7: Evaluating, comparing and expanding models

Taylor

University of Virginia

Introduction

This chapter focuses mostly on quantifying a model's predictive capabilities for the purposes of model selection and expansion.

New Notation!

1. f is the true model
2. y is the data we use to estimate our model
3. $\tilde{y}$ is the future (time series) or alternative (not time series) data that we test our predictions on
4. $p_{\text{post}}(\tilde{y}) = p(\tilde{y} \mid y)$
5. $p_{\text{post}}(\theta) = p(\theta \mid y)$
6. $E_{\text{post}}[\cdot]$ is taken with respect to $p(\theta \mid y)$

Definitions

A **scoring rule/function** $S(p, \tilde{y})$ is a function that takes

1. the distribution you're using to forecast p (ppd, or likelihood with estimated parameters), and
2. a realized value $\tilde{y}$

and then gives you a real-valued number/score/utility. Higher is better, although this convention isn't always followed in the literature.

Keep in mind that the realized value cannot be used to fit the data.

Examples

Example: $S(p, \tilde{y}) = -(\tilde{y} - E_p[\tilde{y}])^2$

Example: $S(p, \tilde{y}) = \log p(\tilde{y})$

Definitions

Future/unseen data is unknown, so we must take the expected score under the true distribution f:

$$E_f[S(p, \tilde{y})].$$

A scoring rule is **proper** if the above expectation is minimized when $f = p$.

A scoring rule is **local** if $S(p, \tilde{y})$ only depends on $p(\tilde{y})$ (don't care about events that didn't happen).

Note, when we are dealing with a logarithmic scoring rule, $E[-2 \log p(\tilde{y})]$ is often called an **information criterion.** The book switches back and forth between dealing with expected score, and information criteria.

Examples

Example: $S(p, \tilde{y}) = -(\tilde{y} - E_p[\tilde{y}])^2$
Most common, perhaps not local or proper for non-Gaussian data.

Example: $S(p, \tilde{y}) = \log p(\tilde{y})$
Obviously local. Proper, too (homework question).

Problem

We are generally not able to evaluate the expectation because we don't know f. However, we may be able to wait for new out-of-sample data and use a Monte-Carlo approach:

$$n^{-1} \sum_{i=1}^{n} S(p, \tilde{y}^i) \to E_f[S(p, \tilde{y})]$$

as $n \to \infty$

Problem

We are generally not able to evaluate the expectation because we don't know f. However, we may be able to wait for new out-of-sample data and use a Monte-Carlo approach:

$$n^{-1} \sum_{i=1}^{n} S(p, \tilde{y}^i) \to E_f[S(p, \tilde{y})]$$

as $n \to \infty$

If we can afford to wait for an infinite amount of data, though, what is the point of trying to predict it?

Problem

NB: the textbook focuses on $S(p, \tilde{y}) = \log p(\tilde{y})$, and the data are iid (after conditioning on the parameter). They call the following quantity the "elppd:"

expected log pointwise predictive density

$$\begin{aligned} E_f[\log p(\tilde{y})] &= E_f[\log p(\tilde{y}_n) \cdots p(\tilde{y}_n)] \\ &= E_f\left[\sum_{i=1}^{n} \log p(\tilde{y}_i)\right] \\ &= \sum_{i=1}^{n} E_f\left[\log p(\tilde{y}_i)\right] \end{aligned}$$

Problem

For the moment let's use $p(\tilde{y}) = p_{\text{post}}(\tilde{y})$

The "elppd" is not obtainable because

1. you don't know f (can't directly integrate)
2. you don't have $\tilde{y}$ (no Monte-Carlo)

Problem

For the moment let's use $p(\tilde{y}) = p_{\text{post}}(\tilde{y})$

The "elppd" is not obtainable because

1. you don't know f (can't directly integrate)
2. you don't have $\tilde{y}$ (no Monte-Carlo)

Using y for $\tilde{y}$, we can come up with a rough elppd estimate called the "lppd"

log pointwise predictive density

$$\text{lppd} = \log p_{\text{post}}(y) = \sum_{i=1}^{n} \log p_{\text{post}}(y_i)$$

Problem

There'a also the problem that arises where we cannot evaluate

$$p_{\text{post}}(y) = \int p(y \mid \theta) p(\theta \mid y) \mathrm{d}\theta = E_{\text{post}}[p(y \mid \theta)]$$

The "computed lppd" again uses y for $\tilde{y}$, but it also uses Monte-Carlo to sample from the posterior

log pointwise predictive density

$$\text{computed lppd} = \log \hat{p}_{\text{post}}(y) = \sum_{i=1}^{n} \log \left(\frac{1}{S} \sum_{j=1}^{n} p(y_i \mid \theta^j) \right)$$

A third problem

Don't know f, don't want to wait for $\tilde{y}$...

and unfortunately, plugging the same data that we used for estimation into the predictive distribution might lead us to overfit because this strategy overestimates the average predictive score.

A third problem

Don't know f, don't want to wait for $\tilde{y}$...

and unfortunately, plugging the same data that we used for estimation into the predictive distribution might lead us to overfit because this strategy overestimates the average predictive score.

However, we can get around this in two ways generally:

1. plug in the already-used y data, but then add an extra penalty term (e.g. AIC, DIC, WAIC, etc.)
2. Cross-Validation: split the data y, many different ways, into a train and test set; estimate and evaluate on each split.

Information Criteria

AIC stands for "an information criterion" or "Akaike's Information Criterion." Let k be the number of parameters:

$$\widehat{\text{elpd}}_{\text{AIC}} = \log p(y \mid \hat{\theta}_{\text{MLE}}) - \overbrace{k}^{\text{penalty}}$$

or

$$\text{AIC} = \underbrace{-2 \log p(y \mid \hat{\theta}_{\text{MLE}})}_{\text{a deviance}} + 2k$$

We estimate $\hat{\theta}_{\text{MLE}}$ using y, and we plug y into the log likelihood.

Information Criteria

DIC replaces the point estimate with $\hat{\theta}_{\text{Bayes}} = E[\theta \mid y]$, and replaces the penalty term with p_{DIC}

$$\widehat{\text{elpd}}_{\text{DIC}} = \log p(y \mid \hat{\theta}_{\text{Bayes}}) - p_{\text{DIC}}$$

or

$$\text{DIC} = -2 \log p(y \mid \hat{\theta}_{\text{Bayes}}) + 2 p_{\text{DIC}}$$

Information Criteria

The book gives two ways to estimate p_{DIC}:

1. $p_{\mathrm{DIC}} = 2\left(\log p(y \mid \hat{\theta}_{\mathrm{Bayes}}) - E_{\mathrm{post}}\left[\log p(y \mid \theta)\right]\right)$
2. $p_{\mathrm{DIC\ alt}} = 2\,\mathrm{Var}_{\mathrm{post}}\left[\log p(y \mid \theta)\right]$

Both of these can be approximated using samples from the posterior.

Motivation for p_{DIC}

$$\begin{aligned}
E_f\left[-2\log p(\tilde{y} \mid \hat{\theta}_{\text{Bayes}})\right] &= -2\log p(y \mid \hat{\theta}_{\text{Bayes}}) \\
&+ E_Y\left[-2\log p(\tilde{y} \mid \hat{\theta}_{\text{Bayes}})\right] + 2\log p(y \mid \hat{\theta}_{\text{Bayes}}) \\
&\approx -2\log p(y \mid \hat{\theta}_{\text{Bayes}}) \\
&+ E_{\theta|y}\left[-2\log p(y \mid \theta)\right] + 2\log p(y \mid \hat{\theta}(y)) \\
&= -2\log p(y \mid \hat{\theta}_{\text{Bayes}}) + p_{\text{DIC}}
\end{aligned}$$

Information Criteria

p_{WAIC} either stands for "widely applicable information criterion" or "Watanabe-Akaike information criterion."

The book refers to it as the most "fully Bayesian" of the three, probably because it doesn't plug in point estimates into the likelihood instead of integrating.

$$\widehat{\text{elppd}}_{\text{WAIC}} = \text{lppd} - p_{\text{WAIC}}$$

or

$$\text{WAIC} = -2\text{lppd} + 2p_{\text{WAIC}}$$

where $\text{lppd} = \sum_{i=1}^{n} \log\left(\frac{1}{S}\sum_{s=1}^{S} p(y_i \mid \theta^s)\right)$

Information Criteria

Two ways to estimate

1. $p_{\text{WAIC 1}} = 2\left(\log p(y \mid y) - E_{\theta|y}\left\{\log p(y \mid \theta)\right\}\right)$
2. $p_{\text{WAIC 2}} = \sum_{i=1}^{n} var_{\text{post}}(\log p(y_i \mid \theta))$

Both of these can be approximated using samples from the posterior.

Information Criteria

Motivation for p_{WAIC}:

$$
\begin{aligned}
E_f\left[E_{\theta|y}\left\{-2\log p(\tilde{y}\mid\theta)\right\}\right] &= -2\log p(y\mid y) \\
&\quad + 2\left(\log p(y\mid y) - E_f\left[E_{\theta|y}\left\{\log p(\tilde{y}\mid\theta)\right\}\right]\right) \\
&= -2\log p(y\mid y) \\
&\quad + 2\left(\log p(y\mid y) - E_{\theta|y}\left\{E_f\left[\log p(\tilde{y}\mid\theta)\right]\right\}\right) \\
&\approx -2\log p(y\mid y) \\
&\quad + 2\left(\log p(y\mid y) - E_{\theta|y}\left\{\log p(y\mid\theta)\right\}\right) \\
&= -2\log p(y\mid y) + p_{\text{WAIC1}}
\end{aligned}
$$

Cross-Validation

To assess prediction performance, one may also use **cross-validation**. Here the data is repeatedly partitioned into different training-set-test-set pairs (aka **folds**).

1. The partitions are nonrandom, test sets are disjoint
2. for each split/estimation/prediction, we never use a data point twice
3. for each split/estimation/prediction, we lose parameter estimation accuracy because each training set is smaller than the full set
4. however, we get to average over many prediction scores, which reduces variance
5. there is still a bias that we have to estimate (but it's usually smaller than AIC/DIC/WAIC/etc.)
6. it can be computationally brutal to calculate for some models

Cross-Validation

To assess prediction performance, one may also use **cross-validation**. Here the data is repeatedly partitioned into different training-set-test-set pairs (aka **folds**).

1. The partitions are nonrandom, test sets are disjoint
2. for each split/estimation/prediction, we never use a data point twice
3. for each split/estimation/prediction, we lose parameter estimation accuracy because each training set is smaller than the full set
4. however, we get to average over many prediction scores, which reduces variance
5. there is still a bias that we have to estimate (but it's usually smaller than AIC/DIC/WAIC/etc.)
6. it can be computationally brutal to calculate for some models

The logo of this QA website illustrates the idea nicely!

Cross-Validation

leave-one-out cross-validation (loo-cv) is a special case where each training set is of size 1.

This necessarily implies that each training set is of size $n - 1$, and there are n possible splits.

If this ends up being too computationally expensive, it is also possible to do k-**fold cross-validation**, which selects k splits/folds. This means the size of each test set is n/k, and the size of each training set is $n - n/k = n(1 - 1/k)$

Cross-Validation Notation

We only discuss loo-cv...

$p_{\text{post}(-i)}(y_i)$ is the prediction for the ith point, using the ppd, which uses the posterior distribution conditioning on all values of the data **except the** ith

If this ppd isn't tractable, we can use draws from the posterior as follows:

$$p_{\text{post}(-i)}(y_i) = \frac{1}{S}\sum_{s=1}^{S} p(y_i \mid \theta^s)$$

where θ^s are draws from $p_{\text{post}(-i)}(\theta)$

Cross-Validation Notation

The Bayesian loo-cv estimate for out-of-sample predictive fit is

$$\text{lppd}_{\text{loo-cv}} = \sum_{i=1}^{n} \log p_{\text{post}(-i)}(y_i)$$

There are also bias-corrected versions as well.