Machine Learning for Economists: Part 1 – Ensemble Learning

Michal Andrle
International Monetary Fund

Washington, D.C.,
October, 2018

Disclaimer #1:

The views expressed herein are those of the authors and should not be attributed to the International Monetary Fund, its Executive Board, or its management.

Ensemble Learning

Often, a **combination of models** performs better than a single model

Both different algorithms or variations of one algorithm can be combined into **model ensembles**

$$y = G\{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_R(\mathbf{x})\} \tag{1}$$

Model Ensembles

Some popular approaches:

- **Bagging** – Bootstrap Aggregating
  Aggregating models estimated on multiple re-sampled datasets
- **Stacking**
  'Optimal' Weighted average of various algorithms. . .
- **Boosting** – Gradient Boosting
  Greedy fitting of adaptive basis-function models, each stage lowering the error

Note that these three approaches are three very different ones!

Bagging – Bootstrap Aggregating

Breiman (1996)

Take data sample $D = \{(y_1, x_1), (y_2, x_2), \ldots, (y_N, x_N), \}$ and create R bootstrap samples by re-sampling with replacement, $D_r^*, r = 1, 2, \ldots, R$.

For each bootsrap sample, D_r^*, train a separate model $f_r^*(x)$.

The ‘bagged’ (bagging) estimate is given by

$$f_{bagg} = F\{f_1(x), f_2(x), \ldots, f_R(x)\}, \tag{2}$$

most often by

$$f_{bagg} = \frac{1}{R}\sum_{r=1}^{R} f_r^*(x). \tag{3}$$

Bagging – Bootstrap Aggregating

Bagged estimate, f_{bagg} **differs** from the estimate on the full sample **only if** $f(.)$ is nonlinear or adaptive function of the data, or if the aggregation, $F(.)$, is nonlinear (e.g. median, etc.)

Bagging is often used with **trees**, where $f(x)$ is non-linear a and for each bootstrap sample D_r^* the tree often has different features and splits. . .

Bagging **may** help with lowering the variance of the final estimator for models sensitive to data-sample changes (unstable)

To some extent, bagged estimate is analogous to Bayesian posterior distribution mean. . .

Bagging – Bootstrap Aggregating

Bagging will work for 'uncorrelated' models.

For uncorrelated variables, x_i, with variance σ^2 the variance of their average is lower:

$$\mathrm{var}(\bar{x}) = \frac{\sigma^2}{n} \tag{4}$$

For variables with pair-wise correlation ρ, the variance of the average is

$$\mathrm{var}(\bar{x}) = \frac{\sigma^2}{n} + \frac{n-1}{n}\rho\,\sigma^2. \tag{5}$$

'Wisdom of crowds' requires diverse and independent members of the crowd...

Model Averaging and Stacking

Model averaging strives to create a superior learner, f_s, by combining a set of learners $\{f_1(x), \ldots, f_K(x)\}$:

$$f_s = \sum_{k=1}^{K} w_k f_k(x) \tag{6}$$

Model Combination

How to choose the weights, w, in

$$f_s = \sum_{k=1}^{K} w_k f_k(x)?$$

An intuitive (and wrong) solution seems to be

$$\mathbf{w} = \arg\max_w \sum_{i=1}^{n} [y_i - \sum_{k=1}^{K} w_k \widehat{f}_k(x)]^2 \tag{7}$$

Problem:
Weights estimation and $\widehat{f}_k$ estimation uses the same data, irrespective of model complexity...

Stacking

Stacking estimates the weights, w_k, using prediction error on a **hold-out** (validation) set...

Let $\widehat{f}_{-\mathbf{H},k}(x)$ be the k-th model estimated excluding datapoints in the hold out set, H: $(y_i, x_i), i \in \mathbf{H}$.

We find the weights by:

$$\mathbf{w} = \text{arg max}_w \sum_{i \in \mathbf{H}}^{n} [y_i - \sum_{k=1}^{K} w_k \widehat{f}_{-\mathbf{H},k}(x)]^2 \tag{8}$$

For good performance on regression problems, Breiman (1996) shows that the weights should be non-negative, $w_k \geq 0$.

Wonkish: Careful, BMA is NOT model combination

Bayesian model averaging (BMA) can improve upon a single model

$$p(y|\mathbf{z}) = \sum_{k=1}^{K} p(M_k|\mathbf{z}) \times p(y|M_k, \mathbf{z}) \tag{9}$$

Models predictions are weighted by the posterior probability of the model, M_k, given the available data, $\mathbf{z} = (\mathbf{x}, \mathbf{y})$.

Note that $p(M_k|\mathbf{z}) \in [0, 1]$.

Note that BMA is not ‘model combination’ (Minka, 2002) and Monteith et al. (2011).

BMA assumes that the K models express mutually exclusive and exhaustive ways how the data were generated. BMA integrates the uncertainty but is more a model selection than combination. . .

Boosting [Gentle Introduction]

Freund and Schapire (1995), Breiman (1996) and Friedman (2001)

Boosting is one of the most powerful approaches to supervised learning. . .

Boosting creates an additive model of the form

$$f(x) = \sum_{k=1}^{K} \beta_k \phi(x; \gamma_k), \tag{10}$$

where $\phi(x; \gamma_k)$ are **base learners** (or ‘weak learners’).

‘Basis expansions’ used in econ/stats all the time (wavelets, func. interpolation, . . .)

With boosting, the parameters (β_k, γ_k) are estimated in a **stage-wise** manner, not simultaneously. . .

Boosting

Boosting estimates the first learner, $\phi_1(x;\gamma_1)$ to explain the data and sets $f_1 = \beta_1\phi_1(x;\gamma_1)$.

The new learner, $\phi(x;\gamma_2)$ is estimated to minimize the **residuals from the previous stage**, i.e.

$$\begin{aligned}(\beta_2,\gamma_2) &= \arg\min \sum_{i=1}^{N} ((y_i - f_1) - \beta_2\phi(x_i;\gamma_2))^2 && (11)\\ &= \arg\min \sum_{i=1}^{N} (y_i - [f_1 + \beta_2\phi(x_i;\gamma_2)])^2 && (12)\end{aligned}$$

and the **aggregate model is updated**

$$f_2 = f_1 + \beta_2\phi(x;\gamma_2) = \beta_1\phi(x;\gamma_1) + \beta_2\phi(x;\gamma_2) \tag{13}$$

At each stage, new learner is fit to remaining residuals. . .

Boosting

Forward Stage-Wise Additive Modeling

For a general loss function, $L[y, f(x)]$, the **boosting** algorithm is as follows:

1. Initialize $f_0 = 0$.
2. For $k = 1$ to K:
   - 2.1 **Estimate** the new base learner

$$(\beta_k, \gamma_k) = \arg\max_{\beta,\gamma} \sum_{i=1}^{N} L[y_i, f_{k-1}(x_i) + \beta_k \phi(x_i; \gamma_k)] \tag{14}$$

   - 2.2 **Update** the model aggregate

$$f_k(x) = f_{k-1}(x) + \alpha \times \beta_k \phi(x; \gamma_k),\ 0 < \alpha \leq 1 \tag{15}$$

Learning speed, α, further slows down learning, acts as a **shrinkage**, and helps to avoid overfitting. . .

Boosting

Historical Perspective…

Boosting is due to Schapire and Freund (1995) and their 'AdaBoost.M1' algorithm for classification…

Boosting worked so well, peole started to wonder WHY?

It became clear that AdaBoost can be viewed as stage-wise additive model optimizing the exponential loss criterion…

Modern understanding of boosting is due to Breiman (1998) and Friedman (2000), who showed it is a **gradient descent** in function space, 'gradient boosting'.

In boosting, base learners are often **trees**, or **stumps** – one-level decision trees…

Wonkish: Gradient Boosting

Follows Friedman (2000) but simplified

For some loss functions and base learners, solving (14) can be difficult...

Then, $\beta\phi(x;\gamma)$ in the update step $f_k = f_{k-1} + \beta\phi(x;\gamma)$ can be viewed as a 'best greedy step' as indicated by the negative **gradient**

The gradient $g_k(x_i) \equiv [\partial L(y_i, F_{k-1}(x_i)/\partial F_{k-1}(x_i)]$ can be evaluated but it's just numbers and we need a function...

So we estimate a functional approximation for the gradient

$$(\beta, \gamma) = \arg\min \sum_{i=1}^{N} [-g_k(x_i) - \beta\phi(x_i;\gamma)]^2. \tag{16}$$

The parameterized estimate of the gradient is used in the function update

$$f_k(x) = f_{k-1}(x) + \alpha \times \beta_k\phi(x;\gamma_k). \tag{17}$$

Note that for the special case $L = (y - f(x))^2$, the gradient is $g(x) = (y - f(x)) = \varepsilon_i$ and $[-g(x) - \beta\phi(x;\gamma)]^2 = [\varepsilon - \beta\phi(x;\gamma)]^2$. Thus, $\phi(x;\gamma)$ is fit to residuals from the previous stage...

Thank you for your patience…