Foundations of Machine Learning
AI2000 and AI5000

FoML-37
Model Combination

Dr. Konda Reddy Mopuri
Department of AI, IIT Hyderabad
July-Nov 2025

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

DiL
Data-driven Intelligence & Learning Lab

So far in FoML

- Intro to ML and Probability refresher
- MLE, MAP, and fully Bayesian treatment
- Supervised learning
    a. Linear Regression with basis functions
    b. Bias-Variance Decomposition
    c. Decision Theory - three broad classification strategies
    d. Neural Networks
- Unsupervised learning
    a. K-Means, Hierarchical, and GMM for clustering
- Kernelizing linear Models
    a. Dual representation, Kernel trick, SVM (max-margin classifier)
- Tree-based Methods

For today

- Model combination

Single vs Multiple models

- Combining multiple models (often) → improved performance

Single vs Multiple models

- Combining multiple models (often) → improved performance
    - E.g., train L different models and use the average of the predictions made by each model

Single vs Multiple models

- Combining multiple models (often) → improved performance
  - E.g., train L different models and use the average of the predictions made by each model
- Such combinations of models → Committees

Model combination - variants

- Boosting
  - Training multiple models in sequence
  - Error function used to train a models depends on the performance of the previous model

Model combination - variants

- Select one of the models to make the prediction
    - Choice of the model is a function of the input
    - Different models are responsible for making predictions in different regions

Model combination - variants

- Select one of the models to make the prediction
  - Choice of the model is a function of the input
  - Different models are responsible for making predictions in different regions
- E.g., decision trees
  - Selection process is a sequence of binary selections

Bayesian Model Averaging vs. Model combination

Model combination

- E.g., density estimation using a mixture of Gaussians (GMM)
- Several Gaussian components are combined probabilistically
  - Binary latent variable z is responsible for generating x

Model combination

$$p(\mathbf{x}, \mathbf{z})$$

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}).$$

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(\mathbf{X}) = \prod_{n=1}^{N} p(\mathbf{x}_n) = \prod_{n=1}^{N} \left[\sum_{\mathbf{z}_n} p(\mathbf{x}_n, \mathbf{z}_n) \right].$$

Each data sample has a corresponding latent variable

Bayesian Model Averaging

- Several different models indexed by h and prior p(h)
  - E.g., GMM or mixture of Cauchy distributions

Bayesian Model Averaging

- Several different models indexed by h and prior p(h)
    - E.g., GMM or mixture of Cauchy distributions

Marginal distribution over data

$$p(\mathbf{X}) = \sum_{h=1}^{H} p(\mathbf{X}|h)p(h).$$

Bayesian Model Averaging

- Several different models indexed by h and prior p(h)
  - E.g., GMM or mixture of Cauchy distributions

Marginal distribution over data

$$p(\mathbf{X}) = \sum_{h=1}^{H} p(\mathbf{X}|h)p(h).$$

One model is responsible for generating the whole data, p(h) captures our uncertainty as to which model that is

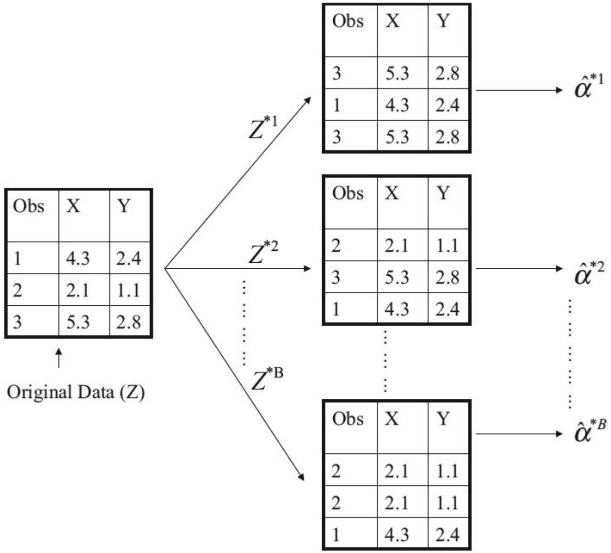
Decision Trees

- Suffer from high variance
  - Different splits of training data → quite different results
- Random Forests, and Boosting reduce the variance
  - These are general purpose procedures

Bagging

Bootstrap

- Creates multiple datasets sampled with replacement
- Used to quantify the uncertainty associated with a given estimator



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Original Data (Z)

Z^{*1}

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

$\hat{\alpha}^{*1}$

Z^{*2}

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

$\hat{\alpha}^{*2}$

Z^{*B}

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

$\hat{\alpha}^{*B}$

Bootstrap

- *Averaging a set of observations reduces the variance*
- Take many training sets, train separate models and average the resulting predictions

Bagging

- Compute B different models using B separate training sets

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x).$$

Bagging

- Useful for decision trees (improves predictions)
- B trees are trained on the bootstrapped datasets
  - Trees are grown deep without pruning
  - High variance and low bias
  - Aggregating → low variance

Bagging

- Prediction aggregation
  - Average for regression
  - Majority voting for classification

Random Forests

Random Forests

- Improvement over bagged trees
    - Via decorrelating them

Random Forests

- Similar to bagging, we build several trees
- When building trees
  - During a split, a random subset of predictors are chosen as candidates
  - Instead of all the ‘p’ predictors, only a random sample of ‘m’ (~√p) are allowed to conduct split

Random Forests

- Suppose one strong predictor and multiple moderate predictors are present in the data
- Bagging → most trees use the strong predictor at the top

Random Forests

- → Most of them will be similar → predictions will be correlated
- Averaging doesn’t lead to a large reduction in variance

Random Forests

- RF overcome this by forcing each split to use a subset of predictors
- Majority of the splits do not consider the strong predictor
- → decorrelating the trees

Boosting

Boosting

- Bagging → multiple copies → trees are learned independently
- Boosting → Trees are grown sequentially
  - each tree is grown using information from previously grown trees

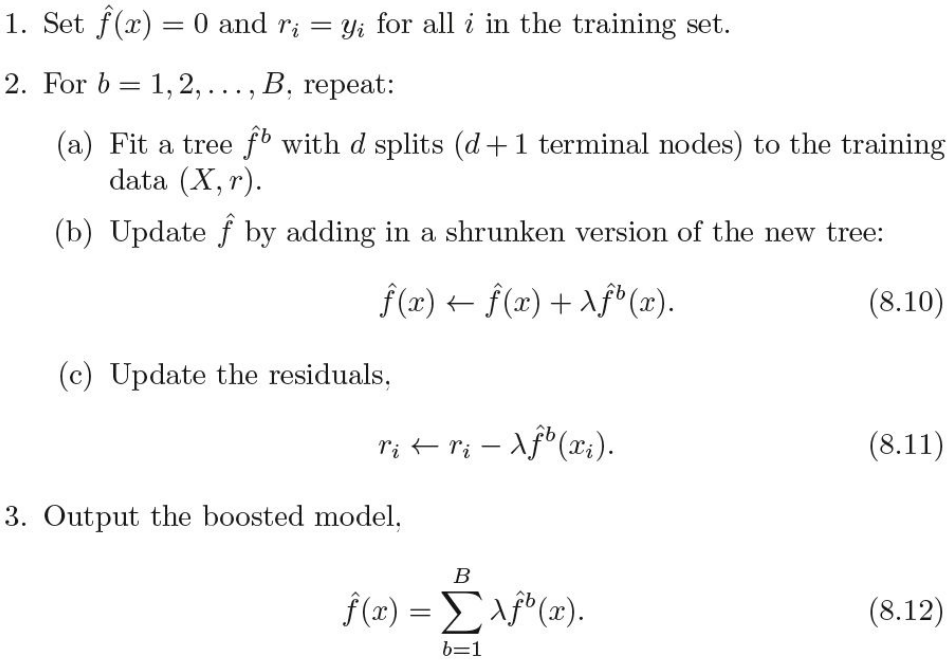
Boosting

- Does not involve bootstrap sampling
- instead each tree is fit on a modified version of the original data set

Boosting

- Given the current model, we fit a decision tree to the residuals from the model.
- Fit a tree using the current residuals, rather than the outcome Y, as the response.

Boosting for Regression Trees



1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.

2. For $b = 1, 2, \ldots, B$, repeat:

   (a) Fit a tree $\hat{f}^b$ with d splits ($d+1$ terminal nodes) to the training data (X, r).

   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:

   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x). \qquad (8.10)$$

   (c) Update the residuals,

   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \qquad (8.11)$$

3. Output the boosted model,

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \qquad (8.12)$$

Rough