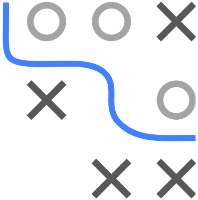
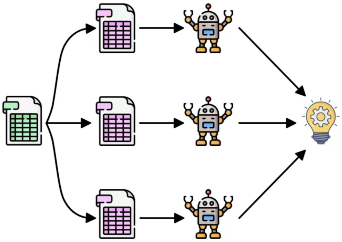
Introduction to Machine Learning



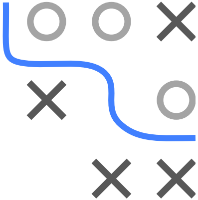
Random Forest Bagging Ensembles



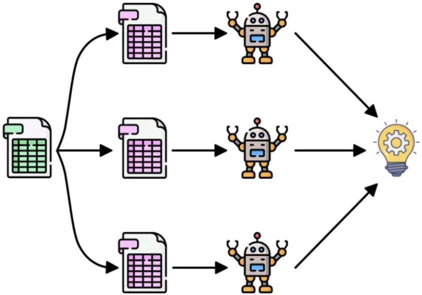
Learning goals

- Understand idea of bagging
- Be able to explain the connection between bagging and bootstrap
- Understand why bagging improves predictive performance

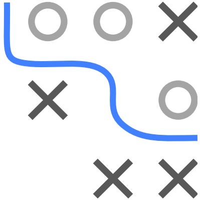
BAGGING



- Bagging is short for **B**ootstrap **Agg**regation
- **Ensemble method**, combines models into large "meta-model"; ensembles usually better than single **base learner**
- Homogeneous ensembles always use same BL class (e.g. CART), heterogeneous ensembles can use different classes
- Bagging is homogeneous

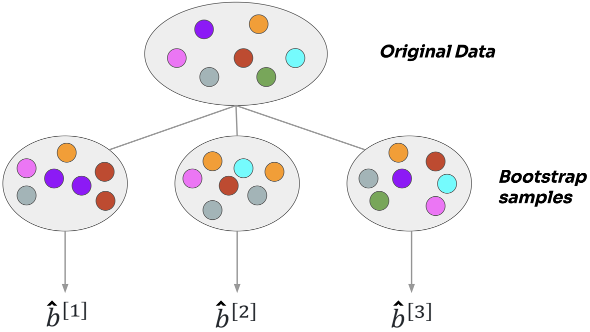


TRAINING BAGGED ENSEMBLES



Train BL on M **bootstrap** samples of training data $\mathcal{D}$:

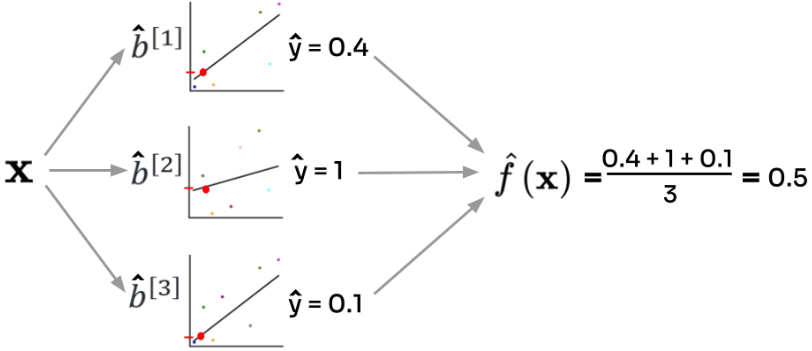
- Draw n observations from $\mathcal{D}$ with replacement
- Fit BL on each bootstrapped data $\mathcal{D}^{[m]}$ to obtain $\hat{b}^{[m]}$

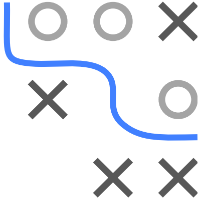


- Data sampled in one iter called “in-bag” (IB)
- Data not sampled called “out-of-bag” (OOB)

PREDICTING WITH A BAGGED ENSEMBLE

Average predictions of *M* fitted models for ensemble:
(here: regression)





BAGGING PSEUDO CODE

Bagging algorithm: Training

```
Input: Dataset D, type of BLs, number of bootstraps M
for m = 1 → M do
    Draw a bootstrap sample D^[m] from D
    Train BL on D^[m] to obtain model b̂^[m]
end for
```

Input: Dataset $\mathcal{D}$, type of BLs, number of bootstraps M
for $m = 1 \rightarrow M$ **do**
  Draw a bootstrap sample $\mathcal{D}^{[m]}$ from $\mathcal{D}$
  Train BL on $\mathcal{D}^{[m]}$ to obtain model $\hat{b}^{[m]}$
end for

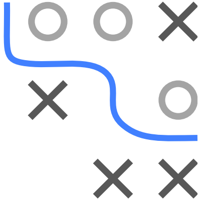
Bagging algorithm: Prediction

Input: Obs. $\mathbf{x}$, trained BLs $\hat{b}^{[m]}$ (as scores $\hat{f}^{[m]}$, hard labels $\hat{h}^{[m]}$ or probs $\hat{\pi}^{[m]}$)
Aggregate/Average predictions

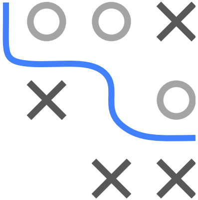
$$\hat{f}(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} \left(\hat{f}^{[m]}(\mathbf{x}) \right) \quad \text{(regression / decision score, use } \hat{f}_k \text{ in multi-class)}$$

$$\hat{h}(\mathbf{x}) = \underset{k \in \mathcal{Y}}{\arg\max} \sum_{m=1}^{M} \mathbb{I} \left(\hat{h}^{[m]}(\mathbf{x}) = k \right) \quad \text{(majority voting)}$$

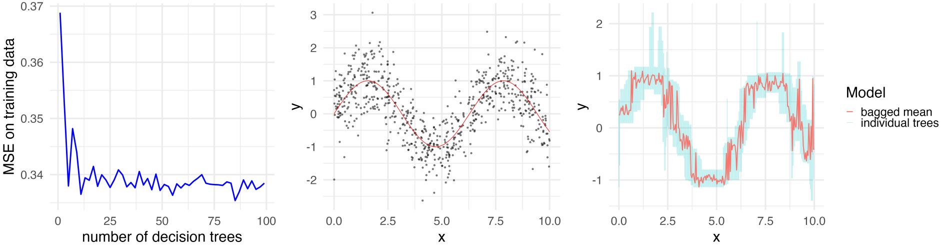
$$\hat{\pi}_k(\mathbf{x}) = \begin{cases} \frac{1}{M} \sum_{m=1}^{M} \hat{\pi}_k^{[m]}(\mathbf{x}) & \text{(probabilities through averaging)} \\ \frac{1}{M} \sum_{m=1}^{M} \mathbb{I} \left(\hat{h}^{[m]}(\mathbf{x}) = k \right) & \text{(probabilities through class frequencies)} \end{cases}$$



WHY/WHEN DOES BAGGING HELP?



- Bagging reduces the variability of predictions by averaging the outcomes from multiple BL models
- It is particularly effective when the errors of a BL are mainly due to (random) variability rather than systematic issues



- Increasing **nr. of BLs** improves performance, up to a point, optimal ensemble size depends on inducer and data distribution

MINI BENCHMARK

Bagged ensembles with 100 BLs each on `spam`:
Bagging seems especially helpful for less stable learners like CART