Bayesian Machine Learning Overview

Zoubin Ghahramani

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

zoubin@cs.cmu.edu

Statistical Machine Learning
CMU 10-702 / 36-702
Spring 2008

Some Canonical Problems

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

Coin Toss

Data: $\mathcal{D} = (H\,T\,H\,H\,H\,T\,T\,\ldots)$

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

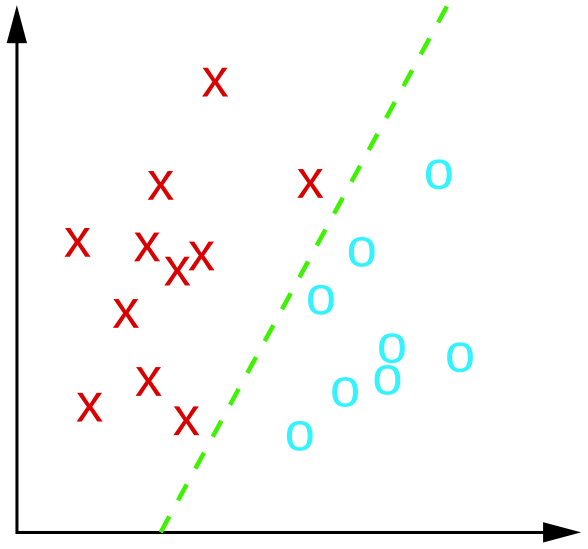
$$P(H|\theta) = \theta$$
$$P(T|\theta) = 1 - \theta$$

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

Linear Classification

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

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



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

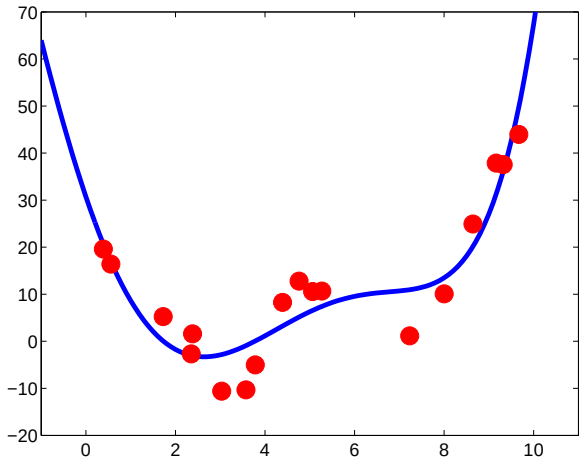
$$P(y^{(n)} = +1 | \boldsymbol{\theta}, \mathbf{x}^{(n)}) = \begin{cases} 1 & \text{if } \sum_{d=1}^{D} \theta_d \, x_d^{(n)} + \theta_0 \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

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 $n = 1, \ldots, N$

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



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

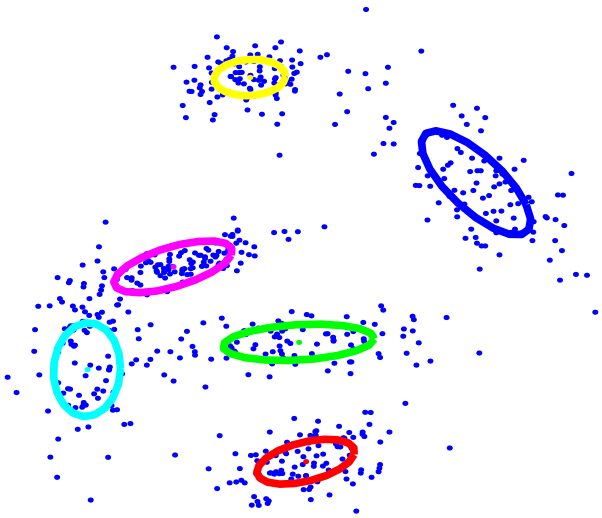
$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

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 $n = 1, \ldots, N$

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



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:

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

That's it.

Questions

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

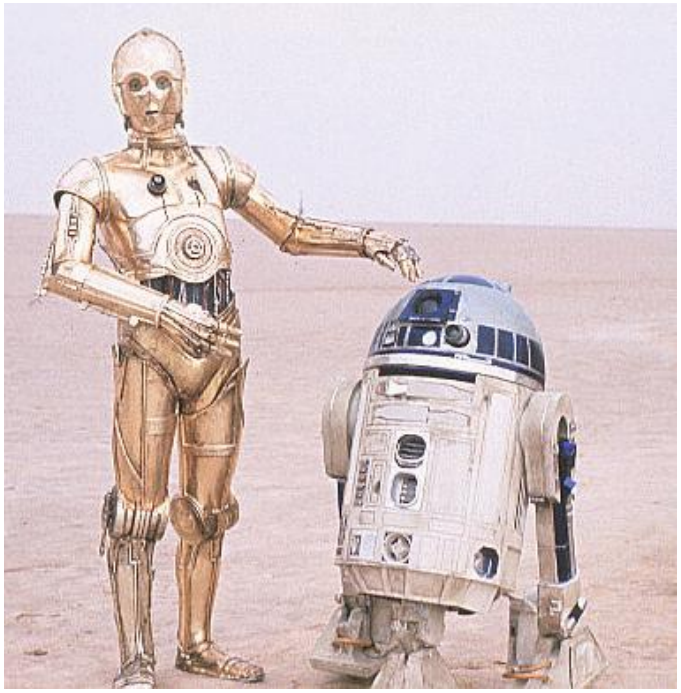
Representing Beliefs (Artificial Intelligence)

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 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.

(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:

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

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 model with true finite-dimensional 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 in the model class, 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

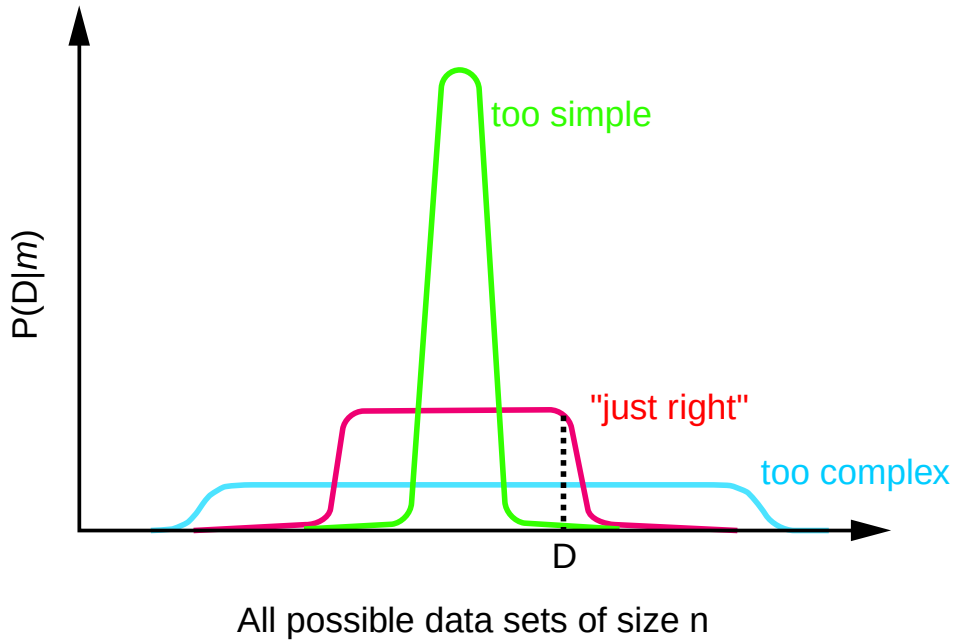
Compare model classes, e.g. m and m', using posterior probabilities given $\mathcal{D}$:

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

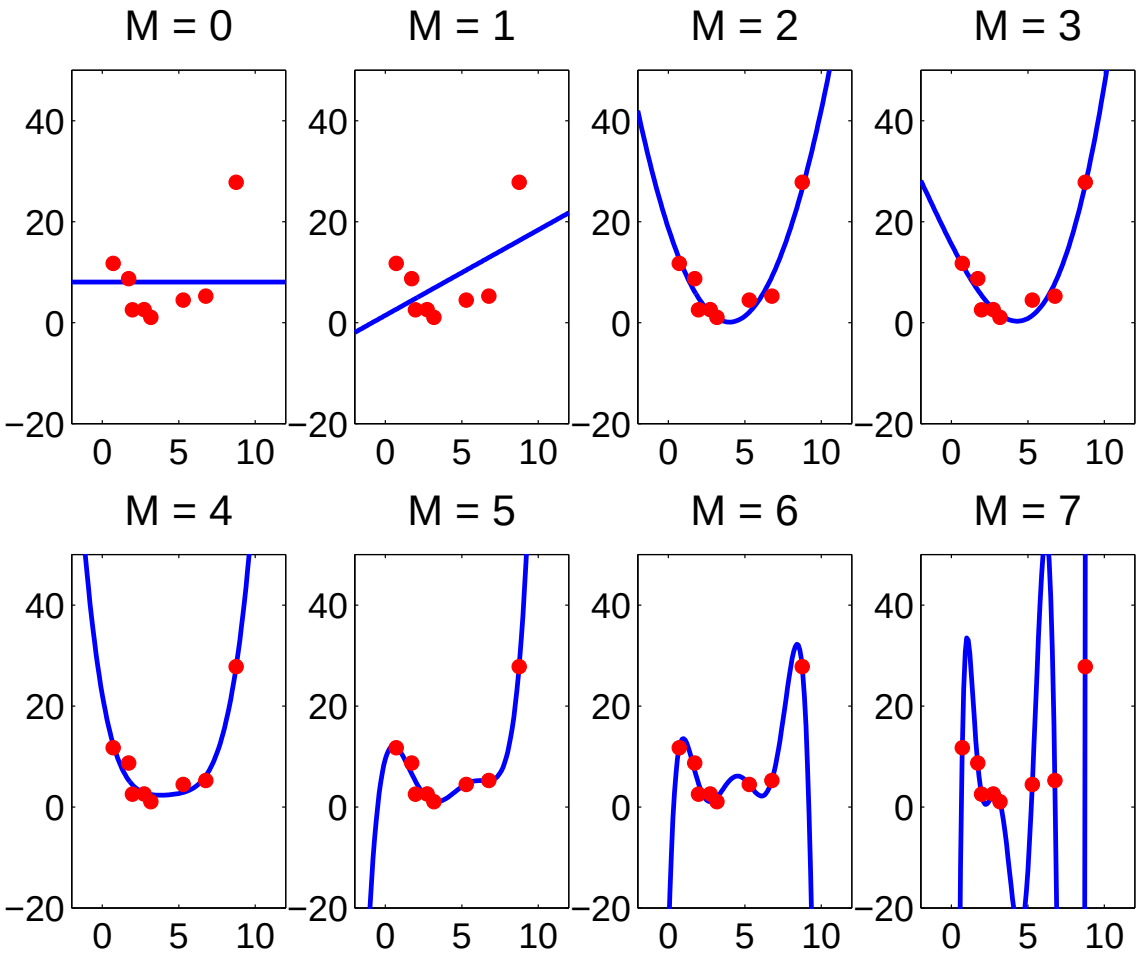
Interpretation of the Marginal Likelihood ("evidence"): The probability that *randomly selected* parameters from the prior would generate $\mathcal{D}$.

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

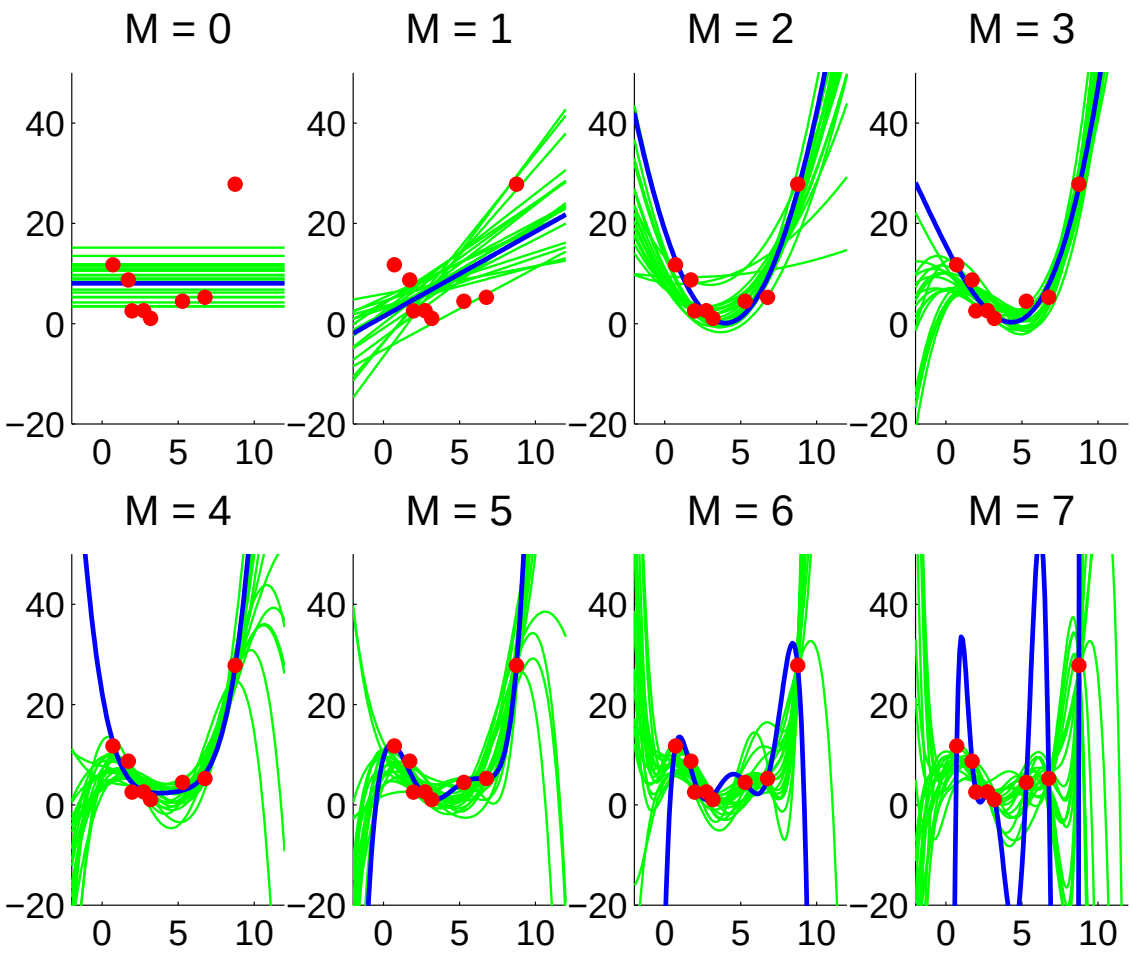
Model classes that are too complex can generate many possible data sets, so again, they are unlikely to generate that particular data set at random.

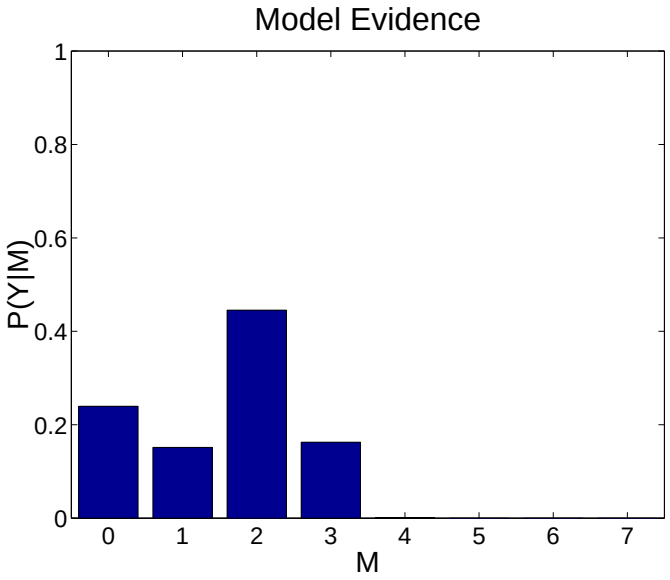


Model structure and overfitting:
A simple example: polynomial regression



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")

Subjective Priors

Priors should capture our 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.

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**.

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

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:

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

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

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

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:

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

in the large sample limit ($n \to \infty$) where n is the number of data points, A grows as nA_0 for some fixed matrix A_0, so $\ln|A| \to \ln|nA_0| = \ln(n^d|A_0|) = d\ln n + \ln|A_0|$. Retaining only terms that grow in n we get:

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

Properties:

- Quick and easy to compute; does not depend on the prior.
- We can use the ML estimate of θ instead of the MAP estimate
- 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)
- It is equivalent to the "Minimum Description Length" (MDL) criterion
- **Danger:** counting parameters can be deceptive! (c.f. sinusoid)

Other Topics

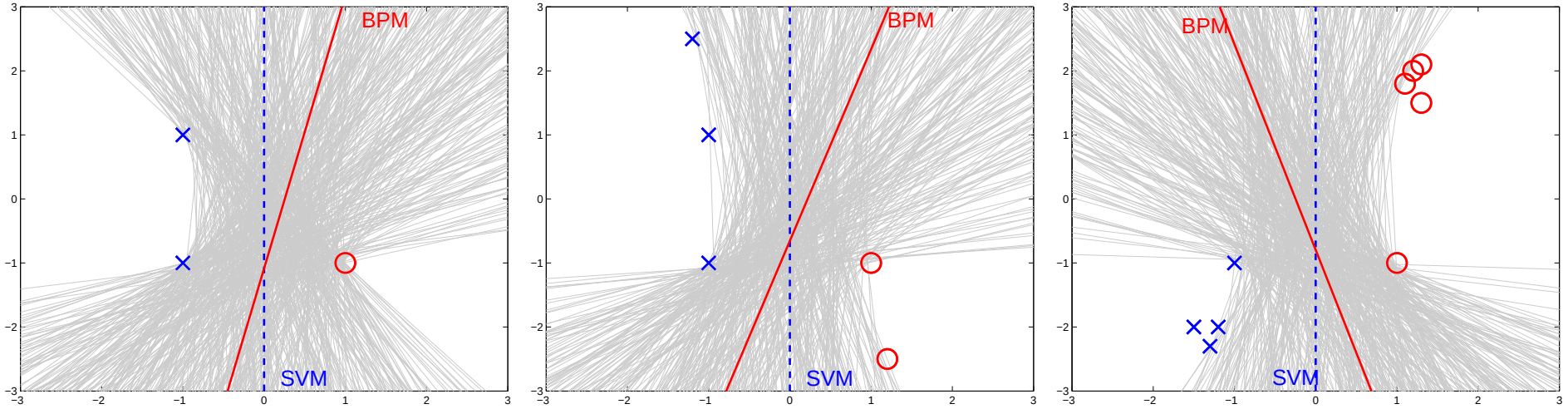
Bayesian Discriminative Modeling

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)

We can often get non-parametric models by considering the infinite-parameter limit of parametric models!

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

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.

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.

Appendix: Cox's Axioms

R1: $(A|X)$, the plausibility of A given X is true, is a single real number. There exists a real number T such that $(A|X) \leq T$ for every X and A.

R2: Plausibility assignments are compatible with the propositional calculus:

- If A is equivalent to A' then $(A|X) = (A'|X)$.
- If A is a tautology then $(A|X) = T$.
- $(A|B, C, X) = (A|(B\&C), X)$
- If X is consistent and $(not(A)|X) < T$ then $A\&X$ is consistent.

R3: There exists a non-decreasing function S_0 such that $(not(A)|X) = S_0(A|X)$ for all A and consistent X.

R4: There exists a non-empty set of real numbers P_0 with the following two properties:

- P_0 is a dense subset of (F, T).
- For every $y_1, y_2, y_3 \in P_0$ there exists some consistent X with a basis of at least three atomic propositions A_1, A_2, A_3 such that $(A_1|X) = y_1$, $(A_2|A_1, X) = y_2$ and $(A_3|A_2, A_1, X) = y_3$.

where $F = S_0(T)$ (see Kevin van Horn, 2003)