36-724 Spring 2006: Model Averaging

Brian Junker

April 5b, 2006

- How to Improve Models?
- Bayesian Approach
- Resampling / Sample Re-Use Approach
How to Improve Models?

Improve for what?

- *Model selection* tries to pick a “best” model (or a few “almost best”…).
  - The models $M_1, \ldots, M_M$ might represent competing “generative” or “scientific” hypotheses for the data-generation process. Maybe…
    - We want to pick the best scientific explanation.
    - We want to pick the best “generative” explanation; etc.
    - We want to pick the most likely/parsimonious/etc. model.
  - What if there is no “best”?
  - What if we just want to know whether the next email is spam?

- *Model averaging* tries to make the “best” prediction by combining predictions of several models.
  - Each of the models $M_1, \ldots, M_M$ might seem equally plausible. Can we make a better prediction by consulting all $M$ models?
  - Each of the models $M_1, \ldots, M_M$ might predict in one region of the data space really well, but be *terrible* by global measures of prediction error. Can we combine them in some way to make a better global predictor?
  - What if we want a simple, generative or scientific, explanation?

It turns out model selection methods can be adapted to the task of model averaging.

We’ll look at this from two perspectives:

- **Bayesian**
- **Resampling / Sample Re-Use**
Bayesian Approach

Model Selection
For $\ell = 1, \ldots, M$ we know that
\[
p(y|\mathcal{M}_\ell) = \int p(y|\theta, \mathcal{M}_\ell)p(\theta|\mathcal{M}_\ell)d\theta
\]
and we may choose model $\mathcal{M}_m$ if it maximizes the posterior probability
\[
p(\mathcal{M}_m|y) = \frac{p(y, \mathcal{M}_m)}{p(y)} = \frac{p(y|\mathcal{M}_m)p(\mathcal{M}_m)}{\sum_{\ell} p(y|\mathcal{M}_\ell)p(\mathcal{M}_\ell)}
\]
If we take $p_\ell = p(\mathcal{M}_\ell) \equiv 1/M$, and we use the Laplace approximation to
the posterior under each model
\[-2 \log p(y|\mathcal{M}_\ell) \approx BIC_\ell = -2 \log p(y|\hat{\theta}, \mathcal{M}_\ell) + d_\ell \log(N)\]
then
\[
p(\mathcal{M}_m|y) \approx \frac{e^{-\frac{1}{2}BIC_m}}{\sum_{\ell} e^{-\frac{1}{2}BIC_\ell}}
\]

Model Averaging
Here the goal is prediction. For one model, the Bayesian paradigm tells us
\[
p(y_{test}|y_{train}) = \int p(y_{test}|\theta)p(\theta|y_{train})d\theta
\]
For the $\ell$th model $\mathcal{M}_\ell$,
\[
p(y_{test}|y_{train}, \mathcal{M}_\ell) = \int p(y_{test}|\theta, \mathcal{M}_\ell)p(\theta|y_{train}, \mathcal{M}_\ell)d\theta
\]
Now letting the model index $\ell$ play the role of parameter $\theta$ and replacing integration with summation, for $M$ models we have
\[
p(y_{test}|y_{train}) = \sum_{\ell=1}^{M} p(y_{test}|y_{train}, \mathcal{M}_\ell)p(\mathcal{M}_\ell|y_{train})
\]
\[
= \sum_{\ell=1}^{M} \int p(y_{test}|\theta, \mathcal{M}_\ell)p(\theta|y_{train}, \mathcal{M}_\ell)d\theta p(\mathcal{M}_\ell|y_{train})
\]
Basically, we weight each model’s prediction with its posterior probability!
Methods

We need methods for calculating $p(M_\ell | y)$.

- BIC approximation

$$p(M_m | y) \approx \frac{e^{-\frac{1}{2}BIC_m}}{\sum_{\ell} e^{-\frac{1}{2}BIC_\ell}}$$

- Formally, the Bayesian model selection model is the same as the latent class / clustering / discrete mixture problem we saw earlier
  - The data-generation process “belongs to” latent class $\ell$ if $y \sim M_\ell$
  - $P(M_\ell | y) = P[\text{data generation process “belongs to” class } \ell | y]$  
  - So we need to calculate these posterior probabilities . . .

- Let $z_\ell = 1$ iff $y \sim M_\ell$; we are trying to estimate $P[z_\ell = 1 | y]$. This is like estimating class membership probabilities in the latent class model

- MCMC methods can work. However, jumping from one model to another, it is difficult to maintain the “detailed balance” equations that were needed for MCMC convergence.
  * Reversible-jump MCMC (Green, *Bmka*, 1995, 711–732) designed for this.
  * Some simple versions appear to be implemented in WinBUGS (I haven’t tried them though!)
- The $z_\ell$’s are missing data and an E-M algorithm can be employed to get posterior-mode estimates of $P[z_\ell = 1 | y]$. More later . . .
Resampling / Sample Re-Use Approach

Model Selection

We have just been exploring these ideas. They are all based on trying to estimate the extra-sample prediction error $\text{Err} = E[L(Y, \hat{f}(X))]$ by re-using the data.

- Training-set prediction error.
- Adjusting training-set error for “optimism” by various complexity penalties (AIC, BIC, SRM, ...).
- Split-half cross-validation prediction error.
- $K$-fold cross-validation prediction error.
- Various Bootstrap prediction error ideas.

We choose the model with the smallest prediction error (or a collection of models with similar small prediction error).

Model Averaging

Here “averaging” has a simple meaning. If we construct $B$ “replicates” of the data, we want to average or aggregate predictions from classifiers built from each “data replicate”.

We will pursue two simple versions of this idea:

- Bagging
- Stacking, aka Boosting
Stacking or Boosting

The idea here is similar to projection-pursuit regression. Basically we build up a prediction function

\[ \hat{y} = \hat{f}_1(x) + \hat{f}_2(x) + \cdots + \hat{f}_B(x) \]

by

- Fitting \( \hat{y} = \hat{f}_1(x) \) alone and calculating the residuals \( r_1 = y - \hat{f}_1(x) \)
- Fitting \( \hat{r}_1 = \hat{f}_2(x) \) and calculating the residuals \( r_2 = r_1 - \hat{f}_2(x) \)
- Fitting \( \hat{r}_2 = \hat{f}_3(x) \) and calculating the residuals \( r_3 = r_2 - \hat{f}_3(x) \)
- \( \ldots \) and repeating until it doesn’t do any more good.

Bagging

Bagging is (B)ootstrap (Ag)gregating models. The idea is simple:

- Construct \( B \) bootstrap samples \( S_1, \ldots, S_B \).
- Train our prediction/classification function separately on each classifier, yielding \( \hat{f}^{a_1}(x), \ldots, \hat{f}^{a_B}(x) \)
- The bagged estimator is then

\[ \hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{i=1}^{B} \hat{f}^{a_i}(x) \]