Lecture 1: Machine Learning Paradigms

Advanced Topics in Machine Learning

Dr. Tom Rainforth

January 22nd, 2020

rainforth@stats.ox.ac.uk

Course Outline

- Slightly unusual course covering different topics in machine learning
- Aim is to get you interacting with actual research
- Fully assessed by coursework
- There are no examples sheets: you are instead expected to take the initiative to investigate areas you find interesting and familiarize yourself will software tools (we will suggest resources and the practicals are there to help with software familiarity)

Course Structure

- 6 lectures on Bayesian Machine Learning from me
- 8 lectures on Natural Language Processing from Dr Alejo Nevado-Holgado
- A few guest lectures at the end
- Many of the lectures we be delivered back-to-back (e.g. I will effectively give 2x1 hour lectures and 2x2 hour lectures)

Course Assessment

- Team project working in groups of 4
- Based on reproducing a research paper
- Each team has a different paper
- Produce a group report + statement of individual contributions + poster
- Individual oral vivas
- Groups will be assigned by department, details are still being sorted
- Check online materials—may end up being some tweaks before you start

Bayesian Machine Learning—Course Outline

Lectures

- Machine Learning Paradigms (1 hour)
- Bayesian Modeling (2 hours)
- Foundations of Bayesian Inference (1 hour)
- Advanced Inference Methods (1 hour)
- Variational Auto-Encoders (1 hour)—key lecture for assessments!

I will upload notes after each lecture. These will not perfectly overlap with the lectures/slides so you will need to separately digest each

What is Machine Learning?

Arthur Samuel, 1959

Field of study that gives computers the ability to learn without being explicitly programmed.

Tom Mitchell, 1997

Any computer program that improves its performance at some task through experience.

Kevin Murphy, 2012

To develop methods that can automatically detect patterns in data, and then to use the uncovered patterns to predict future data or other outcomes of interest.

Motivation: Why Should we Take a Bayesian Approach?

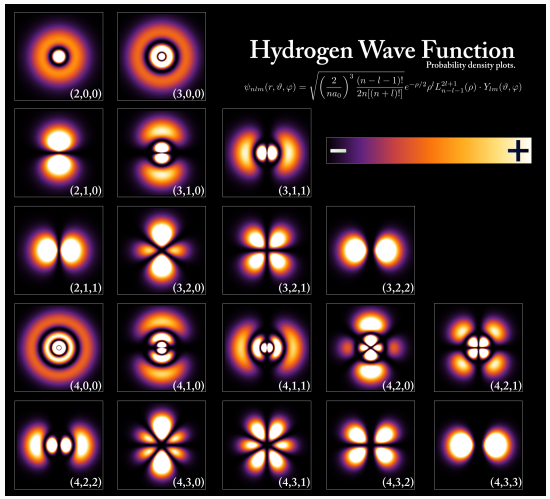
Bayesian Reasoning is the Language of Uncertainty

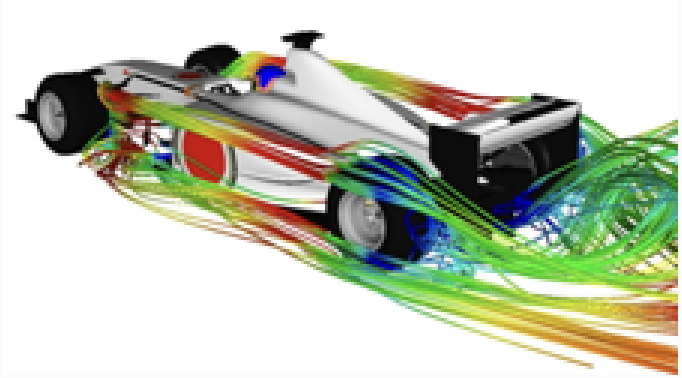
- Bayesian reasoning is the basis for how to make decisions with **incomplete information**
- Bayesian methods allow us to construct models that return principled **uncertainty estimates** rather than just point estimates
- Bayesian models are often **interpretable**, such that they can be easily queried, criticized, and built on by humans

Motivation: Why Should we Take a Bayesian Approach?

Bayesian Modeling Lets us Utilize Domain Expertise

- Bayesian modeling allows us to combine information from data with that from **prior expertise**
- This means we can exploit existing knowledge, rather than purely relying on black-box processing of data
- Models make clear assumptions and are **explainable**
- We can easily update our beliefs as new information becomes available

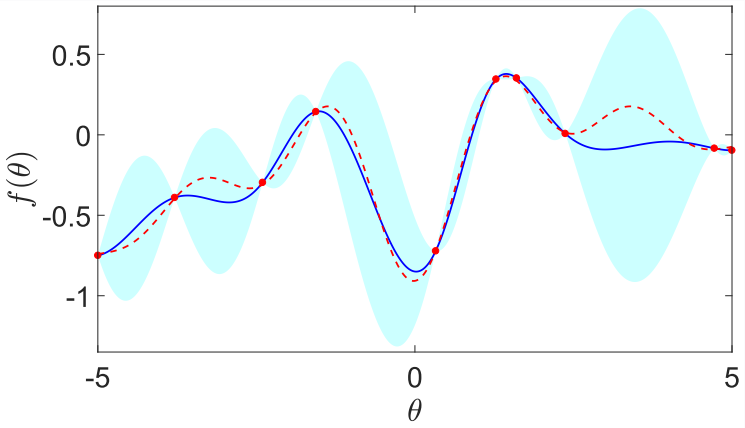




Motivation: Why Should we Take a Bayesian Approach?

Bayesian Modeling is Powerful

- Bayesian models are **state-of-the-art** for a huge variety of prediction and decision making tasks
- They make use of **all** the data and can still be highly effective when data is scarce
- By averaging over possible parameters, they can form rich model classes for explaining **how** data is generated.



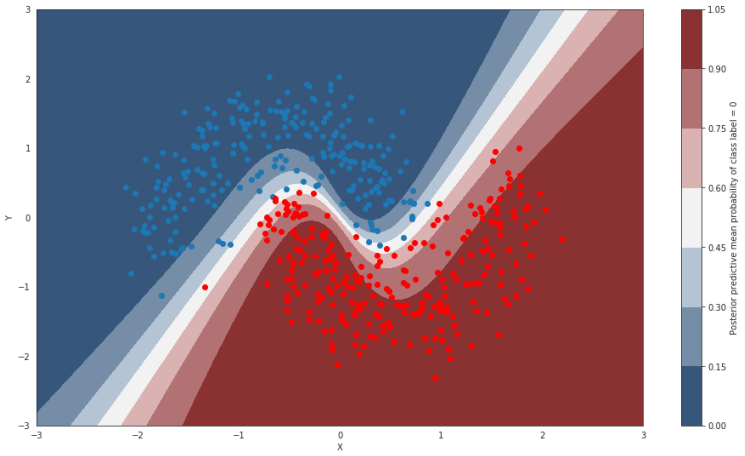


Image Credit: PyMC3 Documentation

Learning From Data

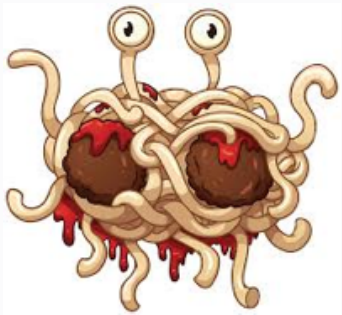
Learning from Data

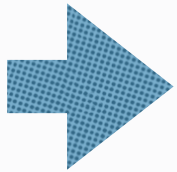
- Machine learning is all about learning from data
- There is generally a focus on making predictions at unseen datapoints
- Starting point is typically a dataset—we can delineate approaches depending on type of dataset

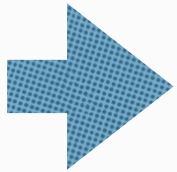
Supervised Learning

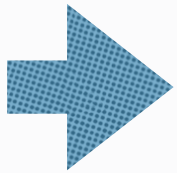
- We have access to a **labeled dataset** of input–output pairs: $\mathcal{D} = \{x_n, y_n\}_{n=1}^{N}$.
- Aim is to learn a **predictive model** f that takes an input $x \in \mathcal{X}$ and aims to predict its corresponding output $y \in \mathcal{Y}$.
- The hope is that these example pairs can be used to "teach" f how to accurately make predictions.

Supervised Learning—Classification









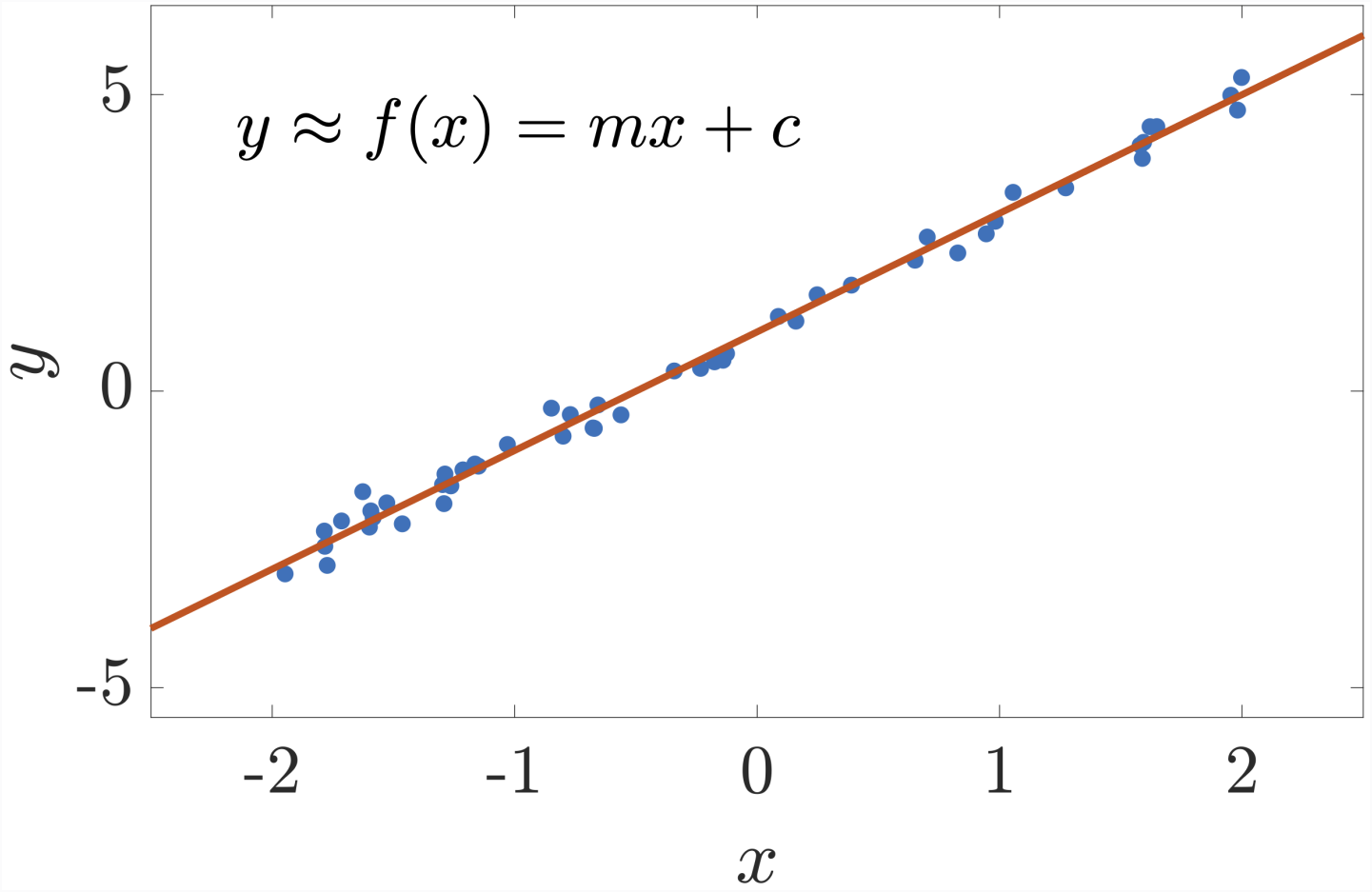
Cat

Dog

Flying Spaghetti Monster

Input x | Predictor $f(x)$ | Class label y

Supervised Learning—Regression



Supervised Learning

Training Data

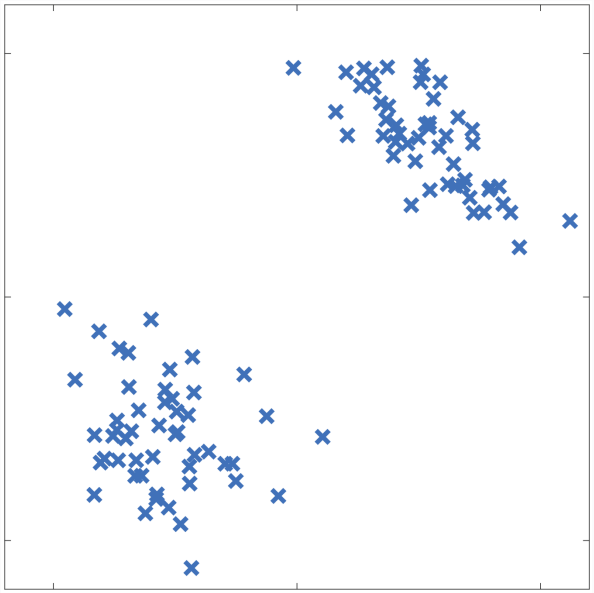
	Input Features					Outputs
Datapoint Index	x_1	x_2	x_3	…	x_M	y
1	0.24	0.12	-0.34	…	0.98	3
2	0.56	1.22	0.20	…	1.03	2
3	-3.20	-0.01	0.21	…	0.93	1
…	…	…	…	…	…	…
N	2.24	1.76	-0.47	…	1.16	2

- Use this data to learn a predictive model $f_\theta : \mathcal{X} \to \mathcal{Y}$ (e.g. by optimizing θ)
- Once learned, we can use this to predict outputs for new input points, e.g. $f_\theta([0.48 \;\; 1.18 \;\; 0.34 \;\; \ldots \;\; 1.13]) = 2$

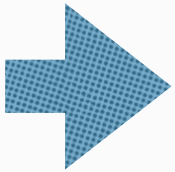
Unsupervised Learning

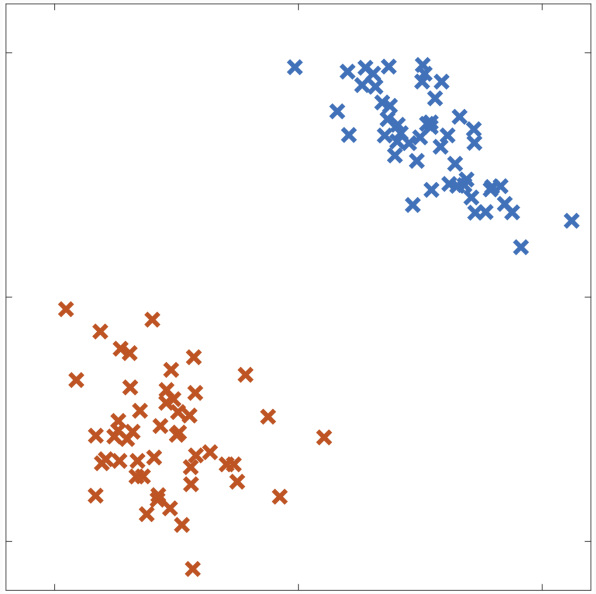
- In unsupervised Learning we have no clear output variable that we are attempting to predict: $\mathcal{D} = \{x_n\}_{n=1}^{N}$
- This is sometimes referred to as **unlabeled data**
- Aim is to exact some salient features for the dataset, such as underlying structure, patterns, or characteristics
- Examples: clustering, feature extraction, density estimation, representation learning, data visualization, data compression

Unsupervised Learning—Clustering



Unlabeled Data





Group into Clusters

Unsupervised Learning—Deep Generative Models

Learn powerful models for generating new datapoints

These are not real faces: they are samples from a learned model!

[1] D P Kingma and P Dhariwal. "Glow: Generative flow with invertible 1x1 convolutions". In: *NeurIPS*. 2018.

Discriminative vs Generative Machine Learning

Discriminative vs Generative Machine Learning

- Discriminative methods try to **directly predict** outputs (they are primary used for supervised tasks)
- Generative methods try to explain **how** the data was generated

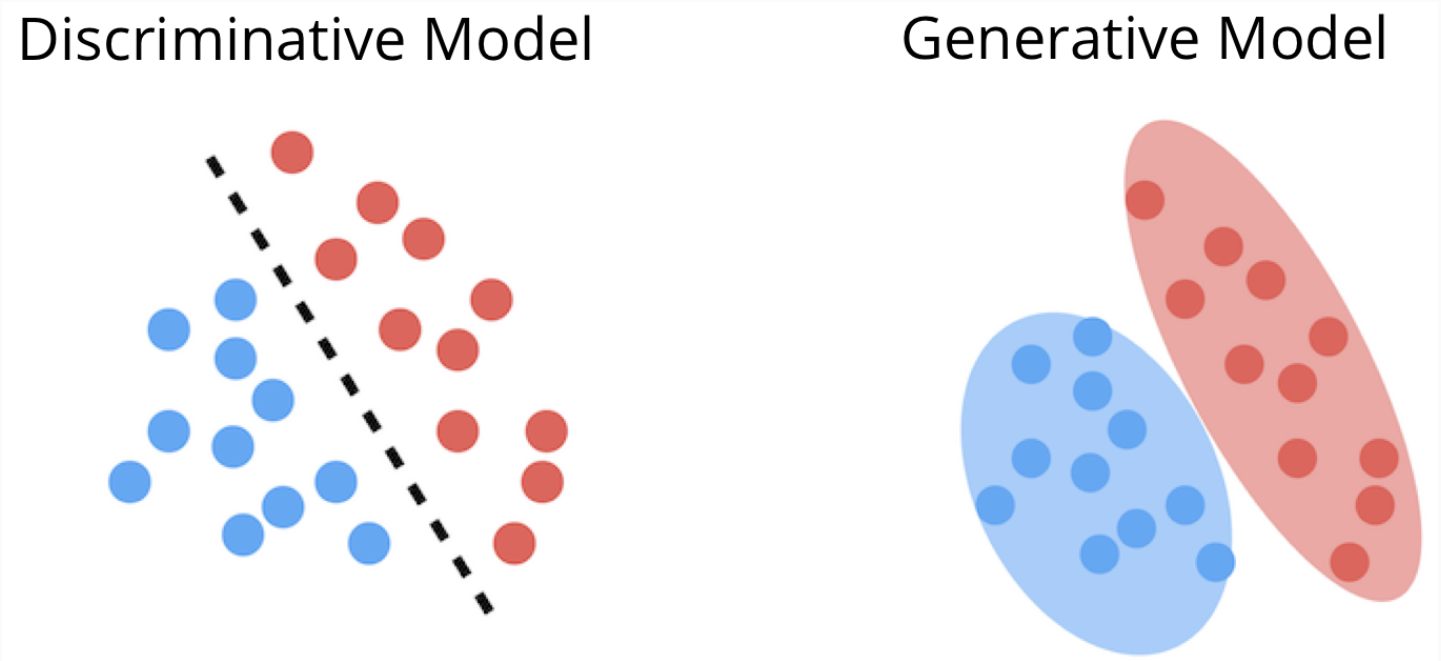


Image credit: Jason Martuscello, medium.com

Discriminative Machine Learning

- Given data $\mathcal{D} = \{x_n, y_n\}_{n=1}^N$, discriminative methods **directly** learn a mapping f_θ from inputs x to outputs y
- **Training** uses $\mathcal{D}$ to estimate optimal values of the parameters θ^*. This is typically done by minimizing an empirical risk over the training data:

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{n=1}^{N} L(y_i, f_\theta(x_i)) \tag{1}$$

  where $L(y, \hat{y})$ is a loss function for prediction $\hat{y}$ and truth y.
- **Prediction** at a new input x involves simply applying $f_{\hat{\theta}}(x)$, where $\hat{\theta}$ is our estimate of θ^*
- Note we often do not predict y directly, e.g. in a classification task we might predict the class probabilities instead
- For **non-parametric** approaches, the dimensionality of θ increases with the dataset size

Discriminative Machine Learning

Common approaches: neural networks, support vector machines, random forests, linear/logistic regression

Pros

- Simpler to directly solve prediction problem than model the whole data generation process
- Few assumptions
  - Often very effective for large datasets
  - Some methods can be used effectively in a black-box manner

Cons

- Can be difficult to impart prior information
- Typically lack interpretability
- Do not usually provide natural uncertainty estimates

Generative Machine Learning

- Generative approaches construct a **probabilistic model** to explain **how** the data is generated
- For example, with labeled data $\mathcal{D} = \{x_n, y_n\}_{n=1}^N$, we might construct a model $p(x, y; \theta)$ of the form $x_n \sim p(x; \theta)$, $y_n|x_n \sim p(y|x = x_n; \theta)$ where θ are model parameters
- This in turns implies a predictive model
- Can also be generative about the **model parameters** θ: e.g. with unsupervised data $\mathcal{D} = \{x_n\}_{n=1}^N$, we can construct a generative model $p(\theta, x)$, such that $\theta \sim p(\theta)$, $x_n|\theta \sim p(x|\theta)$.
    - This is the foundation for Bayesian machine learning

Generative Machine Learning

Common approaches: Bayesian approaches, deep generative models, mixture models

Pros

- Allow us to make stronger modeling assumptions and thus incorporate more problem–specific expertise
- Provide explanation for how data was generated
    - More interpretable
    - Can provide additional information other than just prediction
- Many methods naturally provide uncertainty estimates
- Allow us to use Bayesian methods

Generative Machine Learning

Cons

- Can be difficult to construct—typically require problem specific expertise
- Can impart unwanted assumptions—often less effective for huge datasets
- Tackling an inherently more difficult problem than straight prediction

The Bayesian Paradigm

Bayesian Probability is All About Belief

Frequentist Probability

The frequentist interpretation of probability is that it is the *average proportion of the time an event will occur if a trial is repeated infinitely many times*.

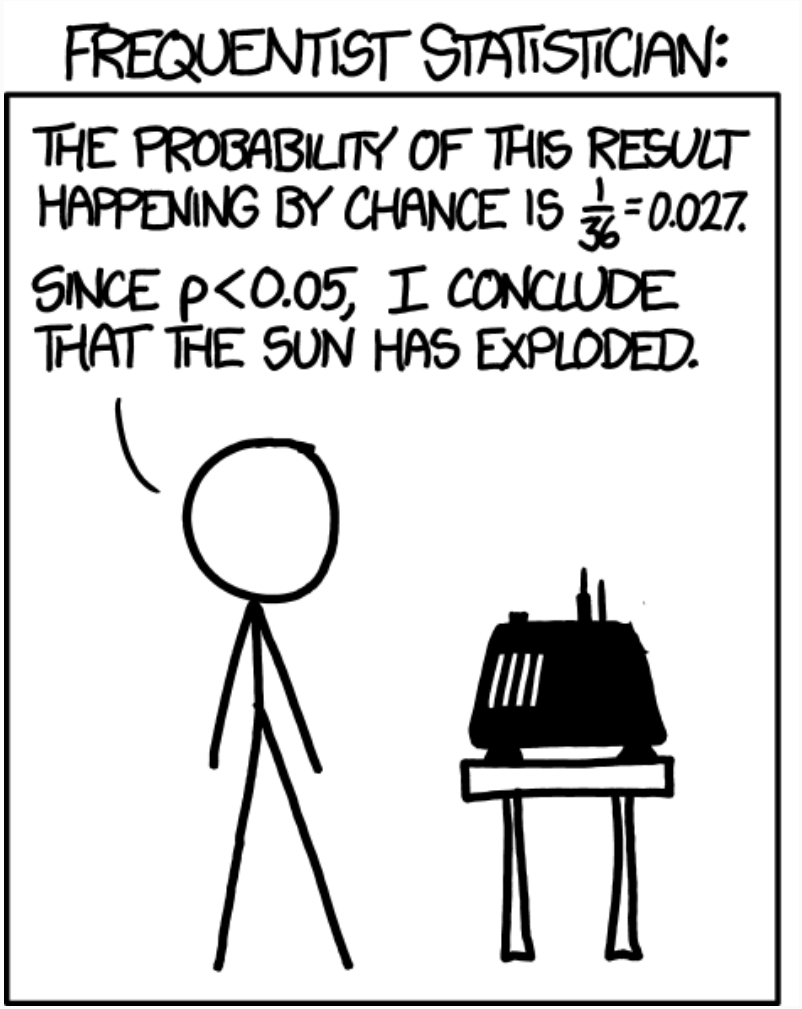
Bayesian Probability

The Bayesian interpretation of probability is that it is the *subjective belief that an event will occur in the presence of incomplete information*

Bayesianism vs Frequentism

https://xkcd.com/1132/

Bayesianism vs Frequentism



https://xkcd.com/1132/

Bayesianism vs Frequentism

https://xkcd.com/1132/

Bayesianism vs Frequentism

Warning

Bayesiansism has its shortfalls too—see the course notes

The Basic Laws of Probability

We can derive most of Bayesian statistics from two rules:

The Product Rule

The probability of two events occurring is the probability of one of the events occurring times the conditional probability of the other event happening given the first event happened:

$$P(A, B) = P(A|B)P(B) = P(B|A)P(A) \tag{2}$$

The Sum Rule

The probability that either A or B occurs, $P(A \cup B)$, is given by

$$P(A \cup B) = P(A) + P(B) - P(A, B). \tag{3}$$

Bayes' Rule

$$p(B|A) = \frac{p(A|B)p(B)}{p(A)}$$

Using Bayes' Rule

- Encode initial belief about parameters θ using a **prior** $p(\theta)$
- Characterize how likely different values of θ are to have given rise to observed data $\mathcal{D}$ using a **likelihood function** $p(\mathcal{D}|\theta)$
- Combined these to give **posterior**, $p(\theta|\mathcal{D})$, using **Bayes' rule**:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} \tag{4}$$

- This represents our **updated belief** about θ once the information from the data has been incorporated
- Finding the posterior is known as **Bayesian inference**
- $p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta)d\theta$ is a normalization constant known as the **marginal likelihood** or **model evidence**
- This does not depend on θ so we have

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta) \tag{5}$$

Multiple Observations: Using the Posterior as the Prior

- One of the key characteristics of Bayes' rule is that it is **self-similar** under multiple observations
- We can use the posterior after our first observation as the prior when considering the next:

$$p(\theta|\mathcal{D}_1, \mathcal{D}_2) = \frac{p(\mathcal{D}_2|\theta, \mathcal{D}_1)p(\theta|\mathcal{D}_1)}{p(\mathcal{D}_2|\mathcal{D}_1)} \tag{6}$$

$$= \frac{p(\mathcal{D}_2|\theta, \mathcal{D}_1)p(\mathcal{D}_1|\theta)p(\theta)}{p(\mathcal{D}_2|\mathcal{D}_1)p(\mathcal{D}_1)} \tag{7}$$

$$= \frac{p(\mathcal{D}_1, \mathcal{D}_2|\theta)p(\theta)}{p(\mathcal{D}_1, \mathcal{D}_2)} \tag{8}$$

- We can thinking of this as continuous updating of beliefs as we receive more information

Example: Positive Cancer Test

We have just had a result back from the Doctor for a cancer screen and it comes back positive. How worried should we be given the test isn't perfect?

Example: Positive Cancer Test (2)

Before these results came in, the chance of us having this type of cancer was quite low: $1/1000$. Let's say θ represents us having cancer so our prior is $p(\theta) = 1/1000$.

For people who do have cancer, the test is 99.9% accurate. Denoting the event of the test returning positive as $\mathcal{D} = 1$, we thus have $p(\mathcal{D} = 1|\theta = 1) = 999/1000$.

For people who do not have cancer, the test is 99% accurate. We thus have $p(\mathcal{D} = 1|\theta = 0) = 1/100$.

Our prospects might seem quite grim at this point given how accurate the test is.

Example: Positive Cancer Test (3)

To figure out the chance we have cancer properly though, we now need to apply Bayes rule:

$$
\begin{aligned}
p(\theta = 1|\mathcal{D} = 1) &= \frac{p(\mathcal{D} = 1|\theta = 1)p(\theta = 1)}{p(\mathcal{D} = 1)} \\
&= \frac{p(\mathcal{D} = 1|\theta = 1)p(\theta = 1)}{p(\mathcal{D} = 1|\theta = 1)p(\theta = 1) + p(\mathcal{D} = 1|\theta = 0)p(\theta = 0)} \\
&= \frac{0.999 \times 0.001}{0.999 \times 0.001 + 0.01 \times 0.999} \\
&= 1/11
\end{aligned}
$$

So the chances are that we actually don't have cancer!

Alternative Viewpoint

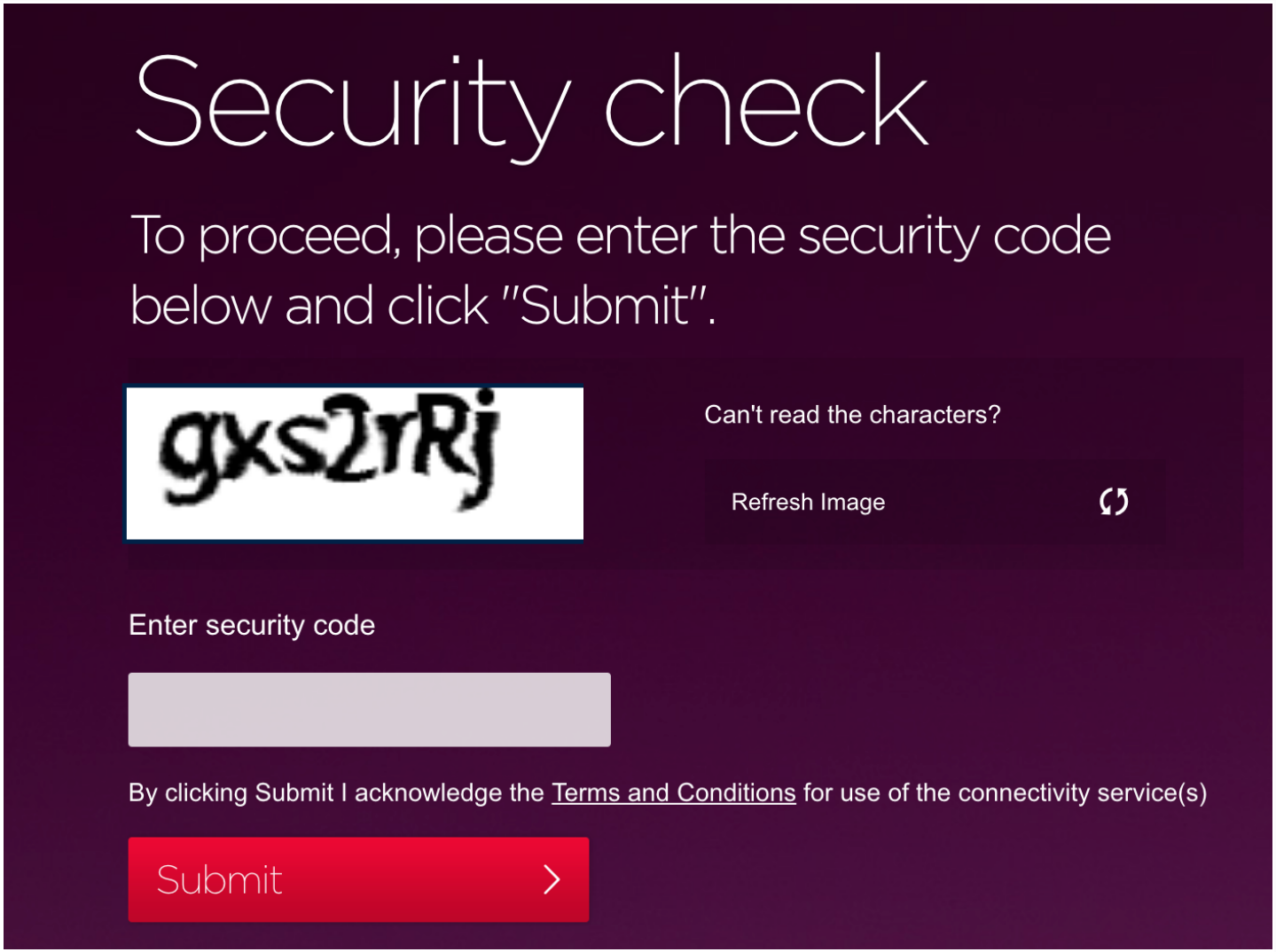
An alternative (equivalent) viewpoint for Bayesian reasoning is that we first define a **joint** model over parameters and data: $p(\theta, \mathcal{D})$

We then **condition** this model on the data taking the observed value, i.e. we fix $\mathcal{D}$

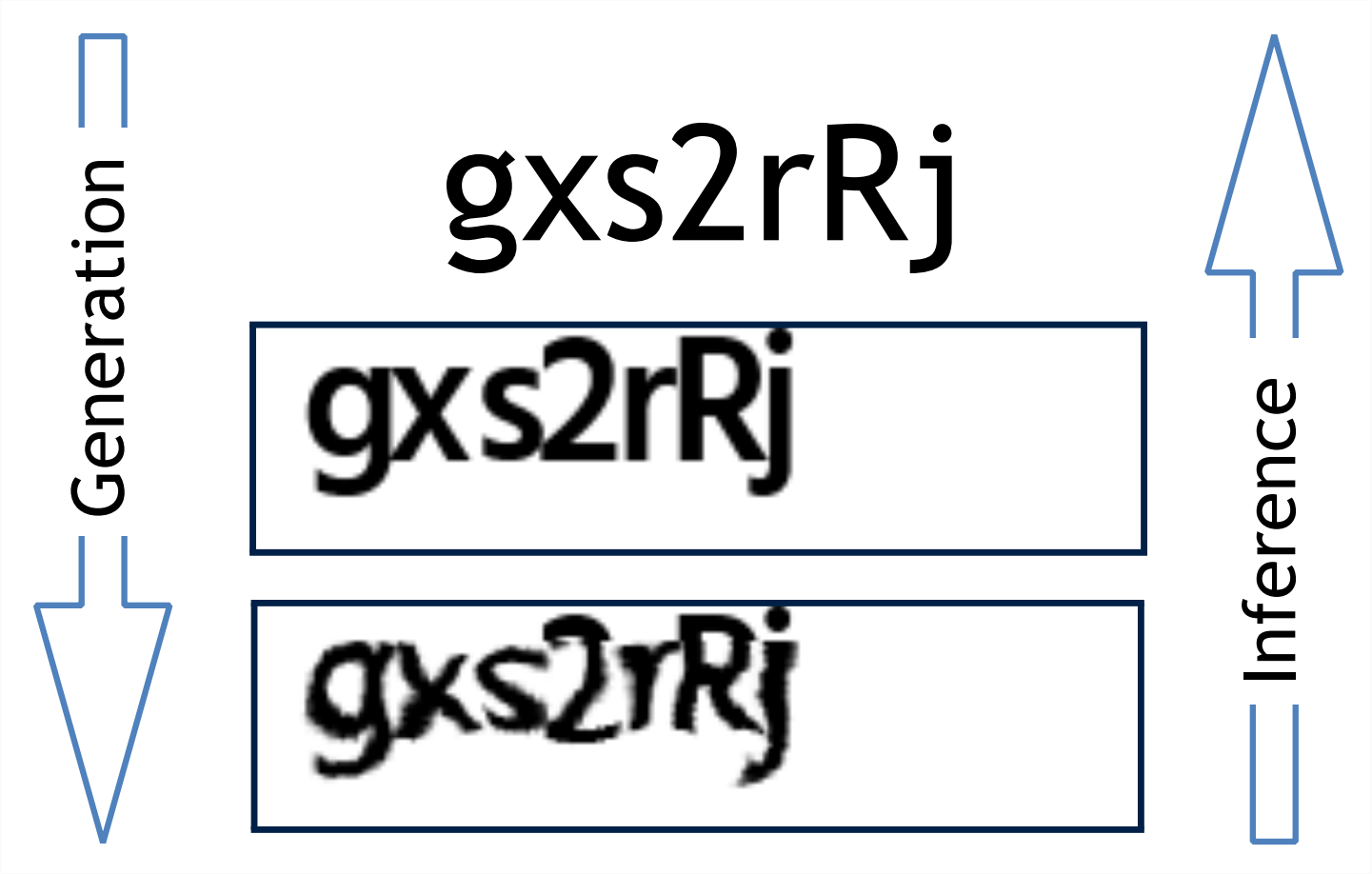
This produces the posterior $p(\theta|\mathcal{D})$ by simply normalizing this to be a valid probability distribution, i.e. the posterior is proportional to the joint for a fixed $\mathcal{D}$:

$$p(\theta|\mathcal{D}) \propto p(\theta, \mathcal{D}) \tag{9}$$

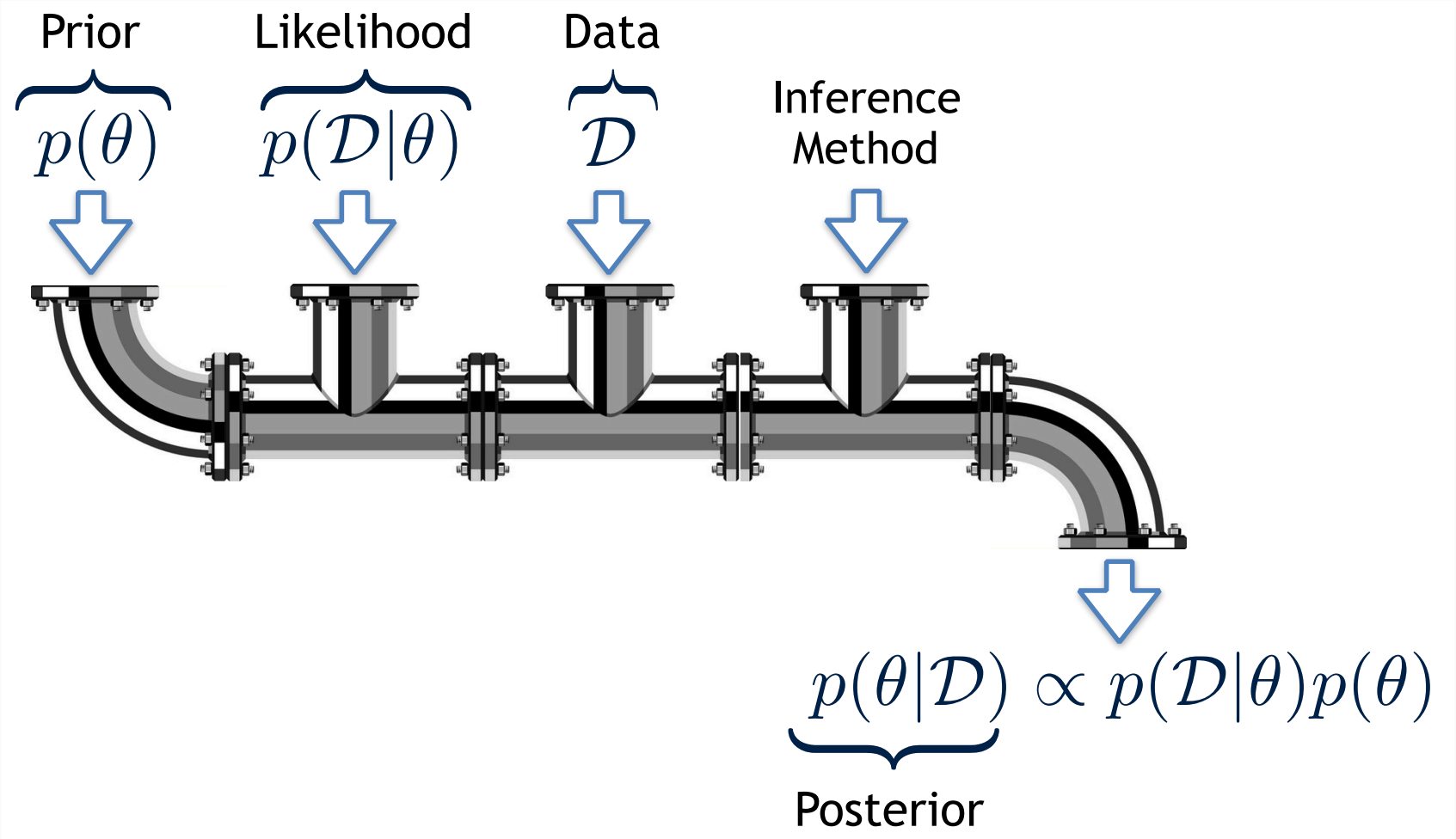
How Might we Write a System to Break Captchas?



Simulating Captchas is Much Easier



The Bayesian Pipeline



Breaking Captchas with Bayesian Models

https://youtu.be/ZTKx4TaqNrQ?t=9

[2] TA Le, A G Baydin, and F Wood. "Inference Compilation and Universal Probabilistic Programming". In: *AISTATS*. 2017.

Making Predictions

- Prediction in Bayesian models is done using the **posterior predictive distribution**
- This is defined by taking the expectation of a predictive model for new data, $p(\mathcal{D}^*|\theta, \mathcal{D})$, with respect to the posterior:

$$p(\mathcal{D}^*|\mathcal{D}) = \int p(\mathcal{D}^*, \theta|\mathcal{D}) d\theta \tag{10}$$

$$= \int p(\mathcal{D}^*|\theta, \mathcal{D}) p(\theta|\mathcal{D}) d\theta \tag{11}$$

$$= \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}^*|\theta, \mathcal{D})]. \tag{12}$$

- This often done dependent on an input point, i.e. we actually calculate $p(y|\mathcal{D}, x) = \mathbb{E}_{p(\theta|\mathcal{D})}[p(y|\theta, \mathcal{D}, x)]$

Making Predictions (2)

Points of Note

- We usually assume that $p(\mathcal{D}^*|\theta, \mathcal{D}) = p(\mathcal{D}^*|\theta)$, i.e. data is conditionally independent given θ
- $p(\mathcal{D}^*|\theta)$ is equivalent to the likelihood model of the new data: in almost all cases we just use the likelihood from the original model
- Calculating the posterior predictive can be computationally challenging: sometimes we resort to approximations, e.g. taking a point estimate for θ (see Lecture 4)
- There are lots of things we might use the posterior for other than just calculating the posterior predictive, e.g. making decisions (see course notes) and calculating expectations

Recap

- Supervised learning has access to **outputs**, unsupervised learning does not
- Discriminative methods try and **directly** make predictions, generative methods try to explain **how** the data is generated
- Bayesian machine learning is a generative approach that allows us to incorporate **uncertainty** and information from **prior expertise**
- Bayes' rule: $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$
- Posterior predictive: $p(\mathcal{D}^*|\mathcal{D}) = \mathbb{E}_{p(\theta|\mathcal{D})}\left[p(\mathcal{D}^*|\theta, \mathcal{D})\right]$

Further Reading

- Look at the course notes! For this lecture there are discussion of Bayesian vs frequentist approaches, and a worked example of Bayesian modeling for a biased coin.
- Chapter 1 of K P Murphy. *Machine learning: a probabilistic perspective*. 2012. https://www.cs.ubc.ca/~murphyk/MLbook/pml-intro-22may12.pdf.
- L Breiman. "Statistical modeling: The two cultures". In: *Statistical science* (2001)
- Chapter 1 of C Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. 2007. https://www.researchgate.net/publication/41222434_The_Bayesian_Choice_From_Decision_Theoretic_Foundations_to_Computational_Implementation.
- Michael I Jordan. Are you a Bayesian or a frequentist? Video lecture, 2009. http://videolectures.net/mlss09uk_jordan_bfway/