Model Comparison

Matthew Talluto

August 18, 2017

UNIVERSITÉ DE SHERBROOKE

Douter de tout ou tout croire sont deux solutions également commodes, qui nous dispensent de réfléchir.

–Henri Poincaré

Introduction to Model Comparison

Why compare models?

Introduction to Model Comparison

Why compare models?

- All models are imperfect
- How good is our model *given the modelling goals?*

Comparing models

Before beginning, evaluate the goals of the comparison

- Predictive performance
- Hypothesis testing
- Reduction of overfitting

If you are asking yourself, “should I use A/B/DIC?”

Remember Betteridge’s law...

Comparing models

Before beginning, evaluate the goals of the comparison

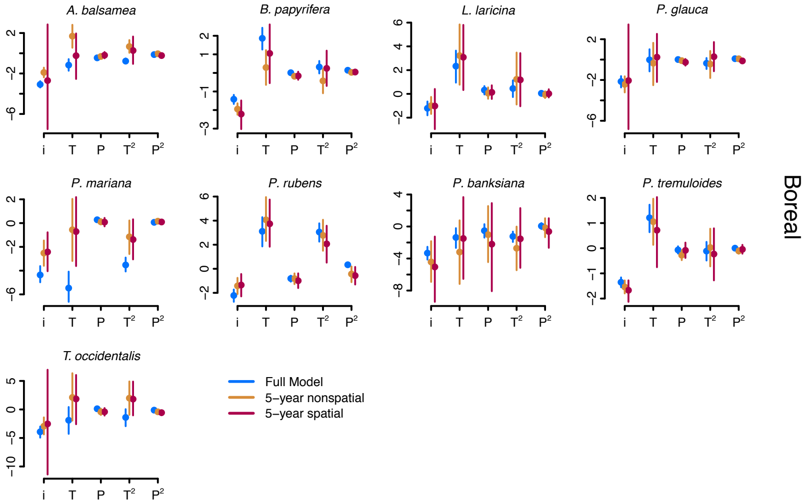
- Predictive performance
- Hypothesis testing
- Reduction of overfitting

If you are asking yourself, "should I use A/B/DIC?"

Remember Betteridge's law...

> *Any headline that ends in a question mark can be answered with the word "NO"*

Informal model comparison



Comparison through evaluation

If the goal is predictive performance, evaluate directly.

- Cross-validation
- k-fold cross validation

Cost: can be computationally intensive (especially for Bayesian). But you are already paying this cost (you ARE evaluating your models, right?)

Comparison through evaluation

If the goal is predictive performance, evaluate directly.

- Cross-validation
- k-fold cross validation

Cost: can be computationally intensive (especially for Bayesian). But you are already paying this cost (you ARE evaluating your models, right?)

Requires selecting an evaluation score

- ROC/TSS (classification)
- RMSE (continuous)
- Goodness of fit
- ...

Bayesian predictive performance

Consider a regression model

$$
\begin{aligned}
\mathrm{pr}(\theta|y, x) &\propto \mathrm{pr}(y, x, |\theta)\mathrm{pr}(\theta) \\
y &\sim \mathcal{N}(\alpha + \beta x, \sigma)
\end{aligned}
$$

From a new value $\hat{x}$ we can compute a posterior prediction $\hat{y} = \alpha + \beta x$

Bayesian predictive performance

We can then compute the *log posterior predictive density* (lppd):

$$\mathrm{lppd} = \mathrm{pr}(\hat{y}|\theta)$$

We can then compute the *log posterior predictive density* (lppd):

$$\mathrm{lppd} = \mathrm{pr}(\hat{y}|\theta)$$

Where is the prior?

Bayesian predictive performance

We want to summarize lppd taking into account:

- an entire set of prediction points $\hat{x} = \{x_1, x_2, \ldots x_n\}$
- the entire posterior distribution of θ
  - (or, realistically, a set of S draws from the posterior distribution)

Bayesian predictive performance

We want to summarize lppd taking into account:

- an entire set of prediction points $\hat{x} = \{x_1, x_2, \ldots x_n\}$
- the entire posterior distribution of θ
    - (or, realistically, a set of S draws from the posterior distribution)

$$\mathrm{lppd} = \sum_{i=1}^{n} \log \left(\frac{1}{S} \sum_{s=1}^{S} \mathrm{pr}(\hat{y}|\theta^s) \right)$$

Bayesian predictive performance

We want to summarize lppd taking into account:

- an entire set of prediction points $\hat{x} = \{x_1, x_2, \ldots x_n\}$
- the entire posterior distribution of θ
    - (or, realistically, a set of S draws from the posterior distribution)

$$\mathrm{lppd} = \sum_{i=1}^{n} \log \left(\frac{1}{S} \sum_{s=1}^{S} \mathrm{pr}(\hat{y}|\theta^s) \right)$$

To compare two competing models θ_1 and θ_2, simply compute lppd_{θ_1} and lppd_{θ_2}, the "better" model (for prediction) is the one with a larger lppd.

What do we do when θ_1 and θ_2 are very different?

What do we do when θ_1 and θ_2 are very different?

Considering the lpd (using the calibration data), it can be proven, when θ_2 is *strictly nested* within θ_1, that $\mathrm{lpd}_{\theta_1} > \mathrm{lpd}_{\theta_2}$.

What do we do when θ_1 and θ_2 are very different?

Considering the lpd (using the calibration data), it can be proven, when θ_2 is *strictly nested* within θ_1, that $\mathrm{lpd}_{\theta_1} > \mathrm{lpd}_{\theta_2}$.

Thus, we require a method for penalizing the larger (or more generally, more flexible) model to avoid simply overfitting, especially when validation data are unavailable.

AIC

$$\mathrm{AIC} = 2k - 2\log \mathrm{pr}(x|\hat{\theta})$$

- $\mathrm{pr}(x|\hat{\theta}) = \max(\mathrm{pr}(x|\theta))$ and k is the number of parameters.
- AIC increases as the model gets worse or the number of parameters gets larger
- $-2\log \mathrm{pr}(x|\hat{\theta})$ is sometimes referred to as *deviance*

$$\mathrm{AIC} = 2k - 2\log \mathrm{pr}(x|\hat{\theta})$$

- $\mathrm{pr}(x|\hat{\theta}) = \max(\mathrm{pr}(x|\theta))$ and k is the number of parameters.
- AIC increases as the model gets worse or the number of parameters gets larger
- $-2\log \mathrm{pr}(x|\hat{\theta})$ is sometimes referred to as *deviance*

What is the number of parameters in a hierarchical model?

DIC

$$D(\theta) = -2\log(\text{pr}(x|\theta))$$

$$D(\theta) = -2 \log(\mathrm{pr}(x|\theta))$$

We still penalize the model based on complexity, but we must estimate how many *effective* parameters there are:

$$p_D = \mathbb{E}[D(\theta)] - D(\mathbb{E}[\theta])$$

DIC

$$D(\theta) = -2\log(\mathrm{pr}(x|\theta))$$

We still penalize the model based on complexity, but we must estimate how many *effective* parameters there are:

$$p_D = \mathbb{E}[D(\theta)] - D(\mathbb{E}[\theta])$$

$$\mathrm{DIC} = D(\mathbb{E}[\theta]) + 2p_D$$

DIC

Pros:

- Easy to estimate
- Widely used and understood
- Effective for a variety of models regardless of nestedness or model size

Cons

- Not Bayesian
- Assume $\theta \sim \mathcal{MN}$
- Modest computational cost

Bayes factor

Consider two competing models θ_1 and θ_2

In classical likelihood statistics, we can compute the likelihood ratio:

$$LR = \frac{MLE(X|\theta_1)}{MLE(X|\theta_2)}$$

Consider two competing models θ_1 and θ_2

In classical likelihood statistics, we can compute the likelihood ratio:

$$LR = \frac{MLE(X|\theta_1)}{MLE(X|\theta_2)}$$

A fully Bayesian approach is to take into account the entire posterior distribution of both models:

$$K = \frac{pr(\theta_1|X)}{pr(\theta_2|X)}$$

Bayes factor

For a single posterior estimate of each model:

$$
\begin{aligned}
K &= \frac{pr(\theta_1|X)}{pr(\theta_2|X)} \\
&= \frac{pr(X|\theta_1)pr(\theta_1)}{pr(X|\theta_2)pr(\theta_2)}
\end{aligned}
$$

Bayes factor

To account for the entire distribution:

$$
\begin{aligned}
K &= \frac{\int \mathrm{pr}(\theta_1|X)d\theta_1}{\int \mathrm{pr}(\theta_2|X)d\theta_2} \\
&= \frac{\int \mathrm{pr}(X|\theta_1)\mathrm{pr}(\theta_1)d\theta_1}{\int \mathrm{pr}(X|\theta_2)\mathrm{pr}(\theta_2)d\theta_2}
\end{aligned}
$$

And others

- Bayesian model averaging
- Reversible jump MCMC

```r
library(mcmc)
suppressMessages(library(bayesplot))


logposterior <- function(params, dat)
{
  if(params[2] <= 0)
    return(-Inf)

  mu <- params[1]
  sig <- params[2]

  lp <- sum(dnorm(dat, mu, sig, log=TRUE)) +
      dnorm(mu, 16, 0.4, log = TRUE) +
      dgamma(sig, 1, 0.1, log = TRUE)
  return(lp)
}
```

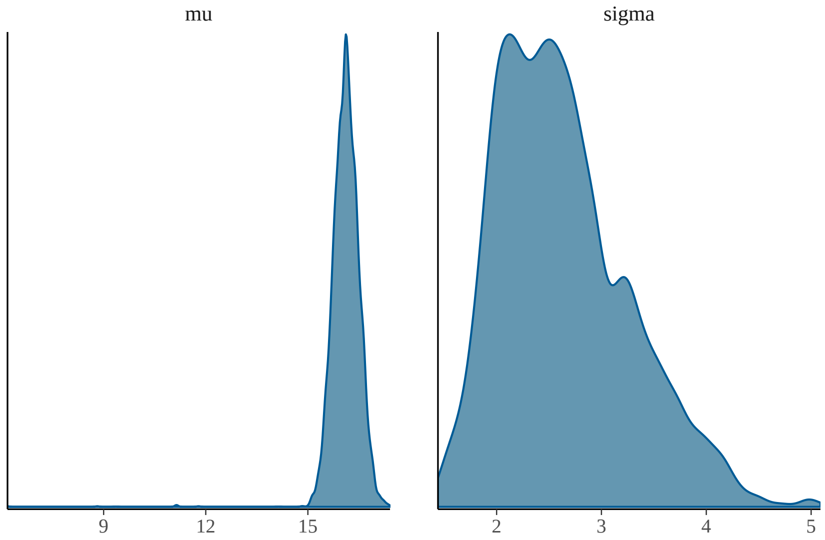
```r
X <- c(15, 19.59, 15.06, 15.71, 14.65, 21.4, 17.64, 18.31,
    15.12, 14.40)
inits <- c(5, 2)
tuning <- c(1.5, 0.5)

model <- metrop(logposterior, initial = inits,
    nbatch = 10000, dat = X, scale = tuning)
model$accept
```

```
## [1] 0.2326
```

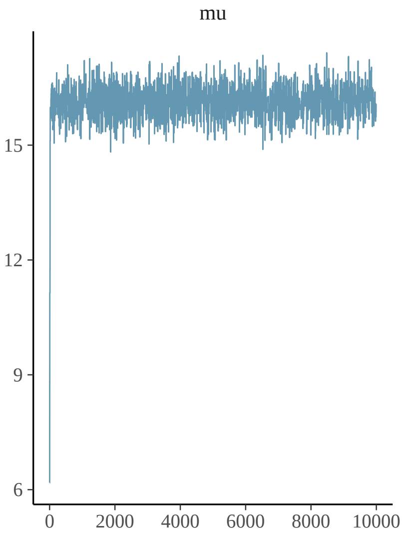
```r
colnames(model$batch) = c('mu', 'sigma')
colMeans(model$batch)
```

```
##        mu     sigma
## 16.114213  2.635871
```

Software



Software



Other software

- mcmc
- LaplacesDemon
- JAGS
- Stan