Should all Machine Learning be Bayesian?
Should all Bayesian models be non-parametric?

Zoubin Ghahramani

Department of Engineering
University of Cambridge, UK

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

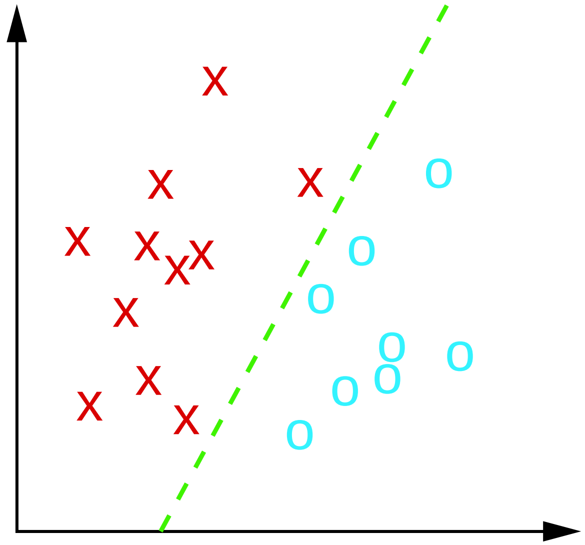
BARK 2008

Some Canonical Machine Learning Problems

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

Example: 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 $P(y|\mathcal{D}, \mathbf{x})$

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

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

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:

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

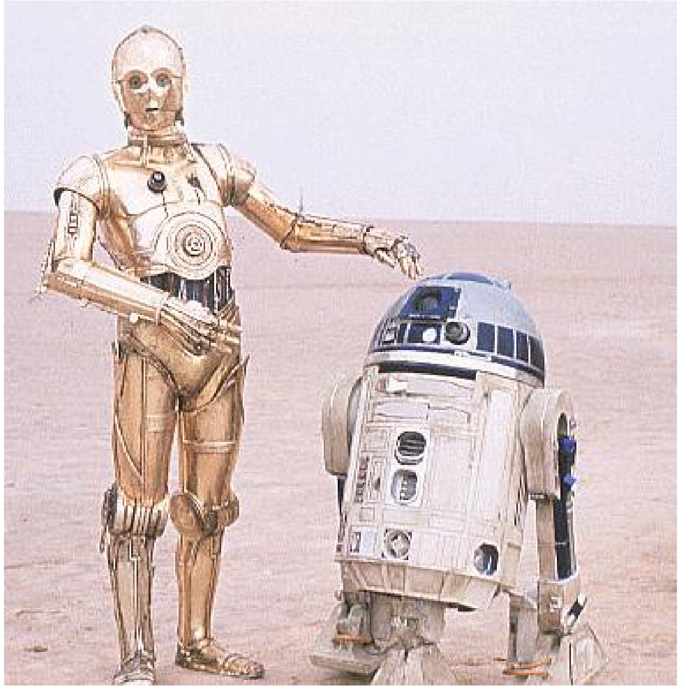
That’s it!

Should all Machine Learning be Bayesian?

- 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:

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

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 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

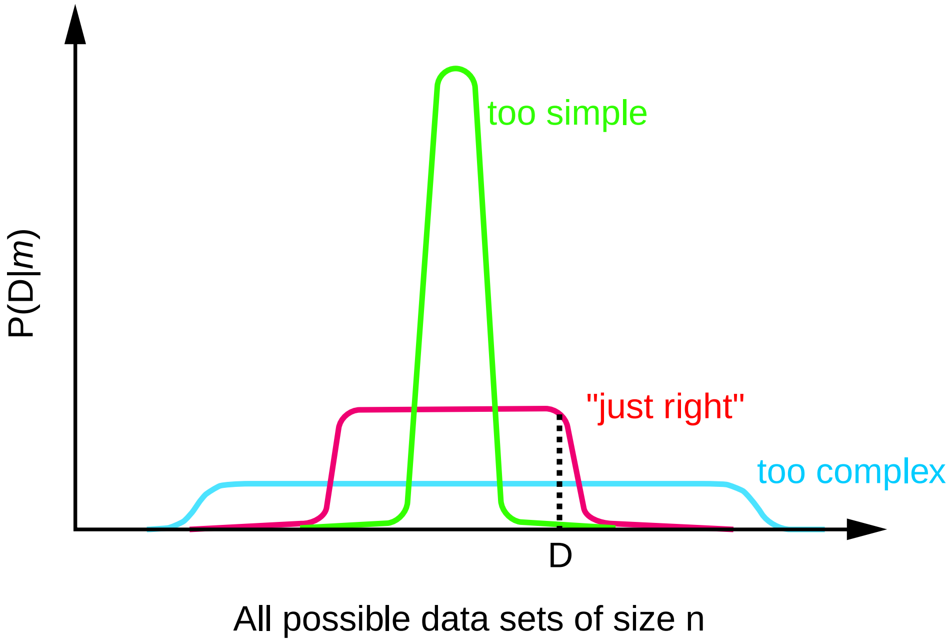
Compare model classes, e.g. m and m', using posterior probabilities given $\mathcal{D}$:

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

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.



Potential advantages of Bayesian Machine Learning over alternatives

- tries to be coherent and honest about uncertainty
- easy to do model comparison, selection
- rational process for model building and adding domain knowledge
- easy to handle missing and hidden data

Disadvantages: to be discussed later :-)

Where does the prior come from?

- **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 out 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 beliefs can be useful.

Two views of machine learning

- The Black Box View
  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.

- The Case Study View
  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.

Bayesian Black Boxes?

Can we meaningfully create Bayesian black-boxes?
If so, what should the prior be?
This seems strange... clearly we can create black boxes, but how can we advocate people blindly using them?
Do we require every practitioner to be a well-trained Bayesian statistician?

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)

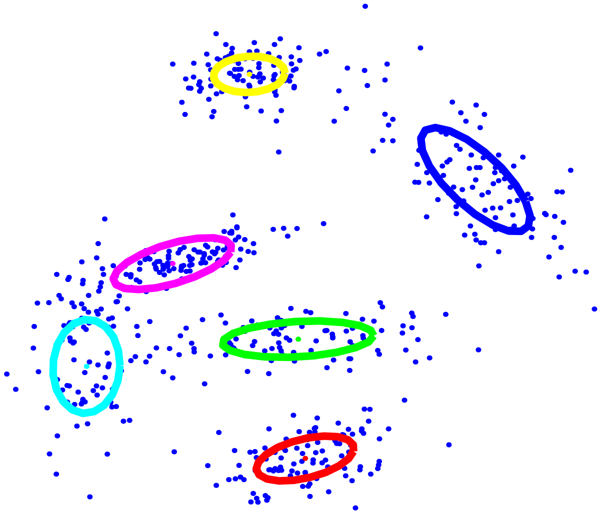
- **Non-parametric Models** allow the number of "parameters" to grow with the data set size, or alternatively we can think of the predictions as depending on the data, and possible a usually small number of parameters $\boldsymbol{\alpha}$

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

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

Example: Clustering

Basic idea: each data point belongs to a cluster



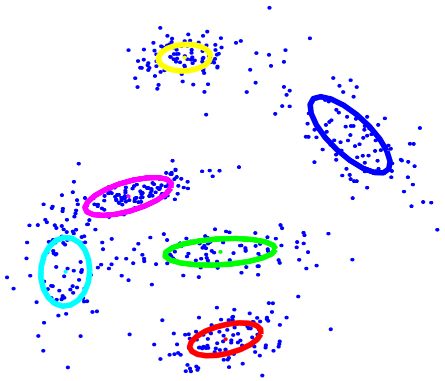
Many clustering methods exist:

- mixture models
- hierarchical clustering
- spectral clustering

Goal: to partition data into groups in an unsupervised manner

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.

Is non-parametrics the only way to go?

- When do we *really* believe our parametric model?

- But, when do we really believe or non-parametric model?

- Is a non-parametric model (e.g. a DPM) really better than a large parametric model (e.g. a mixture of 100 components)?

The Approximate Inference Conundrum

- All interesting models are intractable.
- So we use approximate inference (MCMC, VB, EP, etc).
- Since we often can't control the effect of using approximate inference, are coherence arguments meaningless?
- Is Subjective Bayesianism pointless?

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.

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.

How can we convert the pagan majority of ML researchers to Bayesianism?

Some suggestions:

- Killer apps
- Win more competitions
- Dispel myths, be more proactive, aggressive
- Release good easy to use black-box code
- ??