CS 480/680 Introduction to Machine Learning

Lecture 11

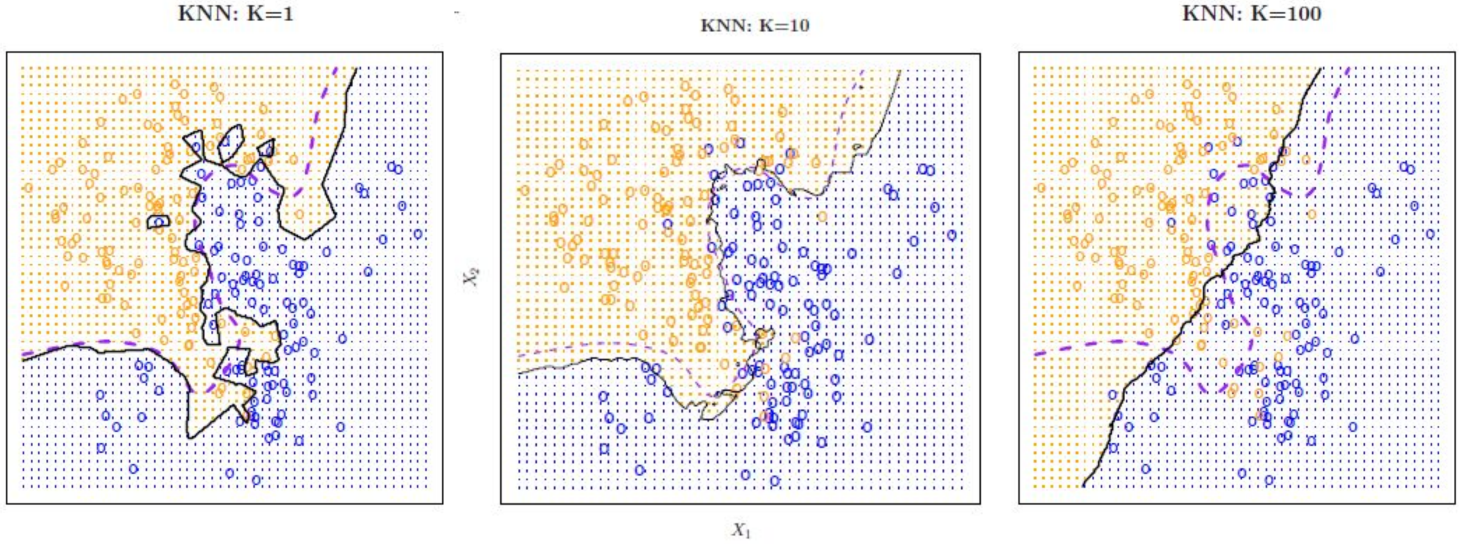
Ensemble Methods

Kathryn Simone

22 October 2024

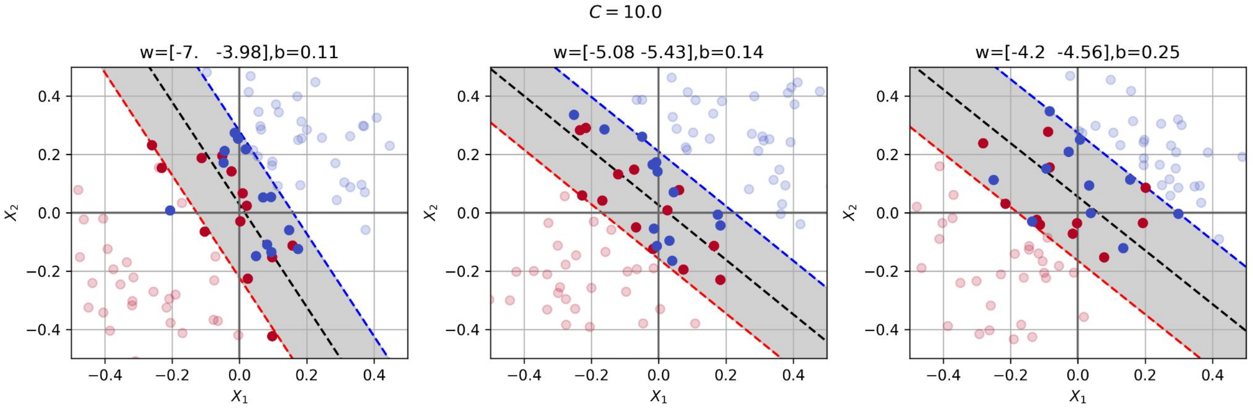
UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

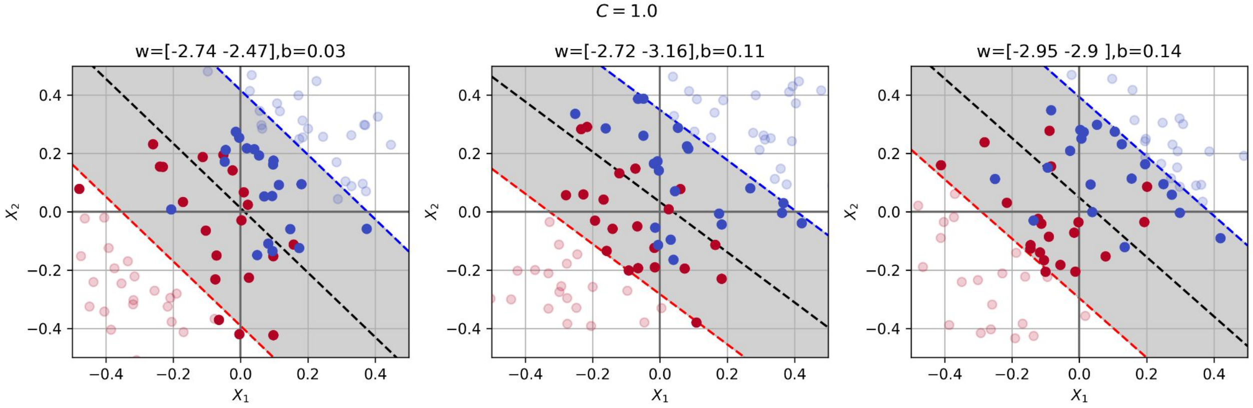
The bias-variance tradeoff in KNN



Introduction to Statistical Learning, Section 2.2

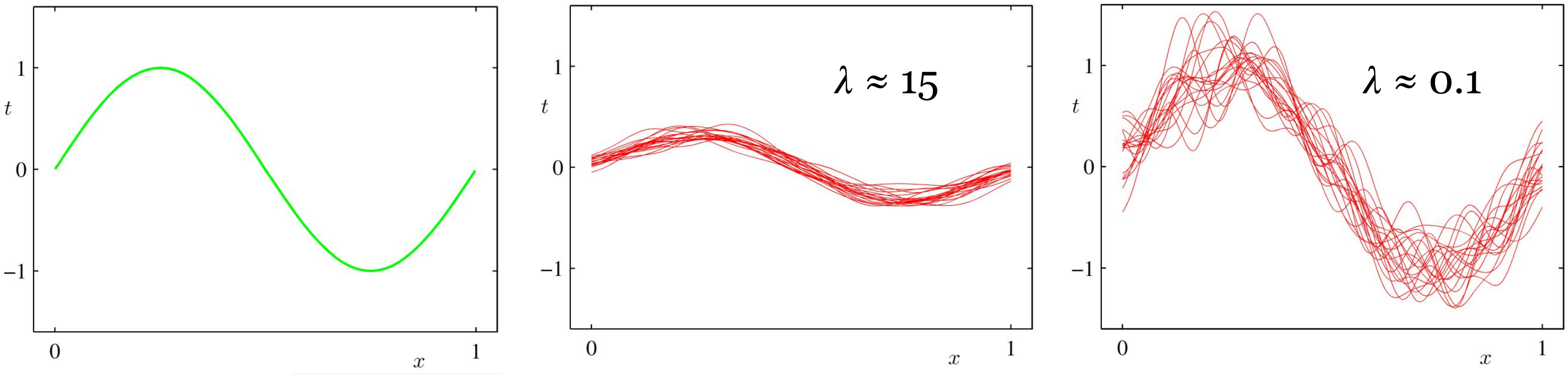
The bias-variance tradeoff in SVMs





The bias-variance tradeoff in linear regression

True function



Pattern Recognition and Machine Learning, Section 3.2

Suppose we are tasked with predicting random variable $Y \in \mathbb{R}$ which is a function of $X \in \mathbb{R}^{\mathrm{d}}$ corrupted by Gaussian noise ϵ:

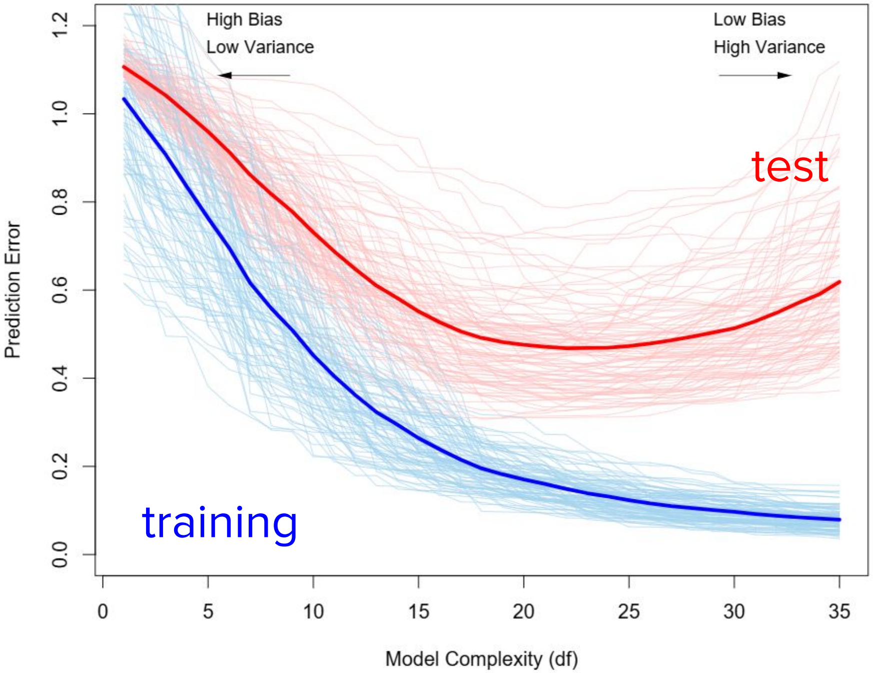
$$Y = f(X) + \epsilon$$
$$\epsilon \sim \mathcal{N}(\mu = 0, \sigma_\epsilon^2)$$
$$f : \ \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$$

Expectation of the squared error for some new test input x_t is:

$$\mathrm{Err}(x_t) = \mathbb{E}\left[(Y - \hat{f}(x_t))^2 \mid X = x_t\right]$$
$$= \sigma_\epsilon^2 + \underbrace{\left[\mathbb{E}[\hat{f}(x_t)] - f(x_t)\right]^2}_{\text{Bias}^2} + \underbrace{\mathbb{E}\left[\left(\hat{f}(x_t) - \mathbb{E}[\hat{f}(x_t)]\right)^2\right]}_{\text{Variance}}$$

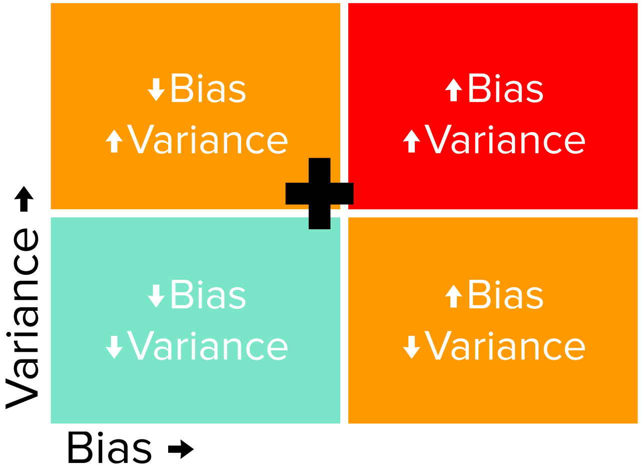
$\hat{f}(x_t)$: Prediction for one approximated function

$\mathbb{E}[\hat{f}(x_t)]$: Average over predictions



Elements of Statistical Learning, Section 7.2-7.3

Strategy so far: Select a *single* model to optimize BV tradeoff



↓Bias
↑Variance

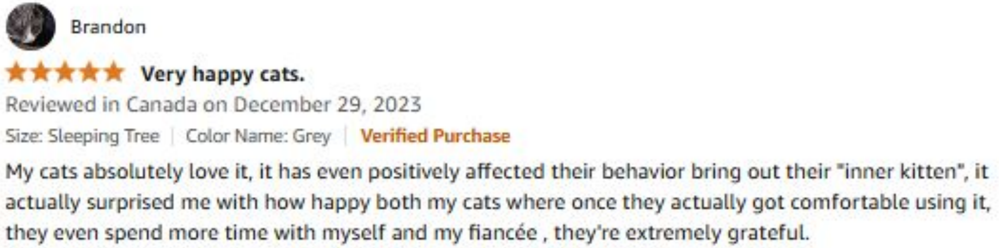
↑Bias
↑Variance

↓Bias
↓Variance

↑Bias
↓Variance

Is this really the best we can do?

This strategy is at odds with our everyday decision-making



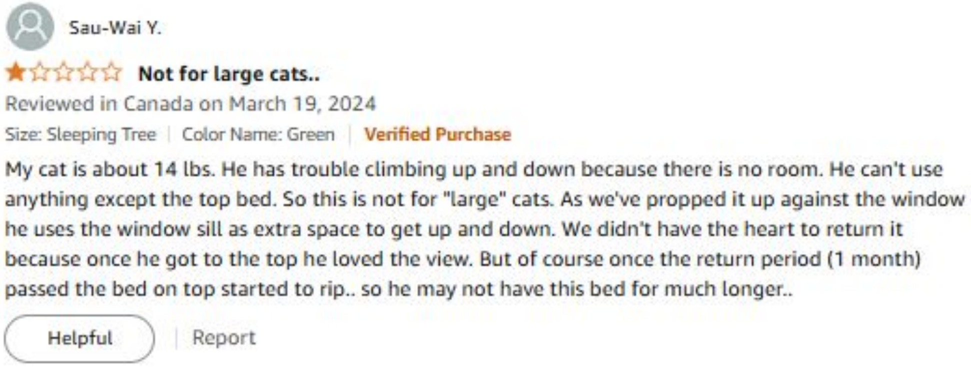
Brandon

★★★★★ **Very happy cats.**

Reviewed in Canada on December 29, 2023

Size: Sleeping Tree | Color Name: Grey | Verified Purchase

My cats absolutely love it, it has even positively affected their behavior bring out their "inner kitten", it actually surprised me with how happy both my cats where once they actually got comfortable using it, they even spend more time with myself and my fiancée , they're extremely grateful.



Sau-Wai Y.

★☆☆☆☆ **Not for large cats..**

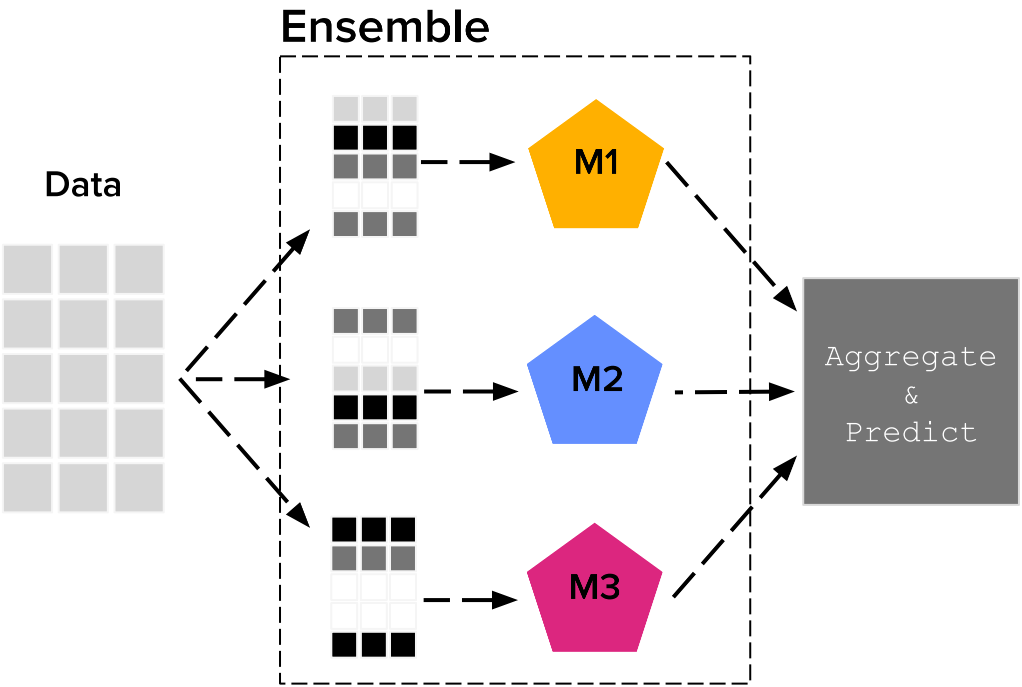
Reviewed in Canada on March 19, 2024

Size: Sleeping Tree | Color Name: Green | Verified Purchase

My cat is about 14 lbs. He has trouble climbing up and down because there is no room. He can't use anything except the top bed. So this is not for "large" cats. As we've propped it up against the window he uses the window sill as extra space to get up and down. We didn't have the heart to return it because once he got to the top he loved the view. But of course once the return period (1 month) passed the bed on top started to rip.. so he may not have this bed for much longer..

This strategy is at odds with our everyday decision-making

Ensemble methods leverage an assembly of base models



Data

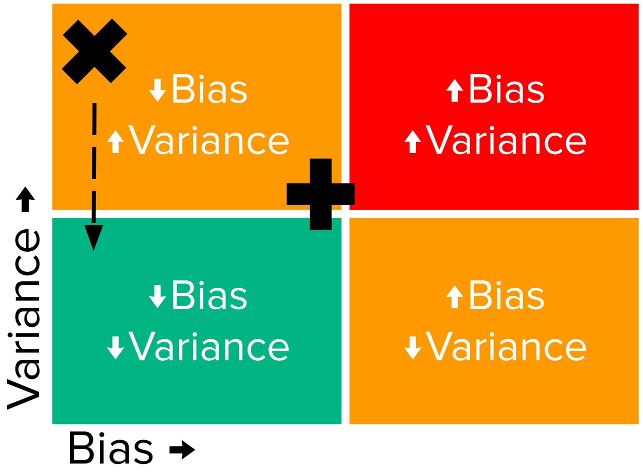
Ensemble

M1

M2

M3

Aggregate & Predict



↓Bias ↑Variance

↑Bias ↑Variance

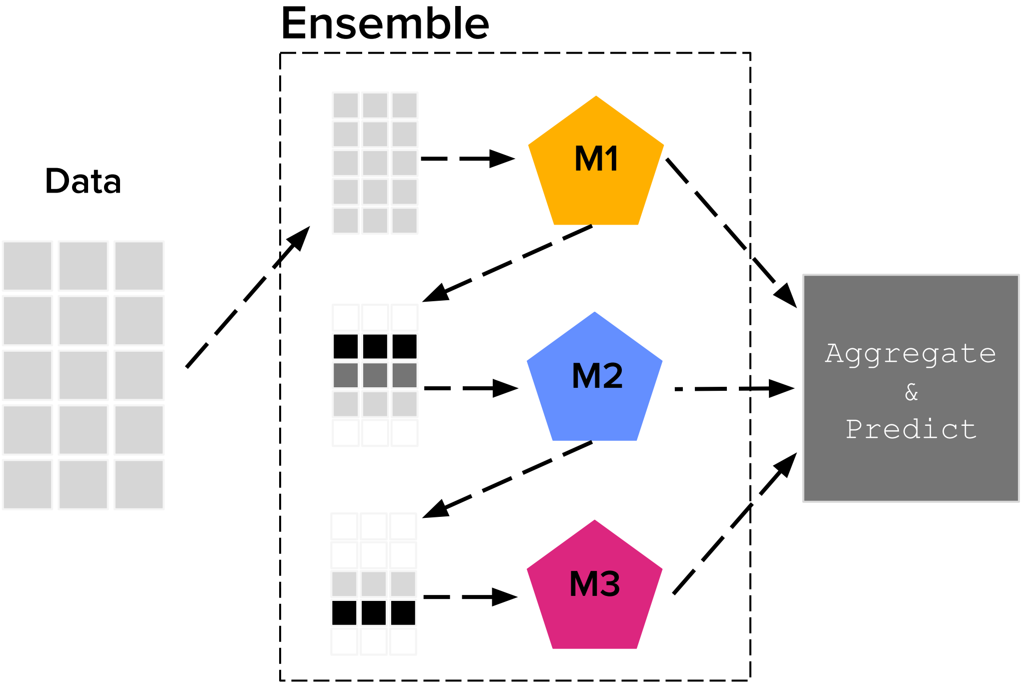
↓Bias ↓Variance

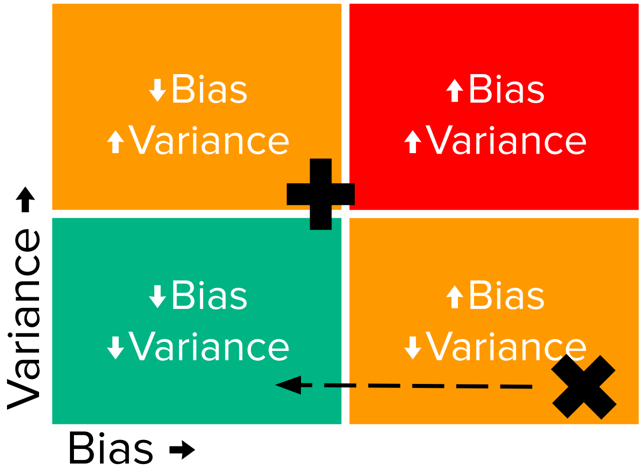
↑Bias ↓Variance

Variance ↑

Bias →

Ensembles can also be trained sequentially





Key Questions

I. **How can ensembles reduce *variance*?**

II. **How can ensembles reduce *bias*?**

III. **How do these methods compare?**

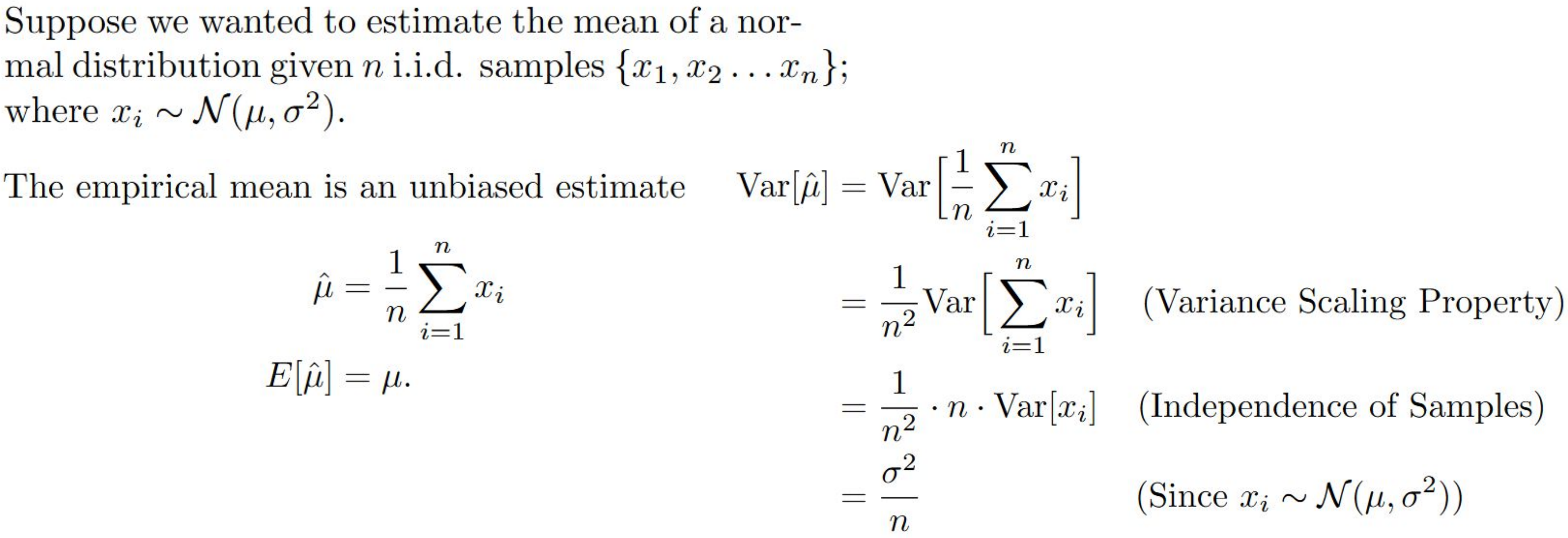
Key Questions

I. **How can ensembles reduce *variance*?**

II. **How can ensembles reduce *bias*?**

III. **How do these methods compare?**

The variance of an estimator depends on dataset size



Suppose we wanted to estimate the mean of a normal distribution given n i.i.d. samples $\{x_1, x_2 \dots x_n\}$; where $x_i \sim \mathcal{N}(\mu, \sigma^2)$.

The empirical mean is an unbiased estimate

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$E[\hat{\mu}] = \mu.$$

$$\begin{aligned}
\mathrm{Var}[\hat{\mu}] &= \mathrm{Var}\left[\frac{1}{n}\sum_{i=1}^{n} x_i\right] \\
&= \frac{1}{n^2}\mathrm{Var}\left[\sum_{i=1}^{n} x_i\right] && \text{(Variance Scaling Property)} \\
&= \frac{1}{n^2} \cdot n \cdot \mathrm{Var}[x_i] && \text{(Independence of Samples)} \\
&= \frac{\sigma^2}{n} && \text{(Since } x_i \sim \mathcal{N}(\mu, \sigma^2))
\end{aligned}$$

If we had more data we could reduce the variance ***without*** increasing bias

If we had Bn points, we could form

$$
\begin{aligned}
S_1 &= \{x_1, \ldots x_n\} \\
S_2 &= \{x_{n+1}, \ldots x_{2n}\} \\
&\vdots \\
S_B &= \{x_{(B-1)n+1}, \ldots x_{Bn}\}
\end{aligned}
$$

$$\hat{\mu}_b = \sum_{(b-1)n+1}^{bn} x_i \qquad \hat{\mu}_B = \frac{1}{B}\sum_b \hat{\mu}_b$$

$$E[\hat{\mu}_B] = \mu$$

$$
\begin{aligned}
\operatorname{Var}[\hat{\mu}_B] &= \operatorname{Var}\Big[\frac{1}{B}\sum_b \hat{\mu}_b\Big] \\
&= \frac{1}{B^2}\operatorname{Var}\Big[\sum_b \hat{\mu}_b\Big] \\
&= \frac{1}{B^2} \cdot B \cdot \operatorname{Var}\big[\hat{\mu}_b\big] \\
&= \frac{1}{B^2} \cdot B \cdot \frac{\sigma^2}{n} \\
&= \frac{\sigma^2}{Bn}
\end{aligned}
$$

Is there anything we can do?

Consider the datasets

$$S_1 = \{x_1 = 0.4, x_2 = 1.0, x_3 = 1.6\}$$
$$S_2 = \{x_4 = 0.5, x_5 = 0.7, x_6 = 1.2\}$$

$$\hat{\mu}_1 = \frac{1}{3}(\mathbf{0.4} + \mathbf{1.0} + 1.6) = \frac{1}{3}(3.0) = 1.0$$

$$\hat{\mu}_2 = \frac{1}{3}(0.5 + 0.7 + 1.2) = \frac{1}{3}(2.4) = 0.8$$

$$\hat{\mu}_2 = \frac{1}{3}(\mathbf{0.4} + \mathbf{1.0} + \mathbf{1.0}) = \frac{1}{3}(2.4) = 0.8$$

Bootstrapping: Mimic the *variability* of drawing more samples from the population by resampling the original data (with replacement)

Bootstrap Sampling: Sample with replacement

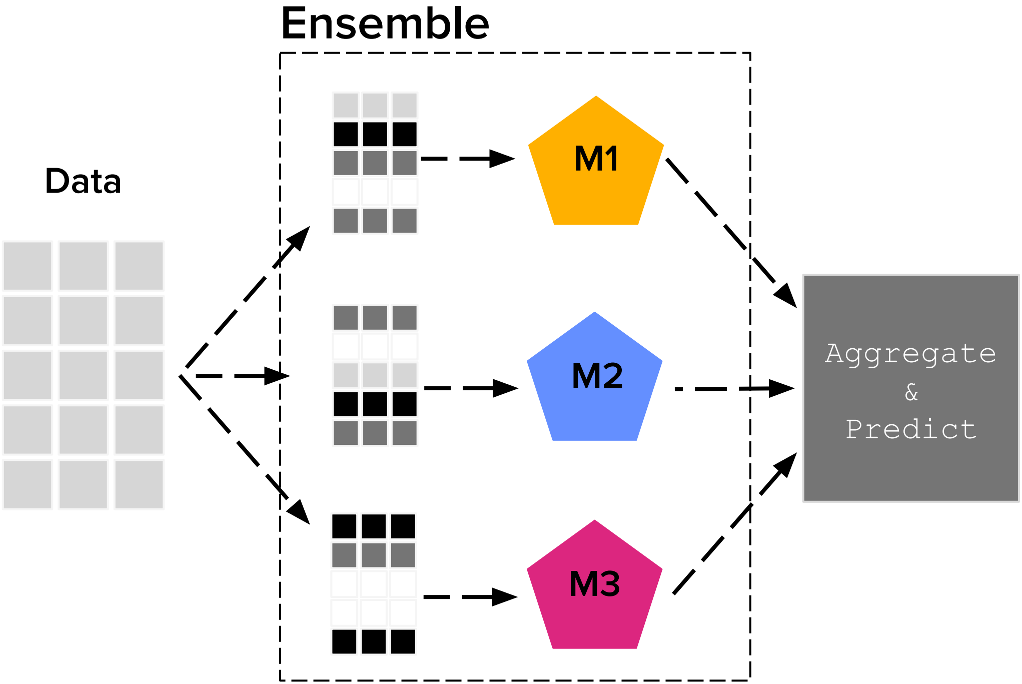
Original Dataset:

$$\{X_1, X_2, X_3, X_4, X_5\}$$

Bootstrap Realizations:

$$\{X_1, X_1, X_4, X_5, X_3\}$$
$$\{X_5, X_2, X_3, X_5, X_1\}$$
$$\{X_3, X_5, X_3, X_2, X_1\}$$

Bagging (Bootstrap Aggregation)



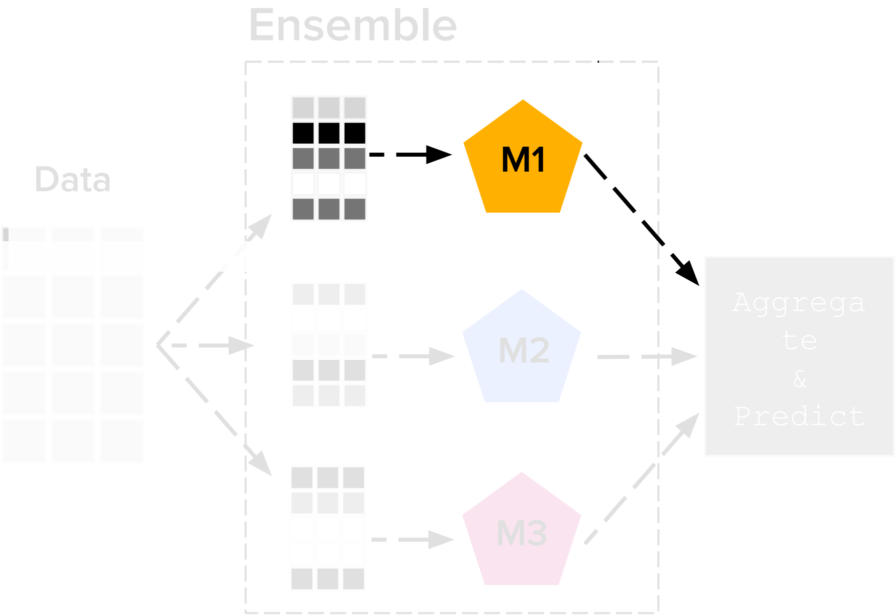
For regression:

$$\hat{y}_E(x_i) = \frac{1}{E}\sum_{e=1}^{E} y_e(x_i)$$

For classification:

$$\hat{y}_E(x_i) = \begin{cases} +1 & \sum_{e=1}^{E} y_e(x_i) > 0 \\ -1 & \text{otherwise} \end{cases}$$

Analyzing the effect of bagging on regression error



$$t(x) = y_e(x) + \epsilon_e(x)$$

$$\implies \epsilon_e(x) = t(x) - y_e(x)$$

Single Bootstrap Model:

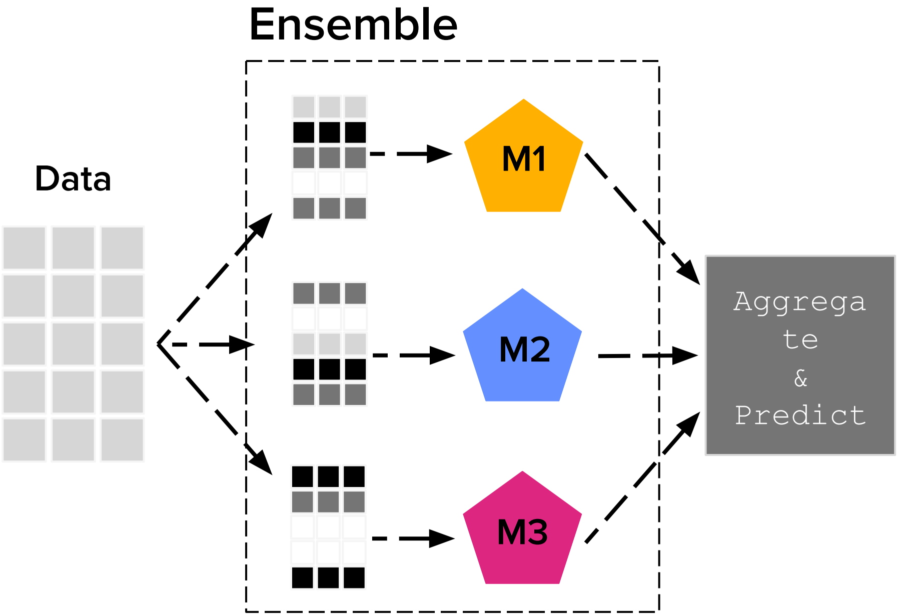
$$\mathbb{E}\left[\left(t(x) - y_e(x)\right)^2\right] = \mathbb{E}\left[\epsilon_e(x)^2\right]$$

$$\epsilon_{\text{avg}} = \frac{1}{E}\sum_{e=1}^{E}\mathbb{E}\left[\epsilon_e(x)^2\right]$$

$$= \mathbb{E}\left[\epsilon(x)^2\right]$$

Pattern Recognition and Machine Learning, Section 14.2

Analyzing the effect of bagging on regression error



$$\hat{y}_E(x_i) = \frac{1}{E}\sum_{e=1}^{E} y_e(x_i)$$

Analyzing the effect of bagging on regression error

$$
\begin{aligned}
\mathbb{E}\left[(t(x) - y_E(x))^2\right] &= \mathbb{E}\left[\left(t(x) - \frac{1}{E}\sum_{e=1}^{E} y_e(x)\right)^2\right] \\
&= \mathbb{E}\left[\left(t(x) - \frac{1}{E}\sum_{e=1}^{E}(t(x) - \epsilon_e(x))\right)^2\right] \\
&= \mathbb{E}\left[\left(\frac{1}{E}\sum_{e=1}^{E} \epsilon_e(x)\right)^2\right] \\
&= \mathbb{E}\left[\frac{1}{E^2}\sum_{i=1}^{E}\sum_{j=1}^{E} \epsilon_i(x)\epsilon_j(x)\right] \\
&= \frac{1}{E^2}\sum_{i=1}^{E}\sum_{j=1}^{E} \mathbb{E}\left[\epsilon_i(x)\epsilon_j(x)\right]
\end{aligned}
$$

Analyzing the effect of bagging on regression error

$$\frac{1}{E^2}\sum_{i=1}^{E}\sum_{j=1}^{E}\mathbb{E}\left[\epsilon_i(x)\varepsilon_j(x)\right] = \frac{1}{E^2}\sum_{i=1}^{E}\mathbb{E}\left[\epsilon_i(x)^2\right] + \frac{1}{E^2}\sum_{i\neq j}\mathbb{E}\left[\epsilon_i(x)\epsilon_j(x)\right]$$

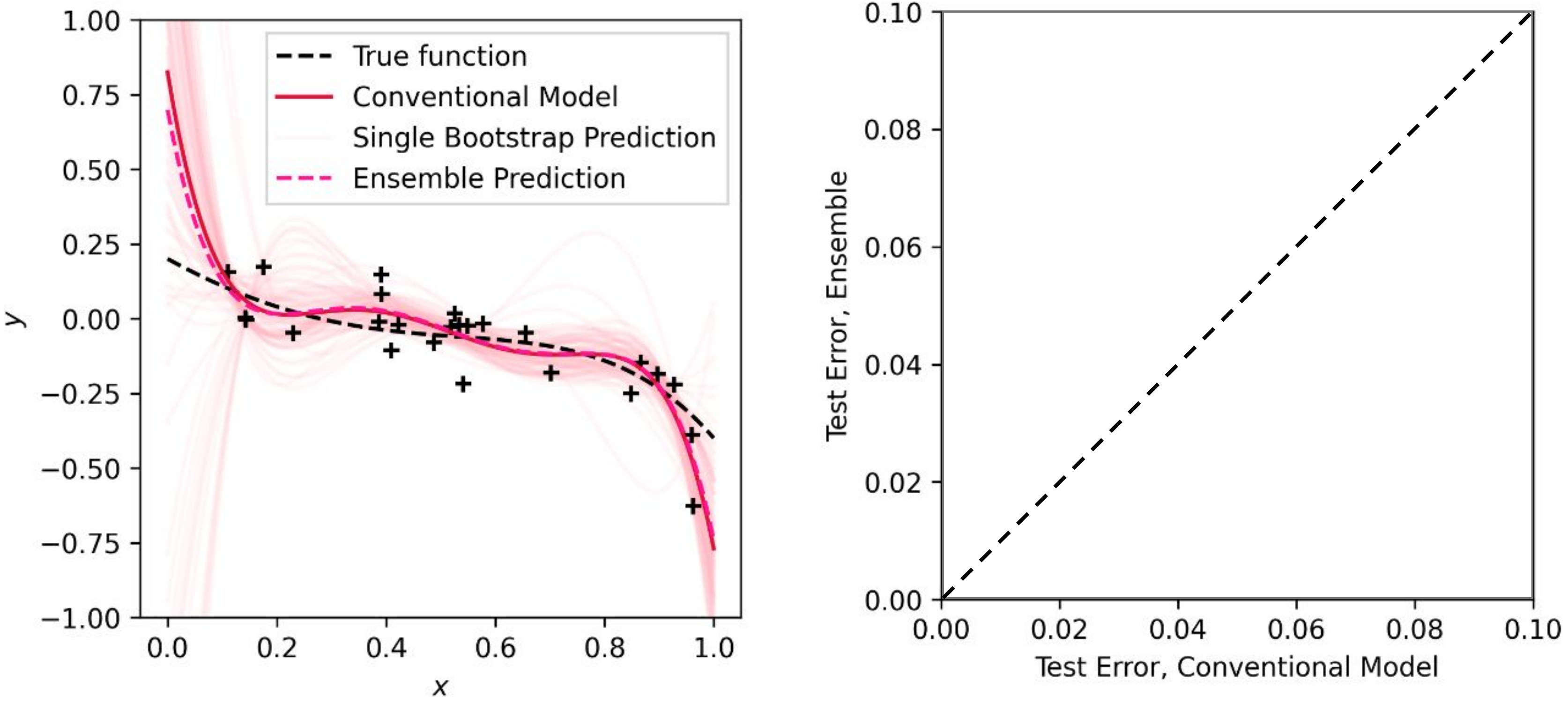
Assume: $\mathbb{E}\left[\epsilon_i(x)\right] = 0 \quad \mathbb{E}\left[\epsilon_i(x)\epsilon_j(x)\right] = 0$

$$= \frac{1}{E^2}\sum_{i=1}^{E}\mathbb{E}\left[\epsilon_i(x)^2\right] + 0$$

Ensemble: $= \frac{1}{E}\mathbb{E}\left[\epsilon(x)^2\right]$

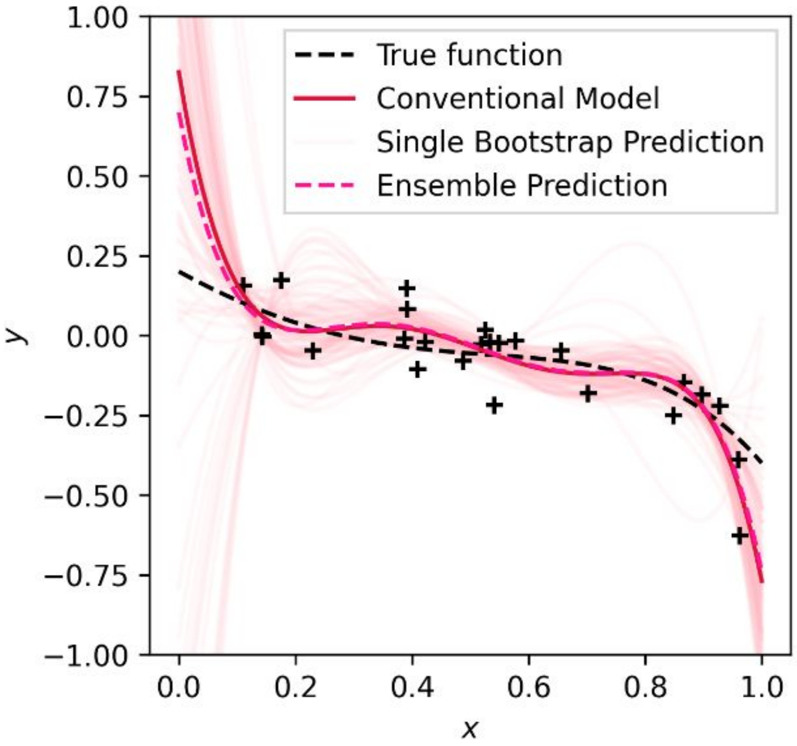
One bootstrap model: $= \mathbb{E}\left[\varepsilon(x)^2\right]$

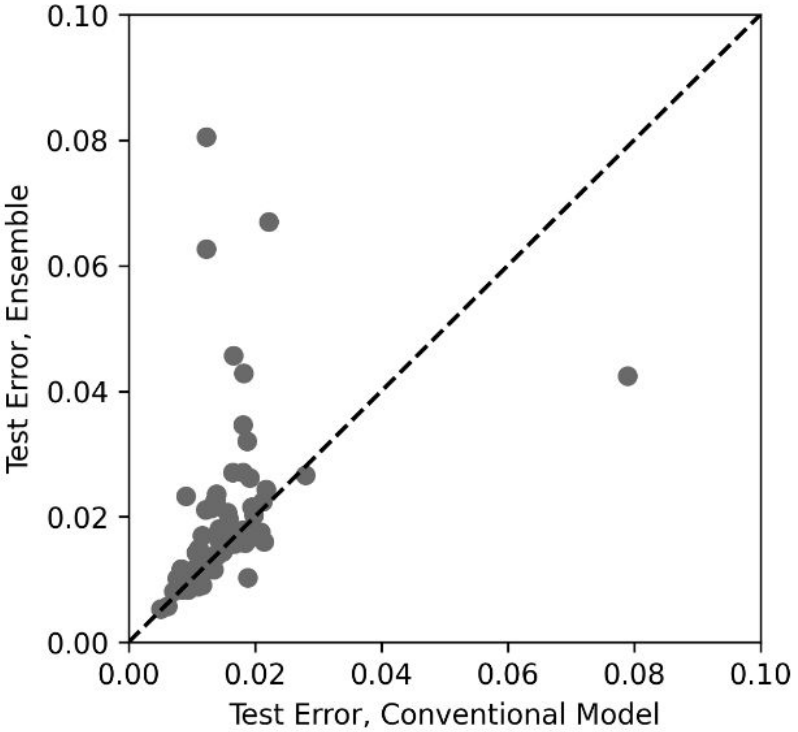
Linear models are not good candidates for bagging



Elements of Statistical Learning 15.2

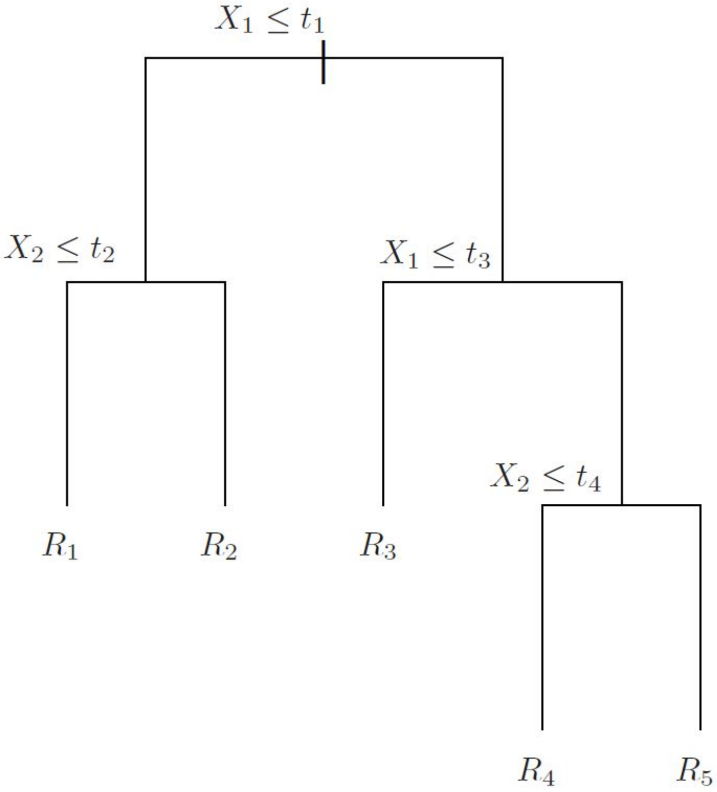
Linear models are not good candidates for bagging



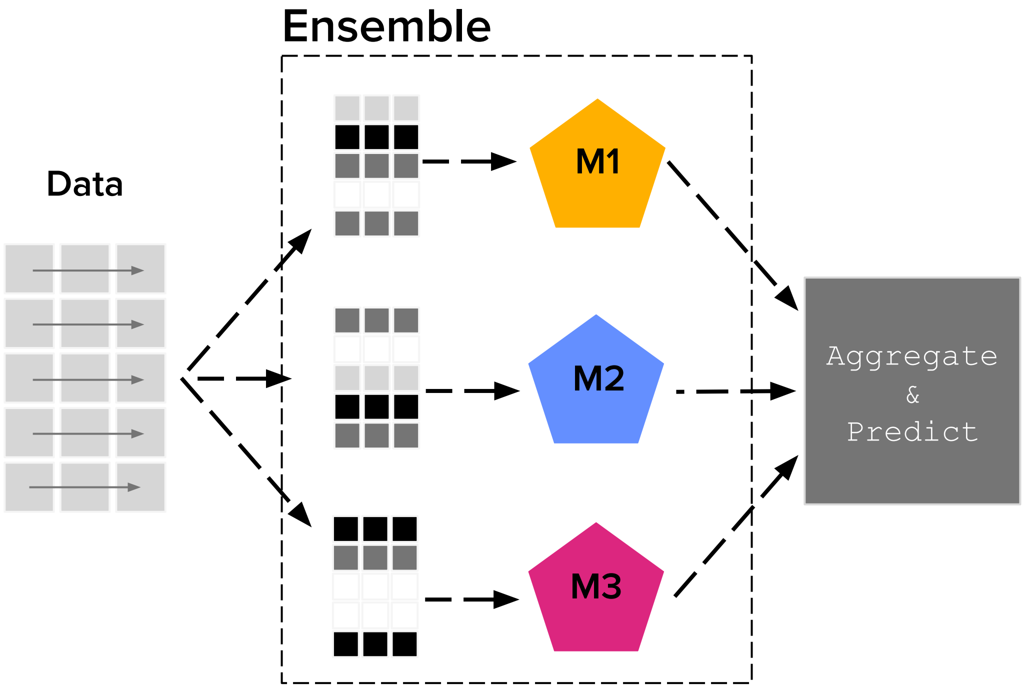


Elements of Statistical Learning 15.2

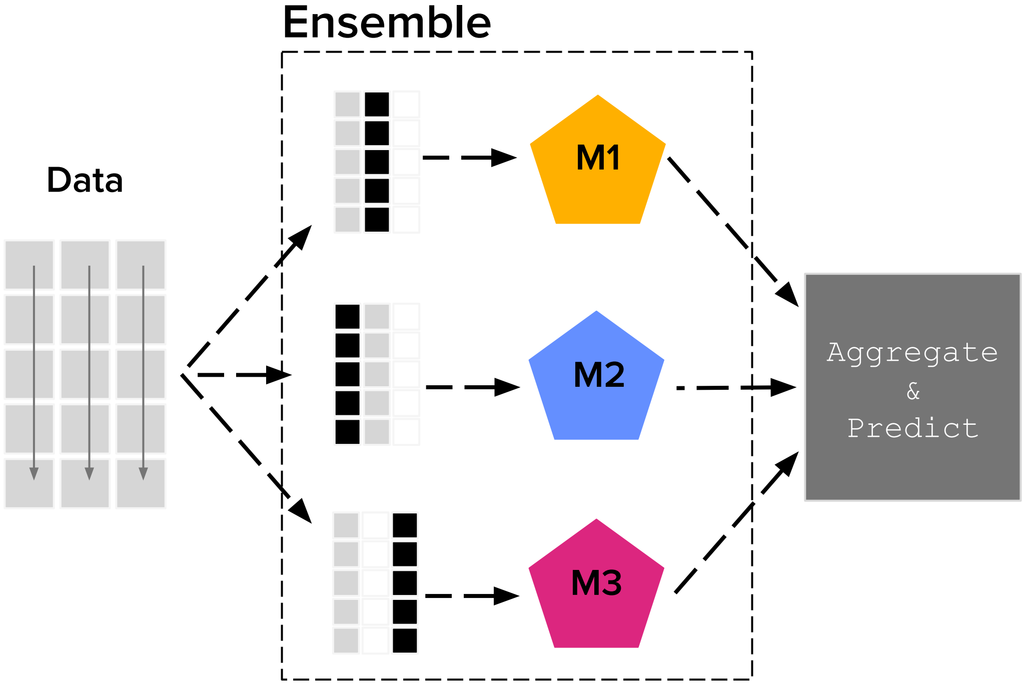
Decision trees are good candidates for bagging



Injecting variability by resampling *observations*

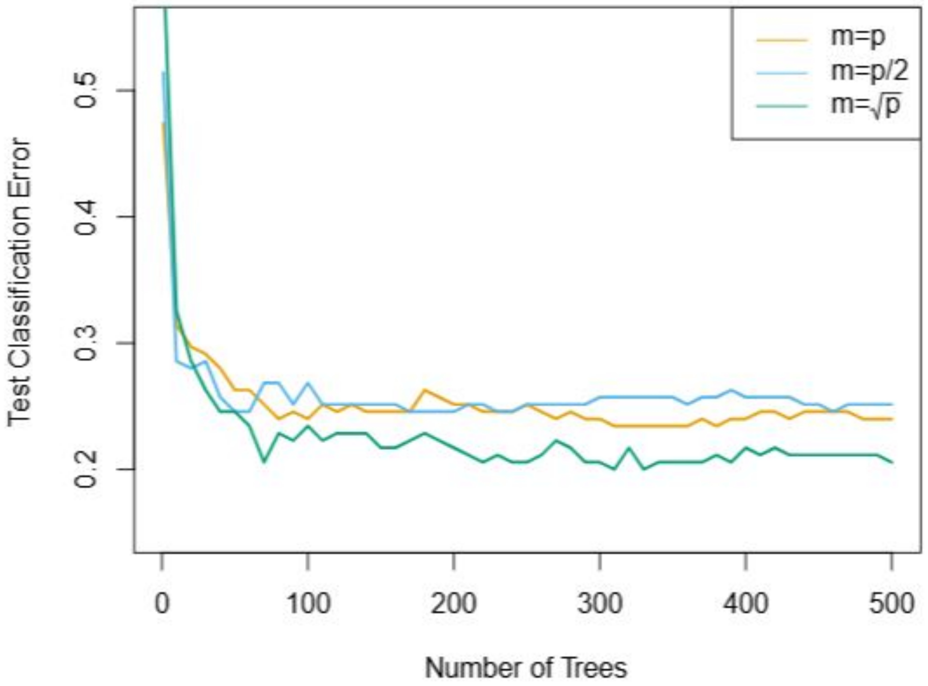


Injecting variability by resampling *features*



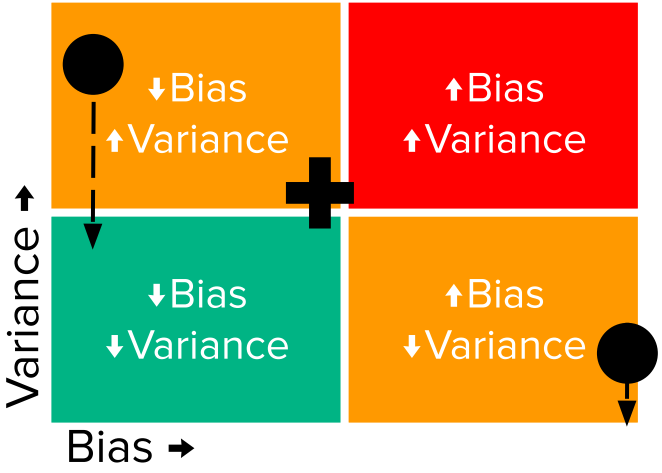
Random Forests

- Reduce correlation between trees by providing a random subset of input features
- Before each split, select m < p input features at random as candidates for splitting



Introduction to Statistical Learning 8.2

Bagging targets variance, has little effect on bias



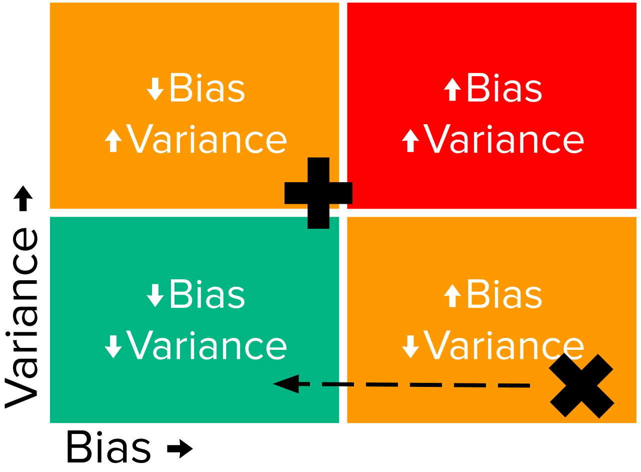
↓Bias
↑Variance

↑Bias
↑Variance

↓Bias
↓Variance

↑Bias
↓Variance

Ensembles can also be trained sequentially



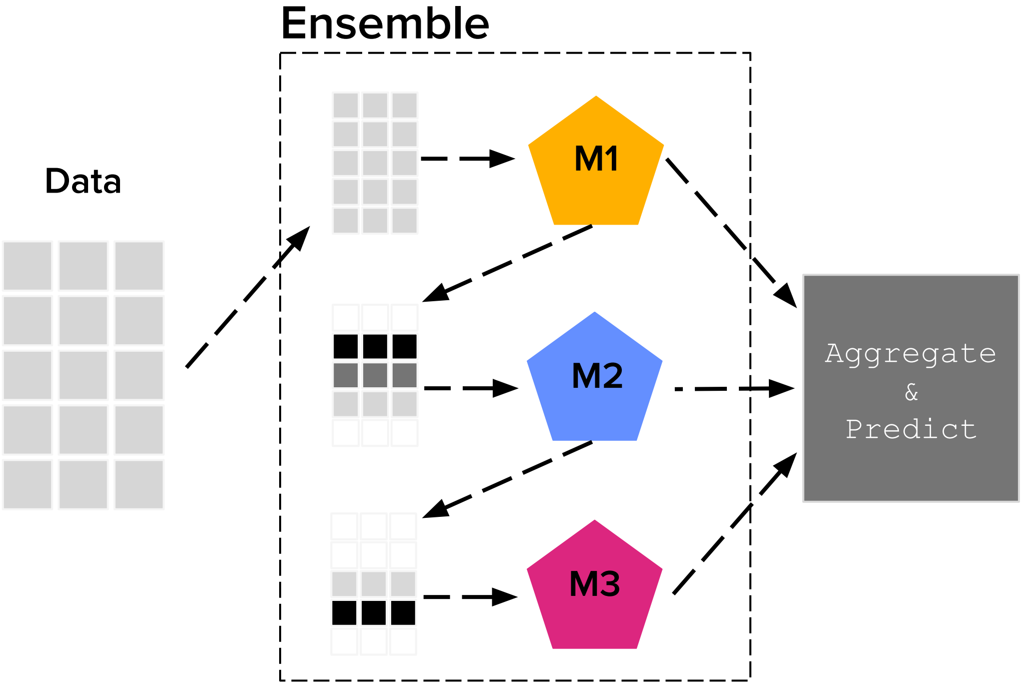
Key Questions

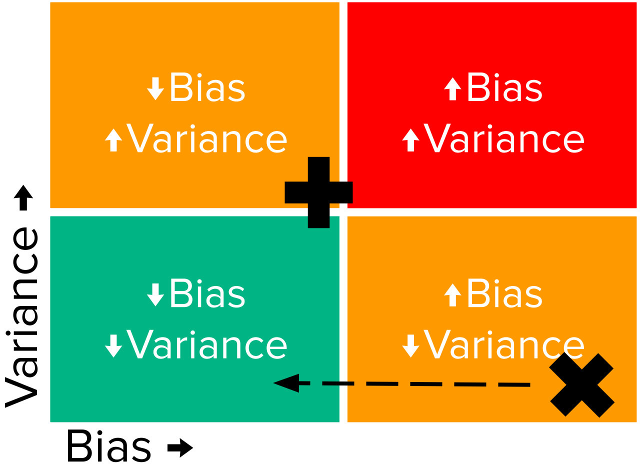
I. How can ensembles reduce *variance*?

II. **How can ensembles reduce *bias*?**

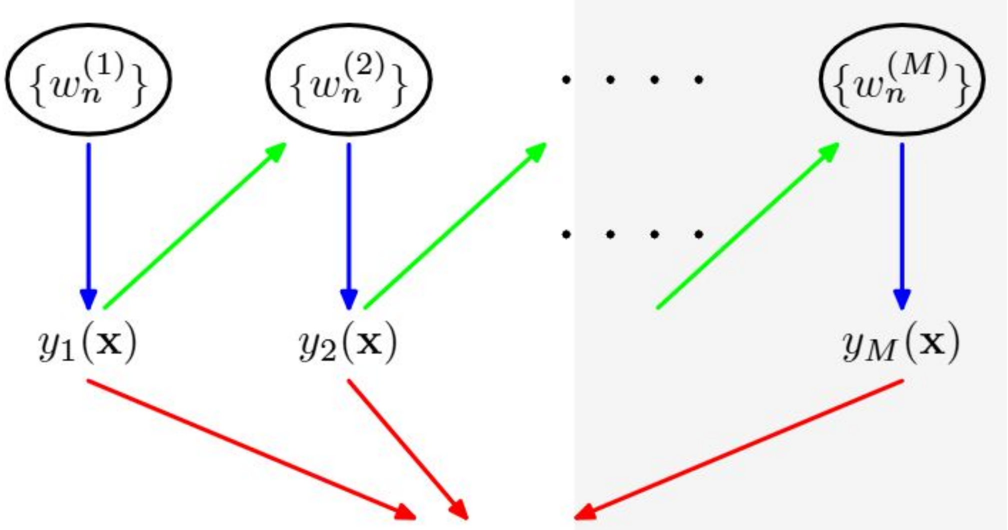
III. When should we use one approach or the other?

Ensembles can also be trained sequentially





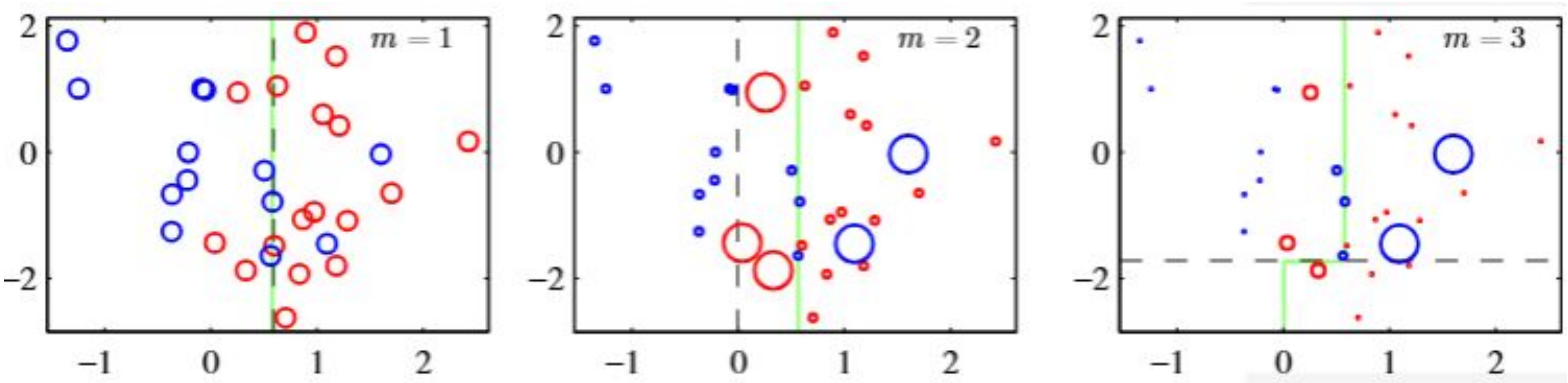
Boosting turns a weak model into a strong model



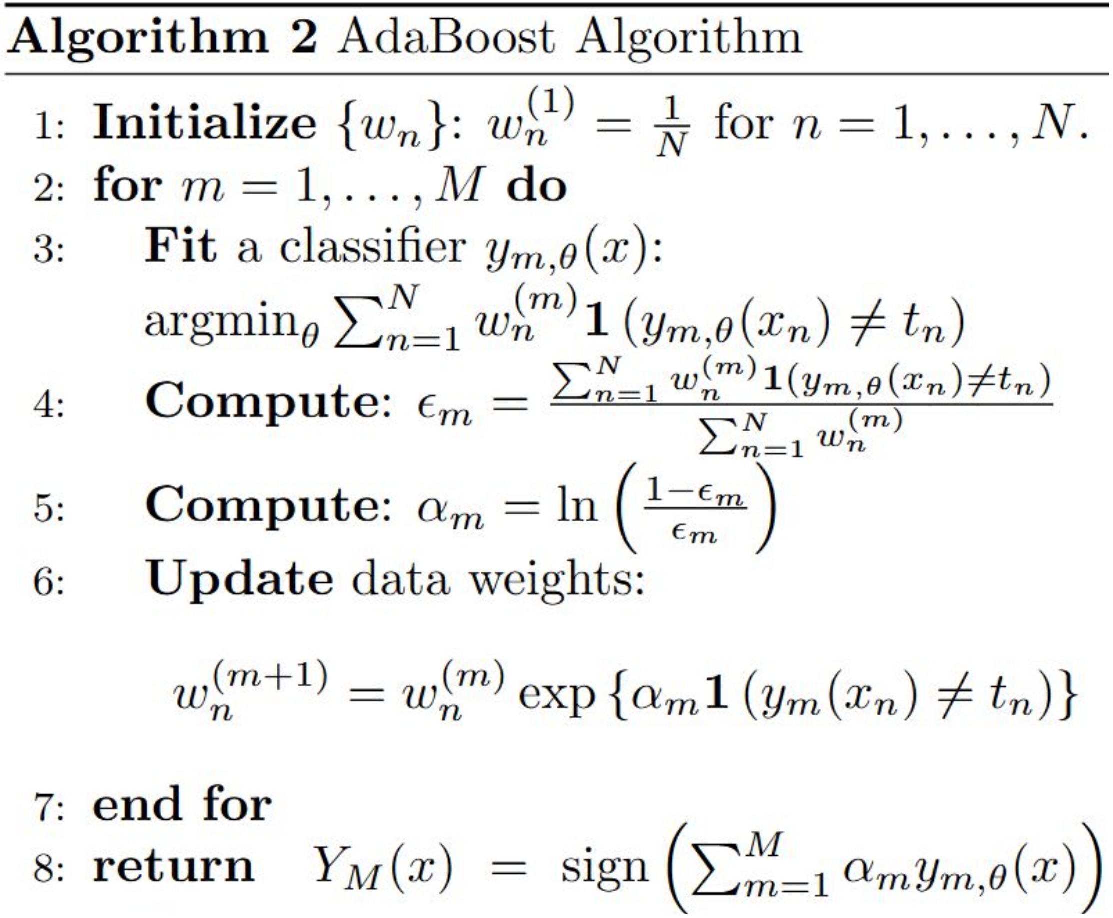
$$Y_M(\mathbf{x}) = \operatorname{sign}\left(\sum_m^M \alpha_m y_m(\mathbf{x})\right)$$

Pattern Recognition and Machine Learning, Section 14.2

AdaBoost (Adaptive Boosting)



Pattern Recognition and Machine Learning, Section 14.2



Algorithm 2 AdaBoost Algorithm

```
Initialize $\{w_n\}$: $w_n^{(1)} = \frac{1}{N}$ for $n = 1, \ldots, N$.
for $m = 1, \ldots, M$ do
    Fit a classifier $y_{m,\theta}(x)$:
    $\text{argmin}_\theta \sum_{n=1}^{N} w_n^{(m)} \mathbf{1}\left(y_{m,\theta}(x_n) \neq t_n\right)$
    Compute: $\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} \mathbf{1}(y_{m,\theta}(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$
    Compute: $\alpha_m = \ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$
    Update data weights:

        $w_n^{(m+1)} = w_n^{(m)} \exp\left\{\alpha_m \mathbf{1}\left(y_m(x_n) \neq t_n\right)\right\}$

end for
return  $Y_M(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_{m,\theta}(x)\right)$
```

Pattern Recognition and Machine Learning, Section 14.2

Key Questions

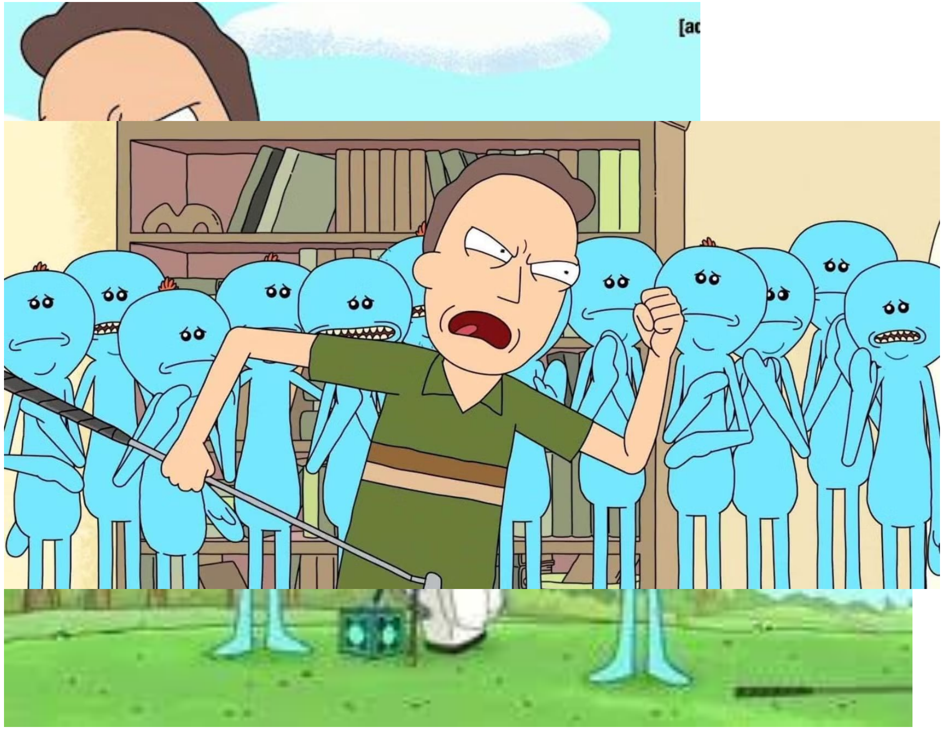
I. How can ensembles reduce *variance*?

II. How can ensembles reduce *bias*?

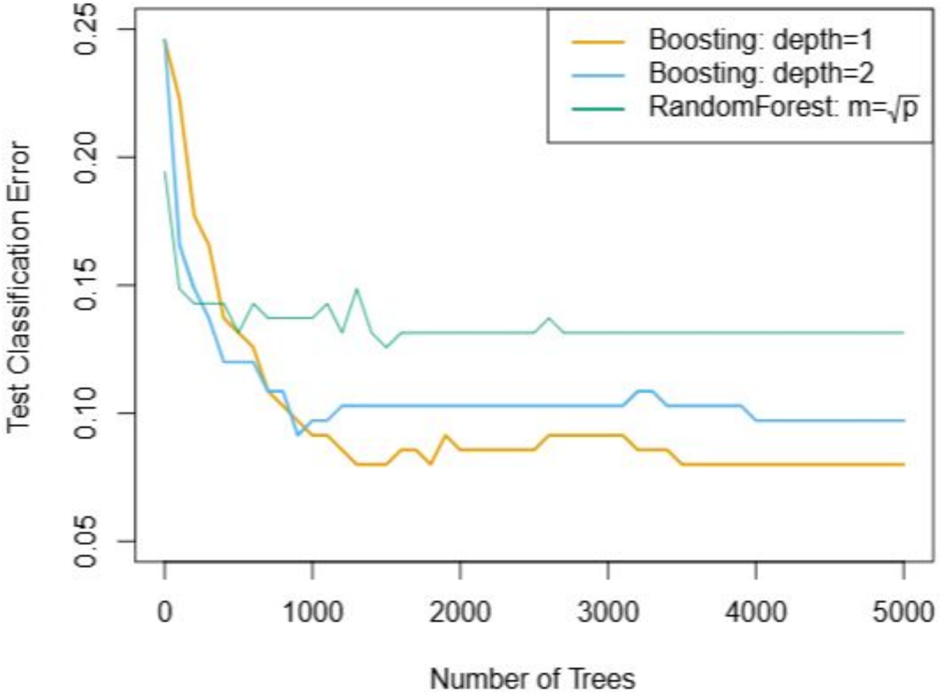
III. **How do these methods compare?**

Bagging

Boosting



Comparing Boosting and Bagging

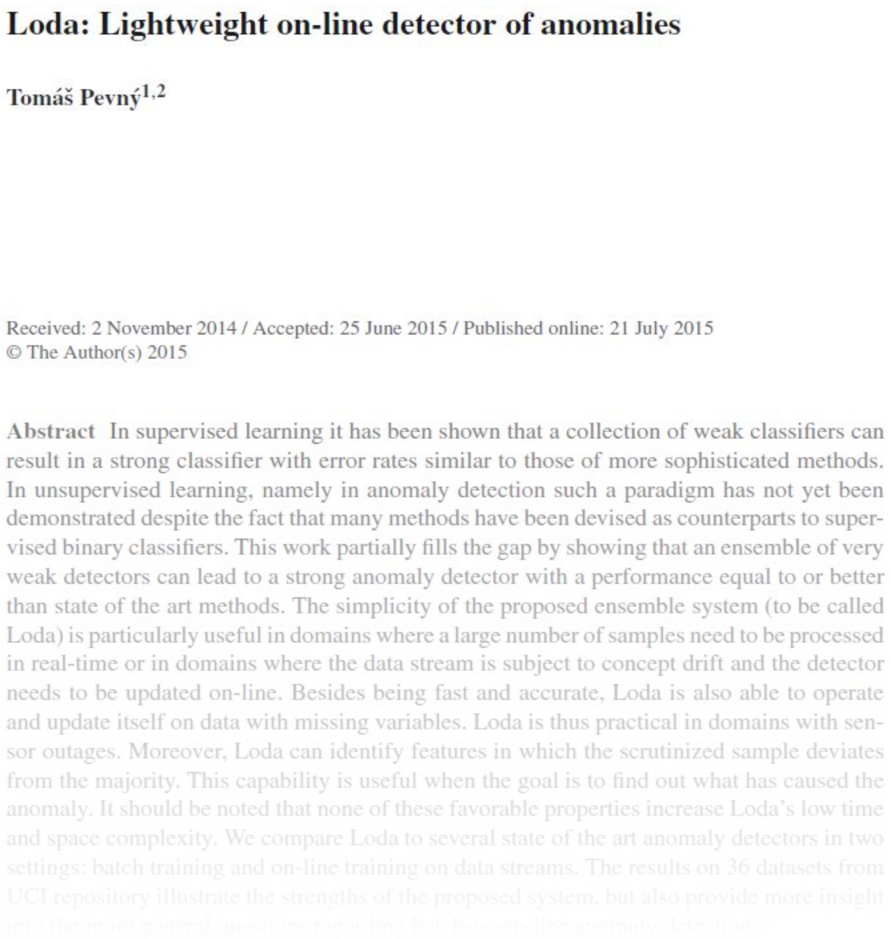


Introduction to Statistical Learning, Section 2.2

Now that we’re at the end of the lecture, you should be able to…

- ★ Explain the **bias-variance decomposition** and demonstrate how it impacts model generalization.
- ★ Implement **bagging techniques** and analyze how aggregating models reduces variance without increasing bias.
- ★ Implement **boosting methods** and illustrate how they sequentially reduce bias by focusing on prior errors.
- ★ Compare bagging and boosting strategies and **recommend appropriate use cases**.
- ★ Develop **random forests** by selecting random subsets of features to **decorrelate decision trees** and reduce overfitting.
- ★ Evaluate the impact of ensemble size on **test error reduction.**
- ★ **Implement ensemble algorithms** for regression and classification tasks and assess their performance on various datasets.

Ensemble methods in unsupervised learning



Loda: Lightweight on-line detector of anomalies

Tomáš Pevný[1,2]

Received: 2 November 2014 / Accepted: 25 June 2015 / Published online: 21 July 2015

Abstract In supervised learning it has been shown that a collection of weak classifiers can result in a strong classifier with error rates similar to those of more sophisticated methods. In unsupervised learning, namely in anomaly detection such a paradigm has not yet been demonstrated despite the fact that many methods have been devised as counterparts to supervised binary classifiers. This work partially fills the gap by showing that an ensemble of very weak detectors can lead to a strong anomaly detector with a performance equal to or better than state of the art methods. The simplicity of the proposed ensemble system (to be called Loda) is particularly useful in domains where a large number of samples need to be processed in real-time or in domains where the data stream is subject to concept drift and the detector needs to be updated on-line. Besides being fast and accurate, Loda is also able to operate and update itself on data with missing variables. Loda is thus practical in domains with sensor outages. Moreover, Loda can identify features in which the scrutinized sample deviates from the majority. This capability is useful when the goal is to find out what has caused the anomaly. It should be noted that none of these favorable properties increase Loda's low time and space complexity. We compare Loda to several state of the art anomaly detectors in two settings: batch training and on-line training on data streams. The results on 36 datasets from UCI repository illustrate the strengths of the proposed system, but also provide more insight into the more general questions regarding batch-vs-on-line anomaly detection.

Errata

- On slide 15, the example datasets S_1 and S_2 used to motivate bootstrap sampling were inconsistent with the two calculations of $\hat{\mu}_2$. These datasets S_1 and S_2 have been fixed to be consistent with the motivating example.
- On slide 17, a previous version of the slide indicated that an ensemble of classifiers would predict the positive class if the sum over base classifier predictions was greater than $E/2$, where E is the number of base classifiers in the ensemble. This has been corrected to specify that the ensemble would predict the positive class if the sum over base classifier predictions is greater than 0.
- On slide 34, the notation used in the algorithm block alternated in the use of m and e to denote properties of one of the base classifiers (weights, error rate, prediction weighting, etc). All such instances have been corrected to use the subscript or superscript m.
- The algorithm block on slide 34 also included a "$= 0$" on the return line. This was due to a LaTeX typesetting error and has been removed.