Ensemble methods
Model averaging

Common strategy in ML: Combine multiple predictors

- Called ensemble methods
Model averaging

Common strategy in ML: Combine multiple predictors
▶ Called ensemble methods

Simplest ensemble method for regression: (uniform) model averaging
▶ Given predictors $f_1, f_2, \ldots, f_M$, return the ensemble predictor $f_{\text{avg}}$ defined by

$$f_{\text{avg}}(\vec{x}) := \frac{1}{M} \sum_{t=1}^{M} f_t(\vec{x})$$
Model averaging

Common strategy in ML: Combine multiple predictors

▶ Called *ensemble methods*

Simplest ensemble method for regression: *(uniform) model averaging*

▶ Given predictors $f_1, f_2, \ldots, f_M$, return the ensemble predictor $f_{\text{avg}}$ defined by

$$f_{\text{avg}}(\bar{x}) := \frac{1}{M} \sum_{t=1}^{M} f_t(\bar{x})$$

**Question:** When is this preferable to *model selection* — i.e., (attempting to) pick the best of the $f_t$?
Theorem. Let \( f_{\text{avg}} := \frac{1}{M} \sum_{t=1}^{M} f_t \). Then

\[
\mathbb{E}[(f_{\text{avg}}(\vec{X}) - Y)^2] = \frac{1}{M} \sum_{t=1}^{M} \mathbb{E}[(f_t(\vec{X}) - Y)^2] - \frac{1}{2M^2} \sum_{s=1}^{M} \sum_{t=1}^{M} \mathbb{E}[(f_s(\vec{X}) - f_t(\vec{X}))^2]
\]

Model averaging is preferable to model selection if:

1. All \( f_t \)'s have similar MSE, and
2. All \( f_t \)'s predict very differently from each other

This may be the case if:

- Same ML algorithm is used to obtain all \( f_t \)
- ML algorithm has "high variability"
Theorem. Let $f_{\text{avg}} := \frac{1}{M} \sum_{t=1}^{M} f_t$. Then

$$
\mathbb{E}[(f_{\text{avg}}(\vec{X}) - Y)^2] = \frac{1}{M} \sum_{t=1}^{M} \mathbb{E}[(f_t(\vec{X}) - Y)^2] - \frac{1}{2M^2} \sum_{s=1}^{M} \sum_{t=1}^{M} \mathbb{E}[(f_s(\vec{X}) - f_t(\vec{X}))^2]
$$

Model averaging is preferable to model selection if:

1. All $f_t$’s have similar MSE, and
Theorem. Let \( f_{\text{avg}} := \frac{1}{M} \sum_{t=1}^{M} f_t \). Then

\[
\mathbb{E}[(f_{\text{avg}}(\vec{X}) - Y)^2] = \frac{1}{M} \sum_{t=1}^{M} \mathbb{E}[(f_t(\vec{X}) - Y)^2] - \frac{1}{2M^2} \sum_{s=1}^{M} \sum_{t=1}^{M} \mathbb{E}[(f_s(\vec{X}) - f_t(\vec{X}))^2]
\]

Model averaging is preferable to model selection if:

1. All \( f_t \)'s have similar MSE, and
2. All \( f_t \)'s predict very differently from each other
**Mean squared error of model averaging**

**Theorem.** Let $f_{\text{avg}} := \frac{1}{M} \sum_{t=1}^{M} f_t$. Then

$$
\mathbb{E}[(f_{\text{avg}}(\vec{X}) - Y)^2] = \frac{1}{M} \sum_{t=1}^{M} \mathbb{E}[(f_t(\vec{X}) - Y)^2] - \frac{1}{2M^2} \sum_{s=1}^{M} \sum_{t=1}^{M} \mathbb{E}[(f_s(\vec{X}) - f_t(\vec{X}))^2]
$$

Model averaging is preferable to model selection if:

1. All $f_t$’s have similar MSE, and
2. All $f_t$’s predict very differently from each other

This may be the case if

- Same ML algorithm is used to obtain all $f_t$
- ML algorithm has “high variability”
Hypothetical scenario

- Running (deterministic) ML algorithm on same training data $M$ times is not helpful.

- Suppose instead we run ML algorithm on multiple (independent) training data $S_1, S_2, \ldots, S_M$.

- Takes advantage of “high variability” ML algorithms!

- However, may not beat running ML algorithm once on $M$ times as many training data!
Hypothetical scenario

- Running (deterministic) ML algorithm on same training data $M$ times is not helpful.
- Suppose instead we run ML algorithm on multiple (independent) training data sets $S_1, S_2, \ldots, S_M$.
  - Takes advantage of “high variability” ML algorithms!
Hypothetical scenario

- Running (deterministic) ML algorithm on same training data $M$ times is not helpful.
- Suppose instead we run ML algorithm on multiple (independent) training data sets $S_1, S_2, \ldots, S_M$.
  - Takes advantage of “high variability” ML algorithms!
- However, may not beat running ML algorithm once on $M$ times as many training data!
“Faking” the multiple training data sets: Bagging

Main idea: Pretend the training data \( S \) is the original population of examples

**Bootstrap aggregating (Bagging):**

- Randomly sample \( M \) independent data sets \( S_1^*, S_2^*, \ldots, S_M^* \) from \( S \), each of size \( n = |S| \)
  - Each \( S_t^* \) is a **bootstrap resampling** of \( S \)
  - Use sampling-with-replacement
- Run ML algorithm on each \( S_t^* \) to get predictors \( f_1, f_2, \ldots, f_M \)
- Return \( f_{avg} = \frac{1}{M} \sum_{t=1}^{M} f_t \)
Random forests

**Random forests**: Bagging + variant of decision tree learning algorithm as the ML algorithm

Main idea: Bagging with greedy training heuristic with stopping rule that leads to large-size trees

To increase "variability", introduce additional randomness in learning algorithm

Only change compared to original greedy training heuristic:
- When finding best split for a tree node, instead of enumerating through all $d$ features, only enumerate through a random subset of $k$ features (Default: $k = d/3$, but this is a hyperparameter)

Leo Breiman  Adele Cutler
Random forests: Bagging + variant of decision tree learning algorithm as the ML algorithm

- Main idea: Bagging with greedy training heuristic with stopping rule that leads to large-size trees

Only change compared to original greedy training heuristic:

- When finding best split for a tree node, instead of enumerating through all $d$ features, only enumerate through a random subset of $k$ features (Default: $k = d/3$, but this is a hyperparameter)

Leo Breiman  
Adele Cutler
Random forests

**Random forests**: Bagging + variant of decision tree learning algorithm as the ML algorithm

- Main idea: Bagging with greedy training heuristic with stopping rule that leads to large-size trees
- To increase “variability”, introduce additional randomness in learning algorithm

Leo Breiman

Adele Cutler
Random forests

Random forests: Bagging + variant of decision tree learning algorithm as the ML algorithm

- Main idea: Bagging with greedy training heuristic with stopping rule that leads to large-size trees
- To increase “variability”, introduce additional randomness in learning algorithm

Only change compared to original greedy training heuristic:

- When finding best split for a tree node, instead of enumerating through all $d$ features, only enumerate through a random subset of $k$ features

  (Default: $k = d/3$, but this is a hyperparameter)

Leo Breiman

Adele Cutler
Recap

- **Ensemble methods:** general term for methods that combine multiple predictors
- **Model averaging:** advantageous when you have collection of predictors of comparable quality but highly variable behavior
- **Bagging:** strategy to “simulate” a scenario where model averaging is advantageous
  - **Random forests:** Bagging + decision trees + extra randomness
- **Many other ensemble methods:** e.g.,
  - Non-uniform model averaging
  - Boosting
  - Stacking

which are all related to *linear models!*