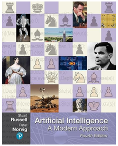
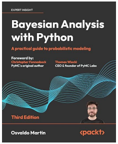
UNIVERSITY OF MARYLAND

MSML610: Advanced Machine Learning

7.5: Bayesian Model Comparison





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)

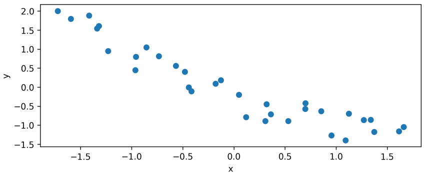
- ***Bayesian Model Comparison***
  - The Balance Between Simplicity and Accuracy
  - Measures of Predictive Accuracy
  - Bayesian Model Selection and Ensemble
  - Bayesian Hypothesis Testing
  - Regularizing Priors

Models as Maps of the Real World

- Typically you need to compare models to understand which one is **"better"**
- Models are a **map, not a copy** of the real world
  - *"All models are wrong, but some are useful"* (Box, 1976)
    - "Wrong": all models are wrong since they aren't the actual territory
    - "Useful" some models describe a problem better than others
- Models have a **purpose**
  - Are approximations to understand a problem
  - A model can't reproduce all aspects equally well
  - Different models capture different data aspects

Posterior Predictive Checks

- **Goal of PPC**:
  - Evaluate model's data explanation
  - Understand model limitations
  - Improve model
- Given data from parabola + noise:
  - Fit with linear model
  - Fit with quadratic model
  - Compare predicted posterior vs observed data

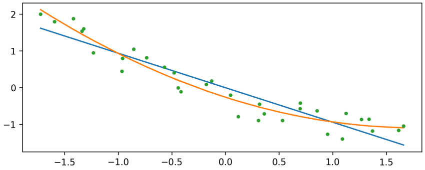


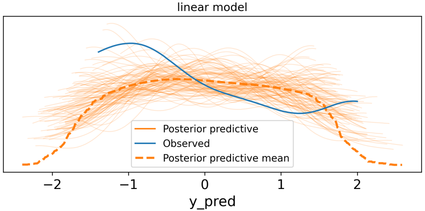

```python
# Linear model.
with pm.Model() as model_l:
    # mu = alpha + beta * x
    alpha = pm.Normal('alpha', mu=0, sigma=1)
    beta = pm.Normal('beta', mu=0, sigma=10)
    mu = alpha + beta * x_c[0]
    #
    sigma = pm.HalfNormal('sigma', 5)
    #
    y_pred = pm.Normal('y_pred', mu=mu, sigma=sigma, observed=y_c)
    #
    idata_l = pm.sample(2000, idata_kwargs={"log_likelihood": True})
    idata_l.extend(pm.sample_posterior_predictive(idata_l))


# Quadratic model.
with pm.Model() as model_p:
    # mu = alpha + beta_1 * x + beta_2 * x^2
    alpha = pm.Normal('alpha', mu=0, sigma=1)
    # Beta is a 2-dim vector.
    beta = pm.Normal('beta', mu=0, sigma=10, shape=order)
    mu = alpha + pm.math.dot(beta, x_c)
    #
    sigma = pm.HalfNormal('sigma', 5)
    #
    y_pred = pm.Normal('y_pred', mu=mu, sigma=sigma, observed=y_c)
    #
    idata_q = pm.sample(2000, idata_kwargs={"log_likelihood": True})
    idata_q.extend(pm.sample_posterior_predictive(idata_q))
```

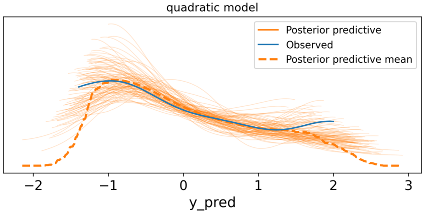
SCIENCE ACADEMY

Posterior Predictive Checks

- **Compare KDE of observed and predicted data**
  - Linear model KDE doesn't match
  - Quadratic model KDE matches better
  - High uncertainty in both models, especially at tails
- Compare mean / interquartile range for data vs model
  - Plot dispersion of mean and IQR for models vs data
  - Data set provides a single point
  - Posterior provides a distribution
- Statistics "orthogonal" to model's focus are more informative, since they are less "overfit"
  - E.g., fit the mean, compare the "median"







SCIENCE ACADEMY

Bayesian P-Value for a Statistic

- A Bayesian p-value summarizes the comparison between simulated and observed data
- **Procedure**
  - Given the posterior predictive $\tilde{Y}$
  - Choose a summary statistic T (E.g., mean, median, standard deviation)
  - Compute T for:
    - The observed data T_{obs}
    - The simulated data T_{sim} from the posterior predictive
  - Compute the Bayesian p-value as the portion of simulated datasets where the test statistic is smaller than the observed data:

$$\text{Bayesian p-value} \triangleq \Pr(T_{sim} \geq T_{obs} | \tilde{Y})$$

  - If observed values agree with predicted ones, the value should be around 0.5

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

- Bayesian Model Comparison
  - ***The Balance Between Simplicity and Accuracy***
  - Measures of Predictive Accuracy
  - Bayesian Model Selection and Ensemble
  - Bayesian Hypothesis Testing
  - 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

- Bayesian Model Comparison
  - The Balance Between Simplicity and Accuracy
  - ***Measures of Predictive Accuracy***
  - Bayesian Model Selection and Ensemble
  - Bayesian Hypothesis Testing
  - 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
  - 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

- Bayesian Model Comparison
    - The Balance Between Simplicity and Accuracy
    - Measures of Predictive Accuracy
    - ***Bayesian Model Selection and Ensemble***
    - Bayesian Hypothesis Testing
    - Regularizing Priors

Bayesian Model Selection

- **Bayesian way to compare k models**
  - Calculate the evidence of each model $\Pr(Y|M_k)$, i.e., the probability of observed data Y given each model M_k
- Fitting a model and model selections are the same process in Bayesian approach
  - The VC framework considers fitting models and selecting models in the same way
  - In the frequentist approach there are different procedures
- **Model fitting**
  - Consider Bayes theorem for parameters θ and data Y, given model M_k

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

  - Find parameters θ that maximize the ratio, independently of evidence probability

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

- **Model selection**
  - To choose the best model among $M_1, ..., M_k$, pick the one that maximizes

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

SCIENCE ACADEMY

Model Averaging

- What do you do when you have multiple models explaining the data?
  1. **Model selection**
     - Select one model
     - Simple solution
  2. **Report all 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

- Bayesian Model Comparison
  - The Balance Between Simplicity and Accuracy
  - Measures of Predictive Accuracy
  - Bayesian Model Selection and Ensemble
  - ***Bayesian Hypothesis Testing***
  - Regularizing Priors

Bayes Factors

- **Bayes factors** are ratio of two marginal likelihoods of the data under competing model hypotheses M_0 and M_1

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

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

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

- **Intuition**
  - Act as a scale weighing evidence for 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

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

- Bayesian Model Comparison
  - The Balance Between Simplicity and Accuracy
  - Measures of Predictive Accuracy
  - Bayesian Model Selection and Ensemble
  - Bayesian Hypothesis Testing
  - ***Regularizing Priors***

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