Model diagnostics and information criteria

ISTA 410 / INFO 510: Bayesian Modeling and Inference

U. of Arizona School of Information

March 15, 2021

Outline

Today:

- Quick aside on plate notation
- Information theory and predictive accuracy
- Scoring models to avoid overfitting
- Information criteria ☆
- Approximate leave-one-out cross-validation ☆

Next week: take-home midterm

Wednesday: summary and review

What the midterm is about

The midterm is similar in format to a homework, but a bit broader in scope:

- “Start-to-finish” exploration of a modeling problem
    - Exploratory plots
    - Setting up a model
    - Prior predictive checks
    - Inference using MCMC
    - Posterior predictive checks

Take home, open book/note/etc.

Plate notation for Bayesian models

Graphical representation of models

Plate notation:

- A way to display models graphically
- Nodes (circles) represent variables in the model
- Edges (arrows) represent dependencies
- Plates (rectangles) represent exchangeable groups / replications

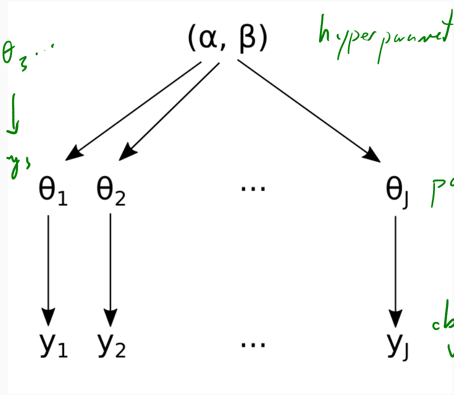
Graphical representation of models

Pooled:

$$\begin{tikzcd}
 & \theta \arrow[dl, dash] \arrow[d, dash] \arrow[dr, dash] & \\
y_1 & y_2 & \cdots
\end{tikzcd}$$

Sep:

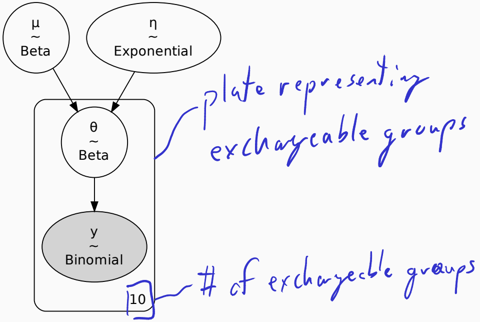
$$\begin{tikzcd}
\theta_1 \arrow[d] & \theta_2 \arrow[d] & \theta_3 \arrow[d] \cdots \\
y_1 & y_2 & y_3
\end{tikzcd}$$



Problem: unwieldy especially if there are multiple layers of hierarchy / many exchangeable groups

Adding plates

Plate notation collapses the exchangeable units:



- Shaded nodes: observed variables
- PyMC3: `pm.model_to_graphviz` (requires `graphviz`)

Information criteria for scoring models

Kullback-Leibler divergence

Entropy: $H = -\sum_i p_i \log_2 p_i$ (measures uncertainty in a distribution)

Kullback-Leibler (KL) divergence:

$$D_{KL}(p, q) = \sum_i \underbrace{p_i(\log_2 p_i}_{\text{Entropy of } p} - \underbrace{\log_2 q_i)}_{\text{cross entropy of } p, q.}$$

"Inflated" entropy when using q to approximate p

Interpretation:

- p is the true outcome distribution
- q is the model predictive distribution
- KL divergence measures incorrectness, in some way

KL divergence for model comparison

We think of $D_{KL}(p, q)$ as measuring the distance from our model, q, to the truth, p.

Problem: we don't know p and never will!

But this isn't an obstacle for comparing models, because if we have two models q and r, then

$$D_{KL}(p, q) - D_{KL}(p, r) = \sum \overset{\text{still unknown}}{\boxed{p_i}} \big(\underbrace{\underline{\log_2(r_i)} - \underline{\log_2(q_i)}}_{\text{these we can calculate.}}\big)$$

i.e. the $H(p)$ term drops out. We still don't know p_i, but we can estimate this from a sample of observations (because the observations are drawn from p_i); e.g. from a MCMC trace

Log score and deviance

Scoring models using log probabilities:

$$\text{log score} \qquad S(q) = \sum_i \log(q_i)$$

where q_i is the probability (density) our model assigns to observation i

need specific values of model params θ

Deviance:

$$D = -2S(q) = -2\sum \log(q_i)$$

- mostly historical reasons

In the Bayesian world, the posterior isn't one model, it's a distribution of models – so we should average:

$$\text{lppd}(y, \theta) = \sum_i \log\left(\frac{1}{S}\sum_s \underline{p(y_i|\theta_s)}\right)$$

probability assigned to each observation by your model with parameters θ_s.

(log pointwise predictive density)

Out-of-sample prediction error

Problem: lppd only looks inside the sample

- Adding parameters nearly always improves fit within the sample
- Eventually, adding parameters reduces accuracy out of the sample (overfitting)
- How can we predict out-of-sample prediction accuracy?
    - Cross-validation
    - Information criteria

Overfitting in action

To demonstrate overfitting, we'll consider a few models fit to fake data.

True data-generating process:

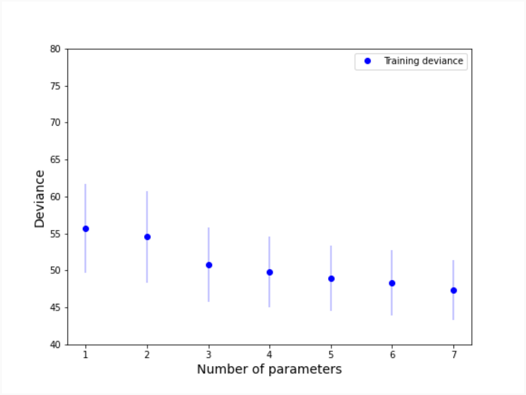
$$y_i \sim \mathrm{Normal}(\mu_i, 1)$$
$$\mu_i = 0.15x_1 - 0.40x_2$$

We'll fit models with the same likelihood and

$$\mu_i = \alpha$$
$$\mu_i = \alpha + \beta_1 x_1$$
$$\mu_i = \alpha + \beta_1 x_1 + \beta_2 x_2$$
$$\mu_i = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$
$$\ldots$$

Overfitting in action

On the training set: Deviance = -2 × log score

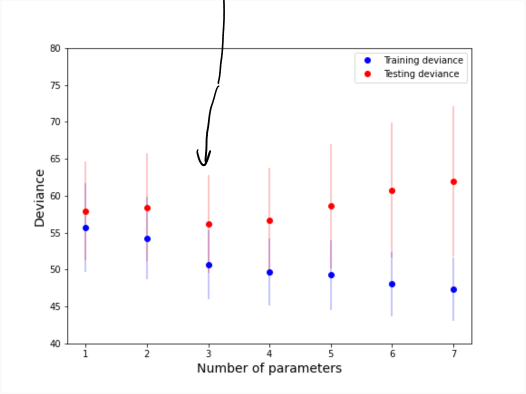


Remember: past 3 parameters, the predictors have no relationship to y in the true data generating process

Overfitting in action

Add in the testing set:

3 = "correct" # of parameters



As expected, the additional parameters just make matters worse.

Goals

Since we use lppd to estimate the fit of our model, our goal with all of these tools is to estimate what our lppd will be on new data.

In other words, ultimately we are trying to estimate some form of:

$$\text{elpd} = \mathbb{E}(\log \underbrace{p(\tilde{y}|y)}_{\text{posterior predictive distribution: distribution of the ``next observation''}})$$

the expected log predictive density of a new data point.

First approach: Akaike information criterion (AIC).

Akaike information criterion (AIC)

AIC: named for Akaike (but he called it "an information criterion"). Attempts to estimate the out-of-sample deviance.

Assuming a point estimate $\hat{\theta}$ for model parameters, calculate the log score/deviance and apply a penalty to correct for overfitting:

max likelihood estimate

$$AIC = D_{\text{train}}|\hat{\theta} + 2k$$

lppd − k ← higher is better

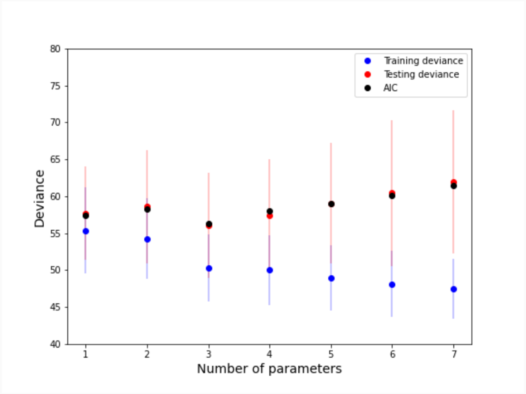
Here $D_{\text{train}}|\hat{\theta}$ is the log predictive density, evaluated at the maximum likelihood estimate, multiplied by -2.

k is the number of parameters. Assumes Gaussian posterior. Where it comes from: Taylor expansion of the log posterior around the posterior mode.

penalty based on # of parameters.

Overfitting in action

Adding the AIC:



We see that for this model, the AIC is a pretty good estimate of the out-of-sample deviance. So, it is reasonable to choose the model with the smallest AIC.

Widely applicable information criterion

WAIC

Introduced by Watanabe (2010); a more Bayesian generalization of the AIC; no requirements about the shape of the posterior.

$$WAIC = -2(\underbrace{\text{lppd}(y,\theta)}_{D_{train}} - \underbrace{\sum_i \text{var} \log(\overbrace{p(y_i|\theta)}^{\text{calculate this log probability score.}})})$$

lppd replaces the training deviance. The overfitting penalty is calculated by taking the log pointwise density and computing the variance over values of the parameters θ.

Reduces to AIC in the special case where AIC is exact: models with flat priors and Gaussian posterior, e.g. ordinary multiple regression

Leave-one-out cross-validation

LOO cross-validation

Idea behind cross-validation:

- Hold out some of your data for evaluation
- Fit the model on the remaining data
- Evaluate the model by estimating lppd on the held-out data
- Repeat, with different partitionings

Types of cross-validation

k-fold CV:

- Partition the data set into *k* equal subsets
- Each subset gets a turn as the hold-out set
- Problem: dependent on the (arbitrary) partition

Leave-one-out CV:

- Each observation gets a turn as the hold-out set
- Exhaustive: no arbitrary choices involved in choosing hold-out sets
- Problem: many re-fits required

Importance sampling

Instead of refitting and resampling the model for each held-out point, we can use the technique of *importance sampling.*

Idea: want to calculate $\mathbb{E}_p(h(\theta))$, but we can't generate samples from $p(\theta)$; however, we can generate random values from an approximation $g(\theta)$. Then,

$$E(h(\theta)) = \frac{\int h(\theta)p(\theta)d\theta}{\int p(\theta)d\theta}$$

Multiply and divide by $g(\theta)$ to get

$$E(h(\theta)) = \frac{\int [h(\theta)p(\theta)/g(\theta)]g(\theta)d\theta}{\int [p(\theta)/g(\theta)]d\theta}$$

Importance sampling

Then, we can estimate using draws θ_s from $g(\theta)$:

$$E(h(\theta)) \approx \frac{\sum_s h(\theta_s) w(\theta_s)}{\sum_s w(\theta_s)}$$

where

$$w(\theta_s) = \frac{p(\theta_s)}{g(\theta_s)}$$

are the *importance weights*.

IS-LOO

In LOO-CV we are trying to estimate $\log p(y_i|y_{-i})$, where y_{-i} denotes the set of observed y values without y_i.

Using importance sampling, we estimate the posterior predictive distribution given y_{-i} by

$$p(\tilde{y}|y_{-i}) = \frac{\sum_s w_{i,s} p(\tilde{y}|\theta_s)}{\sum w_{i_s}}$$

where the weights are $w_{i,s} = \frac{1}{p(y_i|\theta_s)}$

Evaluating this at $\tilde{y} = y_i$, we get

$$p(y_i|y_{-i}) \approx \frac{1}{\frac{1}{S}\sum_s \frac{1}{p(y_i|\theta_s)}}$$

So we can calculate this from a single posterior sample $\{\theta_s\}_{s \in S}$.

Pareto smoothing

In practice, importance sampling behaves poorly if the distribution of importance weights have high variance/long tails.

A fix: Pareto smoothing. Take the top 20% of importance weights, fit a generalized Pareto distribution to them; then replace those top weights with appropriate quantiles from the Pareto distribution.

Pareto smoothing

In practice, importance sampling behaves poorly if the distribution of importance weights have high variance/long tails.

A fix: Pareto smoothing. Take the top 20% of importance weights, fit a generalized Pareto distribution to them; then replace those top weights with appropriate quantiles from the Pareto distribution.

- `pm.compare` can be used to calculate WAIC or PSIS-LOO from sample traces.

Let's see some examples...

Applying WAIC and LOO-CV

We now have two numerical tools for estimating out-of-sample deviance: WAIC and LOO-CV.

- They are two ways to estimate the same thing, so they should give the same results. Best practice is to compute both.
- If they don't agree with one another, you should trust *neither* – hard to tell which, if either, is giving a correct estimate.
- Failure of these two methods is often related to the presence of highly influential points in the data set, and can be used to diagnose these. We'll see a bit more on this later.

Summary

Today:

- information theory / entropy
- Information criteria: AIC and WAIC
- Pareto-smoothed approximate LOO-CV

Next time:

- More details on WAIC and LOO
- Nuances of model comparison
    - Causal inference vs. predictive accuracy
    - "True" models
- Review/questions