Cognitive Modeling Homework IV

Itamar Oren-Naftalovich, Annabelle Choi

April 2025

Problem 1: True-False Questions

Which Are False?

1. **(1) is true.** K-fold CV indeed requires K separate fits and can be expensive.
2. **(2) is true.** BFs are inherently relative; they compare two models' marginal likelihoods, not absolute fit.
3. **(3) is false.** Bayes factors *can* compare models with different likelihood forms.
4. **(4) is false.** The Binomial is a special case of Multinomial, but the Dirichlet is a *prior* over the simplex, not a *special case* of Multinomial.
5. **(5) is true.** LOO-CV uses the posterior predictive distribution for left-out points.
6. **(6) is false.** AIC penalizes complexity via a simple parameter-count term, not the variance of the marginal likelihood.
7. **(7) is false.** The LPD is more about predictive fit rather than directly measuring complexity.
8. **(8) is true.** I typically take $\frac{1}{S}\sum_{s=1}^{S} p(y \mid \theta^{(s)})$ across MCMC draws, then take the log of that average for the LPD.
9. **(9) is true.** By definition, Bayes factors do not incorporate prior model odds.
10. **(10) is false.** It is not always preferable to use information criteria; cross-validation can be more robust if it is computationally feasible.

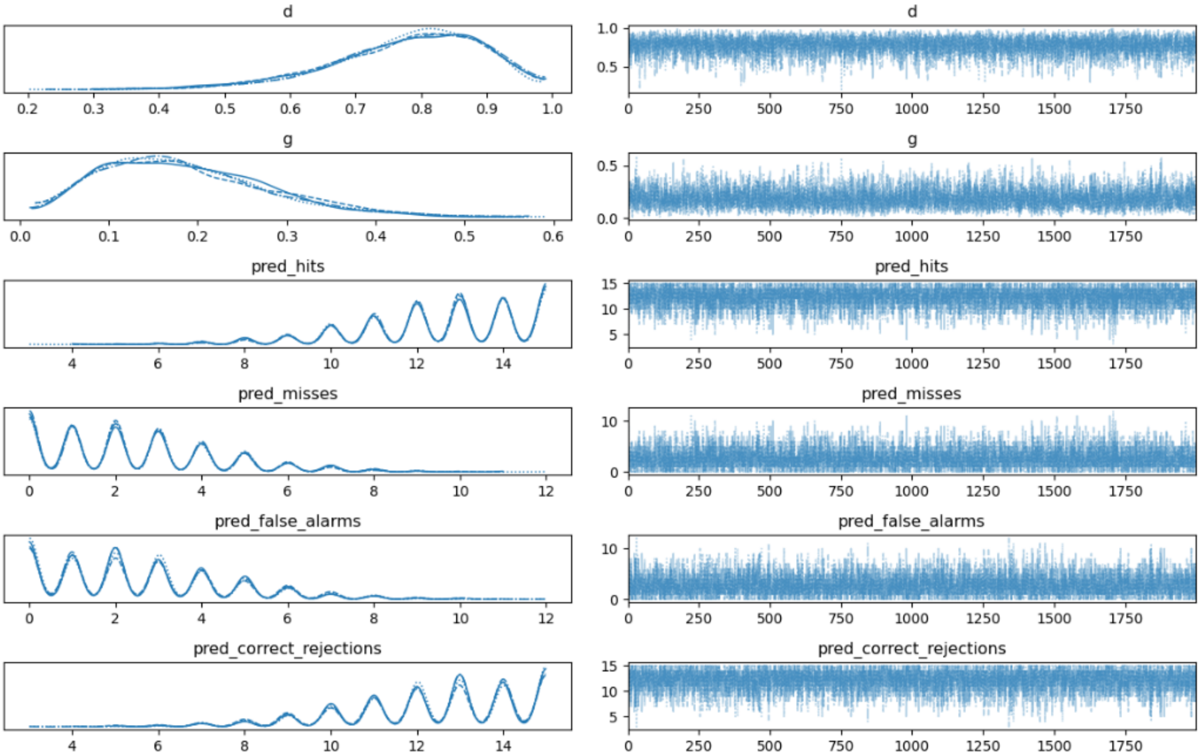
Problem 2 (Optional): Simple MPTs

For this problem, I presented fifteen old words and fifteen new words to a participant. They had to decide whether each word was old or new. Based on their responses, I recorded how many times they were correct or incorrect:

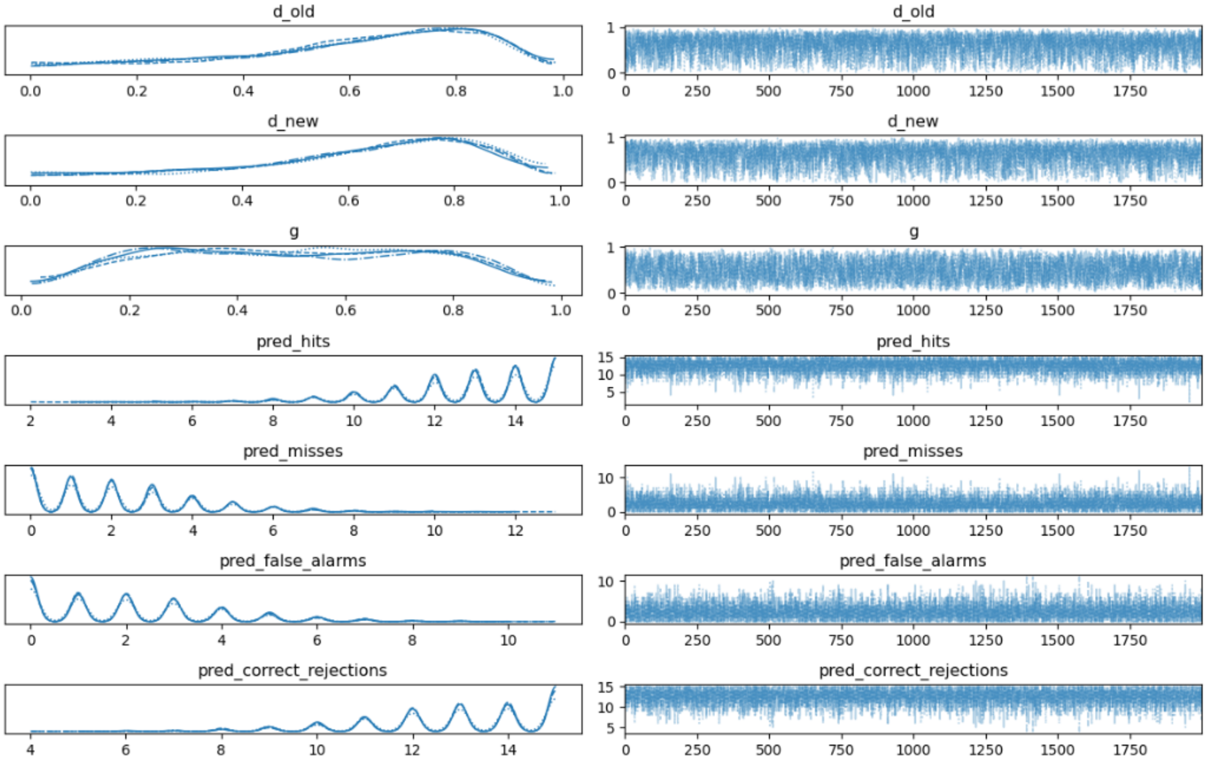
- **Hit = said ”yes” to an old word (correctly remembered seeing it)**
- **Miss = said ”no” to an old word (forgot they had seen it before)**
- **False alarm = said ”yes” to a new word (thought they saw it, but didn't)**
- **Correct rejection = said ”no” to a new word (correctly identified it as new)**

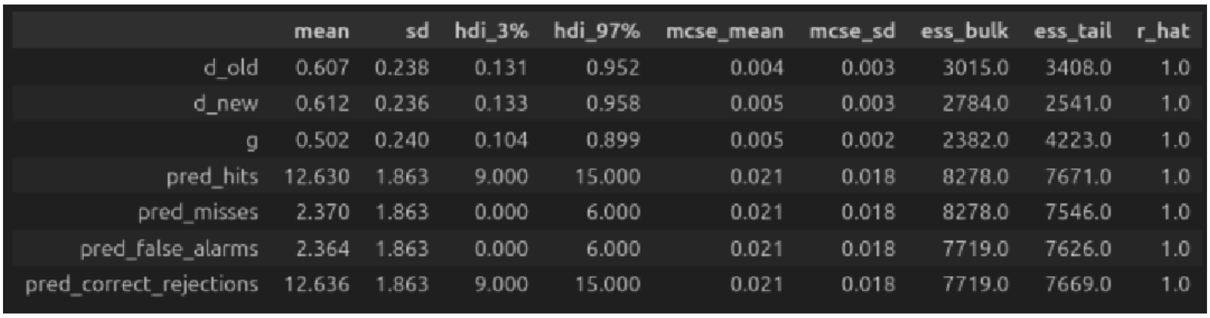
Using this information, I fit two models: the One-High-Threshold (1HT) model and the Two-High-Threshold (2HT) model. The 1HT model assumes participants can only recognize old words and guess when unsure, while the 2HT model assumes participants can recognize both old and new words.

1HT Model Trace Plots



2HT Model Trace Plots





	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
d_old	0.607	0.238	0.131	0.952	0.004	0.003	3015.0	3408.0	1.0
d_new	0.612	0.236	0.133	0.958	0.005	0.003	2784.0	2541.0	1.0
g	0.502	0.240	0.104	0.899	0.005	0.002	2382.0	4223.0	1.0
pred_hits	12.630	1.863	9.000	15.000	0.021	0.018	8278.0	7671.0	1.0
pred_misses	2.370	1.863	0.000	6.000	0.021	0.018	8278.0	7546.0	1.0
pred_false_alarms	2.364	1.863	0.000	6.000	0.021	0.018	7719.0	7626.0	1.0
pred_correct_rejections	12.636	1.863	9.000	15.000	0.021	0.018	7719.0	7669.0	1.0

The 1HT model showed that the participant could recognize old words well and sometimes guessed when unsure. However, it did not measure how well they could reject new words.
The 2HT model gave a more complete view. It showed that the participant was good at recognizing both old and new words, with similar detection rates for each. The guessing rate was also moderate. The model's predictions were close to the actual answers, showing that it fit the data well.
Overall, while both models were accurate, the 2HT model provided more detailed information about the participant's memory and guessing, making it the better model for this task.

Problem 3: Multiple Regression

Goal

I extended our prior Bayesian linear regression to **multiple** regression using the Insurance Costs data set. The target variable is `charges` (medical insurance costs), and this includes:

- **bmi**
- **age**
- **children**
- `smoker` (0 = no, 1 = yes)

Model Specification

I implemented a Normal likelihood with priors:

$$\begin{aligned}
\sigma &\sim \text{Inv-Gamma}(\tau_0, \tau_1),\\
\alpha &\sim \mathcal{N}(0, \sigma_\alpha^2),\\
\beta_j &\sim \mathcal{N}(0, \sigma_\beta^2) \quad (j = 1, \dots, M),\\
y_n &\sim \text{Normal}\left(\alpha + \boldsymbol{\beta}^\top \mathbf{x}_n, \sigma\right).
\end{aligned}$$

Data Preprocessing/Code

See the `HW4.ipynb` file.

Posterior Summaries

From the `az.summary` for the main parameters:

Parameter	Mean	SD	3% HPD	97% HPD	R-hat
alpha	-0.396	0.017	-0.430	-0.364	1.000
β_0 **(bmi)**	0.165	0.016	0.136	0.195	1.000
β_1 **(age)**	0.298	0.016	0.269	0.328	1.000
β_2 **(children)**	0.042	0.015	0.014	0.072	1.000
β_3 **(smoker)**	1.951	0.039	1.875	2.020	1.000
sigma	0.507	0.011	0.486	0.527	1.000

We can see from the above that $\hat{R} \approx 1$.

Posterior Distributions

Interpretation

- In standardized space.
- `smoker` has a large positive coefficient (∼1.95). Switching from 0 to 1 on `smoker` is associated with ∼1.95 SD difference in `charges`.

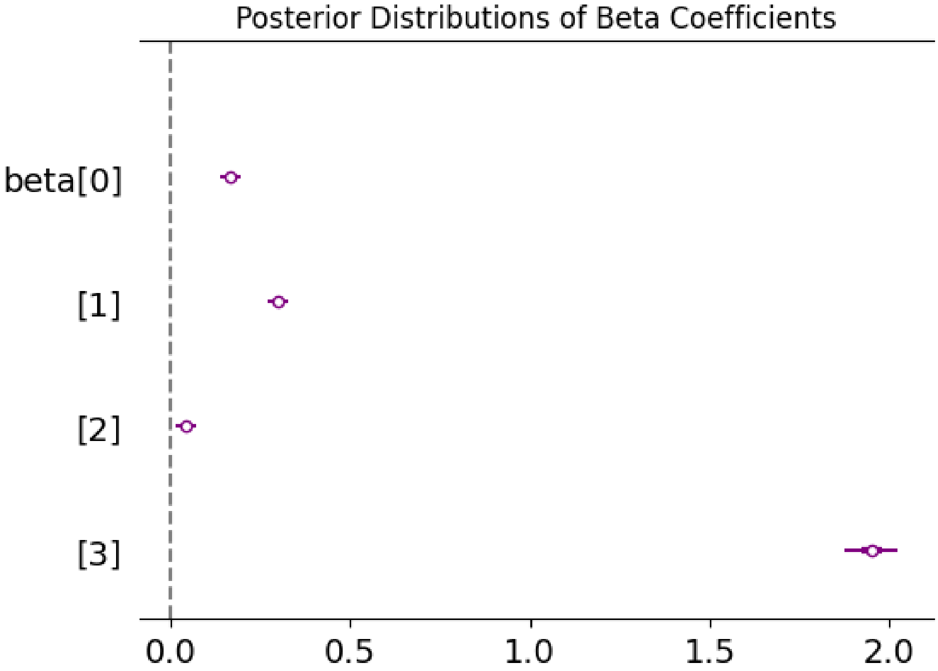


Figure 1: Posterior distributions of β coefficients with 94% credible intervals.

- `age` also shows a positive association (~0.30).
- `bmi` is smaller (~0.16), and `children` is smaller (~0.04).
- $\sigma \approx 0.51$ means the remaining variation in standardized charges is about half an SD after accounting for predictors.

Conclusion: `smoker` is by far the largest effect. Among `bmi`, `age`, and `children`, `age` has the strongest association.

Problem 4: Predictive Distribution and RMSE

I used the Stan model's `generated_quantities` block to generate posterior predictive samples for the *test* set. Then I calculated:

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(\overline{y}_m - y_m)^2},$$

where $\overline{y}_m$ is the posterior mean of the predicted y for test instance m. For the code:

- `rmse_mean` is the average RMSE across all posterior draws.
- `rmse_ci` is a 95% interval showing the uncertainty in out-of-sample RMSE.

Results

Most runs had an RMSE around **0.50–0.55** on the standardized scale of `charges`. Multiplying by `y_std` converts RMSE to the original scale.

What do I lose by using predictive means? I discard the full posterior predictive distribution, ignoring intervals. Computing the RMSE distribution shows the stability of test-set predictions.

Problem 5: Reflection

1. **Posterior predictive checks**: This assignment reinforced the value of predictive distributions over single points. Predictive intervals provide richer insights.
2. **Priors in Bayesian regression**: Even moderate priors (like normal/inverse-gamma) yield stable inferences, but sensitivity checks are crucial for small or skewed data.

Problem 6: Project Pre-Study

This project investigates how different training algorithm configurations—such as optimizer type, learning rate schedules, and regularization strength—and data augmentation strategies like CutMix, RandAugment, and Gaussian noise, influence the generalization and robustness of deep neural networks. The research will begin by formulating hypotheses, such as whether models trained with gradient descent and RandAugment achieve higher accuracy on ImageNet-V2 than those optimized with AdamW and MixUp. The focus will be on evaluating robustness to natural corruptions, adversarial attacks, or both. Clean datasets like CIFAR-10 and ImageNet will be used for training, while separate validation sets will guide early stopping and hyperparameter tuning. To rigorously evaluate robustness, the models will be tested on well-established benchmark suites, including CIFAR-10-C and ImageNet-C for common corruptions, with AutoAttack for adversarial threat scenarios. Parameters of interest include training hyperparameters (e.g., optimizer and augmentation strength) in addition to learned attributes such as weight-space norms. The modeling task combines simulation, via systematic hyperparameter sweeps, with the prediction of performance on unseen corruptions. Performance metrics will include clean accuracy, corruption, adversarial robustness, and expected calibration error (ECE). Prior studies examining the impact of training methods and augmentation on robustness will be reviewed to contextualize this work and highlight how it extends or contrasts with previous findings. The models chosen for evaluation will include architectures like ResNet-18 and Vision Transformer-Tiny, selected for their simplicity and relevance. Training pipelines will vary across optimizer types, learning rate schedules, regularization techniques, and augmentation pipelines, structured within a factorial or grid-based experimental design. To ensure computational faithfulness, the study will incorporate multiple random-seed replicates, track loss, and accuracy over time, apply early stopping, and plan hardware usage accordingly. Tools like PyTorch Lightning and Weights and Biases (WandB) will be used to support reproducibility and automation. Finally, model evaluation will include diagnostics for underfitting and overfitting, robustness stress tests, statistical comparisons between conditions, and visualizations such as reliability diagrams. Statistical methods, including confidence intervals, permutation tests, and Bayesian posterior predictive checks, will be used to determine the practical significance of any observed differences.