Mixture Models

C1 C2 . . . Ck.

+

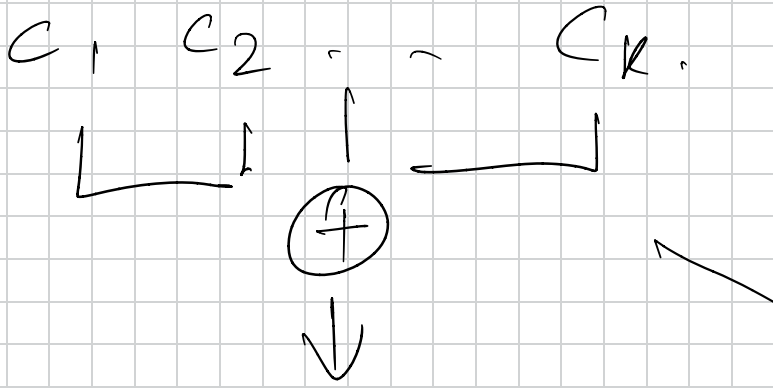
Ensemble Methods & Random Forest

Rishabh Iyer

University of Texas at Dallas

Acknowledgement: Nick Rouzzi, Vibhav Gogate, David Sontag, Killian Weinberger, Aarti Singh

Ensemble Methods



Bagging

↓

↓ Variance

Boosting

↓

↓ Bias

Reduce Variance Without Increasing Bias

$\bar{Z} = \sum_{i=1}^{N} Z_i / N$

- Averaging reduces variance: let $Z_1, \ldots, Z_N$ be i.i.d random variables

$$Var\left(\frac{1}{N}\sum_i Z_i\right) = \frac{1}{N}Var(Z_i)$$

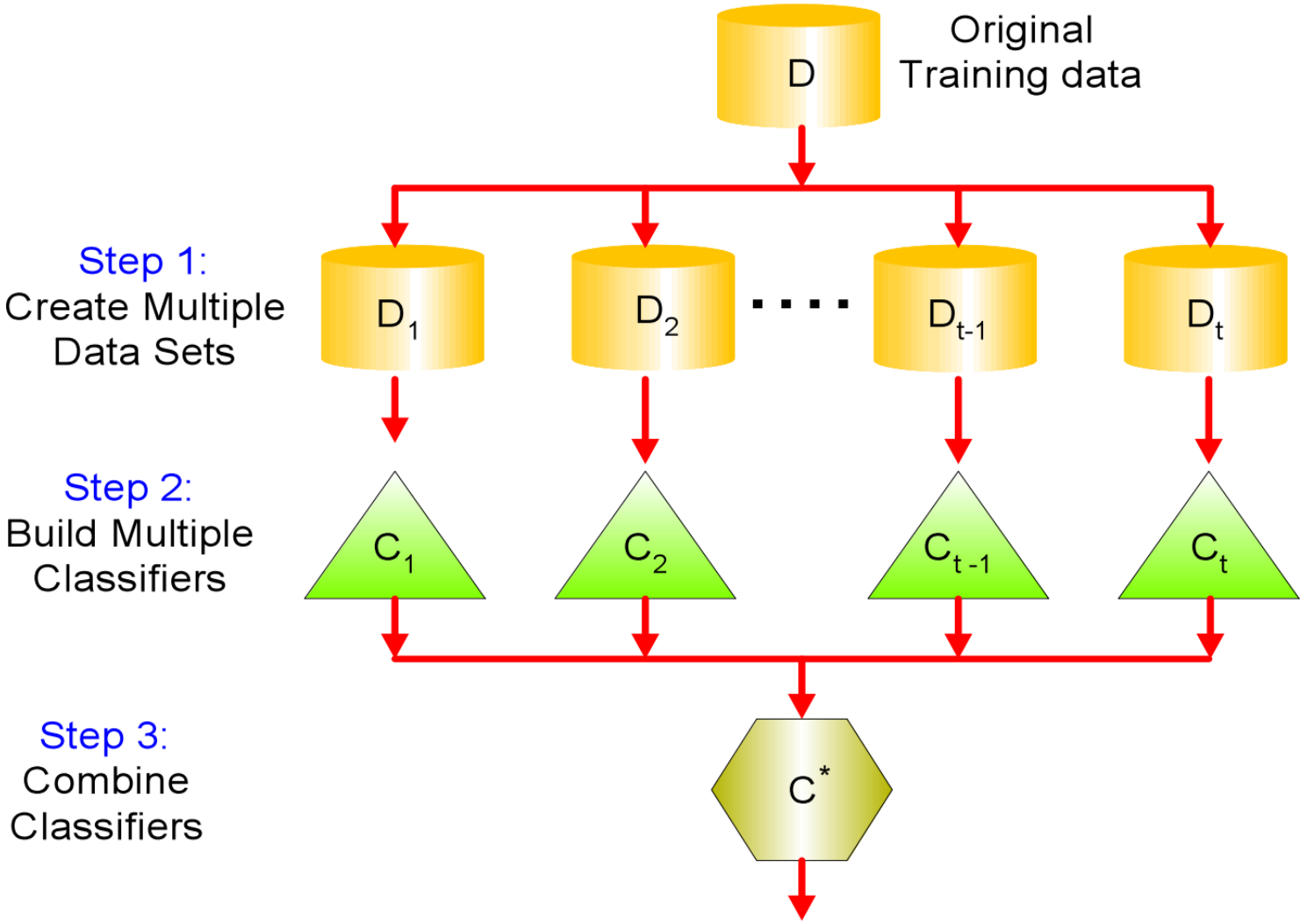
- Idea: average models to reduce model variance
- How to apply it to our setting?
  - Only one training set
  - Where do multiple models come from?

$E[\bar{Z}] = E[Z_i]$

Bagging: Bootstrap Aggregation

- Take repeated bootstrap samples from training set D (Breiman, 1994)
- Bootstrap sampling: Given set D containing N training examples, create D' by drawing N examples at random **with replacement** from D
- Bagging:
  - Create k bootstrap samples $D_1, \ldots, D_k$
  - Train distinct classifier on each D_i
  - Classify new instance by majority vote / average

Bagging: Bootstrap Aggregation



[image from the slides of David Sontag]

Bagging

Data	1	2	3	4	5	6	7	8	9	10
BS 1	7	1	9	10	7	8	8	4	7	2
BS 2	8	1	3	1	1	9	7	4	10	1
BS 3	5	4	8	8	2	5	5	7	8	8

- Build a classifier from each bootstrap sample
- In each bootstrap sample, each data point has probability $\left(1-\frac{1}{N}\right)^N$ of not being selected
  - Expected number of distinct data points in each sample is then

$$N \cdot \left(1-\left(1-\frac{1}{N}\right)^N\right) \approx N \cdot (1-\exp(-1)) = .632 \cdot N$$

$$D \longrightarrow D_i$$

$$D = [(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})]$$

$$h(X_i)$$

D & D_i should have the same statistical property

$$E_D[Y] = \sum_{j \in D} y_j$$

$$E_{D_i}[Y] = \sum_{j \in D_i} y_j$$

h_D & h_{D_i} should be iid.

$$h_{D_1}, h_{D_2}, \ldots, h_{D_t}$$

$$h(x) = \frac{\sum_{i=1}^{t} h_{D_i}(x)}{t}$$

Bagging

Data	1	2	3	4	5	6	7	8	9	10
BS 1	7	1	9	10	7	8	8	4	7	2
BS 2	8	1	3	1	1	9	7	4	10	1
BS 3	5	4	8	8	2	5	5	7	8	8

- Build a classifier from each bootstrap sample
- In each bootstrap sample, each data point has probability $\left(1 - \frac{1}{N}\right)^N$ of not being selected
  - If we have 1 TB of data, each bootstrap sample will be ~ 632GB (this can present computational challenges, e.g., you shouldn't replicate the data)

What does averaging mean?

Regression

$$\bar{h} = \frac{\sum_{i=1}^{t} h_{D_i}(x)}{t}$$

↓

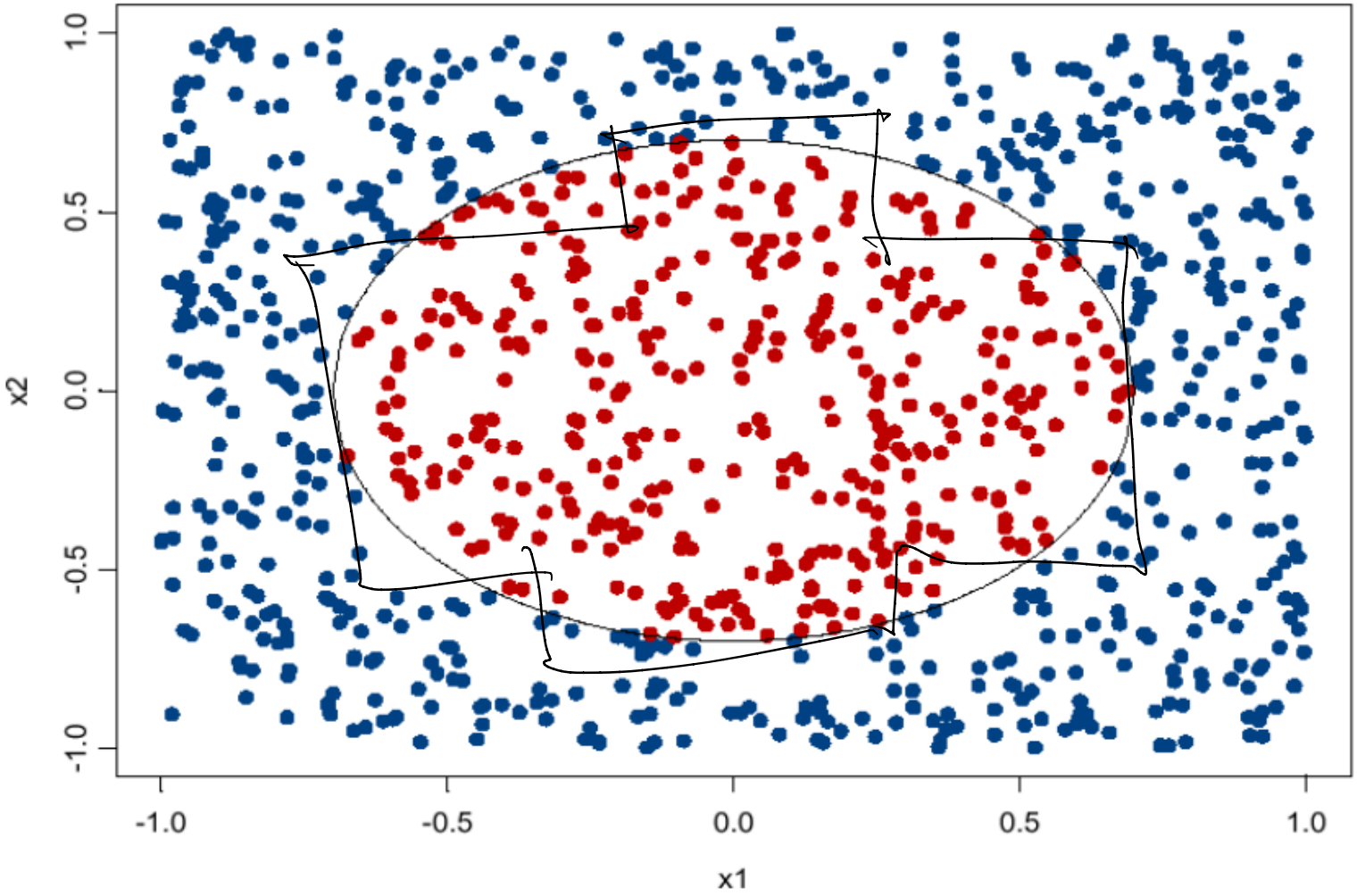
Average.

Classification

$$\bar{h} = \frac{\sum_{i=1}^{t} h_{D_i}(x)}{t} \; ?$$

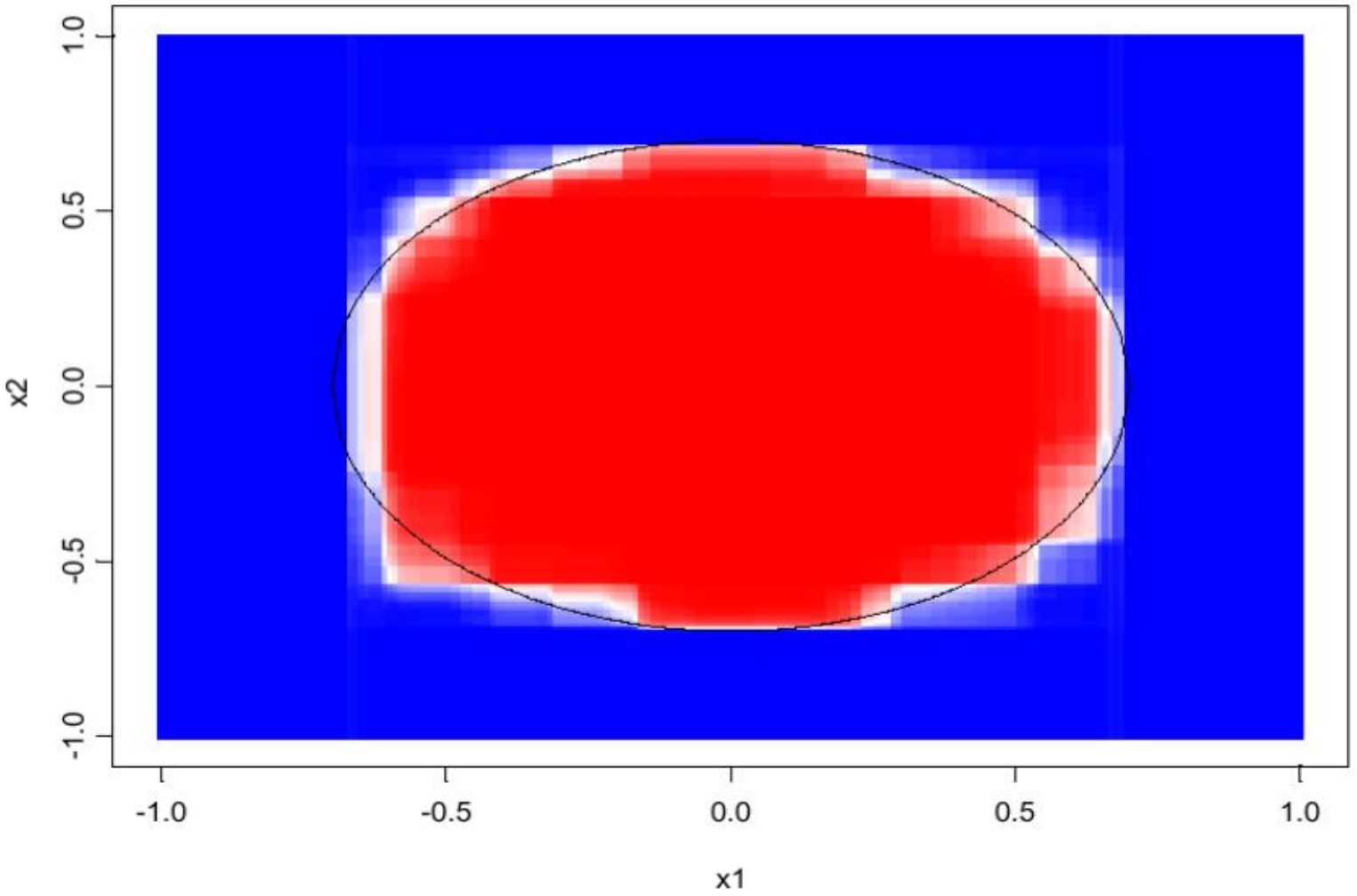
- Binary → $\sum$ 0/1 → majority vote.
- Multiclass → $\begin{bmatrix}0\\1\\0\end{bmatrix} + \begin{bmatrix}0\\1\\0\end{bmatrix} + \begin{bmatrix}0\\1\\0\end{bmatrix}$ → majority vote

Decision Tree Bagging



[image from the slides of David Sontag]

Decision Tree Bagging (100 Bagged Trees)



[image from the slides of David Sontag]

Bagging Results

	Without Bagging	With Bagging	
Data Set	$\bar{e}_S$	$\bar{e}_B$	**Decrease**
waveform	29.1	19.3	34%
heart	4.9	2.8	43%
breast cancer	5.9	3.7	37%
ionosphere	11.2	7.9	29%
diabetes	25.3	23.9	6%
glass	30.4	23.6	22%
soybean	8.6	6.8	21%

Breiman "Bagging Predictors" Berkeley Statistics Department TR#421, 1994

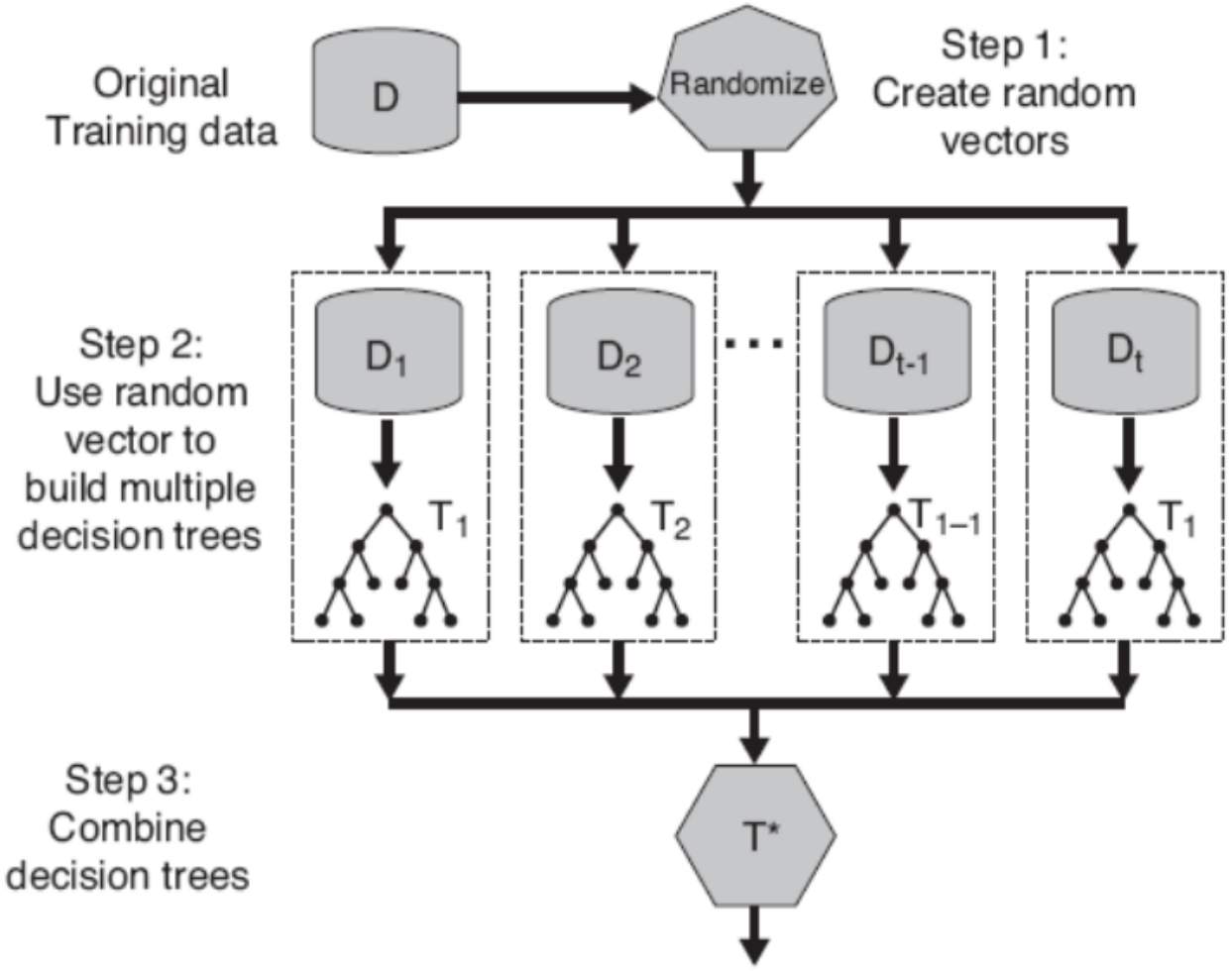
Bagging does not reduce bias.

1. Start with a high complexity model
   [DT, Large depth, small min leaf param]
   [We are currently overfitting]

2. Apply bagging by training the same high complexity model t times.

   ⇒ Bias remains low.
   Variance reduces significantly

Random Forests



Random Forests

- Ensemble method specifically designed for decision tree classifiers
- Introduce two sources of randomness: “bagging” and “random input vectors”
  - Bagging method: each tree is grown using a bootstrap sample of training data
  - Random vector method: best split at each node is chosen from a random sample of m attributes instead of all attributes

Random Forest Algorithm

- For $b = 1$ to B ← Same as "t"
  - Draw a bootstrap sample of size N from the data
  - Grow a tree T_b using the bootstrap sample as follows
    - Choose m attributes uniformly at random from the data
    - Choose the best attribute among the m to split on
    - Split on the best attribute and recurse (until partitions have fewer than s_{min} number of nodes)

- Prediction for a new data point x
  - Regression: $\frac{1}{B}\sum_b T_b(x)$
  - Classification: choose the majority class label among $T_1(x), \ldots, T_B(x)$

Random Forest Demo

A demo of random forests implemented in JavaScript

When Will Bagging Improve Accuracy?

- Depends on the stability of the base-level classifiers
- A learner is unstable if a small change to the training set causes a large change in the output hypothesis ← high variance
  - If small changes in D cause large changes in the output, then there will likely be an improvement in performance with bagging
- Bagging can help unstable procedures, but could hurt the performance of stable procedures
  - Decision trees are unstable
  - k-nearest neighbor is stable

high variance ↓