

Bayesian Modelling

Zoubin Ghahramani

**Department of Engineering
University of Cambridge, UK**

`zoubin@eng.cam.ac.uk`

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

MLSS 2012

La Palma

An Information Revolution?

- We are in an era of abundant data:
 - **Society:** the web, social networks, mobile networks, government, digital archives
 - **Science:** large-scale scientific experiments, biomedical data, climate data, scientific literature
 - **Business:** e-commerce, electronic trading, advertising, personalisation
- We need tools for modelling, searching, visualising, and understanding large data sets.

Modelling Tools

Our modelling tools should:

- Faithfully represent **uncertainty** in our model structure and parameters and **noise** in our data
- Be automated and **adaptive**
- Exhibit **robustness**
- **Scale well** to large data sets

Probabilistic Modelling

- A model describes data that one could observe from a system
- If we use the mathematics of probability theory to express all forms of uncertainty and noise associated with our model...
- ...then *inverse probability* (i.e. Bayes rule) allows us to infer unknown quantities, adapt our models, make predictions and learn from data.

Bayes Rule

$$P(\text{hypothesis}|\text{data}) = \frac{P(\text{data}|\text{hypothesis})P(\text{hypothesis})}{P(\text{data})}$$



Rev'd Thomas Bayes (1702–1761)

- Bayes rule tells us how to do inference about hypotheses from data.
- Learning and prediction can be seen as forms of inference.

Modeling vs toolbox views of Machine Learning

- **Machine Learning seeks to learn models of data:** define a space of possible models; learn the parameters and structure of the models from data; make predictions and decisions
- **Machine Learning is a toolbox of methods for processing data:** feed the data into one of many possible methods; choose methods that have good theoretical or empirical performance; make predictions and decisions

Plan

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

Detailed Plan [Some parts will be skipped]

- **Introduce Foundations**
 - Some canonical problems: classification, regression, density estimation
 - 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
- **Limitations and Discussion**
 - Reconciling Bayesian and Frequentist Views
 - Limitations and Criticisms of Bayesian Methods
 - Discussion

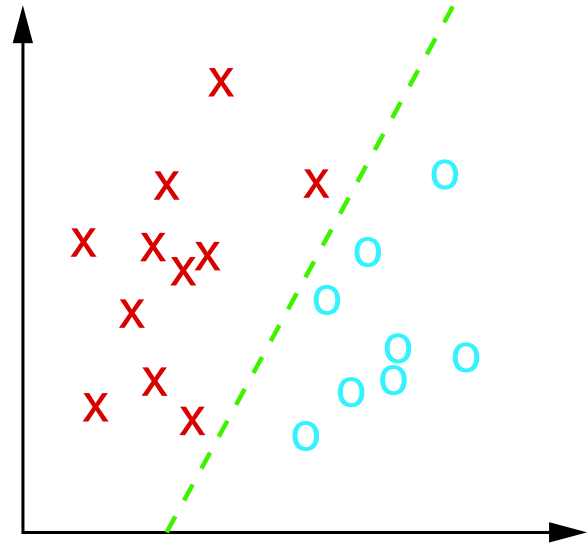
Some Canonical Machine Learning Problems

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

Linear Classification

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

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



Model:

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

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

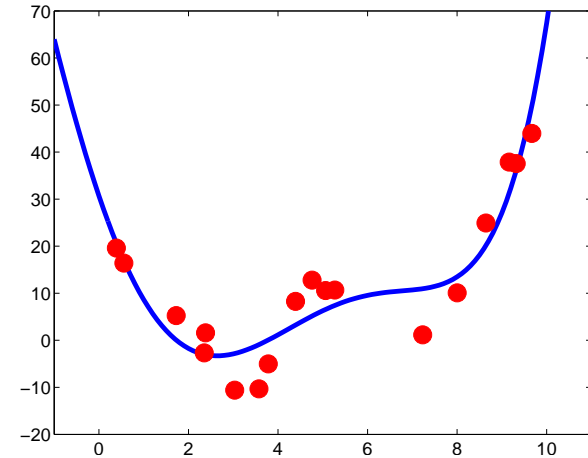
Goal: To infer $\boldsymbol{\theta}$ from the data and to predict future labels $P(y | \mathcal{D}, \mathbf{x})$

Polynomial Regression

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

$$x^{(n)} \in \mathcal{R}$$

$$y^{(n)} \in \mathcal{R}$$



Model:

$$y^{(n)} = a_0 + a_1x^{(n)} + a_2x^{(n)2} \dots + a_mx^{(n)m} + \epsilon$$

where

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

Parameters: $\theta = (a_0, \dots, a_m, \sigma)$

Goal: To infer θ 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 $n = 1, \dots, N$

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

Model:

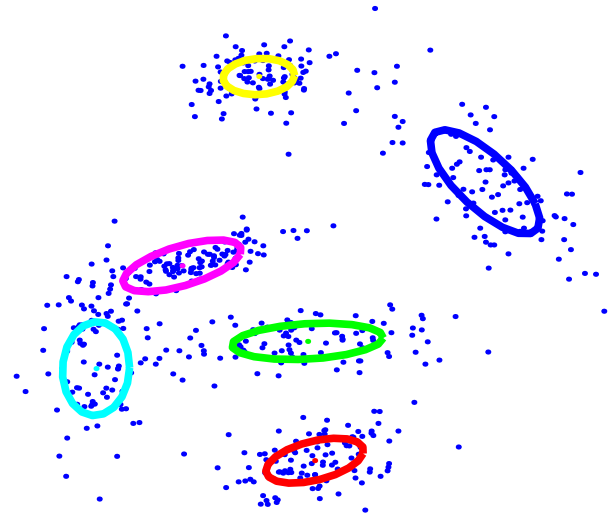
$$\mathbf{x}^{(n)} \sim \sum_{i=1}^m \pi_i p_i(\mathbf{x}^{(n)})$$

where

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

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

Goal: To infer θ from the data, predict the density $p(\mathbf{x}|\mathcal{D}, m)$, and infer which points belong to the same cluster.



Bayesian Machine Learning

Everything follows from two simple rules:

Sum rule: $P(x) = \sum_y P(x, y)$

Product rule: $P(x, y) = P(x)P(y|x)$

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

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

Prediction:

$$P(x|\mathcal{D}, m) = \int P(x|\theta, \mathcal{D}, m)P(\theta|\mathcal{D}, m)d\theta$$

Model Comparison:

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

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

That's it!

Questions

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

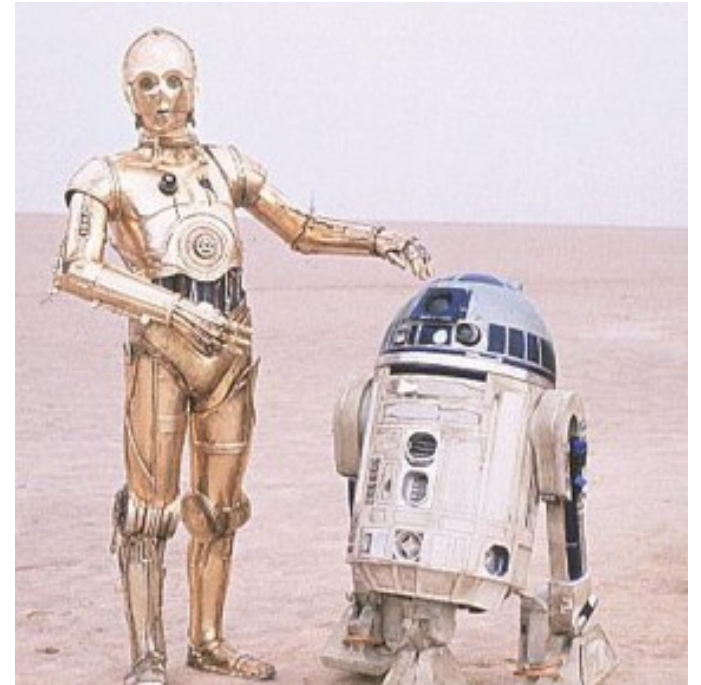
Representing Beliefs (Artificial Intelligence)

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 mathematical rules 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 .

$$0 \leq b(x) \leq 1$$

$$b(x) = 0 \quad x \text{ is definitely **not true**}$$

$$b(x) = 1 \quad x \text{ is definitely **true**}$$

$$b(x|y) \quad \text{strength of belief that } x \text{ is true given that we know } y \text{ is true}$$

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 several ways, then each way should lead to the same answer.
 - The robot must always take 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 sum rule, product rule and therefore Bayes rule.

(Cox 1946; Jaynes, 1996; van Horn, 2003)

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:

$$\left\{ \begin{array}{ll} x \text{ is true} & \text{win } \geq \$1 \\ x \text{ is false} & \text{lose } \$9 \end{array} \right.$$

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 θ^* , then under some regularity conditions, as long as $p(\theta^*) > 0$

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

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

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

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

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

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} .

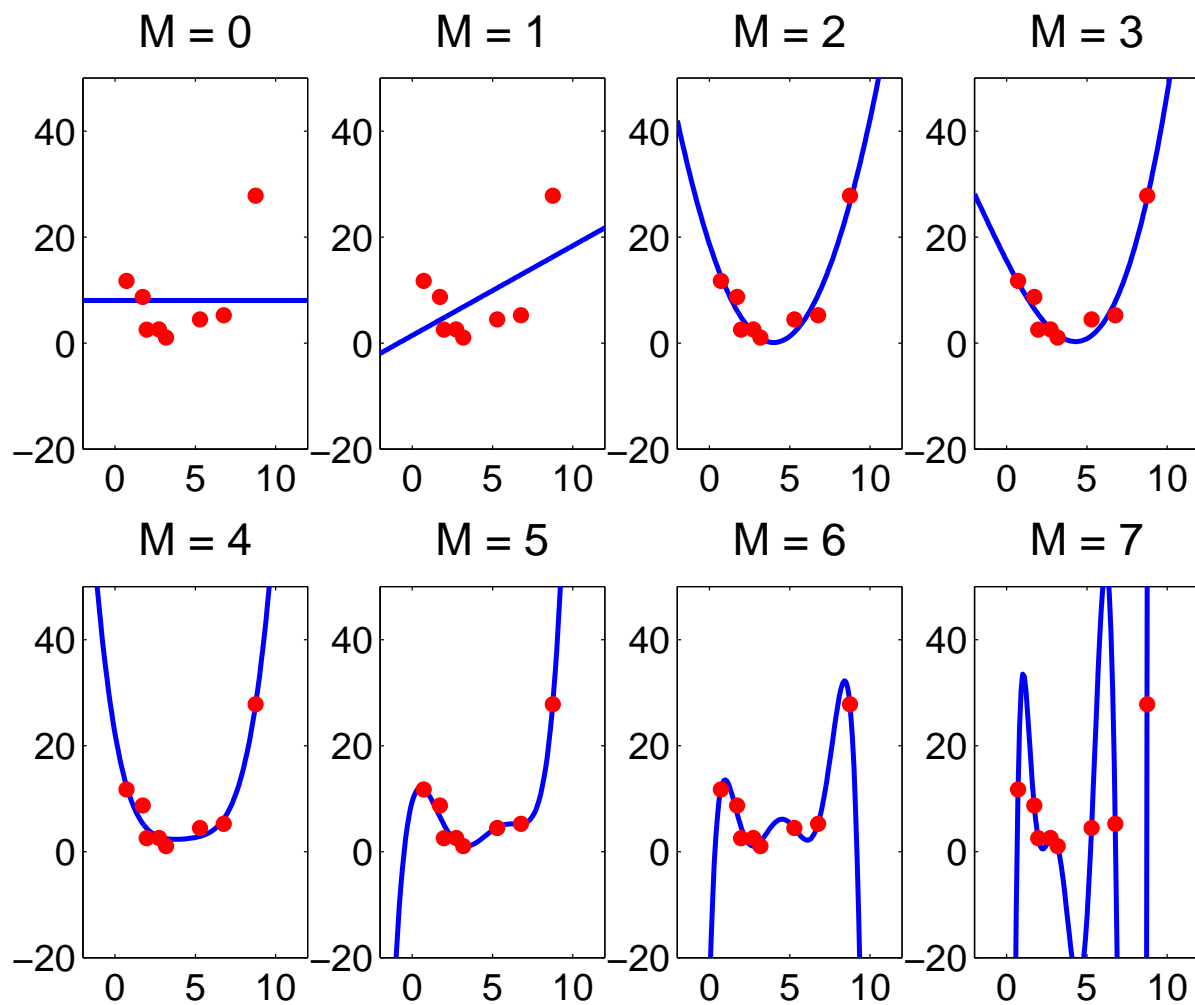
Assume both Bayesians agree on the set of possible and impossible values of θ :

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

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

coin toss demo: bayescoin

Model Selection



Bayesian Occam's Razor and Model Selection

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

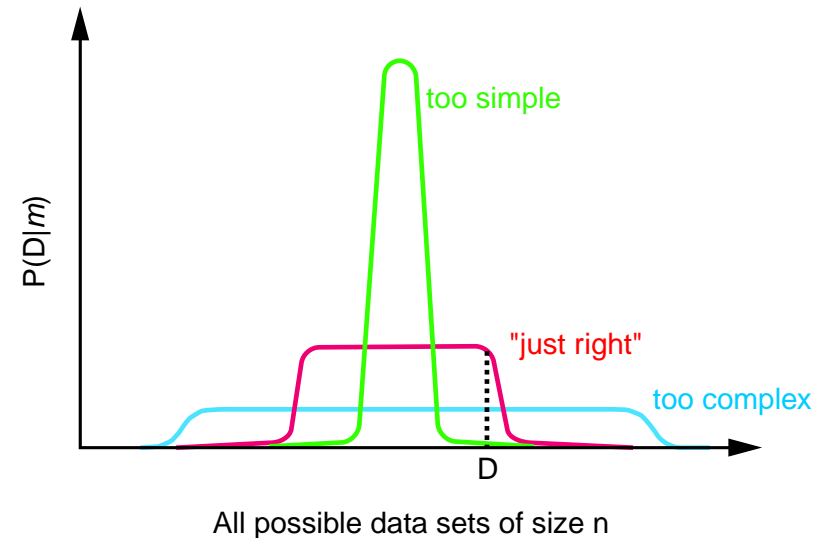
$$p(m|\mathcal{D}) = \frac{p(\mathcal{D}|m) p(m)}{p(\mathcal{D})}, \quad p(\mathcal{D}|m) = \int p(\mathcal{D}|\theta, m) p(\theta|m) d\theta$$

Interpretations of the **Marginal Likelihood** (“model evidence”):

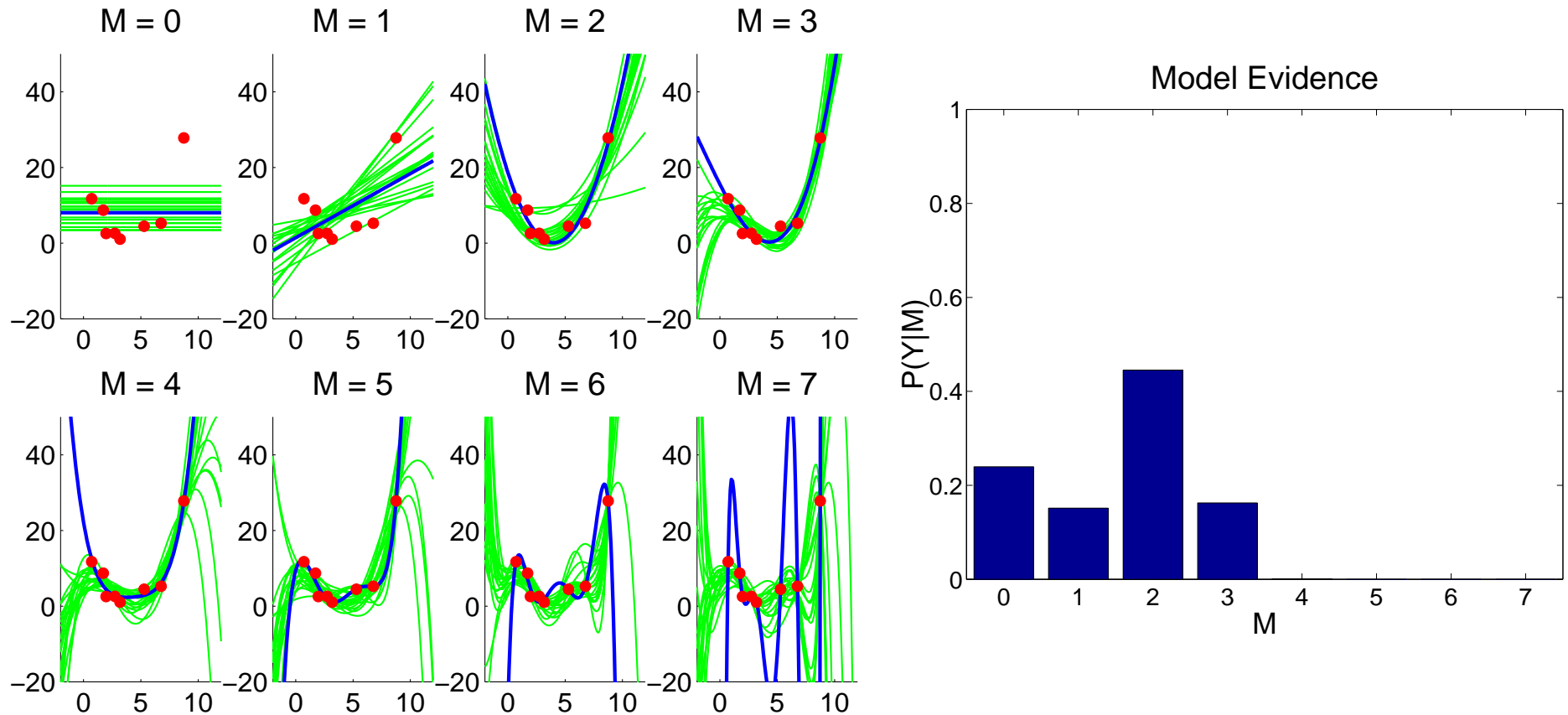
- The probability that *randomly selected* parameters from the prior would generate \mathcal{D} .
- Probability of the data under the model, *averaging* over all possible parameter values.
- $\log_2 \left(\frac{1}{p(\mathcal{D}|m)} \right)$ is the number of *bits of surprise* at observing data \mathcal{D}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



Bayesian Model Selection: Occam's Razor at Work



For example, for quadratic polynomials ($m = 2$): $y = a_0 + a_1x + a_2x^2 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and parameters $\theta = (a_0 \ a_1 \ a_2 \ \sigma)$

```
demo: polybayes  
demo: run_simple
```

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:

$$\begin{aligned} p(\theta) &= \int d\alpha p(\theta|\alpha)p(\alpha) \\ &= \int d\alpha p(\theta|\alpha) \int d\beta p(\alpha|\beta)p(\beta) \end{aligned}$$

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

Subjective Priors

Priors should capture our beliefs as well as possible.

Otherwise we are not coherent.

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 priors beliefs can be useful, since the data will concentrate the posterior around reasonable models.

The key ingredient of Bayesian methods is not the prior, it's the idea of averaging over different possibilities.

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

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

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 data / overfitting.

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**.

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

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:

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

Prediction:

$$P(x|\mathcal{D}, m) = \int P(x|\theta, \mathcal{D}, m)P(\theta|\mathcal{D}, m)d\theta$$
$$P(x|\mathcal{D}, m) = \int P(x|\theta)P(\theta|\mathcal{D}, m)d\theta \quad (\text{for many models})$$

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', \dots , parameter $\boldsymbol{\theta}, \boldsymbol{\theta}' \dots$

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 $A_{ij} = - \left. \frac{d^2}{d\theta_i d\theta_j} \ln p(\boldsymbol{\theta}|\mathbf{y}, m) \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$

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

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 ($n \rightarrow \infty$) 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 $n \rightarrow \infty$, 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:

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

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

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 $n \rightarrow \infty$, 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`

Further Topics

- Bayesian Discriminative Learning (BPM vs SVM)
- From Parametric to Nonparametric Methods
 - Gaussian Processes
 - Dirichlet Process Mixtures
- Feature Selection and ARD

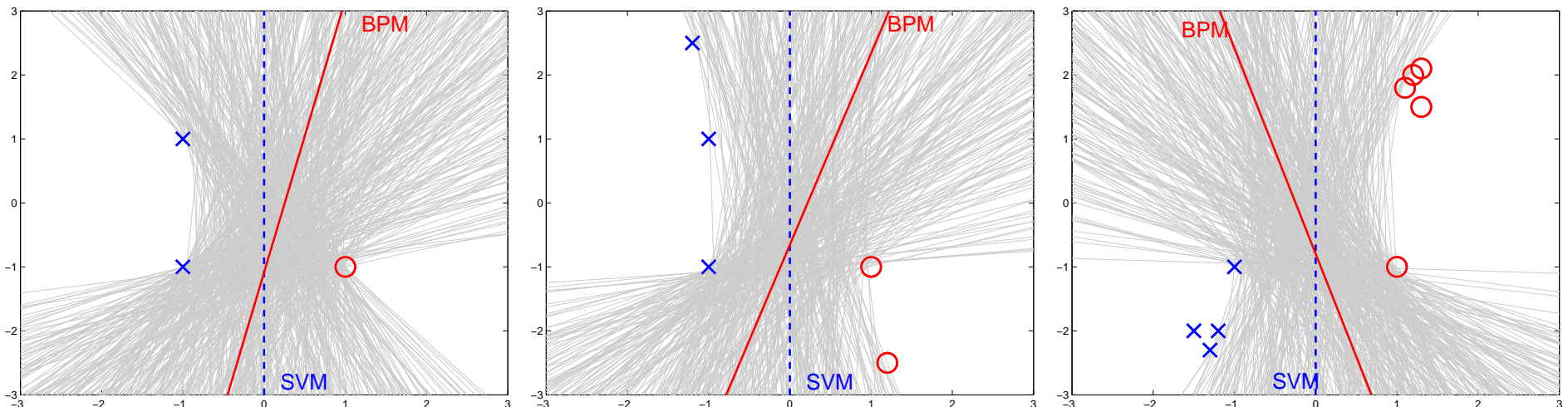
Bayesian Discriminative Modeling

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

- *Parametric models* assume some **finite set of parameters** θ . Given the parameters, future predictions, x , are independent of the observed data, \mathcal{D} :

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

therefore θ capture everything there is to know about the data.

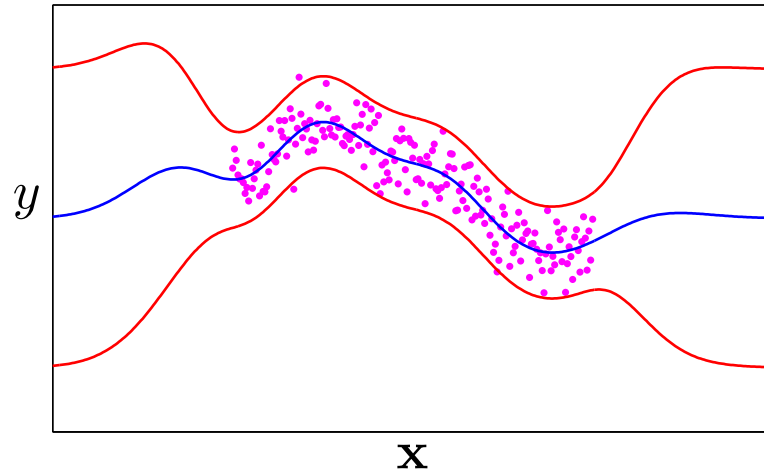
- So the complexity of the model is bounded even if the amount of data is unbounded. This makes them not very flexible.

-
- *Non-parametric models* assume that the data distribution cannot be defined in terms of such a finite set of parameters. But they can often be defined by assuming an **infinite dimensional** θ . Usually we think of θ as a **function**.
 - The amount of information that θ can capture about the data \mathcal{D} can grow as the amount of data grows. This makes them more flexible.
-

Nonlinear regression and Gaussian processes

Consider the problem of **nonlinear regression**:

You want to learn a **function f** with **error bars** from **data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$**



A **Gaussian process** defines a distribution over functions $p(f)$ which can be used for Bayesian regression:

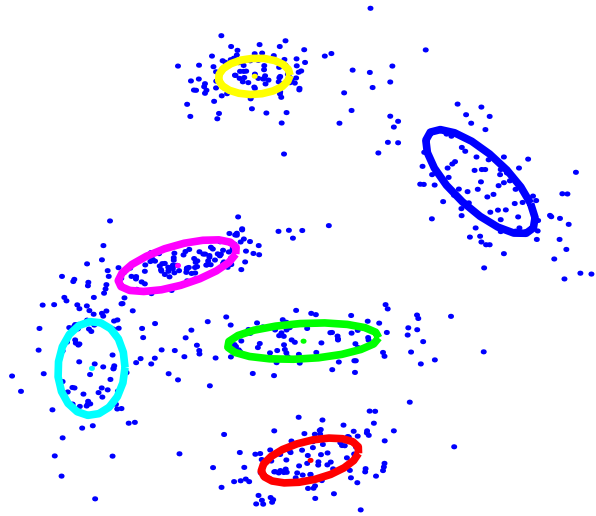
$$p(f|\mathcal{D}) = \frac{p(f)p(\mathcal{D}|f)}{p(\mathcal{D})}$$

Let $\mathbf{f} = (f(x_1), f(x_2), \dots, f(x_n))$ be an n -dimensional vector of function values evaluated at n points $x_i \in \mathcal{X}$. Note, \mathbf{f} is a random variable.

Definition: $p(f)$ is a **Gaussian process** if for *any* finite subset $\{x_1, \dots, x_n\} \subset \mathcal{X}$, the marginal distribution over that subset $p(\mathbf{f})$ is multivariate Gaussian.

Clustering

Basic idea: each data point belongs to a cluster

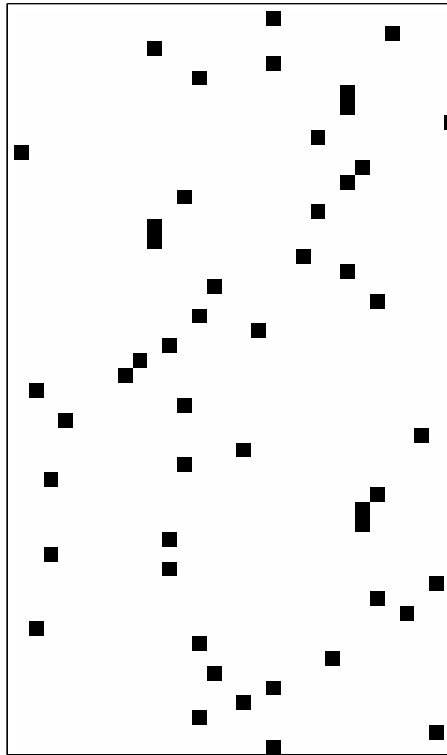


Many clustering methods exist:

- mixture models
- hierarchical clustering
- spectral clustering

Goal: to partition data into groups in an unsupervised manner

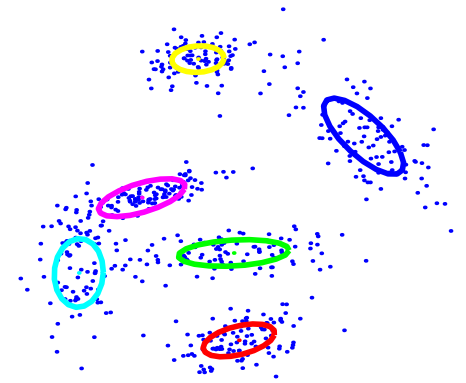
A binary matrix representation for clustering



- Rows are data points
- Columns are clusters
- Since each data point is assigned to one and only one cluster, the rows sum to one.
- **Finite mixture models:** number of columns is finite
- **Infinite mixture models:** number of columns is countably infinite

Infinite mixture models

(e.g. Dirichlet Process Mixtures)



Why?

- You might not believe a priori that your data comes from a finite number of mixture components (e.g. strangely shaped clusters; heavy tails; structure at many resolutions)
- Inflexible models (e.g. a mixture of 6 Gaussians) can yield unreasonable inferences and predictions.
- For many kinds of data, the number of clusters might grow over time: clusters of news stories or emails, classes of objects, etc.
- You might want your method to automatically infer the number of clusters in the data.

Infinite mixture models

$$p(x) = \sum_{k=1}^K \pi_k p_k(x)$$

How?

- Start from a finite mixture model with K components and take the limit¹ as number of components $K \rightarrow \infty$
- But you have infinitely many parameters!
- Rather than optimize the parameters (ML, MAP), you integrate them out (Bayes) using, e.g:
 - MCMC sampling (Escobar & West 1995; Neal 2000; Rasmussen 2000)
 - expectation propagation (EP; Minka and Ghahramani, 2003)
 - variational methods (Blei and Jordan, 2005)
 - Bayesian hierarchical clustering (Heller and Ghahramani, 2005)

¹Dirichlet Process Mixtures; Chinese Restaurant Processes

Discussion

Myths and misconceptions about Bayesian methods

- **Bayesian methods make assumptions where other methods don't**

All methods make assumptions! Otherwise it's impossible to predict. Bayesian methods are transparent in their assumptions whereas other methods are often opaque.

- **If you don't have the right prior you won't do well**

Certainly a poor model will predict poorly but there is no such thing as the right prior! Your model (both prior and likelihood) should capture a reasonable range of possibilities. When in doubt you can choose vague priors (cf nonparametrics).

- **Maximum A Posteriori (MAP) is a Bayesian method**

MAP is similar to regularization and offers no particular Bayesian advantages. The key ingredient in Bayesian methods is to average over your uncertain variables and parameters, rather than to optimize.

Myths and misconceptions about Bayesian methods

- **Bayesian methods don't have theoretical guarantees**

One can often apply frequentist style generalization error bounds to Bayesian methods (e.g. PAC-Bayes). Moreover, it is often possible to prove convergence, consistency and rates for Bayesian methods.

- **Bayesian methods are generative**

You can use Bayesian approaches for both generative and discriminative learning (e.g. Gaussian process classification).

- **Bayesian methods don't scale well**

With the right inference methods (variational, MCMC) it is possible to scale to very large datasets (e.g. excellent results for Bayesian Probabilistic Matrix Factorization on the Netflix dataset using MCMC), but it's true that averaging/integration is often more expensive than optimization.

Reconciling Bayesian and Frequentist Views

Frequentist theory tends to focus on **sampling properties** of estimators, i.e. what would have happened had we observed other data sets from our model. Also look at **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 generally 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 he or she must determine that not only does this inference follow 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 (sampling / 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, see PAC-Bayesian frequentist bounds on Bayesian procedures.

Cons and pros of Bayesian methods

Limitations and Criticisms:

- 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.

Advantages:

- Coherent.
- Conceptually straightforward.
- Modular.
- Often good performance.

Summary

- Bayesian machine learning treats learning as a probabilistic inference problem.
- Bayesian methods work well when the models are flexible enough to capture relevant properties of the data.
- This motivates non-parametric Bayesian methods, e.g.:
 - Gaussian processes for regression.
 - Dirichlet process mixtures for clustering.

Thanks for your patience!

Appendix

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.

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:

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

This can be generalized to experiments with n observations (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.

$$p(\phi) = p(\theta) \left| \frac{d\theta}{d\phi} \right|$$

$$p(\theta) \propto h(\theta)^{1/2}$$

$$h(\theta) = - \int p(x|\theta) \frac{\partial^2}{\partial \theta^2} \log p(x|\theta) dx \quad (\text{Fisher information})$$

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.

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: $q(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \prod_{i=0}^N \tilde{f}_i(\boldsymbol{\theta})$

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

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

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

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:  $q_{\setminus i}(\boldsymbol{\theta}) \leftarrow \frac{q(\boldsymbol{\theta})}{\tilde{f}_i(\boldsymbol{\theta})} = \prod_{j \neq i} \tilde{f}_j(\boldsymbol{\theta})$   
    Projection:  $\tilde{f}_i^{\text{new}}(\boldsymbol{\theta}) \leftarrow \arg \min_{f(\boldsymbol{\theta})} \text{KL}(f_i(\boldsymbol{\theta})q_{\setminus i}(\boldsymbol{\theta}) \| f(\boldsymbol{\theta})q_{\setminus i}(\boldsymbol{\theta}))$   
    Inclusion:  $q(\boldsymbol{\theta}) \leftarrow \tilde{f}_i^{\text{new}}(\boldsymbol{\theta}) q_{\setminus i}(\boldsymbol{\theta})$   
  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:

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

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 $\theta_0 \rightarrow \theta_1 \rightarrow \theta_2 \rightarrow \theta_3 \rightarrow \dots$ 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' . A useful condition that implies invariance of $p^*(\theta)$ is **detailed balance**:

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

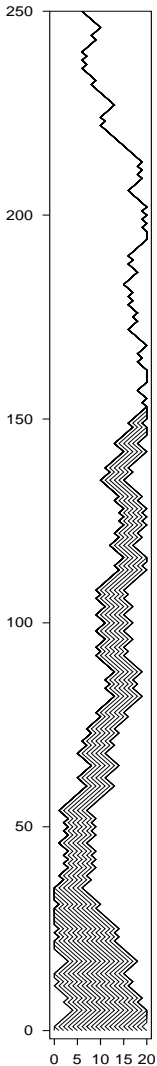
MCMC methods define **ergodic** Markov chains, which converge to a unique stationary distribution (also called an *equilibrium distribution*) regardless of the initial conditions $p_0(\theta)$:

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

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

Exact Sampling

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



- **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 MacKay 2003)

Feature Selection

Example: classification

$$\begin{aligned} \text{input } \mathbf{x} &= (x_1, \dots, x_D) \in \mathbb{R}^D \\ \text{output } y &\in \{+1, -1\} \end{aligned}$$

2^D possible subsets of relevant input features.

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

$$\hat{m} = \operatorname{argmax}_m p(\mathcal{D}|m)$$

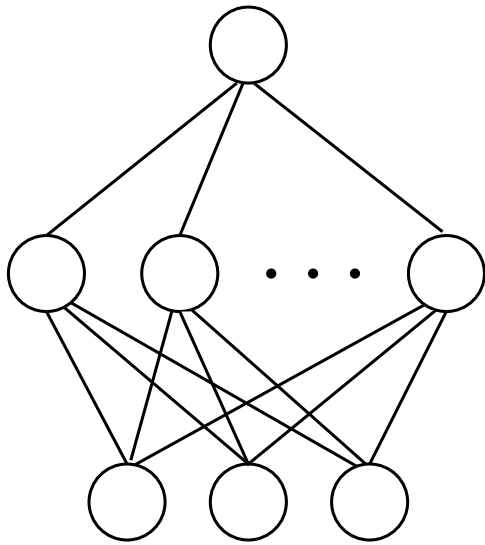
Problems: intractable, overfitting, we should really average

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^{-1} : $p(w_{dj}|\alpha_d) = \mathcal{N}(0, \alpha^{-1})$

Let's think about this:

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

ARD: 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.

Two views of machine learning

- The goal of machine learning is to produce general purpose **black-box algorithms** for learning. I should be able to put my algorithm online, so lots of people can download it. If people want to apply it to problems A, B, C, D... then it should work regardless of the problem, and the user should not have to think too much.
- If I want to solve problem A it seems silly to use some general purpose method that was never designed for A. I should really try to understand what problem A is, learn about the properties of the data, and use as much **expert knowledge** as I can. Only then should I think of designing a method to solve A.



"I know nothing about the subject,
but I'm happy to give you my expert opinion."