Bagging and Random Forests

CS 584 Data Mining (Spring 2022)

Prof. Sanmay Das

George Mason University

Slides are adapted from the available book slides developed by Tan, Steinbach, Karpatne, and Kumar

The Bias-Variance Decomposition

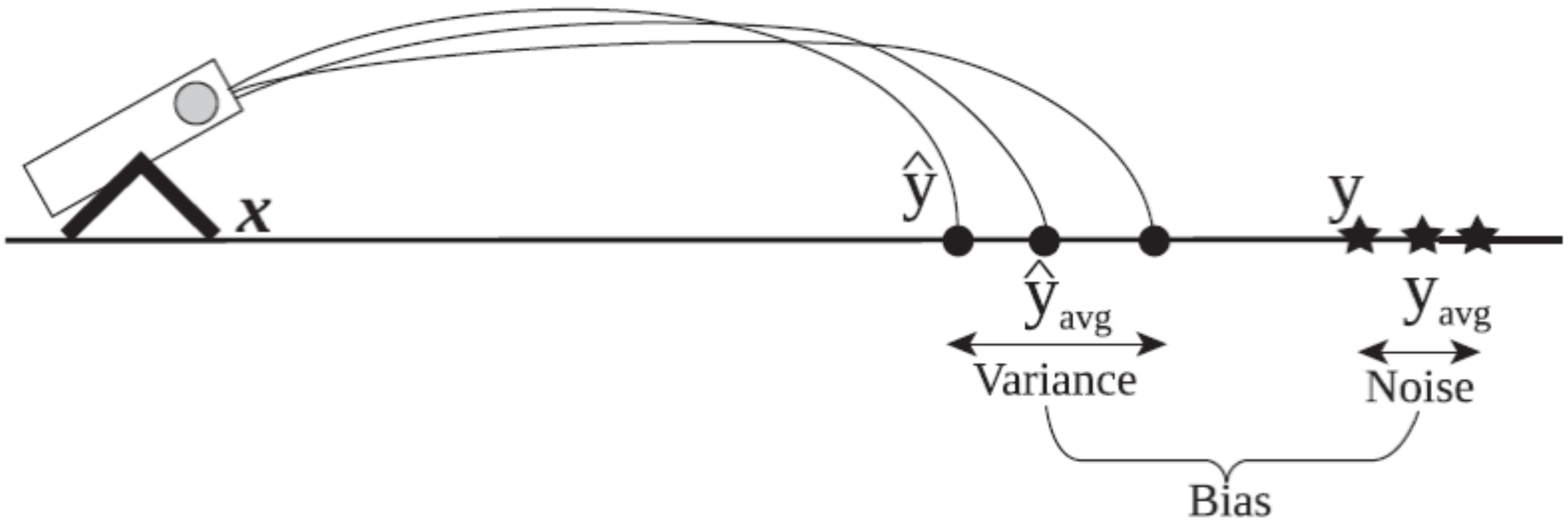
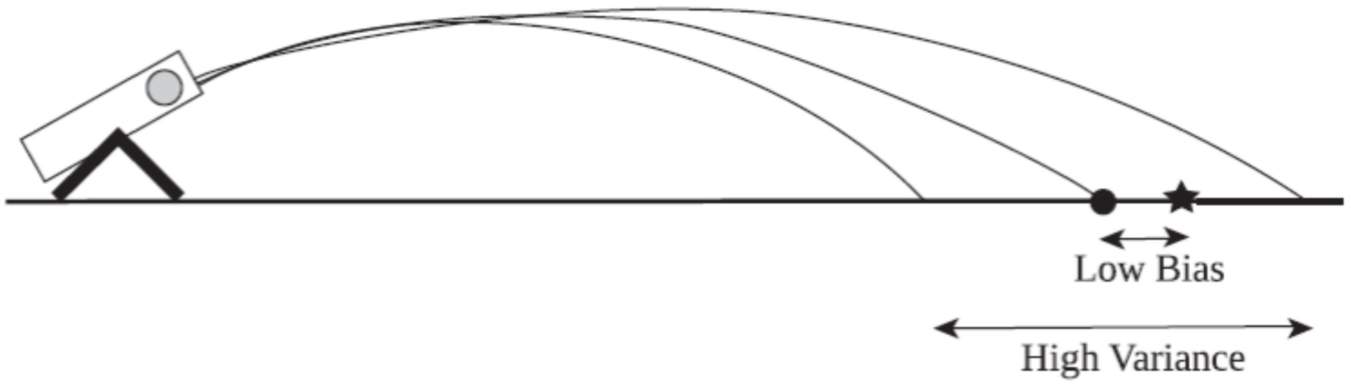
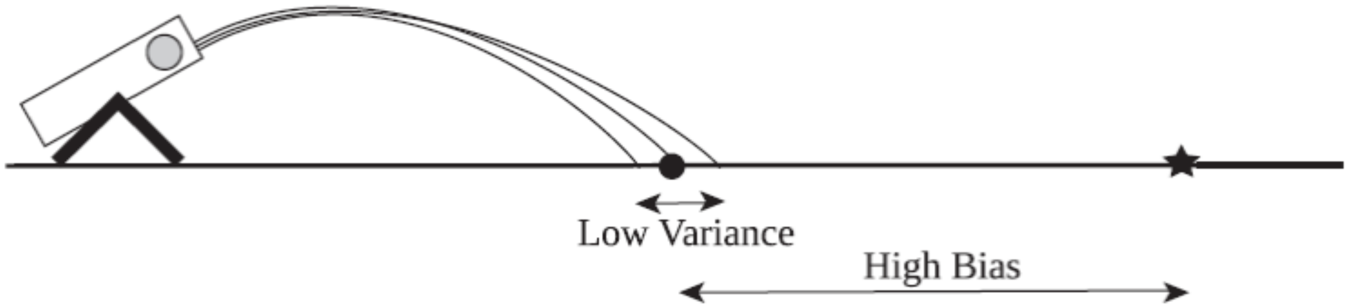


Figure 4.44. Bias-variance decomposition.

gen.error(m) = $c1$ × noise + bias(m) + $c2$ × variance(m)



(a) Phenomena of Overfitting.



(b) Phenomena of Underfitting.

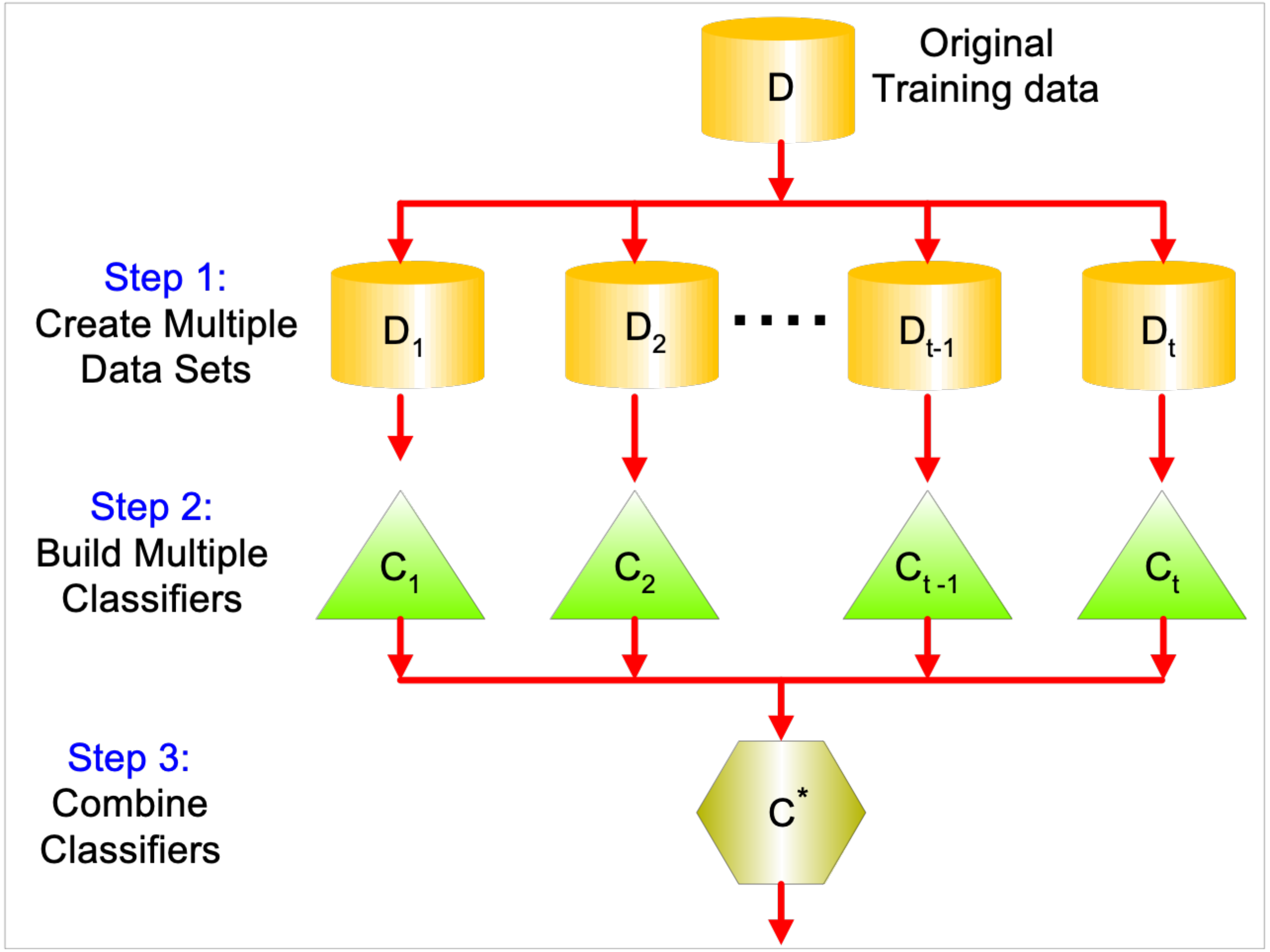
Figure 4.45. Plots showing the behavior of two-dimensional solutions with constant L_2 and L_1 norms.

Decision Trees

- Very high variance. *Unstable*
- Why?
  - The greedy algorithm
  - Small changes can have large effects by changing an early split, hence completely changing the structure underneath!
- Low bias. They are very rich in what they can capture

- Compare with linear models, which are typically low variance, high bias

Bagging: An Approach to Reducing Variance



Bagging

- Sampling with replacement

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Build classifier on each bootstrap sample

- Each data instance has probability $1 - (1 - 1/n)^n$ of being selected as part of the bootstrap sample

Bagging Algorithm

Algorithm 5.6 Bagging Algorithm

Let k be the number of bootstrap samples.
for $i = 1$ to k **do**
  Create a bootstrap sample of size n, D_i.
  Train a base classifier C_i on the bootstrap sample D_i.
end for
$C^*(x) = \arg\max_y \sum_i \delta\big(C_i(x) = y\big)$, $\{\delta(\cdot) = 1$ if its argument is true, and 0 otherwise.$\}$

Random Forests

- One more trick to further decorrelate each bagged tree
- Before each split randomly subsample *k* features (without replacement) and only consider these for your split.
  - Often square root of total number of features
- Empirically *enormously* successful
- Very few hyperparameters (number of bags and number of features to consider at each split; standard approaches to determining both)

Random Forests: Other Benefits

- *Out of bag* error is a nice estimate of test error that comes for free!
  - For each example in the training set, use predictions on it only from each tree constructed from a bag in which it is not present
  - Valid (under)-estimate of accuracy on that example
  - Can construct learning curve to determine when to stop adding bags!
- Many implementations are accompanied by a *feature scoring* method that gives you some sense of how important each feature is to obtaining high accuracy