Bayesian Compromise Estimators

Valentin Zulj

2023-10-24

Introduction

- The common form of a statistical analysis is given by the following two steps:
  - Select the best model
  - Apply the model as if it was the truth/the best overall
- The first step usually relies on either
  - model selection criteria (WAIC/*IC/CV), often data driven, or
  - scientific theory.
- However, it is quite easy to criticize the model selection step. For example:
  - Model selection can be unstable. Small changes of input data may yield radically different model choice.
  - Very different models can have similar performance overall. Then, discriminating between them is often difficult.
- What are the possible consequences? Poor generalizability and inference.
  - Unstable selection means the model may perform badly with out-of-sample data.
  - Ad-hoc choices between similar models may yield sub-optimal selection.
  - Model selection uncertainty not properly represented in final analysis
- Compromise modeling is one way of dealing with the issue. Particularly good at countering poor generalizability.

Compromise Modeling – General Idea

Suppose that φ is some quantity of interest, and that

- There are K candidate models under consideration, and
- Each candidate model produces φ_k as and approximation of φ.

Then, compromise modeling entails us a weighted average of the candidates,

$$\bar{\varphi} = \sum_{k=1}^{K} w_k \varphi_k,$$

as the final approximation of φ. Here w_k are model specific weights that can be estimated to suit the purpose of the analysis.

Example: Variable Selection in Linear Regression

Let $\mathbf{y} = (y_1, \dots, y_n)^T$ be a vector of (continuous) outcomes and $\mathbf{X}$ be a matrix of covariates to use in linear regression.

Problem: Which subset $\mathbf{X}_k \subset \mathbf{X}$ of variables gives the best regression model?

Example: Variable Selection in Linear Regression

Let $\mathbf{y} = (y_1, \ldots, y_n)^T$ be a vector of (continuous) outcomes and $\mathbf{X}$ be a matrix of covariates to use in linear regression.

Problem: Which subset $\mathbf{X}_k \subset \mathbf{X}$ of variables gives the best regression model?

Common solution: Use some off-the-shelf procedure to chose "optimal" subset.

Example: Variable Selection in Linear Regression

Let $\mathbf{y} = (y_1, \ldots, y_n)^T$ be a vector of (continuous) outcomes and $\mathbf{X}$ be a matrix of covariates to use in linear regression.

Problem: Which subset $\mathbf{X}_k \subset \mathbf{X}$ of variables gives the best regression model?

Common solution: Use some off-the-shelf procedure to chose "optimal" subset.

Alternative solution: Consider several different subsets, and combine their information by a weighted average. For example,

- If φ is the posterior predictive distribution, $\bar{\varphi} = \sum_k w_k \cdot p(\tilde{\mathbf{y}}|\mathbf{y}, \mathbf{X}_k)$,
- If φ is the posterior predictive mean, $\bar{\varphi} = \sum_k w_k \cdot E[\tilde{\mathbf{y}}|\mathbf{y}, \mathbf{X}_k]$.

Note that both $p(\tilde{\mathbf{y}}|\mathbf{y}, \mathbf{X}_k)$ and $E[\tilde{\mathbf{y}}|\mathbf{y}, \mathbf{X}_k]$ are derived as usual. The new step in this procedure is finding suitable weights $w_1, \ldots, w_K$. In Bayesian theory, Bayesian model averaging and Bayesian stacking are the prominent ways of doing so.

Bayesian Model Averaging

Bayesian model averaging (BMA) is concerned with $\varphi = p(\Delta|D)$, where

- Δ could be a new observation, $\tilde{\mathbf{y}}$, or a vector of regression coefficients, , and
- D is the data.

Bayesian Model Averaging

Bayesian model averaging (BMA) is concerned with $\varphi = p(\Delta|D)$, where

- Δ could be a new observation, $\tilde{\mathbf{y}}$, or a vector of regression coefficients, , and
- D is the data.

The BMA posterior probability for Δ is given as

$$p(\Delta|D) = \sum_{k=1}^{K} p(\Delta|M_k, D) \cdot p(M_k|D).$$

Here, M_k denotes the k:th candidate model, and

- $p(\Delta|M_k, D)$ is the posterior probability for Δ given model k.
- $p(M_k|D)$ is the posterior probability of the model M_k.

Compare to the compromise posterior predictive on last slide, and note that $w_k = p(M_k|D)$ shows how to weight each model.

Problems in Practice

- Posterior distributions of advanced models can be hard (impossible) to find. For BMA, this amounts to
  - The usual difficulty of finding $p(\Delta | M_k, D)$, and
  - The additional step of finding the model posterior, $p(M_k | D)$. In particular, computing the integrated likelihood.
- Some simple problems have analytical solutions (see e.g. Raftery, Madigan, and Hoeting 1997 for a linear regression example).
- MCMC Model Composition (MCMCMC) simplifies things by generating a Markov chain that moves through the model space (I think it can be applied using the `BMA` package).
- Another issue of BMA has to do with the behavior of the weights in large samples. To discuss this, some further notation has to be introduced.

$\mathcal{M}$-open and $\mathcal{M}$-closed

To formalize the last drawback of BMA, and to motivate the use of Bayesian stacking, the properties of the candidate models employed need to be considered. Thus, let $\mathcal{M} = \{M_1, \dots, M_K\}$ be the set of candidate models. Then

- $\mathcal{M}$-closed means the true data generating model is included in $\mathcal{M}$, although it is not known which of the candidates it is, while
- $\mathcal{M}$-complete means that the true model is *not* in $\mathcal{M}$, but we still use $\mathcal{M}$ since the true model may be too complicated in terms of computations, interpretations, etc.
- $\mathcal{M}$-open means that the true model is *not* in $\mathcal{M}$, and there is no knowledge of how to specify an explicit form of the true model.

$\mathcal{M}$-open and $\mathcal{M}$-closed

To formalize the last drawback of BMA, and to motivate the use of Bayesian stacking, the properties of the candidate models employed need to be considered. Thus, let $\mathcal{M} = \{M_1, \dots, M_K\}$ be the set of candidate models. Then

- $\mathcal{M}$-closed means the true data generating model is included in $\mathcal{M}$, although it is not known which of the candidates it is, while
- $\mathcal{M}$-complete means that the true model is *not* in $\mathcal{M}$, but we still use $\mathcal{M}$ since the true model may be too complicated in terms of computations, interpretations, etc.
- $\mathcal{M}$-open means that the true model is *not* in $\mathcal{M}$, and there is no knowledge of how to specify an explicit form of the true model.

It is known that, as $n \to \infty$, the BMA weight of the candidate closest to the true model (in terms of KL divergence) tends to 1. That is,

- For $\mathcal{M}$-closed, this is great since BMA will chose the true model.
- For $\mathcal{M}$-complete/open, BMA clearly selects the wrong model.

I would argue that $\mathcal{M}$-complete/open is more realistic. So what to do?

Bayesian Stacking

Some preliminaries: Let

- $\mathbf{w} = (w_1, \ldots, w_k)^T$, and suppose it belongs to some set $\mathcal{W}$, and
- $S(P, Q)$ be a scoring rule, measuring the similarity of two distributions P and Q.

Bayesian Stacking

Some preliminaries: Let

- $\mathbf{w} = (w_1, \dots, w_k)^T$, and suppose it belongs to some set $\mathcal{W}$, and
- $S(P, Q)$ be a scoring rule, measuring the similarity of two distributions P and Q.

Then, Bayesian stacking weights are given by

$$\mathbf{w}^* = \underset{\mathbf{w} \in \mathcal{W}}{\operatorname{argmax}} \; S\left(\sum_{k=1}^{K} w_k p(\tilde{\mathbf{y}} | \mathbf{y}, M_k), p_{\text{true}}(\tilde{\mathbf{y}} | \mathbf{y}) \right).$$

That is, $\mathbf{w}^*$ is the weight vector that maximizes the similarity between the stacked posterior predictive and the true posterior predictive distribution

Bayesian Stacking

Of course, $p_{\text{true}}(\tilde{\mathbf{y}}|\mathbf{y})$ is not known, and some kind of empirical approximation is required. One way of estimating $\mathbf{w}^*$ is using leave-one-out cross-validation.

- Clyde and Iversen (2013) introduce it in Bayesian setting,
- Le and Clarke (2017) theoretically motivate use of CV for weight estimation,
- Yao et al. (2018) use general scoring rules.

Bayesian Stacking

Of course, $p_{\text{true}}(\tilde{\mathbf{y}}|\mathbf{y})$ is not known, and some kind of empirical approximation is required. One way of estimating $\mathbf{w}^*$ is using leave-one-out cross-validation.

- Clyde and Iversen (2013) introduce it in Bayesian setting,
- Le and Clarke (2017) theoretically motivate use of CV for weight estimation,
- Yao et al. (2018) use general scoring rules.

Using the weight vector estimated by cross-validation, the stack is given by

$$\sum_{k=1}^{K} w_k^* p(\tilde{\mathbf{y}}|\mathbf{y}, M_k),$$

which is very similar to the *frequentist jackknife model averaging*.

Scoring Rules

For Bayesian stacking, it is common to use either

- The $\log$ score, $LS(P, y) = \log[p(y)]$, or
- The *energy* score, $ES(P, y) = \frac{1}{2} E_P \|Y - Y'\|^\beta - \mathbb{E}_p \|Y - y\|^\beta$. Here, Y and Y' both follow P independently. $\beta = 2$ is common in practice.

The major difference is that stacking using the $\log$ score gives a stacked posterior *distribution*, while stacking using the $\beta = 2$ energy score gives a stacked posterior *mean*.

Depending on the objective of the analysis, either approach may be suitable.

Example Using `rstanarm` and `loo`

`loo::stacking_weights` gives $\log$-score weights. See stylized example below.

```
library("rstanarm"); library("loo")

# Fitting the candidate models
cand1 <- stan_glm(y ~ X1, data = df)
cand2 <- stan_glm(y ~ X1 + X2, data = df)
cand2 <- stan_glm(y ~ X1 + X2 + X3, data = df)

# LOO-CV approximation
loo1 <- loo(cand1); loo2 <- loo(cand2); loo3 <- loo(cand3)

# Pointwise LOO ELPD
lpd_point <- cbind(loo1$pointwise[,"elpd_loo"],
                   loo2$pointwise[,"elpd_loo"],
                   loo3$pointwise[,"elpd_loo"])
stacking_weights(lpd_point) # Estimates the weights
```

An example using real data is given in the enclosed `R` script.

Research Example

Ongoing work aims to evaluate Bayesian stacking using frequentist asymptotics. In particular, the focus is to establish the *oracle property*

$$\frac{\|\mathbf{y} - \sum_k w_k^* E[\mathbf{y}|M_k]\|^2}{\inf_{\mathbf{w}\in\mathcal{W}} \|\mathbf{y} - \sum_k w_k E[\mathbf{y}|M_k]\|^2} \xrightarrow{p} 1.$$

This means that no candidate, nor any other average using the same candidates and weights from $\mathcal{W}$, provides smaller asymptotic error than Bayesian stacking.

Research Example

Ongoing work aims to evaluate Bayesian stacking using frequentist asymptotics. In particular, the focus is to establish the *oracle property*

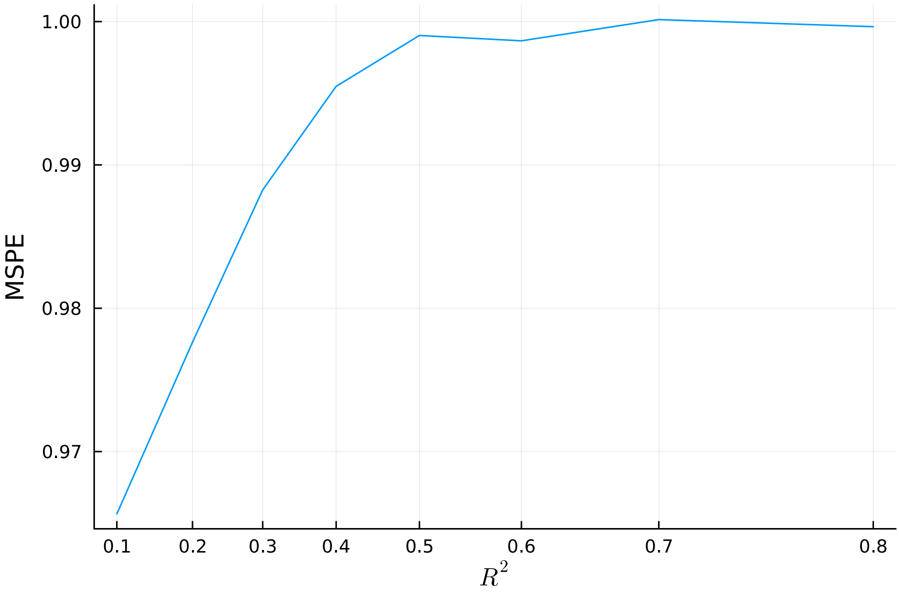
$$\frac{\|\mathbf{y} - \sum_k w_k^* E[\mathbf{y}|M_k]\|^2}{\inf_{\mathbf{w}\in\mathcal{W}} \|\mathbf{y} - \sum_k w_k E[\mathbf{y}|M_k]\|^2} \xrightarrow{p} 1.$$

This means that no candidate, nor any other average using the same candidates and weights from $\mathcal{W}$, provides smaller asymptotic error than Bayesian stacking.

To date, the oracle has been established for Bayesian stacking of linear regression models using $\mathcal{N}(\mathbf{0}, \mathbf{S})$, as the prior for β, where $\mathbf{S} > 0$ is symmetric.

The results of a simulation are given on the next slide. The curve gives the ratio of the Bayesian stacking squared error to the squared error of the best candidate.

Research Example



References

Clyde, Merlise, and Edwin S Iversen. 2013. "Bayesian Model Averaging in the m-Open Framework." In *Bayesian Theory and Applications*. Oxford: Oxford University Press.

Le, Tri, and Bertrand Clarke. 2017. "A Bayes Interpretation of Stacking for m-Complete and m-Open Settings." *Bayesian Analysis* 12 (3): 807–29.

Raftery, Adrian E., David Madigan, and Jennifer A. Hoeting. 1997. "Bayesian Model Averaging for Linear Regression Models." *Journal of the American Statistical Association* 92 (437): 179–91.

Yao, Yuling, Aki Vehtari, Daniel Simpson, and Andrew Gelman. 2018. "Using Stacking to Average Bayesian Predictive Distributions (with Discussion)." *Bayesian Analysis* 13 (3): 917–1003.