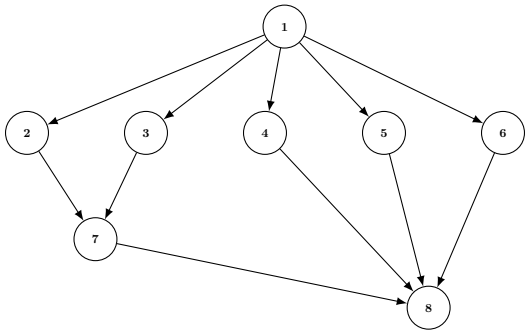
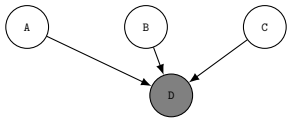


Figure 2: Random example of a graphical model (Bayesian network)

- Conditional independence relations are encoded in the graph
- Inference: from effects to causes – use message passing algorithms, belief propagation, to infer what is the state of variables when e.g. node `D` is observed in the GM of a BN below



Sidemark: GMs are not like Neural Networks (NNs). A few reasons are: NNs are models for *supervised* problems; a NN doesn't define a probabilistic model; transitions are not stochastic in a NN; a NN does not model conditional independence.

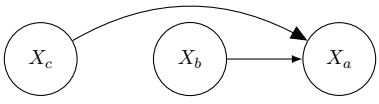
Modeling: Directed Graphical Models

Concrete example

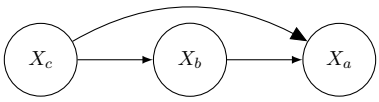
Let's start with something that is generally true: the chain rule:

$$P(X_a, X_b, X_c) = P(X_a|X_b, X_c)P(X_b|X_c)P(X_c) \tag{1}$$

Let's graphically represent this:
$P(X_a|X_b, X_c)$ equals:



$P(X_a|X_b, X_c)P(X_b|X_c)$ equals:



For $P(X_c)$ we do not need to add something because it does not depend on anything.

General

The general steps are thus:

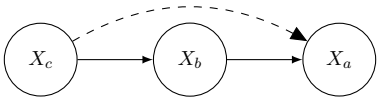
1. Determine ordering of variables
2. In this ordering, call the parents of X_i: $Pa(X_i) = Pa_i$. These are all variables lower in ordering.

$$P(X_i \dots X_M) = \prod_{i=1}^{M} P(X_i|Pa_i) \tag{2}$$

   Assumption: $P(X_i|\varnothing) = P(X_i)$

In the above example X_c and X_b are the *parent set* of X_a, Pa_a. Equation 2 applied to the left side of Equation 1, $P(X_a, X_b, X_c)$, indeed yields the right side of Equation 1, $P(X_a|X_b, X_c)P(X_b|X_c)P(X_c)$. Every equation of that form corresponds with a directed graphical model and vice versa.

To impose structure on a graph is to delete one of the edges. If we stay with the previous example, we can, for example, delete the line from X_c to X_a. Equation 1 has become:



$$P(X_a, X_b, X_c) = P(X_a|X_b)P(X_b|X_c)P(X_c) \tag{3}$$

So the parent set of X_a has changed.

Background remarks

DAGs

This particular BN structure ensures that there are no *directed* loops. Therefore we also call this a Directed Acyclic Graph.

BNs

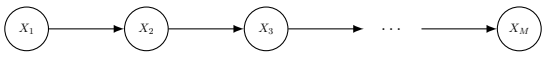
A BN is normalized:

$$\int \prod_{i=1}^{M} P(X_i|Pa_i)\, \mathrm{d}X_1 \dots X_M = 1$$

you can see it if you write out the product.

Markov chains

A chain graph, also called a Markov chain, is graphically modeled below:



It is useful for temporal modeling.

Recall that when everything is fully connected in a model we need $K^M - 1$ parameters, so the parameter complexity is of order $\mathcal{O}(K^M)$. You will *never* have enough data to fit a model of this size. For the Markov chain, however, we need far less:

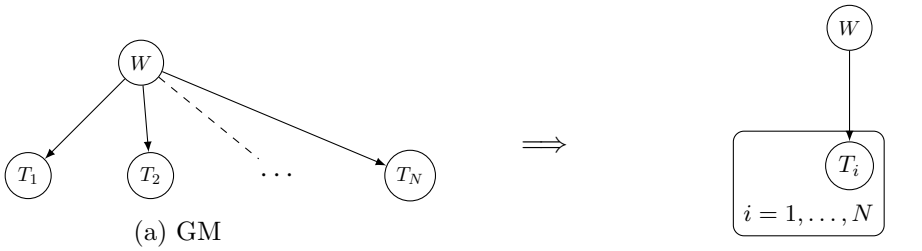
$$\underbrace{\underbrace{P(X_M|X_{M-1})}_{K\cdot(K-1)} \cdot \cdots \cdot \underbrace{P(X_2|X_1)}_{K\cdot(K-1)} \cdot \underbrace{P(X_1)}_{(K-1)}}_{M-1 \text{ of these}}$$

So the total number of parameters needed in a MC model is $(M-1)(K(K-1))+(K-1)$ which is of order $\mathcal{O}(K^2 \cdot M)$.

Example Bayesian Networks

Discriminative

We draw N data points $\{t_1, \ldots, t_N\}$ (data cases), plus some underlying parameter W. We can do this as in Figure 3a or use a shorter notation using "plates" as shown in Figure 3b.



(a) GM

(b) Shorthand GM

Figure 3: Two ways of modeling the same BN.

This plate means we draw N i.d.d. instances/values from the model. You can see a plate as a 3-dimensional box, stacking plates of all N instances.

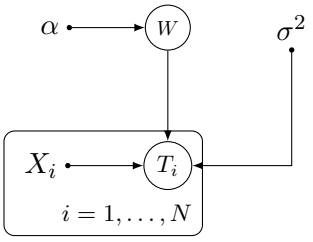
So we have that

$$P(T, W) = p(W) \prod_{i=1}^{N} P(T_i|W)$$

Now we add some parameters X, α, σ^2.

$$P(T, W|X, \alpha, \sigma^2) = P(W|\alpha) \prod_{i=1}^{N} P(T_i|X_i, W, \sigma^2)$$

We want to learn W from T, conditioning on X. This is a regression model (also see previous class). It is also a discriminative model (as opposed to a generative one). The corresponding GM is:



The nodes for X, α, σ^2 are denoted with a small dot. This means that these parameters are fixed – we don't put a distribution over them.

Predictive distribution

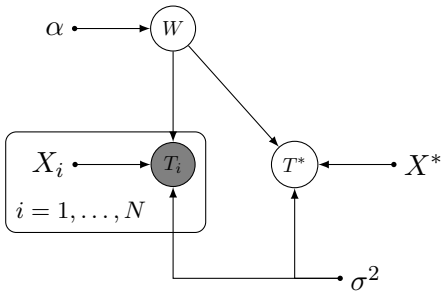
Now consider a model for the predictive distribution. It still corresponds to a regression model. But now we want to make predictions from new input domain variables X_i^* to new output domain variables T_i^*. We assume we know T and we want to know T^*.

$$\begin{aligned} & P(T^*, W, T | X^*, X, \sigma^2, \alpha) \\ &= \left[\prod_{i=1}^{N} P(T_i | X_i, W, \sigma^2) \right] P(W|\alpha) P(T^* | X^*, W, \sigma^2) \end{aligned}$$

The predictive distribution is obtained by integrating over W and substituting T:

$$\begin{aligned} & P(T^*, T | X^*, X, \sigma^2, \alpha) \\ &= \int \left[\prod_{i=1}^{N} P(T_i | X_i, W, \sigma^2) \right] P(W|\alpha) P(T^* | X^*, W, \sigma^2) \, dW \end{aligned}$$

In the new GM shown below, the shaded circle of T_i denotes that we have observed the variable.



From the GM we can directly see that T^* does not directly depend on T_i. From T_i we can learn W and σ^2. So we can throw the rest away after having learned these.

This is a *not* a generative model, because we have not specified a distribution over inputs X_i. If we would do so, it would become a generative model.

Conditional independence relationships in a DAG

Definition

We call two random variables X, Y independent iff

$$p(X, Y) = p(X)p(Y)$$

Intuitively, when told the value of X we do not learn anything about the value of Y (and vice versa).

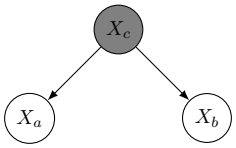
We call two random variables X, Y independent given a (set of) random variables $\mathbf{Z}$ iff

$$p(X, Y \mid \mathbf{Z} = \mathbf{z}) = p(X \mid \mathbf{Z} = \mathbf{z}) p(Y \mid \mathbf{Z} = \mathbf{z})$$

for all possible values $\mathbf{z}$ of $\mathbf{Z}$; this definition works for discrete variables $\mathbf{Z}$ but for continuous variables one has to be more careful and introduce some measure theory. Intuitively, when told the value of X in addition to the value of $\mathbf{Z}$ we do not learn anything new about the value of Y.

Type 1

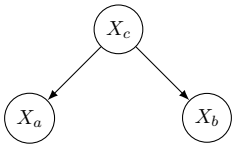
First case



$$X_a \perp\!\!\!\perp X_b | X_c$$

$$\begin{aligned} P(X_a, X_b | X_c) &= \frac{P(X_a, X_b, X_c)}{P(X_c)} && \text{Bayes rule} \\ &= \frac{P(X_a|X_c)P(X_b|X_c)P(X_c)}{P(X_c)} \\ &= P(X_a|X_c)P(X_b|X_c) \end{aligned}$$

Second case

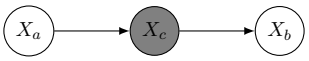


$$X_a \not\perp\!\!\!\perp X_b | \varnothing$$

$$\begin{aligned} P(X_a, X_b) &\stackrel{?}{=} P(X_a)P(X_b) \\ &= \sum_{X_c} P(X_a, X_b, X_c) && \text{marginalizing out } X_c \\ &= \sum_{X_c} P(X_a|X_c)P(X_b|X_c)P(X_c) && \text{indepedence does not hold, counter-example} \end{aligned}$$

Type 2

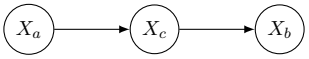
First case



$$X_a \perp\!\!\!\perp X_b | X_c$$

$$\begin{aligned} P(X_a, X_b|X_c) &= \frac{P(X_a, X_b, X_c)}{P(X_c)} && \text{Bayes rule} \\ &= \frac{P(X_b|X_c)P(X_c|X_a)P(X_a)}{P(X_c)} && \text{B.N.} \\ &= P(X_b|X_c)P(X_a|X_c) && \text{Bayes rule} \end{aligned}$$

Second case

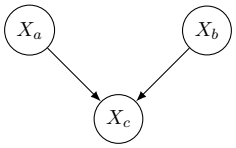


$$X_a \not\perp\!\!\!\perp X_b | \varnothing$$

$$\begin{aligned} P(X_a, X_b) &= \sum_{X_c} P(X_b|X_c)P(X_c|X_a)P(X_a) \\ &= P(X_b|X_a)P(X_a) \\ &\neq P(X_b)P(X_a) \end{aligned}$$

Type 3

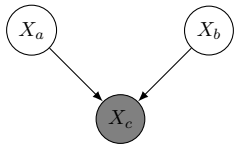
First case



$$X_a \perp\!\!\!\perp X_b | \varnothing$$

$$
\begin{aligned}
P(X_a, X_b) &= \sum_{X_c} P(X_a, X_b, X_c) \\
&= \sum_{X_c} P(X_c|X_a, X_b)P(X_a)P(X_b) \quad \text{normalized distribution inside summation} \\
&= P(X_a)P(X_b)
\end{aligned}
$$

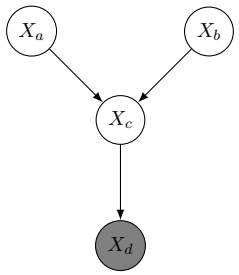
Second case



$$X_a \not\perp\!\!\!\perp X_b|X_c$$

$$
\begin{aligned}
P(X_a, X_b|X_c) &= \frac{P(X_a, X_b, X_c)}{P(X_c)} && \text{Bayes rule} \\
&= \frac{P(X_c|X_a, X_b)P(X_a)P(X_b)}{P(X_c)} \\
&\neq P(X_a|X_c)P(X_b|X_a) && \text{prove with counter-example}
\end{aligned}
$$

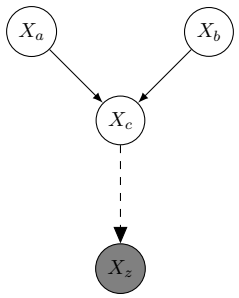
Second case - generalization



$$X_a \not\perp\!\!\!\perp X_b|X_d$$

$$
\begin{aligned}
P(X_a, X_b|X_d) &= \sum_{X_c} \frac{P(X_a, X_b, X_c, X_d)}{P(X_d)} && \text{Bayes rule} \\
&= \sum_{X_c} \frac{P(X_d|X_c)P(X_c|X_a, X_b)P(X_a)P(X_b)}{P(X_d)} \\
&= \frac{P(X_d|X_a, X_b)P(X_a)P(X_b)}{P(X_d)}
\end{aligned}
$$

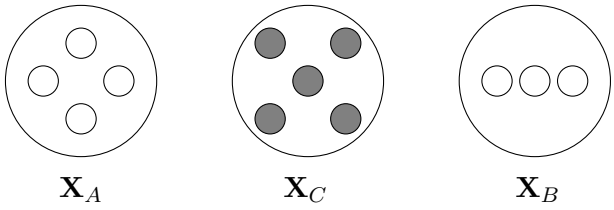
In general, if any descendant node is observed, the independence will be broken:

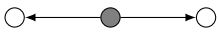
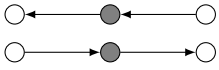
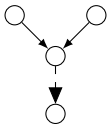


$$X_a \not\perp\!\!\!\perp X_b | X_z$$

d-separation

Is $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_C$ for sets of nodes/variables $\mathbf{X}_A$, $\mathbf{X}_B$ and $\mathbf{X}_C$?



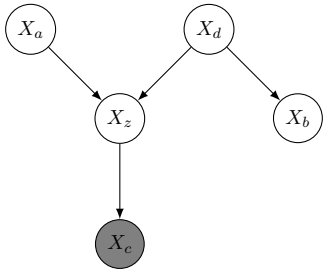
1. Consider all paths from some node in $\mathbf{X}_A$ to some node in $\mathbf{X}_B$
2. A path is **blocked** by $\mathbf{X}_C$ iff it contains a node such that:
   (a)  (the arrows meet tail-to-tail at a node in $\mathbf{X}_C$), or:
   (b)  (the arrows meet head-to-tail the a in $\mathbf{X}_C$), or:
   (c)  (the arrows meet head-to-head at the node, and neither the node nor its descendants are in $\mathbf{X}_C$).
3. If all paths between $\mathbf{X}_A$ and $\mathbf{X}_B$ are blocked by $\mathbf{X}_C$, we say that $\mathbf{X}_A$ and $\mathbf{X}_B$ are **d-separated** by $\mathbf{X}_C$.

If $\mathbf{X}_A$ and $\mathbf{X}_B$ are d-separated by $\mathbf{X}_C$, this implies that $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_C$.

Examples of CIRs in BNs

Ex. 1

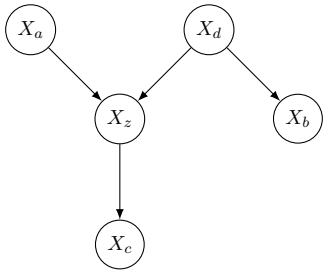
We can go from X_a to X_b.



$$X_a \not\perp\!\!\!\perp X_b | X_c$$

Ex. 2

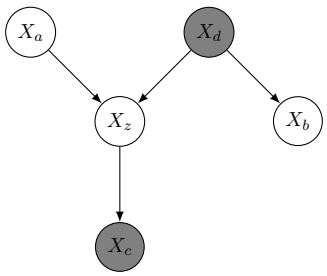
The path is blocked at X_z.



$$X_a \perp\!\!\!\perp X_b | \varnothing$$

Ex. 3

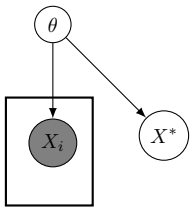
The path is blocked at X_d.



$$X_a \perp\!\!\!\perp X_b | X_d, X_c$$

Ex. 4

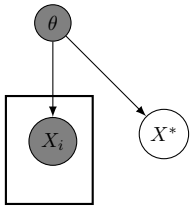
The path is unblocked at θ. We can think of it as when θ hasn't been fixed (e.g. by learning it from data), X_i and X^* are dependent through θ



$$X^* \not\perp\!\!\!\perp X_i | \varnothing$$

Ex. 5

The parth is blocked given θ. We can think of it as once we have learnt parameter θ, we can thrown away X_i, and use only θ to make predictions about unseen data X^*



$$X^* \perp\!\!\!\perp X_i | \theta$$

Machine Learning 2

Lecturer: Max Welling
Scribe: Paris Mavromoustakos, Mart van Baalen
Updated: February 23, 2015; April 08, 2016

Lecture #4
November 6, 2013

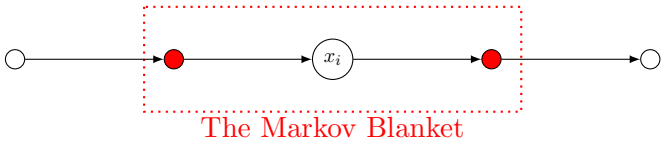
If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

Recap

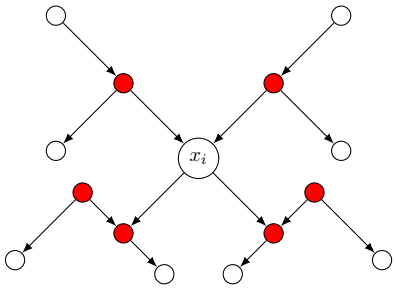
Given a DAG (Directed Acyclic Graph), the family of all $p = \prod_x p(x_i|x_{pa_{x_i}})$ that factorizes according to that DAG is the family for which all Conditional Independence Relations (CIRs) hold defined by a d-separation (the Bayes ball algorithm). This definition applies to a distribution over a **set of variables**. Note: a distribution p that satisfies more CIRs does belong to this family as well.

The Markov Blanket (MB)

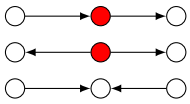
$p(x_i|x_{MB_i}, x_{rest}) = p(x_i|x_{MB_i})$ The **MB** represents the smallest set of nodes needed to make x_i independent of all the other nodes. The **MB** of x_i consists of: the parents of x_i, the children of x_i and the co-parents of the children of x_i.



Example:



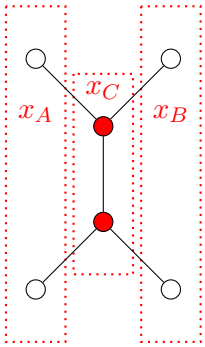
In order to determine the MB, we use the following Blocking Rules:

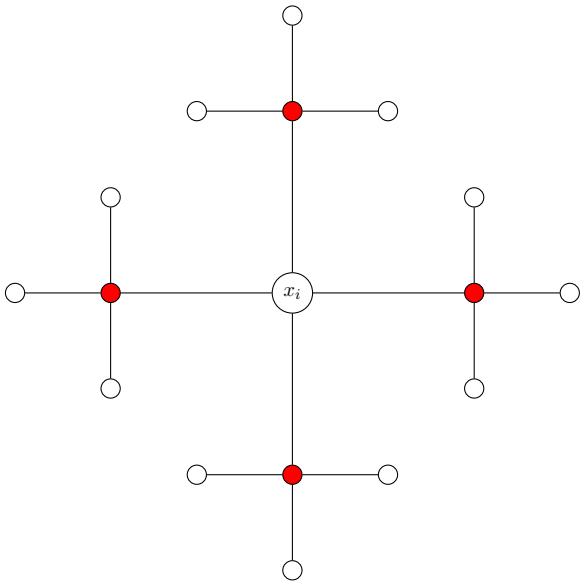


Markov Random Field (Undirected)

Try to find a path from A to B without being blocked (passing through an observed node).
$x_A \perp\!\!\!\perp x_B | x_C$ iff every path from A to B is blocked.

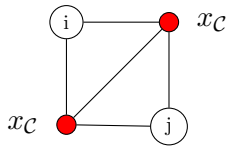
The **MB** of x_i is the set of nodes that are directly connected to x_i.



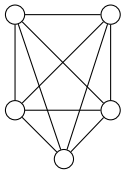


Markov Blanket of x_i

If x_i, x_j are not connected in a graph G, $\Rightarrow \exists$ CIR $p(x_i, x_j|x_\mathcal{C}) = p(x_i|x_\mathcal{C})p(x_j|x_\mathcal{C}) \Rightarrow x_i \perp\!\!\!\perp x_j|x_\mathcal{C}$, where $x_\mathcal{C}$ is not one node but a set of nodes (see example below).



Example where no CIR can exist in a graph:



There can be no CIR because each node is linked to all the remaining nodes. A set of nodes that is fully connected is called a **clique**. A **maximal clique** is a clique that cannot be made bigger (adding any node to the set does not result to a bigger clique).

The distribution represented by such graphs is the following:

$$p(x) = \frac{1}{Z} \prod_A \psi_A(x_A)$$

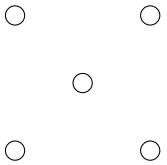
where A ranges over all maximal cliques in the graph, where the **poetntials** $\psi_A \geq 0$ and $Z := \sum_x \prod_A \psi_A(x_A)$ (called **partition function**).

Hammersley-Clifford

Given a joint distribution $p(X) > 0$ with a strictly positive density, one can use the Hammersley-Clifford theorem to represent this as an MRF. The undirected graph G of the MRF is constructed using the conditional independences that hold in $p(X)$. The potentials ψ_A can be expressed in terms of $p(X)$ using the graph G.

D-Map

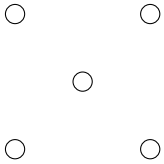
G is a D-Map of a distribution p if every CIR satisfied by p is reflected in G.



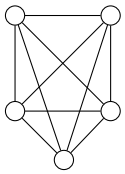
If $p(x_1, \ldots, x_5)$ is a distribution over 5 variables, then the above graph is a D-Map of p, because all of the graph's nodes are independent. Trivial case, empty graph $\overline{K}_n$ is a D-Map for any distribution p over n variables.

I-Map

G is an I-Map of p if every CIR in G is true in p.



The graph above is an I-Map of $\prod_i p(x_i)$

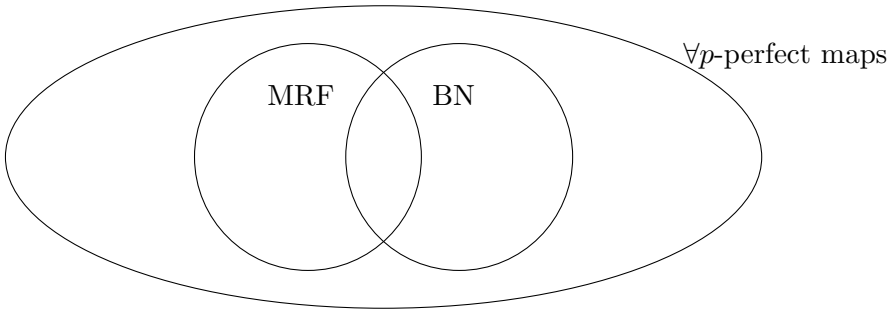


The graph above has no CIRs so it can trivially be an I-Map of any distribution p.

Perfect Map

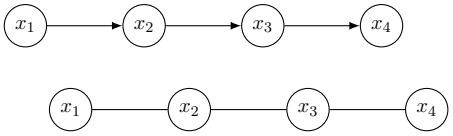
Perfect Maps of p are both D-Maps of p and I-Maps of p.

Venn diagram:



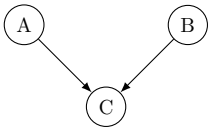
More examples

Case 1 - BN & MRF



The BN and MRF above are perfect Maps of the same distribution p.

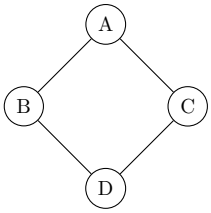
Case 2 - BN and not MRF



The above BN cannot be described by an MRF.

$x_A \perp\!\!\!\perp x_B | \varnothing$
$p(x_C|x_A, x_B)p(x_A)p(x_B) \rightarrow$ distribution

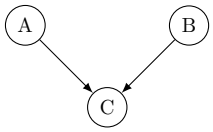
Case 3 - MRF but not BN



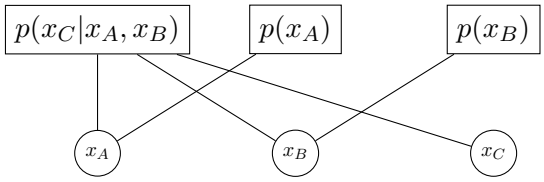
$x_A \perp\!\!\!\perp x_D | x_B, x_C$ (if I block B and C all paths from A to D are blocked). $x_B \perp\!\!\!\perp x_C | x_A, x_D$
There's no BN that can represent this MRF.

Factor Graphs

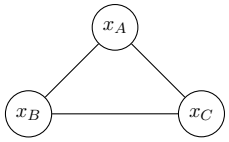
Consider the following BN:



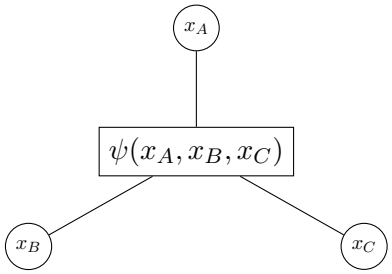
The corresponding factor graph is the following:



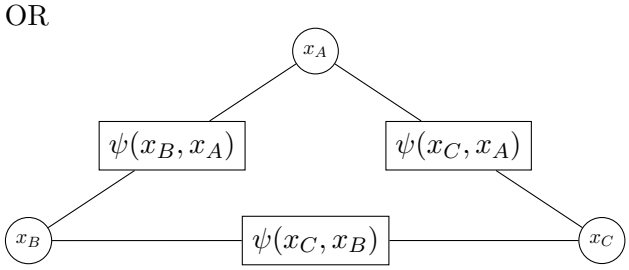
Consider the following MRF:



There are two possible ways of designing the corresponding factor graph:



OR



$\frac{1}{z}\psi(x_A, x_B, x_C)$ and $\frac{1}{z}\psi(x_A, x_B)\psi(x_B, x_C)\psi(x_C, x_A)$ both correspond with the same MRF.

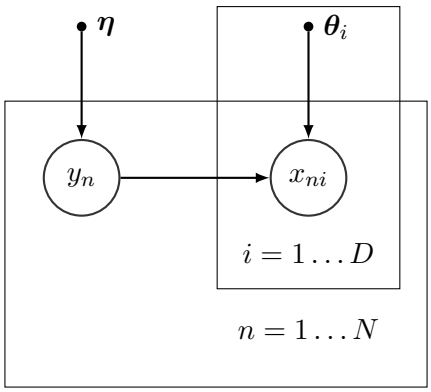


Figure 1: A graphical model representation of the Naive Bayes approach to classification.

1 Application of Graphical Models

In this section we present two examples of how Graphical Models can be used in practice.

1.1 Naive Bayes

In this subsection we describe how a Naive Bayes approach to classification can be modeled using DAGs. Consider the problem of finding a label y^* for some previously unobserved vector of features $\mathbf{x}^*$, while having previously observed the data set $\{\mathbf{x}_n, y_n\}_{n=1:N}$.

We could build a generative model, where a label is chosen from some set Y, and data is generated based on each label. We say that each datapoint $\mathbf{x}_n$ contains D features $i = 1 : D$, $\mathbf{x}_{ni}$ thus denotes feature i of datapoint n. This then implies the graphical model presented in figure 1.

This model implies an important and rather strong assumption: given a label all features are completely independent of each other.

For a single data case $\mathbf{x}$ we now have that $p(x_1, \ldots, x_D, y|\boldsymbol{\eta}, \boldsymbol{\theta}_i) = p(y|\boldsymbol{\eta}) \prod_{i=1}^{D} p(x_i|y, \boldsymbol{\theta}_i)$. The probability of the full dataset $\mathcal{D}$ is now given by $\prod_{n=1}^{N} p(y_n|\boldsymbol{\eta}) \prod_{i=1}^{D} p(x_{ni}|y_n, \boldsymbol{\theta}_i)$.

Note that this generative form is chosen for the model as the other option – data points generating labels – would imply a very highly parameterized model, as we would be considering $p(y|x_1, \ldots, x_D)$.

Finding a label y^* for a new data point $\mathbf{x}^*$ now implies finding

$$y^* = \arg\max_y \left[\ln p(y|\boldsymbol{\eta}) + \sum_{i=1}^{D} \ln p(x_i^*|y, \boldsymbol{\theta}_i) \right]$$

where ln is used for numerical stability.

1.2 Learning in Graphical Models

1.3 Learning in a Bayes Net

The probability of a set of variables $\mathbf{x}$ in a Bayes Net is defined as $p(\mathbf{x}) = \prod_i p(x_i|x_{pa_i})$. If we define a function $\theta_i(x_i, x_{pa_i})$ and use it instead of $p(x_i|x_{pa_i})$, we have $p(\mathbf{x}) = \prod_i \theta_i(x_i, x_{pa_i})$, if we constrain θ_i such that $\sum_{x_i} \theta_i(x_i, x_{pa_i}) = 1, \forall i$.

We now have that the probability of the full data set

$$
\begin{aligned}
p(\{\tilde{x}_{in}\}) &= \prod_n \prod_i \theta_i(\tilde{x}_{in}, \tilde{x}_{pa_i n}) \\
&= \prod_n \prod_i \prod_{x_i} \prod_{x_{pa_i}} \theta_i(x_i, x_{pa_i})^{\mathbb{I}[x_i = \tilde{x}_{in} \wedge x_{pa_i} = \tilde{x}_{pa_i n}]}
\end{aligned}
$$

where $\mathbb{I}(x)$ is the indicator function that returns 1 if x is true and 0 otherwise. It is important to know that x_i and x_{pa_i} indicate the *values* of variables. The expression inside the indicator function in the equation above compares the value of a variable x_i and the values of the variables that form its parents to the values of those variables in a datapoint.

The log-likelihood including Lagrange multipliers is now:

$$
\mathcal{L}(\theta, \tilde{x}) = \sum_n \sum_i \sum_{x_i} \sum_{x_{pa_i}} \mathbb{I}[x_i = \tilde{x}_{in} \wedge x_{pa_i} = \tilde{x}_{pa_i n}] \ln \theta_i(x_i, x_{pa_i}) - \sum_i \sum_{x_{pa_i}} \lambda_{i,x_{pa_i}} \left[\sum_{x_i} \theta_i(x_i, x_{pa_i}) - 1 \right]
$$

where the $\lambda_{i,x_{pa_i}}$ is the Lagrange multiplier (we will see later $\lambda_{i,x_{pa_i}}$ ensures that the distribution $\theta_i(x_i, x_{pa_i})$ is normalized). We can define a function $N(x_i, x_{pa_i})$ to be the function that counts how often the combination of values for x_i and x_{pa_i} co-occur. If we move the sum over n in the expression above inside the sum over i we get:

$$
\begin{aligned}
\mathcal{L}(\theta_i, x) &= \sum_i \sum_n \sum_{x_i} \sum_{x_{pa_i}} \mathbb{I}[x_i = \tilde{x}_{in} \wedge x_{pa_i} = \tilde{x}_{pa_i n}] \ln \theta_i(x_i, x_{pa_i}) - \sum_i \sum_{x_{pa_i}} \lambda_{i,x_{pa_i}} \left[\sum_{x_i} \theta_i(x_i, x_{pa_i}) - 1 \right] \\
&= \sum_i \sum_{x_i} \sum_{x_{pa_i}} N(x_i, x_{pa_i}) \ln \theta_i(x_i, x_{pa_i}) - \sum_i \sum_{x_{pa_i}} \lambda_{i,x_{pa_i}} \left[\sum_{x_i} \theta_i(x_i, x_{pa_i}) - 1 \right]
\end{aligned}
$$

If we now take the derivative of the log-likelihood and set it to 0 we get:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \theta_i(x_i, x_{pa_i})} &= \frac{N(x_i, x_{pa_i})}{\theta_i(x_i, x_{pa_i})} - \lambda_{i,x_{pa_i}} \\
\frac{N(x_i, x_{pa_i})}{\theta_i(x_i, x_{pa_i})} - \lambda_{i,x_{pa_i}} &= 0 \\
\theta_i(x_i, x_{pa_i}) &= \frac{N(x_i, x_{pa_i})}{\underbrace{\sum_{x_i} (x_i, x_{pa_i})}_{\lambda_{i,x_{pa_i}}}} \\
&= \frac{N(x_i, x_{pa_i})}{N(x_{pa_i})}
\end{aligned}
$$

Thus, the maximum likelihood estimation for $\theta_i(x_i, x_{pa_i})$ for values x_i and x_{pa_i} is the number of times the value x_i co-occurred with the value x_{pa_i}, divided by the number of times the value x_{pa_i} occurred.

Maximum Likelihood Learning in a Bayes net is fast because the log-likelihood **decomposes** into a sum over all variables X_i. This means that learning all parameters reduces into a collection of independent learning tasks, where each task corresponds with learning $p(x_i|x_{pa_i})$ for some i.

1.4 Learning in an MRF

From the definition of an MRF we have that the joint probability of a configuration $\mathbf{x}$ of all variables $p(\mathbf{x}) = \frac{1}{Z}\prod_A \psi_A(x_A)$, where x_A is the set of variables associated with ψ_A. The likelihood of a set of N observed configurations over these variables (i.e. datapoints) $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_N$) is

$$p(\tilde{\mathbf{x}}_i, \ldots, \tilde{\mathbf{x}}_N) = \prod_{n=1}^{N} \frac{1}{Z} \prod_A \prod_{x_A} \psi_A(x_A)^{\mathbb{I}[x_A = \tilde{x}_{An}]}$$

where the indicator function is used as before. The log-likelihood of the data is thus:

$$\begin{aligned}\mathcal{L}(\psi, x) &= \sum_{n=1}^{N} \sum_A \sum_{x_A} \mathbb{I}[x_A = \tilde{x}_{An}] \ln \psi_A(x_A) - N \ln Z \\ &= \sum_A \sum_{x_A} N(x_A) \ln \psi_A(x_A) - N \ln Z\end{aligned}$$

where we have used the same trick involving the indicator function as above. The derivative of the log-likelihood is then

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \psi_A(x_A)} &= \frac{N(x_A)}{\psi_A(x_A)} - \frac{N}{\psi_A(x_A)} \mathbb{E}_\psi \left[\mathbb{I}[x_A = \cdot]\right] \\ &= \frac{N}{\psi_A(x_A)} \left[\frac{N(x_A)}{N} - \mathbb{E}_\psi \left[\mathbb{I}[x_A = \cdot]\right]\right]\end{aligned}$$

Note that this is an exponential family function:

$$p(\tilde{x}) \propto \exp\left(\sum_A \sum_{x_A} \mathbb{I}[x_A = \tilde{x}_A] \ln \psi_A(x_A)\right)$$

and thus $\frac{\partial}{\partial \ln \psi_A} \ln Z = \mathbb{E}_\psi \left[\mathbb{I}[x_A = \cdot]\right]$, where $\mathbb{E}_\psi \left[\mathbb{I}[x_A = \cdot]\right]$ is the expected fraction of observations of x_A under model p_ψ.

Note that, in order to set the derivative of the log-likelihood to 0, the expression inside the brackets must evaluate to 0. This occurs only when, for all x_A, the expected ratio of observations of x_A equals the observed ratio of observations of x_A. We can use a sampling procedure to estimate the expected value

$$\mathbb{E}_\psi \left[\mathbb{I}[x_A = \cdot]\right] \approx \frac{N_\psi(x_A)}{N_\psi},$$

where N_ψ indicates the count under a model:

$$\frac{\partial \mathcal{L}}{\partial \psi_A(x_A)} \approx \frac{N}{\psi_A(x_A)} \left[\frac{N(x_A)}{N} - \frac{N_\psi(x_A)}{N_\psi}\right]$$

Machine Learning 2

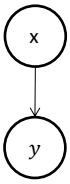
Lecturer: dr. Joris Mooij
Scribe: Ela Gati & Markus Nagel
Updated: April 19, 2016

Lecture #5
November 11, 2013

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

Inference in Graphical Models (8.4)

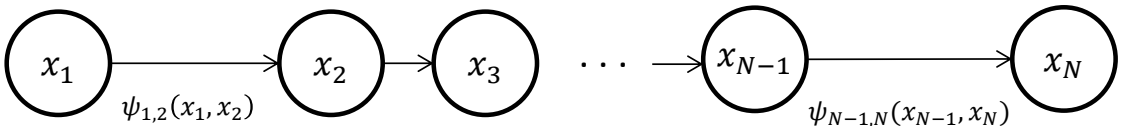
Inference in this context means calculating probabilities (joint, marginal or conditional).
Example 1:



$$p(x) = \mathcal{N}(x|\mu, \sigma^2) \tag{1}$$
$$p(y|x) = \mathcal{N}(y|ax + b, \tau^2) \tag{2}$$

By marginalizing out x we get $p(y)$ (equation 2.115 in Bishop)
Using Bayes rule we get $p(x|y)$ (equation 2.116 in Bishop)
$\Rightarrow$ by doing the inference, we can reverse the arrow of the BN.

Example 2: Markov chain



$$p(\vec{x}) = p(x_1, x_2, ..., x_N) = \frac{1}{Z}\psi_{1,2}(x_1, x_2) \cdot \psi_{2,3}(x_2, x_3) \cdot ... \cdot \psi_{N-1,N}(x_{N-1}, x_N) \tag{3}$$
$$p(x_n) = \sum_{x_1}\sum_{x_2}\cdots\sum_{x_{n-1}}\sum_{x_{n+1}}\cdots\sum_{x_{N-1}}\sum_{x_N} p(\vec{x}) \tag{4}$$
$$= \sum_{x_1}\sum_{x_2}\cdots\sum_{x_{n-1}}\sum_{x_{n+1}}\cdots\sum_{x_{N-1}}\sum_{x_N}\frac{1}{Z}\psi_{1,2}(x_1, x_2) \cdot \psi_{2,3}(x_2, x_3) \cdot ... \cdot \psi_{N-1,N}(x_{N-1}, x_N) \tag{5}$$

If each x_n takes K possible values, a naive computation will take $O(K^N)$.
In order to make it more efficient we can use the distributive law: $a(b + c) = ab + ac$.
The idea is to use this to exchange the order of the sums and products. We can rewrite the above equation as:

$$= \frac{1}{Z}\sum_{x_1}\cdots\sum_{x_{N-1}}\psi_{1,2}(x_1, x_2) \cdot \cdots \cdot \psi_{N-2,N-1}(x_{N-2}, x_{N-1})\sum_{x_N}\psi_{N-1,N}(x_{N-1}, x_N) \tag{6}$$

Repeating the same trick on all sums to the right of x_n, we get:

$$= \frac{1}{Z} \sum_{x_1} \cdots \sum_{x_{n-1}} \psi_{1,2}(x_1, x_2) \dots \psi_{n-1,n}(x_{n-1}, x_n) \cdot \underbrace{\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \sum_{x_{n+2}} \psi_{n+1,n+2}(x_{n+1}, x_{n+2}) \cdots \sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)}_{\text{We call this part the beta message, } \mu_\beta(x_n) \text{ which depends only on } x_n} \tag{7}$$

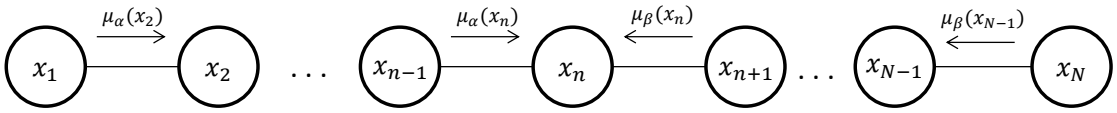
We can do the same trick to the left of x_n, and get:

$$= \frac{1}{Z} \underbrace{\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \sum_{x_{n-2}} \psi_{n-2,n-1}(x_{n-2}, x_{n-1}) \cdots \sum_{x_1} \psi_{1,2}(x_1, x_2)}_{\text{The alpha message, } \mu_\alpha(x_n) \text{ also depends only on } x_n} \cdot \mu_\beta(x_n) \tag{8}$$

Putting both together, we get:

$$p(x_n) = \frac{1}{Z} \mu_\alpha(x_n) \cdot \mu_\beta(x_n) \tag{9}$$

In each sum $\sum_{x_i} \psi_{i-1,i}(x_{i-1}, x_i)$, we sum over K values of x_i and we need to compute it K times, for each value of x_{i-1}, and we have a total of $N-1$ summations, so the computation will take $O(NK^2)$.
If we want the marginal distributions, we can naively repeat the whole process, with complexity of $O(N^2K^2)$, but actually many computations are redundant. By saving intermediate results, we can compute all marginal distributions in $O(2NK^2) = O(NK^2)$.



Recursive message passing equation:

$$\underbrace{\mu_\alpha(x_n)}_{\text{outgoing message}} = \sum_{x_{n-1}} \underbrace{\psi_{n-1,n}(x_{n-1}, x_n)}_{\text{local potential}} \cdot \underbrace{\mu_\alpha(x_{n-1})}_{\text{incoming message}} \tag{10}$$

In a similar fashion:

$$\mu_\beta(x_n) = \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \mu_\beta(x_{n+1}) \tag{11}$$

$$p(x_n) = \frac{1}{Z} \mu_\alpha(x_n) \mu_\beta(x_n) \tag{12}$$

We can compute Z locally in order $O(K)$:

$$Z = \sum_{x_n} \mu_\alpha(x_n) \mu_\beta(x_n) \tag{13}$$

This holds for any $n = 1, ..., N$ in the chain.

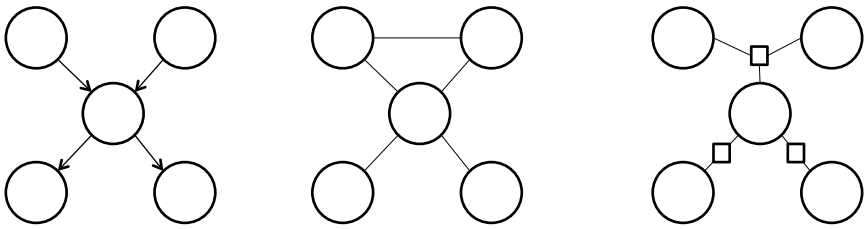


Figure 1: Three representations of the same graph. From left to right: directed graph, undirected graph and a factor graph.

Inference on Trees: the Sum-Product Algorithm (8.4.4)

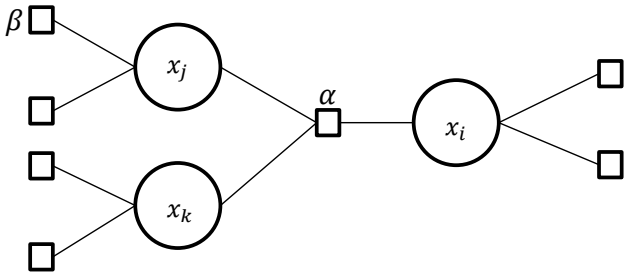
(Also known as "Belief Propagation")

A tree is a graph with no loops. Both directed and undirected trees can be converted to a factor graph tree, but a directed tree could result in a non-tree structure when converted to an undirected representation. An example is given in figure 1, were the leftmost is a directed graph, with no loops. It is called a poly-tree (and not simply a tree) since its undirected representation (middle graph) includes a loop. The factor graph representation is again a tree.

Factor graphs are the most general representation, and since any other tree representation can be easily converted to a factor tree, the sum-product algorithm is defined for factor trees.

Given a factor graph with a tree structure

As an example we use the following (part of a) factor graph:



The probability in a factor graph can be expressed as

$$\underbrace{p(\vec{x})}_{=p(x_1,x_2,\ldots,x_N)} = \frac{1}{Z}\prod_{\alpha} f_\alpha(\vec{x}_\alpha) \tag{14}$$

where $\vec{x}_\alpha$ is a vector containing all nodes dependent on α (in our example $\vec{x}_\alpha = (x_i, x_j, x_k)$). The Sum-Product Algorithm works by passing messages along the edges of the factor graph.

Factor → variable messages:

$$\mu_{\alpha\to i}(x_i) = \sum_{x_{\alpha\setminus i}} f_\alpha(\vec{x}_\alpha) \prod_{j\in\alpha\setminus i} \mu_{j\to\alpha}(x_j) \tag{15}$$

where $\alpha \setminus i$ are the indexes of the variables depending of α excluding i.
Variable $\rightarrow$ factor messages:

$$\mu_{j\to\alpha}(x_j) = \prod_{\beta\in ne(j)\setminus\alpha} \mu_{\beta\to j}(x_j) \tag{16}$$

where $ne(j) \setminus \alpha$ are all neighboring factors of j except α.
Leaf nodes:

$$x_l \text{ is a leaf node:} \qquad \mu_{l\to\delta}(x_l) = 1 \tag{17}$$

$$\epsilon \text{ is a leaf node:} \qquad \mu_{\epsilon\to k}(x_k) = f_\epsilon(x_k) \tag{18}$$

Note, $f_\epsilon(x_k) = \sum_{\vec{x}_{\epsilon\setminus k}} f_\epsilon(x_k)$. Since ϵ is a leaf node $x_{\epsilon\setminus k}$ is the empty set. By definition the sum over a empty set equals the term inside the sum.
After all messages have been computed, the Sum-Product Algorithm calculates "beliefs" for all nodes on the factor graph, which for tree-structured factor graphs are equal to the marginal probabilities of variables and factors.
"variable beliefs"

$$p(x_i) = \sum_{\vec{x}_{-i}} p(\vec{x}) = \frac{1}{Z} \prod_{\alpha\in ne(i)} \mu_{\alpha\to i}(x_i) \tag{19}$$

Note, Z is the normalization constant. This is the same for all x_i. It can easily be calculated by summing over all possible x_i, thus

$$Z = \sum_{x_i} \prod_{\alpha\in ne(i)} \mu_{\alpha\to i}(x_i) \tag{20}$$

"factor beliefs"

$$p(\vec{x}_\alpha) = \frac{1}{Z} f_\alpha(\vec{x}_\alpha) \prod_{i\in ne(\alpha)} \mu_{i\to\alpha}(x_i) \tag{21}$$

By caching intermediate computations, and calculating two messages for each edge on a tree-structured factor graph, the Sum-Product Algorithm efficiently calculates all variable marginals $p(x_i)$ in $O(2EK^M)$ (where E is the number of edges in the factor graph, and each variable x_i is assumed to have K possible values, and we assume that each factor depends on no more than M variables).

Example 3: In this example we consider the following factor graph an calculate the corresponding messages for it.

$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z} f_\alpha(x_1, x_2) f_\beta(x_2, x_3) f_\gamma(x_2, x_4) \tag{22}$$

$$\mu_{1\to\alpha}(x_1) = 1 \tag{23}$$

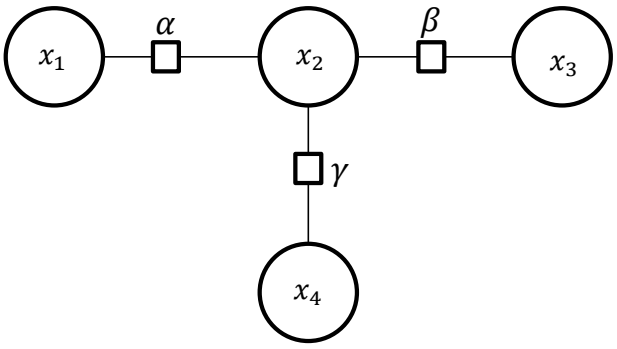
$$\mu_{\alpha\to 2}(x_2) = \sum_{x_1} f_\alpha(x_1, x_2)\mu_{1\to\alpha}(x_1) = \sum_{x_1} f_\alpha(x_1, x_2) \tag{24}$$

$$\mu_{4\to\gamma}(x_4) = 1 \tag{25}$$

$$\mu_{\gamma\to 2}(x_2) = \sum_{x_4} f_\gamma(x_2, x_4)\mu_{4\to\gamma}(x_4) = \sum_{x_4} f_\gamma(x_2, x_4) \tag{26}$$

$$\mu_{2\to\beta}(x_2) = \mu_{\alpha\to 2}(x_2)\mu_{\gamma\to 2}(x_2) \tag{27}$$

$$\mu_{\beta\to 3}(x_3) = \sum_{x_2} f_\beta(x_2, x_3)\mu_{2\to\beta}(x_2) \tag{28}$$



The same example can also be found in Bishop. The six massages in the other direction are similar. Now we can calculate the marginal probability according to equation 19.

$$p(x_2) = \frac{1}{Z}\mu_{\alpha\to 2}(x_2)\mu_{\beta\to 2}(x_2)\mu_{\gamma\to 2}(x_2) \tag{29}$$

$$= \frac{1}{Z}\left(\sum_{x_1} f_\alpha(x_1, x_2)\right)\left(\sum_{x_3} f_\beta(x_2, x_3)\right)\left(\sum_{x_4} f_\gamma(x_2, x_4)\right) \tag{30}$$

$$= \sum_{x_1}\sum_{x_3}\sum_{x_4} \underbrace{\frac{1}{Z} f_\alpha(x_1, x_2) f_\beta(x_2, x_3) f_\gamma(x_2, x_4)}_{=p(x_1,x_2,x_3,x_4)} \tag{31}$$

This is exactly the equation for marginalizing out all other variables other then x_2 from the joint distribution, hence the marginal distribution of x_2. This example is not a proof, however we can see that we get from the defined messages exactly the expected outcome.

The algorithm is defined for trees, but can actually be used even if there are loops in the graph. In this case, the convergence guarantees are very weak, but in practice it often converges, and if the convergence is rapid we can get a very good approximation for the variable beliefs.

Introducing evidence factors

What happens if some of the nodes are observed, i.e. we want to know the conditional probability $p(x_i|x_j) =$? Then we can introduce an "hard evidence" factor with the indicator function

$$f_{\xi_j}(x_j) = \mathbb{I}[x_j = \xi_j] \tag{32}$$

Thus the probability of $p(\vec{x})$ with observing $x_j = \xi_j$ gets

$$p(x_1, \ldots, x_j = \xi_j, \ldots, x_N) = \sum_{x_j} p(x_1, \ldots, x_N) f_{\xi_j}(x_j) \tag{33}$$

And the conditional probability of x_i given $x_j = \xi_j$ is

$$p(x_i|x_j = \xi_j) = \frac{p(x_i, x_j = \xi_j)}{\sum_{x_i} p(x_i, x_j = \xi_j)} \propto p(x_i, x_j = \xi_j) = \sum_{x_j} p(x_i, x_j) f_{\xi_j}(x_j) \tag{34}$$

This means that we can use the Sum-Product Algorithm on an extended factor graph that contains one evidence factor (indicator function) for each variable with evidence. Instead of multiplying with f_{ξ_j} we could redefine the existing factors according to the evidence. If f_γ is a factor dependent on x_j, we can rewrite it as:

$$\tilde{f}_\gamma = f_\gamma \cdot f_{\xi_j} \tag{35}$$

Machine Learning 2

Lecturer: Joris Mooij
Scribes: Sander Nugteren & Chiel Kooijman
Updated: February 23, 2015

Lecture #6
13 November, 2013

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

8.4.5 The Max-sum algorithm

$$\begin{aligned}
\boldsymbol{x}^* &= \arg\max_{\boldsymbol{x}} \prod_\alpha f_\alpha(\boldsymbol{x}_\alpha) \\
p(\boldsymbol{x}^*) &= \max_{\boldsymbol{x}} \frac{1}{Z} \prod_\alpha f_\alpha(\boldsymbol{x}_\alpha) \\
\log p(\boldsymbol{x}^*) &= \log \left[\max_{\boldsymbol{x}} \frac{1}{Z} \prod_\alpha f_\alpha(\boldsymbol{x}_\alpha) \right] \\
&= \max_{\boldsymbol{x}} -\log Z + \sum_\alpha \log f_\alpha(\boldsymbol{x}_\alpha)
\end{aligned}$$

The Max-Sum Algorithm is analogous to the Sum-Product Algorithm, but sums are replaced with max-operators, products with sums and factors with log-factors.
Factor $\rightarrow$ variable messages:

$$\nu_{\alpha \to i}(x_i) = \max_{x_{\alpha \setminus i}} \left[(\ln f_\alpha(\boldsymbol{x}_\alpha)) + \sum_{j \in \alpha \setminus i} \nu_{j \to \alpha}(x_j) \right]$$

Variable $\rightarrow$ factor messages:

$$\nu_{j \to \alpha}(x_j) = \sum_{\beta \in ne(j) \setminus \alpha} \nu_{\beta \to j}(x_j)$$

where $ne(j) \setminus \alpha$ are all neighboring factors of j except α.

max-beliefs/max-marginals

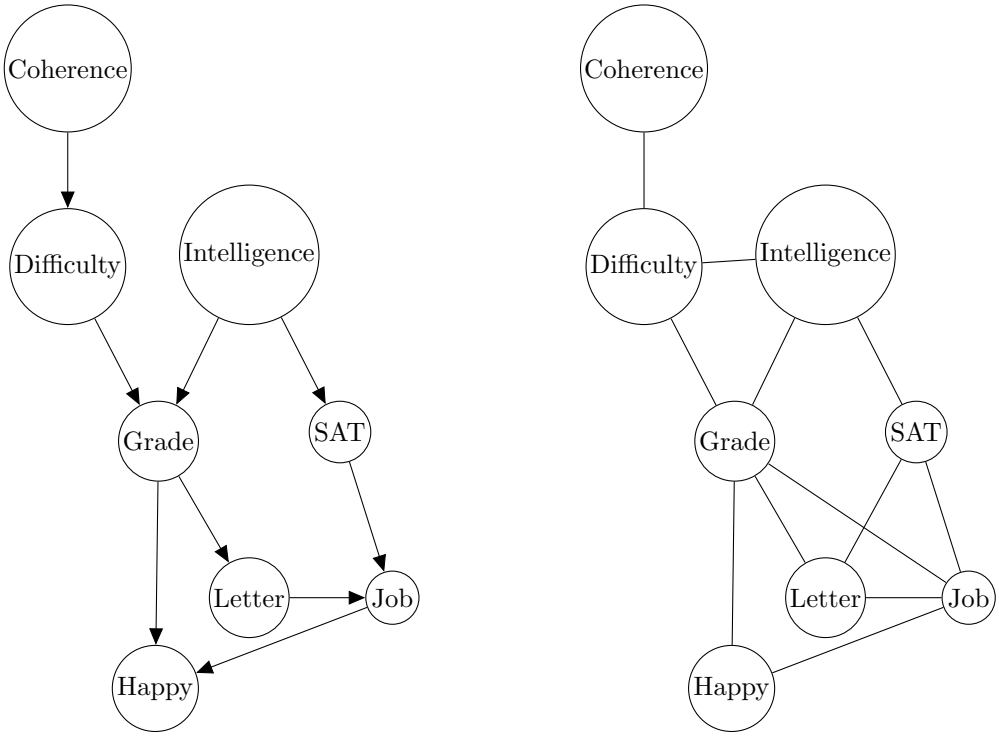
$$\begin{aligned}
q(x_i) &= \max_{\boldsymbol{x}_{\setminus i}} \log p(\boldsymbol{x}_{\setminus i}, x_i) \\
&= -\log Z + \sum_{\alpha \in ne(i)} \nu_{\alpha \to i}(x_i) \qquad\qquad ne(i) \text{ denotes the neighbours of } x_i
\end{aligned}$$

Given max-marginals you have to perform a decoding step (the Viterbi algorithm) in order to find the global optimum. If $q(x_i)$ has a unique maximum, we can use $x_i^* = \arg\max_{x_i} q(x_i)$.

8.4.6 Exact inference on general graphs

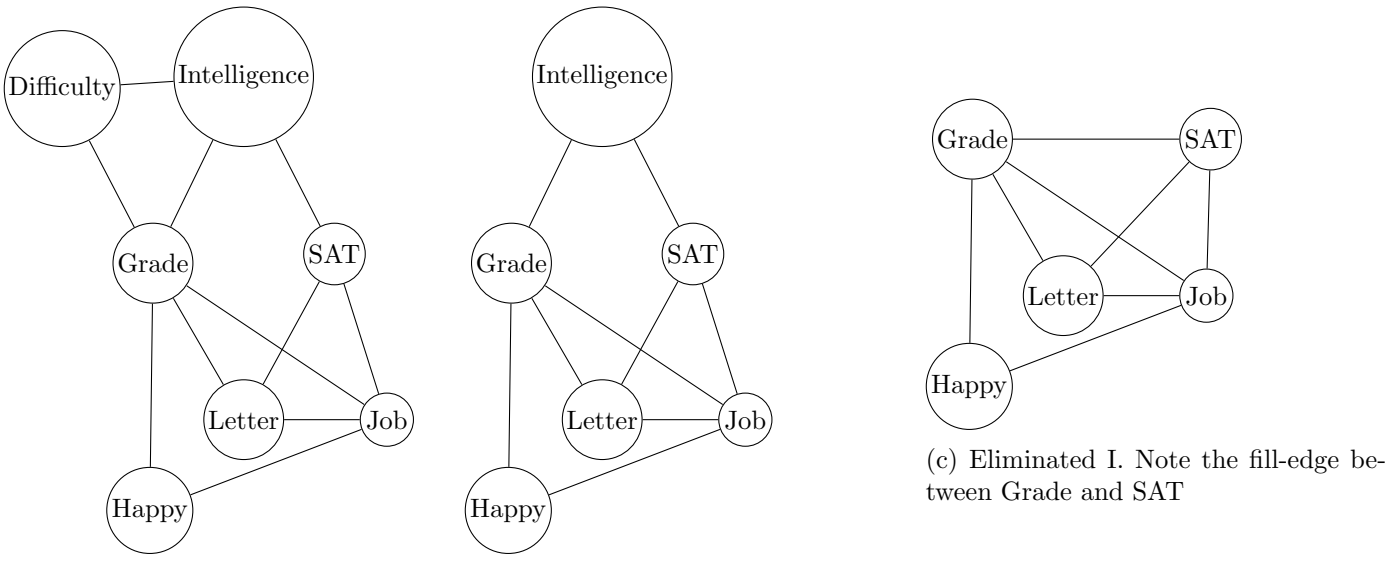
See also chapter 20 of Murphy (BB).

Variable Elimination: Example



(a) Directed graph (b) Markov random field

Figure 1: The example network



(a) Eliminated C (b) Eliminated D

(c) Eliminated I. Note the fill-edge between Grade and SAT

Figure 2: Variable elimination

Step	Eliminated	Factors used	Valiables involved	New factor	
1	C	$\phi_C,\ \phi_D$	$C,\ D$	$\tau_1(D)$	
2	D	$\phi_G,\ \tau_1$	$G,\ I,\ D$	$\tau_2(G,I)$	
3	I	$\phi_S,\ \phi_I,\ \tau_2$	$S,\ G,\ I$	$\tau_3(S,G)$	Add fill-edge between G and S
4	H	ϕ_H	$H,\ G,\ J$	$\tau_4(G,J)$	
5	G	$\phi_L,\ \tau_3,\ \tau_4$	$G,\ L,\ S,\ J$	$\tau_5(J,L,S)$	
6	S	$\phi_J,\ \tau_5$	$J,\ L,\ S$	$\tau_6(J,L)$	
7	L	τ_6	$J,\ L$	$\tau_7(J)$	

$$
\begin{aligned}
p(J) &= \sum_L \sum_S \sum_G \sum_H \sum_I \sum_D \sum_C p(C,D,I,H,G,S,L,J) \\
&= \sum_L \sum_S \sum_G \sum_H \sum_I \sum_D \sum_C \phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)\phi_S(S,I)\phi_L(L,G)\phi_J(J,L,S),\phi_H(H,G,J) \\
&= \sum_L \sum_S \phi_J(J,L,S) \sum_G \phi_L(L,G) \sum_H \phi_H(H,G,J) \sum_I \phi_I(S,I)\phi_I(I) \sum_D \phi_G(G,I,D) \sum_C \phi_C(C)\phi_D(C,D)
\end{aligned}
$$

$$
\begin{aligned}
\tau_1(D) &= \sum_C \phi_C(C)\phi_D(C,D) \\
\tau_2(G,I) &= \sum_D \phi_G(G,I,D) \sum_C \phi_C(C)\phi_D(C,D) \\
\tau_3(S,G) &= \sum_I \phi_I(S,I)\phi_I(I) \sum_D \phi_G(G,I,D) \sum_C \phi_C(C)\phi_D(C,D) \\
\tau_4(G,J) &= \sum_H \phi_H(H,G,J) \\
\tau_5(J,L,S) &= \sum_G \phi_L(L,G) \sum_H \phi_H(H,G,J) \sum_I \phi_I(S,I)\phi_I(I) \sum_D \phi_G(G,I,D) \sum_C \phi_C(C)\phi_D(C,D) \\
\tau_6(J,L) &= \sum_S \phi_J(J,L,S) \sum_G \phi_L(L,G) \sum_H \phi_H(H,G,J) \sum_I \phi_I(S,I)\phi_I(I) \sum_D \phi_G(G,I,D) \sum_C \phi_C(C)\phi_D(C,D) \\
\tau_7(J) &= \sum_L \sum_S \phi_J(J,L,S) \sum_G \phi_L(L,G) \sum_H \phi_H(H,G,J) \sum_I \phi_I(S,I)\phi_I(I) \sum_D \phi_G(G,I,D) \sum_C \phi_C(C)\phi_D(C,D)
\end{aligned}
$$

Machine Learning 2

Lecturer: Max Welling
Scribe: Marios Tzakris & Georgios Methenitis
Updated: February 23, 2015; May 2, 2016

Lecture #7
November 18, 2013

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

Variational Expectation Maximization (VEM)

Assuming we have a distribution $p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$, in which:

- $\boldsymbol{x}_n$: N observed random variables
- $\boldsymbol{z}_n$: N hidden random variables
- $\boldsymbol{\theta}$, parameters

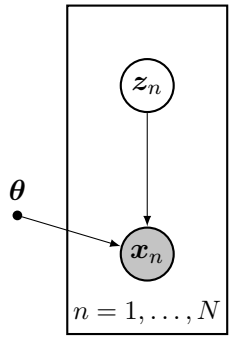


Figure 1: Model described above.

Suppose we want to learn $\boldsymbol{\theta}$ using Maximum Likelihood. The Log-Likelihood of the data is:

$$L(\boldsymbol{\theta}) = \log p(\boldsymbol{X}|\boldsymbol{\theta}) \tag{1}$$

$$= \sum_n \log p(\boldsymbol{x}_n|\boldsymbol{\theta}) \tag{2}$$

$$= \sum_n \log \left(\sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}) \right) \tag{3}$$

Due to the sum over $\boldsymbol{z}_n$ inside the logarithm, this can be difficult to optimize with respect to $\boldsymbol{\theta}$. The Expectation-Maximization (EM) algorithm comes to the rescue. It is derived by using the following trick. We want to approximate the posterior distributions $p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta})$ by distributions $q_n(\boldsymbol{z}_n)$ (for $n = 1, \ldots, N$). For arbitrary distributions $q_n(\boldsymbol{z}_n)$:

$$\log p(\boldsymbol{x}_n|\boldsymbol{\theta}) = \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q_n(\boldsymbol{z}_n)} \frac{q_n(\boldsymbol{z}_n)}{p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta})} \qquad \forall \boldsymbol{z}_n \tag{4}$$

Therefore the log-likelihood becomes:

$$L(\boldsymbol{\theta}) = \sum_n \log p(\boldsymbol{x}_n|\boldsymbol{\theta}) = \sum_n \sum_{\boldsymbol{z}_n} q_n(\boldsymbol{z}_n) \log p(\boldsymbol{x}_n|\boldsymbol{\theta}) = \sum_n \sum_{\boldsymbol{z}_n} q_n(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q_n(\boldsymbol{z}_n)} \frac{q_n(\boldsymbol{z}_n)}{p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta})} \tag{5}$$

Entropy and Kullback-Leibler Divergence:

$$H(q) = -\sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log q(\boldsymbol{z}) \tag{6}$$

$$KL[q||p] = \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log \frac{q(\boldsymbol{z})}{p(\boldsymbol{z})} \geq 0, \ (= 0 \ if \ q = p) \tag{7}$$

So eq.(1) becomes:

$$L(\boldsymbol{\theta}) = \sum_n \left(\mathbb{E}_{q_n}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})] + H[q_n] + KL[q_n(\boldsymbol{z}_n)||p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta})] \right). \tag{8}$$

Because KL is non-negative:

$$L(\boldsymbol{\theta}) \geq \sum_n \left(\mathbb{E}_{q_n}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})] + H[q_n] \right) \tag{9}$$

$$= \mathcal{L}(\boldsymbol{\theta}, q) \tag{10}$$

$L(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}, q)$ if and only if $q_n(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta})$. At the moment, we have arbitrary distributions but the idea is to make them close to the posterior probabilities.

Variational EM Algorithm

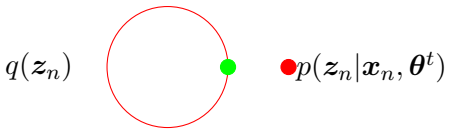
Variational EM (Expectation-Maximization) algorithm has two steps: one optimizes $\mathcal{L}(\boldsymbol{\theta}, q)$ over $\boldsymbol{\theta}$, the other over q.

- **E-Step:**
  Given $\boldsymbol{\theta}^t$, evaluate

  $$q_n^t(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta}^t) \tag{11}$$

  (or increase $\mathcal{L}(\boldsymbol{\theta}^t, q)$ over q). This is not easy to solve, but sometimes it can be done (for example in Mixture of Gaussians).

  It could be a situation where:

  The green dot upon the circle will be the best q_n that approximates $p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta}^t)$ as it is the shortest point from p.

- **M-Step:**
  Given q^t, solve

  $$\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}^t} \sum_n \mathbb{E}_{q_n^t}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)] \tag{12}$$

  (or increase $\mathcal{L}(\boldsymbol{\theta}, q^t)$ over $\boldsymbol{\theta}$).

We maximize the lower bounds until we find the maximum. The convergence is rather slow. In terms of speed is not the best algorithm.

This Variational EM algorithm is a slight generalization of the EM algorithm discussed by Bishop. The E-step of the EM algorithm is to calculate the expectation:

$$\sum_n \mathbb{E}_{q_n^t}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)]$$

The M-step of the EM algorithm is to maximize that expectation over $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}^t} \sum_n \mathbb{E}_{q_n^t}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)]$$

Example: mixture of Bernoulli's

Consider a multivariate Bernoulli distribution

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{i=1}^{D} \mu_i^{x_i}(1-\mu_i)^{1-x_i}, \mu_i \in [0,1], \;\; x_i \in \{0,1\} \tag{13}$$

where $\mathbf{x} = (x_1, \ldots, x_D)$ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_D)$. Consider a mixture of these distributions

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\boldsymbol{\mu}_k) \tag{14}$$

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

We introduce a hidden variable: $z \in \{1, 2, \ldots, K\}$. Assuming each datapoint is assigned to one cluster, the hidden variable z tells us to which cluster each datapoint is assigned.

$$p(z|\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{\delta_{z,k}}$$

$$p(\mathbf{x}|z, \boldsymbol{\mu}) = \prod_{k=1}^{K} p(\mathbf{x}|\boldsymbol{\mu}_k)^{\delta_{z,k}}$$

It is easy to see that

$$\sum_z p(\mathbf{x}|z, \boldsymbol{\mu}) p(z|\boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\boldsymbol{\mu}_k) = p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\pi})$$

Rewrite log-likelihood as

$$L(\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_n \log \sum_{k=1}^{K} \pi_k p(\mathbf{x}_n|\boldsymbol{\mu}_k) \tag{15}$$

$$= \sum_n \log \sum_{z_n} p(\mathbf{x}_n|z_n, \boldsymbol{\mu}) p(z_n|\boldsymbol{\pi}) \tag{16}$$

The EM functional is:

$$\mathcal{L}(q, \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_n \sum_{z_n} q_n(z_n) \log p(\mathbf{x}_n, z_n | \boldsymbol{\mu}, \boldsymbol{\pi}) - \sum_n \sum_{z_n} q_n(z_n) \log q_n(z_n) \tag{17}$$

$$= \sum_n \sum_{z_n} q_n(z_n) \left[\log \pi_{z_n} + \left(\sum_{i=1}^{D} x_{ni} \log \mu_{z_n,i} + (1 - x_{ni}) \log(1 - \mu_{z_n,i}) \right) - \log q_n(z_n) \right] \tag{18}$$

Including Lagrange multipliers for the constraints:

$$\tilde{\mathcal{L}}(q, \boldsymbol{\mu}, \boldsymbol{\pi}, \lambda, \lambda_n) = \mathcal{L}(q, \boldsymbol{\mu}, \boldsymbol{\pi}) + \lambda \left(\sum_k \pi_k - 1 \right) + \sum_n \lambda_n \left(\sum_{z_n} q_n(z_n) - 1 \right) \tag{19}$$

E-Step:

$$\frac{\partial \tilde{\mathcal{L}}}{\partial q_n(z_n)} = \log \pi_{z_n} + \left[\sum_{i=1}^{D} x_{ni} \log \mu_{z_n,i} + (1 - x_{ni}) \log(1 - \mu_{z_n,i}) \right] - \log q_n(z_n) - 1 + \lambda_n = 0 \tag{20}$$

$$q_n(z_n) = \exp(\lambda_n - 1) \, \pi_{z_n} \prod_i \mu_{z_n,i}^{x_{ni}} (1 - \mu_{z_n,i})^{1 - x_{ni}} \tag{21}$$

M-Step:

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \pi_k} = \sum_n \sum_{z_n} \delta_{z_n,k} \frac{q_n(z_n)}{\pi_k} + \lambda = 0 \Rightarrow$$

$$\pi_k = -\frac{1}{\lambda} \sum_n q_n(k) \Rightarrow \boxed{\frac{N_k}{N} = \pi_k}$$

Finally:

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \mu_{ki}} = \sum_n q_n(k) \left(\frac{x_{ni}}{\mu_{ki}} - \frac{(1 - x_{ni})}{(1 - \mu_{ki})} \right) = 0 \Rightarrow \tag{22}$$

$$\boxed{\mu_{ki} = \frac{\sum_n q_n(k) x_{ni}}{N_k}} \tag{23}$$

which is the average over all datapoints assigned to each cluster.

Variational Inference

Instead of treating $\boldsymbol{\theta}$ as parameter, let us treat it as a hidden random variable and calculate its posterior.

We are interested in the evidence $p(\boldsymbol{X})$ and the posterior $p(\boldsymbol{\theta}|\boldsymbol{X})$:

$$L = \log p(\boldsymbol{X}) \geq \mathcal{L}(q) \tag{24}$$

(in which the equality holds when $q(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{X})$), with:

$$\mathcal{L}(q) = \int q(\boldsymbol{\theta}) \log(p(\boldsymbol{X}|\boldsymbol{\theta}) p(\boldsymbol{\theta})) \, \mathrm{d}\boldsymbol{\theta} - \int q(\boldsymbol{\theta}) \log q(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \tag{25}$$

Let's now assume distribution q factorizes over parameters θ_i:

$$q(\boldsymbol{\theta}) = \prod_{i=1}^{D} q_i(\theta_i) \tag{26}$$

Together with Lagrange multipliers ensuring normalization of $q_i(\theta_i)$:

$$\tilde{\mathcal{L}}(q) = \int \left(\prod_{i=1}^{D} q_i(\theta_i) \right) \log p(\boldsymbol{X}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} - \sum_i \int q_i(\theta_i) \log q_i(\theta_i) \mathrm{d}\theta_i + \sum_i \lambda_i \left(\int q_i(\theta_i) \mathrm{d}\theta_i - 1 \right)$$

Maximizing the bounds separately for each term (note: this is a functional derivative, see App. D in Bishop):

$$\frac{\partial \tilde{\mathcal{L}}}{\partial q_i(\theta_i)} = \int \left(\prod_{j \neq i} q_j(\theta_j) \right) \log(p(\boldsymbol{X}|\boldsymbol{\theta}) p(\boldsymbol{\theta})) \, \mathrm{d}\boldsymbol{\theta}_{\setminus i} - \log q_i(\theta_i) - 1 + \lambda_i \tag{27}$$

The update becomes:

$$\begin{aligned} q_i(\theta_i) &= \exp(\lambda_i - 1) \exp \int \left(\prod_{j \neq i} q_j(\theta_j) \right) \log(p(\boldsymbol{X}|\boldsymbol{\theta}) p(\boldsymbol{\theta})) \, \mathrm{d}\boldsymbol{\theta}_{\setminus i} & (28) \\ &\propto \exp\left(\mathbb{E}_{q_{\setminus i}}[\log p(\boldsymbol{X}, \boldsymbol{\theta})] \right) & (29) \end{aligned}$$

where: $\exp(\lambda_i - 1) = \dfrac{1}{Z}$, ensures $q_i(\theta_i)$ is correctly normalized.
This is called **variational Bayes**: $p(\boldsymbol{\theta}|\boldsymbol{X}) \approx \prod_i q_i(\theta_i)$

Example of Variational Bayes

Consider the following Bayesian model for a Gaussian distribution:

$$\begin{aligned} p(X|\mu, \tau) &= \prod_n \mathcal{N}(x_n|\mu, \tau^{-1}) = \left(\frac{\tau}{2\pi} \right)^{\frac{N}{2}} \exp\left(\frac{-\tau}{2} \sum_n (x_n - \mu)^2 \right) & (30) \\ p(\tau) &= \mathrm{Gam}(\tau|a_0, b_0) = c\tau^{a_0 - 1} e^{-b_0 \tau} & (31) \\ p(\mu|\tau) &= \mathcal{N}(\mu|\mu_0, (\lambda_0 \tau)^{-1}) & (32) \end{aligned}$$

Let's assume now that $q(\tau, \mu) = q(\tau)q(\mu)$, $\boldsymbol{\theta} = (\tau, \mu)$

$$\begin{aligned} \mathcal{L}(q_\mu, q_\tau) = &\int q_\mu(\mu) q_\tau(\tau) \log(p(X|\mu, \tau) p(\mu|\tau) p(\tau)) \mathrm{d}\mu \mathrm{d}\tau \\ &- \int q_\mu(\mu) \log q_\mu(\mu) \mathrm{d}\mu - \int q_\tau(\tau) \log q_\tau(\tau) \mathrm{d}\tau \end{aligned}$$

Adding Lagrange multipliers:

$$\tilde{\mathcal{L}}(q_\mu, q_\tau) = \mathcal{L}(q_\mu, q_\tau) + \lambda_\mu \left(\int q_\mu(\mu) \mathrm{d}\mu - 1 \right) + \lambda_\tau \left(\int q_\tau(\tau) \mathrm{d}\tau - 1 \right)$$

Hence,

$$\frac{\partial \tilde{\mathcal{L}}}{\partial q_\mu(\mu)} = \int \mathrm{d}\tau \; q(\tau) \log p(X, \mu, \tau) - \log q(\mu) - 1 + \lambda_\mu \tag{33}$$

Set $\dfrac{\partial \tilde{\mathcal{L}}}{\partial q_\mu(\mu)} = 0$, we find

$$\begin{aligned} q_\mu(\mu) &= \frac{1}{Z} \exp\left(\int q(\tau) \log p(X, \mu, \tau) \, \mathrm{d}\tau \right) \\ &= \mathcal{N}\left(\frac{\lambda_0 \mu_0 + N\bar{x}}{\lambda_0 + N}, (\lambda_0 + N)\mathbb{E}_{q_\tau}[\tau] \right) \end{aligned}$$

Similarly:

$$q_\tau(\tau) = \mathrm{Gam}\left(\tau | a_0 + \frac{N}{2}, b_0 + \frac{1}{2}\mathbb{E}_{q_\mu}\left[\sum_n (x_n - \mu)^2 + \lambda_0(\mu - \mu_0)^2 \right] \right) \tag{34}$$

and, after solving these fixed point equations, one obtains the approximation

$$\begin{aligned} p(\boldsymbol{\theta}|X) &\approx q(\mu) q(\tau) \\ \log p(X) &\approx \mathcal{L}(q_\mu, q_\tau) \end{aligned}$$

Machine Learning 2

Lecturer: Max Welling
Scribe: Sammie Katt
Updated: May 2, 2016

Lecture #8
November 20, 2013

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

1 Variational EM (recap)

We have seen that variational EM consists of a two-step algorithm for maximizing log-likelihood function $\ln p(X|\theta)$, which is rewritten as in equation.

$$\ln p(X|\theta) = -\mathbb{E}_q[\ln p(X, Z|\theta)] + H[q] + KL[q||p(Z|X, \theta)] \tag{1}$$

These steps involve maximizing $\mathcal{L}$:

1. E-step: maximize over $q(Z)$ while holding θ fixed
2. M-step: maximize over θ while holding $q(Z)$ fixed

2 Variational Bayes (recap)

Variational Bayes approach considers θ parameters as latent variables (Z). We are interested in evidence $p(X)$ and the posterior $p(\theta|X)$ (eq 2).

$$\ln p(X) = -\mathbb{E}_q[\ln p(X, \theta)] + H[q] + KL[q||p(\theta|X)] \tag{2}$$

Assuming q factorizes $q(\theta) = \prod_i q_i(\theta_i)$ we get to the following update rules

$$q_i(\theta_i) \propto \exp\left(\mathbb{E}_{q \setminus q_i}[\ln p(X, \theta)]\right)$$

3 Hybrid

Although not further discussed in college a hybrid form exists, combining both parameters θ and hidden variables Z. We still try to calculate our evidence $p(X)$ (as we did in variational bayes) but we introduce hidden variables (eq 3).

$$\ln p(X) = \underbrace{\mathbb{E}_q\left[\ln p(X, Z, \theta)\right] + H[q_Z] + H[q_\theta]}_{\mathcal{L}(q_z, q_\theta)} + KL[q_Z q_\theta||p(Z, \theta|X)] \tag{3}$$

where

$$q = q(Z|X) \times q(\theta)$$

The EM steps are a combination of variational EM and variational Bayes:

1. $\mathbb{E}_z$-step: $\max_{q_z} \mathcal{L}(q_z, q_\theta)$
2. $\mathbb{E}_\theta$-step: $\max_{q_\theta} \mathcal{L}(q_z, q_\theta)$

4 Types of maximizing

4.1 Maximizing with EM

A direct consequence the way (variational) EM maximizes is the underestimation of the variance of the approximated distribution. EM maximizes the lower bound by minimizing $KL[q(\theta)|||p(\theta|X)]$. As we see in equation 4, taking an outlier would lead to a small $p(\theta)$ (taken 0 as an extreme example). If $q(\theta) > p(\theta)$ this could easily lead to a high number and thus is penalized. Thus minimizing leads directly to penalizing high variance and underestimates the variance.

$$\begin{aligned} KL[q(\theta)||p(\theta)] &= \int q(\theta) \ln p(\theta) - q(\theta) \ln q(\theta) d\theta \\ &= \int q(\theta) \ln 0 - q(\theta) \ln q(\theta) d\theta \\ &= \int q(\theta) \times -\infty - q(\theta) \ln q(\theta) d\theta \end{aligned} \tag{4}$$

4.2 Maximizing with EP (expectation propagation)

Though to be described later, EP minimizes $KL[p(\theta|X)||q(\theta)]$ and, as expected, overestimates the variance. This is shown in derivation 5, which assumes $q(\theta) \in$ exponential family $(\frac{1}{Z(\eta)} \exp(\sum_k \eta_k \phi_k(\theta))$.

$$\begin{aligned} KL[p(\theta)||q(\theta)] &= -\int p(\theta) \left(-\ln Z(\eta) + \sum_k \eta_k \phi_k(\theta) \right) \mathrm{d}\theta - H[p] \\ &= \ln Z(\eta) - \sum_k \eta_k \mathbb{E}_p[\phi_k(\theta)] - H[p] \end{aligned}$$

minimize by setting derivative to 0

$$\begin{aligned} \frac{\delta KL}{\delta \eta_j} &= \mathbb{E}_q(\phi_j(\theta)) - \mathbb{E}_p[\phi_j(\theta)] \\ \mathbb{E}_q(\phi_j(\theta)) &= \mathbb{E}_p[\phi_j(\theta)] \end{aligned} \tag{5}$$

The result 5 tells us that by moment matching (match the parameters of two distributions) we find our desired $q(\theta)$. As we are dealing with an exponential family, the μ and Σ are found as follows:

$$\begin{aligned} \mu &= \int \theta p(\theta) \\ \Sigma &= \int \theta \theta^T p(\theta) \end{aligned}$$

If $p(\theta)$ is irregular in shape (which is often is in real data), $q(\theta)$ will try to match its variance and ends up becoming larger in order to catch the irregular shape of $p(\theta)$.

5 Expectation propagation

We are, again, interested in $p(\theta|X)$ (posterior for prediction) and $p(X)$ (for model evaluation). We assume these distributions factorize in some mixture of independent factors (for example nodes in a graph or data points)

In BN:

$$p(X,\theta) = \prod_j f_j(\theta) = \prod_k p(x_k|x_{pa_k},\theta)$$

In MRF:

$$p(X,\theta) = \prod_j f_j(\theta) = \frac{1}{Z_j}\prod_k \psi_k(\theta)$$

The posterior is given by

$$p(\theta|X) = \frac{1}{p(X)}\prod_j f_j(\theta)$$

and the model evidence is given by

$$p(X) = \int \prod_j f_j(\theta)\mathrm{d}\theta$$

We approximate the posterior by a product of factors

$$q(\theta) = \frac{1}{Z}\prod_j \widetilde{f}_j(\theta)$$

6 EP method (Skipped in 2016)

Ideally we would like to minimize equation 6 (remember KL of EP is different from that of EM), but for that we require the moments of $p(X)$ (which we are trying to approximate as we do not know these!).

$$\min_\theta KL\left[\frac{1}{p(D)}\prod_i f_i(\theta)||\frac{1}{Z}\prod_i \widetilde{f}_i(\theta)\right] \tag{6}$$

Instead, we iterate over each $\widetilde{f}_j(\theta)$ and minimize the difference between this one with respect to the original $f_j(\theta)$ while fixing all other $\widetilde{f}_{i\neq j}$ (derivation 7).

$$\begin{aligned} q^{\text{new}}(\theta) &\propto \frac{1}{Z_j}\widetilde{f}_j(\theta)q^{\setminus j}(\theta) \\ &\approx f_j(\theta)q^{\setminus j}(\theta) \\ &\rightarrow KL\left[\frac{f_j(\theta)\prod_{i\neq j}\widetilde{f}_i(\theta)}{Z_j}||q^{\text{new}}(\theta)\right] \end{aligned} \tag{7}$$

where

$$\begin{aligned} q^{\setminus j} &= \frac{q(\theta)}{\widetilde{f}_j(\theta)} \\ Z_j &= \int f_j(\theta)q^{\setminus j}(\theta)\delta\theta \end{aligned}$$

6.1 Algorithm

These steps conclude in the following steps for doing EP (do this iteratively over all j):

1. Choose a factor $\widetilde{f}_j(\theta)$ to refine
2. Remove $\widetilde{f}_j(\theta)$ from the posterior by division

$$q^{\setminus j}(\theta) = \frac{q(\theta)}{\widetilde{f}_j(\theta)}$$

3. Match the moments of $q^{\text{new}}(\theta)$ to those of $q^{\setminus j}(\theta) f_j(\theta)$, including evaluation of the normalizing constant Z_j
4. Evaluate and store the new factor

$$\widetilde{f}_j(\theta) = Z_j \frac{q^{\text{new}}(\theta)}{q^{\setminus j}(\theta)}$$

Lastly $p(\theta|X) = \frac{1}{Z}\prod_j \widetilde{f}_j(\theta)$. If applied on MRF this becomes the loopy believe algorithm. Additionally, EP is not guaranted to converge.

7 EP on evidence

The same trick is applied for calculating evidence:

$$p(X) = \int \prod_j f_j(\theta)\delta\theta \approx \int \prod_j \widetilde{f}_j(\theta)\delta\theta$$

which leads to the same calculations as in the section above, (see 10.202-10.208 Bishop)

Fundamental Lemma of Variational Inference

Consider model distribution $p(x, z)$ (dep. on parameters θ)
where value of x might be known/observed, z unknown.

Consider another model distribution $q(z)$ (dep. on x, and other parameters λ)
meant to approximate conditional $p(z|x)$ ("posterior")
(which is usually difficult to compute)

We are interested in

i) evidence lower bound (ELBO)

$$\mathcal{L}_x(q|p) := H(q) + \mathbb{E}_{q(z)}[\log p(x,z)]$$

ii) $\underbrace{\log\text{-likelihood}}_{\text{(frequentist)}} / \underbrace{\text{evidence}}_{\text{(Bayesian)}}$:

$$\mathcal{L}_x(p) := \log p(x)$$

We then always have:

$$\mathcal{L}_x(p) \geq \mathcal{L}_x(q|p)$$

Equality holds if and only if:

$$q(z) = p(z|x) \quad \forall z$$

proof we have:

$$0 \leq KL(q(z) \| p(z|x)) \quad \text{with "=" iff } q(z) = p(z|x) \ \forall z.$$

but

$$\int q(z) \log \frac{q(z)}{p(z|x)}\, dz = \int q(z) \log \frac{p(x)\cdot q(z)}{p(z,x)}$$

$$= \int q(z) \log q(z)\, dz + \int q(z) \log p(x)\, dz - \int q(z) \log p(z,x)\, dz$$

$$= -H(q) - \mathbb{E}_{q(z)}[\log p(z,x)] + \log p(x)$$

$$= -\mathcal{L}(q|p) + \mathcal{L}(p)$$

□

Variational Autoencoders

model : $p(x, z|\theta) = p(x|z,\theta_1) \cdot p(z|\theta_2)$ ⇒ neural network
(decoder) (prior)

model : $q(z|x,\lambda)$ (encoder)

evidence: $\mathcal{L}(\theta) := \log p(x|\theta)$ ✓ data want $\max_\theta \mathcal{L}(\theta)$

ELBO : $\mathcal{L}(\theta,\lambda) := H(q(z|x,\lambda)) + \mathbb{E}_{q(z|x,\lambda)}\left[\log p(x|z,\theta_1)\cdot p(z|\theta_2)\right]$

since $\mathcal{L}(\theta) \geq \mathcal{L}(\theta,\lambda)$

we instead optimize:

$$\max_{\theta,\lambda} \mathcal{L}(\theta,\lambda)$$ with SGD

we further approximate:

$$\hat{\mathcal{L}}(\theta,\lambda) := H(q(z|x,\lambda)) + \frac{1}{K}\sum_{z^{(1)},\dots z^{(K)} \sim q(z|x,\lambda)} \left[\log p(x|z^{(k)},\theta_1)\, p(z^{(k)}|\theta_2)\right]$$

(minibatch)

↳ reparametrization trick / stochasticity in z.

$$\max_{\theta,\lambda} \hat{\mathcal{L}}(\theta,\lambda)$$

Train

- sample minibatch $x^{(1)},\dots x^{(M)}$ from data ($p(x) = \text{true}$)
- ↳ (compute $q(z|x^{(i)},\lambda^t)$ (parameters of))
- for $i=1\dots M$ sample $k=1\dots K$ (reparametrization trick)
  $z^{(i,k)} \sim q(z|x^{(i)},\lambda^{(t)})$
- compute $\hat{\mathcal{L}}(\theta^t,\lambda^t)$
- compute $\nabla_{\theta,\lambda}\hat{\mathcal{L}}(\theta^t,\lambda^t)$ on all $x^{(i)},\dots$, $z^{(i,k)}$, $i=1\dots M$, $k=1\dots K$
- → Backpropagation:

$$\theta^{t+1} = \theta^t + \alpha\cdot\nabla_\theta\hat{\mathcal{L}}(\theta^t,\lambda^t)$$
$$\lambda^{t+1} = \lambda^t + \alpha\,\nabla_\lambda\hat{\mathcal{L}}(\theta^t,\lambda^t)$$

repeat. α-learning rate

low-dim ↝ Latent representation in z-space of the data.

Machine Learning 2

Lecturer: Max Welling
Scribe: Lydia Mennes & Auke Wiggers
Updated: May 2, 2016

Lecture # 9
November 25, 2013

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

Recap of what we've done so far

Inference in GMs (frequentist approach) We estimate parameters from the data.

$$p_\theta(z|x) = p(z|x,\theta)$$
$$p(X|\theta) = \sum_z p(x,z|\theta)$$

Inference in GMs (Bayesian approach) Parameter is another hidden variable, use $p(z,\theta|x)$.

$$p(x) = \sum_z \int p(x,z,\theta)d\theta$$

Variational EM (frequentist approach) (See lecture nr. 7)

Variational Bayes (Bayesian approach) Performs inference over parameters. (See lecture nr. 7)

Expectation Propagation (See lecture nr. 8)

Belief propagation As well as loopy belief propagation and variable elimination algorithm

All of the above methods are deterministic! Of course, they may contain stochastic methods (e.g., random initialisation for loopy belief prop.) but in general the outcome will always be the same. All methods also compute the full distribution $q(z_k|x)$: they optimize to find q.

Monte Carlo methods

The goal is to compute $\mathbb{E}_p[f] = \int p(z)f(z)dz$. This can be used for prediction:

$$p(y^*|x^*) = \int p(y^*|x^*,\theta)p(\theta|X,Y)d\theta$$

Or for the estimation of evidence:

$$p(Y|X) = \sum_Z \int p(Y,Z|X,\theta)p(\theta)d\theta$$

What would we do if $p(\theta|X,Y)$ or $p(\theta)$ is difficult to compute? An alternative to the methods mentioned above is sampling.

Regular sampling

For regular sampling you draw N samples from the distribution p:

$$z_i \sim p(z) \qquad i = 1, ..., N$$

Then:

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{i=1}^{N} f(z_i)$$

The idea is that you draw the function values proportional to p, therefore the average approximates $\int p(z)f(z)dz$. Such an estimate is called a Monte Carlo estimate and the used notation is $\langle f \rangle$. If ∞ Monte Carlo estimates are made, the average equals $\mathbb{E}[f]$.

The expected value of $\langle f \rangle$ is:

$$\mathbb{E}[\langle f \rangle] = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}[f] = \mathbb{E}[f]$$

Since $\mathbb{E}[\langle f \rangle]$ is the same as $\mathbb{E}[f]$, $\langle f \rangle$ is an unbiased estimate. The expected error of $\langle f \rangle$ is:

$$\begin{aligned}
\mathbb{E}\big[(\langle f \rangle - \mathbb{E}f)^2\big] &= \mathbb{E}\big[(\langle f \rangle - \mathbb{E}\langle f \rangle)^2\big] \\
&= \mathbb{V}(\langle f \rangle) \\
&= \mathbb{V}\left(\frac{1}{N} \sum_i f(z_i)\right) \\
&= \frac{1}{N^2} \sum_i \mathbb{V}\big(f(z_i)\big) \\
&= \frac{1}{N} \mathbb{V}(f)
\end{aligned}$$

It can be seen that large N reduces the expected error of $\langle f \rangle$, the magnitude of the error is order $O(\frac{1}{\sqrt{N}})$.

Regular sampling for discrete random variables

For regular sampling for discrete random variables a cumulative probability distribution (CDF) is needed for the distribution p that you wish to sample from. An example can be seen in Figure 2. The procedure for regular sampling in this case is:

- Draw a value from a uniform distribution: $u \sim U(0, 1)$.
- See where the value is located on the y-axis and note state k that is associated with the value as your sample.

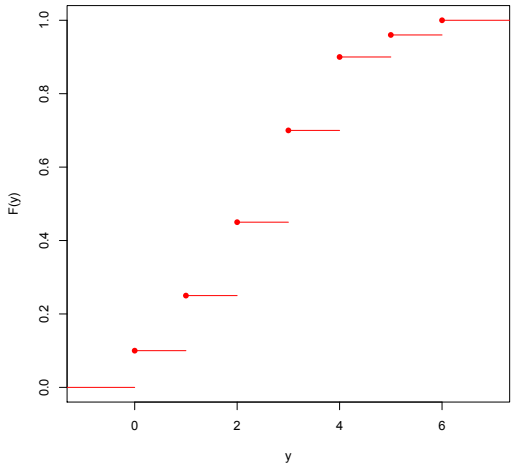


Figure 1: Cumulative probability distribution of a discrete random variable with $K = 7$ states

Regular sampling for continuous random variables

For regular sampling for continuous variables the CDF is defined as:

$$\begin{aligned} F(x) &= \int_{-\infty}^{x} p(z)dz \\ &= p(z \leq x) \end{aligned}$$

The procedure for regular sampling is than as follows:

- Draw a value from a uniform distribution: $u \sim U(0, 1)$.
- Get your sample: $x_i \sim F^{-1}(u_i)$

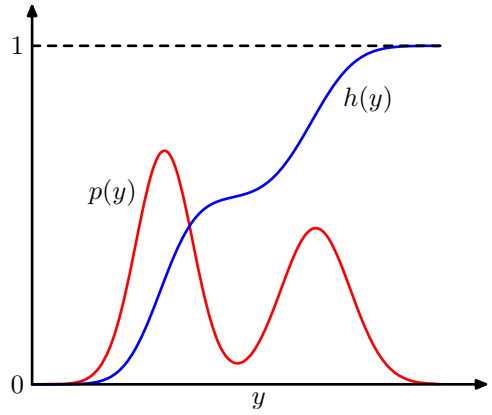


Figure 2: Cdf of continuous random variable

Rejection sampling

If the exact CDF is not available a different approach is needed and Rejection sampling is one option. The procedure for rejection sampling is as follows:

- Use a (unnormalized) distribution $\tilde{p} \propto p$, such that there is no need to use the actual distribution p. Since we are only interested in the sample and the ratios stay intact it is valid to use $\tilde{p}$.
- Use some (unnormalized) distribution $\tilde{q}$ that upperbounds $\tilde{p}$ as tight as possible (otherwise you reject too many samples) but that has finite normalization constant, i.e., $\int \tilde{q}(z)dz < \infty$. See Figure 3.
- Sample $z_i \sim q$
- Sample $u_i \sim U(0, \tilde{q}(z_i))$
- If $u_i > \tilde{p}(z_i)$ the sample is trashed, otherwise the sample is kept.

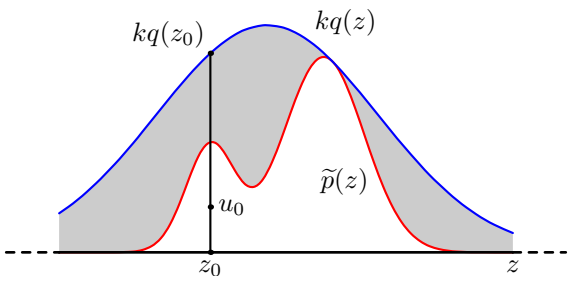


Figure 3: Distributions $\tilde{p}$ and $\tilde{q}$

Since the probability density of a sample is $q(z)$ and the probability of accepting the sample is $\frac{\tilde{p}(z)}{\tilde{q}(z)}$ it can easily be seen that this results in a correct sample. The density of accepted samples is $\propto q(z)\frac{\tilde{p}(z)}{\tilde{q}(z)} \propto p(z)$.

With a large number of dimensions a large amount of volume is present between $\tilde{p}$ and $\tilde{q}$, even if the fit is as tight as possible. Because of this most samples are rejected: the curse of dimensionality.

Adaptive rejection sampling

Read Section 11.1.3 in Bishop for this topic. Skip the equations in the book as they only complicate things, but understand the concept.

Importance sampling

Used when it is not easy to find a function that forms an upper bound. Note that q does not necessarily has to be a bound on p, as we weigh by the quotient of the two distributions.

Assume that

$$p = \frac{\tilde{p}}{Z_p}, \qquad q = \frac{\tilde{q}}{Z_q}$$

where $\tilde{p}$ and $\tilde{q}$ can be evaluated easily, whereas the normalizing constants Z_p and Z_q are unknown. Samples are drawn from (unnormalized) distribution $\tilde{q}$ and weighed correspondingly:

$$z_i \sim q$$
$$w_i = \frac{\tilde{p}(z_i)}{\tilde{q}(z_i)}$$

Then

$$\mathbb{E}f = \int p(z)f(z)dz = \frac{\int dz\, q(z)\frac{p(z)}{q(z)}f(z)}{\underbrace{\int dz\, q(z)\frac{p(z)}{q(z)}}_{1}} = \frac{\int dz\, q(z)\frac{p(z)Z_p}{q(z)Z_q}f(z)}{\int dz\, q(z)\frac{p(z)Z_p}{q(z)Z_q}} = \frac{\int dz\, q(z)\frac{\tilde{p}(z)}{\tilde{q}(z)}f(z)}{\int dz\, q(z)\frac{\tilde{p}(z)}{\tilde{q}(z)}}$$

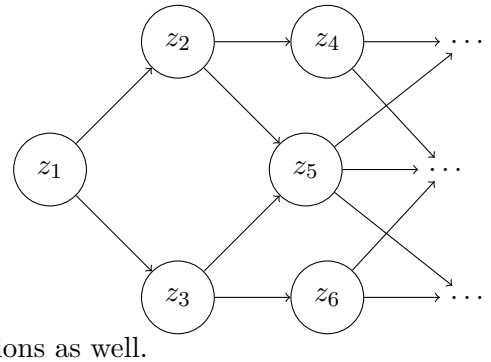
$$\approx \frac{\sum_i w_i f(z_i)}{\sum_j w_j}$$

Works bad in high dimensions as it is hard to find an z where both $\tilde{q}(z)$ and $\tilde{p}(z)$ are high in the high-dimensional space. If $\tilde{p}$ and $\tilde{q}$ overlap just a little, this will give a bad estimate.

Ancestral sampling (AKA Likelihood Weighted sampling)

Given a Bayesian network, we can write down the joint probability as $p(z_1)p(z_2|z_1)\ldots$. That means there is always a node in the DAG that is **not** dependent on any other variable. We sample this node, and use the sample to draw samples for subsequent nodes:

$$\tilde{z}_1 \sim p(z_1)$$
$$\tilde{z}_2 \sim p(z_2|\tilde{z}_1)$$
$$\tilde{z}_3 \sim p(z_3|\tilde{z}_1)$$
$$\tilde{z}_4 \sim p(z_5|\tilde{z}_2, \tilde{z}_3)$$
$$\vdots$$

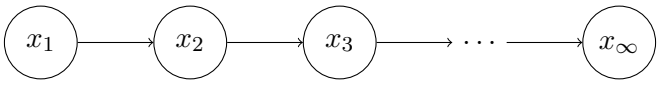


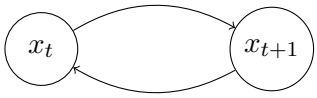
Works bad in high dimensions as well.

Markov Chain Monte-Carlo (MCMC) sampling

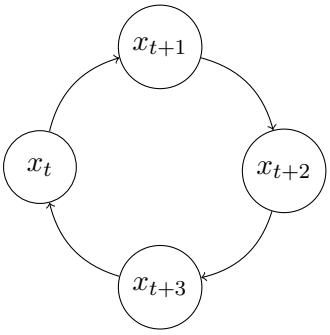
MCMC runs ancestral sampling on a chain. Samples can no longer be drawn independently!

$$x_t \overset{t\to\infty}{\sim} q_\infty \qquad \textit{Equilibrium} \text{ or } \textit{invariant} \text{ distribution.}$$





(a) Figure showing a case of detailed balance



(b) Figure showing a case of invariance where there is no detailed balance

We transition from q_1 to q_2 according to transition probability $T(x_2|x_1)$:

$$q(x_1, x_2) = T(x_2|x_1)q(x_1)$$
$$q(x_2) = \int T(x_2|x_1)q(x_1)dx_1$$

Generalization of this rule:

$$q(\underbrace{x_{t+1}, x_t)}_{\text{May be correlated!}} = T(x_{t+1}|x_t)q(x_t) \tag{1}$$

Invariance

We want to find T for a given p, such that $p = Tp$ holds:

$$\underbrace{p(x_{t+1}) = \int dx_t T(x_{t+1}|x_t)p(x_t)}_{\text{Eigenvalue equation for } \lambda = 1} \tag{2}$$

Proof of invariance of the current method: if $T(x_{t+1}|x_t)p(x_t) = T(x_t|x_{t+1})p(x_{t+1})$ for all x_t, x_{t+1}, then:

$$\int dx_t T(x_{t+1}|x_t)p(x_t) = \underbrace{\int dx_t T(x_t|x_{t+1})}_{1} p(x_{t+1}) = p(x_{t+1}).$$

Detailed balance (or reversibility)

We want to find T so that transition from one state to another has probability mass equal to the transition from that next state to the current:

$$T(x_{t+1}|x_t)p(x_t) = T(x_t|x_{t+1})p(x_{t+1}) \tag{3}$$

Ergodicity

This algorithm needs *ergodicity*: A positive probability for every state (so all will be reached).

Let's say we have two transition functions, T_1 and T_2, that satisfy detailed balance and are ergodic. We can then construct a new one by taken a weighted average, or by combining the operations:

$$T_3 = \alpha T_1 + (1 - \alpha) T_2$$
$$T_4 = T_1 \circ T_2$$

Metropolis-Hastings algorithm

Algorithm 1 Metropolis-Hastings algorithm

1: $x_{t+1} \sim q(x_{t+1}|x_t)$
2: $\alpha = \min\left(1, \frac{p(x_{t+1})q(x_t|x_{t+1})}{p(x_t)q(x_{t+1}|x_t)}\right)$
3: $u \sim U(0, 1)$
4: if $u \leq \alpha$, accept, otherwise, keep copy of the old sample instead.

Proof of detailed balance: if the new sample is accepted, then:

$$\begin{aligned} p(x_t)q(x_{t+1}|x_t)\min\left(1, \frac{p(x_{t+1})q(x_t|x_{t+1})}{p(x_t)q(x_{t+1}|x_t)}\right) &= \min\left(p(x_t)q(x_{t+1}|x_t), p(x_{t+1})q(x_t|x_{t+1})\right) \\ &= \min\left(p(x_{t+1})q(x_t|x_{t+1}), p(x_t)q(x_{t+1}|x_t)\right) \\ &= p(x_{t+1})q(x_t|x_{t+1})\min\left(1, \frac{p(x_t)q(x_{t+1}|x_t)}{p(x_{t+1})q(x_t|x_{t+1})}\right) \end{aligned}$$

So, if the algorithm satisfies *ergodicity* and *detailed balance*, it will eventually sample from the desired distribution $p(z)$. The first ("burn-in") samples should be discarded as they are not yet sampling from the equilibrium distribution. It is hard to say in general how many burn-in samples there are.

Will work better in high dimensions than previous methods, but is quite slow (as we perform a random walk).

Quasi-MC sampling

Fun fact: If we sample in a smart way (e.g. not sample the same point twice, ensure that areas there hasn't been sampled from have higher probability of being chosen) instead of random, these methods may converge faster.

Machine Learning 2

Lecturer: Max Welling
Scribe: Mircea Traichioiu
Updated: May 2, 2016

Lecture #10
November 27, 2013

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

MCMC Review

Being given a distribution $\tilde{p}(x)$, the goal is to design an algorithm with transition kernel $T(\mathbf{x_{t+1}}|\mathbf{x_t})$ that maps sampled point $\mathbf{x_t}$ to $\mathbf{x_{t+1}}$.

This transition kernel must satisfy either the *invariant distribution* or *detailed balance* properties (described below), in addition to *ergodizity* (each state must have a non-zero chance to be explored).

Furthermore, if T_1 and T_2 are valid transition kernels, then also a linear combination $T_3 = \pi T_1 + (1-\pi)T_2$ and a composition $T_4 = T_2 \circ T_1$ are valid transition kernels.

Invariant Distribution

$$p(\mathbf{x_{t+1}}) = \int T(\mathbf{x_{t+1}}|\mathbf{x_t})p(\mathbf{x_t})\mathrm{d}\mathbf{x}_t$$

The above relation can be seen as a marginalization over $\mathbf{x_t}$, with $T(\mathbf{x_{t+1}}|\mathbf{x_t})p(\mathbf{x_t}) = p(\mathbf{x_{t+1}}, \mathbf{x_t})$

Detailed Balance (reversibility)

$$p(\mathbf{x_t})T(\mathbf{x_{t+1}}|\mathbf{x_t}) = p(\mathbf{x_{t+1}})T(\mathbf{x_t}|\mathbf{x_t} + \mathbf{1})$$

Detailed balance implies the invariant distribution property, but the other way around is not necesarly true. However, in general it is easier to design a transition kernel that satisfies detailed balance.

Metropolis Hastings Algorithm

1. Sample $\mathbf{x_{t+1}} \sim Q(\mathbf{x_{t+1}}|\mathbf{x_t})$ (proposal distribution)

2. Compute "acceptance probability"

$$\alpha(\mathbf{x_t} \to \mathbf{x_{t+1}}) = \min\left(1, \frac{p(\mathbf{x_{t+1}})Q(\mathbf{x_t}|\mathbf{x_{t+1}})}{p(\mathbf{x_t})Q(\mathbf{x_{t+1}}|\mathbf{x_t})}\right)$$

   Thus, the transition kernel is $T = Q \circ \alpha$

3. $u \sim U(0,1)$ (random uniform)

   - if $u \leq \alpha$: accept $\Rightarrow S_{t+1} = \{S_t, \mathbf{x_{t+1}}\}$
   - if $u > \alpha$: reject $\Rightarrow S_{t+1} = \{S_t, \mathbf{x_t}\}$

   where S_t denotes the sample set at time t.

Gibbs sampling

Let a sample at a given moment t be a D-dimensional vector $(x_1^t, x_2^t, \dots x_D^t)$. Gibbs sampling involves sampling on a single dimension at a time:

$$
\begin{aligned}
x_1^{t+1} &\sim p(x_1 | x_2^t, x_3^t, \dots x_D^t) \\
x_2^{t+1} &\sim p(x_2 | x_1^{t+1}, x_3^t, \dots x_D^t) \\
&\dots \\
x_D^{t+1} &\sim p(x_D | x_1^{t+1}, x_2^{t+1}, \dots x_{D-1}^{t+1})
\end{aligned}
$$

The order in which the dimensions are chosen can either be random or fixed. If the order is fixed, detailed balance is not guaranteed, but the transition kernel is still valid (i.e., leads to invariant p).

The underlying principle for the Gibbs sampling is the fact that sampling a variable from an unidimensional distribution is easier than drawing from a multi-dimensional one. In particular for graphical models, this involves only fixing the values for the nodes in the Markov blanket of the desired variable.

Hamiltonian Monte Carlo (HMC) (Skipped in 2016)

This algorithm is the preferred sampling method for working with continuous variables. Consider the following model that generates joint samples $\{(\mathbf{x_t}, r_t)\}$:

$$
\begin{aligned}
p(\mathbf{x}) &= \frac{1}{Z_{\mathbf{x}}} e^{-E(\mathbf{x})} && (1) \\
p(r) &= \mathcal{N}(r|0,1) = \frac{1}{r} e^{-\frac{1}{2}\|\overrightarrow{r}\|^2} && (2) \\
p(\mathbf{x}, r) &= p(\mathbf{x})p(r) \propto e^{-E(\mathbf{x})-\frac{1}{2}\|\overrightarrow{r}\|^2} && (3) \\
p(\mathbf{x}) &= \int p(\mathbf{x}, r) \mathrm{d}r && (4)
\end{aligned}
$$

In equation 3 an analogy with a physical process is made, with $E(\mathbf{x})$ denoting *potential energy* and $\frac{1}{2}\|\overrightarrow{r}\|^2$ denoting *kinetic energy*. Under Newtonian physics, mechanic energy is conserved, i.e. $H = E + K$. Thus, the following conditions must hold:

$$
\begin{aligned}
\frac{\partial x_i}{\partial t} &= \frac{\partial H}{\partial r_i} = r_i \\
\frac{\partial r_i}{\partial t} &= -\frac{\partial H}{\partial x_i} = -\frac{\partial E}{\partial x_i}
\end{aligned}
$$

We have the following properties:

1. $H(\mathbf{x}(t), r(t)) = H(\mathbf{x}(0), r(0))$ (Total energy does not change)

$$
\begin{aligned}
\frac{\partial H}{\partial t} &= \frac{\partial H}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial H}{\partial r}\frac{\partial r}{\partial t} \\
&= \frac{\partial H}{\partial x}\frac{\partial H}{\partial r} + \frac{\partial H}{\partial r}\left(-\frac{\partial H}{\partial x}\right) = 0
\end{aligned}
$$

2. Volume does not change.

$$p(\mathbf{x}_0, r_0)\mathrm{d}V_0 = p(\mathbf{x}_t, r_t)\mathrm{d}V_t$$
$$p(\mathbf{x}_t, r_t) = p(\mathbf{x}_0, r_0)\underbrace{\frac{\mathrm{d}V_t}{\mathrm{d}V_0}}_{=1}$$

3. Procedure is reversible (detailed balance holds)

Algorithm HMC

1. $\mathbf{x_0}$
2. draw $r_0 \sim \mathcal{N}(0, 1)$
3. "Leapfrog stepping"
   - $r_i(t + \frac{\epsilon}{2}) = r_i(t) - \frac{\epsilon}{2}\frac{\partial E}{\partial x_i}\Big|_{x(t)}$
   - $x_i(t + \epsilon) = x_i(t) + \epsilon r_i(t + \frac{\epsilon}{2})$
   - $r_i(t + \epsilon) = r_i(t + \frac{\epsilon}{2}) - \frac{\epsilon}{2}\frac{\partial E}{\partial x_i}\Big|_{x_i(t+\epsilon)}$
   - Iterate for T rounds
4. Accept $(\mathbf{x}(t), r(t))$ with probability $\alpha = \min(1, e^{H_0 - H_t})$
5. Repeat

Regarding step 4, it must be noted that the acceptance probability would always be 1 if the Hamiltonian dynamics would be modelled perfectly (i.e. $H_t = H_0$). However, in practice this is not the case due to numerical errors.

Machine Learning 2

Lecturer: Joris Mooij
Scribes: Alberto Ferreira & Jorge Sáez Gómez
Updated: March 25, 2015; May 10, 2016

Lecture #11
2 December, 2013

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

(13) Sequential Data

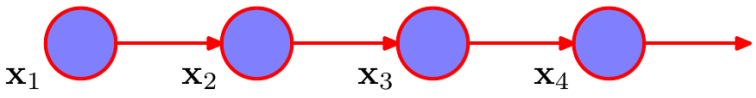
So far, we have studied models for which:

- The likelihood of a single data point is expressed as $p(\boldsymbol{x}|\boldsymbol{\theta})$
- Multiple data points are assumed to be i.i.d.: $\prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\theta})$

We might want to drop the i.i.d. assumption, since this is too restrictive for time-dependent data points: weather, stock prices...

(13.1) Markov models

If we have a graphical model with nodes $\boldsymbol{x}_1$ to $\boldsymbol{x}_N$ (all observed), we might connect them in a chain as follows (Markov model):



And thus, it follows that:

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

Note that the arrows in the previous graph do not necessarily represent time (e.g.: words in sentences, space...).

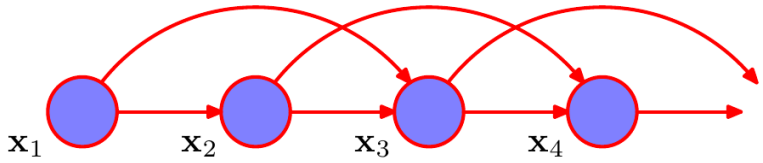
Markov property/assumption:

$$p(\boldsymbol{x}_n|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}) = p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) \tag{1}$$

Homogeneous M.M. $\Rightarrow$ $p(\boldsymbol{x_{n+1}}|\boldsymbol{x_n})$" = "$p(\boldsymbol{x_n}|\boldsymbol{x_{n-1}})$, where by " = " an equality between, for instance, the probability tables is meant, not between particular probability values. This assumption allows to reduce the number of parameters needed to specify the Markov chain.

So far, we hava only considered first order Markov chains, but we may write higher order Markov chains by adding dependences on more previous nodes. For example, a second order Markov chain would be:

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



Note that in such cases the Markov property needs to be adapted accordingly.

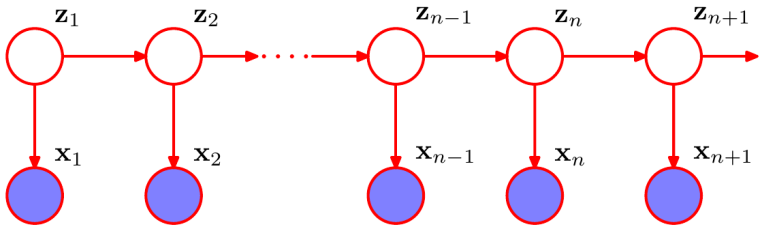
The number of parameters of a Markov chain increases exponentially with its order (where K is the number of states of the variables):

- 1) $K - 1 + K(K-1)$
- 2) $K^2 - 1 + K^2(K-1)$
- ...
- M) $K^M - 1 + K^M(K-1)$

By grouping variables (i.e.: $\boldsymbol{y}_n = (\boldsymbol{x}_n, \boldsymbol{x}_{n-1})$) we can define higher order Markov chains as a first order Markov chain.

Also known as AR model (Auto-Regressive model) for linear-Gaussian case (i.e. $p(\boldsymbol{x_n}|\boldsymbol{x_{n-1}}) = \mathcal{N}(\boldsymbol{x}_n|B\,\boldsymbol{x}_{n-1}, \Sigma)$).

(13.2) Hidden Markov Models (HMMs)



Useful in speech recognition, natural language processing, online character recognition... Comes in two flavours:

- Discrete latent variables: *HMM*
- Continuous latent variables, linear-Gaussian interactions: *LDS* (Linear Dynamical System)

For the homogeneous case:

$$p(\boldsymbol{x}, \ldots, \boldsymbol{x}_N|\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{\phi}) = \sum_{\boldsymbol{z}_1} \cdots \sum_{\boldsymbol{z}_N} p(\boldsymbol{z}_1|\boldsymbol{\pi}) \left(\prod_{n=2}^{N} \underbrace{p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}, \boldsymbol{A})}_{\text{Transition probabilities}} \right) \prod_{n=1}^{N} \underbrace{p(\boldsymbol{x}_n|\boldsymbol{z}_n, \boldsymbol{\phi})}_{\text{Emission probabilities}} \tag{2}$$

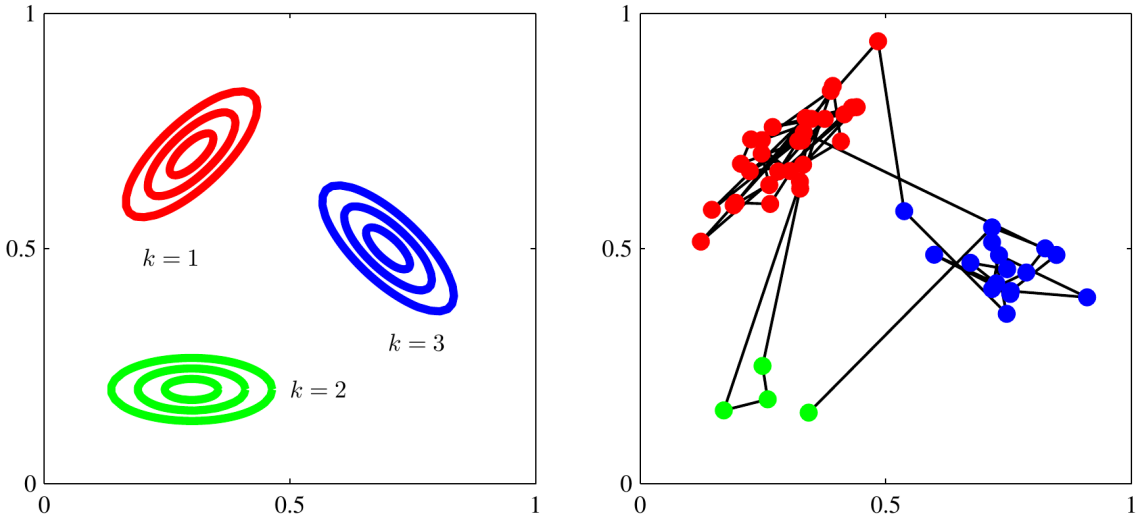
We will use 1-of-K coding for $\boldsymbol{z}_n$ (e.g.: $\boldsymbol{z}_n = (0, 1, 0, 0)$ corresponds with the 2^{nd} state)

$$p(\boldsymbol{z}_1|\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{z_{1k}}$$

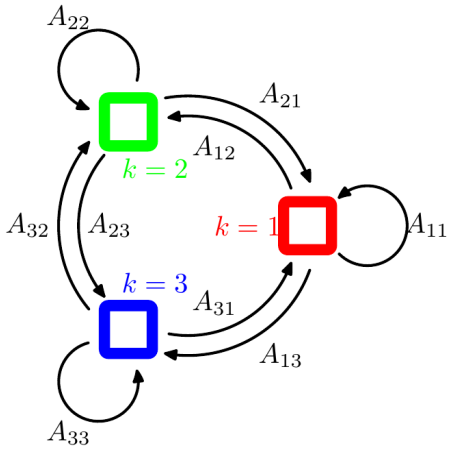
$$p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}, \boldsymbol{A}) = \prod_{k=1}^{K} \prod_{j=1}^{K} A_{jk}^{z_{n-1,j} z_{nk}}$$

$$p(\boldsymbol{x}_n|\boldsymbol{z}_n, \boldsymbol{\phi}) = \prod_{k=1}^{K} p(\boldsymbol{x}_n|\phi_k)^{z_{nk}}$$

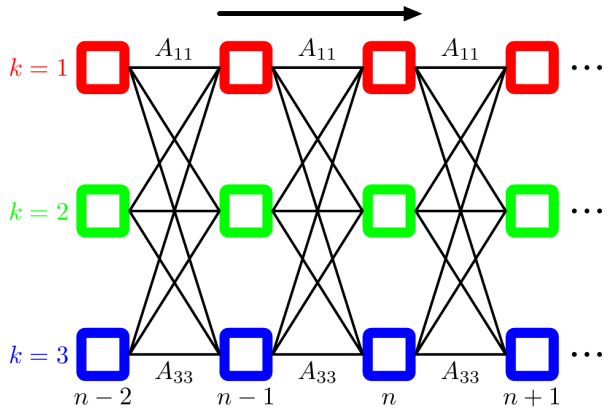
Where $\boldsymbol{A}$, $\boldsymbol{\phi}$ and $\boldsymbol{\pi}$ are the parameters of the model. $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \boldsymbol{\phi})$ is model-dependent (e.g.: mixture of Gaussians, multinomial if discrete, etc). The next figure shows an example with a mixture of Gaussians for the emission probabilities:



The model parameter $\boldsymbol{A}$ can be interpreted as the state transition probabilities: A_{jk} is the probability to go from state j to state k. We can picture the HMM as a state transition diagram:



The HMM is most useful when $\boldsymbol{A}$ is sparse. If homogeneity is assumed, then all parameters can be estimated from a single, long observation of data. The following *lattice trellis diagram* shows the unfolding state transitions of the latent variables over time:



(13.2.1) Maximum Likelihood for HMMs

We have:

- Data $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$
- Latent state $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N)$
- Parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{\phi})$

And thus:

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$$

This represents a sum of K^N terms, which quickly becomes intractable, so we can use for instance the EM algorithm.

E-step:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \tag{3}$$

M-step:

$$\boldsymbol{\theta}^{\text{new}} = \text{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) \tag{4}$$

We will first introduce a new notation:

$$\begin{aligned} \gamma(\boldsymbol{z}_n) &\leftarrow p(\boldsymbol{z}_n|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}) \\ \xi(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n) &\leftarrow p(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}) \end{aligned} \tag{5}$$

We can now write Q more explicitly:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{k=1}^{K} \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{nk}) \ln A_{jk} + \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \ln p(\boldsymbol{x}_n|\phi_k) \tag{6}$$

M-step

We can finally solve for the model parameters for the M-step, taking care of adding Lagrange multipliers where needed in order to enforce the probabilities contraints:

$$\begin{aligned} \pi_k^{\text{new}} &\leftarrow \text{argmax}_{\pi_k} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})} \\ A_{jk}^{\text{new}} &\leftarrow \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nl})} \end{aligned} \tag{7}$$

For a Gaussian measurement model, i.e.,

$$p(\boldsymbol{x}_n|\boldsymbol{\phi}, \boldsymbol{z}_n) = \sum_{k=1}^{K} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \Sigma_k) z_{nk}$$

with $\boldsymbol{\phi}_k = (\boldsymbol{\mu}_k, \Sigma_k)$, we get update equations (13.20) and (13.21) for $\boldsymbol{\phi}$.

For a multinomial measurement model, i.e.,

$$p(\boldsymbol{x}_n|\boldsymbol{z}_n, \boldsymbol{\phi}) = \prod_{i=1}^{D} \prod_{k=1}^{K} \mu_{ik}^{x_{ni} z_{nk}}$$

with $\mu \in \mathcal{R}^{D \times K}$ we get update equations (13.23) for $\boldsymbol{\phi}$.

E-step

For the E-step, we need to calculate marginal probabilities in the graphical model. We can use the sum-product algorithm for this purpose. Bishop starts with treating the "forward-backward (Baum-Welch) algorithm" in 13.2.2 from scratch. In 13.2.3 he then shows that this is actually a special case of the sum-product algorithm. Instead, we will directly write down the sum-product algorithm for the case at hand.

We start by representing the graphical model as a factor graph. Since the data $\boldsymbol{X}$ are observed and the parameters are fixed at $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{old}}$, we are only interested in the dependence on $\boldsymbol{Z}$. In other words, we will represent the (unnormalized) probability distribution $p(\boldsymbol{Z}, \boldsymbol{X}|\boldsymbol{\theta}^{\text{old}})$ as a factor graph. The normalization output by the sum-product algorithm will then be $\sum_{\boldsymbol{Z}} p(\boldsymbol{Z}, \boldsymbol{X}|\boldsymbol{\theta}^{\text{old}})$. The (normalized) beliefs output by the sum-product algorithm are marginal probabilities of the conditional distribution $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}})$.

If we define the factors as follows:

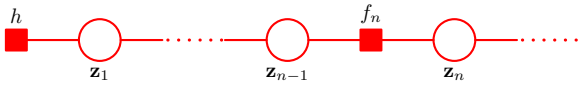
$$h(\boldsymbol{z}_1) = p(\boldsymbol{z}_1|\boldsymbol{\pi}^{\text{old}}) p(\boldsymbol{x}_1|\boldsymbol{z}_1, \boldsymbol{\phi}^{\text{old}})$$

$$f_n(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}, \boldsymbol{A}^{\text{old}}) p(\boldsymbol{x}_n|\boldsymbol{z}_n, \boldsymbol{\phi}^{\text{old}})$$

then the factor graph of

$$p(\boldsymbol{Z}, \boldsymbol{X}|\boldsymbol{\theta}^{\text{old}}) = p(\boldsymbol{z}_1|\boldsymbol{\pi}^{\text{old}}) \left(\prod_{n=2}^{N} p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}, \boldsymbol{A}^{\text{old}}) \right) \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{z}_n, \boldsymbol{\phi}^{\text{old}})$$

looks like:



The message update equations of the sum-product algorithm flowing from left to right in the factor graph become:

$$\mu_{\boldsymbol{z}_{n-1}\to f_n}(\boldsymbol{z}_{n-1}) = \mu_{f_{n-1}\to \boldsymbol{z}_{n-1}}(\boldsymbol{z}_{n-1})$$
$$\alpha_n(\boldsymbol{z}_n) := \mu_{f_n\to \boldsymbol{z}_n}(\boldsymbol{z}_n) = \sum_{\boldsymbol{z}_{n-1}} f_n(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n)\mu_{\boldsymbol{z}_{n-1}\to f_n}(\boldsymbol{z}_{n-1})$$

Substituting the first equation into the second, we get a more compact representation (also known as "alpha recursions" in the context of the Baum-Welch algorithm):

$$\alpha_n(\boldsymbol{z}_n) = \sum_{\boldsymbol{z}_{n-1}} f_n(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n)\alpha_{n-1}(\boldsymbol{z}_{n-1})$$

Similarly, we get the following message update equations for the messages flowing from right to left:

$$\mu_{\boldsymbol{z}_n\to f_n}(\boldsymbol{z}_n) = \mu_{f_{n+1}\to \boldsymbol{z}_n}(\boldsymbol{z}_n)$$
$$\beta_n(\boldsymbol{z}_n) := \mu_{f_{n+1}\to \boldsymbol{z}_n}(\boldsymbol{z}_n) = \sum_{\boldsymbol{z}_{n+1}} f_{n+1}(\boldsymbol{z}_n, \boldsymbol{z}_{n+1})\mu_{\boldsymbol{z}_{n+1}\to f_{n+1}}(\boldsymbol{z}_{n+1})$$

and by substitution we obtain the "beta recursions":

$$\beta_n(\boldsymbol{z}_n) = \sum_{\boldsymbol{z}_{n+1}} f_{n+1}(\boldsymbol{z}_n, \boldsymbol{z}_{n+1})\beta_{n+1}(\boldsymbol{z}_{n+1})$$

After one forward and one backward pass, the variable beliefs are given by:

$$p(\boldsymbol{z}_n, \boldsymbol{X}|\boldsymbol{\theta}^{\text{old}}) = \mu_{f_n\to \boldsymbol{z}_n}(\boldsymbol{z}_n)\mu_{f_{n+1}\to \boldsymbol{z}_n}(\boldsymbol{z}_n) = \alpha_n(\boldsymbol{z}_n)\beta_n(\boldsymbol{z}_n)$$

and the factor beliefs by:

$$p(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n, \boldsymbol{X}|\boldsymbol{\theta}^{\text{old}}) = \mu_{f_{n-1}\to \boldsymbol{z}_{n-1}}(\boldsymbol{z}_{n-1})\mu_{f_{n+1}\to \boldsymbol{z}_n}(\boldsymbol{z}_n)f_n(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n)$$

and the normalization constant by:

$$p(\boldsymbol{X}|\boldsymbol{\theta}^{\text{old}}) = \sum_{\boldsymbol{z}_n} \alpha_n(\boldsymbol{z}_n)\beta_n(\boldsymbol{z}_n)$$

(for any n).

Therefore, the quantities we need in the E-step in in the EM algorithm are given by

$$\gamma(\boldsymbol{z}_n) = \frac{p(\boldsymbol{z}_n, \boldsymbol{X}|\boldsymbol{\theta}^{\text{old}})}{p(\boldsymbol{X}|\boldsymbol{\theta}^{\text{old}})} = \frac{\alpha_n(\boldsymbol{z}_n)\beta_n(\boldsymbol{z}_n)}{p(\boldsymbol{X}|\boldsymbol{\theta}^{\text{old}})}$$

and

$$\xi(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n) = \frac{p(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n, \boldsymbol{X}|\boldsymbol{\theta}^{\text{old}})}{p(\boldsymbol{X}|\boldsymbol{\theta}^{\text{old}})} = \frac{\alpha_{n-1}(\boldsymbol{z}_{n-1})\beta_n(\boldsymbol{z}_n)p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}, \boldsymbol{A}^{\text{old}})p(\boldsymbol{x}_n|\boldsymbol{z}_n, \boldsymbol{\phi}^{\text{old}})}{p(\boldsymbol{X}|\boldsymbol{\theta}^{\text{old}})}$$

Now that the E-step has been solved, the M-step can be performed.

EM for HMM

Overall, the EM algorithm for HMM becomes:

1. Choose initial parameters $\boldsymbol{\theta}^{\text{old}}$ where $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{\phi})$.

2. Iterate until convergence:

   E-step "Calculate $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$":

   (a) Run forward α recursion to calculate $\alpha(\boldsymbol{z}_1)$, ..., $\alpha(\boldsymbol{z}_N)$;
   (b) Run backward β recursion to calculate $\beta(\boldsymbol{z}_N)$, ..., $\beta(\boldsymbol{z}_N)$;
   (c) Calculate sufficient statistics $\{\gamma(\boldsymbol{z}_n)\}, \{\xi(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n)\}, p(\boldsymbol{X}|\boldsymbol{\theta}^{\text{old}})$;

   M-step "Calculate $\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$":
   Use M-step equations (13.18) to update $\boldsymbol{\pi}$, (13.19) to update $\boldsymbol{A}$, and (13.20-21) or (13.23) or other model-dependent equations to update $\boldsymbol{\phi}$.

Predictive distribution

Add $\boldsymbol{x}_{N+1}$ and $\boldsymbol{z}_{N+1}$ at the end of the chain and use the calculated parameters to predict the new values (assuming the chain is homogeneous).

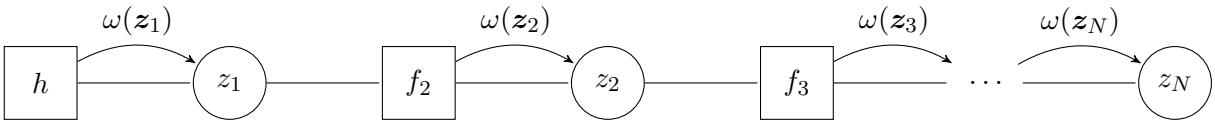
Machine Learning 2

Lecturer: Joris Mooij — Lecture #12
Scribe: Nikolaas Steenbergen &Efstathios Charitos & Jerome Cremers — December 4, 2013
Updated: March 25, 2015; May 10, 2016

If you spot any error, please email to Joris Mooij (j.m.mooij@uva.nl)

13.2.5 Viterbi Algorithm

The Viterbi algorithm answers the question: calculate the most probable sequence of latent states, given the observations. This is a special case of the max-sum algorithm, it follows the principle of dynamic programming. The factors include both emission and transition probabilities, except h. We define: $\omega(\boldsymbol{z}_n) = \nu_{f_n \to z_n}(\boldsymbol{z}_n)$, where $\nu_{f_n \to z_n}$ is the message in the max-sum algorithm, for $2 \leq n \leq N$.



Bishop is not clear on this, thus following pseudo algorithm:

//first message;
$\omega(\boldsymbol{z}_1) = \ln p(\boldsymbol{z}_1) + \ln p(\boldsymbol{x}_1|\boldsymbol{z}_1)$;
//message passing from left to right;
for $n = 1 : N - 1$ **do**
  $\omega(\boldsymbol{z}_{n+1}) = \ln p(\boldsymbol{x}_{n+1}|\boldsymbol{z}_{n+1}) + \max_{\boldsymbol{z}_n} [\ln p(\boldsymbol{z}_{n+1}|\boldsymbol{z}_n) + \omega(\boldsymbol{z}_n)]$;
  //keep track of value that maximizes;
  $\psi_n(\boldsymbol{z}_{n+1}) = \arg\max_{\boldsymbol{z}_n} [\ln p(\boldsymbol{z}_{n+1}|\boldsymbol{z}_n) + \omega(\boldsymbol{z}_n)]$;
end

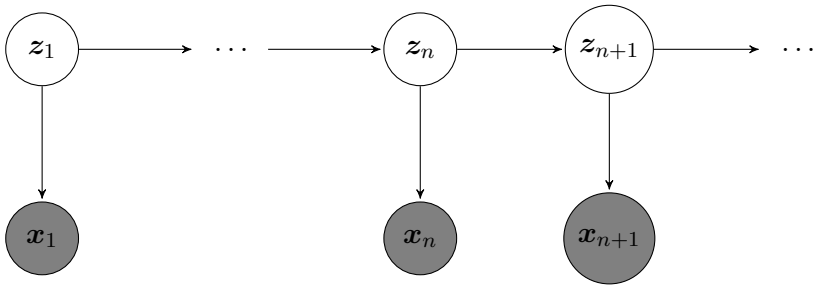
Algorithm 1: forward pass

//start from the rightmost side and reason backwards;
$\boldsymbol{z}_N^{max} = \arg\max_{\boldsymbol{z}_N} \omega(\boldsymbol{z}_N)$
for $n = N - 1 : 1$ **do**
  $\boldsymbol{z}_n^{max} = \psi_n(\boldsymbol{z}_{n+1}^{max})$
end

Algorithm 2: backward pass

The sequence $(\boldsymbol{z}_1^{max}, \ldots, \boldsymbol{z}_N^{max})$ then contains the most probable sequence of latent states, given the observations.

13.3 Linear Dynamic Systems



where the $\boldsymbol{x}_n$ are observed and the $\boldsymbol{z}_n$ are latent.

Note that: $\boldsymbol{z}_n$ is now continuous. Linear Gaussian Model: conditional distributions are Gaussian with means that depend linearly on their parents.
$\rightarrow$ All conditional / marginal distributions are Gaussian.
$\rightarrow$ mode = mean (no need for Viterbi)

Forward message passing equation: Kalman filter equations.
Backward message passing equation: Kalman smoother equations.

Notation transitions:

$$p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}) = \mathcal{N}(\boldsymbol{z}_n|A\boldsymbol{z}_{n-1}, \Gamma)$$

where Γ is transition noise
Emissions:

$$p(\boldsymbol{x}_n|\boldsymbol{z}_n) = \mathcal{N}(\boldsymbol{x}_n|C\boldsymbol{z}_n, \Sigma)$$

where Σ is observation noise.
Initial state:

$$p(\boldsymbol{z}_1) = \mathcal{N}(\boldsymbol{z}_1|\boldsymbol{\mu}_0, V_0)$$

six parameters: A,Γ,C,Σ,μ_0,V_0, to be learned form the data using EM.

13.3.1 Inference in Linear dynamical systems (for E-Step)

Forward equations

$$\hat{\alpha}(\boldsymbol{z}_n) = \mathcal{N}(\boldsymbol{z}_n|\boldsymbol{\mu}_n, V_n)$$

Recursion equations (similar to the discrete case):

$$c_n\hat{\alpha}(\boldsymbol{z}_n) = p(\boldsymbol{x}_n|\boldsymbol{z}_n)\int \hat{\alpha}(\boldsymbol{z}_{n-1})p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1})d_{\boldsymbol{z}_{n-1}}$$

$$c_n\mathcal{N}(\boldsymbol{z}_n|\boldsymbol{\mu}_n, V_n) = \mathcal{N}(\boldsymbol{x}_n|C\boldsymbol{z}_n, \Sigma)\int \mathcal{N}(\boldsymbol{z}_{n-1}|\mu_{n-1}, V_{n-1})\mathcal{N}(\boldsymbol{z}_n|A\boldsymbol{z}_{n-1}, \Gamma)d\boldsymbol{z}_{n-1}$$

where c_n is a scaling factor to ensure proper normalization. Using the equations from Bishop (2.115) and (2.116)[1] we can first calculate the integral term as:

$$\int \mathcal{N}(\boldsymbol{z}_{n-1}|\mu_{n-1}, V_{n-1})\mathcal{N}(\boldsymbol{z}_n|A\boldsymbol{z}_{n-1}, \Gamma)d\boldsymbol{z}_{n-1} = \mathcal{N}(\boldsymbol{z}_n|A\boldsymbol{\mu}_{n-1}, \Gamma + AV_{n-1}A^T)$$

we define:$P_{n-1} = \Gamma + AV_{n-1}A^T$ and we are left with:

$$c_n\mathcal{N}(\boldsymbol{z}_n|\boldsymbol{\mu}_n, V_n) = \mathcal{N}(\boldsymbol{x}_n|C\boldsymbol{z}_n, \Sigma)\mathcal{N}(\boldsymbol{z}_n|A\boldsymbol{\mu}_{n-1}, P_{n-1})$$

where we can use the footnote equations again seeing that: $c_n \leftrightarrow p(y)$, $\mathcal{N}(\boldsymbol{z}_n|\boldsymbol{\mu}_n, V_n) \leftrightarrow p(x|y)$, $\mathcal{N}(\boldsymbol{x}_n|C\boldsymbol{z}_n, \Sigma) \leftrightarrow p(y|x)$, $\mathcal{N}(\boldsymbol{z}_n|A\boldsymbol{\mu}_{n-1}, P_{n-1}) \leftrightarrow p(x)$. To make things clear we make the following matrix relating the quantities in Bishop with our variables:

x, y	$\boldsymbol{\mu}$	Λ^{-1}	A	$\boldsymbol{b}$	L^{-1}	Σ
$\boldsymbol{z}_n, \boldsymbol{x}_n$	$A\boldsymbol{\mu}_{n-1}$	P_{n-1}	C	$\mathbf{0}$	Σ	$\left(P_{n-1}^{-1} + C^T\Sigma^{-1}C\right)^{-1}$

We get the following results:

$$c_n = \mathcal{N}(\boldsymbol{x}_n|CA\boldsymbol{\mu}_{n-1}, \Sigma + CP_{n-1}C^T)$$

$$\mu_n = \left(P_{n-1}^{-1} + C^T\Sigma^{-1}C\right)^{-1}\left(C^T\Sigma^{-1}\boldsymbol{x}_n + P_{n-1}^{-1}A\boldsymbol{\mu}_{n-1}\right)$$

$$V_n = \left(P_{n-1}^{-1} + C^T\Sigma^{-1}C\right)^{-1}$$

Alternatively, one could use the formula for a product of two Gaussians (see lecture notes of lecture # 1) to derive these results.

Finally we can use the following identities to simplify the results and reduce the computation time by doing less inverse matrix calculations.

We apply Bishop C.7 (Woodbury identity)

$$\left(A + BD^{-1}C\right)^{-1} = A^{-1} - A^{-1}B\left(D + CA^{-1}B\right)^{-1}CA^{-1}$$

to the equation for V_n. By defining the *Kalman Gain Matrix* as:

$$K_n = P_{n-1}C^T\left(CP_{n-1}C^T + \Sigma\right)^{-1}$$

we get:

$$V_n = \left(P_{n-1}^{-1} + C^T\Sigma^{-1}C\right)^{-1} = P_{n-1} - P_{n-1}C^T(\Sigma + CP_{n-1}C^T)^{-1}CP_{n-1} = (I - K_nC)P_{n-1}$$

[1]

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \Lambda^{-1})$$
$$p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|A\boldsymbol{x} + \boldsymbol{b}, L^{-1})$$

(see Bishop 2.115)

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|A\boldsymbol{\mu} + \boldsymbol{b}, L^{-1} + A\Lambda^{-1}A^T)$$

(also see Bishop 2.116)

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{x}|\Sigma(A^TL(\boldsymbol{y} - \boldsymbol{b}) + \Lambda\boldsymbol{\mu}), \Sigma)$$

where $\Sigma = (\Lambda + A^TLA)^{-1}$.

Applying Bishop C.5

$$\left(P^{-1} + B^T R^{-1} B\right)^{-1} B^T R^{-1} = P B^T \left(BPB^T + R\right)^{-1}$$

to the first part of the expression for $\boldsymbol{\mu}_n$ leads to:

$$\boldsymbol{\mu}_n = K_n \boldsymbol{x}_n + (I - K_n C) A \boldsymbol{\mu}_{n-1} = A\boldsymbol{\mu}_{n-1} + K_n(\boldsymbol{x}_n - CA\boldsymbol{\mu}_{n-1})$$

The term $\boldsymbol{x}_n - CA\boldsymbol{\mu}_{n-1}$ represents the error between the predicted observation and the actual observation. Similarly we can derive $\hat{\alpha}(\boldsymbol{z}_1)$ and obtain Bishop equations 13.94 to 13.97 (exercise!).

Backward equations

In the LDS literature backward recursion is formulated in terms of $\gamma(\boldsymbol{z}_n) = \hat{\alpha}(\boldsymbol{z}_n)\hat{\beta}(\boldsymbol{z}_n)$

$$\gamma(\boldsymbol{z}_n) = \mathcal{N}(\boldsymbol{z}_n | \hat{\mu}_n, \hat{V}_n)$$

exercise: Derive 13.99-13.104 in Bishop

For the EM algorithm we also need:

$$\xi(\boldsymbol{z}_{n-1}, \boldsymbol{z}_n) = \mathcal{N}\left(\begin{pmatrix} \boldsymbol{z}_{n-1} \\ \boldsymbol{z}_n \end{pmatrix} \middle| \begin{pmatrix} \hat{\mu}_{n-1} \\ \hat{\mu}_n \end{pmatrix}, \left(\begin{array}{c|c} \hat{V}_{n-1} & J_{n-1}\hat{V}_n \\ \hline \hat{V}_n J_{n-1}^T & \hat{V}_n \end{array}\right)\right)$$

13.3.2 Learning in LDS using EM

Complete data log-likelihood:

$$\ln p(X, Z|\boldsymbol{\theta}) = \ln p(\boldsymbol{z}_1|\boldsymbol{\mu}_0, \boldsymbol{V}_0) + \sum_{n=2}^{N} \ln p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}, A, \Gamma) + \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n|\boldsymbol{z}_n, C, \Sigma)$$

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{\theta}^{old}}[\ln p(X, Z|\boldsymbol{\theta})]$$

Use results from inference (note: Bishop (13.105-13.107) are sloppy!)

$$\mathbb{E}(\boldsymbol{z}_n|\boldsymbol{\theta}^{old}) = \hat{\mu}_n$$

$$\mathbb{E}(\boldsymbol{z}_n \boldsymbol{z}_{n-1}^T|\boldsymbol{\theta}^{old}) = J_{n-1}\hat{V}_n + \hat{\mu}_n \hat{\mu}_{n-1}^T$$

$$\mathbb{E}(\boldsymbol{z}_n \boldsymbol{z}_n^T|\boldsymbol{\theta}^{old}) = \hat{V}_n + \hat{\mu}_n \hat{\mu}_n^T$$

M-step:

$$\boldsymbol{\theta}^{new} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$$

e.g.

$$\boldsymbol{\mu}_0, V_0 : Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = -\frac{1}{2}\ln|V_0| - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}_1|\boldsymbol{\theta}^{old}}[(\boldsymbol{z}_1 - \boldsymbol{\mu}_0)^T V_0^{-1}(\boldsymbol{z}_1 - \boldsymbol{\mu}_0)] + const$$

Now use calculation similar to that for finding maximum likelihood estimation for Gaussians (section 2.3.4):

$$\boldsymbol{\mu}_0^{new} = \mathbb{E}(\boldsymbol{z}_1|\boldsymbol{\theta}^{old})$$

$$V_0^{new} = \mathbb{E}(\boldsymbol{z}_1 \boldsymbol{z}_1^T|\boldsymbol{\theta}^{old}) - \mathbb{E}(\boldsymbol{z}_1|\boldsymbol{\theta}^{old})\mathbb{E}(\boldsymbol{z}_1|\boldsymbol{\theta}^{old})^T$$

Similarly for A, Γ, C, Σ. (Exercise 13.33, 13.34).

ML2 lecture *Causality*

Joris Mooij
j.m.mooij@uva.nl

Informatics Institute

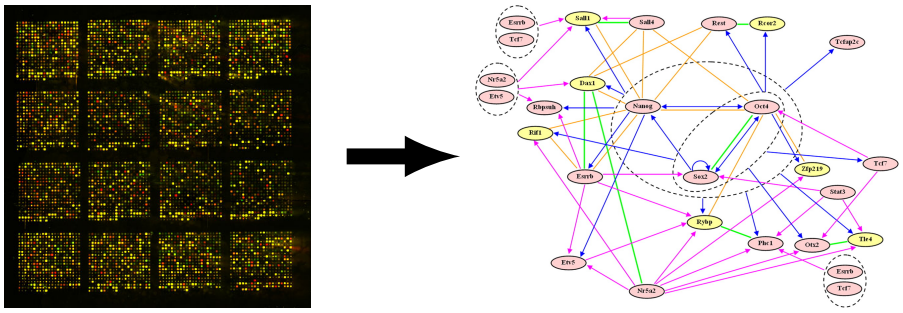

University of Amsterdam

May 09th, 2018

Outline

Causality: ubiquitous in the sciences

Genetics:
how to infer gene regulatory networks from micro-array data?

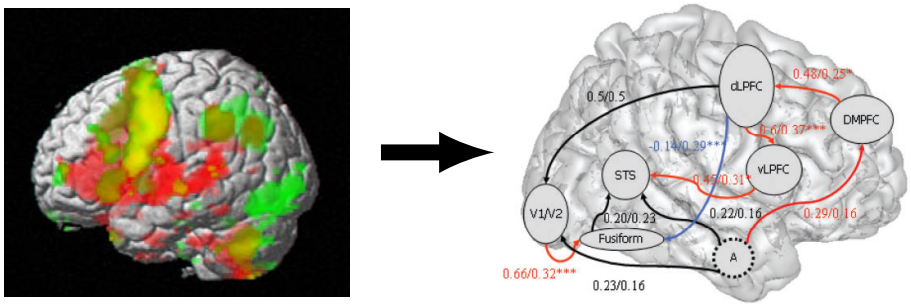


Causality: ubiquitous in the sciences

Social sciences:
does playing violent computer games cause aggressive behavior?

Causality: ubiquitous in the sciences

Neuroscience:
how to infer functional connectivity networks from fMRI data?



Causality: ubiquitous in the sciences

Economy:
Does austerity reduce national debt?

Causality: what is it?

Causality is central notion in science, decision-taking and daily life.

How to reason formally about cause and effect?

Question: give a definition of cause and effect.

Hume on Causality

The subject of *causality* has a long history in philosophy. For example, this is what Hume had to say about it:

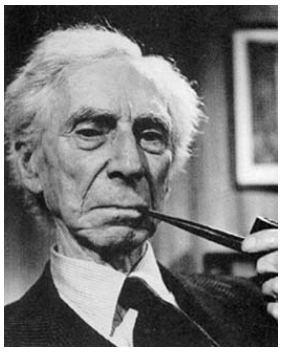
"Thus we remember to have seen that species of object we call *flame*, and to have felt that species of sensation we call *heat*. We likewise call to mind their constant conjunction in all past instances. Without any farther ceremony, we call the one *cause* and the other *effect*, and infer the existence of the one from that of the other."

David Hume, *Treatise of Human Nature*

But: does the rooster's crow really cause the sun to rise?

Russell on Causality

Some philosophers even proposed to abandon the concept of causality completely.

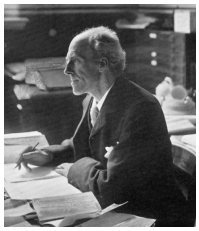


"All philosophers, of every school, imagine that causation is one of the fundamental axioms or postulates of science, yet, oddly enough, in advanced sciences such as gravitational astronomy, the word 'cause' never occurs. The law of causality, I believe, like much that passes muster among philosophers, is a relic of a bygone age, surviving, like the monarchy, only because it is erroneously supposed to do no harm."

Bertrand Russell, *On The Notion Of Cause*

Causality in Statistics

Karl Pearson (one of the founders of modern statistics, well-known from his work on the *correlation coefficient*) writes:



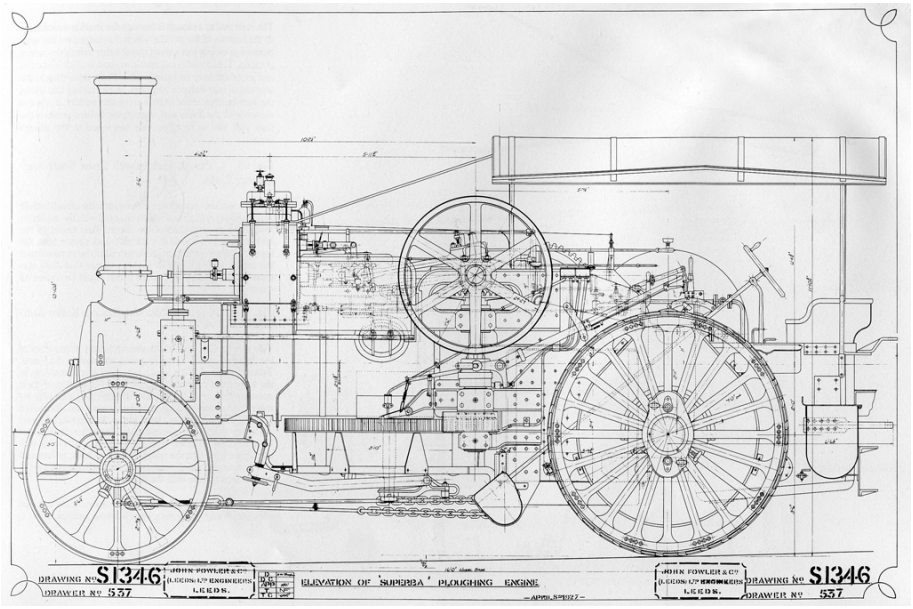
"Beyond such discarded fundamentals as 'matter' and 'force' lies still another fetish amidst the inscrutable arcana of even modern science, namely, the category of cause and effect."

Karl Pearson, *The Grammar of Science*

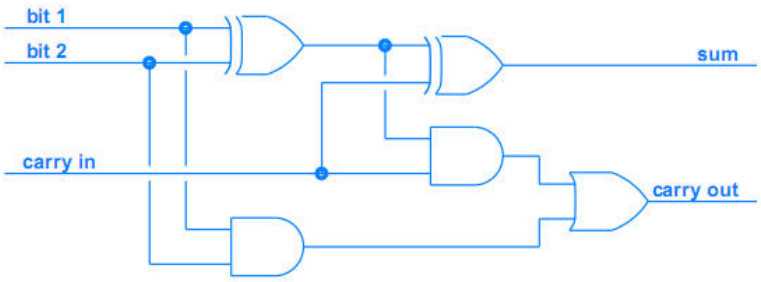
Since then, many statisticians tried to avoid causal reasoning:

- "Considerations of causality should be treated as they have always been in statistics: preferably not at all." (Terry Speed, former president of the Biometric Society).
- "It would be very healthy if more researchers abandon thinking of and using terms such as cause and effect." (Prominent social scientist).

Causality in engineering (1)



Causality in engineering (2)

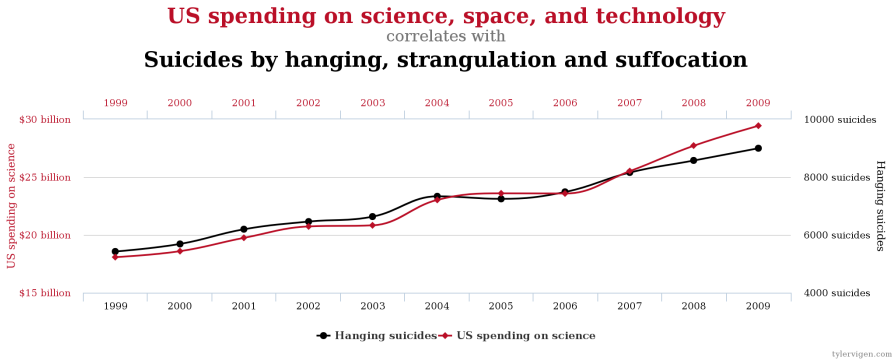


Causality is a very useful concept in engineering.

Using causal reasoning, engineers can not only predict what happens when a system operators normally, but also when an external *intervention* changes part of the system.

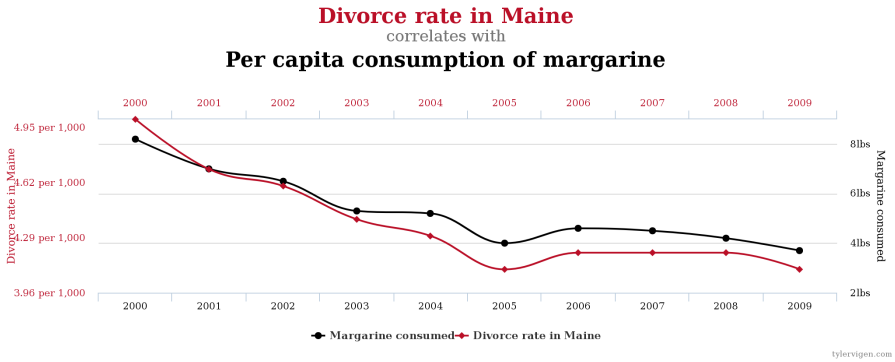
Being able to predict what happens under interventions allows to exert *control*.

Correlation vs. Causation (1)



Source: `http://tylervigen.com/spurious-correlations`

Correlation vs. Causation (2)



Source: `http://tylervigen.com/spurious-correlations`

A formal theory of causality?

Question

Can we formalize causal reasoning?

Exercise 1

Please make Exercise 1...

Problems in formalizing causal reasoning: probabilities

Example (Simpson's paradox)

We collect electronic patient records to investigate the effectiveness of a new drug against a certain disease. It can happen that:

1. The probability of recovery is higher for patients that took the drug:

$$p(\text{recovery} \mid \text{drug}) > p(\text{recovery} \mid \text{no drug})$$

2. For **both male and female** patients, the relation is **opposite**:

$$p(\text{recovery} \mid \text{drug}, \text{male}) < p(\text{recovery} \mid \text{no drug}, \text{male})$$

$$p(\text{recovery} \mid \text{drug}, \text{female}) < p(\text{recovery} \mid \text{no drug}, \text{female})$$

Would you use this drug for treatment?

Problems in formalizing causal reasoning: probabilities

Example (Simpson's paradox)

We collect electronic patient records to investigate the effectiveness of a new drug against a certain disease. It can happen that:

1. The probability of recovery is higher for patients that took the drug:

$$p(\text{recovery} \mid \text{drug}) > p(\text{recovery} \mid \text{no drug})$$

2. For **both male and female** patients, the relation is **opposite**:

$$p(\text{recovery} \mid \text{drug}, \text{male}) < p(\text{recovery} \mid \text{no drug}, \text{male})$$

$$p(\text{recovery} \mid \text{drug}, \text{female}) < p(\text{recovery} \mid \text{no drug}, \text{female})$$

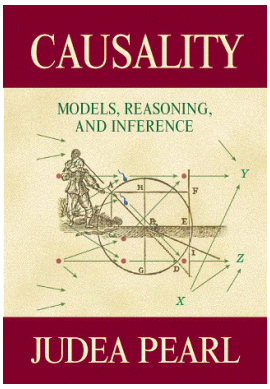
Would you use this drug for treatment?

Note

Fancy classifiers, deep learning and big data do not help us here!

An important step forwards

Judea Pearl



ACM Turing Award 2011: "For fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning."

Pearl's contribution: the do-operator

- Probability theory has a semantics for updating probabilities given *observations*: conditioning.
- Pearl extends probability calculus by introducing a new operator for describing *interventions*, the **do-operator**.

Example (Do-operator)

- $p(\text{recovery} \mid \text{drug})$: the probability that somebody recovers, given (the observation) that the person took the drug.
- $p(\text{recovery} \mid \text{do(drug)})$: the probability that somebody recovers, if we *force* the person to take the drug.

Resolution of Simpson's paradox:

- Simpson's paradox is only paradoxical if we misinterpret $p(\text{recovery} \mid \text{drug})$ as $p(\text{recovery} \mid \text{do(drug)})$.
- We should prescribe the drug if $p(\text{recovery} \mid \text{do(drug)}) > p(\text{recovery} \mid \text{do(no drug)})$.

Do-calculus

Pearl recognized that the rules of probability theory do not suffice for causal reasoning. He formulated three additional rules (the "**do-calculus**"):

1. **Ignoring observations**:

$$p(\boldsymbol{y} \mid \text{do}(\boldsymbol{x}), \boldsymbol{w}, \boldsymbol{z}) = p(\boldsymbol{y} \mid \text{do}(\boldsymbol{x}), \boldsymbol{w}) \qquad \text{if } (\boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{W})_{\mathcal{G}_{\overline{\boldsymbol{X}}}}$$

2. **Action/observation exchange**:

$$p(\boldsymbol{y} \mid \text{do}(\boldsymbol{x}), \text{do}(\boldsymbol{z}), \boldsymbol{w}) = p(\boldsymbol{y} \mid \text{do}(\boldsymbol{x}), \boldsymbol{z}, \boldsymbol{w}) \qquad \text{if } (\boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{W})_{\mathcal{G}_{\overline{\boldsymbol{X}},\underline{\boldsymbol{Z}}}}$$

3. **Ignoring actions**:

$$p(\boldsymbol{y} \mid \text{do}(\boldsymbol{x}), \text{do}(\boldsymbol{z}), \boldsymbol{w}) = p(\boldsymbol{y} \mid \text{do}(\boldsymbol{x}), \boldsymbol{w}) \qquad \text{if } (\boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{W})_{\mathcal{G}_{\overline{\boldsymbol{X}},\overline{\boldsymbol{Z(W)}}}}$$

   where $\boldsymbol{Z}(\boldsymbol{W}) = \boldsymbol{Z} \setminus An_{\mathcal{G}_{\overline{\boldsymbol{X}}}}(\boldsymbol{W})$.

The do-calculus allows us to reason with (probabilistic) causal statements, given (partial) knowledge of the causal structure.

Outline

1. Introduction
2. **Causality: Basic terminology**
3. Causal Bayesian Networks
4. Causal Reasoning: Back-door Criterion

Causal relations

Definition

A causes B if changing A leads to a change of B.

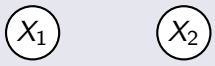
Causal relations

Definition

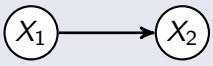
A causes *B* if changing *A* leads to a change of *B*.

Causal graph represents the causal relationships between variables (nodes are variables, edges encode causal relations between variables).

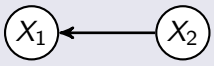
Example



X_1 and X_2 are causally unrelated

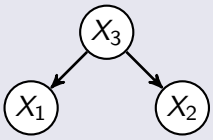


X_1 causes X_2

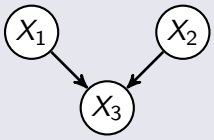


X_2 causes X_1

X_1 and X_2 cause each other



X_1 and X_2 have a common cause X_3



X_1 and X_2 have a common effect X_3

Direct causation

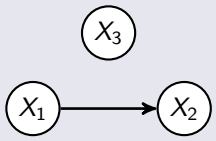
Let $\boldsymbol{V} = \{X_1, \ldots, X_N\}$ be a set of variables.

Definition

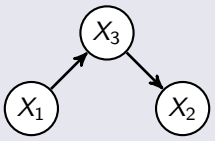
If X_i causes X_j even if all other variables $\boldsymbol{V} \setminus \{X_i, X_j\}$ are hold fixed at arbitrary values, then

- we say that X_i causes X_j directly with respect to $\boldsymbol{V}$
- we indicate this in the causal graph on $\boldsymbol{V}$ by a directed edge $X_i \to X_j$

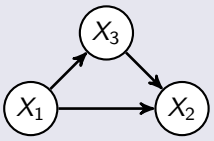
Example



X_1 causes X_2;
X_1 causes X_2 directly w.r.t. $\{X_1, X_2, X_3\}$



X_1 causes X_2;
X_1 does not cause X_2 directly w.r.t. $\{X_1, X_2, X_3\}$



X_1 causes X_2;
X_1 causes X_2 directly w.r.t. $\{X_1, X_2, X_3\}$

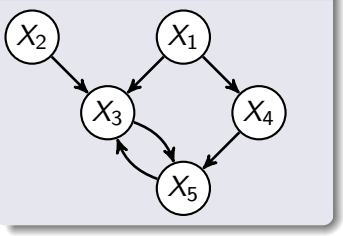
Terminology of directed graphs

Let $\mathcal{G}$ be a directed graph with nodes $\boldsymbol{V} = \{X_1, \ldots, X_N\}$.

Definition

- If $X_i \to X_j$ we call X_i parent of X_j and X_j a child of X_i.
- If $X_i \to X_j$ or $X_j \to X_i$ then we call X_i and X_j adjacent.
- If $X_{i_1} \to X_{i_2} \to X_{i_3} \to \cdots \to X_{i_k}$ we say that there is a directed path from X_{i_1} to X_{i_k}.
- If there is a directed path from X_i to X_j (or if $X_i = X_j$), X_i is called a ancestor of X_j, and X_j is called a descendant of X_i.
- $\mathrm{An}_{\mathcal{G}}(\boldsymbol{X})$ denotes the set of all ancestors of nodes in subset $\boldsymbol{X} \subseteq \boldsymbol{V}$.

Example

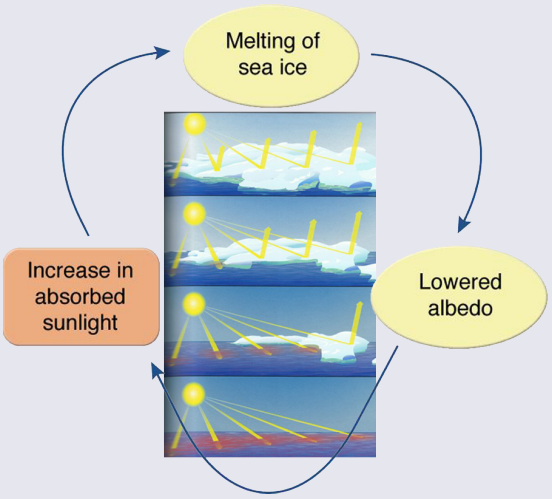


Causal interpretation

parent	= direct cause
child	= direct effect
ancestor	= cause
descendant	= effect

Feedback loops: Example

Example



Cycles, Feedback loops: Definitions

Let $\mathcal{G}$ be a directed graph with nodes $\boldsymbol{V} = \{X_1, \ldots, X_N\}$.

Definition

$\mathcal{G}$ is cyclic if it contains a directed cycle

$$X_{i_1} \to X_{i_2} \to \cdots \to X_{i_k}, \qquad X_{i_1} = X_{i_k}$$

If it does not contain such a directed cycle, the graph is called acyclic. This is also known as a DAG (Directed Acyclic Graph).

Definition

If A causes B and B causes A, then we say that A and B are involved in a causal feedback loop.

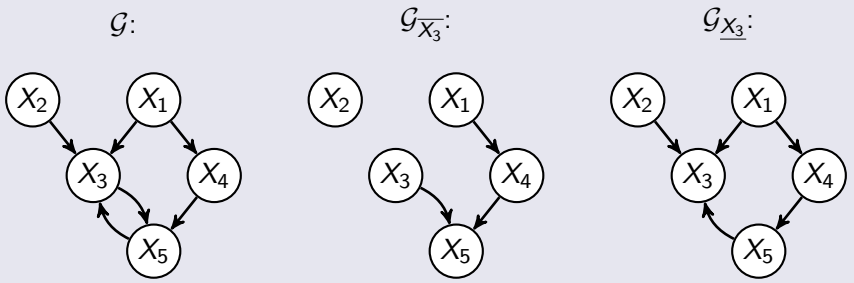
Mutilated graphs

Definition

Given a directed graph $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ and a subset $\boldsymbol{X} \subseteq \boldsymbol{V}$, we define

- $\mathcal{G}_{\overline{\boldsymbol{X}}}$ to be $\mathcal{G}$ without the incoming edges on nodes in $\boldsymbol{X}$;
- $\mathcal{G}_{\underline{\boldsymbol{X}}}$ to be $\mathcal{G}$ without the outgoing edges from nodes in $\boldsymbol{X}$.

Example



Perfect interventions

Definition

A perfect intervention $\text{do}(\boldsymbol{X} = \boldsymbol{\xi})$ on a set of variables $\boldsymbol{X} \subseteq \boldsymbol{V}$ is an externally enforced change of the system that ensures that $\boldsymbol{X} = \boldsymbol{\xi}$ but leaves the rest of the system untouched.

The concept of perfect intervention assumes "modularity": the causal system can be divided into two parts, $\boldsymbol{X}$ and $\boldsymbol{V} \setminus \boldsymbol{X}$, and we can make changes to one part while keeping the other part intact.

Note

The causal graph $\mathcal{G}$ changes into $\mathcal{G}_{\overline{\boldsymbol{X}}}$ after a perfect intervention $\text{do}(\boldsymbol{X} = \boldsymbol{\xi})$ (because none of the other variables can now cause $\boldsymbol{X}$).

Confounders: Example

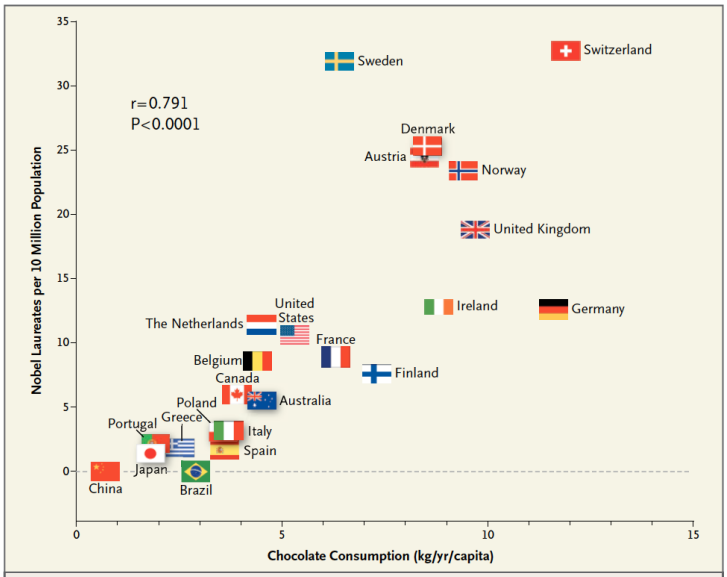


Figure 1. Correlation between Countries' Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.

Confounders: Definition

Definition

Let X, Y be observed variables and H an latent (unobserved) variable. H confounds X and Y if:

1. there exists a directed path from H to X that does not contain Y
2. there exists a directed path from H to Y that does not contain X

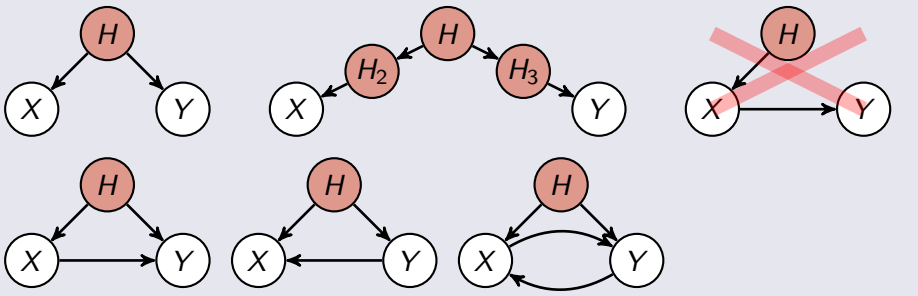
Confounders: Definition

Definition

Let X, Y be observed variables and H an latent (unobserved) variable.
H confounds X and Y if:

1. there exists a directed path from H to X that does not contain Y
2. there exists a directed path from H to Y that does not contain X

Example



(Conditional) independences

Definition: independence

Given two random variables X, Y, we write $X \perp\!\!\!\perp Y$ and say that X is independent of Y if

$$p(X, Y) = p(X)p(Y).$$

Intuitively, X is independent of Y if we do not learn anything about X when told the value of Y (or vice versa).

(Conditional) independences

Definition: independence

Given two random variables X, Y, we write $X \perp\!\!\!\perp Y$ and say that X is independent of Y if

$$p(X, Y) = p(X)p(Y).$$

Intuitively, X is independent of Y if we do not learn anything about X when told the value of Y (or vice versa).

Definition: conditional independence

Given a third random variable Z, we write $X \perp\!\!\!\perp Y \mid Z$ and say that X is (conditionally) independent from Y, given Z, if

$$p(X, Y \mid Z) = p(X \mid Z)p(Y \mid Z).$$

Intuitively, X is independent of Y if, given the value of Z, we do not learn anything new about X when told the value of Y.

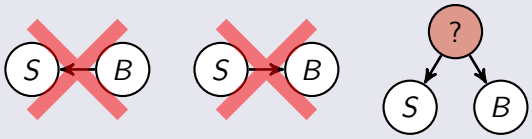
Reichenbach's Principle

Reichenbach's Principle of Common Cause

A dependence between X, Y implies that $X \to Y$, $Y \to X$, or there exists a confounder of X and Y (or any combination of these three).

Example

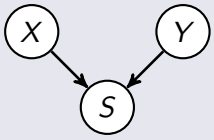
- Significant correlation ($p = 0.008$) between human birth rate and number of stork populations in European countries [Matthews, 2000]
- Most people nowadays do not believe that storks deliver babies (nor that babies deliver storks)
- There must be some confounder explaining the correlation



Selection Bias

Reichenbach's Principle may fail in case of selection bias.
If a data set is obtained by only including samples conditional on some event, *selection bias* may be introduced.

Example



X: the battery is empty
Y: the start engine is broken
S: the car does not start

- In general, X and Y are independent events: $X \perp\!\!\!\perp Y$.
- A car mechanic (who only observes cars for which $S = 1$) will observe a dependence between X and Y: $X \not\perp\!\!\!\perp Y \mid S$.
- When the car mechanic invokes Reichenbach's Principle without realizing that he is selecting on the value of S (maybe S is a latent variable), a wrong conclusion will be drawn.

Outline

1. Introduction
2. Causality: Basic Terminology
3. **Causal Bayesian Networks**
4. Causal Reasoning: Back-door Criterion

Assumptions

For simplicity, in this lecture we restrict our attention to a subclass of causal models.

Causal Bayesian Networks: Assumptions

Causal Bayesian Networks are a class of causal models that incorporate the following assumptions:

1. No confounding
2. No feedback
3. No selection bias
4. No measurement error
5. No time dependence

Extensions of the theory that drop one or more of these assumptions exist (see e.g. the literature on Acyclic Directed Mixed Graphs, Semi-Markov Causal Models, Maximal Ancestral Graphs, Structural Equation Models, d-connection graphs). This is an active area of research.

Bayesian Networks

Definition

A Bayesian Network is a pair $(\mathcal{G}, p)$ where:

- $\mathcal{G}$ is a Directed Acyclic Graph
- p is a joint probability density on the nodes $X_1, \ldots, X_N$ of $\mathcal{G}$ s.t.

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i \mid \boldsymbol{x}_{\mathrm{pa}(i)})$$

where $\mathrm{pa}(i)$ are the parents of X_i in $\mathcal{G}$.

Causal Bayesian Networks

Definition

A Bayesian Network is causal if:

- Directed edges correspond with direct causal relations
- After a perfect intervention $\text{do}(\boldsymbol{X}_I = \boldsymbol{x}_I)$, the incoming arrows on $\boldsymbol{X}_I$ are removed and the probability density becomes:

$$p\left(x_{\boldsymbol{V}\setminus\boldsymbol{I}} \mid \text{do}(\boldsymbol{X}_I = \boldsymbol{x}_I)\right) = \prod_{i \in \boldsymbol{V}\setminus\boldsymbol{I}} p(x_i \mid \boldsymbol{x}_{\text{pa}(i)})$$

(also known as the **Truncated Factorization Theorem**).

In other words, a perfect intervention $\text{do}(\boldsymbol{X}_I = \boldsymbol{x}_I)$ on a subset of variables $\boldsymbol{X}_I$ simply "divides out" the conditional densities $p(x_i \mid \boldsymbol{x}_{\text{pa}(i)})$ from the joint density for all $i \in I$, and substitutes the variables $\boldsymbol{X}_I$ by their values $\boldsymbol{x}_I$.

Local Markov Condition

Theorem

For any (Causal) Bayesian Network with variables $\{X_1, \ldots, X_N\}$*, the following "Local Markov Condition" holds:*

$$X_i \perp\!\!\!\perp X_{\mathrm{nd}(i)} \mid X_{\mathrm{pa}(i)}$$

for all $i = 1, \ldots, N$*. Here,* $\mathrm{nd}(i)$ *are the non-descendants of* X_i*.*

Remember: the descendants of X_i are all variables X_j such that there is a directed path $X_i \to \cdots \to X_j$.

Paths and colliders

Definition

Let $\mathcal{G}$ be a DAG with nodes $\boldsymbol{V} = \{X_1, \ldots, X_N\}$.

- A path $X_{i_1} \ldots X_{i_2} \ldots X_{i_k}$ is a sequence of distinct nodes such that X_{i_j} and $X_{i_{j+1}}$ are adjacent (for $j = 1, \ldots, k-1$).

Paths and colliders

Definition

Let $\mathcal{G}$ be a DAG with nodes $\boldsymbol{V} = \{X_1, \ldots, X_N\}$.

- A path $X_{i_1} \ldots X_{i_2} \ldots X_{i_k}$ is a sequence of distinct nodes such that X_{i_j} and $X_{i_{j+1}}$ are adjacent (for $j = 1, \ldots, k-1$).
- A collider on a path is a (non-endpoint) node X_{i_j} $(j = 2, \ldots, k-1)$ on the path with precisely two "incoming" arrow heads: $X_{i_{j-1}} \rightarrow X_{i_j} \leftarrow X_{i_{j+1}}$.

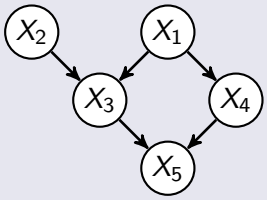
Paths and colliders

Definition

Let $\mathcal{G}$ be a DAG with nodes $\boldsymbol{V} = \{X_1, \ldots, X_N\}$.

- A path $X_{i_1} \ldots X_{i_2} \ldots X_{i_k}$ is a sequence of distinct nodes such that X_{i_j} and $X_{i_{j+1}}$ are adjacent (for $j = 1, \ldots, k-1$).
- A collider on a path is a (non-endpoint) node X_{i_j} $(j = 2, \ldots, k-1)$ on the path with precisely two "incoming" arrow heads: $X_{i_{j-1}} \rightarrow X_{i_j} \leftarrow X_{i_{j+1}}$.
- A non-collider on a path is any node X_{i_j} $(j = 1, \ldots, k)$ on the path which is not a collider.

Example



$X_1 \rightarrow X_3 \leftarrow X_1$ is not a path.
$X_2 \rightarrow X_3 \leftarrow X_1$ is a path.
$X_1 \rightarrow X_3 \rightarrow X_5 \leftarrow X_4 \leftarrow X_1$ is not a path.
The path $X_3 \rightarrow X_5 \leftarrow X_4$ contains a collider X_5.
The path $X_4 \leftarrow X_1 \rightarrow X_3$ contains no collider.

Blocked paths

Definition

Let $\mathcal{G}$ be a directed graph with nodes $\boldsymbol{V}$. Given a path p between nodes X and Y in $\boldsymbol{V}$, and a set of nodes $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, Y\}$, we say that $\boldsymbol{Z}$ blocks p if p contains

- a non-collider which is in $\boldsymbol{Z}$, or
- a collider which is *not* an ancestor of $\boldsymbol{Z}$.

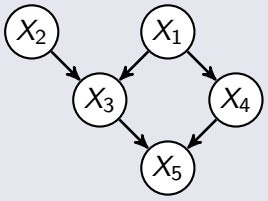
Blocked paths

Definition

Let $\mathcal{G}$ be a directed graph with nodes $\boldsymbol{V}$. Given a path p between nodes X and Y in $\boldsymbol{V}$, and a set of nodes $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, Y\}$, we say that $\boldsymbol{Z}$ blocks p if p contains

- a non-collider which is in $\boldsymbol{Z}$, or
- a collider which is *not* an ancestor of $\boldsymbol{Z}$.

Example



$X_3 \to X_5 \leftarrow X_4$ is blocked by $\emptyset$.
$X_3 \to X_5 \leftarrow X_4$ is blocked by $\{X_1\}$.
$X_3 \to X_5 \leftarrow X_4$ is not blocked by $\{X_5\}$.
$X_3 \leftarrow X_1 \to X_4$ is not blocked by $\emptyset$.
$X_2 \to X_3 \leftarrow X_1 \to X_4$ is blocked by $\{X_1\}$.
$X_2 \to X_3 \leftarrow X_1 \to X_4$ is not blocked by $\{X_5\}$.

d-separation

Let $\mathcal{G}$ be a directed graph with nodes $\boldsymbol{V}$.

Definition

Given two distinct nodes $X, Y \in \boldsymbol{V}$ and a set of nodes $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, Y\}$, we say that X and Y are d-separated by $\boldsymbol{Z}$ iff all paths between X and Y are blocked by $\boldsymbol{Z}$.
For three disjoint subsets $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \subseteq \boldsymbol{V}$ of nodes, we say that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated by $\boldsymbol{Z}$ iff all paths between any node in $\boldsymbol{X}$ and any node in $\boldsymbol{Y}$ are blocked by $\boldsymbol{Z}$.

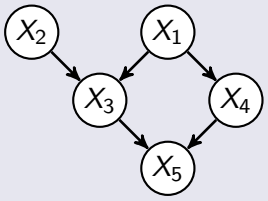
d-separation

Let $\mathcal{G}$ be a directed graph with nodes $\boldsymbol{V}$.

Definition

Given two distinct nodes $X, Y \in \boldsymbol{V}$ and a set of nodes $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, Y\}$, we say that X and Y are d-separated by $\boldsymbol{Z}$ iff all paths between X and Y are blocked by $\boldsymbol{Z}$.
For three disjoint subsets $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \subseteq \boldsymbol{V}$ of nodes, we say that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated by $\boldsymbol{Z}$ iff all paths between any node in $\boldsymbol{X}$ and any node in $\boldsymbol{Y}$ are blocked by $\boldsymbol{Z}$.

Example



X_2 and X_1 are d-separated by $\emptyset$.
X_2 and X_1 are d-separated by X_4.
X_2 and X_1 are not d-separated by X_5.
X_3 and X_4 are not d-separated by $\emptyset$.
X_3 and X_4 are d-separated by X_1.
X_3 and X_4 are not d-separated by $\{X_1, X_5\}$.

Global Markov Condition

Theorem

In any (Causal) Bayesian Network, the following "Global Markov Condition" holds:

$$\boldsymbol{X}, \boldsymbol{Y} \text{ d-separated by } \boldsymbol{Z} \quad \Longrightarrow \quad \boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{Z}$$

for all subsets $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ of nodes.

In other words, we can read off conditional independences from the graph of a Bayesian Network by using the Global Markov Condition.

Outline

1. Introduction
2. Causality: Basic Terminology
3. Causal Bayesian Networks
4. **Causal Reasoning: Back-door Criterion**

Identifiability

Given i.i.d. data of the observational distribution $p(x, y, \dots)$.
From this we can estimate $p(y \mid X = x)$.

Question

Can we also estimate $p(y \mid \mathrm{do}(X = x))$ from the observational data?

Identifiability

Given i.i.d. data of the observational distribution $p(x, y, \dots)$.
From this we can estimate $p(y \mid X = x)$.

Question

Can we also estimate $p(y \mid \text{do}(X = x))$ from the observational data?

Given enough assumptions, the answer is yes. In that case, we do not have to actually perform the intervention experiment!

Definition

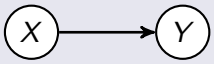
If a quantity like $p(y \mid \text{do}(X = x))$ can be expressed in terms of the observational distribution $p(x, y, \dots)$, we say that it is identifiable (from the observational distribution).

Identifiability: Example

Example

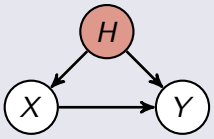
Is $p(y \mid \text{do}(X = x))$ identifiable?

identifiable:



$p(y \mid \text{do}(X = x)) = p(y \mid X = x)$

not identifiable:



$p(y \mid \text{do}(X = x)) \neq p(y \mid X = x)$

Indeed, for the graph with the latent variable H:

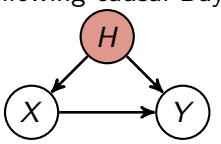
$$p(y \mid \text{do}(X = x)) = \int p(h)p(y \mid x, h)\, dh$$

which is generally different from

$$p(y \mid X = x) = \int p(h \mid x)p(y \mid x, h)\, dh.$$

Adjustment for covariates

- We have seen that for the following causal Bayesian network,



  adjusting for the confounder H, i.e.,

$$p(y \mid \mathrm{do}(X = x)) = \int p(h)p(y \mid x, h)\, dh$$

  yields the causal effect of X on Y.

- More generally, given a causal Bayes network: which covariates $\boldsymbol{S}$ could we adjust for, in order to express the causal effect on Y of intervening on X in terms of the observed distribution?
- A sufficient condition is given by Pearl's Back-door criterion.

The Back-door Criterion

The following result is known as the "Back-door Criterion":

Theorem

A set $\boldsymbol{S}$ of nodes is "admissible" for adjustment to find the causal effect of X on Y, if:

1. $X, Y \notin \boldsymbol{S}$;
2. *no element of $\boldsymbol{S}$ is a descendant of X;*
3. *$\boldsymbol{S}$ blocks all back-door paths $X \leftarrow \ldots Y$ (all paths between X and Y that start with an incoming edge on X).*

In that case,

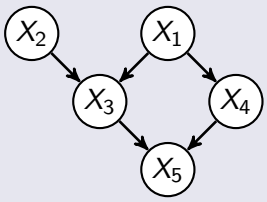
$$p(y \mid \mathrm{do}(X = x)) = \int p(y \mid x, \boldsymbol{s}) p(\boldsymbol{s}) \, d\boldsymbol{s}.$$

For the special case $\boldsymbol{S} = \emptyset$, this simply should be read as:

$$p(y \mid \mathrm{do}(X = x)) = p(y \mid x).$$

The Back-door Criterion: Example

Example



- $\{X_1\}$ is admissible for adjustment to find the causal effect of X_4 on X_5.
- $\emptyset$ is admissible for adjustment to find the causal effect of X_2 on X_5.
- $\{X_1\}$ is admissible for adjustment to find the causal effect of X_2 on X_5.
- $\{X_1, X_4\}$ is admissible for adjustment to find the causal effect of X_2 on X_5.
- $\{X_3\}$ is not admissible for adjustment to find the causal effect of X_2 on X_5.
- $\{X_1, X_3\}$ is admissible for adjustment to find the causal effect of X_5 on X_2.

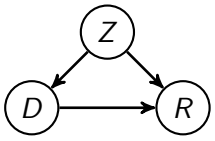
Exercise 2

Please make Exercise 2...

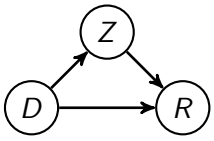
Simpson's paradox resolved

R stands for *Recovery*, D for taking the *Drug*, Z for *Gender*.
Two possible causal models:

(1)



(2)



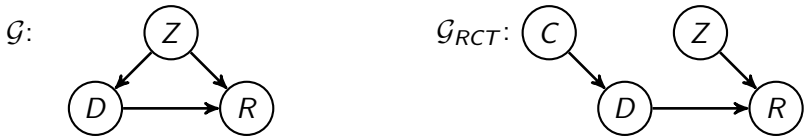
Using the back-door criterion (or do-calculus) one can derive:

1. $p(R \mid \mathrm{do}(D)) = \sum_Z p(R \mid D, Z)p(Z)$
   We should **not prescribe** the drug (for both males and females, probability of recovery is lower for those who took the drug).
2. $p(R \mid \mathrm{do}(D)) = p(R \mid D)$
   We should **prescribe** the drug (in the general population, probability of recovery is higher for those who took the drug).

Note that (2) seems unlikely, but if we would replace *gender* by e.g. *blood pressure* it is no longer obvious which model is more likely *a priori*.

Randomized controlled trials

If possible, the best way to find causal relationships and effect sizes is to use a randomized controlled trial.



R: Recovery, D: Drug, Z: latent confounders (e.g., genetics), C: coin flip.

- Divide patients into two groups: treatment and control.
- Which patient is assigned to which group is completely random.
- Patients in the treatment group are forced to take a drug, and patients in the control group are forced to not take the drug (but rather a placebo).
- Estimating the causal effect of the drug now becomes a standard statistical exercise, as $p(R \mid C) = p(R \mid \mathrm{do}(C))$.
- The RCT intervention breaks any back-door paths.

Conclusion: Causal vs. probabilistic reasoning

Traditional statistics, machine learning

- About **associations** (stork population and human birth rate are correlated)

Causality

- About **causation** (storks do not causally affect human birth rate)

Conclusion: Causal vs. probabilistic reasoning

Traditional statistics, machine learning

- About **associations** (stork population and human birth rate are correlated)
- Model the **distribution** of the data

Causality

- About **causation** (storks do not causally affect human birth rate)
- Model the **mechanism** that generates the data

Conclusion: Causal vs. probabilistic reasoning

Traditional statistics, machine learning

- About **associations** (stork population and human birth rate are correlated)
- Model the **distribution** of the data
- Predict given **observations** (if we **observe** a certain number of storks, what is our best estimate of human birth rate?)

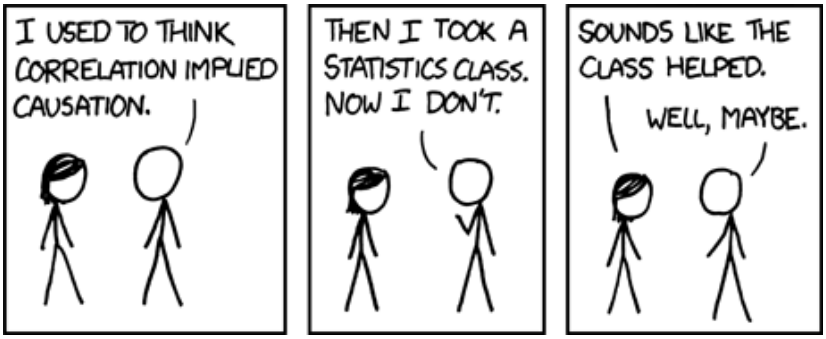
Causality

- About **causation** (storks do not causally affect human birth rate)
- Model the **mechanism** that generates the data
- Predict results of **interventions** (if we **change** the number of storks, what will happen with the human birth rate?)

Further reading

- Pearl, J. (1999).
  Simpson's paradox: An anatomy.
  Technical Report R-264, UCLA Cognitive Systems Laboratory.
- Pearl, J. (2000).
  Causality: Models, Reasoning, and Inference.
  Cambridge University Press.
- Pearl, J. (2009).
  Causal inference in statistics: An overview.
  Statistics Surveys, 3:96–146.
- Spirtes, P., Glymour, C., and Scheines, R. (2000).
  Causation, Prediction, and Search.
  The MIT Press.

Thank you for your attention!



Randall Munroe, www.xkcd.org