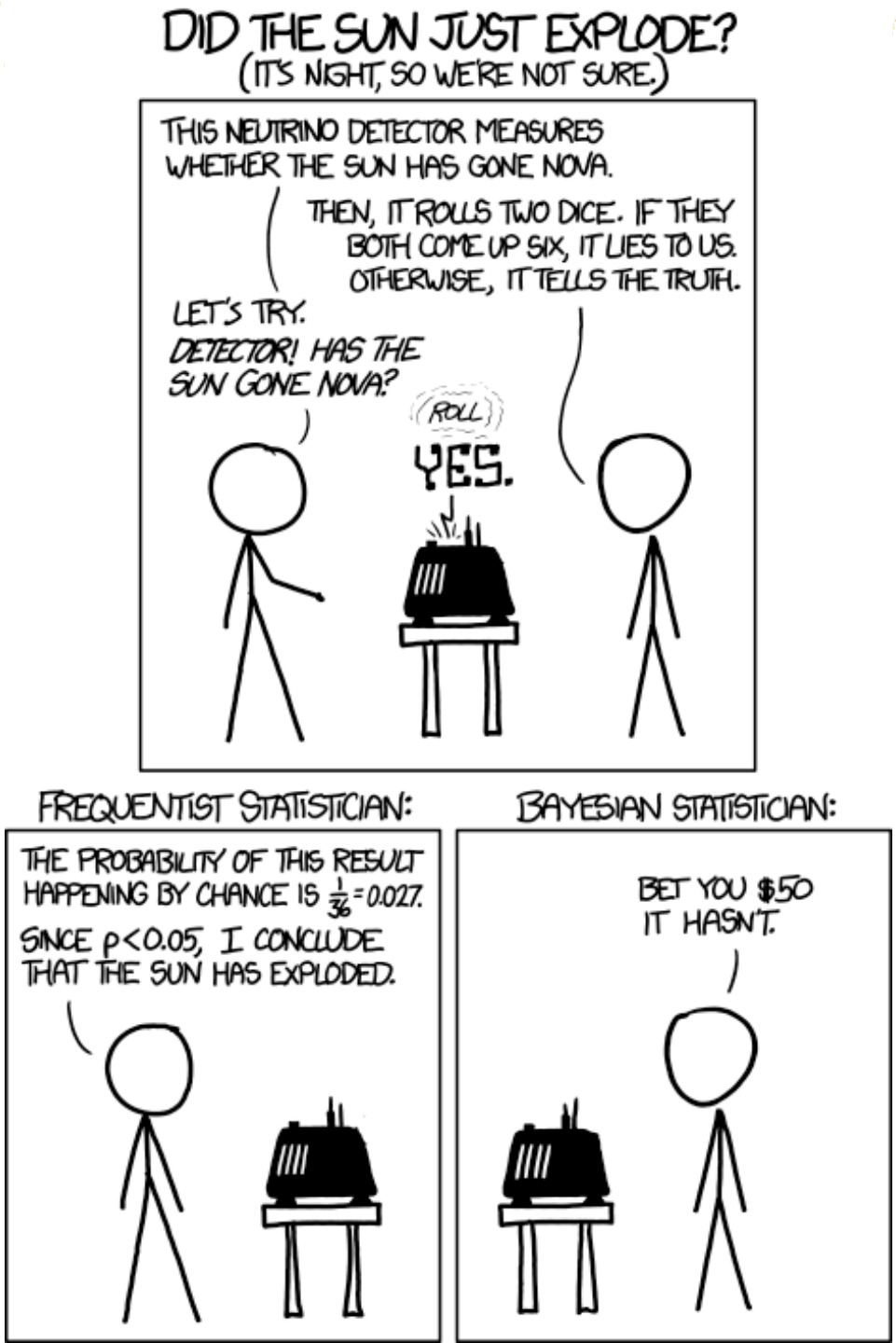
UNIVERSITY of WASHINGTON

Data Science UW Methods for Data Analysis

Introduction to Neural Networks
Extra Topics
Nick McClure


DID THE SUN JUST EXPLODE?
(IT'S NIGHT, SO WE'RE NOT SURE.)
THIS NEUTRINO DETECTOR MEASURES WHETHER THE SUN HAS GONE NOVA.
THEN, IT ROLLS TWO DICE. IF THEY BOTH COME UP SIX, IT LIES TO US. OTHERWISE, IT TELLS THE TRUTH.
LET'S TRY.
DETECTOR! HAS THE SUN GONE NOVA?
ROLL
YES.
FREQUENTIST STATISTICIAN:
THE PROBABILITY OF THIS RESULT HAPPENING BY CHANCE IS $\frac{1}{36} = 0.027$.
SINCE $p < 0.05$, I CONCLUDE THAT THE SUN HAS EXPLODED.
BAYESIAN STATISTICIAN:
BET YOU $50 IT HASN'T.

Review

- NLP Methods:
  - Text normalization
  - Word Clouds
  - Documents and Corpus’
  - Term Document Matrices (and vice versa)
  - TF-IDF
  - Naïve Bayes

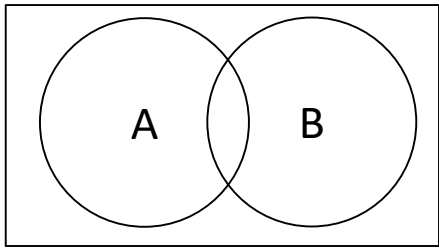
Topics

- Bayesian Statistics
  - Bayesian Inference
  - MCMC distributions
- Computational Statistics
  - Bootstrapping
  - Generating P-values via simulating the Null Hypothesis
  - Cross Validation

Remember Bayes Law: $P(A|B) = P(B|A)\frac{P(A)}{P(B)}$

- Important points to make:
  - Tests are not the event. We have a disease test, which is different than the event of actually having the disease.
  - Tests are flawed. Tests have false positives and false negatives.
  - Tests return test probabilities, not the event probabilities.
  - False positives skew results.
    - E.g. If fraud is rare, then the likelihood of a positive result of fraud is probably due to a false positive

Revisiting Conditional Probability



Fun Fact #1 $$P(B) = P(B \cap A) + P(B \cap not\ A)$$

Fun Fact #2 $$P(B|E) = \frac{P(B \cap E)}{P(E)}$$ OR $$P(B|E)P(E) = P(B \cap E)$$

Combining these results in:

$$P(B) = P(B|A)P(A) + P(B|not\ A)P(not\ A)$$

Another way to write Bayes Law:

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

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

If $P(B) = P(B|A)P(A) + P(B|not\ A)P(not\ A)$

Then $P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|not\ A)P(not\ A)}$

Why? Because usually P(B) is hard to estimate.

A Simpler Way to Write Bayes Law:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|not\ A)P(not\ A)}$$

Or $$P(A|B) = k \cdot P(B|A)P(A)$$

Or $$P(A|B) \propto P(B|A)P(A)$$

Posterior Distribution

The Likelihood

Prior Distribution

Interpretation with Modeling

$$P(A|B) \propto P(B|A)P(A)$$

Posterior ∝ Likelihood * Prior

- As this applies to parameters in a model (partial slopes, intercept, error distributions, lasso constant,…) and the observed data:

$$P(parameters|data) \propto P(data|parameters)P(parameters)$$

- The idea is that we have a prior assumption about the behavior of the parameters (the prior), we then produce a model which tells us the probability of observing our data, and then we come up with a new probability of our parameters.

Interpretation with Modeling

$$P(parameters|data) \propto P(data|parameters)P(parameters)$$

- Steps:
  - Identify data relevant to the research question. E.g.: what are the measurement scales of the data? (Helps set uninformative priors)
  - Define a descriptive model for the data. E.g.: pick a linear model formula.
  - Specify a prior distribution of the parameters. E.g. We think the error in the linear model is Normally distributed as $N(0, \sigma^2)$.
  - Use the Bayesian inference formula (above) to re-assess parameter probabilities.
  - Optionally, iterate if more data is observed.

An overused example, but for good reason.

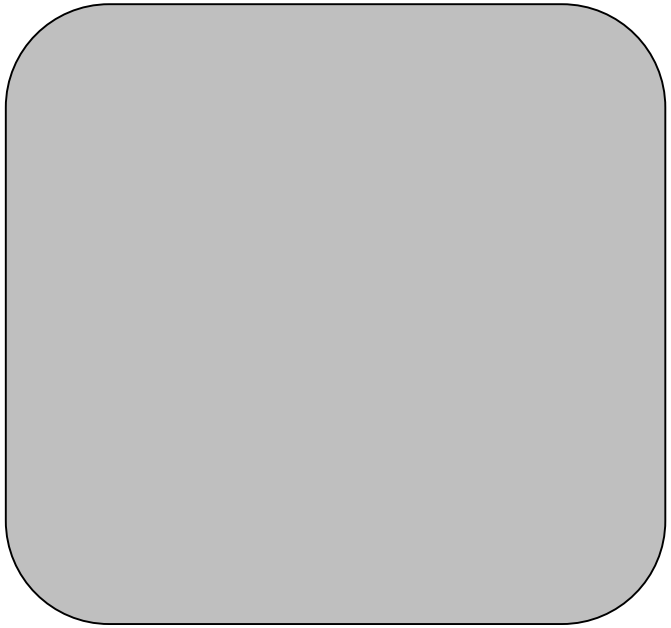
> Tasked with identifying where on a target archery board the bullseye is. But we can only see the back of the target and where the arrows puncture through as a marksman fires at it.

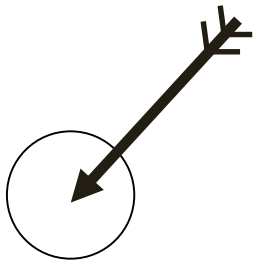
Back of target: What we see.

Front of target?

Archery Example

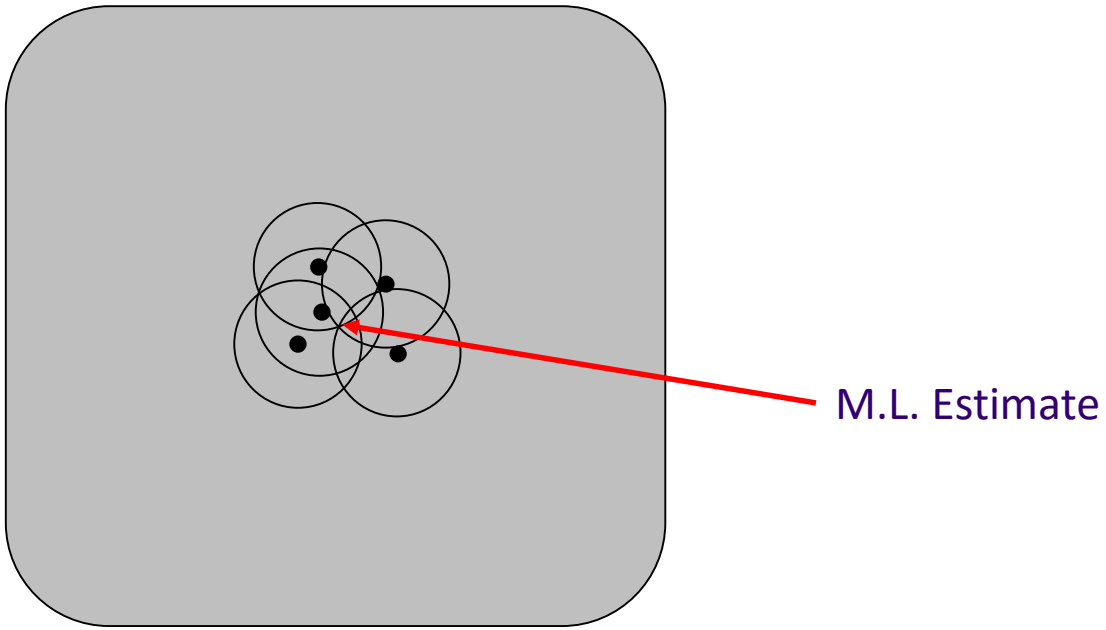
> We are told that a marksman is firing at it and they are always within 10 centimeters of the target 95% of the time.





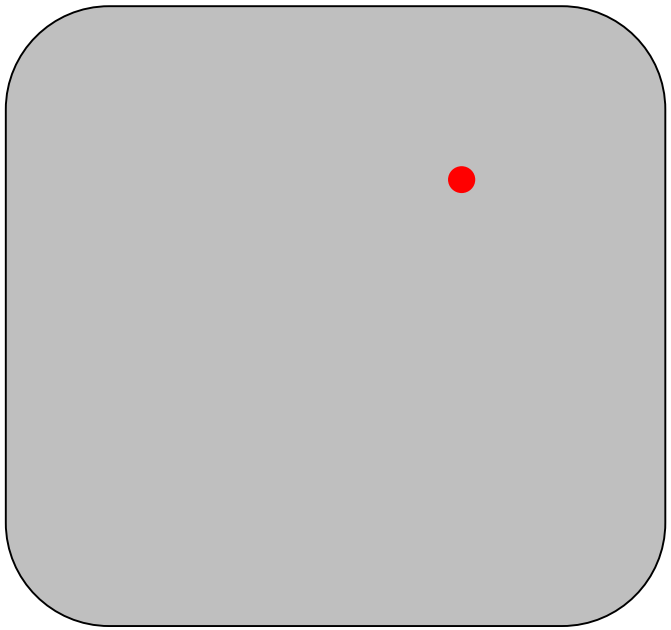
Archery Example

- Here are the archer's first 5 shots with a 10 cm radius around it.
- A frequentist observes the maximum likelihood point as the best guess.



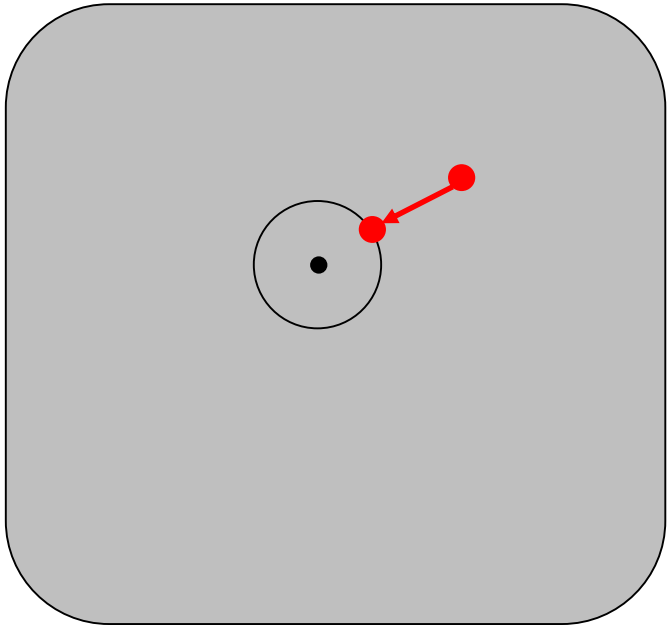
Archery Example

> A Bayesian approach is to create a ‘Prior’, or a previous belief of where the target is. (Red point)



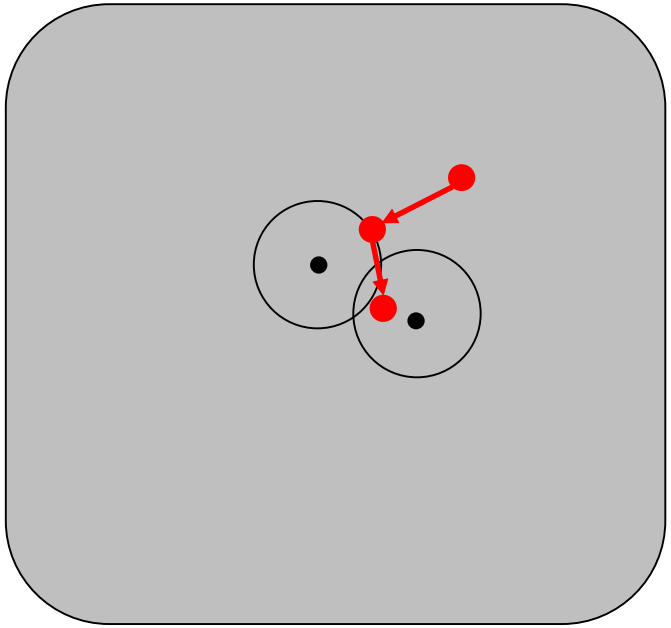
Archery Example

- As the next arrow fires, we update our prior via the posterior distribution.
- We iterate on this until our final target is chosen.



Archery Example

- As the next arrow fires, we update our prior via the posterior distribution.
- We iterate on this until our final target is chosen.

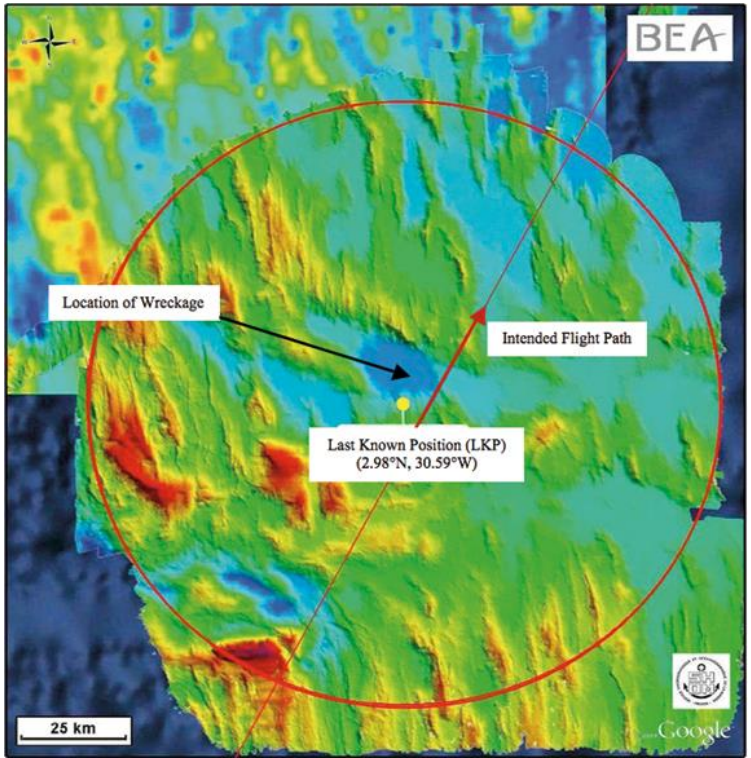


- This procedure is called Bayesian inference.
- How is this used in the real world? (Remember lost keys ex.)

Bayesian Inference Successes

$$P(parameters|data) \propto P(data|parameters)P(parameters)$$

- In practice, Bayesian inference has been used successfully to find lost planes. E.g. Air France 447
- https://www.informs.org/ORMS-Today/Public-Articles/August-Volume-38-Number-4/In-Search-of-Air-France-Flight-447



Frequentist Estimation of Heads in a Coin Flip

- We will flip a coin N times. We count the number of heads and want to estimate the p(H). E.g. if it is a fair coin, we would expect (with enough trails) that we would estimate p(H) = 0.5.
- Frequentist probability:
  - Most likely (maximum likelihood) answer would be:

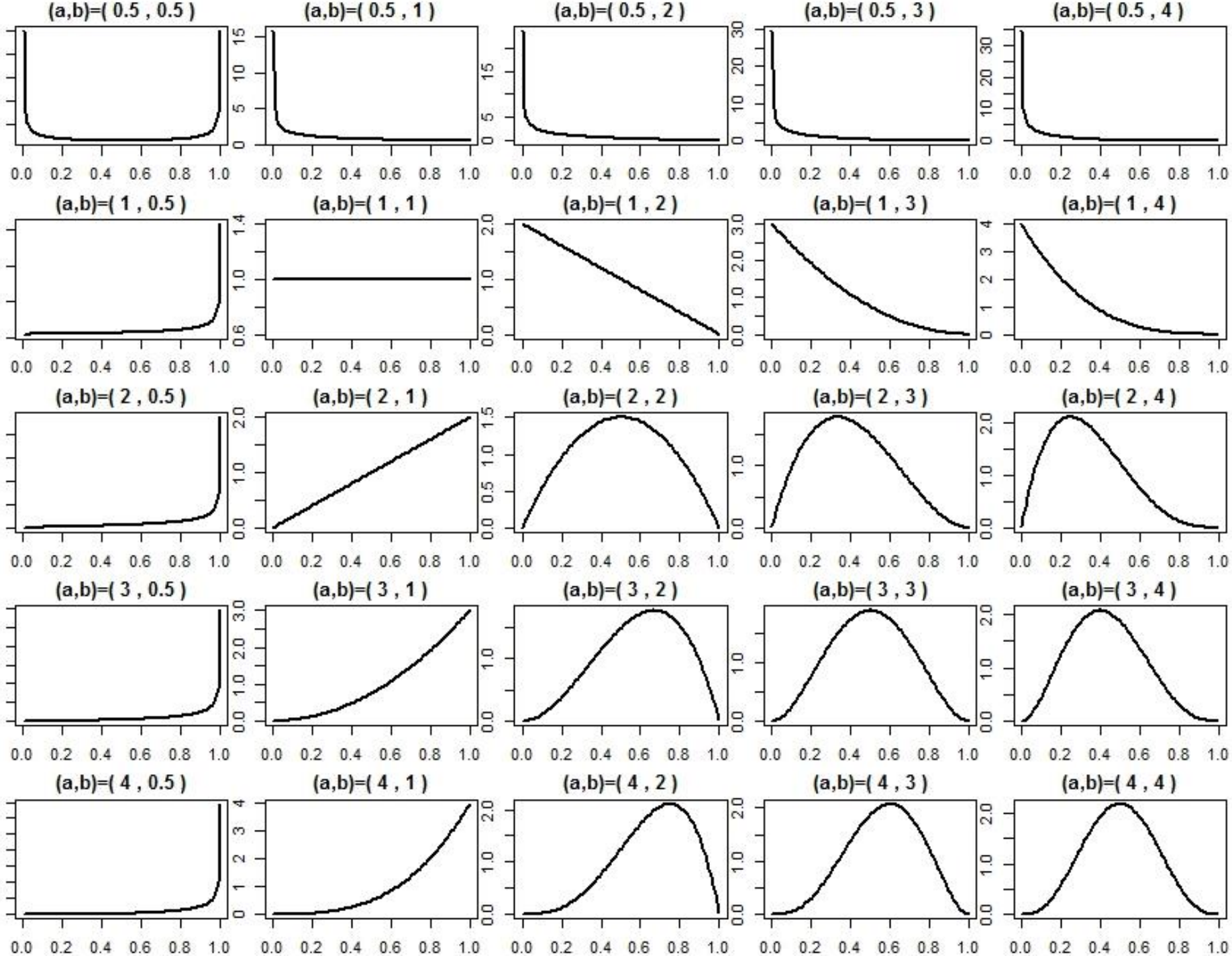
$$p(H) = \frac{n(H)}{N}$$

Bayesian Estimation of Heads in a Coin Flip

- We will need to define a Prior probability for the estimation of p(H).
- The best choice in this case is a Beta distribution:

$$f(x) = x^{a-1}(1-x)^{b-1} \cdot (normalizing\ constant)$$

- a, b are constants that define the distribution (similar to how the mean and variance define different normals).
- X is only defined between 0 and 1.


(a,b)=(0.5 , 0.5)
(a,b)=(0.5 , 1)
(a,b)=(0.5 , 2)
(a,b)=(0.5 , 3)
(a,b)=(0.5 , 4)
(a,b)=(1 , 0.5)
(a,b)=(1 , 1)
(a,b)=(1 , 2)
(a,b)=(1 , 3)
(a,b)=(1 , 4)
(a,b)=(2 , 0.5)
(a,b)=(2 , 1)
(a,b)=(2 , 2)
(a,b)=(2 , 3)
(a,b)=(2 , 4)
(a,b)=(3 , 0.5)
(a,b)=(3 , 1)
(a,b)=(3 , 2)
(a,b)=(3 , 3)
(a,b)=(3 , 4)
(a,b)=(4 , 0.5)
(a,b)=(4 , 1)
(a,b)=(4 , 2)
(a,b)=(4 , 3)
(a,b)=(4 , 4)

Bayesian Estimation of a Coin Flip Probability

$$P(parameters|data) = P(data|parameters)\frac{P(parameters)}{P(data)}$$

$$f(x) = x^{a-1}(1-x)^{b-1} \cdot (normalizing\ constant)$$

- After we choose a prior, we compute the posterior:

$$Posterior = Likelihood\frac{Prior}{P(data)}$$

- Always a problem estimating the P(data). So…

$$Posterior = Likelihood\frac{Prior}{P(data|all\ parameters)}$$

$$Posterior = Likelihood\frac{Prior}{\sum P(data|theta)}$$

- R demo

Bayesian Estimation of Multiple Parameters

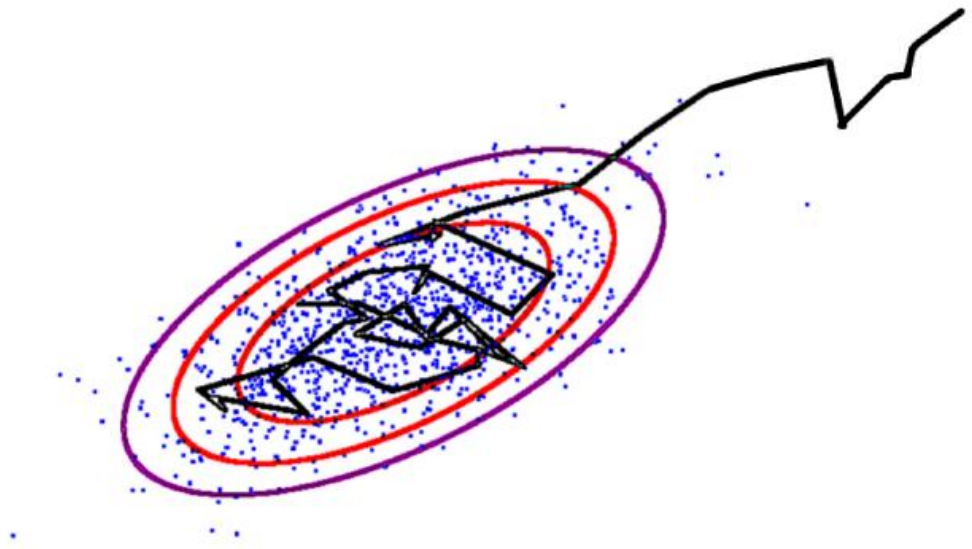
- We only had one parameter to estimate for the coin flip example, p(H).
- We created a grid to check (seq(0.01,0.99,length=100) and used this to calculate the p(data), by checking all the values.
- What if we had several parameters? If we had 6 parameters with a length 100 grid… = 100^6 = 1,000,000,000,000 = 1 trillion points to check.
- Maybe we don't have to sample everything, just enough points to understand and estimate the distribution of how p(data) behaves under the 6 parameters.

Introducing the Metropolis (Hastings) Algorithm

- The Metropolis algorithm tells us how to explore a space of parameters, according to an unknown distribution.
- Algorithm:
  - 1. Pick a starting point in your parameter space and evaluate it according to your model. (find p(data)).
  - 2. Choose a nearby point randomly and evaluate this point.
    - If the value of the new point is greater than your previous points, accept new point and move there.
    - If the value of the new point is less than your previous point, only accept with probability according to the ratio: value(new) / value(old).
  - 3. Repeat # 2 many times.
- Eventually the distribution of the points you accept will represent the underlying distribution.
- We only have to visit N points, not 1 Trillion points.

The Metropolis Algorithm and MCMC

- Terminology:
  - Assessing a distribution via representative sampled points is called a Monte-Carlo approximation.
  - Any method that uses only the prior state to find the next state is called a Markov-Chain.
- So, using the Metropolis Algorithm to approximate the p(data) is also called Markov-Chain Monty-Carlo, or MCMC.
- R-Demo.



MCMC For Linear Regression

- We want to estimate the distribution of the slopes, the intercept, and the error's standard deviation.

$$y_i = mx_i + b + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma)$$

- We start with a prior on the three parameters (m, b, sigma). Usually quite uninformative priors (uniform distribution).
- We choose parameter points (m, b, sigma) according to the MCMC algorithm and see the resulting distribution.

- R demo.

Computational Statistics

- Why?
  - Statistics is hard! Deriving formulas and keeping track of relationships takes time.
  - Use that time computationally. Let your computer do the work.

Small Datasets

- Why worry when approximations apply and don't apply when your data set isn't large enough?
- The idea is to "create data and samples from nothing".
- Key assumption:
  - Our sample, however small, was created by randomly sampling.
  - This means that our sample is 'representative' of the population.
- We create more data by sampling our sample **WITH** replacement.

Bootstrapping

- Bootstrapping is called as such because of a passage is *Ulysses:*
  - "There were others who had forced their way to the top from the lowest rung only by the aid of their bootstraps…"
- We treat these resamples of the data as representatives of the whole population. In fact, under bootstrapping, we think of the population as the set of infinite resamples of the smaller sample.
- E.g. we poll the students in our class for estimates of the instructors age.
- The sample is small, so an error bound on the mean is large. We can reduce this by bootstrapping

- R-demo

Bootstrapping for Linear Modeling

- Just like we sometimes won't have enough data points for a mean/standard deviation, we sometimes don't have enough points to find the error in fitting a line.
- Two methods:
  - Bootstrapping the selection of points. (Parametric)
  - Bootstrapping the residuals. (non-Parametric)

- R demo

Estimating Probabilities With Sampling

- If we can generate or sample large enough data sets, we should be able to use that data, as is, to estimate p-values.
- We can simulate hypothesis testing as well.
  - Simulate the null hypothesis many times and arrive at a null-hypothesis distribution for comparison.
- R-demo

Evaluating Binary Classifiers (Logistic Regression)

- Logistic Regression outputs a probability of success between 0 – 1.
- The usual cutoff we have been using is 0.5:
  - If f(y) > 0.5, predict success or 1
  - If f(y) < 0.5, predict failure or 0
- Remember that:
  - True Positives: Predict 1 and Actual 1
  - True Negatives: Predict 0 and Actual 0
  - False Negatives: Predict 0 and Actual 1
  - False Positives: Predict 1 and Actual 0
- We compose results in a 'Confusion Matrix':

	Predicted Success	**Predicted Failure**
Actual Success	True Positives	False Negatives
Actual Failure	False Positives	True Negatives

Confusion Matrix

> We are also interested in the Specificity, Sensitivity and Predictive Values:

		Condition (as determined by "Gold standard")		
		Condition Positive	Condition Negative	
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)	Positive predictive value = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Test Outcome Positive}}$
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**	Negative predictive value = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Test Outcome Negative}}$
		Sensitivity = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Condition Positive}}$	**Specificity** = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Condition Negative}}$	

R-Demo

Image Source: https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Confusion Matrix

> We are also interested in the Specificity, Sensitivity and Predictive Values:

		Condition (as determined by "Gold standard")		
		Condition Positive	Condition Negative	
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)	Positive predictive value = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Test Outcome Positive}}$
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**	Negative predictive value = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Test Outcome Negative}}$
		Sensitivity = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Condition Positive}}$	**Specificity** = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Condition Negative}}$	

R-Demo

Image Source: https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Model Validation

- So far we have only evaluated predictions on the same set of data we have been training our models on.
- Normally, a smaller set is held out, and the prediction on that set is compared to the actuals.
- This is called a train-test split.
  - 90-10 Split (90% of data randomly used in training, rest for testing)
  - 80-20 Split
  - 70-30 Split
  - …

- R-demo

Cross Validation

- Concerns with Train-Test split:
  - Exact test results could be dependent on which random 10% we chose.
  - Cannot estimate error of accuracy.
- Types of Cross Validation:
  - K-fold cross validation: (1) Partition your data randomly into k-subsets. (2) Create k-models, testing on a different subset each time.
    - Note that k is bounded: $2 < k < N$
    - K=2: is the same as a 50-50 test-train split.
    - K=N: is the same as….
  - Leave-One-Out-Cross-Validation (LOOCV).

- R-demo

Cross Validation

- How to choose k? Arbitrary recommendations put it at k=5 or 10.
- There is a bias-variance trade off in the results.
  - If k is large, the training set is large and the model will have less bias in the outcomes. But the predictions will have larger variance.
  - If k is small, the training set is smaller and the model will have more bias in the outcomes. But we will be more confident in the predictions.
- Also, there is a computational consideration when k gets larger, we have to compute k models.

Another Application of Bayes Theorem

- Naïve Bayes Classification.
- Let's say we want to test a hypothesis, h_0:

$$P(h_0|Data) = P(Data|h_0)\frac{P(h_0)}{P(Data)}$$

- We pick either the null or alternative, based on which has the highest probability given the data.

Another Application of Bayes Theorem

$$P(h_0|Data) = P(Data|h_0)\frac{P(h_0)}{P(Data)}$$

- We want to predict if a consumer will buy an ad:
- Null hypothesis is yes (or could be no)
- We have observed 10 outcomes in the past:

Age	Income	Student	Credit	Buys Ad
35	Medium	Yes	Fair	Yes
30	High	No	Average	No
40	Low	Yes	Good	No
35	Medium	No	Fair	Yes
45	Low	No	Fair	Yes
35	High	No	Excellent	Yes
35	Medium	No	Good	No
25	Low	No	Good	No
28	High	No	Average	No
35	Medium	Yes	Average	Yes

Another Application of Bayes Theorem

Age	Income	Student	Credit	Buys Ad
35	Medium	Yes	Fair	Yes
30	High	No	Average	No
40	Low	Yes	Good	No
35	Medium	No	Fair	Yes
45	Low	No	Fair	Yes
35	High	No	Excellent	Yes
35	Medium	No	Good	No
25	Low	No	Good	No
28	High	No	Average	No
35	Medium	Yes	Average	Yes

$$P(h_0|Data) = P(Data|h_0)\frac{P(h_0)}{P(Data)}$$

New Data: Consumer is 35 years old
And has a medium income

- P(buys Ad)=0.5, P(not buy Ad) = 0.5
- P(35yrs & med income) = 4/10 = 0.4
- P(35yrs & med income | buys ad) = 3/5 = 0.6
- P(35yrs & med income | not buy ad) = 1/5 = 0.2

- P(buy | demographics) = $P(demographics|buy)\frac{P(buy)}{P(demographics)}$
- =0.6 * (0.5 / 0.4) = 0.75
- Similarly, P(not buy | demographics) = 0.2 * (0.5 / 0.4)=0.25

Further interesting links

- Microsoft Research 2008: Bayesian Inference in Traffic Patterns. http://research.microsoft.com/pubs/101948/TAinfer.pdf
- Google Analytics (Multi armed bandit experiments for AB testing): https://support.google.com/analytics/answer/2844870?hl=en
- Google Research: Voice recognition: http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/37567.pdf
- CNA: Forecasting insurance loss: http://www.casact.org/education/annual/2010/handouts/C4-Zhang.pdf
- Swype texting: http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/39190.pdf

Assignment

- Finish your project.
  - No extensions.