Bagging and Model Averaging

Wei Li

Syracuse University

Spring 2021

Introduction

Introduction

Bootstrap aggregating (bagging) and boosting are useful techniques to improve the predictive performance of models.

- Boosting may also be useful in connection with many other models, e.g. for additive models with high-dimensional predictors
- bagging is most prominent for improving tree algorithms.

Bagging in Regression problems

Bagging in Regression problems

Let training data be $z = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, from z, obtain the prediction $\hat{f}(x)$ at input x.

Bagging works as follows:

1. Generate a bootstrap sample
$$(X_1^*, Y_1^*), \ldots, (X_n^*, Y_n^*)$$
and compute the bootstrapped estimator $\hat{f}^*(\cdot)$.
2. Repeat step B times, yielding
$$\hat{f}^{*1}(\cdot), \ldots, \hat{f}^{*B}(\cdot)$$
3. Aggregate the bootstrap estimates
$$\hat{f}_{bag}(\cdot) = B^{-1} \sum_{i=1}^{B} \hat{f}^{*i}(\cdot)$$

The bagging algorithm is nothing else than an approximation

$$\hat{f}_{bag}(\cdot) \approx E^* \left[\hat{f}^*(\cdot)\right]$$

here, E^* is with respect to the bootstrap distribution

The bagged estimate will differ from the original estimate $\hat{f}(x)$ **only when** the latter is a nonlinear or adaptive function of the data:

A trivial identity hints at some properties of bagging: write (the theoretical version of bagging with $B = \infty$)

$$\hat{f}_{bag}(\cdot) \cong \hat{f}(\cdot) + \left(E^* \left[\hat{f}^*(\cdot)\right] - \hat{f}(\cdot)\right)$$
$$\cong \hat{f}(\cdot) + \text{ bootstrap bias estimate.}$$

- Adding the bootstrap bias estimate
- What we can hope for is a variance reduction at the price of a higher bias.
    - True if $\hat{f}(\cdot)$ is a tree-based estimator

Bagging in Classification

Hard classification

Hard classification: If $\hat{f}^{*b}(x)$ is indicator-vector, with one 1 and $K-1$ 0's (**hard classification**), equivalently, $\hat{h}^{*b}(x) \in \{1, \ldots, K\}$.

- This is called "Consensus vote" (majority vote): selects the most "votes" from the B classifiers.

$$\hat{f}_{bag,k}(x) = \frac{1}{B} \sum_{b=1}^{B} 1\{\hat{h}^{*b}(x) = k\}$$

so the classifier is

$$\hat{f}_{bag}(x) = \arg\max_{k} \hat{f}_{bag,k}(x)$$

Soft classification

If $\hat{f}^{*b}(x) = \left(\hat{p}_1^{*b}, \ldots, \hat{p}_K^{*b}\right)$, the estimates of class probabilities $\hat{p}_k^{*b} := \hat{P}^b(Y = k \mid X = x), k = 1, \cdots, K$.

Consider "averaging probabilities": the bagged estimates are the average prediction at x from B classifiers

$$\hat{p}_{\text{bag},k}(x) = B^{-1} \sum_{b=1}^{B} \hat{p}_k^{*b}(x), \quad k = 1, \cdots, K$$

$$\hat{p}_{bag}(x) = \arg max_k \hat{f}_{\text{bag},k}(x)$$

Effects of bagging

In the classification under $0-1$ loss, bagging a good classifier can make it better, but bagging a bad classifier can make it worse.

The **Wisdom of Crowds** asserts that the collective knowledge of a diverse and independent body of people typically exceeds the knowledge of any single individual and can be harnessed by voting (Surowiecki, 2004).

The Wisdom of Crowds

Consider an example: optimal decision at x be $h^*(x) = 1$ in a two-class example.

- Suppose each of the weak learners h^{*b} have an error-rate $e_b = e < 0.5$,
- let $S_1(x) = \sum_{b=1}^{B} I\left(h^{*b}(x) = 1\right)$ be the consensus vote for class 1.
- the weak learners are assumed to be independent

So

$$S_1(x) \sim \text{Bin}(B, 1-e),$$

$$\Pr(S_1/B > 1/2) \to 1,$$

as B gets large.

Note the assumption here is that all weak learners are independent of each other.

Model Averaging

Bayesian model averaging

- J models, denoted by $M_j, j = 1, 2, \ldots, J$. T
- ζ be some quantity of interest, e.g., prediction $f(x)$ at some x.

The posterior distribution of ζ is

$$\Pr(\zeta \mid Z) = \sum_{m=1}^{M} \Pr(\zeta \mid M_m, Z) \Pr(M_m \mid Z)$$

with posterior mean

$$\mathrm{E}(\zeta \mid Z) = \sum_{m=1}^{M} \mathrm{E}(\zeta \mid M_m, Z) \Pr(M_m \mid Z)$$

Note:

- $\Pr(M_m \mid Z)$ are the weights proportional to posterior probability of each model.
- The **committee method** simply uses a simple unweighted average (equal probability).

$Pr(M_m|Z)$

To obtain $Pr(M_m|Z)$, there are two methods

- Use BIC
- carry out a full Bayesian procedure

Frequentist model averaging

Given predictions $\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_M(x)$, under squared-error loss, we can seek the weights $w = (w_1, w_2, \ldots, w_M)$ such that

$$\hat{w} = \underset{w}{\operatorname{argmin}} \mathrm{E}_P \left[Y - \sum_{m=1}^{M} w_m \hat{f}_m(x) \right]^2$$

The solution is the population LS of Y on
$\hat{F}(x)^T \equiv \left[\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_M(x) \right]$:

$$\hat{w} = \mathrm{E}_P \left[\hat{F}(x) \hat{F}(x)^T \right]^{-1} \mathrm{E}_P [\hat{F}(x) Y]$$

One can show that

$$\mathrm{E}_P \left[Y - \sum_{m=1}^{M} \hat{w}_m \hat{f}_m(x) \right]^2 \leq \mathrm{E}_P \left[Y - \hat{f}_m(x) \right]^2, \quad \forall m$$

so combining models never makes things worse, at the population level.

Stacking

- $\hat{f}_m^{-(i)}(x)$ is the **cross validation prediction** at x,i.e.,using the m th model learned from the dataset with the i th training example removed.
- Then the stacking weights are given by

$$\hat{w}^{\text{st}} = \arg\min_{w} \sum_{i=1}^{n} \left(y_i - \sum_{m=1}^{M} w_m \hat{f}_m^{-(i)}(x_i) \right)^2$$

- The final prediction at point x is

$$\sum_{m} \hat{w}_m^{\text{st}} \hat{f}_m(x)$$

- the weights may be restriced to be sum up to 1.

- Stacking and model selection with via leave-one-out cross-validation are closely related.
- If we restrict the minimization to weight vectors w that have one unit weight and the rest zero, this leads to a model choice m with smallest leave-one-out cross-validation error.
- Rather than choose a single model, stacking combines them with estimated optimal weights. This will often lead to better prediction, but less interpretability than the choice of only one of the M models.

Bumping

Bumping

A technique for finding a better single model:

Bumping uses bootstrap sampling to move randomly through model space.

- Draw bootstrap samples $\mathbf{Z}^{*1}, \ldots, \mathbf{Z}^{*B}$, for $b = 1, \ldots, B$
  - By convention, the original training sample is included in the set of bootstrap samples.
- Fit the model to Z^{*b} giving $\hat{f}^{*b}(x)$.
- Choose the model obtained from bootstrap sample $\hat{b}$ which minimizes training error:

$$\hat{b} = \arg\min_b \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{f}^{*b}\left(x_i\right) \right)^2$$

- The prediction is

$$\hat{f}^{*\hat{b}}(x)$$

- By perturbing the data, bumping tries to move the fitting procedure arround to good areas of model space. If a few data points are causing the procedure to find a poor solution, any bootstrap sample that omits those data points should produce a better solution.
- Bumping compares different models on the training data, one must ensure models to compare have roughly similar complexity. That is, for each b, the fitted model could be different but should have similar complexity. Often, the fitted model is from the same class of model. (In the case of trees, this mean growing trees to the same number of terminal nodes on each bootstrap sample).