Bayesian P-Value for Entire Distribution

- Instead of using a summary statistic, one can compute "the probability of predicting a lower or equal value for each observed value"
- If the model is well calibrated, it captures all observations equally well, the probability should be the same for all observed values
    - The output should be a uniform distribution

Bayesian P-Value: Example

- Study the height of people in a population
- **Fit the Bayesian model**
  - Assume a normal distribution with unknown mean and variance
  - Collect observed data of heights (e.g., 100 people)
  - Specify a prior distribution for mean and variance
  - Combine observed data with prior to obtain a posterior distribution of mean and variance of population height
- **Compute Bayesian p-value**
  - From posterior distribution:
    - Generate new simulated datasets
    - For each dataset, compute mean height
  - Use test statistic T, as the difference between the mean of the replicated dataset and the observed mean
  - Compute Bayesian p-value: the proportion of replicated datasets where the test statistic is $>=$ test statistic for observed data
    - A value close to 0.5 means the observed data is covered by the model
    - A value close to 0 or 1 indicates a poor fit

Bayesian vs Frequentist P-Value

- **Frequentist p-value** is the probability of getting observed data as or more extreme, assuming the null hypothesis is true
- **Bayesian p-value** is the probability that simulated data from the model (i.e., posterior predictive check) is as or more extreme than the observed data
- P-value measures inconsistency between observed data and:
  - A null hypothesis (frequentist approach)
  - Model (Bayesian approach)
- Does p-value incorporate uncertainty?
  - (Frequentist) No, it uses single point estimates
  - (Bayesian) Yes, it incorporates uncertainty of parameter estimates

- ***The Balance Between Simplicity and Accuracy***
- Measures of Predictive Accuracy
- Regularizing priors
- Regularizing Priors
- Mixture Models

Occam's Razor

- *"If you have **equivalent** explanations for the same phenomenon, you should choose the **simpler** one"*
  - Quality of explanation $\approx$ accuracy
  - Simpler $\approx$ number of model parameters
- **Complexity vs accuracy**
  - Increasing model complexity (e.g., number of model parameters) is accompanied by:
    - Increasing in-sample accuracy
    - Not necessarily out-of-sample accuracy
  - The complex model:
    - Did not "learn" from the data but just "memorize" it
    - Does a bad job generalizing to predict potentially observable data
- Ideally balance complexity and accuracy in a quantitative way

Overfitting and Underfitting

- A model is **overfit** when it has many parameters, fitting the training data well but unseen data poorly
  - Overfitting in terms of signal/noise:
    - Each dataset has "signal" and "noise"
    - We want the model to learn the signal
    - A model overfits when it learns the noise, obscuring the signal
- A model is **underfit** when it has few parameters, fitting the dataset poorly
  - An underfit model doesn't learn the signal well
  - E.g., a constant fits a dataset, only learning the mean

Bias-Variance Trade-Off

- A model has **high bias** when:
  - It has low ability to accommodate the data
  - I.e., underfitting
  - E.g., a polynomial of degree 0
- A model has **high variance** when:
  - It has high capacity and it is sensitive to details in the data, capturing noise
  - I.e., overfitting
  - E.g., a polynomial of degree 100
- Trade-off between bias and variance
  - Goal: balance simplicity and goodness of fit
  - Aim for a model that "fits the data right," avoiding overfitting or underfitting

- The Balance Between Simplicity and Accuracy
- ***Measures of Predictive Accuracy***
- Regularizing priors
- Regularizing Priors
- Mixture Models

Accuracy Measures

- **In-sample accuracy** is measured on the data used to fit a model
- **Out-of-sample accuracy** is measured on data not used to fit a model
  - Aka "predictive accuracy"
- In-sample accuracy $>$ out-of-sample accuracy
- There is a trade-off between how much data is used for training and for evaluating true accuracy

Information Criteria: Intuition

- **Information criteria** compare models in terms of fitting the data taking into account their complexity through a penalization term
  - Out-of-sample accuracy $\approx$ in-sample accuracy $+$ a term penalizing model complexity
  - It's the VC equation

$$E_{out}[h] = E_{in}[h] + \Omega(\mathcal{H})$$

Model Parameters for Bayesian vs Non-Bayesian Set-Up

Maximum Likelihood Estimation (MLE)

- **MLE** finds the parameter values that make the observed data most probable (given a model)
  - Denoted by $\hat{\theta}_{MLE}$
  - It's a point not a distribution
- **Procedure**:
  - Given the data $x_1, x_2, ..., x_n$
  - Assume it comes from a distribution with an unknown parameter θ
  - Pick the value of θ that makes the data most likely given a likelihood function

$$\begin{cases} L(\theta) = \log \Pr(x_1, x_2, ..., x_n|\theta) \\ \hat{\theta}_{MLE} = \text{argmax}_\theta L(\theta) \end{cases}$$

- In Bayesian terms, MLE is equivalent to the mode of θ using flat priors
  - Aka MAP (maximum a posteriori)

Akaike Information Criterion (AIC)

- AIC is defined as

$$AIC = -2 \sum \log \Pr(y_i | \hat{\theta}_{MLE}) + 2\text{num}_{params}$$

  where:

  - $\hat{\theta}_{MLE}$ is the maximum likelihood estimation of θ
  - num_{params} is the number of parameters

- **Interpretation**:

  - The first term (log likelihood) measures how well the model fits the data
  - The second term penalizes complex models

- **Cons**:

  - Discard information about uncertainty of posterior estimation
  - MLE assumes flat priors (vs informative and weakly informative priors)
  - Number of parameters is not always a good measure of complexity
    - E.g., in hierarchical models the effective number of params is smaller

Bayesian Information Criteria

- **Bayesian Information Criteria (BIC)**
  - Like AIC, it assumes flat priors and uses MLE
  - It is not Bayesian
- **Widely Applicable Information Criteria (WAIC)**
  - Bayesian version of AIC
  - It has two terms:
    - One that measures how good the fit is
    - One that penalizes complex models
  - WAIC uses the posterior distribution to estimate both terms

Cross-Validation

- **Cross-validation** (CV)
  - **Procedure**
    - Partition data into K portions of equal size and similar statistics
    - Use $K-1$ partitions to train the model and test on remaining partition
    - Repeat for all K folds
    - Average the results
  - **Pros**
    - Simple and effective solution to use all data to compare models
- **Leave-one-out cross-validation** (LOO-CV)
  - **Procedure**:
    - The model is fit for all data, excluding one observation
    - The model's predictive accuracy is tested on the left out observation
    - Repeat the process for all observations
    - Average the results
  - **Cons**
    - It is very computationally expensive since one needs to refit the model
- How to adapt **cross-validation to a Bayesian approach**?
  - CV and LOO require multiple model fits and fitting a Bayesian model is very expensive
  - Yes! There is a way to approximate using a single fit to the data

ELPD with LOO-CV

- 💀 Math alert
- We want to compute $ELPD_{LOO-CV}$ where:
  - "Expected Log-Pointwise predictive Density" (ELPD)
    - It should be ELPPD and not ELPD!
  - "Leave-One-Out Cross-Validation" (LOO-CV) is used to compute it
- The definition of ELPD with LOO-CV is:

$$ELPD_{LOO-CV} = \sum_{i=1}^{n} \log \int p(y_i|\theta) p(\theta|y_{-i}) d\theta$$

  where:
  - Fit model using all the data without y_{-i}
  - Predict with the model the unseen y_i
  - Integrate on all the posterior values
  - Repeat for all the points
- How to compute it efficiently?
  - Use "Pareto smooth importance sampling leave-one-out cross-validation"

Pointwise Predictive Density (PPD)

- The **pointwise predictive density** for a given data point y_i is defined as the posterior predictive probability, given the rest of the data

$$PPD \triangleq \Pr(y_i|data - \{i\}) = \int p(y_i|\theta)p(\theta|y_{-i})d\theta$$

  - y_i: observed data point
  - $p(y_i|\theta)$: likelihood given model parameters θ
  - $p(\theta|y_{-i})$: posterior distribution of the model parameters given rest of data
  - Integral: averages over posterior distribution, capturing parameter uncertainty
- **Interpretation**
  - PPD measures model's predictive ability for y_i when trained on data excluding y_i
  - Similar to cross-validation, using Bayesian parameter averaging over the model parameters

Expected Log Pointwise Predictive Density

- The ELPD is the average over unseen points of the log PPD

$$ELPD \triangleq \sum_{i=1}^{n} \log \int p(y_i|\theta_{-i}) p(\theta_{-i}|y_{-i}) d\theta$$

- **Interpretation**
  - It can be used to determine which model generalizes better to new data
  - ELPD measures the predictive accuracy of a Bayesian model on unseen data
  - Train on y_{-i}, i.e., all data excluding y_i
  - For each point y_i excluded from the training set, there is a new distribution of the params θ_{-i}
  - Test on y_i

PSIS-LOO-CV

- Compute the Expected Log Pointwise Predictive Density (ELPD) using Leave-One-Out Cross-Validation (LOO-CV):

$$ELPD_{LOO-CV} \triangleq \sum_i \log \int p(y_i|\theta)p(\theta|y_{-i})d\theta$$

- **Problem**: Need to train N models, one for every data point
- **Solution**:
  - Pareto-Smoothed Importance Sampling (PSIS) Leave-One-Out Cross-Validation (LOO-CV) estimates without refitting for every point
  - **Importance sampling**:
    - Use full dataset to approximate posterior distribution when one observation is left out
    - Re-weight posterior samples based on importance
  - **Pareto-smoothing**:
    - Stabilize importance weights, reducing extreme weights' impact
    - E.g., if an observation left out influences the posterior distribution
    - Provide diagnostics to assess reliability of importance weights

Evidence of Data Given a Model

- The Bayesian way to compare k models is to calculate the evidence of each model $\Pr(Y|M_k)$, i.e., the probability of observed data Y given each model M_k
  - Typically we ignore the evidence when we do parameter inference
- Consider the Bayes theorem for the parameters θ and the data Y, given a model M_k

$$\Pr(\theta|Y, M_k) = \frac{\Pr(Y|\theta, M_k)\Pr(\theta|M_k)}{\Pr(Y|M_k)}$$

- We find the parameters θ that maximizes the ratio, independently of the probability of the evidence

$$\text{argmax}_\theta \Pr(\theta|y, M_k) = \text{argmax}_\theta \Pr(y|\theta, M_k)\Pr(\theta|M_k)$$

- Even if we need to choose the best model among $M_1, ..., M_k$ we can pick the one that maximizes

$$\text{argmax}_k \Pr(M_k|y) \propto \Pr(y|M_k)\Pr(M_k)$$

Bayes Factors

- The Bayes factors are defined as the ratio of the two marginal likelihoods under competing hypotheses

$$BF = \frac{\Pr(y|M_0)}{\Pr(y|M_1)}$$

where $BF > 1$ means that the model 0 explains the data better than model 1

Bayes factor	Support
1-3	Anecdotal
3-10	Moderate
10-30	Strong
30-100	Very strong
>100	Extreme

- Intuition
  - Bayes factors are a quantitative tool that helps compare how likely two competing explanations (i.e., models) are, given the evidence you find
  - Bayes factors are like a scale that weigh how much evidence supports one theory over another

Assumption of Bayes Factors

- The assumption of Bayes factor is that the models have the same prior probability
- Otherwise we need to compute the "posterior odds" as "Bayes factors" x "prior odds"

$$\frac{\Pr(M_0|y)}{\Pr(M_1|y)} = \frac{\Pr(y|M_0)}{\Pr(y|M_1)} \frac{\Pr(M_0)}{\Pr(M_1)} = \text{Bayes factors} \times \text{prior odds}$$

Bayes Factors: Pros and Cons

- Looking at the definition of marginal likelihood (aka evidence):

$$p(y) = \int_{\theta} p(y|\theta)p(\theta)d\theta$$

- Making the dependency of the model M_k explicit

$$p(y|M_k) = \int_{\theta_k} p(y|\theta_k, M_k)p(\theta_k, M_k)d\theta_k$$

- Pros
  - Models with more parameters have a larger prior, so the Bayes factor has a built-in Occam's Razor
- Cons
  - The marginal likelihood needs to be computed numerically over a large dimensional space
  - The marginal likelihood depends on the value of the prior
    - Changing the prior might not affect the inference of θ but have a direct effect on the marginal likelihood

Hierarchical Models: Candies in a Jar Examples

- Each classroom has a jar filled with candies, each different but coming from the same candy shop
- Kids in each classroom need to guess the number of candies in each jar
- Individual guesses
  - Think of each jar as its own little puzzle
  - E.g., guess based on how big the jar is, how filled it is
  - Each jar has certain "parameters"
- Group learning
  - Consider what you learn from other jars since they come from the same candy shop
  - E.g., the shop prefers to use a certain type of candies, or fills the jar up to a certain level
  - The jars have certain "hyper-parameters"
- Sharing info
  - As you make more guesses, you start sharing what you have learned with your friends about each jar
  - The hierarchical model lets the info flow across models for individual jar

Computing Bayes Factors as Hierarchical Models

- The computation of Bayes factors can be framed as a hierarchical model
  - The high-level parameter is an index assigned to each model and sampled from a categorical distribution
- We perform inference of the competing models at the same time, using a discrete variable jumping between models
  - The proportion we use to sample each model is proportional to $\Pr(M_k|y)$
- Then we compute the Bayes factors
- The models can be different in the prior, in the likelihood, or both

Common Problems When Computing Bayes Factors

1. If one model is better than the other, then we will spend more time sampling from it
   - Cons: under-sample one of the models
2. Values of the parameters are updated, even when the parameters are not used to fit that model
   - E.g., when model 0 is chosen, the parameters in model 1 are updated, but they are only restricted by the prior
   - If the prior is too vague, the parameter values might be too far from previous accepted values and the step is rejected
   - TODO: ?

- Solutions to improve sampling
  - Force both models to be visited equally
  - Use "pseudo priors"

Using Sequential Monte Carlo to Compute Bayes Factors

- TODO

Bayes Factors and Information Criteria

-
- If we take the log of Bayes factors, we turn ratio of marginal likelihood into a difference, which is similar to comparing differences in information criteria
- We can interpret each marginal likelihood as having:
  - a fitting term (i.e., how well the model fits the data)
  - penalizing term (i.e., averaging over the prior)
    - more parameters $\rightarrow$ more diffused the prior $\rightarrow$ greater penalty
-
- TODO

- The Balance Between Simplicity and Accuracy
- Measures of Predictive Accuracy
- ***Regularizing priors***
- Regularizing Priors
- Mixture Models

- The Balance Between Simplicity and Accuracy
- Measures of Predictive Accuracy
- Regularizing priors
- ***Regularizing Priors***
- Mixture Models

Priors and Regularization

- Using weakly/informative priors is a way of pushing a model to prevent overfitting and generalize well
- This is similar to the idea of "regularization"
- Regularization
  - Reduce information that a model can represent and reduce chances to capture noise instead of signal
  - E.g., penalize large values for the parameters in a model
  - E.g., ridge and Lasso regression applies regularization to least square method

Popular Regularization Methods in Bayesian Framework

- Ridge regression
  - Normal distribution for coefficients of linear model, pushing them toward zero
- Lasso regression
  - MAP of posterior using Laplace priors for coefficients
  - Because Laplace distribution looks like Gaussian with a sharp peak at zero, it provides "variable selection" since it induces sparsity of model

- The Balance Between Simplicity and Accuracy
- Measures of Predictive Accuracy
- Regularizing priors
- Regularizing Priors
- ***Mixture Models***

Mixture Models

- Mixture models are models that assume that data comes from a mixture of distributions or where the solution can be approximated as a mix of simpler distributions
- Mixtures can model data from sub-populations
  - E.g., distribution of heights in adult human population, made of male and female
  - E.g., clustering of handwritten digits (e.g., 10 sub-populations)
- Mixture models as statistical trick to handle complex distributions
  - E.g., mix of Gaussians to describe an arbitrary distribution
    - E.g., multimodal, skewed distributions
    - The number of distributions depends on the accuracy of the approximation and the details of the data
  - E.g., Kernel density estimation (KDE) is a (non-Bayesian) non-parametric estimation technique
    - Place a Gaussian with a fixed variance on top of each data points
    - Use the sums of all the individual Gaussians to approximate the empirical distribution of the data

Finite Mixture Models

- $=$ the PDF of the observed data is a weighted sum of the PDFs of K subgroups

$$p(y) = \sum_{i=1}^{K} w_i p(y|theta_i)$$

where:

- K is finite
- $w_i \in [0, 1]$ (it's like the probability of component i)
- $\sum_i w_i = 1$
- $p(y|\theta_i)$ are simpler distributions (e.g., Gaussian or Poisson)

Conceptual Solution of a Mixture Model

- The assumption of a mixture model is that the data is generated by K underlying distributions / components
  - In other words, each data point is assumed to come from one of the components, but we don't know which
- The goal is to determine which component of the model k the data point x most likely belongs to
  - For each point x, assign probabilities of belonging to one of the K components $\Pr(k|x)$
  - This is modeled as a random variable, which is called "latent variable" since it can't be observed
  - E.g., for two components ($K = 2$) we use a Bernoulli, using a Beta distribution as prior (since it is conjugate prior for Bernoulli)
  - E.g., for K outcomes we can use as prior
    - A Categorial distribution to generalize the Bernoulli
    - A Dirichlet distribution to generalize the Beta distribution
- One can estimate the likelihood of the observed data based on:
  - Parameters of the individual components
  - Probabilities that a given data point belongs to each component

Categorical Distribution

- Discrete distribution representing "rolling a K-sided unfair die" or "choosing a random card from a deck of cards"
  - It generalizes a Bernoulli distribution $K = 2$
- It is parametrized with probabilities for each possible outcome $(p_1, p_2, ..., p_K)$, where p_i is the probability of outcome i

Dirichlet Distribution

- Defined in a simplex which is a generalization of a triangle in K-dimensions, where K vectors elements are in $[0, 1]$ and sum to 1
  - A 1d simplex is an interval
  - A 2d simplex is a triangle
  - A 3d simplex is a tetrahedron
  - ...
- Dirichlet distribution generalizes the Beta distribution
  - Beta models the probability of a single proportion (e.g., the probability of success in a Bernoulli trial)
  - Beta models 2 outcomes, one with probability p and the other with $1 - p$
  - It uses two parameters to describe its shape
- The Dirichlet distribution models the probability of K outcomes
  - It represents the uncertainty over the proportions of different outcomes in a multinomial experiment

Dirichlet Distribution: Examples

- Bucket of colored balls
  - You have a bucket of colored balls, each ball comes in 3 colors
  - You want to figure out the probability of picking a ball of each color
  - You are uncertain about the probability and you model that uncertainty
- You want to divide a pie into 4 different slices
  - You want to model the size of the slices (making sure they split the entire pie), modeling the uncertainty

Re-Parametrization Using Marginalization

- Consider a mixture model that include latent variables representing the component from which each observation is drawn
- Performing posterior sampling on these models is inefficient for several reasons:
  - Discrete variables introduce discontinuities
  - Latent variable introduces dependencies
- A solution is "marginalization"
  - Removing the latent variable by integrating out from the model
  - The observations are directly modeled from a mixture distribution
  - This makes sampling more efficient
  - The problem is that the model becomes more complex mathematically
  - `pymc` has distributions (e.g., `NormalMixture`) where the latent variable is already marginalized

Non-Identifiability of Parameters

- Parameters in a model are "not identifiable" when one or more parameters cannot be uniquely determined
- In practice multiple model fits give different parameters corresponding to the same model
  - E.g., given a bimodal distribution sum of two Gaussians, solving the same model can switch the value of the means (aka "label switching problem")
  - E.g., variables with high-correlation in linear models
- Solutions
  - Arrange the means of the components to be in increasing order
    - E.g., in `pymc` add a "potential" to the likelihood (which doesn't depend on the data) so that a constraint is not violated
  - Use informative priors to guide a model towards a canonical representation

How to Choose K

- How to decide the number of components K in a finite mixture model?
- Solution
  - Start with a small number of components K
  - Increase K to improve the model-fit evaluation
  - Use model selection techniques to find best trade-off
    - E.g., using posterior-predictive checks, WAIC, LOO, or domain expertise