CO902
Probabilistic and statistical inference

Lecture 3

Tom Nichols
Department of Statistics &
Warwick Manufacturing Group

t.e.nichols@warwick.ac.uk

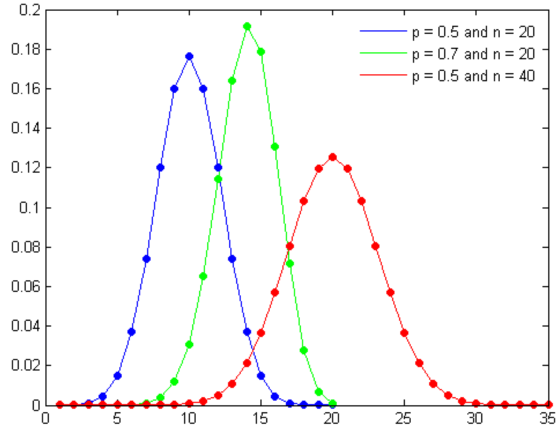
Outline

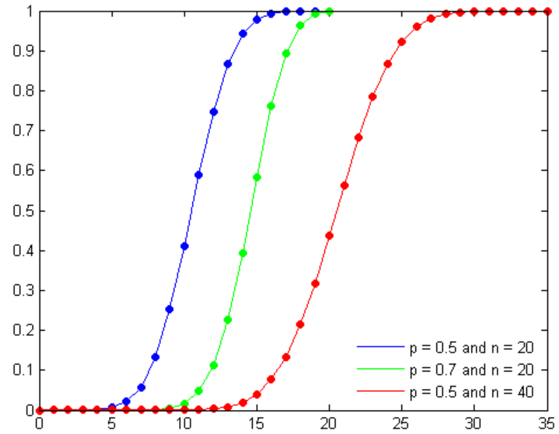
- **Estimation**
  - Parameterized families
  - Data, estimators
  - Likelihood function, Maximum likelihood
- **(In)dependence**
  - The role of *structure* in probabilistic models
  - Dependent RVs, Markov assumptions
  - Markov chains as structural models
- **Properties of estimators**
  - Bias
  - Consistency
  - Law of large numbers

Cumulative distribution function

- For RV *X*, the **cumulative distribution function** or **CDF** is a function which gives the probability that the RV is less than or equal to its argument:

$$F_X(x) \quad = \quad P(X \leq x)$$





Probability density functions

- Let X be a *continuous* RV (i.e. X can take any value in a finite or infinite interval)
- Let F_X be the cdf of X
- Then for $a<b$:

$$
\begin{aligned}
P(X \leq b) &= P(X \leq a) + P(a < X \leq b) \\
P(a < X \leq b) &= P(X \leq b) - P(X \leq a) \\
&= F_X(b) - F_X(a)
\end{aligned}
$$

Probability density functions

$$
\begin{aligned}
P(X \leq b) &= P(X \leq a) + P(a < X \leq b) \\
P(a < X \leq b) &= P(X \leq b) - P(X \leq a) \\
&= F_X(b) - F_X(a)
\end{aligned}
$$

- Assume cdf differentiable:

$$
\frac{\mathrm{d}}{\mathrm{d}x} F_X(x) = f_X(x)
$$

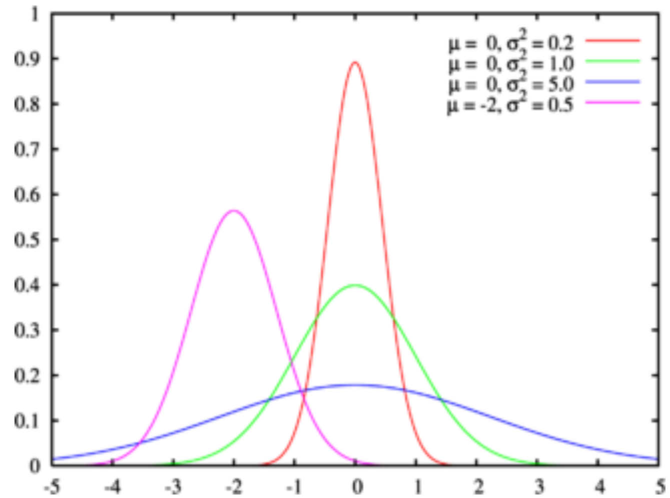
- This gives:

$$
P(a < X \leq b) = \int_a^b f_X(x)\,\mathrm{d}x
$$

Probability density functions

$$P(a < X \leq b) \;=\; \int_a^b f_X(x)\,\mathrm{d}x$$

- The function f_X is called the **probability density function** or **pdf** of RV *X*
- For small dx, probability that X lies between x and x+dx is $f_X(x)dx$
- Intuitively, shape of pdf tells us which regions the RV is more likely to fall into
- We will use:
  - $p(x)$ to refer to a pdf
  - $P(x)$ for either a pmf or a direct probability statement



PDFs: properties

$$P(a < X \leq b) = \int_a^b p(x)\,\mathrm{d}x$$

$$\int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1$$

$$\forall x \cdot p(x) \geq 0$$

- Note that the density at *x*, *p(x)* is *not* a probability: it can exceed 1
- The pdf has to integrate to one, because the RV must take *some* value
- The pdf has to be everywhere non-negative because of the monotonicity of the cdf
- pdf value is not a probability!

$$P(X = x) \neq p(x)$$

  For a continuous r.v., probability that it takes on value exactly x is 0
- Easy to confuse pdf and pmf; be careful!

Expectation

$$\begin{aligned}\mu_X &= \mathbb{E}[X] \\ &= \int_{x \in \mathcal{X}} x\, p(x)\, \mathrm{d}x\end{aligned}$$

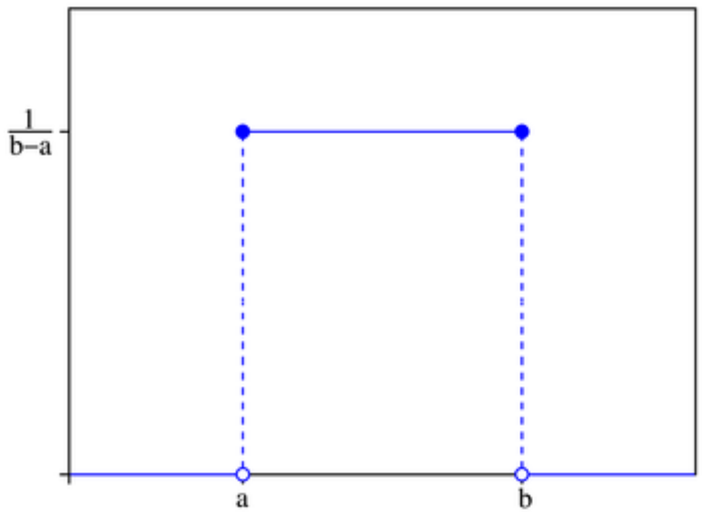
is the **expectation** or **expected value** or **mean** of continuous RV *X*

- More generally, if *g(X)* is a function of RV *X*, *g(X)* is also an RV, with expected value:

$$\mathbb{E}[g(X)] = \int_{x \in \mathcal{X}} g(x)\, p(x)\, \mathrm{d}x$$

- Similarly, we get the **variance** and **standard deviation** of *X*

Uniform pdf



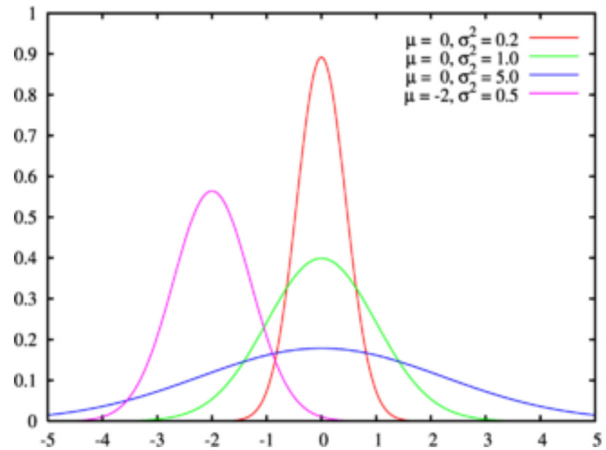
$$p(x) = \begin{cases} \frac{1}{b-a} & \text{if } x \in [a, b] \\ 0 & \text{otherwise} \end{cases}$$

- Intuitively: description of a RV all of whose values over some range are equally likely

Normal or Gaussian pdf

$$p(x \mid \mu, \sigma^2) \quad = \quad \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$$-\infty < x < \infty$$

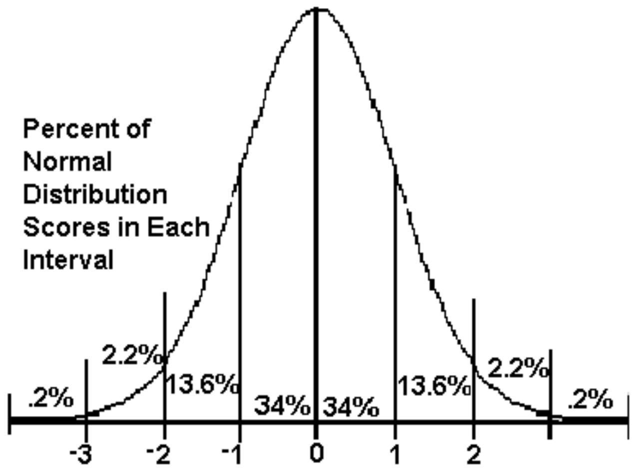


- Arguably single most important PDF
- Parameters are the **mean** and **variance**
- Many interesting properties: CLT, maximum entropy etc.
- Note that this is a **family of pdfs**

Normal or Gaussian pdf

$$p(x \mid \mu, \sigma^2) \quad = \quad \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$$-\infty < x < \infty$$



- Exponent is square of #of std deviations distance from the mean
- This makes it fall off quickly away from the mean: the density has "light tails"
- 68% of mass lies within 1 std dev either side of the mean, 95% within 2 and 98% within 3

- We'll encounter other pdfs as we need them

Covariance

- For two RVs X and Y, the **covariance** *COV(X,Y)* is defined as:

$$COV(X,Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

- **Q: What is COV(X,X)?**
- **Q: If *X,Y* are independent, what is *COV(X,Y)* ?**

Random vectors

- A **random vector** is a vector whose components are RVs:

$$\mathbf{X} = [X_1 X_2 \ldots X_d]^T$$

- The **mean vector** is a vector whose components are the means of the components of ***X***:

$$\begin{aligned} \boldsymbol{\mu} &= \mathbb{E}[\mathbf{X}] \\ &= [\mathbb{E}[X_1]\mathbb{E}[X_2] \ldots \mathbb{E}[X_d]]^T \end{aligned}$$

Covariance matrix

- The **covariance matrix** $\boldsymbol{\Sigma}$ of a random vector is a matrix whose components are the covariances of pairs of vector components:

$$\mathbf{X} = [X_1 X_2 \ldots X_d]^T$$

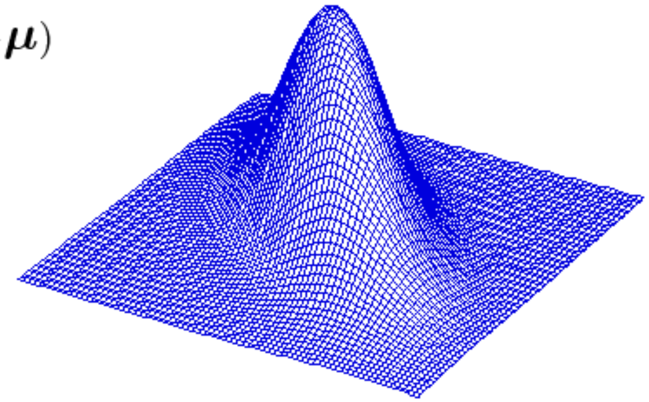
$$\Sigma_{ij} = COV(X_i, X_j)$$

- **Q: what are the entries along the diagonal?**

Multivariate normal pdf

$$p(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

$$\mathbf{x} \in \mathbb{R}^d$$



- **Multivariate** is statistics-speak for multi-dimensional
- To get the probability that the RV lies in some region, we have to integrate the pdf over that region
- Exponent is a weighted distance between ***x*** and ***μ***, and is sometimes called the **Mahalanobis distance**

Sum, product and Bayes rules for pdfs

$$p(y) = \int_{-\infty}^{\infty} p(x, y)\,\mathrm{d}x$$ *(sum)*

$$= \int_{x \in \mathcal{X}} p(x, y)\,\mathrm{d}x$$ *(sum; support)*

$$p(x, y) = p(x \mid y)p(y)$$ *(product)*

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)}$$ *(Bayes)*

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{\int_{x \in \mathcal{X}} p(y \mid x)p(x)\,\mathrm{d}x}$$

Bayesian inference

- **Bayesian inference** is a different approach to characterizing unknown parameters which uses the rules of probability to get a probability distribution *over* the unknown parameter
  - The distribution *before* we see any data is called the **prior**
  - The distribution after we see the data is called the **posterior**
- The prior brings a non-likelihood element into inference
- Today:
  - Intro to Bayesian inference and
  - Application to the Bernoulli model

Bernoulli MLE

- We've seen that the Bernoulli MLE has some nice properties:
  - Intuitively appealing
  - Unbiased
  - Consistent
- These kinds of properties are nice, but in modelling what we're really after is **predictive power**
- Suppose we get the following sequence of coin tosses:

H, H, H

- What's the Bernoulli MLE's prediction for the next toss?

Overfitting

- What's going on is a kind of “overfitting”
- The model has tuned itself too closely to the data
- Another example: curve-fitting...
- These are toy examples but overfitting is a serious concern in real-life models in many areas:
  - Biology (e.g. large gene networks)
  - Finance (recent events?)
  - Climate models
- In these cases models can have 100s or 1000s of parameters, maybe more if you consider model uncertainty: for sufficiently complicated models overfitting remains a concern even when there seems to be “lots” of data
- Likelihood is important, but it's entirely data-driven
- In practice, with finite data, can be helpful to have a non-data term...

Bayesian inference

- **Bayesian inference** is an approach to statistical problems in which
  - Uncertainty about the parameter of interest is captured by a **probability distribution over the parameter**, and
  - The rules of probability are used to characterize this distribution, with **Bayes' rule** front and centre (hence the name)
- The idea of having a distribution for the parameter may seem a bit odd
- But if probability distributions are meant to capture **uncertainty,** it's actually pretty natural: we are **uncertain** about the value of the parameter, and want to say capture our state of knowledge about it

Posterior distribution

- Distribution over parameter, *given* the data we've observed:

$$p(\theta \mid X_1 \ldots X_n)$$

- This is a **posterior distribution**, because it comes after the data
- In this case the parameter is continuous, so it's going to be a density

- But our original data model gives us $P(X_1 \ldots X_n \mid \theta)$
- *Not* $p(\theta \mid X_1 \ldots X_n)$

... use Bayes' rule to “flip around”

Prior distribution

- Using Bayes' rule:

$$
\begin{aligned}
p(\theta \mid X_1 \ldots X_n) &= \frac{P(X_1 \ldots X_n \mid \theta) p(\theta)}{P(X_1 \ldots X_n)} \\
&\propto P(X_1 \ldots X_n \mid \theta) \times p(\theta)
\end{aligned}
$$

- What does $p(\theta)$ represent?
- This is the distribution over the parameter *before* seeing any data
- It's therefore called the **prior distribution**

Bayesian inference for the Bernoulli

- Data: *n* tosses

$$X_1, X_2 \ldots X_n$$

- Likelihood:

$$\begin{aligned} X_i &\overset{iid}{\sim} Bernoulli(\theta) \\ P(X_1, X_2 \ldots X_n \mid \theta) &= \prod_{i=1}^{n} P(X_i \mid \theta) \\ &= \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{(1-x_i)} \end{aligned}$$

- In the Bayesian approach we aim to get a *distribution* over the parameter, *given* the data we've observed...

Prior for Bernoulli model: desiderata

- We need a **prior distribution** $p(\theta)$
- This should be:
  - A density over the range [0,1]
  - Tunable, to give us flexibility in different situations (e.g. expect nothing in particular, expect coin to be nearly fair, expect coin to be grossly unfair etc.)

Beta pdf

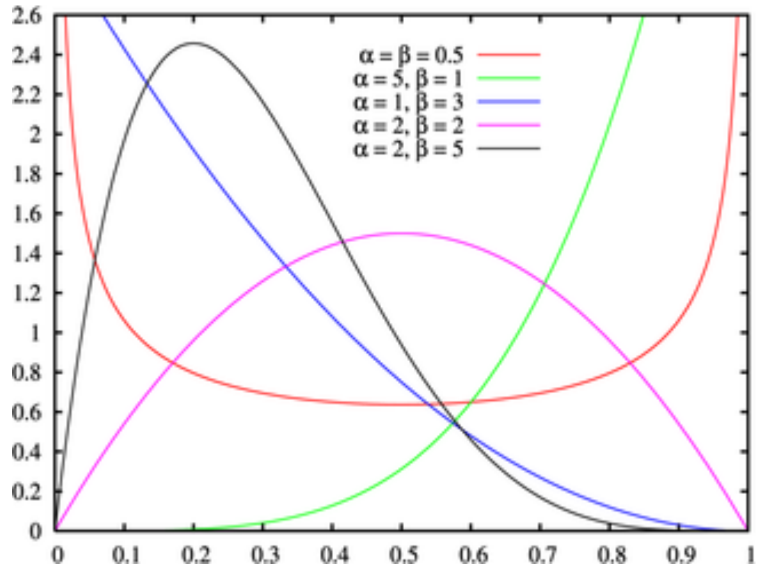
$$p(x \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

$$x \in [0,1]$$

$$\alpha, \beta > 0$$

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t}\, dt$$

$$x > 0$$



- PDF for RVs taking values in the unit interval
- Parameters can be adjusted to give bell-shaped, u-shaped, or skewed densities
- Much used in **Bayesian inference**, as a **prior density** for probability parameters
- We'll use the Beta a great deal

Beta prior

- We'll use a Beta pdf as a prior for the Bernoulli parameter**:**

$$\begin{aligned} p(\theta \mid \alpha, \beta) &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\,\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ \theta &\in [0,1] \\ \alpha, \beta &> 0 \end{aligned}$$

- Parameters of the prior are then called **hyperparameters**
- Consider two options:
  - Most typically a fair coin, but sometimes weighted towards H's or T's with diminishing probability. E.g. Beta(2,2)
  - Or, if we want to start off completely uninformed, we could make the prior uniform over [0,1]. This corresponds to Beta(1,1)

Posterior

- Using the Beta prior and Bernoulli likelihood, let's work out the posterior density:

$$
\begin{aligned}
p(\theta \mid X_1 \ldots X_n) &\propto P(X_1 \ldots X_n \mid \theta) \times p(\theta) \\
&\propto \theta^{n_1}(1-\theta)^{(n-n_1)} \times \theta^{\alpha-1}(1-\theta)^{\beta-1} \\
&= \theta^{n_1+\alpha-1}(1-\theta)^{(n-n_1+\beta-1)}
\end{aligned}
$$

- **Q: Does this look familiar?**
- **Q: What is the normalizing factor?**

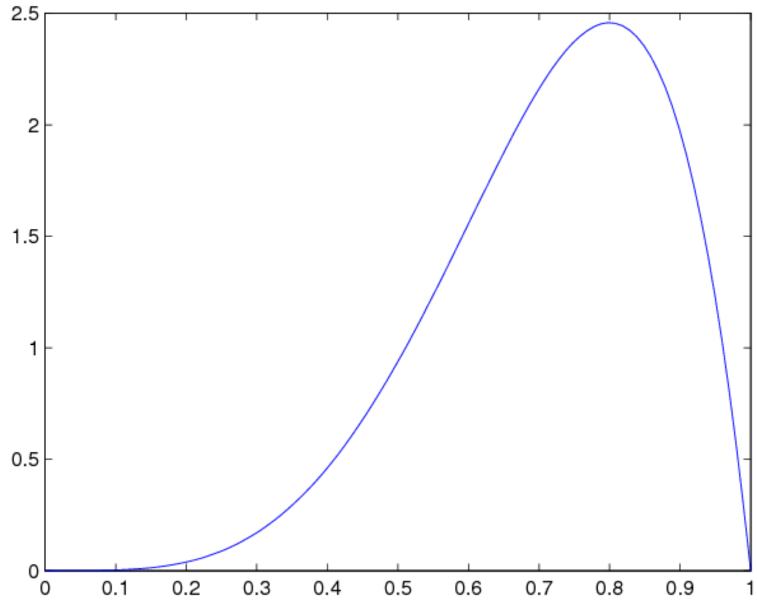
Posterior

- Recognizing the Beta “kernel”, we can see that the posterior distribution is $Beta(n_1 + \alpha, n - n_1 + \beta)$:

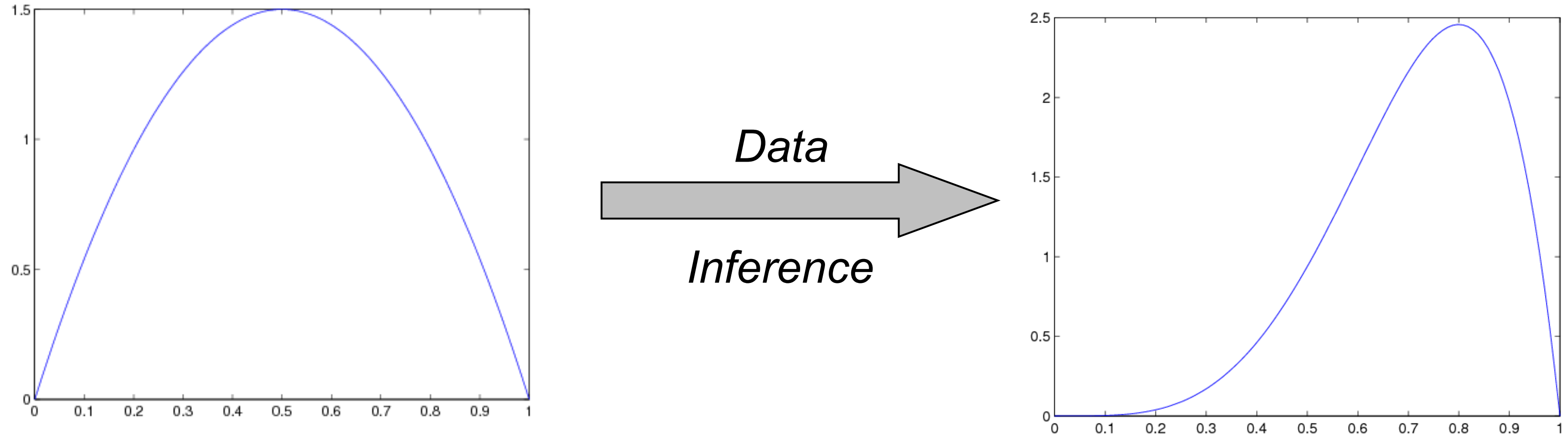
$$p(\theta \mid X_1 \ldots X_n) \quad = \quad \frac{\Gamma(n+\alpha+\beta)}{\Gamma(n_1+\alpha)\Gamma(n-n_1+\beta)} \theta^{n_1+\alpha-1}(1-\theta)^{n-n_1+\beta-1}$$

- **What does the posterior look like?**

Posterior



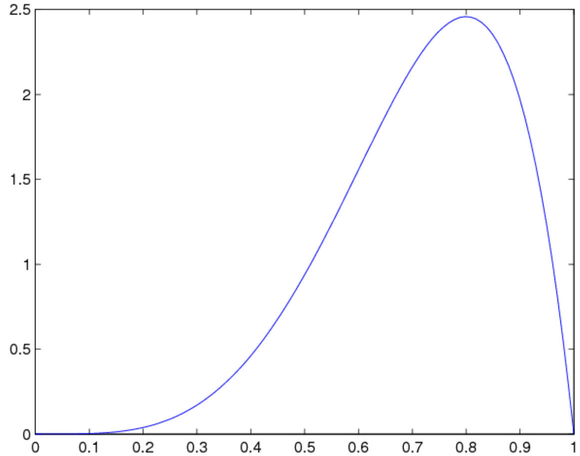
Posterior



Prior

Posterior

Posterior



Posterior

- **This looks reasonable, no?**
- This object is *the* key element of any Bayesian analysis, because it describes our current state of knowledge about the unknown parameter
- We can therefore use it to say something about other quantities which depend on the parameter

Conjugate priors

Prior $$p(\theta) \quad \propto \quad \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

Posterior $$p(\theta \mid X_1 \ldots X_n) \quad \propto \quad \theta^{n_1+\alpha-1}(1-\theta)^{n-n_1+\beta-1}$$

- The posterior ended up being of the **same form** as the prior
- This helped us to characterize the posterior distribution, in this case by recognizing the Beta kernel
- This property – of a posterior having the same form as a prior - is called **conjugacy**
- In this case the **Beta is a conjugate prior for the Bernoulli**

MAP estimators

- The posterior distribution is not a (point) estimate, in the sense that it doesn't give a single “answer”
  - Prior – Belief about different possible values of the parameter **before** seeing the data
  - Posterior – Belief about possible values of the parameter **after** seeing the data
- The following point estimator is often derived from the posterior:

$$\hat{\theta}_{MAP} = \underset{\theta}{\operatorname{argmax}}\, p(\theta \mid X_1 \ldots X_n)$$

- This is called a **maximum a posteriori** or **MAP** estimator
- **Q: Using the posterior distribution we have derived, write down the Bernoulli MAP estimate**

MAP estimate for the Bernoulli

- Log-posterior:

$$\log(p(\theta|X_1, \ldots, X_n)) \propto (n_1 + \alpha - 1)\log(\theta) + (n - n_1 + \beta - 1)\log(1 - \theta)$$

- Setting derivative to zero and solving, we get:

$$\hat{\theta}_{MAP} = \frac{n_1 + \alpha - 1}{n + \alpha + \beta - 2}$$

- **For our dataset of three heads, and the Beta(2,2) prior, what *is* the MAP estimate?**
- **Does this feel more or less reasonable than the MLE?**
- **What is the MAP estimate with the flat prior Beta(1,1)?**

Properties of the MAP estimator

- The MAP estimator is just another estimator, so we can look into it's properties, like **bias** and **consistency**
- This would proceed along the same lines as we saw for the MLE
  - i.e., in practice, we use numerical simulation
- Generally, Bayesian approaches tend to agree with ML in the limit of lots of data, because the effect of the prior gets "wiped out" by the likelihood, which makes sense
- But for sample sizes which are small-to-moderate in relation to the complexity of the model (and this can mean pretty *large* for a complex model) the answers can be very different, as we've seen

Bayesian computation

- In practice, relevant computations (characterizing posteriors, intergrating out things you're not interested in) are rarely as “nice” as our Bernoulli example

- This has meant that approximate, computational approaches like *Markov chain Monte Carlo* are important in Bayesian inference

- This is one reason Bayesian methods are now vastly more popular than a few decades ago: today you can perform pretty “heavy-duty” approximate inference on a desktop PC...

Bayesian inference generally

- So this is how Bayesian inference works, no matter how complicated the situation:

$$posterior \propto likelihood \times prior$$

- As we've seen, the prior is *not* data-dependent
- This is one thing which has, over the years, made Bayesian inference somewhat controversial
- Some people feel uncomfortable specifying a prior because it seems too subjective

Bayesian inference generally

- However, nowadays Bayesian approaches are popular in many practical applications, including:
  - Engineering (e.g. robotics)
  - CS (e.g. language, AI)
  - Biology (e.g. gene networks) etc.
- One appealing feature is the ability to **incorporate background knowledge** in a *principled* manner
  - Often, it's natural enough to say *something* about the system of interest, *a priori*
  - Bayes then tells us *how* to combine our possibly vague prior knowledge with data
- Equally, using "uninformative" priors, Bayes is a nice way (but certainly not the only way) to "regularize" problems
- Finally, opens up a principled way of doing model comparison

Bayes Conclusions

- In conclusion: shouldn't accept any method uncritically, but both Bayes and ML are important ideas to have in your **conceptual toolbox**

Outline of course

A. Basics: Probability, random variables (RVs), common distributions, introduction to statistical inference

B. Supervised learning: Classification, regression; including issues of over-fitting; penalized likelihood & Bayesian approaches

C. Unsupervised learning: Dimensionality reduction, clustering and mixture models

D. Networks: Probabilistic graphical models, learning in graphical models, inferring network structure

Outline

(1) Introduction to **supervised learning**

(2) **Classification**

(3) Generic classifier based on **generative model** and **class-conditional distributions**

(4) Discrete “Naive Bayes” classifier

Supervised learning

- **Supervised learning:** prediction problems where you start with a dataset in which the "right" answers are given
- Supervised in the sense of "learning with a teacher"

- This is a topic with a **huge range of applications...**

Predicting drug response

???

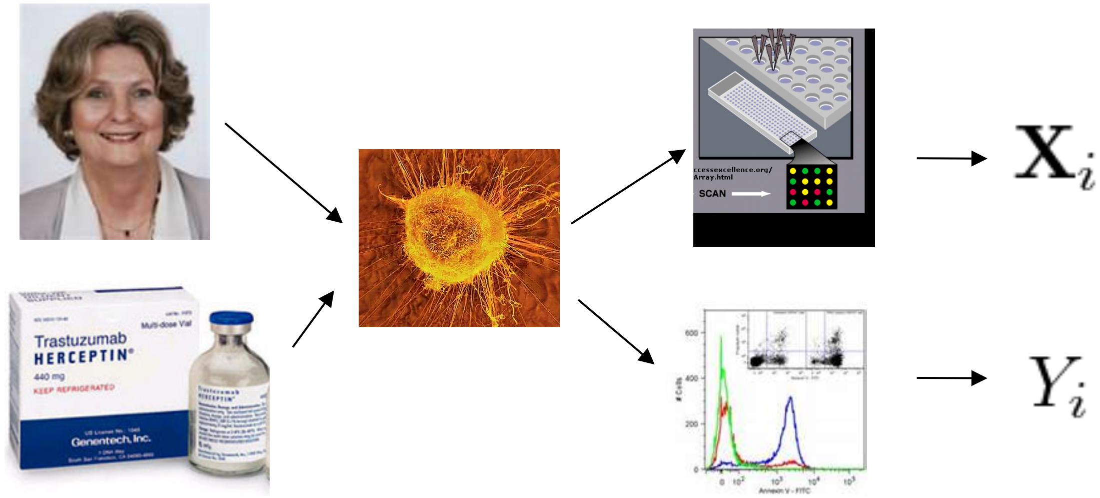
- Cancer drugs don't work equally well for everyone
- Response comes about via complex interplay between drug and individual genetics, gene expression, protein levels etc. (not to mention social and psychological factors...)
- **Individual differences** in genetic and molecular factors can lead to very different outcomes – e.g. Herceptin
- Much interest in understanding the factors which contribute to such heterogeneity and how to **personalize therapy** to individuals

Predicting drug response

???

- Genomic data can tell us about the individual's gene code
- Equally, technologies like microarrays & protein chips allow us to **capture the molecular state** of an individual: that is, extent to which each of 10000s of genes are "switched on", which proteins are present etc.
- Such data offer possibility of **molecular prediction of drug response**
- A (good) predictor could play a **clinical role** and also point to **molecular mechanisms** underlying heterogeneity in drug response

Predicting drug response

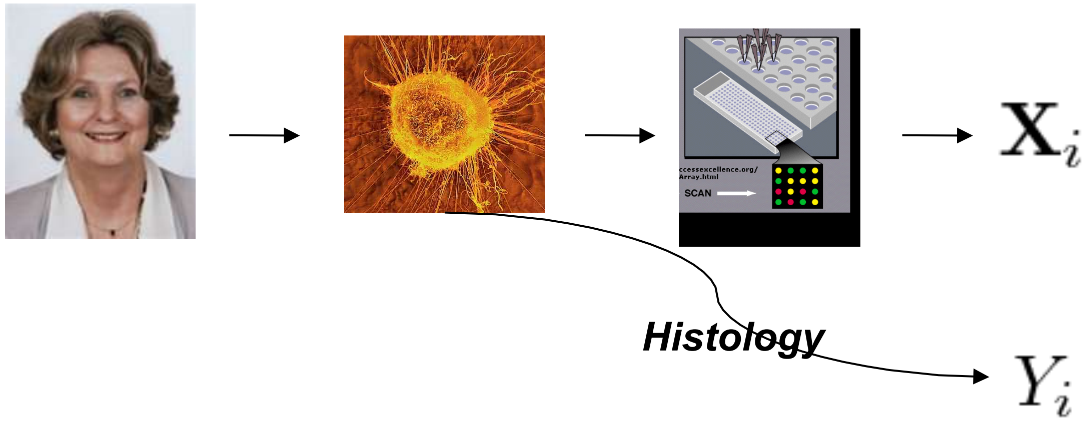


- Suppose we collect **data** of the following kind:
  - For each of *n* patients, we get a tumour sample, and using a microarray obtain **expression measurements** for d=10k genes
  - Also, we administer the drug to each of the *n* patients, and record a numerical measure of **drug response**
- This gives us data of the following kind:

$$\{\mathbf{X}_i, Y_i\},\ i = 1..n$$

$$\mathbf{X}_i \in \mathbb{R}^d,\ Y_i \in \mathbb{R}$$

Class of cancer



- Many subtly different forms of cancer
- These can be hard to distinguish by examination or under the microscope
- Instead, we can use high-throughput data to try to recognize molecular signatures which are predictive of the type of cancer
- Here, the thing being predicted is a "class" rather than a number
- Data:

$$\{\mathbf{X}_i, Y_i\},\ i = 1..n$$

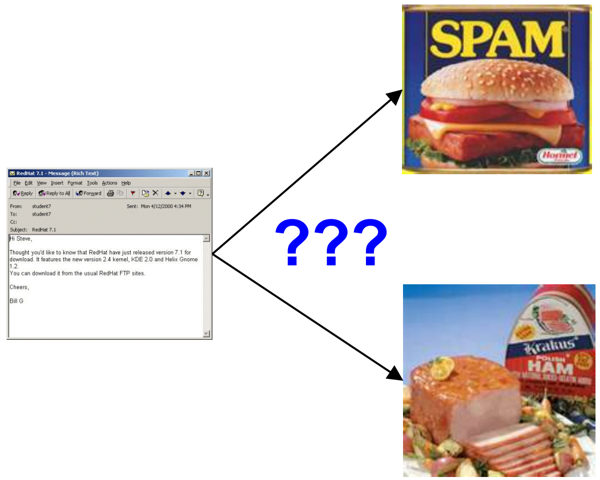
$$\mathbf{X}_i \in \mathbb{R}^d,\ Y_i \in \{1, 2, \ldots k\}$$

Spam prediction

- Drowning in **spam –** One statistic: of the 4 billion emails Hotmail receive each day, they only deliver 600 million
- We can recognize spam when we see it
- Doing this **automatically** involves introspection and hand-coding of the heuristics we use, and/or **learning from examples** what the difference is
- That is, given *n* email messages, each flagged as spam/non-spam, we seek to learn a rule which will tell the two apart
- Emails might be described by the presence/ absence of each of *d* words
- Then, **data:**

$$\{\mathbf{X}_i, Y_i\},\ i = 1..n$$

$$\mathbf{X}_i \in \{0,1\}^d,\ Y_i \in \{0,1\}$$



Object recognition

- Object recognition: recognizing the class of an object from an image
- Our facility with this belies the fact that this is ***very*** hard problem
- Applications in image processing, image search, but also interest from cognitive psychology

“duck”

“tiger”

*Input **X***

Output Y

- Here again the thing being predicted is discrete
- Data would look like:

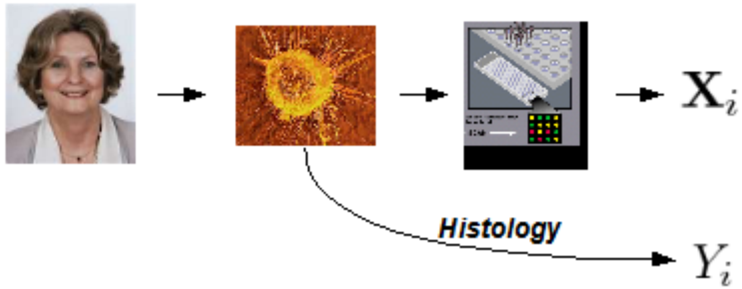
$$\{\mathbf{X}_i, Y_i\},\ i = 1..n$$

$$\mathbf{X}_i \in \mathbb{R}^d,\ Y_i \in \{1, 2, \ldots k\}$$

Supervised learning

- In general terms:
  - we have $\{\mathbf{X}_i, Y_i\}$
  - want to predict *Y* from *X*
- We can **learn** a predictor from the data $\{\mathbf{X}_i, Y_i\}$
- This is called **supervised learning**, because it's like learning with a teacher: you get told the right answer for the examples you learn from
- In contrast, **unsupervised learning** is about finding interesting regularities or patterns in data *without* a labelled dataset:
  - Examples: *clustering*, or finding interesting groups in data, *dimensionality reduction*, or finding informative low-dimensional data representations
- Today, **classification**

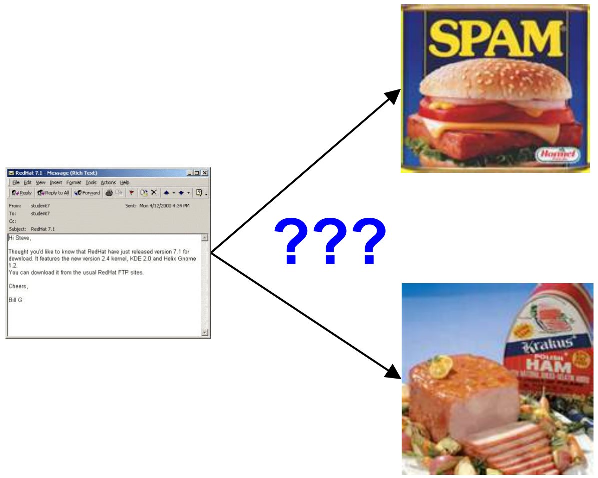
Classification

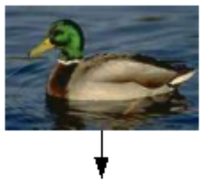


All these problems share a common structure

$$\{\mathbf{X}_i, Y_i\},\ i = 1..n$$

$$\mathbf{X}_i \in \mathbb{R}^d,\ Y_i \in \{1, 2, \ldots k\}$$





"duck"

"tiger"

*Input **X***

Output Y

Classification

- These are all examples of **classification problems**
- Classification: supervised learning problem in which the output is a (finite) set of classes or categories (rather than real-valued, as in regression, e.g. drug response)

$$\{\mathbf{X}_i, Y_i\},\ i = 1..n$$

$$\mathbf{X}_i \in \mathbb{R}^d,\ Y_i \in \{1, 2, \ldots k\}$$

- This is a **very general class of problems**

Generative model

- Question: given vector-valued input data, with each datapoint belonging to one of two classes, can we learn a probability model to automatically classify such observations?

- Data:

$$\{\mathbf{X}_i, Y_i\}, \qquad i = 1..n$$
$$\mathbf{X}_i \in \mathbb{R}^d$$
$$Y_i \in \{0, 1\}$$

- One way to approach this sort of problem is to
  - think of a model which could have *generated* the data, and
  - then use it to both make predictions and answer questions about features of interest

- This is called a **generative model**

Class-conditional generative model

- Data:

$$\begin{aligned} \{\mathbf{X}_i, Y_i\}, & \quad i = 1..n \\ \mathbf{X}_i & \in \mathbb{R}^d \\ Y_i & \in \{0, 1\} \end{aligned}$$

- What kind of model do we want?
- There are two distinct classes, so we certainly don't expect all of the data to come from the *same* distribution
- We can instead use two distributions, one for each class...

$$\begin{aligned} p(\mathbf{X} \mid Y = k) &= p_k(\mathbf{X}) \\ &= p(\mathbf{X} \mid \theta_k) \end{aligned}$$

(same family, different parameters)

- These are called **class-conditional distributions**
- Idea is very intuitive: consider M/F by height

Class posterior

- We want to classify a data-vector, i.e. determine it's class
- Using Bayes' rule:

$$P(Y = 1 \mid \mathbf{X}) \quad = \quad \frac{p(\mathbf{X} \mid Y = 1)P(Y = 1)}{p(\mathbf{X} \mid Y = 1)P(Y = 1) + p(\mathbf{X} \mid Y = 0)P(Y = 0)}$$

- If we
  – Assume some prior on class membership and
  – Can estimate the two class-conditional pdfs/pmfs

then we can classify data-points

Inference

- Intuitively
  - We have two groups, labelled by Y=0, Y=1
  - We want the parameters for each group
  - We can just estimate the parameters for all datapoints having Y = k
- This can be described more formally in likelihood terms
- We'll start with a discrete classifier