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

SCIENCE ACADEMY

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

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

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
  - Test on y_i

SCIENCE ACADEMY

Approximating PPD

- Calculating analytically the pointwise posterior density integral

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

  is difficult

  - The posterior $p(\theta|y_{-i})$ rarely has a closed form
  - The integral on θ is on a high-dimensional space

- It can be approximated numerically given posterior samples s of the model parameters $\theta^{(s)}$

$$PPD \approx \frac{1}{S}\sum_{s} p(y_i|\theta^{(s)}_{-i})$$

  - Suppose we already have posterior samples $\theta^{(s)} \sim p(\theta|y)$ from the full dataset

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**: Train once per point
- **Solution**:
  - Pareto-Smoothed Importance Sampling (PSIS) Leave-One-Out Cross-Validation (LOO-CV) estimates the formula without refitting the model for every point
  - **Importance sampling**:
    - Use the full dataset to approximate the posterior distribution when a single observation is left out
    - Re-weight posterior samples based on importance
  - **Pareto-smoothing**:
    - Stabilize importance weights, reducing the impact of extreme weights
    - E.g., if an observation left out has a large influence on the posterior distribution
    - Provide diagnostics to assess the reliability of importance weights

Predictive Accuracy with Arviz

- If the inference data has the log-likelihood group

  `pm.sample(idata_kwargs="log_likelihood": True)`

  metrics such as WAIC and LOO (with / without ELPD) can be automatically computed

- In the first section
  - The first row is ELPD
  - The second row is the effective number of parameters

- In the second section, there is the Pareto k diagnostic
  - Since all the values are between 0 and 0.7, the approximation can be trusted

Comparing Predictive Accuracy with Arviz

- In general the predictive accuracy metrics should be interpreted in relation to other models

Model Averaging

- You have multiple models explaining the data: what do you do?
  1. Select a single model
     - Simple solution used in frequentist approach
     - "Model selection"
  2. Report all the models with their informations (e.g., standard errors, posterior predictive checks)
     - Express advantages and shortcomings of the models
  3. Average all the models
     - Build a meta-model using a weighted average of each model
     - Weight prediction by the difference between information criteria (e.g., WAIC, LOO) of the models
     - A hierarchical model is a continuous versions of multiple discrete models

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 $M_1, \ldots, M_k$ 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"

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

SCIENCE ACADEMY

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

SCIENCE ACADEMY

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.g., averaging over the prior)
    - more parameters $\rightarrow$ more diffused the prior $\rightarrow$ greater penalty
-
- TODO

UNIVERSITY OF MARYLAND 18 56 SCIENCE ACADEMY

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

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

SCIENCE ACADEMY

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