Bayesian Methods for Machine Learning

Zoubin Ghahramani

Gatsby Computational Neuroscience Unit
University College London, UK

Center for Automated Learning and Discovery
Carnegie Mellon University, USA

zoubin@gatsby.ucl.ac.uk
http://www.gatsby.ucl.ac.uk

Chicago Machine Learning Summer School
May 2005

Plan

- Introduce Foundations
- The Intractability Problem
- Approximation Tools
- Advanced Topics
- Limitations and Discussion

Detailed Plan

- **Introduce Foundations**
  - Some canonical problems: classification, regression, density estimation, coin toss
  - Representing beliefs and the Cox axioms
  - The Dutch Book Theorem
  - Asymptotic Certainty and Consensus
  - Occam's Razor and Marginal Likelihoods
  - Choosing Priors
    - Objective Priors: Noninformative, Jeffreys, Reference
    - Subjective Priors
    - Hierarchical Priors
    - Empirical Priors
    - Conjugate Priors
- **The Intractability Problem**
- **Approximation Tools**
  - Laplace's Approximation
  - Bayesian Information Criterion (BIC)
  - Variational Approximations
  - Expectation Propagation
  - MCMC
  - Exact Sampling
- **Advanced Topics**
  - Feature Selection and ARD
  - Bayesian Discriminative Learning (BPM vs SVM)
  - From Parametric to Nonparametric Methods
    - Gaussian Processes
    - Dirichlet Process Mixtures
    - Other Non-parametric Bayesian Methods
  - Bayesian Decision Theory and Active Learning
  - Bayesian Semi-supervised Learning
- **Limitations and Discussion**
  - Reconciling Bayesian and Frequentist Views
  - Limitations and Criticisms of Bayesian Methods
  - Discussion

This is a modified and shortened version of my 2004 ICML tutorial.

Some Canonical Problems

- Weather Prediction
- Linear Classification
- Polynomial Regression
- Clustering with Gaussian Mixtures (Density Estimation)

Weather Prediction

Assume that the weather in London is independent and identically distributed across days. It can either rain (R) or be cloudy (C).

Data: $\mathcal{D} = (R\,C\,C\,R\,R\,C\,R\,\ldots)$

Parameters: $\theta \stackrel{\text{def}}{=}$ Probability of rain

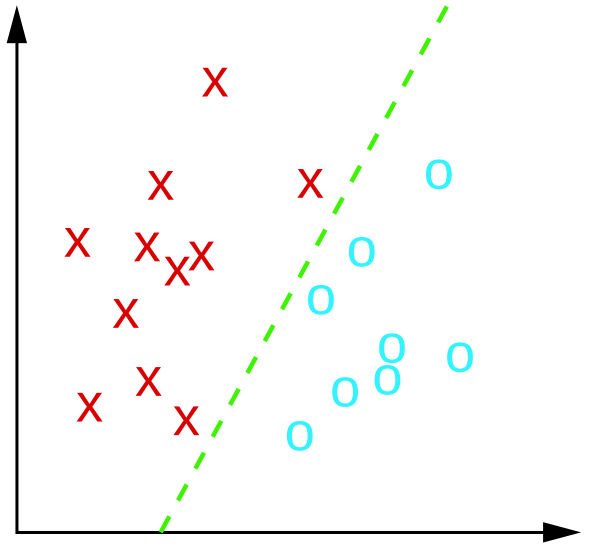
$$P(R|\theta) = \theta$$
$$P(C|\theta) = 1 - \theta$$

Goal: To infer θ from the data and predict future outcomes $P(R|\mathcal{D})$.

Linear Classification

Data: $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}$ for $n = 1, \ldots, N$ data points

$$\begin{aligned} \mathbf{x}^{(n)} &\in \Re^D \\ y^{(n)} &\in \{+1, -1\} \end{aligned}$$



Parameters: $\boldsymbol{\theta} \in \Re^{D+1}$

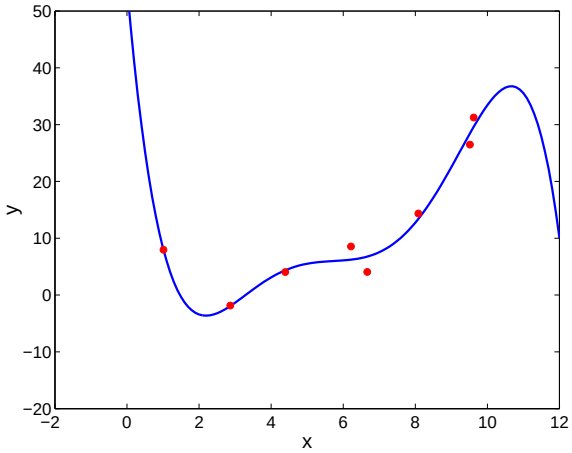
$$P(y^{(n)} = +1 | \boldsymbol{\theta}, \mathbf{x}^{(n)}) = \begin{cases} 1 & \text{if } \sum_{d=1}^{D} \theta_d \, x_d^{(n)} + \theta_0 \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Goal: To infer $\boldsymbol{\theta}$ from the data and to predict future labels $P(y|\mathcal{D}, \mathbf{x})$

Polynomial Regression

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

$$\begin{aligned} x^{(n)} &\in \Re \\ y^{(n)} &\in \Re \end{aligned}$$



Parameters: $\boldsymbol{\theta} = (a_0, \ldots, a_m, \sigma)$

Model:

$$y^{(n)} = a_0 + a_1 x^{(n)} + a_2 {x^{(n)}}^2 \ldots + a_m {x^{(n)}}^m + \epsilon$$

where

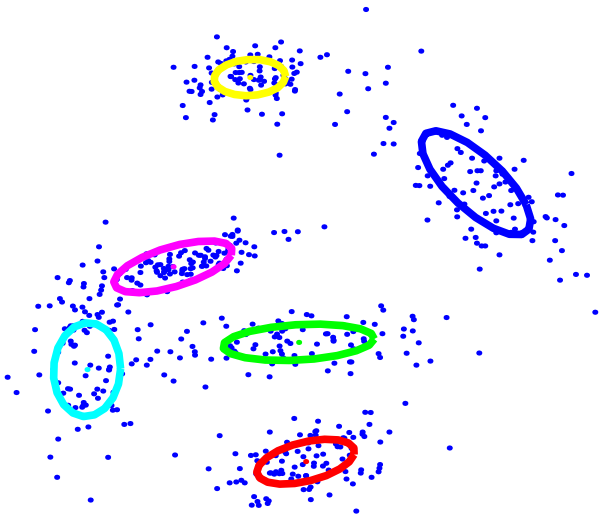
$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

Goal: To infer $\boldsymbol{\theta}$ from the data and to predict future outputs $P(y|\mathcal{D}, x, m)$

Clustering with Gaussian Mixtures (Density Estimation)

Data: $\mathcal{D} = \{\mathbf{x}^{(n)}\}$ for $n = 1, \ldots, N$

$$\mathbf{x}^{(n)} \in \Re^D$$



Parameters: $\boldsymbol{\theta} = \left((\mu^{(1)}, \Sigma^{(1)}) \ldots, (\mu^{(m)}, \Sigma^{(m)}), \boldsymbol{\pi}\right)$

Model:

$$\mathbf{x}^{(n)} \sim \sum_{i=1}^{m} \pi_i \, p_i(\mathbf{x}^{(n)})$$

where

$$p_i(\mathbf{x}^{(n)}) = \mathcal{N}(\mu^{(i)}, \Sigma^{(i)})$$

Goal: To infer $\boldsymbol{\theta}$ from the data and predict the density $p(\mathbf{x}|\mathcal{D}, m)$

Basic Rules of Probability

$P(x)$	probability of x
$P(x\|\theta)$	conditional probability of x given θ
$P(x,\theta)$	joint probability of x and θ

$$P(x,\theta) = P(x)P(\theta|x) = P(\theta)P(x|\theta)$$

Bayes Rule:

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}$$

Marginalization

$$P(x) = \int P(x,\theta)\, d\theta$$

Warning: I will not be obsessively careful in my use of p and P for probability density and probability distribution. Should be obvious from context.

Bayes Rule Applied to Machine Learning

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

$P(\mathcal{D}\|\theta)$	likelihood of θ
$P(\theta)$	prior probability of θ
$P(\theta\|\mathcal{D})$	posterior of θ given $\mathcal{D}$

Model Comparison:

$$\begin{aligned} P(m|\mathcal{D}) &= \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})} \\ P(\mathcal{D}|m) &= \int P(\mathcal{D}|\theta, m)P(\theta|m)\, d\theta \end{aligned}$$

Prediction:

$$\begin{aligned} P(x|\mathcal{D}, m) &= \int P(x|\theta, \mathcal{D}, m)P(\theta|\mathcal{D}, m)d\theta \\ P(x|\mathcal{D}, m) &= \int P(x|\theta)P(\theta|\mathcal{D}, m)d\theta \qquad \text{(for many models)} \end{aligned}$$

End of Tutorial

Questions

- Why be Bayesian?
- Where does the prior come from?
- How do we do these integrals?

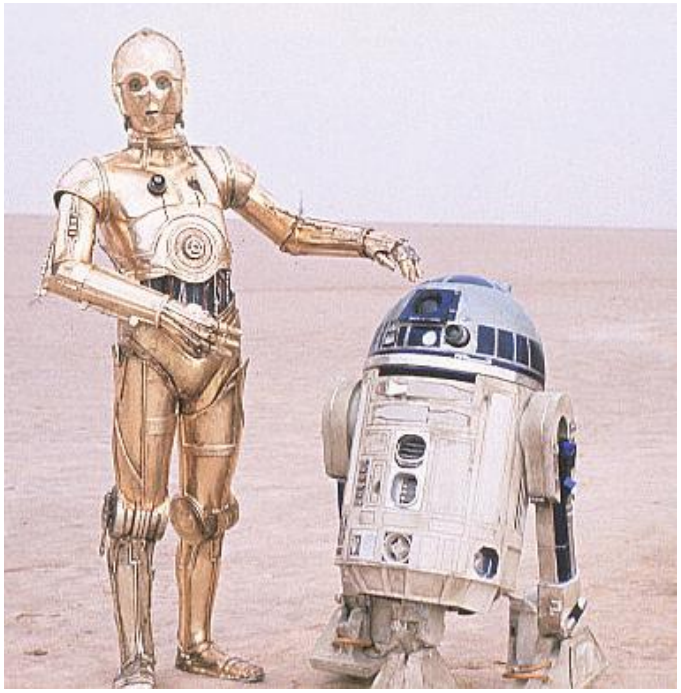
Representing Beliefs (Artificial Intelligence)

Consider a robot. In order to behave intelligently the robot should be able to represent beliefs about propositions in the world:

"my charging station is at location (x,y,z)"

"my rangefinder is malfunctioning"

"that stormtrooper is hostile"



We want to represent the **strength** of these beliefs numerically in the brain of the robot, and we want to know what rules ("calculus") we should use to manipulate those beliefs.

Representing Beliefs II

Let's use $b(x)$ to represent the strength of belief in (plausibility of) proposition x.

$0 \leq b(x) \leq 1$		
$b(x) = 0$	x	is definitely **not true**
$b(x) = 1$	x	is definitely **true**
$b(x\|y)$		strength of belief that x is true given that we know y is true

Cox Axioms (Desiderata):

- Strengths of belief (degrees of plausibility) are represented by real numbers
- Qualitative correspondence with common sense
- Consistency
  - If a conclusion can be reasoned in more than one way, then every way should lead to the same answer.
  - The robot always takes into account all relevant evidence.
  - Equivalent states of knowledge are represented by equivalent plausibility assignments.

Consequence: Belief functions (e.g. $b(x)$, $b(x|y)$, $b(x, y)$) must satisfy the rules of probability theory, including Bayes rule. (see Jaynes, *Probability Theory: The Logic of Science*)

The Dutch Book Theorem

Assume you are willing to accept bets with odds proportional to the strength of your beliefs. That is, $b(x) = 0.9$ implies that you will accept a bet:

$$\begin{cases} x & \text{is true} & \text{win} & \geq \$1 \\ x & \text{is false} & \text{lose} & \$9 \end{cases}$$

Then, unless your beliefs satisfy the rules of probability theory, including Bayes rule, there exists a set of simultaneous bets (called a "Dutch Book") which you are willing to accept, and for which **you are guaranteed to lose money, no matter what the outcome**.

The only way to guard against Dutch Books to to ensure that your beliefs are coherent: i.e. satisfy the rules of probability.

Asymptotic Certainty

Assume that data set $\mathcal{D}_n$, consisting of n data points, was generated from some true finite dimensional model with parameters θ^*, then under some regularity conditions, as long as $p(\theta^*) > 0$

$$\lim_{n\to\infty} p(\theta|\mathcal{D}_n) = \delta(\theta - \theta^*)$$

In the **unrealizable case**, where data was generated from some $p^*(x)$ which cannot be modelled by any θ, then the posterior will converge to

$$\lim_{n\to\infty} p(\theta|\mathcal{D}_n) = \delta(\theta - \hat{\theta})$$

where $\hat{\theta}$ minimizes $\mathrm{KL}(p^*(x), p(x|\theta))$:

$$\hat{\theta} = \operatorname*{argmin}_\theta \int p^*(x) \log \frac{p^*(x)}{p(x|\theta)}\, dx = \operatorname*{argmax}_\theta \int p^*(x) \log p(x|\theta)\, dx$$

Warning: careful with the regularity conditions, these are just sketches of the theoretical results

Asymptotic Consensus

Consider two Bayesians with *different priors*, $p_1(\theta)$ and $p_2(\theta)$,
who observe the *same data* $\mathcal{D}$.

Assume both Bayesians agree on the set of possible and impossible values of θ:

$$\{\theta : p_1(\theta) > 0\} = \{\theta : p_2(\theta) > 0\}$$

Then, in the limit of $n \to \infty$, the posteriors, $p_1(\theta|\mathcal{D}_n)$ and $p_2(\theta|\mathcal{D}_n)$ will converge
(in uniform distance between distibutions $\rho(P_1, P_2) = \sup_E |P_1(E) - P_2(E)|$)

coin toss demo: `bayescoin`

Bayesian Occam's Razor and Model Comparison

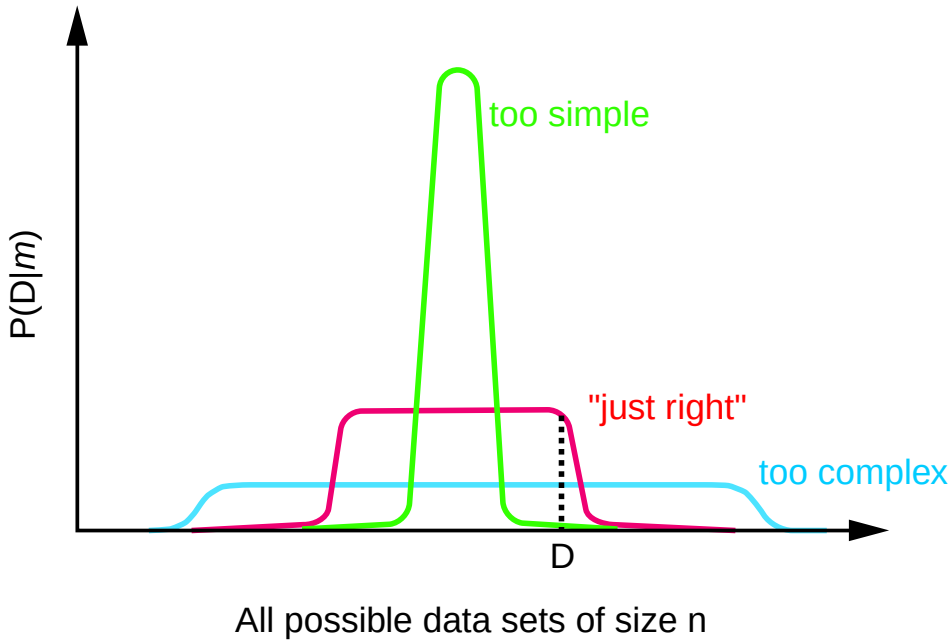
Compare model classes, e.g. m and m', using posterior probabilities given $\mathcal{D}$:

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

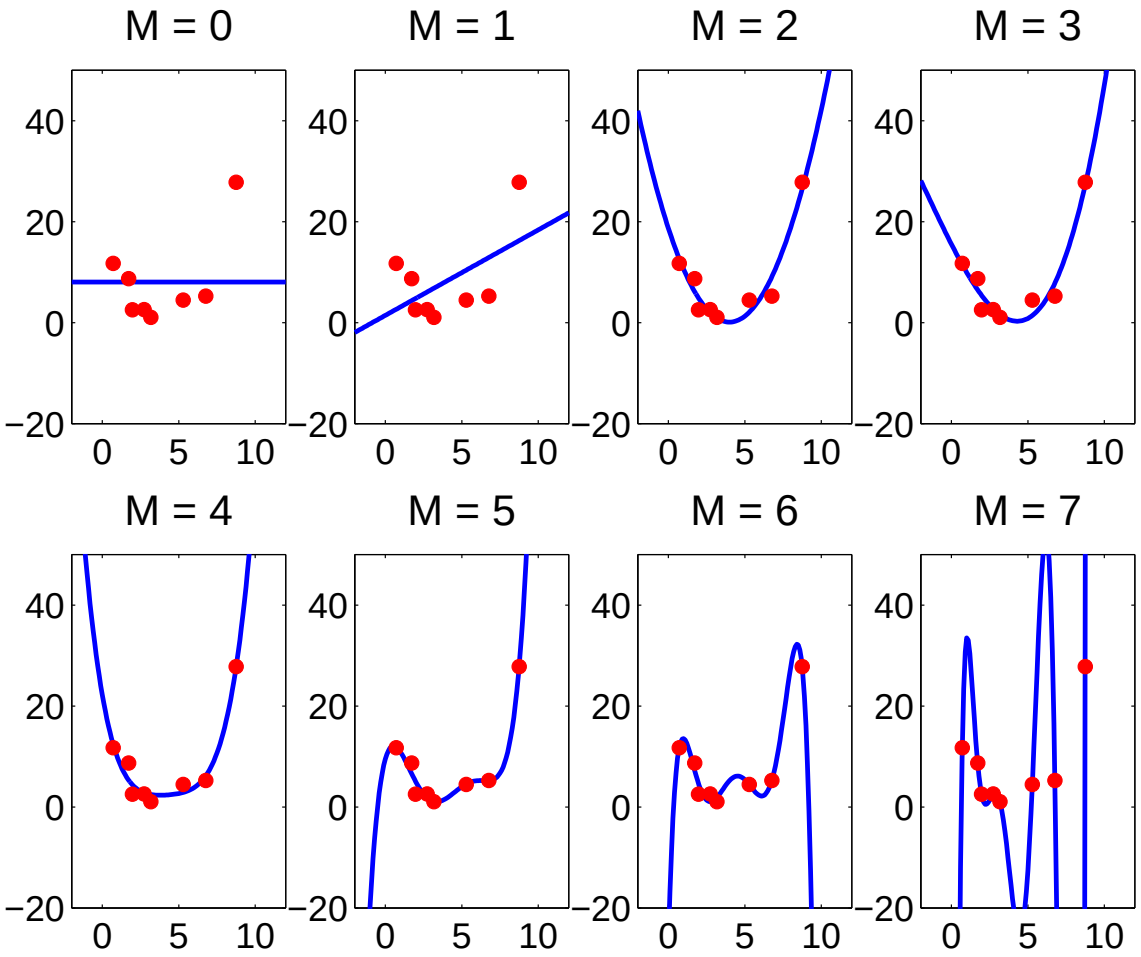
Interpretation of the Marginal Likelihood ("evidence"): The probability that *randomly selected* parameters from the prior would generate $\mathcal{D}$.

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

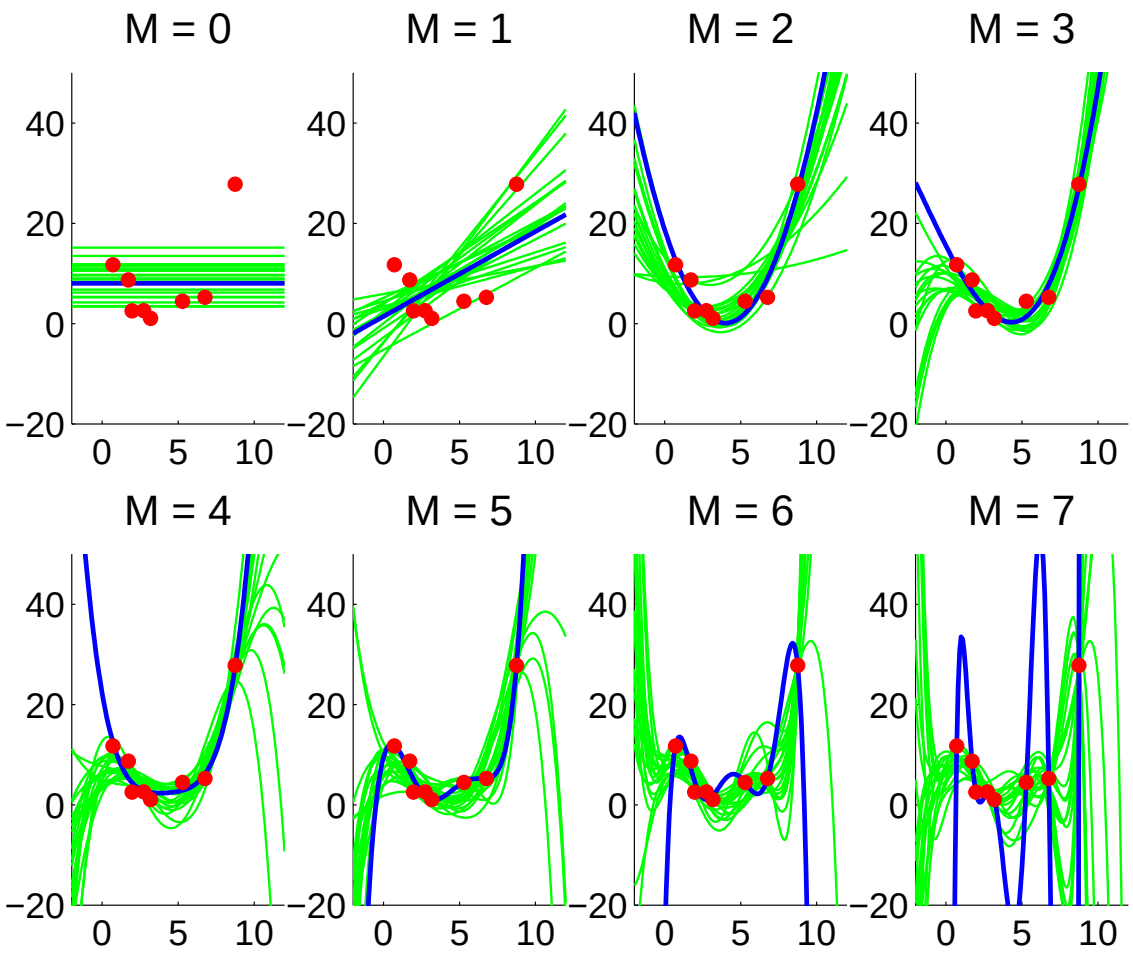
Model classes that are too complex can generate many possible data sets, so again, they are unlikely to generate that particular data set at random.

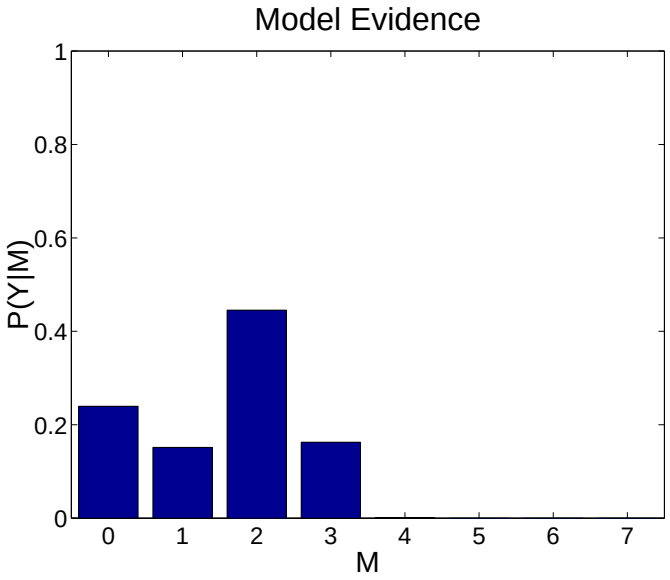


Model structure and overfitting:
A simple example: polynomial regression



Bayesian Model Comparison: Occam's Razor at Work





demo: polybayes

On Choosing Priors

- **Objective Priors**: noninformative priors that attempt to capture ignorance and have good frequentist properties.

- **Subjective Priors**: priors should capture our beliefs as well as possible. They are subjective but not arbitrary.

- **Hierarchical Priors**: multiple levels of priors:

$$\begin{aligned} p(\theta) &= \int d\alpha\, p(\theta|\alpha)p(\alpha) \\ &= \int d\alpha\, p(\theta|\alpha) \int d\beta\, p(\alpha|\beta)p(\beta) \quad \text{(etc...)} \end{aligned}$$

- **Empirical Priors**: learn some of the parameters of the prior from the data ("Empirical Bayes")

Objective Priors

Non-informative priors:

Consider a Gaussian with mean μ and variance σ^2.

The parameter μ informs about the *location of the data*.
If we pick $p(\mu) = p(\mu - a)\ \forall a$ then predictions are location invariant

$$p(x|x') = p(x - a|x' - a)$$

But $p(\mu) = p(\mu - a)\ \forall a$ implies $p(\mu) = \text{Unif}(-\infty, \infty)$ which is **improper**.

Similarly, σ informs about the *scale of the data*, so we can pick $p(\sigma) \propto 1/\sigma$

Problems: It is hard (impossible) to generalize to all parameters of a complicated model. Risk of incoherent inferences (e.g. $E_x E_y[Y|X] \neq E_y[Y]$), paradoxes, and improper posteriors.

Subjective Priors

Priors should capture out beliefs as well as possible.

Otherwise we are not coherent.

End of story!

How do we know our beliefs?

- Think about the problems domain (no black box view of machine learning)
- Generate data from the prior. Does it match expectations?

Even very vague beliefs can be useful.

Empirical "Priors"

Consider a hierarchical model with parameters θ and hyperparameters α

$$p(\mathcal{D}|\alpha) = \int p(\mathcal{D}|\theta)p(\theta|\alpha)\,d\theta$$

Estimate hyperparameters from the data

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmax}} p(\mathcal{D}|\alpha) \qquad \text{(level II ML)}$$

Prediction:

$$p(x|\mathcal{D}, \hat{\alpha}) = \int p(x|\theta)p(\theta|\mathcal{D}, \hat{\alpha})\,d\theta$$

Advantages: Robust—overcomes some limitations of mis-specification of the prior.

Problem: Double counting of evidence / overfitting.

Bayes Rule Applied to Machine Learning

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

$P(\mathcal{D}\|\theta)$	likelihood of θ
$P(\theta)$	prior on θ
$P(\theta\|\mathcal{D})$	posterior of θ given $\mathcal{D}$

Model Comparison:

$$\begin{aligned} P(m|\mathcal{D}) &= \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})} \\ P(\mathcal{D}|m) &= \int P(\mathcal{D}|\theta, m)P(\theta|m)\, d\theta \end{aligned}$$

Prediction:

$$\begin{aligned} P(x|\mathcal{D}, m) &= \int P(x|\theta, \mathcal{D}, m)P(\theta|\mathcal{D}, m)d\theta \\ P(x|\mathcal{D}, m) &= \int P(x|\theta)P(\theta|\mathcal{D}, m)d\theta \qquad \text{(for many models)} \end{aligned}$$

Computing Marginal Likelihoods can be Computationally Intractable

Observed data $\mathbf{y}$, hidden variables $\mathbf{x}$, parameters $\boldsymbol{\theta}$, model class m.

$$p(\mathbf{y}|m) = \int p(\mathbf{y}|\boldsymbol{\theta}, m)\, p(\boldsymbol{\theta}|m)\, d\boldsymbol{\theta}$$

- This can be a very **high dimensional integral**.
- The presence of **latent variables** results in additional dimensions that need to be marginalized out.

$$p(\mathbf{y}|m) = \int \int p(\mathbf{y}, \mathbf{x}|\boldsymbol{\theta}, m)\, p(\boldsymbol{\theta}|m)\, d\mathbf{x}\, d\boldsymbol{\theta}$$

- The likelihood term can be **complicated**.

Approximation Methods for Posteriors and Marginal Likelihoods

- Laplace approximation
- Bayesian Information Criterion (BIC)
- Variational approximations
- Expectation Propagation (EP)
- Markov chain Monte Carlo methods (MCMC)
- Exact Sampling
- ...

Note: there are other deterministic approximations; we won't review them all.

Laplace Approximation

data set $\mathbf{y}$, models $m, m', \ldots,$ parameter $\boldsymbol{\theta}, \boldsymbol{\theta}' \ldots$

Model Comparison: $P(m|\mathbf{y}) \propto P(m)p(\mathbf{y}|m)$

For large amounts of data (relative to number of parameters, d) the parameter posterior is approximately Gaussian around the MAP estimate $\hat{\boldsymbol{\theta}}$:

$$p(\boldsymbol{\theta}|\mathbf{y}, m) \approx (2\pi)^{-\frac{d}{2}}|A|^{\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top A\,(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right\}$$

where $-A$ is the $d \times d$ Hessian of the log posterior $A_{ij} = -\frac{d^2}{d\theta_i d\theta_j} \ln p(\boldsymbol{\theta}|\mathbf{y}, m)\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$

$$p(\mathbf{y}|m) = \frac{p(\boldsymbol{\theta}, \mathbf{y}|m)}{p(\boldsymbol{\theta}|\mathbf{y}, m)}$$

Evaluating the above expression for $\ln p(\mathbf{y}|m)$ at $\hat{\boldsymbol{\theta}}$:

$$\ln p(\mathbf{y}|m) \approx \ln p(\hat{\boldsymbol{\theta}}|m) + \ln p(\mathbf{y}|\hat{\boldsymbol{\theta}}, m) + \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|A|$$

This can be used for model comparison/selection.

Bayesian Information Criterion (BIC)

BIC can be obtained from the Laplace approximation:

$$\ln p(\mathbf{y}|m) \approx \ln p(\hat{\boldsymbol{\theta}}|m) + \ln p(\mathbf{y}|\hat{\boldsymbol{\theta}}, m) + \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln |A|$$

by taking the large sample limit $(n \to \infty)$ where n is the number of data points:

$$\ln p(\mathbf{y}|m) \approx \ln p(\mathbf{y}|\hat{\boldsymbol{\theta}}, m) - \frac{d}{2}\ln n$$

Properties:

- Quick and easy to compute
- It does not depend on the prior
- We can use the ML estimate of θ instead of the MAP estimate
- It is equivalent to the MDL criterion
- Assumes that as $n \to \infty$, all the parameters are well-determined (i.e. the model is identifiable; otherwise, d should be the number of well-determined parameters)
- **Danger:** counting parameters can be deceiving! (c.f. sinusoid, infinite models)

Lower Bounding the Marginal Likelihood

Variational Bayesian Learning

Let the latent variables be $\mathbf{x}$, observed data $\mathbf{y}$ and the parameters $\boldsymbol{\theta}$.
We can lower bound the marginal likelihood (Jensen's inequality):

$$\begin{aligned}
\ln p(\mathbf{y}|m) &= \ln \int p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}|m)\, d\mathbf{x}\, d\boldsymbol{\theta} \\
&= \ln \int q(\mathbf{x}, \boldsymbol{\theta}) \frac{p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}|m)}{q(\mathbf{x}, \boldsymbol{\theta})}\, d\mathbf{x}\, d\boldsymbol{\theta} \\
&\geq \int q(\mathbf{x}, \boldsymbol{\theta}) \ln \frac{p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}|m)}{q(\mathbf{x}, \boldsymbol{\theta})}\, d\mathbf{x}\, d\boldsymbol{\theta}.
\end{aligned}$$

Use a simpler, factorised approximation for $q(\mathbf{x}, \boldsymbol{\theta}) \approx q_{\mathbf{x}}(\mathbf{x}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$:

$$\begin{aligned}
\ln p(\mathbf{y}|m) &\geq \int q_{\mathbf{x}}(\mathbf{x}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln \frac{p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}|m)}{q_{\mathbf{x}}(\mathbf{x}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}\, d\mathbf{x}\, d\boldsymbol{\theta} \\
&\stackrel{\text{def}}{=} \mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \mathbf{y}).
\end{aligned}$$

Variational Bayesian Learning . . .

Maximizing this lower bound, $\mathcal{F}_m$, leads to **EM-like** iterative updates:

$$q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) \quad \propto \quad \exp\left[\int \ln p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta},m)\, q_{\boldsymbol{\theta}}^{(t)}(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right] \qquad \text{E-like step}$$

$$q_{\boldsymbol{\theta}}^{(t+1)}(\boldsymbol{\theta}) \quad \propto \quad p(\boldsymbol{\theta}|m)\ \exp\left[\int \ln p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta},m)\, q_{\mathbf{x}}^{(t+1)}(\mathbf{x})\, d\mathbf{x}\right] \qquad \text{M-like step}$$

Maximizing $\mathcal{F}_m$ is equivalent to minimizing KL-divergence between the *approximate posterior*, $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})\ q_{\mathbf{x}}(\mathbf{x})$ and the *true posterior*, $p(\boldsymbol{\theta},\mathbf{x}|\mathbf{y},m)$:

$$\ln p(\mathbf{y}|m) - \mathcal{F}_m(q_{\mathbf{x}}(\mathbf{x}), q_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \mathbf{y}) = \int q_{\mathbf{x}}(\mathbf{x})\ q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln \frac{q_{\mathbf{x}}(\mathbf{x})\ q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{p(\boldsymbol{\theta},\mathbf{x}|\mathbf{y},m)}\, d\mathbf{x}\, d\boldsymbol{\theta} = \mathbf{KL}(q\|p)$$

In the limit as $n \to \infty$, for identifiable models, the variational lower bound approaches the BIC criterion.

The Variational Bayesian EM algorithm

EM for MAP estimation

Goal: maximize $p(\boldsymbol{\theta}|\mathbf{y}, m)$ w.r.t. $\boldsymbol{\theta}$

E Step: compute

$$q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^{(t)})$$

M Step:

$$\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \int q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) \ln p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) \, d\mathbf{x}$$

Variational Bayesian EM

Goal: lower bound $p(\mathbf{y}|m)$

VB-E Step: compute

$$q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}, \bar{\boldsymbol{\phi}}^{(t)})$$

VB-M Step:

$$q_{\boldsymbol{\theta}}^{(t+1)}(\boldsymbol{\theta}) \propto \exp\left[\int q_{\mathbf{x}}^{(t+1)}(\mathbf{x}) \ln p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) \, d\mathbf{x}\right]$$

Properties:

- Reduces to the EM algorithm if $q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$.
- $\mathcal{F}_m$ increases monotonically, and incorporates the model complexity penalty.
- Analytical parameter distributions (but not constrained to be Gaussian).
- VB-E step has same complexity as corresponding E step.
- We can use the junction tree, belief propagation, Kalman filter, etc, algorithms in the VB-E step of VB-EM, but **using expected natural parameters**, $\bar{\boldsymbol{\phi}}$.

Variational Bayesian EM

The Variational Bayesian EM algorithm has been used to approximate Bayesian learning in a wide range of models such as:

- probabilistic PCA and factor analysis
- mixtures of Gaussians and mixtures of factor analysers
- hidden Markov models
- state-space models (linear dynamical systems)
- independent components analysis (ICA)
- discrete graphical models...

The main advantage is that it can be used to **automatically do model selection** and does not suffer from overfitting to the same extent as ML methods do.

Also it is about as computationally demanding as the usual EM algorithm.

See: `www.variational-bayes.org`

mixture of Gaussians demo: `run_simple`

Expectation Propagation (EP)

Data (iid) $\mathcal{D} = \{\mathbf{x}^{(1)} \dots, \mathbf{x}^{(N)}\}$, model $p(\mathbf{x}|\boldsymbol{\theta})$, with parameter prior $p(\boldsymbol{\theta})$.

The parameter posterior is: $$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{1}{p(\mathcal{D})} p(\boldsymbol{\theta}) \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$$

We can write this as product of factors over $\boldsymbol{\theta}$: $$p(\boldsymbol{\theta}) \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) = \prod_{i=0}^{N} f_i(\boldsymbol{\theta})$$

where $f_0(\boldsymbol{\theta}) \stackrel{\text{def}}{=} p(\boldsymbol{\theta})$ and $f_i(\boldsymbol{\theta}) \stackrel{\text{def}}{=} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$ and we will ignore the constants.

We wish to approximate this by a product of *simpler* terms: $$q(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \prod_{i=0}^{N} \tilde{f}_i(\boldsymbol{\theta})$$

$$\min_{q(\boldsymbol{\theta})} \mathrm{KL}\left(\prod_{i=0}^{N} f_i(\boldsymbol{\theta}) \middle\| \prod_{i=0}^{N} \tilde{f}_i(\boldsymbol{\theta}) \right) \quad \text{(intractable)}$$

$$\min_{\tilde{f}_i(\boldsymbol{\theta})} \mathrm{KL}\left(f_i(\boldsymbol{\theta}) \| \tilde{f}_i(\boldsymbol{\theta}) \right) \quad \text{(simple, non-iterative, inaccurate)}$$

$$\min_{\tilde{f}_i(\boldsymbol{\theta})} \mathrm{KL}\left(f_i(\boldsymbol{\theta}) \prod_{j \neq i} \tilde{f}_j(\boldsymbol{\theta}) \middle\| \tilde{f}_i(\boldsymbol{\theta}) \prod_{j \neq i} \tilde{f}_j(\boldsymbol{\theta}) \right) \quad \text{(simple, iterative, accurate)} \leftarrow \text{EP}$$

Expectation Propagation

Input $f_0(\boldsymbol{\theta}) \dots f_N(\boldsymbol{\theta})$
Initialize $\tilde{f}_0(\boldsymbol{\theta}) = f_0(\boldsymbol{\theta})$, $\tilde{f}_i(\boldsymbol{\theta}) = 1$ for $i > 0$, $q(\boldsymbol{\theta}) = \prod_i \tilde{f}_i(\boldsymbol{\theta})$
repeat
  for $i = 0 \dots N$ **do**
    Deletion: $q_{\backslash i}(\boldsymbol{\theta}) \leftarrow \dfrac{q(\boldsymbol{\theta})}{\tilde{f}_i(\boldsymbol{\theta})} = \prod_{j \neq i} \tilde{f}_j(\boldsymbol{\theta})$
    Projection: $\tilde{f}_i^{\text{new}}(\boldsymbol{\theta}) \leftarrow \arg\min_{f(\boldsymbol{\theta})} \mathrm{KL}(f_i(\boldsymbol{\theta}) q_{\backslash i}(\boldsymbol{\theta}) \| f(\boldsymbol{\theta}) q_{\backslash i}(\boldsymbol{\theta}))$
    Inclusion: $q(\boldsymbol{\theta}) \leftarrow \tilde{f}_i^{\text{new}}(\boldsymbol{\theta})\, q_{\backslash i}(\boldsymbol{\theta})$
  end for
until convergence

The EP algorithm. Some variations are possible: here we assumed that f_0 is in the exponential family, and we updated sequentially over i. The names for the steps (deletion, projection, inclusion) are not the same as in (Minka, 2001)

- Minimizes the opposite KL to variational methods
- $\tilde{f}_i(\boldsymbol{\theta})$ in exponential family $\rightarrow$ projection step is moment matching
- Loopy belief propagation and assumed density filtering are special cases
- No convergence guarantee (although convergent forms can be developed)

An Overview of Sampling Methods

Monte Carlo Methods:

- Simple Monte Carlo
- Rejection Sampling
- Importance Sampling
- etc.

Markov Chain Monte Carlo Methods:

- Gibbs Sampling
- Metropolis Algorithm
- Hybrid Monte Carlo
- etc.

Exact Sampling Methods

Markov chain Monte Carlo (MCMC) methods

Assume we are interested in drawing samples from some desired distribution $p^*(\theta)$, e.g. $p^*(\theta) = p(\theta|\mathcal{D}, m)$.

We define a Markov chain:

$$\theta_0 \rightarrow \theta_1 \rightarrow \theta_2 \rightarrow \theta_3 \rightarrow \theta_4 \rightarrow \theta_5 \ldots$$

where $\theta_0 \sim p_0(\theta)$, $\theta_1 \sim p_1(\theta)$, etc, with the property that:

$$p_t(\theta') = \int p_{t-1}(\theta)\ T(\theta \rightarrow \theta')\, d\theta$$

where $T(\theta \rightarrow \theta')$ is the **Markov chain transition probability** from θ to θ'.

We say that $p^*(\theta)$ is an **invariant (or stationary) distribution** of the Markov chain defined by T iff:

$$p^*(\theta') = \int p^*(\theta)\, T(\theta \rightarrow \theta')\, d\theta$$

Markov chain Monte Carlo (MCMC) methods

We have a Markov chain $\theta_0 \to \theta_1 \to \theta_2 \to \theta_3 \to \ldots$ where $\theta_0 \sim p_0(\theta)$, $\theta_1 \sim p_1(\theta)$, etc, with the property that:

$$p_t(\theta') = \int p_{t-1}(\theta)\ T(\theta \to \theta')\, d\theta$$

where $T(\theta \to \theta')$ is the Markov chain transition probability from θ to θ'.
A useful condition that implies invariance of $p^*(\theta)$ is **detailed balance**:

$$p^*(\theta')T(\theta' \to \theta) = p^*(\theta)T(\theta \to \theta')$$

MCMC methods define **ergodic** Markov chains, which converge to a unique stationary distribution (also called an *equilibrium distribution*) regardless of the initial conditions $p_0(\theta)$:

$$\lim_{t\to\infty} p_t(\theta) = p^*(\theta)$$

Procedure: define an MCMC method with equilibrium distribution $p(\theta|\mathcal{D}, m)$, run method and collect samples. There are also sampling methods for $p(\mathcal{D}|m)$.

BREAK AND QUESTIONS

Further Topics

- Feature Selection and ARD
- Bayesian Discriminative Learning (BPM vs SVM)
- From Parametric to Nonparametric Methods
  - Gaussian Processes
  - Dirichlet Process Mixtures
  - Other Non-parametric Bayesian Methods
- Bayesian Decision Theory and Active Learning
- Bayesian Semi-supervised Learning

Feature Selection

Example: classification

$$\text{input } \mathbf{x} = (x_1, \ldots, x_D) \in \Re^D$$
$$\text{output } y \in \{+1, -1\}$$

2^D possible subsets of relevant input features.

One approach, consider all models $m \in \{0, 1\}^D$ and find

$$\hat{m} = \underset{m}{\text{argmax}}\ p(\mathcal{D}|m)$$

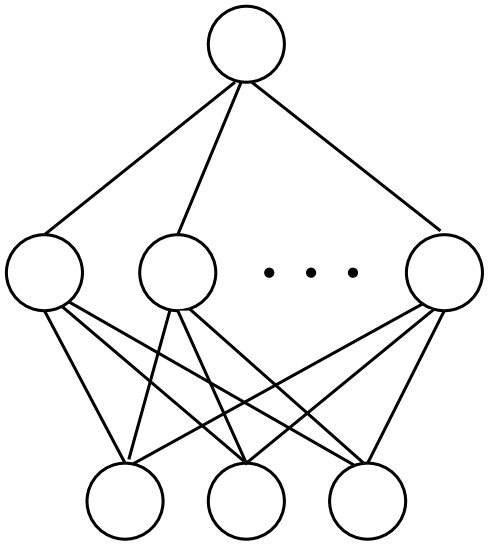
Problems: intractable, overfitting, we should really average

Feature Selection

- Why are we doing feature selection?
- What does it cost us to keep all the features?
- Usual answer (overfitting) does not apply to fully Bayesian methods, since they don't involve any fitting.
- We should only do feature selection if there is a cost associated with measuring features or predicting with many features.

Note: Radford Neal won the NIPS feature selection competition using Bayesian methods that used 100% of the features.

Feature Selection: Automatic Relevance Determination



Bayesian neural network

Data: $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N} = (X, \mathbf{y})$
Parameters (weights): $\boldsymbol{\theta} = \{\{w_{ij}\}, \{v_k\}\}$

prior	$p(\boldsymbol{\theta}\|\boldsymbol{\alpha})$
posterior	$p(\boldsymbol{\theta}\|\boldsymbol{\alpha}, \mathcal{D}) \propto p(\mathbf{y}\|X, \boldsymbol{\theta})p(\boldsymbol{\theta}\|\boldsymbol{\alpha})$
evidence	$p(\mathbf{y}\|X, \boldsymbol{\alpha}) = \int p(\mathbf{y}\|X, \boldsymbol{\theta})p(\boldsymbol{\theta}\|\boldsymbol{\alpha})\, d\boldsymbol{\theta}$
prediction	$p(y'\|\mathcal{D}, \mathbf{x}', \boldsymbol{\alpha}) = \int p(y'\|\mathbf{x}', \boldsymbol{\theta})p(\boldsymbol{\theta}\|\mathcal{D}, \boldsymbol{\alpha})\, d\boldsymbol{\theta}$

Automatic Relevance Determination (ARD):

Let the weights from feature x_d have variance α_d^{-1}: $\quad p(w_{dj}|\alpha_d) = \mathcal{N}(0, \alpha_d^{-1})$

Let's think about this:

$\alpha_d \to \infty$	variance $\to 0$	weights $\to 0$	(irrelevant)
$\alpha_d \ll \infty$	finite variance	weight can vary	(relevant)

ARD: Infer relevances $\boldsymbol{\alpha}$ from data. Often we can optimize $\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmax}}\ p(\mathbf{y}|X, \boldsymbol{\alpha})$.

During optimization some α_d will go to ∞, so the model will discover irrelevant inputs.

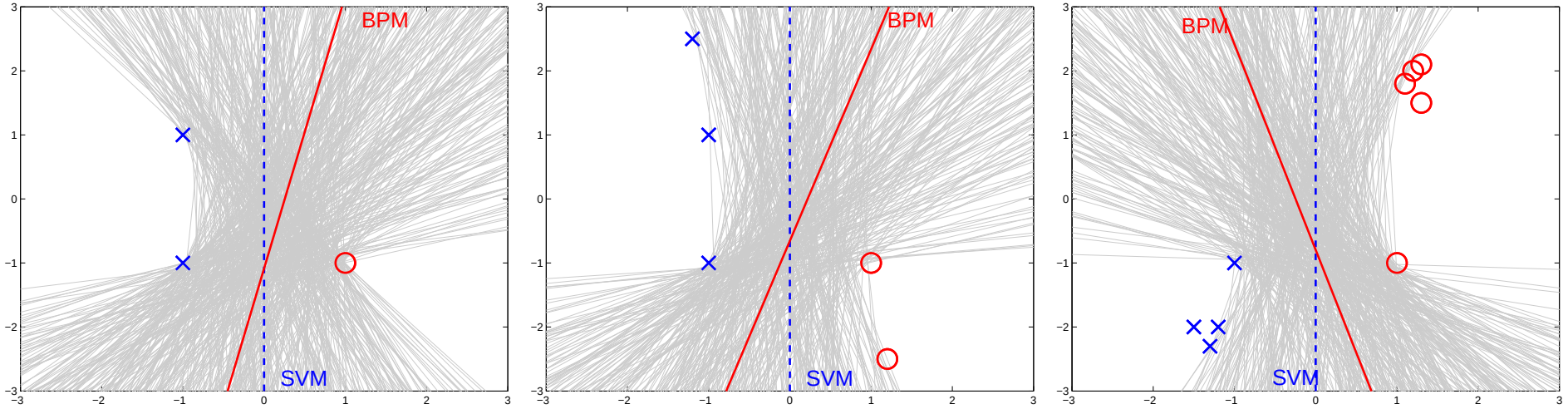
Bayesian Discriminative Modeling

Terminology for classification with inputs $\mathbf{x}$ and classes y:

- **Generative Model:** models prior $p(y)$ and class-conditional density $p(\mathbf{x}|y)$
- **Discriminative Model:** directly models the conditional distribution $p(y|\mathbf{x})$ or the class boundary e.g. $\{\mathbf{x} : p(y = +1|\mathbf{x}) = 0.5\}$

Myth: Bayesian Methods = Generative Models

For example, it is possible to define Bayesian kernel classifiers (e.g. Bayes point machines, and Gaussian processes) analogous to support vector machines (SVMs).



(figure adapted from Minka, 2001)

Parametric vs Nonparametric Models

Terminology (roughly):

- **Parametric Models** have a finite fixed number of parameters $\boldsymbol{\theta}$, regardless of the size of the data set. Given $\boldsymbol{\theta}$, the predictions are independent of the data $\mathcal{D}$:

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

  The parameters are a finite summary of the data. We can also call this model-based learning (e.g. mixture of k Gaussians)

- **Non-parametric Models** allow the number of "parameters" to grow with the data set size, or alternatively we can think of the predictions as depending on the data, and possible a usually small number of parameters $\boldsymbol{\alpha}$

$$p(x|\mathcal{D}, \alpha)$$

  We can also call this memory-based learning (e.g. kernel density estimation)

A key question: are a fixed finite number of sufficient statistics of the data needed to make predictions?

Nonparametric Bayesian Methods (Infinite Models)

We ought not to limit the complexity of our model a priori (e.g. number of hidden states, number of basis functions, number of mixture components, etc) since we don't believe that the real data was actually generated from a statistical model with a small number of parameters.

Therefore, regardless of how much training data we have, we should consider models with as many parameters as we can handle computationally.

Here there is no model order selection task:

- No need to compare marginal likelihoods to select model order (which is often difficult).
- No need to use Occam's razor to limit the number of parameters in the model.

In fact, we may even want to consider doing inference in models with an infinite number of parameters...

Gaussian Processes for Regression

Two ways of understanding Gaussian processes (GPs)...

- Starting from multivariate Gaussians
- Starting from linear regression

...from multivariate Gaussians to GPs...

univariate Gaussian density for t

$$p(t) = (2\pi\sigma^2)^{-1/2} \exp\left\{-\frac{t^2}{2\sigma^2}\right\}$$

multivariate Gaussian density for $\mathbf{t} = (t_1, t_2, t_3, \ldots t_N)^\top$

$$p(\mathbf{t}) = |2\pi\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}\mathbf{t}^\top\Sigma^{-1}\mathbf{t}\right\}$$

Σ is an $N \times N$ covariance matrix.

Imagine that Σ_{ij} depends on i and j and we plot samples of $\mathbf{t}$ as if they were functions...

`gpdemogen` and `gpdemo`

...from linear regression to GPs...

- Linear regression with inputs $\mathbf{x}_i$ and outputs t_i: $$t_i = \sum_d w_d \, x_{id} + \epsilon_i$$

- Linear regression with basis functions ("kernel trick"): $$t_i = \sum_d w_d \, \phi_d(\mathbf{x}_i) + \epsilon_i$$

- Bayesian linear regression with basis functions:

$$w_d \sim \mathcal{N}(0, \beta_d) \quad \text{(independent of } w_\ell, \, \forall \ell \neq d), \qquad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

- Integrating out the weights, w_d, we find:

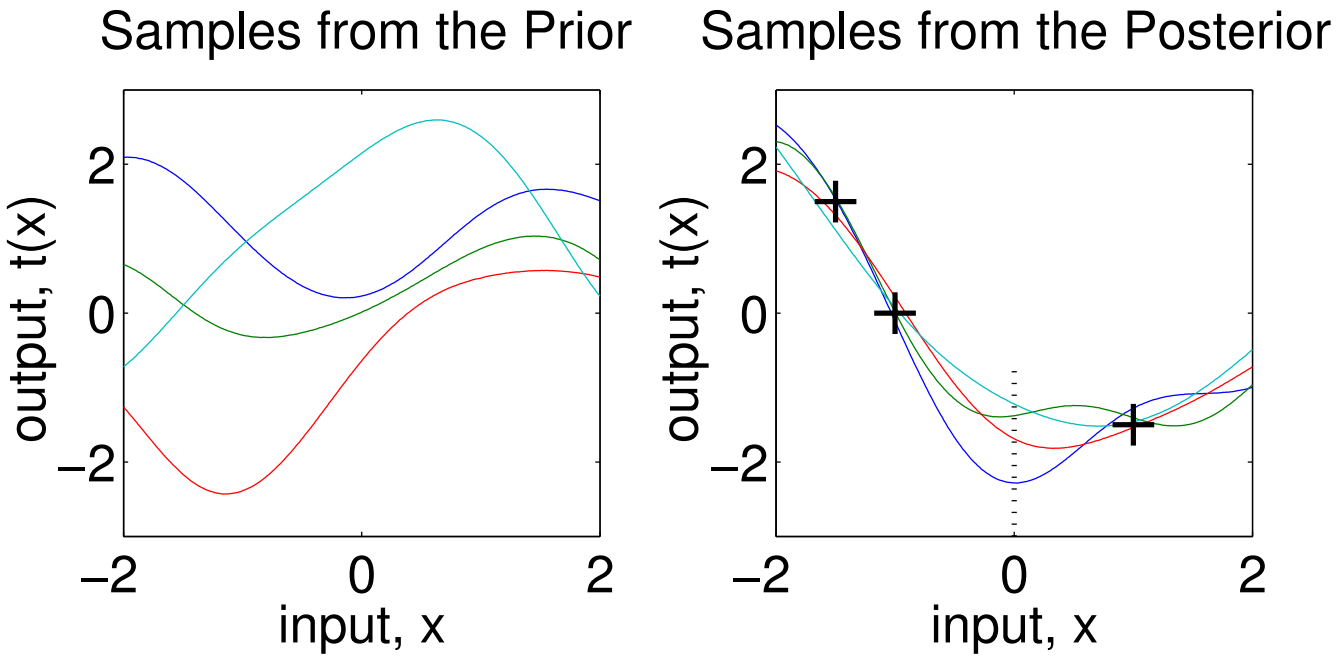
$$E[t_i] = 0, \qquad E[t_i t_j] = C_{ij} \stackrel{\text{def}}{=} \sum_d \beta_d \, \phi_d(\mathbf{x}_i) \, \phi_d(\mathbf{x}_j) + \delta_{ij} \sigma^2$$

This is a Gaussian process with covariance function $C(\mathbf{x}_i, \mathbf{x}_j) \stackrel{\text{def}}{=} C_{ij}$.

This Gaussian process has a finite number of basis functions. Many useful GP covariance functions correspond to infinitely many basis functions.

Gaussian Process Regression

A Gaussian Process (GP) places a prior directly on the space of functions such that at any finite selection of points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ the corresponding function values $t^{(1)}, \ldots, t^{(N)}$ have a multivariate Gaussian distribution.

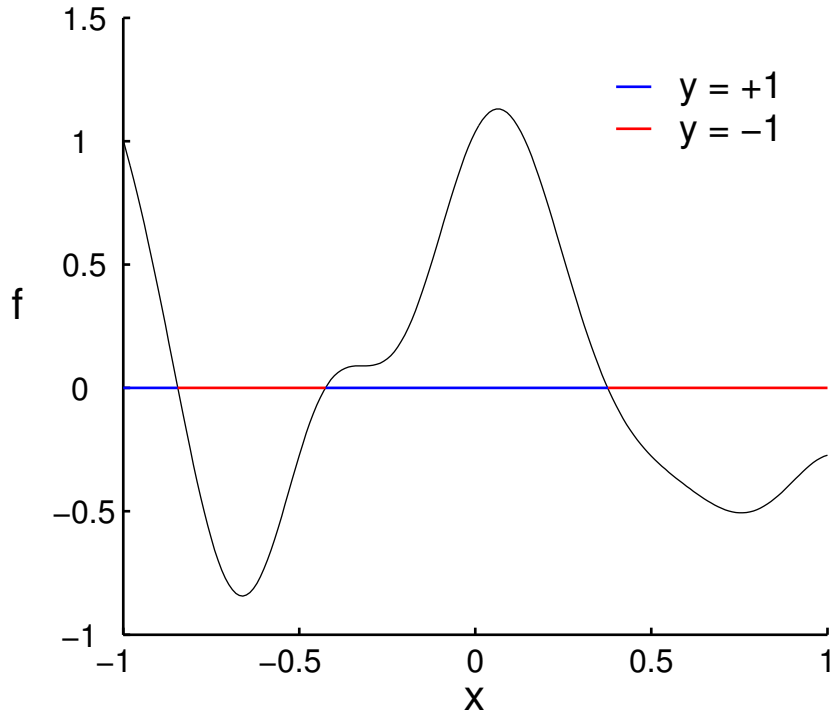


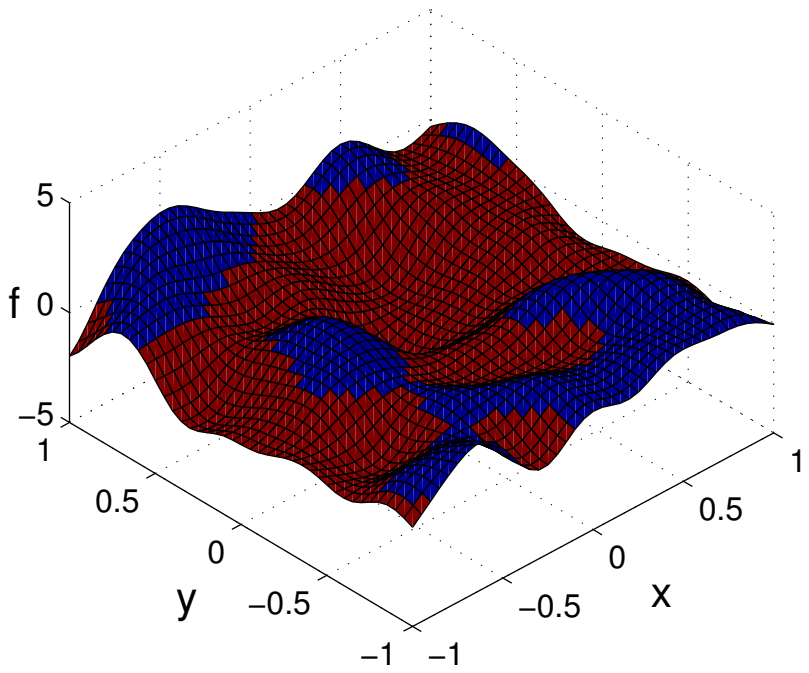
The covariance between two function values $t^{(i)}$ and $t^{(j)}$ under the prior is given by the covariance function $C(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, which typically decays monotonically with $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|$, encoding smoothness.

GPs are "Bayesian Kernel Regression Machines"

Using Gaussian Processes for Classification

Binary classification problem: Given a data set $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, where $y^{(n)} \in \{-1, +1\}$, infer class label probabilities at new points.





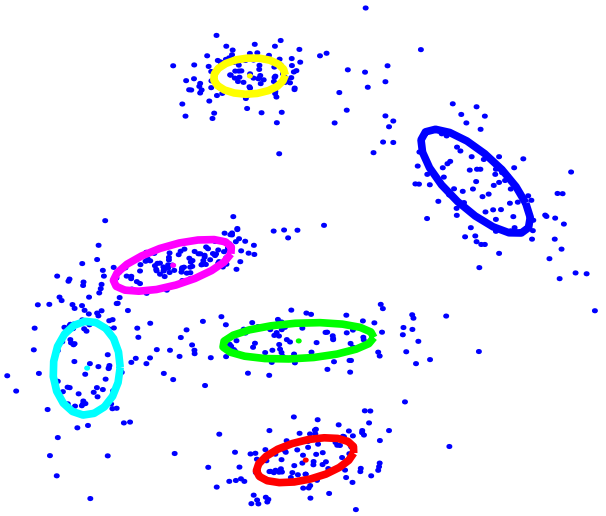
There are many ways to relate function values $f^{(n)}$ to class probabilities:

$$p(y|f) = \begin{cases} \frac{1}{1+\exp(-yf)} & \text{sigmoid (logistic)} \\ \Phi(yf) & \text{cumulative normal (probit)} \\ \boldsymbol{H}(yf) & \text{threshold} \\ \epsilon + (1-2\epsilon)\boldsymbol{H}(yf) & \text{robust threshold} \end{cases}$$

Dirichlet Process Mixtures (Infinite Mixtures)

Consider using a finite mixture of K components to model a data set $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$

$$
\begin{aligned}
p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) &= \sum_{j=1}^{K} \pi_j \, p_j(\mathbf{x}^{(i)}|\boldsymbol{\theta}_j) \\
&= \sum_{j=1}^{K} P(s^{(i)} = j|\boldsymbol{\pi}) \, p_j(\mathbf{x}^{(i)}|\boldsymbol{\theta}_j, s^{(i)} = j)
\end{aligned}
$$



Distribution of indicators $\mathbf{s} = (s^{(1)}, \ldots, s^{(n)})$ given $\boldsymbol{\pi}$ is **multinomial**

$$
P(s^{(1)}, \ldots, s^{(n)}|\boldsymbol{\pi}) = \prod_{j=1}^{K} \pi_j^{n_j}, \qquad n_j \stackrel{\text{def}}{=} \sum_{i=1}^{n} \delta(s^{(i)}, j) \; .
$$

Assume mixing proportions $\boldsymbol{\pi}$ have a given symmetric conjugate **Dirichlet prior**

$$
p(\boldsymbol{\pi}|\alpha) = \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_{j=1}^{K} \pi_j^{\alpha/K - 1}
$$

Dirichlet Process Mixtures (Infinite Mixtures) - II

Distribution of indicators $\mathbf{s} = (s^{(1)}, \ldots, s^{(n)})$ given $\boldsymbol{\pi}$ is **multinomial**

$$P(s^{(1)}, \ldots, s^{(n)}|\boldsymbol{\pi}) = \prod_{j=1}^{K} \pi_j^{n_j}, \qquad n_j \stackrel{\text{def}}{=} \sum_{i=1}^{n} \delta(s^{(i)}, j) \ .$$

Mixing proportions $\boldsymbol{\pi}$ have a symmetric conjugate **Dirichlet prior**

$$p(\boldsymbol{\pi}|\alpha) = \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_{j=1}^{K} \pi_j^{\alpha/K-1}$$

Integrating out the mixing proportions, $\boldsymbol{\pi}$, we obtain

$$P(s^{(1)}, \ldots, s^{(n)}|\alpha) = \int d\boldsymbol{\pi} \ P(\mathbf{s}|\boldsymbol{\pi})P(\boldsymbol{\pi}|\alpha) = \frac{\Gamma(\alpha)}{\Gamma(n+\alpha)} \prod_{j=1}^{K} \frac{\Gamma(n_j + \alpha/K)}{\Gamma(\alpha/K)}$$

Dirichlet Process Mixtures (Infinite Mixtures) - III

Starting from $$P(\mathbf{s}|\alpha) = \frac{\Gamma(\alpha)}{\Gamma(n+\alpha)} \prod_{j=1}^{K} \frac{\Gamma(n_j + \alpha/K)}{\Gamma(\alpha/K)}$$

Conditional Probabilities: Finite K

$$P(s^{(i)} = j|\mathbf{s}_{-i}, \alpha) = \frac{n_{-i,j} + \alpha/K}{n - 1 + \alpha}$$

where $\mathbf{s}_{-i}$ denotes all indices except i, and $n_{-i,j} \stackrel{\text{def}}{=} \sum_{\ell \neq i} \delta(s^{(\ell)}, j)$

DP: more populous classes are more more likely to be joined

Conditional Probabilities: Infinite K

Taking the limit as $K \to \infty$ yields the conditionals

$$P(s^{(i)} = j|\mathbf{s}_{-i}, \alpha) = \begin{cases} \frac{n_{-i,j}}{n-1+\alpha} & j \text{ represented} \\ \frac{\alpha}{n-1+\alpha} & \text{all } j \text{ not represented} \end{cases}$$

Left over mass, α, $\Rightarrow$ **countably infinite** number of indicator settings.
Gibbs sampling from posterior of indicators is often easy!

Other Non-parametric Bayesian Models

- Infinite Hidden Markov models
- Hierarchical Dirichlet Processes
- Dirichlet Diffusion Trees
- Infinite mixtures of Gaussian Processes
- Indian Buffet Processes
- ...

Bayesian Decision Theory

Bayesian decision theory deals with the problem of making optimal decisions—that is, decisions or actions that minimize an expected loss.

- Let's say we have a choice of taking one of k possible **actions** $a_1 \ldots a_k$.
- Assume that the world can be in one of m different **states** $s_1, \ldots, s_m$.
- If we take action a_i and the world is in state s_j we incur a **loss** ℓ_{ij}
- Given all the observed data $\mathcal{D}$ and prior background knowledge $\mathcal{B}$, our **beliefs** about the state of the world are summarized by $p(s|\mathcal{D}, \mathcal{B})$.
- *The optimal action is the one which is expected to minimize loss (or maximize utility)*:

$$a^* = \underset{a_i}{\operatorname{argmin}} \sum_{j=1}^{m} \ell_{ij}\, p(s_j|\mathcal{D}, \mathcal{B})$$

Bayesian sequential decision theory	(statistics)
Optimal control theory	(engineering)
Reinforcement learning	(computer science / psychology)

Bayesian Active Learning

Active Learning is a special case of Bayesian Decision Theory

An example:

- Consider an active classification problem with a labeled data set $\mathcal{D}_\ell = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$ and a pool of unlabeled data points $\mathcal{D}_u = \{x^{(n+1)}, \ldots, x^{(n+m)}\}$.
- Assume there is a **cost** associated with finding the true label of a point.
- The **action** is picking which unlabeled point to find the true label for. The remaining points will be labeled according to most probable predicted label after including this true label.
- The **state** of the world is the true labels of all the points.
- The **beliefs** are $p(y^{(n+1)}, \ldots, y^{(n+m)} | \mathcal{D}_\ell, \mathcal{D}_u, \mathcal{B})$
- The **loss** is the misclassification loss on the remaining points and the loss due to the cost of labeling the point.

Bayesian Semi-supervised Learning

In semi-supervised learning you have a labeled data set $\mathcal{D}_\ell = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$ and an unlabeled data set $\mathcal{D}_u = \{x^{(n+1)}, \ldots, x^{(n+m)}\}$. Often $m \gg n$.

Goal: to learn a model $p(y|x)$ (e.g. a classifier, $y \in \{\pm 1\}$)

Question: how should knowledge about $p(x)$ from $\mathcal{D}_u$ affect the classifier $p(y|x)$?

Answer: it all depends on your priors! It's just a missing data problem.

Two Bayesian approaches:

- Generative: Express your beliefs about the generative process $p(y)$ and $p(x|y)$—this induces a relationship between $p(x)$ and $p(y|x)$.

- Discriminative: Directly express some prior that relates parameters of $p(y|x)$ to the parameters in $p(x)$. One simple example is the notion that the decision boundary should prefer to go through regions of low density.

Reconciling Bayesian and Frequentist Views

Frequentist theory tends to focus on **sampling properties**, or on **minimax performance** of methods – i.e. what is the worst case performance if the environment is adversarial. Frequentist methods often optimize some penalized cost function.

Bayesian methods focus on **expected loss** under the posterior. Bayesian methods, in theory, do not make use of optimization, except at the point at which decisions are to be made.

There are some reasons why frequentist procedures are useful to Bayesians:

- **Communication:** If Bayesian A wants to convince Bayesians B, C, and D of the validity of some inference (or even non-Bayesians) then she must determine that not only does this inference follows from prior p_A but also would have followed from p_B, p_C and p_D, etc. For this reason it's useful sometimes to find a prior which has good frequentist (worst-case) properties, even though acting on the prior would not be coherent with our beliefs.
- **Robustness:** Priors with good frequentist properties can be more robust to mis-specifications of the prior. Two ways of dealing with robustness issues are to make sure that the prior is vague enough, and to make use of a loss function to penalize costly errors.

also, recently, PAC-Bayesian frequentist bounds on Bayesian procedures.

Limitations and Criticisms of Bayesian Methods

- They are subjective
- It is hard to come up with a prior, the assumptions are usually wrong.
- The closed world assumption: need to consider all possible hypotheses for the data before observing the data
- They can be computationally demanding
- The use of approximations weakens the coherence argument.

Discussion Questions and Challenges

The following are some of my (possibly) controvertial thoughts:

- "I have no idea why anyone would want to use non-subjective priors. 'Objective' priors are fraught with inconsistencies and no modelling is truly objective anyway. If you want robustness make sure your prior captures a wide range of reasonable outcomes and use decision theory to capture your losses"
- "The above robustness leads to the need for good non-parameteric Bayesian methods – come to my UAI tutorial"
- "Bayesian methods don't overfit, because they don't fit anything! Approximate Bayesian methods can have failure modes that look like overfitting. "
- "Anything you can do easily with an SVM you can do with a Gaussian Process better"
- "Learning theory is useful to analyze bounds on the performance of algorithms but I'm not sure it should be used to design algorithms. Algorithms should be designed to be sensible given the problem at hand, ignoring prior knowledge seems very silly"
- "Well designed MCMC methods can sometimes be much faster and perform better than optimization algorithms"
- "MAP methods, ie using a log prior as a regularizer, are **not** Bayesian."

Summary

- **Introduce Foundations**
  - Some canonical problems: classification, regression, density estimation, coin toss
  - Representing beliefs and the Cox axioms
  - The Dutch Book Theorem
  - Asymptotic Certainty and Consensus
  - Occam's Razor and Marginal Likelihoods
  - Choosing Priors
    - Objective Priors: Noninformative, Jeffreys, Reference
    - Subjective Priors
    - Hierarchical Priors
    - Empirical Priors
    - Conjugate Priors
- **The Intractability Problem**
- **Approximation Tools**
  - Laplace's Approximation
  - Bayesian Information Criterion (BIC)
  - Variational Approximations
  - Expectation Propagation
  - MCMC
  - Exact Sampling
- **Advanced Topics**
  - Feature Selection and ARD
  - Bayesian Discriminative Learning (BPM vs SVM)
  - From Parametric to Nonparametric Methods
    - Gaussian Processes
    - Dirichlet Process Mixtures
    - Other Non-parametric Bayesian Methods
  - Bayesian Decision Theory and Active Learning
  - Bayesian Semi-supervised Learning
- **Limitations and Discussion**
  - Reconciling Bayesian and Frequentist Views
  - Limitations and Criticisms of Bayesian Methods
  - Discussion

Conclusions

- Bayesian methods provide a coherent framework for doing inference under uncertainty and for learning from data
- The ideas are simple, although execution can be hard
- There are still many open research directions
- You can find out more from my tutorial paper:
  Ghahramani (2004) Unsupervised Learning. In Bousquet, O., Raetsch, G. and von Luxburg, U. (eds) Advanced Lectures on Machine Learning LNAI 3176. Springer-Verlag.
  `http://www.gatsby.ucl.ac.uk/~zoubin/course04/ul.pdf`

`http://www.gatsby.ucl.ac.uk/~zoubin`

(for more resources, also to contact me
if interested in a PhD or postdoc)

Thanks for your patience!

Appendix

Objective Priors

Reference Priors:

Captures the following notion of noninformativeness. Given a model $p(x|\theta)$ we wish to find the prior on θ such that an experiment involving observing x is expected to provide the most information about θ.

That is, most of the information about θ will come from the experiment rather than the prior. The information about θ is:

$$I(\theta|x) = -\int p(\theta)\log p(\theta)d\theta - (-\int p(\theta, x)\log p(\theta|x)d\theta\ dx)$$

This can be generalized to experiments with n obserations (giving different answers)

Problems: Hard to compute in general (e.g. MCMC schemes), prior depends on the size of data to be observed.

Objective Priors

Jeffreys Priors:

Motivated by invariance arguments: the principle for choosing priors should not depend on the parameterization.

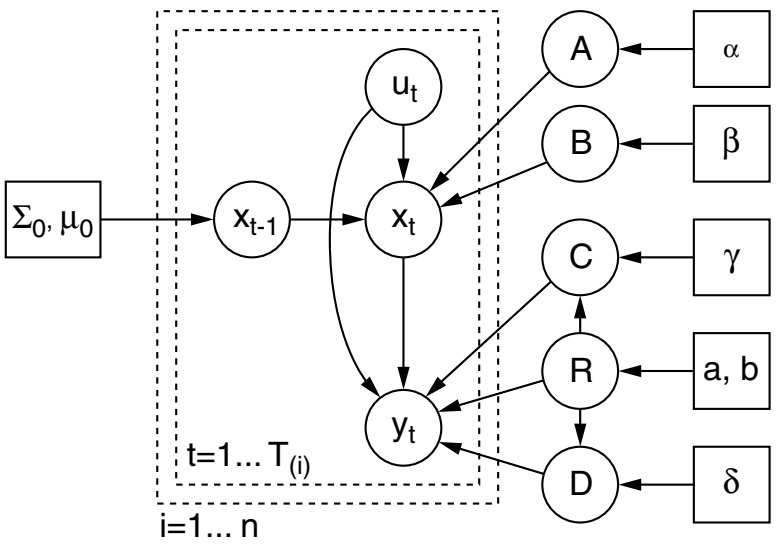
$$
\begin{aligned}
p(\phi) &= p(\theta)\left|\frac{d\theta}{d\phi}\right| \\
p(\theta) &\propto h(\theta)^{1/2} \\
h(\theta) &= -\int p(x|\theta)\,\frac{\partial^2}{\partial\theta^2}\log p(x|\theta)\,dx \qquad \text{(Fisher information)}
\end{aligned}
$$

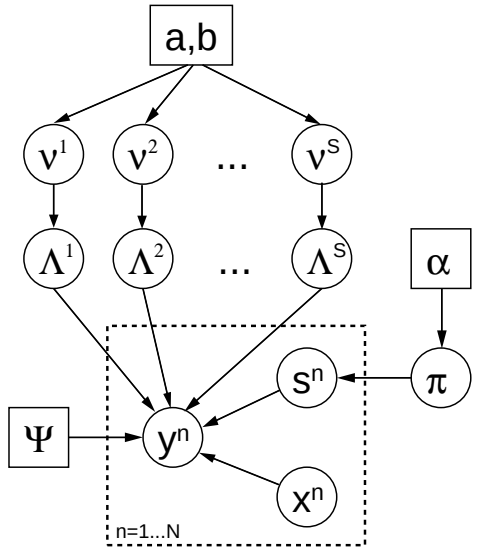
Problems: It is hard (impossible) to generalize to all parameters of a complicated model. Risk of incoherent inferences (e.g. $E_x E_y[Y|X] \neq E_y[Y]$), paradoxes, and improper posteriors.

Hierarchical Priors

$$
\begin{aligned}
p(\theta) &= \int d\alpha\, p(\theta|\alpha)p(\alpha) \\
&= \int d\alpha\, p(\theta|\alpha) \int d\beta\, p(\alpha|\beta)p(\beta) \\
&= \int d\alpha\, p(\theta|\alpha) \int d\beta\, p(\alpha|\beta) \int d\gamma\, p(\beta|\gamma)p(\gamma) \quad \text{(etc...)}
\end{aligned}
$$

In models with many parameters, priors over parameters have hyperparameters. These in turn can also have priors with hyper-hyperparameters, etc.





Exponential Family and Conjugate Priors

$p(x|\theta)$ in the **exponential family** if it can be written as:

$$p(x|\theta) = f(x)g(\theta)\exp\{\phi(\theta)^\top s(x)\}$$

ϕ	vector of *natural parameters*
$s(x)$	vector of *sufficient statistics*
f and g	positive functions of x and θ, respectively.

The **conjugate prior** for this is

$$p(\theta) = h(\eta, \nu)\, g(\theta)^\eta \exp\{\phi(\theta)^\top \nu\}$$

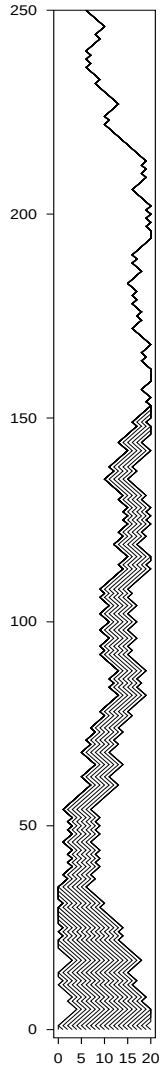
where η and ν are hyperparameters and h is the normalizing function.

The posterior for N data points is also conjugate (by definition), with hyperparameters $\eta + N$ and $\nu + \sum_n s(x_n)$. This is **computationally convenient**.

$$p(\theta|x_1, \ldots, x_N) = h\Big(\eta + N, \nu + \sum_n s(x_n)\Big)\, g(\theta)^{\eta+N} \exp\left\{\phi(\theta)^\top (\nu + \sum_n s(x_n))\right\}$$

Exact Sampling

a.k.a. perfect simulation, coupling from the past



(from MacKay 2003)

- Coupling: running multiple Markov chains (MCs) using the same random seeds. E.g. imagine starting a Markov chain at each possible value of the state (θ).
- Coalescence: if two coupled MCs end up at the same state at time t, then they will forever follow the same path.
- Monotonicity: Rather than running an MC starting from every state, find a partial ordering of the states preserved by the coupled transitions, and track the highest and lowest elements of the partial ordering. When these coalesce, MCs started from all initial states would have coalesced.
- Running from the past: Start at $t = -K$ in the past, if highest and lowest elements of the MC have coalesced by time $t = 0$ then all MCs started at $t = -\infty$ would have coalesced, therefore the chain must be at equilibrium, therefore $\theta_0 \sim p^*(\theta)$.

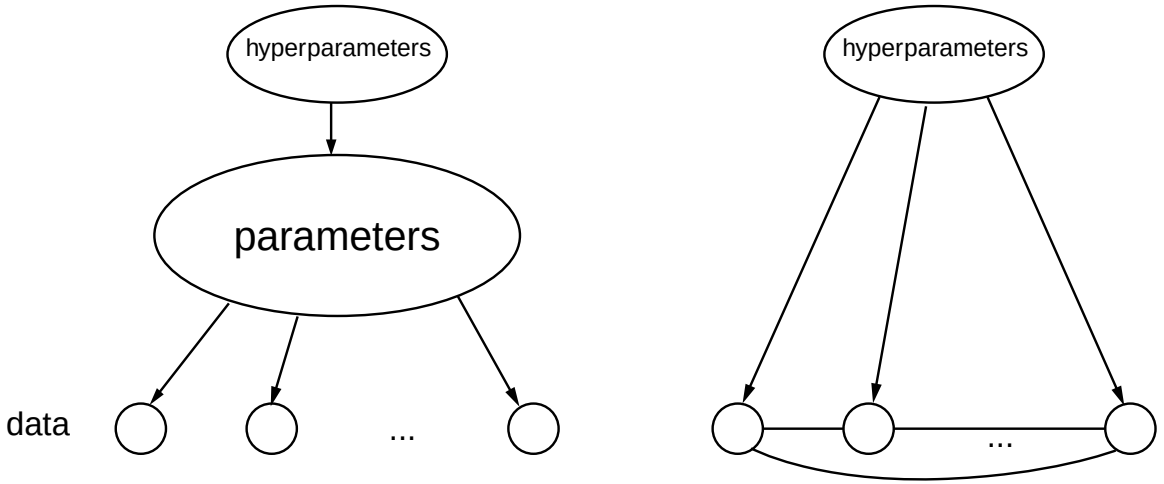
Bottom Line This procedure, *when* it produces a sample, will produce one from the *exact* distribution $p^*(\theta)$.

From Parametric to Nonparametric

Consider data set $\mathcal{D} = \{x^{(1)}, \ldots, x^{(n)}\}$, new data point x, parameters $\boldsymbol{\theta}$, and hyperparameters $\boldsymbol{\alpha}$

By integrating out parameters one can seemingly turn a parametric method into a non-parametric method:

$$
\begin{array}{lcll}
p(x, \mathcal{D}, \boldsymbol{\theta}|\boldsymbol{\alpha}) & = & p(x|\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})p(\theta|\boldsymbol{\alpha}) & \text{(parametric)} \\
p(x|\mathcal{D}, \boldsymbol{\alpha}) & \propto & \int p(x|\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})p(\theta|\boldsymbol{\alpha})\, d\boldsymbol{\theta} & \text{(non-parametric)}
\end{array}
$$



A key question: are a fixed finite number of sufficient statistics of the data needed to make predictions?