Calibration and Prediction with Gaussian Process Emulators - Exercise Prep

J. Coleman, R. Wolpert, S. Bass

January 4, 2018

Overview of Analysis

Designing the Training Points - Latin Hypercube

Training and Validating GP Emulators

GP Introduction

Multivariate Output - PCA

Calibration

Intro To Bayesian Analysis

Overview

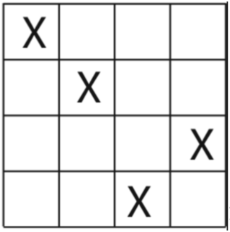
Designing the Training Points - Latin Hypercube

Training and Validating GP Emulators
- GP Introduction
- Multivariate Output - PCA

Calibration
- Intro To Bayesian Analysis

Latin Hypercube Design

- Ensures that there is only one design point in each row and column
- Every design point is in exactly one "bin" for each dimension

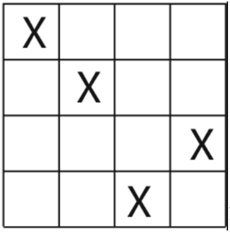


[1]

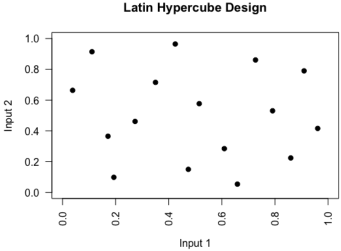
[1] https://upload.wikimedia.org/wikipedia/commons/f/fb/LHSsampling.png

Latin Hypercube Design

- Ensures that there is only one design point in each row and column
- Every design point is in exactly one “bin” for each dimension



[1]



---

[1] https://upload.wikimedia.org/wikipedia/commons/f/fb/LHSsampling.png

Exercise

- Create a Latin Hypercube design of 20 points in 2 dimensions
- Plot the result

Overview

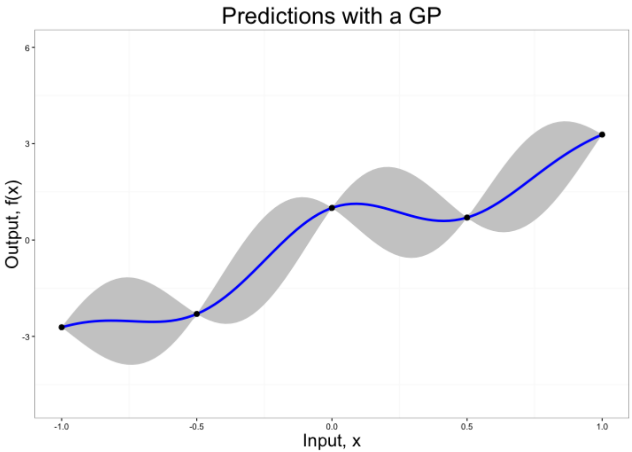
Designing the Training Points - Latin Hypercube

Training and Validating GP Emulators
- GP Introduction
- Multivariate Output - PCA

Calibration
- Intro To Bayesian Analysis

GPs In Action

Prediction = mean + uncertainty



The gray bands are 95% confidence intervals.

What is a GP?

Technical Terms: A Gaussian Process (GP) is a stochastic process Y indexed by $\mathbf{x} \in \mathcal{X}$ such that realizations are jointly Multivariate Normal.

What is a GP?

Technical Terms: A Gaussian Process (GP) is a stochastic process Y indexed by $\mathbf{x} \in \mathcal{X}$ such that realizations are jointly Multivariate Normal.

Ok, but what does that mean?

What is a GP?

Technical Terms: A Gaussian Process (GP) is a stochastic process Y indexed by $\mathbf{x} \in \mathcal{X}$ such that realizations are jointly Multivariate Normal.

Ok, but what does that mean?

- If given $x_1, \ldots x_n$ locations, we can find the joint distribution for outputs $(Y(x_1), \ldots Y(x_n))$ - and it's Multivariate Normal

What is a GP?

Technical Terms: A Gaussian Process (GP) is a stochastic process Y indexed by $\mathbf{x} \in \mathcal{X}$ such that realizations are jointly Multivariate Normal.

Ok, but what does that mean?

- If given $x_1, \dots x_n$ locations, we can find the joint distribution for outputs $(Y(x_1), \dots Y(x_n))$ - and it's Multivariate Normal
- Our function $Y()$ is random, but we can make guesses based on input x and other observed values of Y.

Some Housekeeping

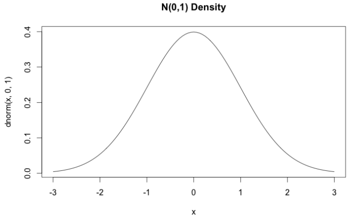
A random variable $Y \in \mathbb{R}^1$ is said to come from a **Gaussian** or **Normal** distribution with mean μ and variance σ^2 if it has the probability density function (pdf)

$$Y \sim N(\mu, \sigma^2) \Rightarrow p(Y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma}(y-\mu)^2}$$

Some Housekeeping

A random variable $Y \in \mathbb{R}^1$ is said to come from a **Gaussian** or **Normal** distribution with mean μ and variance σ^2 if it has the probability density function (pdf)

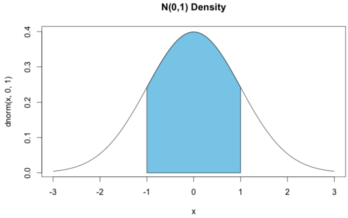
$$Y \sim N(\mu, \sigma^2) \Rightarrow p(Y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma}(y-\mu)^2}$$



Some Housekeeping

A random variable $Y \in \mathbb{R}^1$ is said to come from a **Gaussian** or **Normal** distribution with mean μ and variance σ^2 if it has the probability density function (pdf)

$$Y \sim N(\mu, \sigma^2) \Rightarrow p(Y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma}(y-\mu)^2}$$



$P(Y \in [a, b]) = \int_a^b p(Y \mid \mu, \sigma^2) dy$

Some Housekeeping

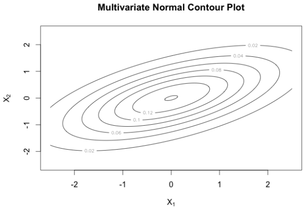
A random variable $\mathbf{Y} \in \mathbb{R}^n$ is said to come from a **Multivariate Gaussian** or **Multivariate Normal** with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ if has pdf

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \sim MVN(\boldsymbol{\mu}, \Sigma) \Rightarrow p(\mathbf{Y} \mid \boldsymbol{\mu}, \Sigma) = |2\pi\Sigma|^{-1/2} e^{-\frac{1}{2}(\mathbf{Y}-\boldsymbol{\mu})'\Sigma^{-1}(\mathbf{Y}-\boldsymbol{\mu})}$$

Some Housekeeping

A random variable $\mathbf{Y} \in \mathbb{R}^n$ is said to come from a **Multivariate Gaussian** or **Multivariate Normal** with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ if has pdf

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \sim MVN(\boldsymbol{\mu}, \Sigma) \Rightarrow p(\mathbf{Y} \mid \boldsymbol{\mu}, \Sigma) = |2\pi\Sigma|^{-1/2} e^{-\frac{1}{2}(\mathbf{Y}-\boldsymbol{\mu})'\Sigma^{-1}(\mathbf{Y}-\boldsymbol{\mu})}$$



Some Housekeeping

Multivariate Normal properties

If $\mathbf{Y} \sim MVN(\boldsymbol{\mu}, \Sigma)$:

Some Housekeeping

Multivariate Normal properties

If $\mathbf{Y} \sim MVN(\boldsymbol{\mu}, \Sigma)$:

- $\mathbf{Y}_i \sim N(\boldsymbol{\mu}_i, \Sigma_{ii})$

Some Housekeeping

Multivariate Normal properties

If $\mathbf{Y} \sim MVN(\boldsymbol{\mu}, \Sigma)$:

- $\mathbf{Y}_i \sim N(\boldsymbol{\mu}_i, \Sigma_{ii})$
- $Cov(\mathbf{Y}_i, \mathbf{Y}_j) = \Sigma_{ij} = \Sigma_{ji}$

Some Housekeeping

Multivariate Normal properties

If $\mathbf{Y} \sim MVN(\boldsymbol{\mu}, \Sigma)$:

- $\mathbf{Y}_i \sim N(\boldsymbol{\mu}_i, \Sigma_{ii})$
- $Cov(\mathbf{Y}_i, \mathbf{Y}_j) = \Sigma_{ij} = \Sigma_{ji}$
- $Cov(\mathbf{Y}_i, \mathbf{Y}_j) = 0 \Leftrightarrow \mathbf{Y}_i$ and $\mathbf{Y}_j$ are independent (special for Gaussians)

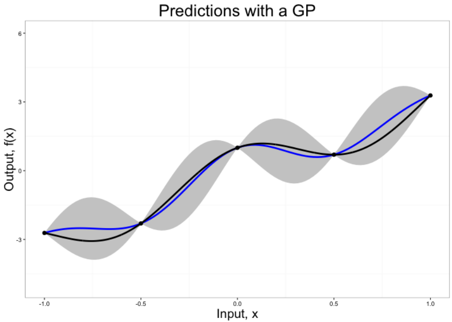
Some Housekeeping

Multivariate Normal properties

If $\mathbf{Y} \sim MVN(\boldsymbol{\mu}, \Sigma)$:

- $\mathbf{Y}_i \sim N(\boldsymbol{\mu}_i, \Sigma_{ii})$
- $Cov(\mathbf{Y}_i, \mathbf{Y}_j) = \Sigma_{ij} = \Sigma_{ji}$
- $Cov(\mathbf{Y}_i, \mathbf{Y}_j) = 0 \Leftrightarrow \mathbf{Y}_i$ and $\mathbf{Y}_j$ are independent (special for Gaussians)
- Σ must be symmetric, positive definite

Don't Lose Sight!



The prediction and estimated errors are just going to come from multivariate normals

Remember the Definition

- Formally, a Gaussian Process is a stochastic process Y indexed by $\mathbf{x} \in \mathcal{X}$ such that realizations are jointly Multivariate Normal.

Remember the Definition

- Formally, a Gaussian Process is a stochastic process Y indexed by $\mathbf{x} \in \mathcal{X}$ such that realizations are jointly Multivariate Normal.
- I.e. given locations $\mathbf{x}_1, \ldots \mathbf{x}_n$, then $Y(\mathbf{x}_1), \ldots Y(\mathbf{x}_n)$ are MVN

Remember the Definition

- Formally, a Gaussian Process is a stochastic process Y indexed by $\mathbf{x} \in \mathcal{X}$ such that realizations are jointly Multivariate Normal.
- I.e. given locations $\mathbf{x}_1, \ldots \mathbf{x}_n$, then $Y(\mathbf{x}_1), \ldots Y(\mathbf{x}_n)$ are MVN
- It is completely determined by a **mean function** $\mu(\cdot)$ and a positive-definite **covariance function** $c(\cdot, \cdot)$ through

$$\mu_i = \mu(\mathbf{x}_i) \qquad \Sigma_{ij} = c(\mathbf{x}_i, \mathbf{x}_j)$$

Remember the Definition

- Formally, a Gaussian Process is a stochastic process Y indexed by $\mathbf{x} \in \mathcal{X}$ such that realizations are jointly Multivariate Normal.
- I.e. given locations $\mathbf{x}_1, \ldots \mathbf{x}_n$, then $Y(\mathbf{x}_1), \ldots Y(\mathbf{x}_n)$ are MVN
- It is completely determined by a **mean function** $\mu(\cdot)$ and a positive-definite **covariance function** $c(\cdot, \cdot)$ through

$$\mu_i = \mu(\mathbf{x}_i) \qquad \Sigma_{ij} = c(\mathbf{x}_i, \mathbf{x}_j)$$

- We can think of a GP as a distribution over functions

A Concrete example

- Let points $\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_n \in \mathcal{X}$, where $\mathcal{X}$ is the input space.
- Let $Y(\cdot) \sim GP(\mu(\cdot), c(\cdot, \cdot))$. Then

$$\begin{pmatrix} Y(\mathbf{x}_1) \\ Y(\mathbf{x}_2) \\ \vdots \\ Y(\mathbf{x}_n) \end{pmatrix} \sim MVN \left[\begin{pmatrix} \mu(\mathbf{x}_1) \\ \mu(\mathbf{x}_2) \\ \vdots \\ \mu(\mathbf{x}_n) \end{pmatrix}, \quad \begin{pmatrix} c(\mathbf{x}_1, \mathbf{x}_1) & \dots & c(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ c(\mathbf{x}_n, \mathbf{x}_1) & \dots & c(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix} \right]$$

A Concrete example

- Let points $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n \in \mathcal{X}$, where $\mathcal{X}$ is the input space.
- Let $Y(\cdot) \sim GP(\mu(\cdot), c(\cdot, \cdot))$. Then

$$\begin{pmatrix} Y(\mathbf{x}_1) \\ Y(\mathbf{x}_2) \\ \vdots \\ Y(\mathbf{x}_n) \end{pmatrix} \sim MVN \left[\begin{pmatrix} \mu(\mathbf{x}_1) \\ \mu(\mathbf{x}_2) \\ \vdots \\ \mu(\mathbf{x}_n) \end{pmatrix}, \quad \begin{pmatrix} c(\mathbf{x}_1, \mathbf{x}_1) & \ldots & c(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ c(\mathbf{x}_n, \mathbf{x}_1) & \ldots & c(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix} \right]$$

- Examples: $\mu(\cdot) \equiv 0$; $\mu(\cdot) \equiv \mu$; $\mu(\mathbf{x}) \equiv \sum_i \mathbf{x}_i \beta, \ldots,$
- $c(\cdot, \cdot)$ are special functions that give rise to symmetric positive definite matrices

Conditional Normal Theory

Let $Y(\mathbf{x_d}) = [Y(\mathbf{x}_{d_1}), \ldots, Y(\mathbf{x}_{d_n})]' \in \mathbb{R}^n$, and similarly $c(\mathbf{x_d}, \mathbf{x_d}) \in \mathbb{R}^{n \times n}$

$$\begin{pmatrix} Y(\mathbf{x}^*) \\ Y(\mathbf{x_d}) \end{pmatrix} \sim MVN \left[\begin{pmatrix} \mu(\mathbf{x}^*) \\ \mu(\mathbf{x_d}) \end{pmatrix}, \quad \begin{pmatrix} c(\mathbf{x}^*, \mathbf{x}^*) & c(\mathbf{x}^*, \mathbf{x_d}) \\ c(\mathbf{x_d}, \mathbf{x}^*) & c(\mathbf{x_d}, \mathbf{x_d}) \end{pmatrix} \right]$$

Conditional Normal Theory

Let $Y(\mathbf{x_d}) = [Y(\mathbf{x}_{d_1}), \ldots, Y(\mathbf{x}_{d_n})]' \in \mathbb{R}^n$, and similarly $c(\mathbf{x_d}, \mathbf{x_d}) \in \mathbb{R}^{n \times n}$

$$\begin{pmatrix} Y(\mathbf{x}^*) \\ Y(\mathbf{x_d}) \end{pmatrix} \sim MVN \left[\begin{pmatrix} \mu(\mathbf{x}^*) \\ \mu(\mathbf{x_d}) \end{pmatrix}, \quad \begin{pmatrix} c(\mathbf{x}^*, \mathbf{x}^*) & c(\mathbf{x}^*, \mathbf{x_d}) \\ c(\mathbf{x_d}, \mathbf{x}^*) & c(\mathbf{x_d}, \mathbf{x_d}) \end{pmatrix} \right]$$

then $Y(\mathbf{x}^*) \mid (Y(\mathbf{x_d}) = \mathbf{y}) \sim N(\mu^*, \Sigma^*)$ where

$$\mu^* = \boldsymbol{\mu}_Z + c(\mathbf{x}^*, \mathbf{x_d}) c(\mathbf{x_d}, \mathbf{x_d})^{-1} (\mathbf{y} - \boldsymbol{\mu}_Y)$$
$$\Sigma^* = c(\mathbf{x}^*, \mathbf{x}^*) - c(\mathbf{x}^*, \mathbf{x_d}) c(\mathbf{x_d}, \mathbf{x_d})^{-1} c(\mathbf{x_d}, \mathbf{x}^*)$$

Conditional Normal Theory

Let $Y(\mathbf{x_d}) = [Y(\mathbf{x}_{d_1}), \ldots, Y(\mathbf{x}_{d_n})]' \in \mathbb{R}^n$, and similarly $c(\mathbf{x_d}, \mathbf{x_d}) \in \mathbb{R}^{n \times n}$

$$\begin{pmatrix} Y(\mathbf{x}^*) \\ Y(\mathbf{x_d}) \end{pmatrix} \sim MVN \left[\begin{pmatrix} \mu(\mathbf{x}^*) \\ \mu(\mathbf{x_d}) \end{pmatrix}, \quad \begin{pmatrix} c(\mathbf{x}^*, \mathbf{x}^*) & c(\mathbf{x}^*, \mathbf{x_d}) \\ c(\mathbf{x_d}, \mathbf{x}^*) & c(\mathbf{x_d}, \mathbf{x_d}) \end{pmatrix} \right]$$

then $Y(\mathbf{x}^*) \mid (Y(\mathbf{x_d}) = \mathbf{y}) \sim N(\mu^*, \Sigma^*)$ where

$$\mu^* = \boldsymbol{\mu}_Z + c(\mathbf{x}^*, \mathbf{x_d}) c(\mathbf{x_d}, \mathbf{x_d})^{-1} (\mathbf{y} - \boldsymbol{\mu}_Y)$$
$$\Sigma^* = c(\mathbf{x}^*, \mathbf{x}^*) - c(\mathbf{x}^*, \mathbf{x_d}) c(\mathbf{x_d}, \mathbf{x_d})^{-1} c(\mathbf{x_d}, \mathbf{x}^*)$$

Important!: The diagonal of Σ^* gives the marginal variance of the predicted points. A Gaussian random variable has 95% probability of falling within $\pm$ 1.96 standard deviations from the mean.

GP Exercises

Assume you want to estimate the (unknown) function $y(x) = 3x + \cos(5x)$, but you only know y at $x = \{-1, -0.5, 0, 0.5, 1\}$

- Find and plot the mean and variance at all points $x^* = \{-1, -0.99, \ldots, 0.99, 1\}$. Compare to the truth
- Draw five possible sample paths using the above mean and variance

PCA - Orthogonality and Dimension Reduction

$$\mathbf{Y} = \mathbf{USV}' \quad \rightarrow \quad \mathbf{Z} = \mathbf{YV}$$

- All of the columns of **Z** are orthogonal, and thus independent (if **Z** is Multivariate Normal)

PCA - Orthogonality and Dimension Reduction

$$\mathbf{Y} = \mathbf{USV}' \quad \rightarrow \quad \mathbf{Z} = \mathbf{YV}$$

- All of the columns of $\mathbf{Z}$ are orthogonal, and thus independent (if $\mathbf{Z}$ is Multivariate Normal)
- The principal components can also tell us about the percent of variance explained by each each component

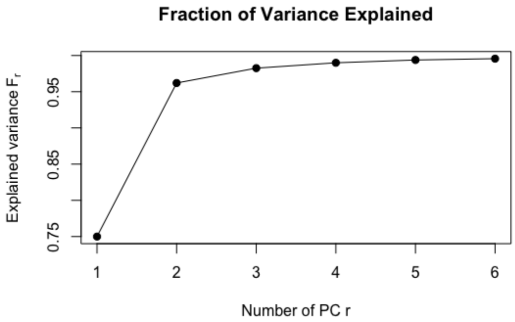
PCA - Orthogonality and Dimension Reduction

$$\mathbf{Y} = \mathbf{USV}' \quad \rightarrow \quad \mathbf{Z} = \mathbf{YV}$$

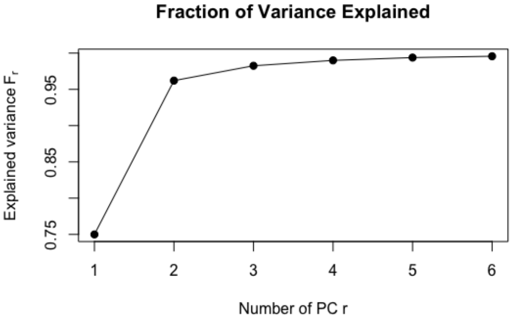
- All of the columns of $\mathbf{Z}$ are orthogonal, and thus independent (if $\mathbf{Z}$ is Multivariate Normal)
- The principal components can also tell us about the percent of variance explained by each each component
- Let $\{s_1, \ldots, s_p\} = diag(\mathbf{S})$. Then the fraction of variance explained by the first R columns of $\mathbf{Z}$ is

$$F_R = \frac{\sum_1^R s^2}{\sum_1^p s^2}$$

Look for the Elbow

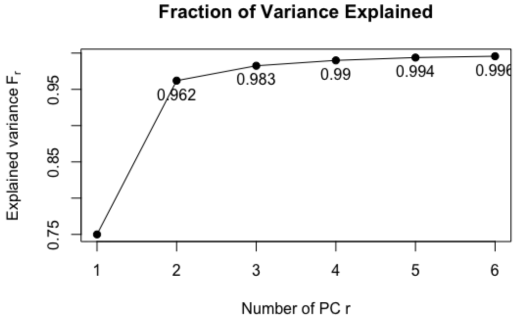


Look for the Elbow



Decide on number of components from “elbow,” or some threshold of fraction of variance explained

Look for the Elbow



Decide on number of components from “elbow,” or some threshold of fraction of variance explained

PCA Exercise

Load the *dev_indices* dataset

- Visualize fraction of variance explained, and choose a number of PCs
- Plot PC1 vs PC2. What is their correlation?

The Bayesian Paradigm

Suppose you have a coin, and you're not sure if it's biased. You flip it 10 times, and get 8 heads. Consider the following scenarios:

The Bayesian Paradigm

Suppose you have a coin, and you're not sure if it's biased. You flip it 10 times, and get 8 heads. Consider the following scenarios:

- You think it could be biased, so getting 8/10 is suspicious.

The Bayesian Paradigm

Suppose you have a coin, and you're not sure if it's biased. You flip it 10 times, and get 8 heads. Consider the following scenarios:

- You think it could be biased, so getting 8/10 is suspicious.
- You're *very* confident it's not biased, so getting 8/10 is definitely not convincing.

The Bayesian Paradigm

Suppose you have a coin, and you're not sure if it's biased. You flip it 10 times, and get 8 heads. Consider the following scenarios:

- You think it could be biased, so getting $8/10$ is suspicious.
- You're *very* confident it's not biased, so getting $8/10$ is definitely not convincing.
- You have no idea whether it's biased or not, so getting $8/10$ is pretty convincing.

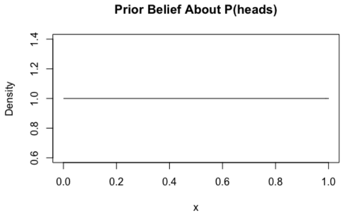
The Bayesian Paradigm

Suppose you have a coin, and you're not sure if it's biased. You flip it 10 times, and get 8 heads. Consider the following scenarios:

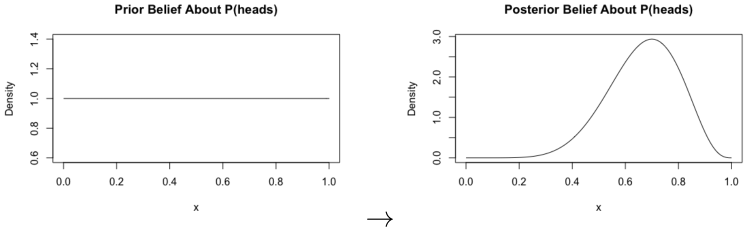
- You think it could be biased, so getting 8/10 is suspicious.
- You're *very* confident it's not biased, so getting 8/10 is definitely not convincing.
- You have no idea whether it's biased or not, so getting 8/10 is pretty convincing.

Bayesian analysis gives us a mathematical framework to insert our prior beliefs

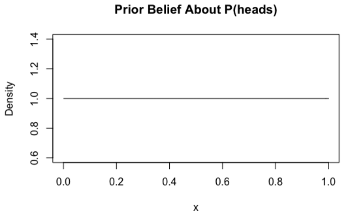
Bayesian Paradigm, In Pictures



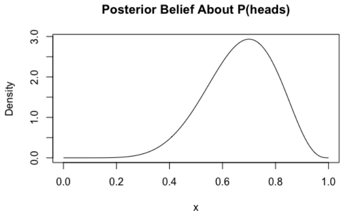
Bayesian Paradigm, In Pictures

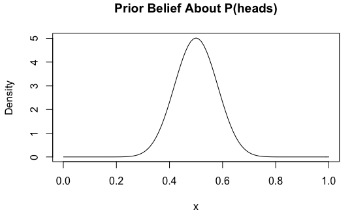


Bayesian Paradigm, In Pictures

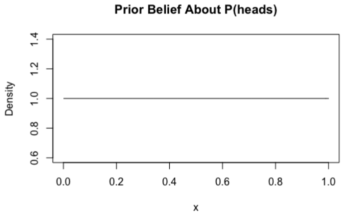


→

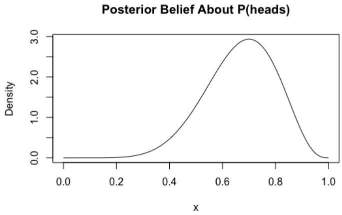


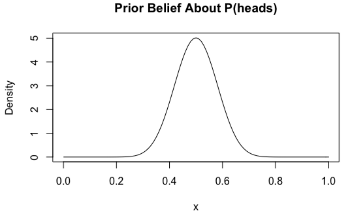


Bayesian Paradigm, In Pictures

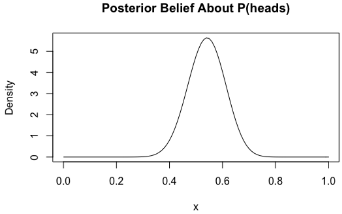


$\longrightarrow$





$\longrightarrow$



Proper Math

Let θ be a parameter of interest, upon which data y depends. Bayesian analysis has three main components

Proper Math

Let θ be a parameter of interest, upon which data y depends. Bayesian analysis has three main components

1. A chosen *prior* distribution on θ: $p(\theta)$

Proper Math

Let θ be a parameter of interest, upon which data y depends. Bayesian analysis has three main components

1. A chosen *prior* distribution on θ: $p(\theta)$
2. A chosen *likelihood* of y: $p(y \mid \theta)$

Proper Math

Let θ be a parameter of interest, upon which data y depends. Bayesian analysis has three main components

1. A chosen *prior* distribution on θ: $p(\theta)$
2. A chosen *likelihood* of y: $p(y \mid \theta)$
3. A resulting (of interest) *posterior* of θ: $p(\theta \mid y)$

Proper Math

Let θ be a parameter of interest, upon which data y depends. Bayesian analysis has three main components

1. A chosen *prior* distribution on θ: $p(\theta)$
2. A chosen *likelihood* of y: $p(y \mid \theta)$
3. A resulting (of interest) *posterior* of θ: $p(\theta \mid y)$

In other words: given some prior belief of θ and data from a model that depends on θ, what are our posterior beliefs of θ given the data? We explore this through *Bayes Rule*:

Proper Math

Let θ be a parameter of interest, upon which data y depends. Bayesian analysis has three main components

1. A chosen *prior* distribution on θ: $p(\theta)$
2. A chosen *likelihood* of y: $p(y \mid \theta)$
3. A resulting (of interest) *posterior* of θ: $p(\theta \mid y)$

In other words: given some prior belief of θ and data from a model that depends on θ, what are our posterior beliefs of θ given the data? We explore this through *Bayes Rule*:

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{\int_{\Theta} p(y \mid \theta)p(\theta)d\theta}$$
$$\propto p(y \mid \theta)p(\theta)$$

Understanding the Posterior

- If $p(\theta \mid y)$ is a known distribution, we can summarize with mean/variance or draw samples directly

Understanding the Posterior

- If $p(\theta \mid y)$ is a known distribution, we can summarize with mean/variance or draw samples directly
- If $p(\theta \mid y)$ is too complicated or unknown, must resort to sampling methods

Understanding the Posterior

- If $p(\theta \mid y)$ is a known distribution, we can summarize with mean/variance or draw samples directly
- If $p(\theta \mid y)$ is too complicated or unknown, must resort to sampling methods
  - Markov Chain Monte Carlo (MCMC) builds a sequence of draws

Understanding the Posterior

- If $p(\theta \mid y)$ is a known distribution, we can summarize with mean/variance or draw samples directly
- If $p(\theta \mid y)$ is too complicated or unknown, must resort to sampling methods
  - Markov Chain Monte Carlo (MCMC) builds a sequence of draws
  - Constructed so in "long run," draws are samples from $p(\theta \mid y)$

Using emcee

- emcee is a Python library that facilitates posterior inference by constructing an MCMC sampler. It computes a bunch chains in parallel.
- The user supplies a function that calculates the (proportional) log posterior pdf given parameters to sample
- The object EnsemblerSampler takes the number of chains (*nwalkers*) and number of parameters to find posteriors of (*dim*), and the above function
- The above sampler object has a method *run_mcmc()* takes a starting point and runs the chains for a number of specified samples.
- After the chains are run, the sampler object will have an attribute *chain* containing the posterior draws.

Exercises

Load the data *coin_tosses.txt*.

- Use MCMC through the package emcee to explore different priors, and how those priors impact the posterior
- Compare the results for a couple priors to the analytical posterior (we can calculate it directly here because it's a simple model)