Graphical Models: Learning parameters and structure

Zoubin Ghahramani

zoubin@eng.cam.ac.uk

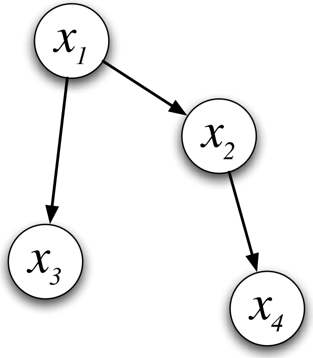
http://learning.eng.cam.ac.uk/zoubin/

Department of Engineering
University of Cambridge, UK

Machine Learning Department
Carnegie Mellon University, USA

2010

Learning parameters



$$p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2)$$

θ_2

	x_2		
x_1	0.2	0.3	0.5
	0.1	0.6	0.3

Assume each variable x_i is discrete and can take on K_i values.

The parameters of this model can be represented as 4 tables: θ_1 has K_1 entries, θ_2 has $K_1 \times K_2$ entries, etc.

These are called **conditional probability tables** (CPTs) with the following semantics:

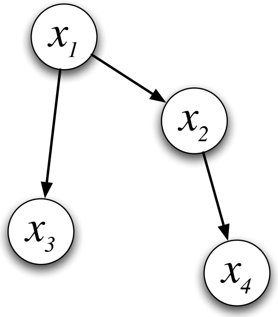
$$p(x_1 = k) = \theta_{1,k} \qquad p(x_2 = k'|x_1 = k) = \theta_{2,k,k'}$$

If node i has M parents, θ_i can be represented either as an $M+1$ dimensional table, or as a 2-dimensional table with $\left(\prod_{j \in \mathrm{pa}(i)} K_j\right) \times K_i$ entries by collapsing all the states of the parents of node i. Note that $\sum_{k'} \theta_{i,k,k'} = 1$.

Assume a data set $\mathcal{D} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$.

How do we learn θ from $\mathcal{D}$?

Learning parameters



Assume a data set $\mathcal{D} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$. How do we learn $\boldsymbol{\theta}$ from $\mathcal{D}$?

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(x_1|\theta_1)p(x_2|x_1,\theta_2)p(x_3|x_1,\theta_3)p(x_4|x_2,\theta_4)$$

Likelihood:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\mathbf{x}^{(n)}|\boldsymbol{\theta})$$

Log Likelihood:

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{i} \log p(x_i^{(n)}|x_{\mathrm{pa}(i)}^{(n)}, \theta_i)$$

This decomposes into sum of functions of θ_i. Each θ_i can be optimized separately:

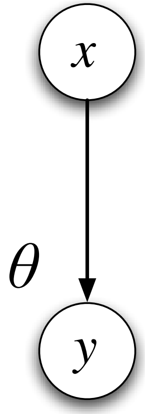
$$\hat{\theta}_{i,k,k'} = \frac{n_{i,k,k'}}{\sum_{k''} n_{i,k,k''}}$$

where $n_{i,k,k'}$ is the number of times in $\mathcal{D}$ where $x_i = k'$ and $x_{\mathrm{pa}(i)} = k$, where k represents a joint configuration of all the parents of i (i.e. takes on one of $\prod_{j \in \mathrm{pa}(i)} K_j$ values)

$$n_2 \quad \begin{array}{c|ccc} & & x_2 & \\ \hline x_1 & 2 & 3 & 0 \\ & 3 & 1 & 6 \end{array} \quad \Rightarrow \quad \theta_2 \quad \begin{array}{c|ccc} & & x_2 & \\ \hline x_1 & 0.4 & 0.6 & 0 \\ & 0.3 & 0.1 & 0.6 \end{array}$$

ML solution: Simply calculate frequencies!

Deriving the Maximum Likelihood Estimate



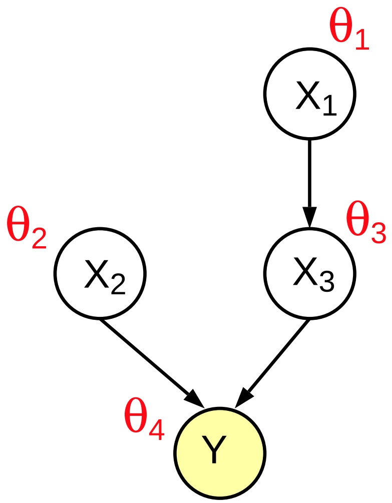
$$p(y|x,\theta) = \prod_{k,\ell} \theta_{k,\ell}^{\delta(x,k)\delta(y,\ell)}$$

Dataset $\mathcal{D} = \{(x^{(n)}, y^{(n)}) : n = 1 \ldots, N\}$

$$\begin{aligned}
\mathcal{L}(\theta) &= \log \prod_n p(y^{(n)}|x^{(n)},\theta) \\
&= \log \prod_n \prod_{k,\ell} \theta_{k,\ell}^{\delta(x^{(n)},k)\delta(y^{(n)},\ell)} \\
&= \sum_{n,k,\ell} \delta(x^{(n)},k)\delta(y^{(n)},\ell) \log \theta_{k,\ell} \\
&= \sum_{k,\ell} \left(\sum_n \delta(x^{(n)},k)\delta(y^{(n)},\ell) \right) \log \theta_{k,\ell} \;\; = \sum_{k,\ell} n_{k,\ell} \log \theta_{k,\ell}
\end{aligned}$$

Maximize $\mathcal{L}(\theta)$ w.r.t. θ subject to $\sum_\ell \theta_{k,\ell} = 1$ for all k.

Maximum Likelihood Learning with Hidden Variables

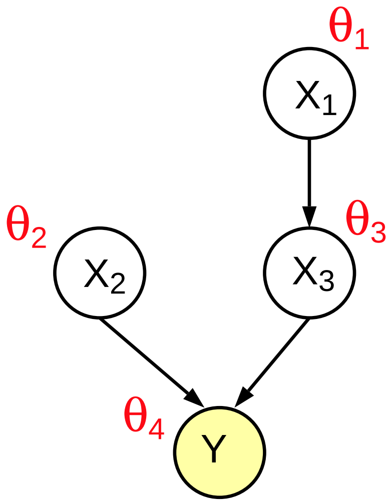


Assume a model parameterised by θ with observable variables Y and hidden variables X

Goal: maximize parameter log likelihood given observed data.

$$\mathcal{L}(\theta) = \log p(Y|\theta) = \log \sum_X p(Y, X|\theta)$$

Maximum Likelihood Learning with Hidden Variables: The EM Algorithm



Goal: maximise parameter log likelihood given observables.

$$\mathcal{L}(\theta) = \log p(Y|\theta) = \log \sum_X p(Y, X|\theta)$$

The Expectation Maximization (EM) algorithm (intuition):

Iterate between applying the following two steps:

- **The E step:** fill-in the hidden/missing variables
- **The M step:** apply complete data learning to filled-in data.

Maximum Likelihood Learning with Hidden Variables: The EM Algorithm

Goal: maximise parameter log likelihood given observables.

$$\mathcal{L}(\theta) = \log p(Y|\theta) = \log \sum_X p(Y, X|\theta)$$

The EM algorithm (derivation):

$$\mathcal{L}(\theta) \quad = \quad \log \sum_X q(X) \frac{p(Y, X|\theta)}{q(X)} \geq \sum_X q(X) \log \frac{p(Y, X|\theta)}{q(X)} = \mathcal{F}(q(X), \theta)$$

- **The E step:** maximize $\mathcal{F}(q(X), \theta^{[t]})$ wrt $q(X)$ holding $\theta^{[t]}$ fixed:

$$q(X) = p(X|Y, \theta^{[t]})$$

- **The M step:** maximize $\mathcal{F}(q(X), \theta)$ wrt θ holding $q(X)$ fixed:

$$\theta^{[t+1]} \leftarrow \mathrm{argmax}_\theta \sum_X q(X) \log p(Y, X|\theta)$$

The E-step requires solving the *inference* problem, finding the distribution over the hidden variables $p(X|Y, \theta^{[t]})$ given the current model parameters. This can be done using **belief propagation** or the **junction tree algorithm**.

Maximum Likelihood Learning without and with Hidden Variables

ML Learning with Complete Data (No Hidden Variables)

Log likelihood decomposes into sum of functions of θ_i. Each θ_i can be optimized separately:

$$\hat{\theta}_{ijk} \leftarrow \frac{n_{ijk}}{\sum_{k'} n_{ijk'}}$$

where n_{ijk} is the number of times in $\mathcal{D}$ where $x_i = k$ and $x_{\mathrm{pa}(i)} = j$.

Maximum likelihood solution: Simply calculate frequencies!

ML Learning with Incomplete Data (i.e. with Hidden Variables)

Iterative EM algorithm

E step: compute expected counts given previous settings of parameters $E[n_{ijk}|\mathcal{D}, \boldsymbol{\theta}^{[t]}]$.

M step: re-estimate parameters using these expected counts

$$\theta_{ijk}^{[t+1]} \leftarrow \frac{E[n_{ijk}|\mathcal{D}, \boldsymbol{\theta}^{[t]}]}{\sum_{k'} E[n_{ijk'}|\mathcal{D}, \boldsymbol{\theta}^{[t]}]}$$

Bayesian Learning

Apply the basic rules of probability to learning from data.

Data set: $\mathcal{D} = \{x_1, \dots, x_n\}$ Models: m, m' etc. Model parameters: θ

Prior probability of models: $P(m)$, $P(m')$ etc.
Prior probabilities of model parameters: $P(\theta|m)$
Model of data given parameters (likelihood model): $P(x|\theta, m)$

If the data are independently and identically distributed then:

$$P(\mathcal{D}|\theta, m) = \prod_{i=1}^{n} P(x_i|\theta, m)$$

Posterior probability of model parameters:

$$P(\theta|\mathcal{D}, m) = \frac{P(\mathcal{D}|\theta, m)P(\theta|m)}{P(\mathcal{D}|m)}$$

Posterior probability of models:

$$P(m|\mathcal{D}) = \frac{P(m)P(\mathcal{D}|m)}{P(\mathcal{D})}$$

Bayesian Occam's Razor and Model Comparison

Compare model classes, e.g. m and m', using posterior probabilities given $\mathcal{D}$:

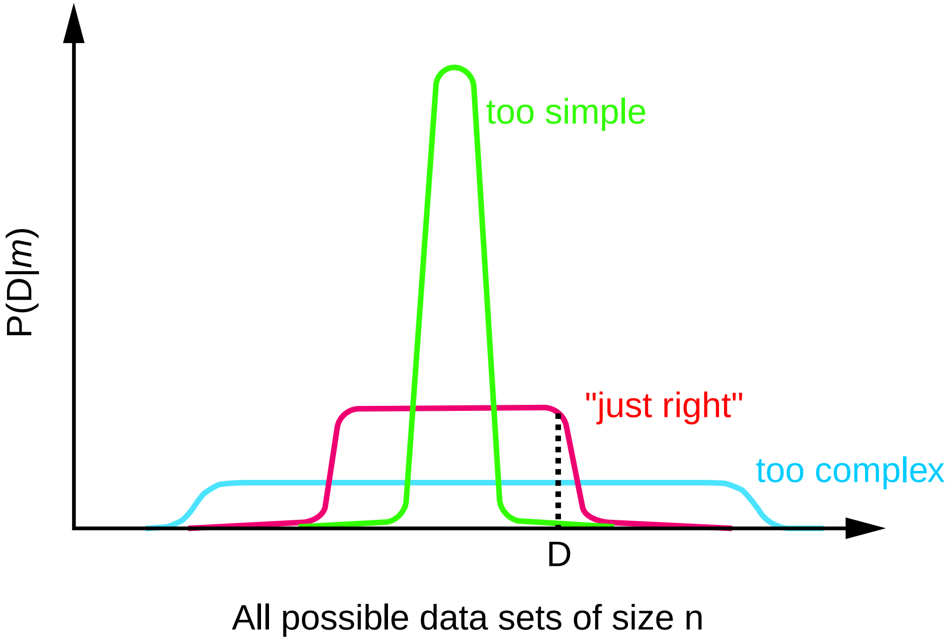
$$p(m|\mathcal{D}) = \frac{p(\mathcal{D}|m)\, p(m)}{p(\mathcal{D})}, \qquad p(\mathcal{D}|m) = \int p(\mathcal{D}|\boldsymbol{\theta}, m)\, p(\boldsymbol{\theta}|m)\, d\boldsymbol{\theta}$$

Interpretations of the Marginal Likelihood ("model evidence"):

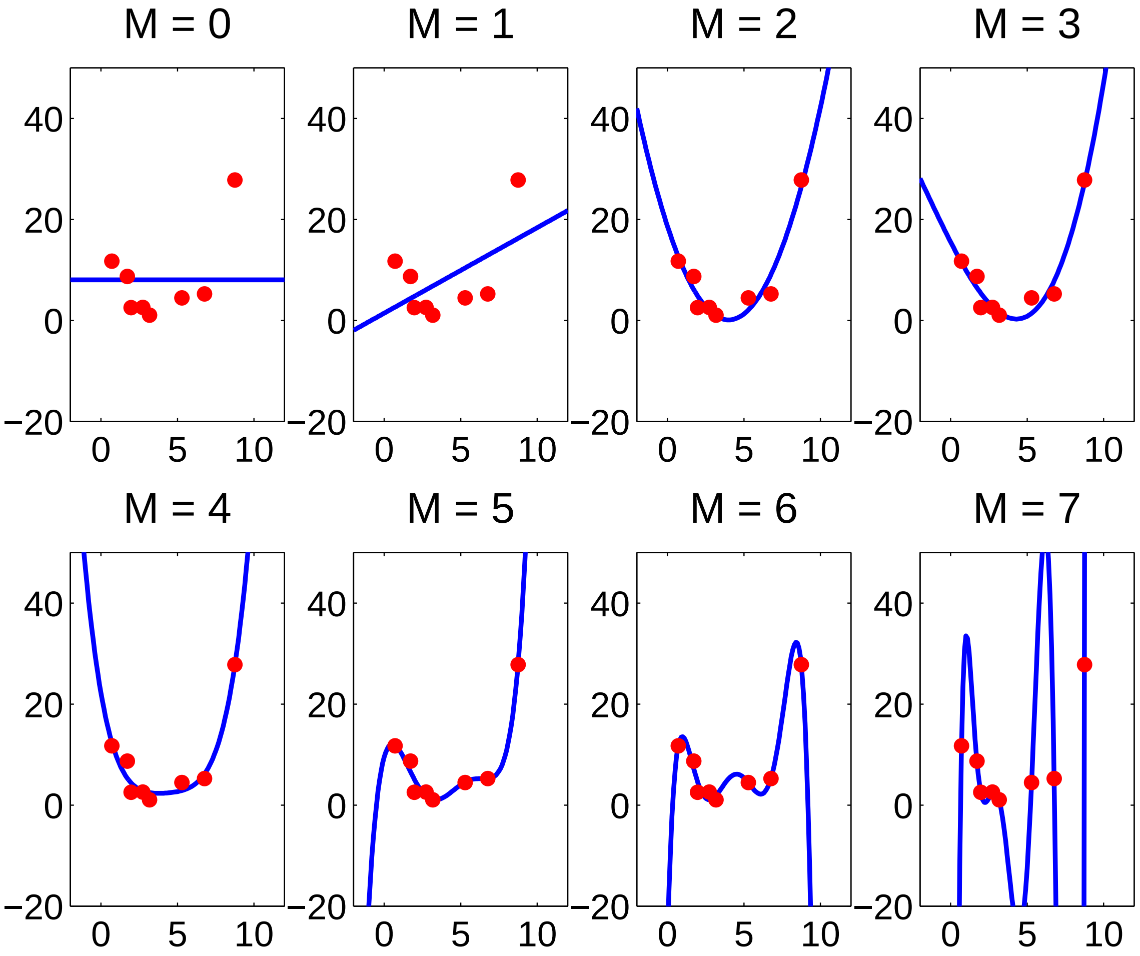
- The probability that *randomly selected* parameters from the prior would generate $\mathcal{D}$.
- Probability of the data under the model, *averaging* over all possible parameter values.
- $\log_2 \left(\frac{1}{p(\mathcal{D}|m)} \right)$ is the number of *bits of surprise* at observing data $\mathcal{D}$ under model m.

Model classes that are too simple are unlikely to generate the data set.

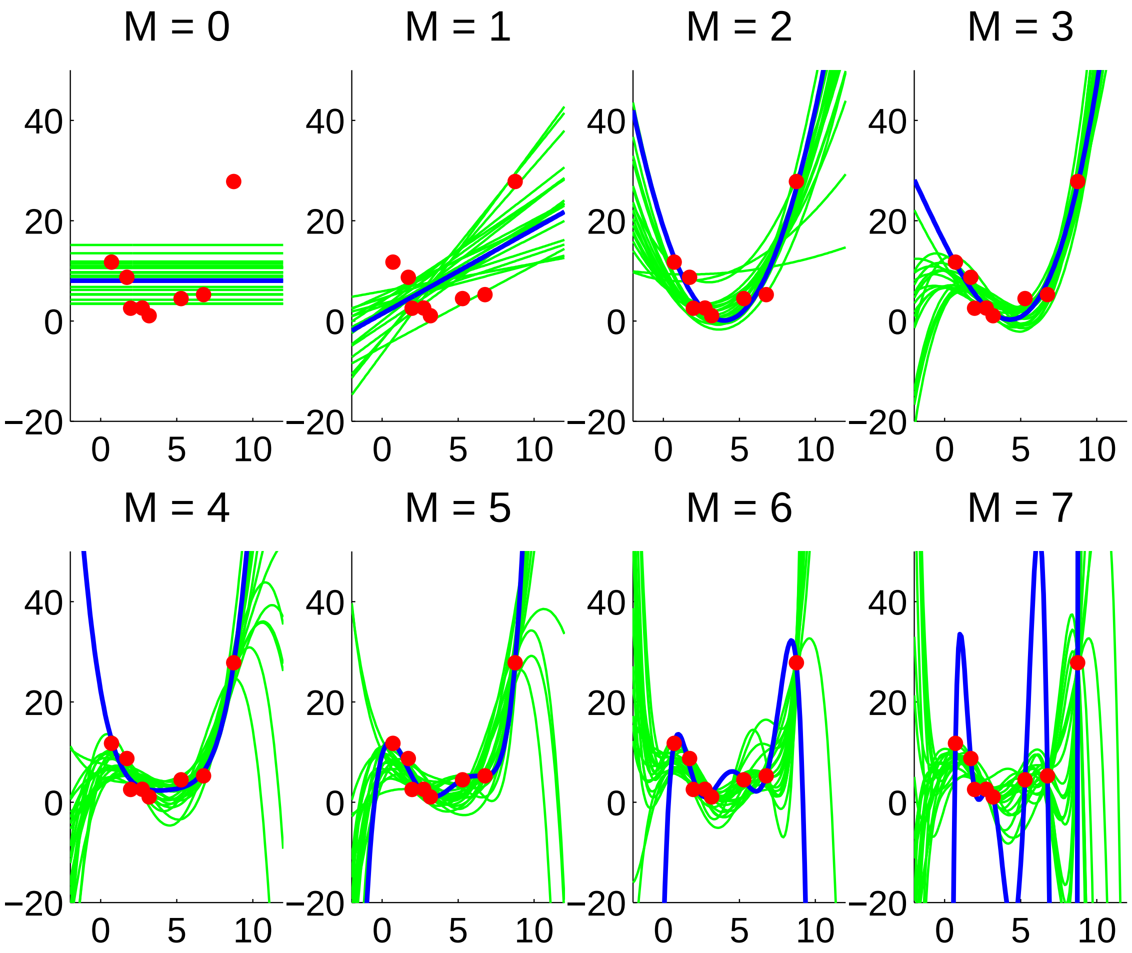
Model classes that are too complex can generate many possible data sets, so again, they are unlikely to generate that particular data set at random.

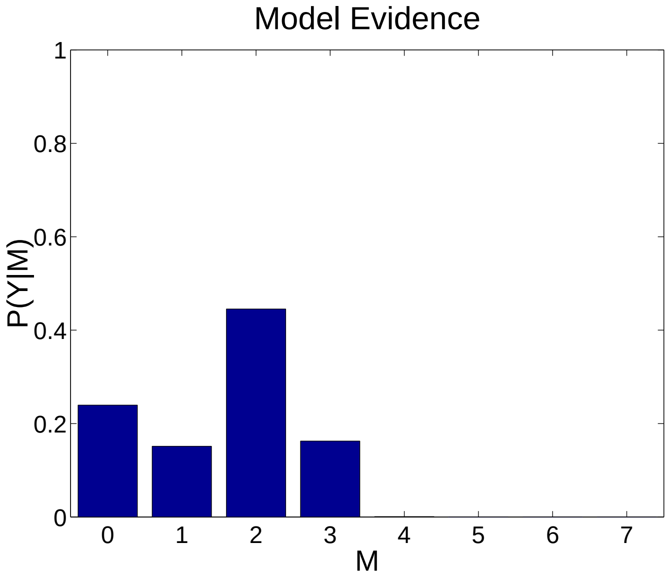


Model structure and overfitting:
A simple example: polynomial regression



Bayesian Model Comparison: Occam's Razor at Work



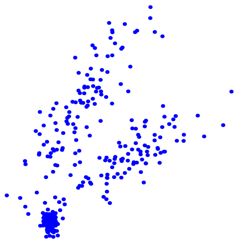


For example, for quadratic polynomials ($m = 2$): $y = a_0 + a_1 x + a_2 x^2 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and parameters $\boldsymbol{\theta} = (a_0\ a_1\ a_2\ \sigma)$

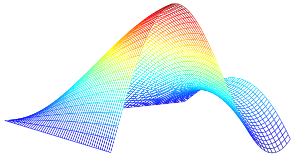
`demo: polybayes`

Learning Model Structure

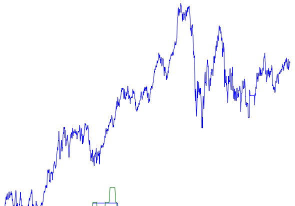
How many clusters in the data?



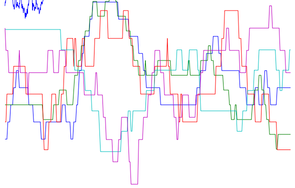
What is the intrinsic dimensionality of the data?



Is this input relevant to predicting that output?



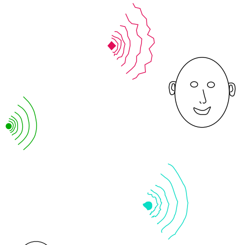
What is the order of a dynamical system?



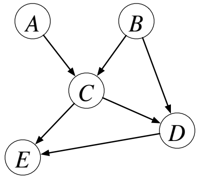
How many states in a hidden Markov model?

SVYDAAAQLTADVKKDLRDSWKVIGSDKKGNGVALMTTY

How many auditory sources in the input?



Which graph structure best models the data?



`demo: run_simple`

Bayesian parameter learning with no hidden variables

Let n_{ijk} be the number of times $(x_i^{(n)} = k$ and $x_{\mathrm{pa}(i)}^{(n)} = j)$ in $\mathcal{D}$.
For each i and j, $\boldsymbol{\theta}_{ij\cdot}$ is a probability vector of length $K_i \times 1$.

Since x_i is a discrete variable with probabilities given by $\boldsymbol{\theta}_{i,j,\cdot}$, the likelihood is:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_n \prod_i p(x_i^{(n)}|x_{\mathrm{pa}(i)}^{(n)}, \boldsymbol{\theta}) = \prod_i \prod_j \prod_k \theta_{ijk}^{n_{ijk}}$$

If we choose a prior on $\boldsymbol{\theta}$ of the form:

$$p(\boldsymbol{\theta}) = c \prod_i \prod_j \prod_k \theta_{ijk}^{\alpha_{ijk}-1}$$

where c is a normalization constant, and $\sum_k \theta_{ijk} = 1 \; \forall i, j$, then the posterior distribution also has the same form:

$$p(\boldsymbol{\theta}|\mathcal{D}) = c' \prod_i \prod_j \prod_k \theta_{ijk}^{\tilde{\alpha}_{ijk}-1}$$

where $\tilde{\alpha}_{ijk} = \alpha_{ijk} + n_{ijk}$.

This distribution is called the Dirichlet distribution.

Dirichlet Distribution

The Dirichlet distribution is a distribution over the K-dim probability simplex.

Let $\boldsymbol{\theta}$ be a K-dimensional vector s.t. $\forall j : \theta_j \geq 0$ and $\sum_{j=1}^{K} \theta_j = 1$

$$p(\boldsymbol{\theta}|\alpha) = \text{Dir}(\alpha_1, \ldots, \alpha_K) \overset{\text{def}}{=} \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \Gamma(\alpha_j)} \prod_{j=1}^{K} \theta_j^{\alpha_j - 1}$$

where the first term is a normalization constant[1] and $E(\theta_j) = \alpha_j / (\sum_k \alpha_k)$

The Dirichlet is conjugate to the multinomial distribution. Let

$$x|\boldsymbol{\theta} \sim \text{Multinomial}(\cdot|\boldsymbol{\theta})$$

That is, $p(x = j|\boldsymbol{\theta}) = \theta_j$. Then the posterior is also Dirichlet:

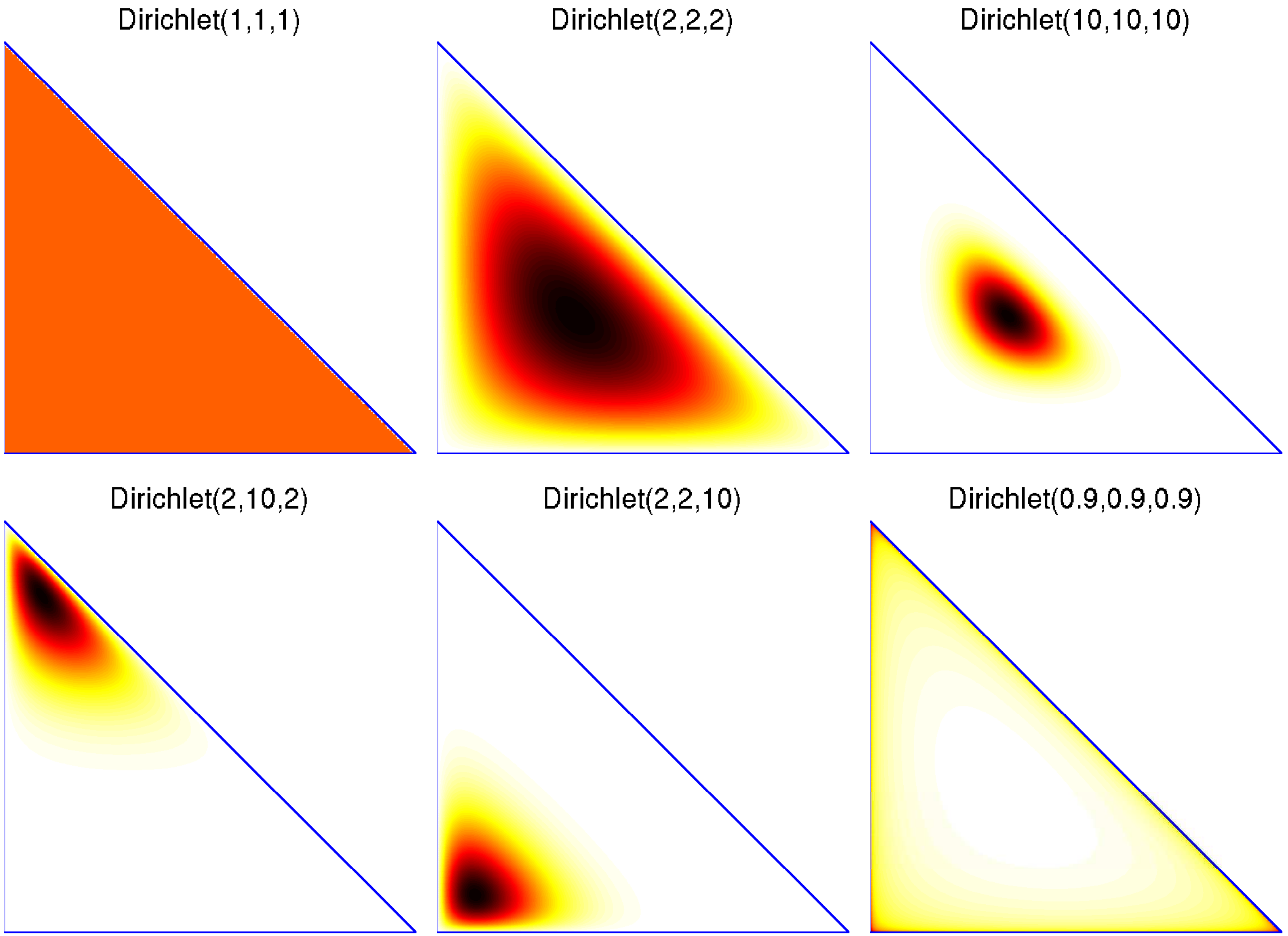
$$\boldsymbol{p(\theta|x = j, \alpha) = \frac{p(x = j|\theta)p(\theta|\alpha)}{p(x = j|\alpha)}} = \text{Dir}(\tilde{\boldsymbol{\alpha}})$$

where $\tilde{\alpha}_j = \alpha_j + 1$, and $\forall \ell \neq j : \tilde{\alpha}_\ell = \alpha_\ell$

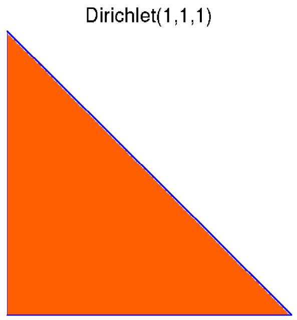
[1] $\Gamma(x) = (x-1)\Gamma(x-1) = \int_0^\infty t^{x-1} e^{-t} dt$. For integer n, $\Gamma(n) = (n-1)!$

Dirichlet Distributions

Examples of Dirichlet distributions over $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ which can be plotted in 2D since $\theta_3 = 1 - \theta_1 - \theta_2$:



Example



Assume $\alpha_{ijk} = 1 \ \forall i, j, k$.

This corresponds to a **uniform** prior distribution over parameters θ. This is not a very strong/dogmatic prior, since any parameter setting is assumed a priori possible.

After observed data $\mathcal{D}$, what are the parameter posterior distributions?

$$p(\theta_{ij\cdot}|\mathcal{D}) = \mathrm{Dir}(n_{ij\cdot} + 1)$$

This distribution predicts, for future data:

$$p(x_i = k | x_{\mathrm{pa}(i)} = j, \mathcal{D}) = \frac{n_{ijk} + 1}{\sum_{k'}(n_{ijk'} + 1)}$$

Adding 1 to each of the counts is a form of smoothing called “Laplace’s Rule”.

Bayesian parameter learning with hidden variables

Notation: let $\mathcal{D}$ be the observed data set, $\mathcal{X}$ be hidden variables, and $\boldsymbol{\theta}$ be model parameters. Assume discrete variables and Dirichlet priors on $\boldsymbol{\theta}$

Goal: to infer $p(\boldsymbol{\theta}|\mathcal{D}) = \sum_{\mathcal{X}} p(\mathcal{X}, \boldsymbol{\theta}|\mathcal{D})$

Problem: since (a)

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

and (b) for every way of filling in the missing data, $p(\boldsymbol{\theta}|\mathcal{X}, \mathcal{D})$ is a Dirichlet distribution, and (c) there are exponentially many ways of filling in $\mathcal{X}$, it follows that $p(\boldsymbol{\theta}|\mathcal{D})$ is a mixture of Dirichlets with exponentially many terms!

Solutions:

- Find a single best ("Viterbi") completion of $\mathcal{X}$ (Stolcke and Omohundro, 1993)
- Markov chain Monte Carlo methods
- Variational Bayesian (VB) methods (Beal and Ghahramani, 2006)

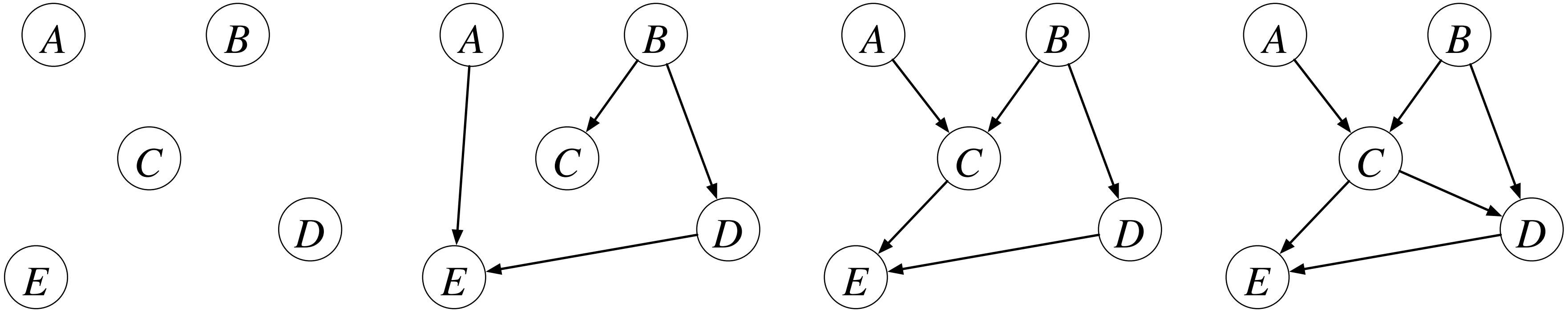
Summary of parameter learning

	Complete (fully observed) data	Incomplete (hidden /missing) data
ML	calculate frequencies	EM
Bayesian	update Dirichlet distributions	MCMC / Viterbi / VB

- For complete data Bayesian learning is not more costly than ML
- For incomplete data VB $\approx$ EM time complexity
- Other parameter priors are possible but Dirichlet is pretty flexible and intuitive.
- For non-discrete data, similar ideas but generally harder inference and learning.

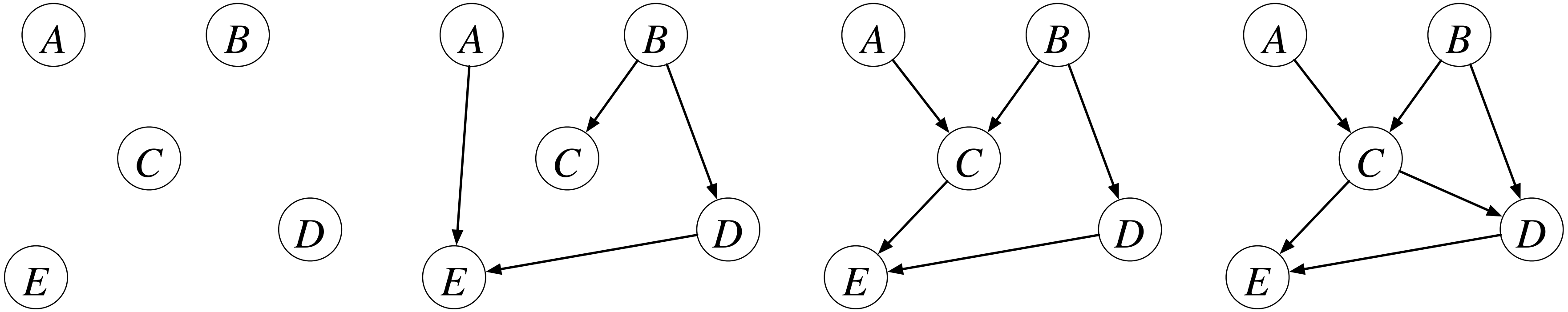
Structure learning

Given a data set of observations of (A, B, C, D, E) can we learn the structure of the graphical model?



Let m denote the graph structure = the set of edges.

Structure learning



Constraint-Based Learning: Use statistical tests of marginal and conditional independence. Find the set of DAGs whose d-separation relations match the results of conditional independence tests.

Score-Based Learning: Use a global score such as the BIC score or Bayesian marginal likelihood. Find the structures that maximize this score.

Score-based structure learning for complete data

Consider a graphical model with structure m, discrete observed data $\mathcal{D}$, and parameters θ. Assume Dirichlet priors.

The Bayesian marginal likelihood score is easy to compute:

$$\text{score}(m) = \log p(\mathcal{D}|m) = \log \int p(\mathcal{D}|\theta, m) p(\theta|m) d\theta$$

$$\text{score}(m) = \sum_i \sum_j \left[\log \Gamma(\sum_k \alpha_{ijk}) - \sum_k \log \Gamma(\alpha_{ijk}) - \log \Gamma(\sum_k \tilde{\alpha}_{ijk}) + \sum_k \log \Gamma(\tilde{\alpha}_{ijk}) \right]$$

where $\tilde{\alpha}_{ijk} = \alpha_{ijk} + n_{ijk}$. **Note that the score decomposes over i.**

One can incorporate structure prior information $p(m)$ as well:

$$\text{score}(m) = \log p(\mathcal{D}|m) + \log p(m)$$

Greedy search algorithm: Start with m. Consider modifications $m \to m'$ (edge deletions, additions, reversals). Accept m' if score(m') > score(m). Repeat.

Bayesian inference of model structure: Run MCMC on m.

Bayesian Structural EM for *in*complete data

Consider a graph with structure m, observed data $\mathcal{D}$, hidden variables $\mathcal{X}$ and parameters θ

The Bayesian score is generally intractable to compute:

$$\text{score}(m) = p(\mathcal{D}|m) = \int \sum_{\mathcal{X}} p(\mathcal{X}, \theta, \mathcal{D}|m) d\theta$$

Bayesian Structure EM (Friedman, 1998):

1. compute MAP parameters $\hat{\theta}$ for current model m using EM

2. find hidden variable distribution $p(\mathcal{X}|\mathcal{D}, \hat{\theta})$

3. for a small set of candidate structures compute or approximate

$$\text{score}(m') = \sum_{\mathcal{X}} p(\mathcal{X}|\mathcal{D}, \hat{\theta}) \log p(\mathcal{D}, \mathcal{X}|m')$$

4. $m \leftarrow m'$ with highest score

Alternatively we can use variational Bayesian learning (Beal and Ghahramani, 2006).

Directed Graphical Models and Causality

Causal relationships are a fundamental component of cognition and scientific discovery.

Even though the independence relations are identical, there is a **causal** difference between

- “smoking” → “yellow teeth”
- “yellow teeth” → “smoking”

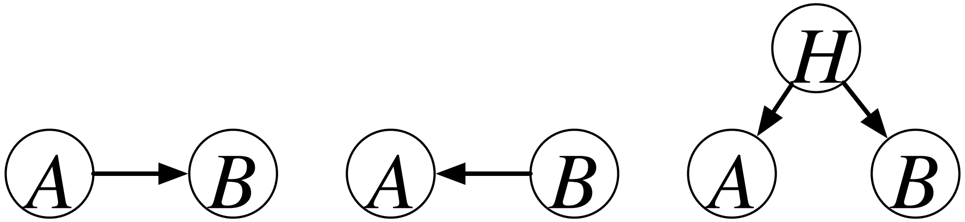
Key idea: interventions and the do-calculus:

$$p(S|Y=y) \neq p(S|\text{do}(Y=y))$$

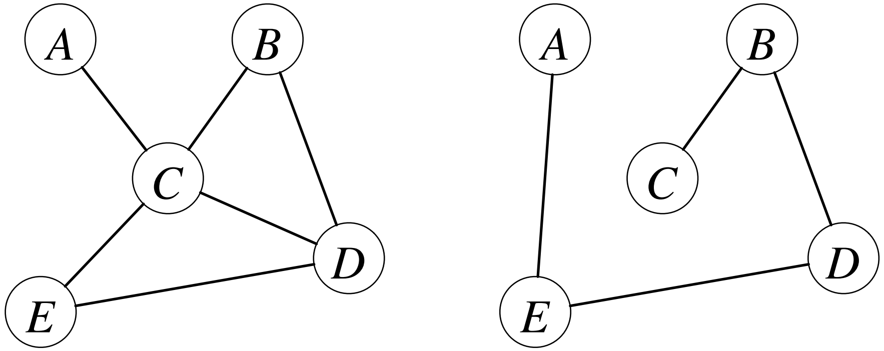
$$p(Y|S=s) = p(Y|\text{do}(S=s))$$

Causal relationships are robust to interventions on the parents.

The **key difficulty** in learning causal relationships from observational data is the presence of **hidden common causes**:



Learning parameters and structure in undirected graphs



$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_j g_j(\mathbf{x}_{C_j}; \boldsymbol{\theta}_j)$ where $Z(\boldsymbol{\theta}) = \sum_{\mathbf{x}} \prod_j g_j(\mathbf{x}_{C_j}; \boldsymbol{\theta}_j)$.

Problem: computing $Z(\boldsymbol{\theta})$ is computationally intractable for general (non-tree-structured) undirected models. Therefore, maximum-likelihood learning of parameters is generally intractable, Bayesian scoring of structures is intractable, etc.

Solutions:

- directly approximate $Z(\boldsymbol{\theta})$ and/or its derivatives (cf. Boltzmann machine learning; contrastive divergence; pseudo-likelihood)

- use approx inference methods (e.g. loopy belief propagation, bounding methods, EP).

See: (Murray and Ghahramani, 2004; Murray et al, 2006) for Bayesian learning in undirected models.

Summary

- Parameter learning in directed models:
  - complete and incomplete data;
  - ML and Bayesian methods
- Bayesian model comparison and Occam's Razor
- Structure learning in directed models: complete and incomplete data
- Causality
- Parameter and Structure learning in undirected models

Readings and References

- Beal, M.J. and Ghahramani, Z. (2006) Variational Bayesian learning of directed graphical models with hidden variables. *Bayesian Analysis* 1(4):793–832.
  http://learning.eng.cam.ac.uk/zoubin/papers/BeaGha06.pdf
- Friedman, N. (1998) The Bayesian structural EM algorithm. In *Uncertainty in Artificial Intelligence (UAI-1998)*. http://robotics.stanford.edu/ nir/Papers/Fr2.pdf
- Friedman, N. and Koller, D. (2003) Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning.* 50(1): 95–125.
  http://www.springerlink.com/index/NQ13817217667435.pdf
- Ghahramani, Z. (2004) Unsupervised Learning. In Bousquet, O., von Luxburg, U. and Raetsch, G. *Advanced Lectures in Machine Learning*. 72-112.
  http://learning.eng.cam.ac.uk/zoubin/papers/ul.pdf
- Heckerman, D. (1995) A tutorial on learning with Bayesian networks. In *Learning in Graphical Models.*
  http://research.microsoft.com/pubs/69588/tr-95-06.pdf
- Murray, I.A., Ghahramani, Z., and MacKay, D.J.C. (2006) MCMC for doubly-intractable distributions. In *Uncertainty in Artificial Intelligence (UAI-2006).*
  http://learning.eng.cam.ac.uk/zoubin/papers/doubly_intractable.pdf
- Murray, I.A. and Ghahramani, Z. (2004) Bayesian Learning in Undirected Graphical Models: Approximate MCMC algorithms. In *Uncertainty in Artificial Intelligence (UAI-2004).*
  http://learning.eng.cam.ac.uk/zoubin/papers/uai04murray.pdf
- Stolcke, A. and Omohundro, S. (1993) Hidden Markov model induction by Bayesian model merging. In *Advances in Neural Information Processing Systems (NIPS).*
  http://omohundro.files.wordpress.com/2009/03/stolcke_omohundro93_hmm_induction_bayesian_model_merging.pdf