UNIVERSITY OF MARYLAND

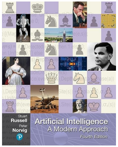
MSML610: Advanced Machine Learning

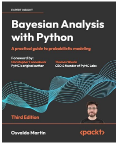
7.1: Introduction to Probabilistic Programming

Instructor: Dr. GP Saggese - gsaggese@umd.edu

References:

- AIMA (Artificial Intelligence: a Modern Approach)
  - Chap 15: Probabilistic programming
- Martin, Bayesian Analysis with Python, 2018 (2e)





SCIENCE ACADEMY

- ***Concepts***
- Coin Example

EDA vs Inference

- **Exploratory data analysis**
  - Summarize, interpret, check data
  - Visually inspect the data
  - Compute descriptive statistics
  - Communicate results
- **Inferential statistics / inference**
  - Draw insights from a limited set of data
  - Make predictions for future unobserved data points
  - Understand a phenomenon
  - Choose among competing explanations for the same observations

Good vs Bad Way to Do Statistics

- **Bad** 😈
  - Learn a collection of "statistical recipes"
    - Make assumption / approximate to make math workable
  - Given data and problem
    - Pick one recipe
    - Try until you get a "low" p-value
  - For machine learning
    - Iterate until you get a "good" fit on out-of-sample data
- **Good** 😊
  - General approach to statistical inference (Bayesian statistics)
    - Remove limitations from closed analytical form
  - Probabilistic approach unifies (seemingly) disparate methods
    - E.g., statistical methods and machine learning
    - E.g., `statsmodels` linear regression vs `sklearn` decision tree
    - Deep unity of different recipes
  - Modern tools (e.g., PyMC3) solve previously unsolvable models

Data

- Data **comes from**:
    - Experiments
    - Simulations
    - Surveys
    - Field observations
- Data is stochastic due to **uncertainty**
    - Ontological: system is intrinsically stochastic
    - Technical: measurement precision is limited or noisy
    - Epistemic: conceptual limitations in understanding
- Collecting data is **costly**
    - Consider questions before collecting data
    - Experiment design is a branch of statistics for data collection
- Data is **rarely clean and tidy**
- Data needs to be **interpreted** through mental and formal models

Models

- **Models** are simplified descriptions of a given system/process
  - A more complex model is not always a better one
  - VC dimension made it mathematical precise
    - *"You need at least 10 data points per effective degree of freedom of the hypothesis set"*
- **Goals**
  - Capture the most relevant aspects of the system
  - Ignore minor details

Bayes' Theorem: Recap

- **Bayes' theorem** posits that for model parameters θ and data X

$$\Pr(\theta|X) = \frac{\Pr(X|\theta)\cdot\Pr(\theta)}{\Pr(X)}$$

  where:
  - $\Pr(\theta|X)$
    - **Posterior**: probability for parameters θ after seeing data X
  - $\Pr(X|\theta)$
    - **Likelihood** (aka "statistical model"): plausibility of data X given parameters θ
  - $\Pr(\theta)$
    - **Prior**: knowledge about parameter θ before any data
  - $\Pr(X)$
    - **Evidence** ("marginal likelihood"): probability of observing data X
    - "Marginal" as it averages over all possible parameter values
- In other words:

$$\text{Posterior} = \frac{\text{Likelihood}\cdot\text{Prior}}{\text{Evidence}}$$

Bayesian Models

- **Probability** measures uncertainty about parameters
- **Bayes' theorem** updates probabilities with new data, reducing uncertainty (hopefully)

$$\Pr(hyp|data) = \frac{\Pr(data|hyp)\Pr(hyp)}{\Pr(data)}$$

- **Bayesian modeling workflow**

  1. Design a model using probabilities based on data and assumptions
     - Assumptions on data generation
     - Model can be a crude approximation
  2. Apply Bayes' theorem to "condition" the model on data
  3. Validate model against:
     - Data
     - Subject expertise
     - Related models

  - Steps may involve backtracking:
    - Correct coding errors
    - Improve model
    - Gather more or different data

- Concepts
- ***Coin Example***
  - Analytical Approach
  - Frequentist vs Bayesian
  - Probabilistic Programming

- Concepts
- Coin Example
  - ***Analytical Approach***
  - Frequentist vs Bayesian
  - Probabilistic Programming

Coin Example: Problem

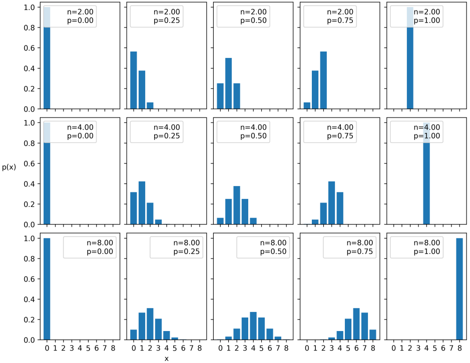
- **Problem**:
  - Toss a coin N times
  - Record the number of heads Y and tails $N - Y$
  - Question: *"How biased is the coin?"*
- There is **true uncertainty**
  - An underlying parameter exists, but it is unknown
  - θ represents the coin bias
    - 0: always tails
    - 1: always heads
    - 0.5: half tails, half heads
- **Model assumptions**:
  - Independent Identically Distributed (IID)
    - Independence: coin tosses don't affect each other
    - Identically distributed: coin's bias is constant
  - Likelihood $Y|\theta$ as a binomial distribution
    - Probability of Y heads out of N tosses, given θ
  - Prior θ as a beta distribution
    - Adopts several shapes
    - Beta is the conjugate prior of the binomial distribution

Binomial Distribution

Probability of k heads out of n tosses given bias p

$$X \sim Binomial(n, p)$$

$$\Pr(k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

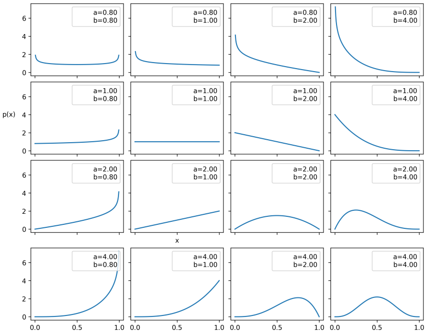


Beta Distribution

- Continuous PDF in [0, 1]
- Adopts several shapes
  - Uniform, increasing, decreasing, Gaussian-like, U-like
  - α: “success” parameter
  - β: “failure” parameter
  - $\alpha > \beta$: Skews toward 1, higher probability of success
  - $\alpha = \beta$: Symmetric, centered around 0.5
- Models probability or proportion
  - E.g., probability of success in a Bernoulli trial θ
- Beta is the conjugate prior of the binomial distribution

$$X \sim Beta(\alpha, \beta)$$

$$\Pr(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$



Conjugate Prior of a Likelihood

- **Conjugate prior** is a prior that, when combined with a likelihood, returns a posterior with the same functional form as the prior
  - E.g.,

Prior	**Likelihood**	**Posterior**
Beta	Binomial	Beta
Normal	Normal	Normal

- **Properties**
  - Prior and posterior have the same distribution
  - Posterior has a closed analytical form
    - Update parameters from the prior using data in multiple iterations
  - Ensures tractability of the posterior

Coin Example: Analytical Solution

- The **posterior** is proportional to **likelihood** × **prior**

$$\Pr(\theta|y) \propto \Pr(y|\theta)\Pr(\theta)$$

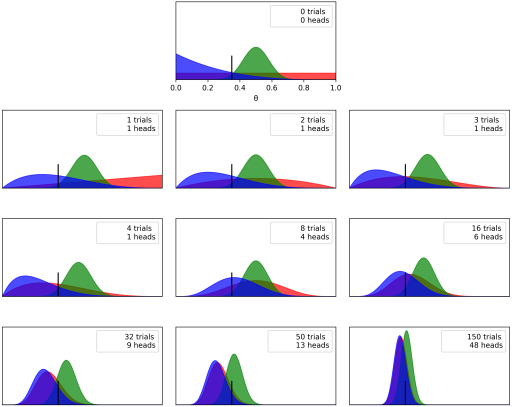
- Substituting **likelihood** with a Binomial and **prior** with a Beta

$$\begin{aligned}
&\Pr(\theta \mid Y) \\
&= \underbrace{\frac{N!}{y!(N-y)!}\theta^{y}(1-\theta)^{N-y}}_{\text{likelihood}} \underbrace{\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}}_{\text{prior}} \\
&\propto \underbrace{\theta^{y}(1-\theta)^{N-y}}_{\text{likelihood}} \underbrace{\theta^{\alpha-1}(1-\theta)^{\beta-1}}_{\text{prior}} \\
&= \theta^{y+\alpha-1}(1-\theta)^{N-y+\beta-1} \\
&= \text{Beta}\left(\alpha_{\text{prior}} + y, \beta_{\text{prior}} + N - y\right)
\end{aligned}$$

- This is how **the posterior is updated** given the data

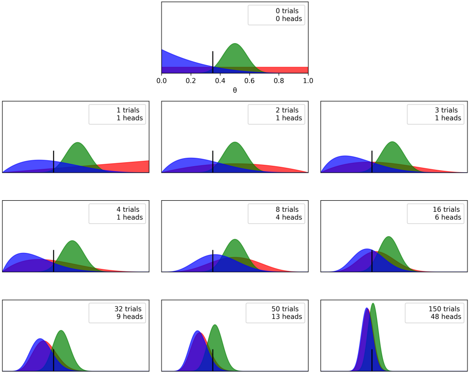
Coin Example: Effect of Priors (1/2)

- The true (unknown) value of the coin bias is 0.35
- Start with 3 different priors and update the model
  - Red: uniform prior
    - All bias values equally probable
  - Green: Gaussian-like prior around 0.5
    - Coin mostly unbiased
  - Blue: skewed towards tail
    - Coin biased
- Apply data to update the posterior distribution
- Update model



Coin Example: Effect of Priors (2/2)

- **Outcome of Bayesian analysis**
  - Posterior distribution, not a single value
- **Spread of posterior**
  - Proportional to uncertainty
  - Decreases with more data
  - Decreases faster if aligned with prior
  - With enough data, models with different priors converge to same result
- Applying posterior sequentially or at once yields same result



- Concepts
- Coin Example
  - Analytical Approach
  - ***Frequentist vs Bayesian***
  - Probabilistic Programming

Frequentist Approach vs Priors

- **Detractors of Bayesian approach** complain that:
  - *"One should let the data speak"*
  - The prior doesn't let the data speak for itself
- ⚖ **Counterpoints**
  - *"Data doesn't speak, but murmurs"*
    - Data doesn't have meaning per-se
    - Make sense of data only in context of models (e.g., mental models, mathematical models)
    - A prior is a mathematical model
  - Every statistical model has a prior, even if not explicit
    - Frequentist statistics still makes assumptions (i.e., has a prior), but are hidden
    - E.g., maximum likelihood estimate (MLE) in frequentist approach corresponds to a uniform prior and mode of the posterior
    - E.g, MLE is a point-estimate, not a distribution of plausible values

Advantages of Using Prior

- **Assumptions are clear and explicit**
  - Instead of hidden by frequentist or hacker ML approach
- **Prior**
  - Encourages deeper analysis of problem and data
  - Forces understanding before seeing data
- Posterior averaged over priors is **less prone to overfitting**
- Spread of distribution measures **uncertainty**
- Well-chosen prior simplifies and **speeds up inference**
  - "*When you encounter computational problems, there's often an issue with your model*" (Gelman, 2008)

How to Choose Priors

- **Weakly-informative priors** (aka “flat”, “vague”, “diffuse priors”)
  - Provide minimal information
    - Coefficient of linear regression centered around 0: $\beta \sim \mathit{Normal}(0, 10)$
- **Regularizing priors**
  - Known information about the parameter
    - Parameter is positive: $\sigma \sim \mathit{HalfCauchy}(0, 5)$
    - Parameter close to zero, above/below a number, or in a range
    - $\beta \sim \mathit{Laplace}(0, 1)$ (lasso prior) encourages sparsity
    - $\beta \sim \mathit{Normal}(0, 1)$ discourages extreme values
- **Informative priors**
  - Strong priors from previous knowledge (expert opinion, studies)
    - From experimental data: $\beta_1 \sim \mathit{Normal}(2.5, 0.5^2)$
    - From previous data, about 5% of cases positive: $p \sim \mathit{Beta}(2, 38)$
- **Prior elicitation**
  - Compute least informative distribution given constraints
    - Estimate distribution using maximum entropy to satisfy constraints
    - E.g., beta distribution with 90% of mass between 0.1 and 0.7

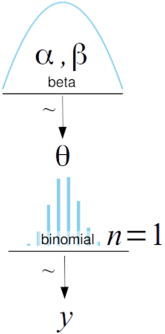
Communicating the Model of a Bayesian Analysis

1. **Communicate assumptions / hypothesis**
   - Describe priors and probabilistic models
   - E.g., coin-flip distributions:

$$\begin{cases} \theta \sim \text{Beta}(\alpha, \beta) \\ y \sim \mathit{Binomial}(n = 1, p = \theta) \end{cases}$$

2. **Communicate Bayesian analysis result**
   - Describe posterior distribution
   - Summarize location and dispersion
   - Mean (or mode, median)
   - Std dev
     - Misleading for skewed distributions
   - Highest-posterior density (HPD)
     - Shortest interval containing a portion of probability density (e.g., 95% or 50%)
     - Amount is arbitrary (e.g., `ArviZ` defaults to 94%)

Kruschke diagram



Confidence Intervals vs Credible Intervals

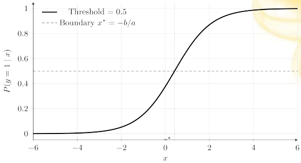
- People confuse:
  - **Frequentist confidence intervals**
  - **Bayesian credible intervals**
- In the frequentist framework, there is a true (unknown) parameter value
  - A **confidence interval** may or may not contain the true parameter value
  - Interpretation of a 95% confidence interval
    - ❌ No: *"There is a 95% probability that the true value is in this interval"*
    - ✅ Yes: *"If repeated many times, 95% of intervals would contain the true value"*
- In the Bayesian framework, parameters are random variables
  - Interpretation of a 95% **Bayesian credible interval**
    - *"There is a 95% probability that the true parameter lies within this interval, given the observed data"*
    - Bayesian **credible interval** is intuitive

Confidence Intervals vs Credible Intervals (ELI5)

- **Confidence Interval (Frequentist)**
  - Imagine fishing in a lake without seeing the fish
  - You throw your net
  - 95% confidence interval: *"If I threw this net 100 times, about 95 nets would catch the fish."*
  - Important: Once the net is thrown, it either caught the fish or not. The 95% makes sense across many attempts
- **Credible Interval (Bayesian)**
  - Imagine a magical map showing where fish *probably* are, based on past observations
  - 95% credible interval: *"Given my map, there's a 95% chance the fish is inside this part of the lake."*
  - The fish's location is uncertain, and probability describes your belief

- Concepts
- Coin Example
    - Analytical Approach
    - Frequentist vs Bayesian
    - ***Probabilistic Programming***

Bayesian Statistics



- Given:
  - The **"knows"**
    - Model structure (modeled as a graph of probability distributions)
    - Data, observations (modeled as constants)
  - The **"unknowns"**
    - Model parameters (modeled as probability distributions)
- Use Bayes' theorem to:
  - Condition unknowns to knowns
  - Reduce the uncertainty about the unknowns
- **Problem**
  - Most probabilistic models are analytically intractable
- **Solution**
  - Probabilistic programming
    - Specify a probabilistic model using code
    - Solve models using numerical techniques

Probabilistic Programming Languages

- **Steps**:
  1. Specify models using code
  2. Numerical models solve inference problems without need of user to understand how
     - Universal inference engines
     - `PyMC3`: flexible Python library for probabilistic programming
     - `Theano`: library to define, optimize, evaluate mathematical expressions using tensors
     - `ArviZ`: library to interpret probabilistic model results
- **Pros**:
  - Compute results without analytical closed form
  - Treat model solving as a black box
  - Focus on model design, evaluation, interpretation
- **Probabilistic programming languages**
  - Similar impact as Fortran on scientific computing
  - Build algorithms but ignore computational details

Coin Example: Numerical Solution (1/3)

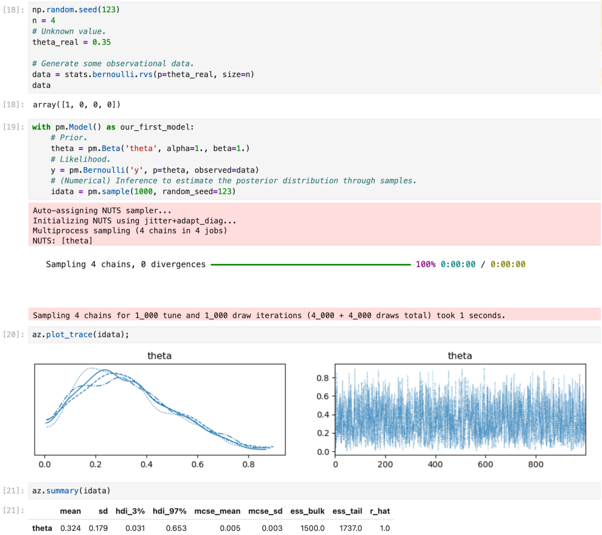
- It's a synthetic example!
  - Assume you know the true value of θ (not true in general)
- **Workflow**
  - Model the prior θ and the likelihood $Y|\theta$

$$\begin{cases} \theta \sim \text{Beta}(\alpha = 1, \beta = 1) \\ Y \sim \text{Binomial}(n = 1, p = \theta) \end{cases}$$

  - Observe samples of the variable Y
  - Run inference
  - Generate samples of the posterior
  - Summarize posterior
    - E.g., Highest-Posterior Density (HPD)
  - ...

Coin Example: Numerical Solution (2/3)

- Generate data from ground truth model
- Build PyMC model matching mathematical model
- PyMC uses NUTS sampler, computes 4 chains
- No trace diverges
- Kernel density estimation (KDE) for posterior
- Should be Beta
- Traces appear “noisy” and non-diverging (good)
- Numerical summary of posterior: mean, std dev, HDI
- $\mathbb{E}[\hat{\theta}] \approx 0.324$
- $\Pr(\hat{\theta} \in [0.031, 0.653]) = 0.94$



```
[18]: np.random.seed(123)
      n = 4
      # Unknown value.
      theta_real = 0.35

      # Generate some observational data.
      data = stats.bernoulli.rvs(p=theta_real, size=n)
      data
```

```
[18]: array([1, 0, 0, 0])
```

```
[19]: with pm.Model() as our_first_model:
          # Prior.
          theta = pm.Beta('theta', alpha=1., beta=1.)
          # Likelihood.
          y = pm.Bernoulli('y', p=theta, observed=data)
          # (Numerical) Inference to estimate the posterior distribution through samples.
          idata = pm.sample(1000, random_seed=123)
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (4 chains in 4 jobs)
NUTS: [theta]

 Sampling 4 chains, 0 divergences ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00 / 0:00:00

Sampling 4 chains for 1_000 tune and 1_000 draw iterations (4_000 + 4_000 draws total) took 1 seconds.
```

```
[20]: az.plot_trace(idata);
```

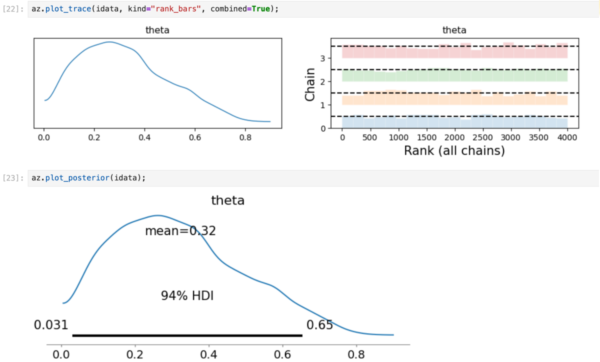
```
[21]: az.summary(idata)
```

[21]:

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
theta	0.324	0.179	0.031	0.653	0.005	0.003	1500.0	1737.0	1.0

Coin Example: Numerical Solution (3/3)

- Compute single KDE for all chains
- Rank plot to check results
- Histograms should look uniform, exploring different (and all) posterior regions
- Plot single KDE with all statistics



```
[22]: az.plot_trace(idata, kind="rank_bars", combined=True);
```

```
[23]: az.plot_posterior(idata);
```