Reasoning and Decision-Making under Uncertainty

11. Session:

Learning Bayesian Models (Parameter learning)

Prof. Dr.-Ing. Stefan Kopp
Center of Excellence „Cognitive Interaction Technology“
AG Sociable Agents

D. Barber (2012), Bayesian Reasoning and Machine Learning, Ch. 8+9

CITEC

Sociable Agents

Bayesian networks as models

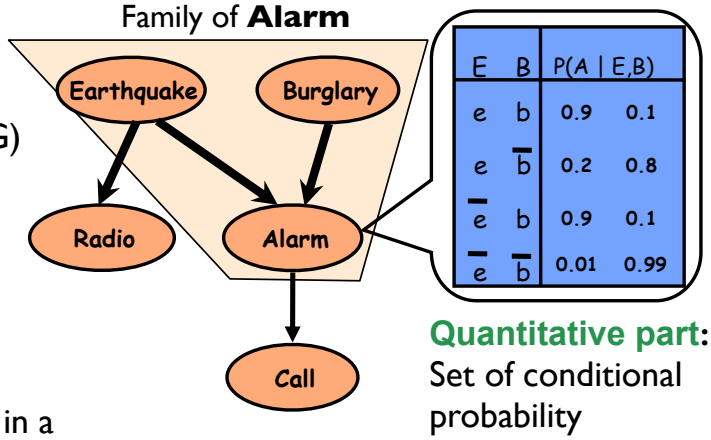
Compact representation of probability distributions via conditional independence

Qualitative part:

Directed acyclic graph (DAG)

Nodes - random variables

Edges - direct influence



Quantitative part:
Set of conditional probability distributions

Together:
Define a unique distribution in a factored form:

$$P(B,E,A,C,R) = P(B)P(E)P(A \mid B,E)P(R \mid E)P(C \mid A)$$

How to get a Bayesian network in general?

Method I: Model from given facts/knowledge, where possible

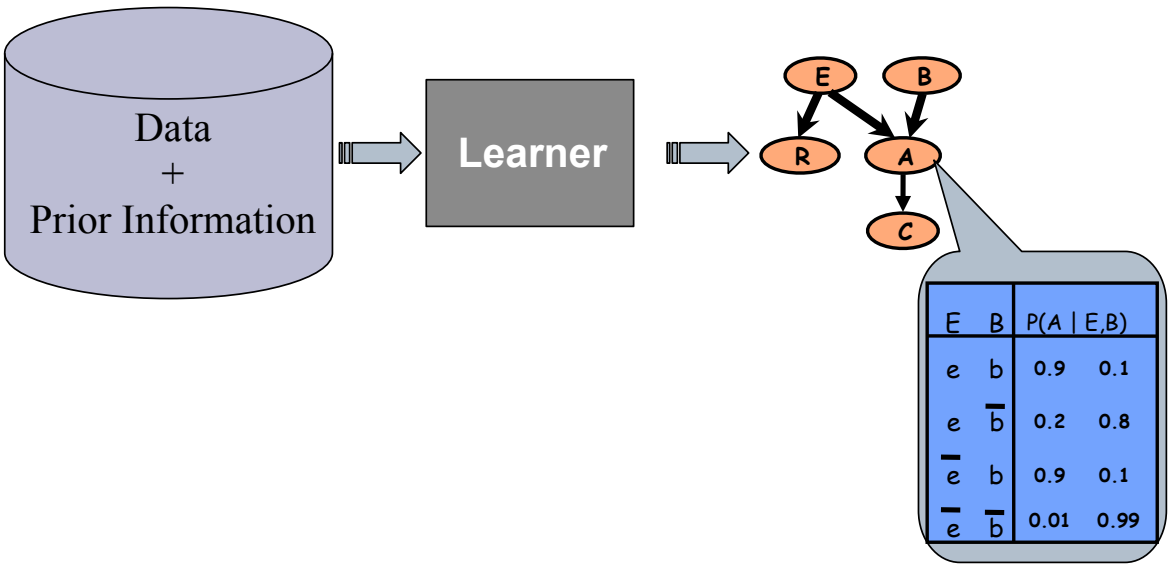
- variables (evidence, query, intermediary) and their possible values
- network structure (cause-effect relations, conditional independencies)
- network parameters (CPTs)

Method II: Learn from data, either partially or completely

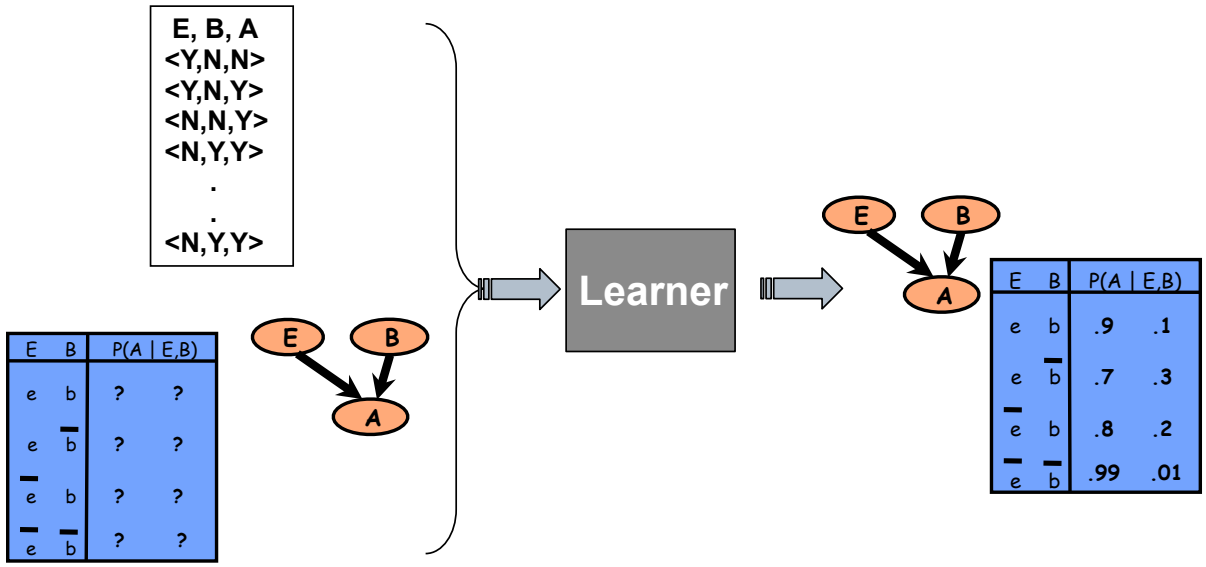
- collect or gather data and prior information: enough samples, cover all relevant variables, cover all relevant events (variations)
- use to learn network parameters (distributions) or structure

Also combination of both methods if possible, but most often one has to use learning to build a Bayesian model

Learning Bayesian networks

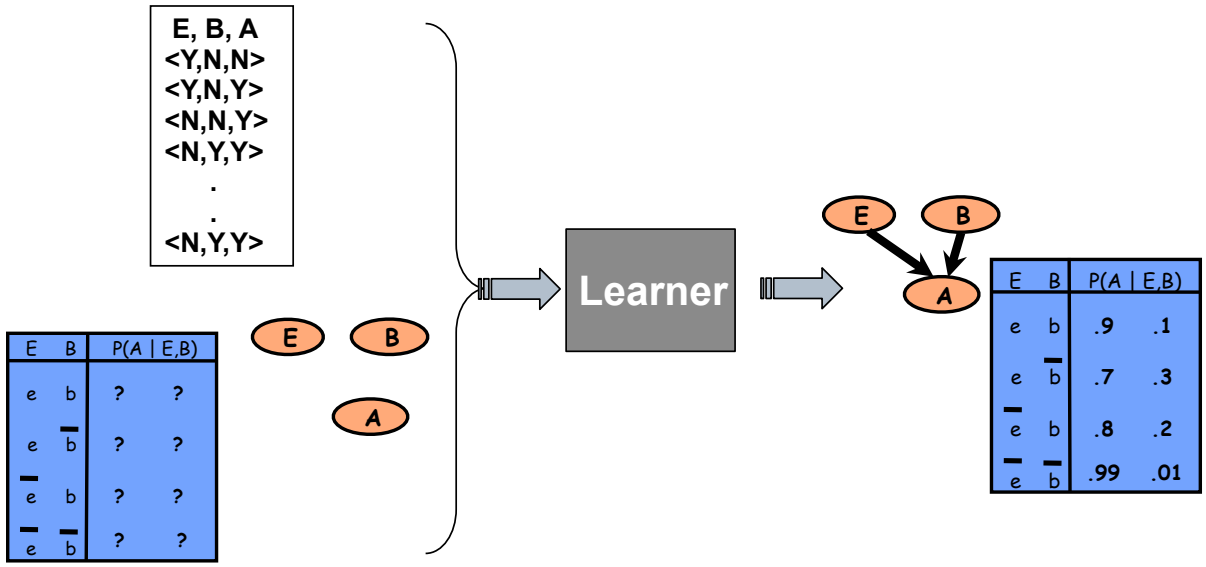


Known Structure, Complete Data



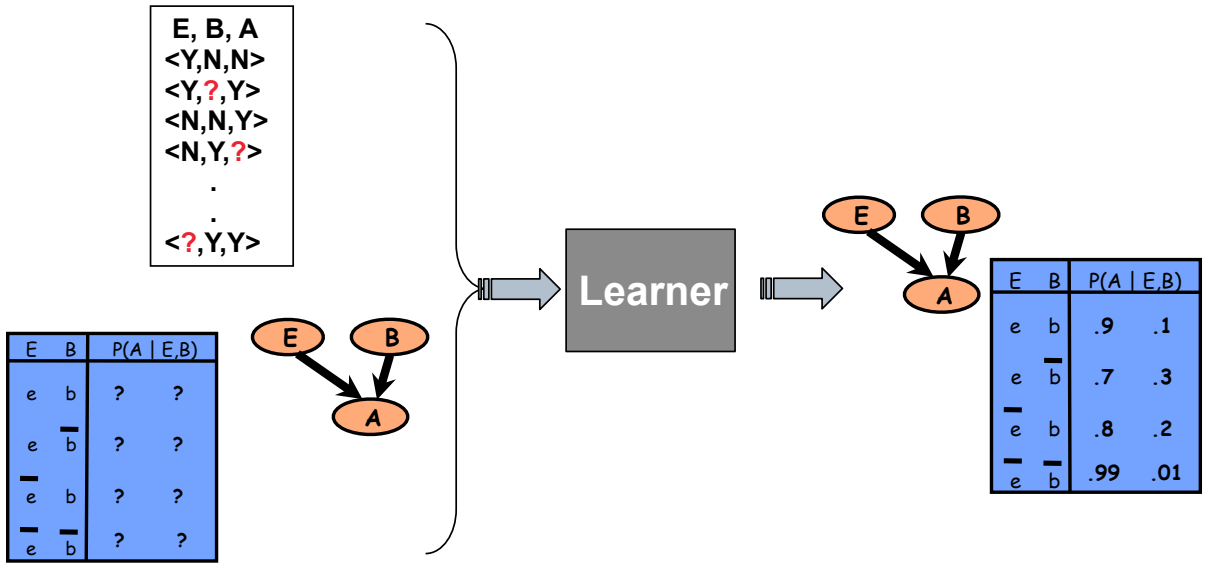
Network structure is specified, learner needs to estimate parameters

Unknown Structure, Complete Data



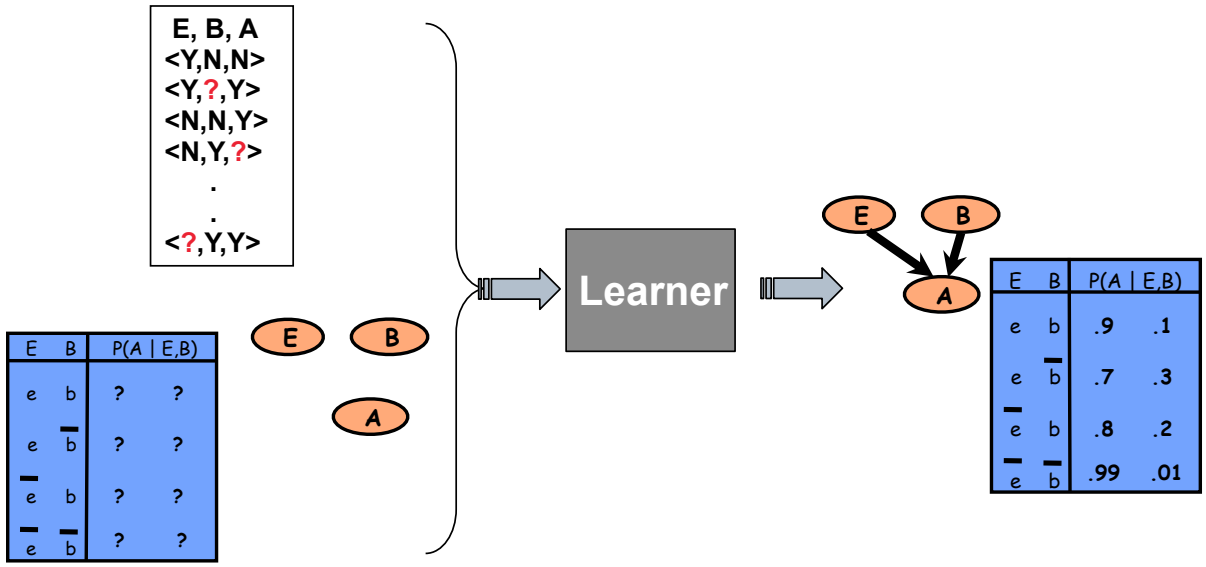
Network structure is *not* specified, learner needs to select graph structure & estimate parameters

Known Structure, Incomplete Data



Network structure is specified but data contains missing values, learner need to consider assignments to missing values

Unknown Structure, Incomplete Data



Network structure is *not* specified and data contains missing values, learner needs to select graph structure, estimate parameters, assign missing values

Example: Learning the bias of a coin

$$v^n = \begin{cases} 1 & \text{if on toss } n \text{ the coin comes up heads} \\ 0 & \text{if on toss } n \text{ the coin comes up tails} \end{cases}$$

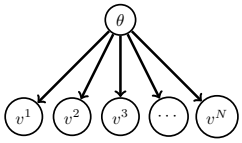
Our aim is to estimate the probability θ that the coin will be a head, $p(v^n = 1|\theta) = \theta$ – called the 'bias' of the coin.

Approach: Learning as probabilistic inference over variables

Building a model

The variables are $v^1, \ldots, v^N$ and θ and we require a model of the probabilistic interaction of the variables, $p(v^1, \ldots, v^N, \theta)$. Assuming there is no dependence between the observed tosses, except through θ, we have the belief network

$$p(v^1, \ldots, v^N, \theta) = p(\theta) \prod_{n=1}^{N} p(v^n|\theta)$$



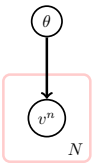


Plate notation

Example: Learning the bias of a coin

The posterior

$$\begin{aligned} p(\theta|v^1, \ldots, v^N) &\propto p(\theta) \prod_{n=1}^{N} p(v^n|\theta) \\ &= p(\theta) \prod_{n=1}^{N} \theta^{\mathbb{I}[v^n=1]} (1-\theta)^{\mathbb{I}[v^n=0]} \\ &\propto p(\theta) \theta^{\sum_{n=1}^{N} \mathbb{I}[v^n=1]} (1-\theta)^{\sum_{n=1}^{N} \mathbb{I}[v^n=0]} \end{aligned}$$

Hence

$$p(\theta|v^1, \ldots, v^N) \propto p(\theta) \theta^{N_H} (1-\theta)^{N_T}$$

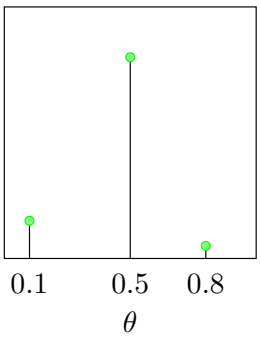
$N_H = \sum_{n=1}^{N} \mathbb{I}[v^n = 1]$ is the number of occurrences of heads.
$N_T = \sum_{n=1}^{N} \mathbb{I}[v^n = 0]$ is the number of tails.

Example: Learning the bias of a coin

The prior

We still need to fully specify the prior $p(\theta)$. To avoid complexities resulting from continuous variables, we'll consider a discrete θ with only three possible states, $\theta \in \{0.1, 0.5, 0.8\}$. Specifically, we assume

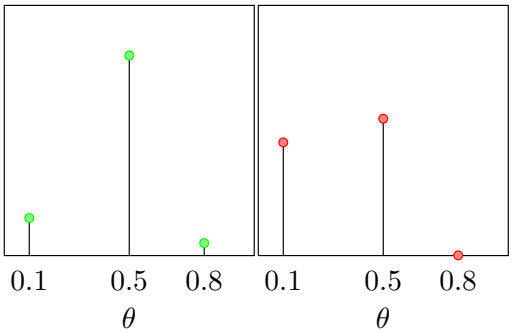
$$p(\theta = 0.1) = 0.15,\ p(\theta = 0.5) = 0.8,\ p(\theta = 0.8) = 0.05$$



Example: Learning the bias of a coin

Coin posterior after observations

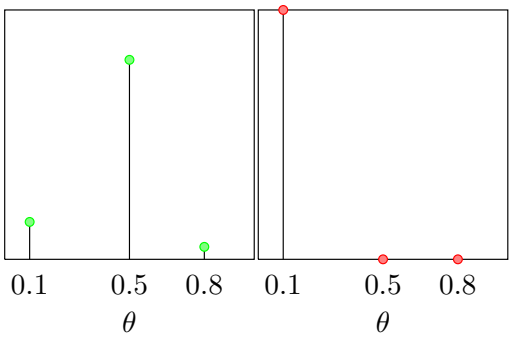
For an experiment with $N_H = 2$, $N_T = 8$, the posterior distribution is



If we were asked to choose a single *a posteriori* most likely value for θ, it would be $\theta = 0.5$, although our confidence in this is low since the posterior belief that $\theta = 0.1$ is also appreciable. This result is intuitive since, even though we observed more Tails than Heads, our prior belief was that it was more likely the coin is fair.

Example: Learning the bias of a coin

Repeating the above with $N_H = 20$, $N_T = 80$, the posterior changes to



so that the posterior belief in $\theta = 0.1$ dominates. There are so many more tails than heads that this is unlikely to occur from a fair coin. Even though we *a priori* thought that the coin was fair, *a posteriori* we have enough evidence to change our minds.

The posterior effect

Note that in both examples, $N_T/N_H = 4$, although in the latter we are much more confident that $\theta = 0.1$

Math background: Distributions

- univariate p(x) or multivariate p(**x**)
- mode: state(s) where p takes its maximum value $x_* = \underset{x}{\operatorname{argmax}}\, p(x)$
- average or expectation of f(x) w.r.t. distr. p(x) $\mathbb{E}(x)$

  $\langle f(x) \rangle \equiv \sum_{\mathrm{x}} f(x = \mathrm{x}) p(x = \mathrm{x})$ $\langle f(x) \rangle \equiv \int_{-\infty}^{\infty} f(x) p(x) dx$

  - average of f(x) conditioned on y (w.r.t. to distribution p(x|y)): $\langle f(x)|y \rangle$
- mean: first moment of a distr. $\mu \equiv \langle x \rangle$
- variance: $\sigma^2 \equiv \left\langle (x - \langle x \rangle)^2 \right\rangle_{p(x)}$ (square root = standard deviation)
- covariance matrix of multivariate distr.: $\Sigma_{ij} = \langle (x_i - \mu_i)(x_j - \mu_j) \rangle$
- correlation matrix: $\rho_{ij} = \left\langle \frac{(x_i - \mu_i)}{\sigma_i} \frac{(x_j - \mu_j)}{\sigma_j} \right\rangle$

Math background: Distributions

- Skewness of p:
  - >0: p is heavy-tail to the right
  - <0: p is heavy-tail to the left

$$\gamma_1 \equiv \frac{\left\langle (x - \langle x \rangle)^3 \right\rangle_{p(x)}}{\sigma^3}$$

- Kurtosis of p: how peaked p is around the mean
  - >0: more mass around the mean than a Gaussian with similar mean and variance (super Gaussian)
  - <0: less mass around the mean (sub Gaussian)

$$\gamma_2 \equiv \frac{\left\langle (x - \langle x \rangle)^4 \right\rangle_{p(x)}}{\sigma^4} - 3$$

- Dirac delta function: continuous function that is 0 everywhere except at x0, with

$\delta(x - x_0)$

$$\int_{-\infty}^{\infty} \delta(x - x_0) dx = 1 \qquad \int_{-\infty}^{\infty} \delta(x - x_0) f(x) dx = f(x_0)$$

  - discrete case: Kronecker delta δ_{x,x_0}

Math background: Distributions

Unbiased estimator: given data **X** from a distr. $p(x|\theta)$, we can use x1,...,xn to estimate the parameter d that was used to generate **X**. The estimator is a function of the data: $\hat{\theta}(\mathcal{X})$

- for an unbiased estimator: $\left\langle \hat{\theta}(\mathcal{X}) \right\rangle_{p(\mathcal{X}|\theta)} = \theta$

Kullback-Leibler Divergence KL(q|p)

- measure of the „difference“ between two distr. q and p

$$\mathrm{KL}(q|p) \equiv \langle \log q(x) - \log p(x) \rangle_{q(x)} \geq 0$$

- KL(q|p) = 0 ***iff*** p and q are exactly the same

Math background: Distributions

Classical discrete distributions

- Bernoulli distribution: discrete variable x∈{0,1}, with

$$p(x=1)=\theta \qquad p(x=0)=1-\theta$$

$$\langle x\rangle = 0\times p(x=0)+1\times p(x=1)=\theta \qquad \mathrm{var}(x)=\theta\,(1-\theta)$$

- Categorical distribution: generalization to states {1,...,C}

$$p(x=c)=\theta_c, \qquad \sum_c \theta_c = 1$$

- Binomial distribution: discrete, two-state symbolic variable. Prob. to observe k `success´states 1 in n *Bernoulli* trials is

$$p(y=k|\theta)=\binom{n}{k}\theta^k\,(1-\theta)^{n-k} \qquad \langle y\rangle = n\theta, \qquad \mathrm{var}(x)=n\theta\,(1-\theta)$$

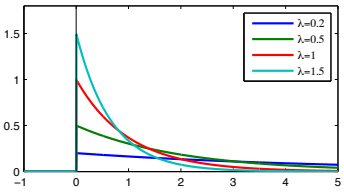
Math background: Distributions

Classical continuous distributions

- Exponential distribution: for x≥0

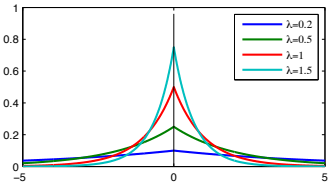
$$p(x|\lambda)\equiv \lambda e^{-\lambda x} \qquad \langle x\rangle = \frac{1}{\lambda}, \qquad \mathrm{var}(x)=\frac{1}{\lambda^2}$$

$b=1/\lambda$ is called the scale.



- Laplace (double exponential) distribution:

$$p(x|\lambda)\equiv \lambda e^{-\frac{1}{b}|x-\mu|} \qquad \langle x\rangle = \mu, \qquad \mathrm{var}(x)=2b^2$$



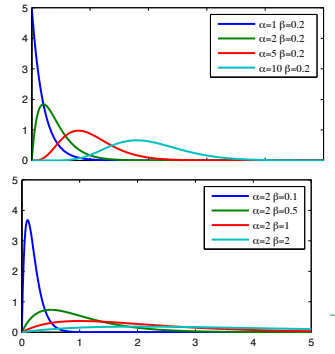
- Gamma distribution:

$$Gam\,(x|\alpha,\beta)=\frac{1}{\beta\Gamma(\gamma)}\left(\frac{x}{\beta}\right)^{\alpha-1}e^{-\frac{x}{\beta}}, \qquad x\geq 0, \alpha>0, \beta>$$

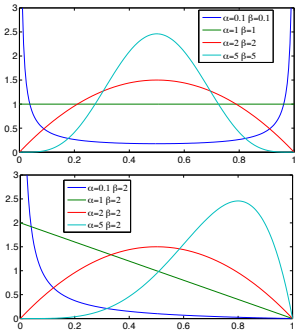
α is called the shape parameter, β is the scale parameter and

$$\Gamma(a)=\int_0^\infty t^{a-1}e^{-t}dt \qquad \alpha=\left(\frac{\mu}{s}\right)^2 \quad \beta=\frac{s^2}{\mu}$$

(s std. deviation)



Math background: Distributions



Classical continuous distributions

- Beta distribution:

$$p(x|\alpha,\beta) = B(x|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad 0 \le x \le 1$$

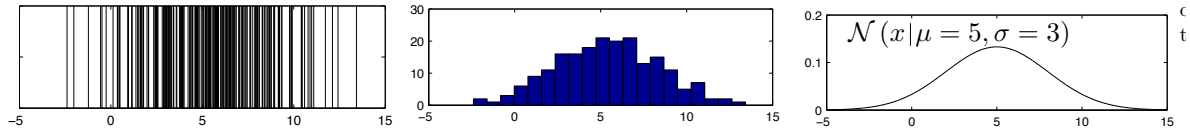
where the beta function is defined as

$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \qquad \langle x \rangle = \frac{\alpha}{\alpha+\beta} \qquad \mathrm{var}(x) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

- Univariate Gaussian distribution (Normal distribution)

$$p(x|\mu,\sigma^2) = \mathcal{N}(x|\mu,\sigma^2) \equiv \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

$$\mu = \langle x \rangle_{\mathcal{N}(x|\mu,\sigma^2)}, \qquad \sigma^2 = \left\langle (x-\mu)^2 \right\rangle_{\mathcal{N}(x|\mu,\sigma^2)}$$



Math background: Distributions

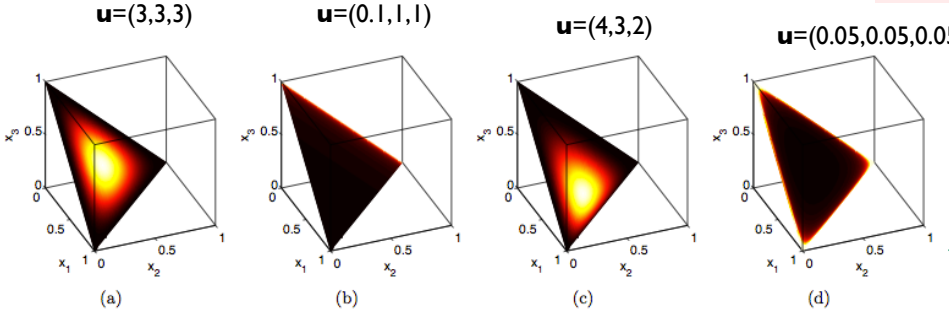
Continuous multivariate distributions:

- Dirichlet distribution: a distribution on distributions

$$p(\alpha) = \frac{1}{Z(\mathbf{u})} \delta\left(\sum_{i=1}^{Q} \alpha_i - 1\right) \prod_{q=1}^{Q} \alpha_q^{u_q - 1} \qquad Z(\mathbf{u}) = \frac{\prod_{q=1}^{Q} \Gamma(u_q)}{\Gamma\left(\sum_{q=1}^{Q} u_q\right)} \qquad \mathrm{Dirichlet}(\alpha|\mathbf{u})$$

- **u**: controls how much the mass is pushed to corners of the simplex
- closed under multiplication: $\mathrm{Dirichlet}(\theta|\mathbf{u}_1)\,\mathrm{Dirichlet}(\theta|\mathbf{u}_2) = \mathrm{Dirichlet}(\theta|\mathbf{u}_1+\mathbf{u}_2)$
- Marginal of a single component is a Beta distribution

$$p(\theta_i) = B\left(\theta_i|u_i, \sum_{j\neq i} u_j\right)$$



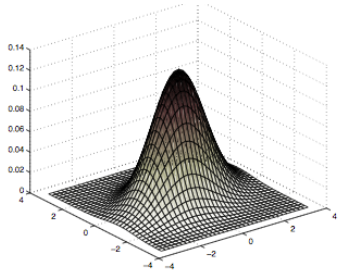
Math background: Distributions

Continuous multivariate distributions:

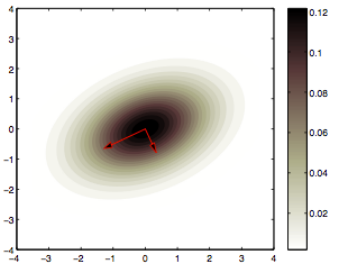
- Multivariate Gaussian distribution

$$p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \equiv \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

$\boldsymbol{\mu}$ is the mean vector of the distribution, and $\boldsymbol{\Sigma}$ the covariance matrix.



$$\boldsymbol{\mu} = \langle \mathbf{x} \rangle_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})} \qquad \boldsymbol{\Sigma} = \left\langle (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}} \right\rangle_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}$$



- Some useful properties
  - product of two Gaussians is a Gaussian
  - can be conveniently transformed into Eigenvalues of the Covariance matrix
  - can be shifted into linearly transformed parameters
  - Entropy independent of mean:

$$H(\mathbf{x}) \equiv -\langle \log p(\mathbf{x}) \rangle_{p(\mathbf{x})} = \frac{1}{2}\log\det(2\pi\boldsymbol{\Sigma}) + \frac{D}{2} \qquad D = \dim \mathbf{x}$$

Learning distributions

For a distribution p(x|θ) and data $\mathbf{X}$={x1,...,xN}, learning corresponds to inferring the parameter θ that best explains data $\mathbf{X}$. Using Bayes:

posterior p(θ|$\mathbf{X}$) = likelihood p($\mathbf{X}$|θ) * prior p(θ) / evidence p($\mathbf{X}$)

- Bayesian methods: examine posterior p(θ|x) $\propto$ p($\mathbf{X}$|θ) p(θ). This gives rise to a distribution over θ.
- Maximum a posteriori: θ^{MAP}=argmax$_\theta$ p(θ|$\mathbf{X}$)
- Maximum likelihood: Under a flat prior p(θ)=const., the MAP solution is equivalent to setting θ to the value that maximizes the likelihood of observing the data: θ^{ML}=argmax$_\theta$ p($\mathbf{X}$|θ)

Learning distributions

Often, a numerical optimization is required to single out the best parameter value. Thus it is important to find a good model that makes computation feasible, or to find good approximations.

Often, the distributions are also conditioned on the model *M*

- $p(\theta|\mathbf{X},M) = p(\mathbf{X}|\theta,M)\ p(\theta|M) / p(\mathbf{X}|M)$
  - model likelihood $p(\mathbf{X}|M)$

For a set of observations x1,...,xN, conditioned on θ, we say the $\mathbf{X}$ are independent and identically distributed (i.i.d.) if there is no dependence between the observations: $p(\mathbf{X}|\theta)=\prod p(xi|\theta)$

Maximum likelihood estimation (MLE)

With i.i.d. data samples D={x[m]}, m=1...N, what are the parameters Θ that makes sampling x from p(x|Θ) as likely as possible?

Maximize $P(D \mid \theta) = \prod_m P(x[m] \mid \theta)$

Direct approach:

- maximize the log likelihood: $\log(L) = \sum_m \log p(x[m]|\Theta)$
- write down derivative $dL/d\Theta$ with respect to each parameter and solve for 0

In practice, one is often interested in (assumed) certain distributions, whose parameter(s) should be learned

Maximum likelihood learning

Given a set of training data $\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$, drawn from a Gaussian $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with unknown mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, how can we find these parameters? Assuming the data are drawn i.i.d. the log likelihood is

$$L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \equiv \sum_{n=1}^{N} \log p(x|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{1}{2} \sum_{n=1}^{N} (\mathbf{x}^n - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\mathbf{x}^n - \boldsymbol{\mu}) - \frac{N}{2} \log \det (2\pi\boldsymbol{\Sigma}) \qquad (8.6.37)$$

Direct approach:

- optimal mean: search for zero vector derivative

$$\nabla_{\boldsymbol{\mu}} L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \boldsymbol{\Sigma}^{-1} (\mathbf{x}^n - \boldsymbol{\mu}) \qquad \sum_{n=1}^{N} \boldsymbol{\Sigma}^{-1} \mathbf{x}^n = N\boldsymbol{\mu}\boldsymbol{\Sigma}^{-1} \qquad \boldsymbol{\mu} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}^n$$

- optimal covariance: setting derivative w.r.t. the covariance matrix to zero gives $\boldsymbol{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}^n - \boldsymbol{\mu})(\mathbf{x}^n - \boldsymbol{\mu})^\mathsf{T}$

➔ max. likelihood solution for training data ✘ simply sets parameters to sample statistics of the empirical distribution, i.e. we can count.

Example

Consider the following model of the relationship between exposure to asbestos (a), being a smoker (s) and the incidence of lung cancer (c)

$$p(a, s, c) = p(c|a, s)p(a)p(s)$$

Each variable is binary, $\mathrm{dom}(a) = \{0, 1\}$, $\mathrm{dom}(s) = \{0, 1\}$, $\mathrm{dom}(c) = \{0, 1\}$. Furthermore, we assume that we have a list of patient records, where each row represents a patient's data.

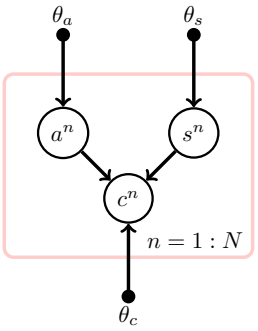
a	s	c
1	1	1
1	0	0
0	1	1
0	1	0
1	1	1
0	0	0
1	0	1

A database containing information about the Asbestos exposure (1 signifies exposure), being a Smoker (1 signifies the individual is a smoker), and lung Cancer (1 signifies the individual has lung Cancer). Each row contains the information for an individual, so that there are 7 individuals in the database.

Example

Choosing a structure and learning the table

a	s	c
1	1	1
1	0	0
0	1	1
0	1	0
1	1	1
0	0	0
1	0	1



To learn the table entries $p(c|a,s)$ we can do so by counting the number of c is in state 1 for each of the 4 parental states of a and s:

$$p(c=1|a=0,s=0)=0, \quad p(c=1|a=0,s=1)=0.5$$
$$p(c=1|a=1,s=0)=0.5 \quad p(c=1|a=1,s=1)=1$$

Similarly, based on counting, $p(a=1)=4/7$, and $p(s=1)=4/7$. These three CPTs then complete the full distribution specification.

Maximum likelihood learning

Maximum likelihood and KL divergence

Let q be the empirical distribution:

$$q(x) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[x = x^n\right]$$

Then

$$\begin{aligned}\mathrm{KL}(q|p(x|\theta)) &= \langle \log q(x) \rangle_{q(x)} - \langle \log p(x|\theta) \rangle_{q(x)} \\ &= -\frac{1}{N}\sum_{n=1}^{N} \log p(x^n|\theta) + \text{const.}\end{aligned}$$

Hence setting parameters of p that maximise the likelihood is equivalent to setting parameters of p that minimise the KL divergence between p and the empirical distribution.

Maximum likelihood learning

Maximum likelihood BN training and counting

A BN takes the form:

$$p(x) = \prod_{i=1}^{K} p(x_i|\mathrm{pa}\left(x_i\right))$$

For the BN $p(x)$, and empirical distribution $q(x)$ we have

$$\begin{aligned}
\mathrm{KL}(q|p) &= -\left\langle \sum_{i=1}^{K} \log p\left(x_i|\mathrm{pa}\left(x_i\right)\right) \right\rangle_{q(x)} + \text{const.} \\
&= -\sum_{i=1}^{K} \left\langle \log p\left(x_i|\mathrm{pa}\left(x_i\right)\right) \right\rangle_{q(x_i,\mathrm{pa}(x_i))} + \text{const.} \\
&= \sum_{i=1}^{K} \left[\left\langle \log q(x_i|\mathrm{pa}\left(x_i\right)) \right\rangle_{q(x_i,\mathrm{pa}(x_i))} - \left\langle \log p\left(x_i|\mathrm{pa}\left(x_i\right)\right) \right\rangle_{q(x_i,\mathrm{pa}(x_i))} \right] + \\
&= \sum_{i=1}^{K} \left\langle \mathrm{KL}(q(x_i|\mathrm{pa}\left(x_i\right))|p(x_i|\mathrm{pa}\left(x_i\right))) \right\rangle_{q(\mathrm{pa}(x_i))} + \text{const.}
\end{aligned}$$

Maximum likelihood learning

Maximum likelihood BN training and counting

The minimal Kullback-Leibler setting, and that which corresponds to Maximum Likelihood, is therefore

$$p(x_i|\mathrm{pa}\left(x_i\right)) = q(x_i|\mathrm{pa}\left(x_i\right))$$

In terms of the original data, this is

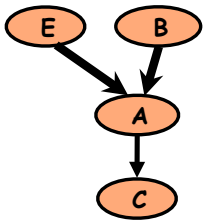
$$p(x_i = \mathsf{s}|\mathrm{pa}\left(x_i\right) = \mathsf{t}) \propto \sum_{n=1}^{N} \mathbb{I}\left[x_i^n = \mathsf{s}\right] \prod_{x_j \in \mathrm{pa}(x_i)} \mathbb{I}\left[x_j^n = \mathsf{t}^j\right]$$

The table entry $p(x_i|\mathrm{pa}\left(x_i\right))$ can be set by counting the number of times the state $\{x_i = \mathsf{s}, \mathrm{pa}\left(x_i\right) = \mathsf{t}\}$ occurs in the dataset (where t is a vector of parental states). The table is then given by the relative number of counts of being in state s compared to the other states s′, for fixed joint parental state t.

Maximum likelihood learning

ML learning, more generally, requires:

- Network structure specified
- Complete training data



Training data D has the form:

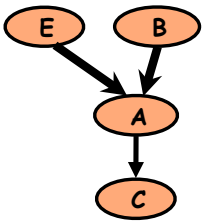
$$D = \begin{bmatrix} E[1] & B[1] & A[1] & C[1] \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ E[M] & B[M] & A[M] & C[M] \end{bmatrix}$$

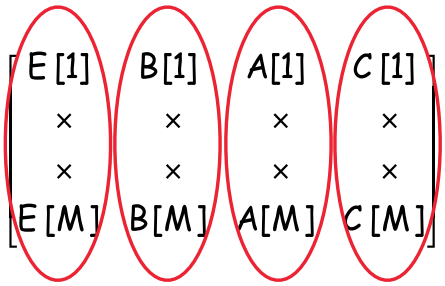
Maximum likelihood learning

Assume i.i.d. data samples D, we have

$$P(D|\Theta) = \prod_m P(E[m], B[m], A[m], C[m] : \Theta)$$

$$= \prod_m P(E[m] : \Theta) \prod_m P(B[m] : \Theta) \prod_m P(A[m] | B[m], E[m] : \Theta) \prod_m P(C[m] | A[m] : \Theta)$$





Maximum likelihood learning

Generalizing for any Bayesian network:

$$P(D|\Theta) = \prod_m P(x_1[m],\ldots,x_n[m] : \Theta)$$

$$= \prod_i \prod_m P(x_i[m] \mid Pa_i[m] : \Theta_i)$$

$$= \prod_i P(D|\Theta_i)$$

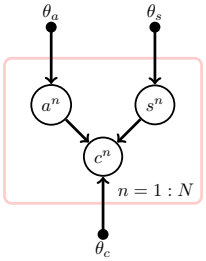
With complete data, parameter learning for a Bayesian network decomposes into independent estimation (learning) problems, one for each parameter Θ_i

Summary: Maximum likelihood learning

- assumes uniform priors
  - ok for large data sets
- either sets tables from sample statistics of the empirical distribution
- ... or chooses a parameterized family of models to describe the data and
  - write down likelihood of the data as a function of the parameters
  - write down derivative of the log likelihood w.r.t. each parameter
  - find the parameter values such that the derivatives are zero
  - may be hard/impossible; computational optimization techniques help

If only small data sets available or if we have additional knowledge, we need to place a prior on the tables ➔ Bayesian learning approach

Bayesian Belief Net training



We continue with the Asbestos, Smoking, Cancer scenario,

$$p(a, c, s) = p(c|a, s)p(a)p(s)$$

and a set of visible observations, $\mathcal{V} = \{(a^n, s^n, c^n), n = 1, \ldots, N\}$. With all variables binary we have parameters such as

$$p(a = 1|\theta_a) = \theta_a, \quad p(c = 1|a = 0, s = 1, \theta_c) = \theta_c^{0,1}$$

The parameters are

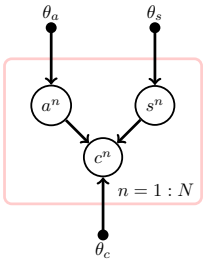
$$\theta_a, \theta_s, \underbrace{\theta_c^{0,0}, \theta_c^{0,1}, \theta_c^{1,0}, \theta_c^{1,1}}_{\theta_c}$$

In Bayesian learning of BNs, we need to specify a prior on the joint table entries. Since in general dealing with multi-dimensional continuous distributions is computationally problematic, it is useful to specify only uni-variate distributions in the prior. As we show below, this has a pleasing consequence that for i.i.d. data the posterior also factorises into uni-variate distributions.

Global parameter independence

A convenient assumption is that the prior factorises over parameters. For our Asbestos, Smoking, Cancer example, we assume

$$p(\theta_a, \theta_s, \theta_c) = p(\theta_a)p(\theta_s)p(\theta_c)$$



Assuming the data is i.i.d., we then have the joint model

$$p(\theta_a, \theta_s, \theta_c, \mathcal{V}) = p(\theta_a)p(\theta_s)p(\theta_c)\prod_n p(a^n|\theta_a)p(s^n|\theta_s)p(c^n|s^n, a^n, \theta_c)$$

Learning then corresponds to inference of

$$p(\theta_a, \theta_s, \theta_c|\mathcal{V}) = \frac{p(\mathcal{V}|\theta_a, \theta_s, \theta_c)p(\theta_a, \theta_s, \theta_c)}{p(\mathcal{V})} = \frac{p(\mathcal{V}|\theta_a, \theta_s, \theta_c)p(\theta_a)p(\theta_s)p(\theta_c)}{p(\mathcal{V})}$$

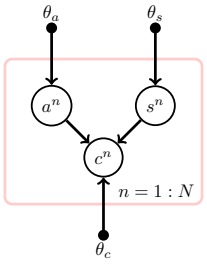
The posterior also factorises, since

$$\begin{aligned} p(\theta_a, \theta_s, \theta_c|\mathcal{V}) &\propto p(\theta_a, \theta_s, \theta_c, \mathcal{V}) \\ &= \left\{p(\theta_a)\prod_n p(a^n|\theta_a)\right\}\left\{p(\theta_s)\prod_n p(s^n|\theta_s)\right\}\left\{p(\theta_c)\prod_n p(c^n|s^n, a^n, \theta_c)\right\} \\ &\propto p(\theta_a|\mathcal{V}_a)p(\theta_s|\mathcal{V}_s)p(\theta_c|\mathcal{V}_c) \end{aligned}$$

Local parameter independence

If we further assume that the prior for the table factorises over all states a, c:

$$p(\theta_c) = p(\theta_c^{0,0})p(\theta_c^{1,0})p(\theta_c^{0,1})p(\theta_c^{1,1})$$



then the posterior

$$\begin{aligned} p(\theta_c|\mathcal{V}_c) &\propto p(\mathcal{V}_c|\theta_c)p(\theta_c^{0,0})p(\theta_c^{1,0})p(\theta_c^{0,1})p(\theta_c^{1,1}) \\ &= \underbrace{\left[\theta_c^{0,0}\right]^{\sharp(a=0,s=0)} p(\theta_c^{0,0})}_{\propto p(\theta_c^{0,0}|\mathcal{V}_c)} \underbrace{\left[\theta_c^{0,1}\right]^{\sharp(a=0,s=1)} p(\theta_c^{0,1})}_{\propto p(\theta_c^{0,1}|\mathcal{V}_c)} \\ &\times \underbrace{\left[\theta_c^{1,0}\right]^{\sharp(a=1,s=0)} p(\theta_c^{1,0})}_{\propto p(\theta_c^{1,0}|\mathcal{V}_c)} \underbrace{\left[\theta_c^{1,1}\right]^{\sharp(a=1,s=1)} p(\theta_c^{1,1})}_{\propto p(\theta_c^{1,1}|\mathcal{V}_c)} \end{aligned}$$

so that the posterior also factorises over the parental states of the local conditional table.

Using a Beta prior

$$p(\theta_a) = B\left(\theta_a|\alpha_a, \beta_a\right) = \frac{1}{B(\alpha_a, \beta_a)} \theta_a^{\alpha_a - 1} \left(1 - \theta_a\right)^{\beta_a - 1}$$

for which the posterior is also a Beta distribution:

$$p(\theta_a|\mathcal{V}_a) = B\left(\theta_a|\alpha_a + \sharp\left(a = 1\right), \beta_a + \sharp\left(a = 0\right)\right)$$

The marginal table is given by

$$p(a = 1|\mathcal{V}_a) = \int_{\theta_a} p(\theta_a|\mathcal{V}_a)\theta_a = \frac{\alpha_a + \sharp\left(a = 1\right)}{\alpha_a + \sharp\left(a = 1\right) + \beta_a + \sharp\left(a = 0\right)}$$

Corresponds in this case to adding 'pseudo counts' to the data.

hyperparameters

The prior parameters α_a, β_a are called hyperparameters. If one had no preference, one would set $\alpha_a = \beta_b = 1$.

Summary: Learning Parameters

Estimation relies on sufficient statistics

Maximum-likelihood (estimation) (ML/MLE)

- standard (non-bayesian) statistical learning
- useful for large data sets, where priors get irrelevant

Bayesian parameter learning

- Include prior probabilities, useful when data sets smaller
- Prediction is standard Bayesian inference

MLE vs. Bayesian learning

- Both are asymptotically equivalent and consistent
- Both can be implemented in an on-line manner by accumulating sufficient statistics

Outlook

Next week:

Learning with missing data (hidden variables)

- Expectation Maximization

Learning network structure

- PC (local search)
- scoring (global search)