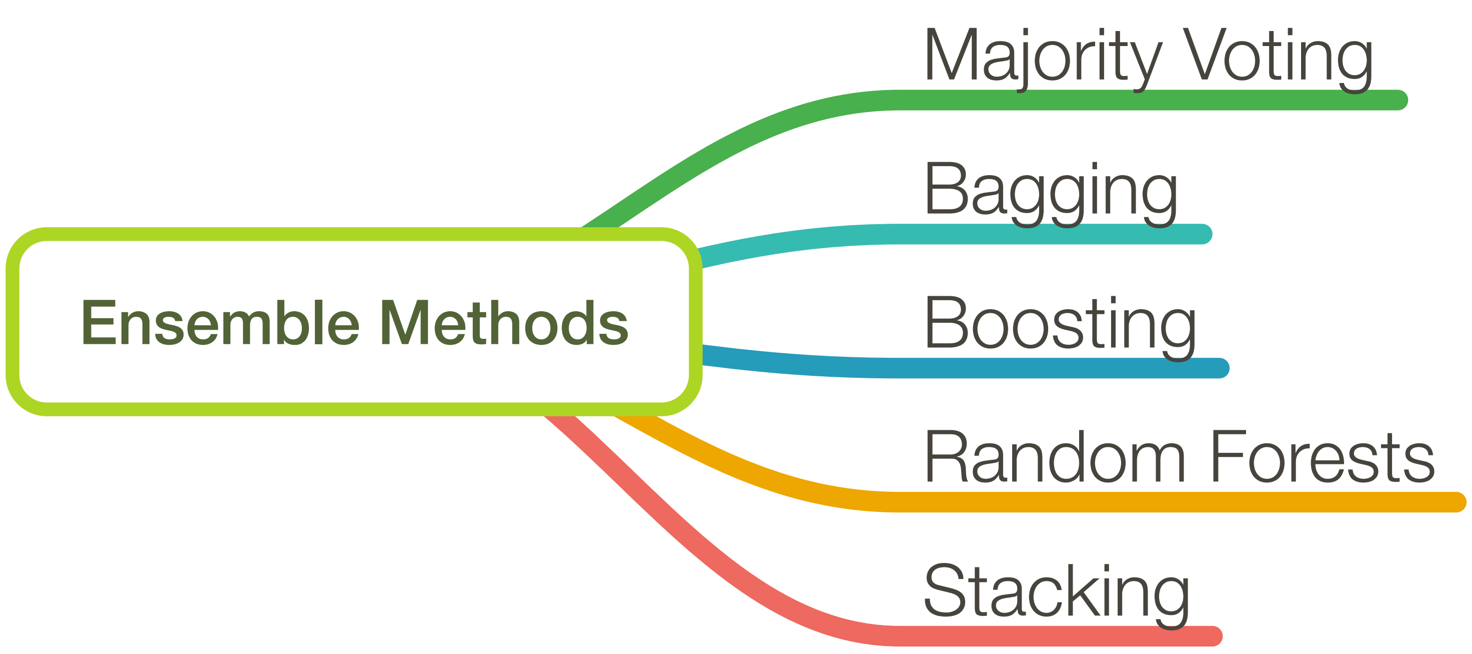
Lecture 07

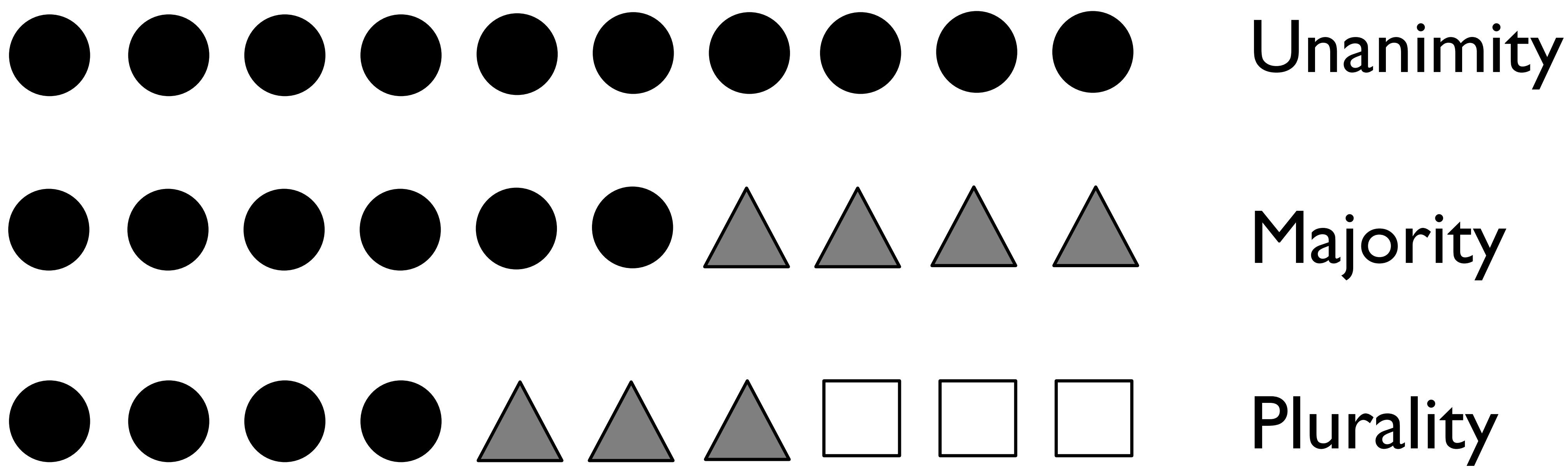
Ensemble Methods

STAT 479: Machine Learning, Fall 2018
Sebastian Raschka
http://stat.wisc.edu/~sraschka/teaching/stat479-fs2018/

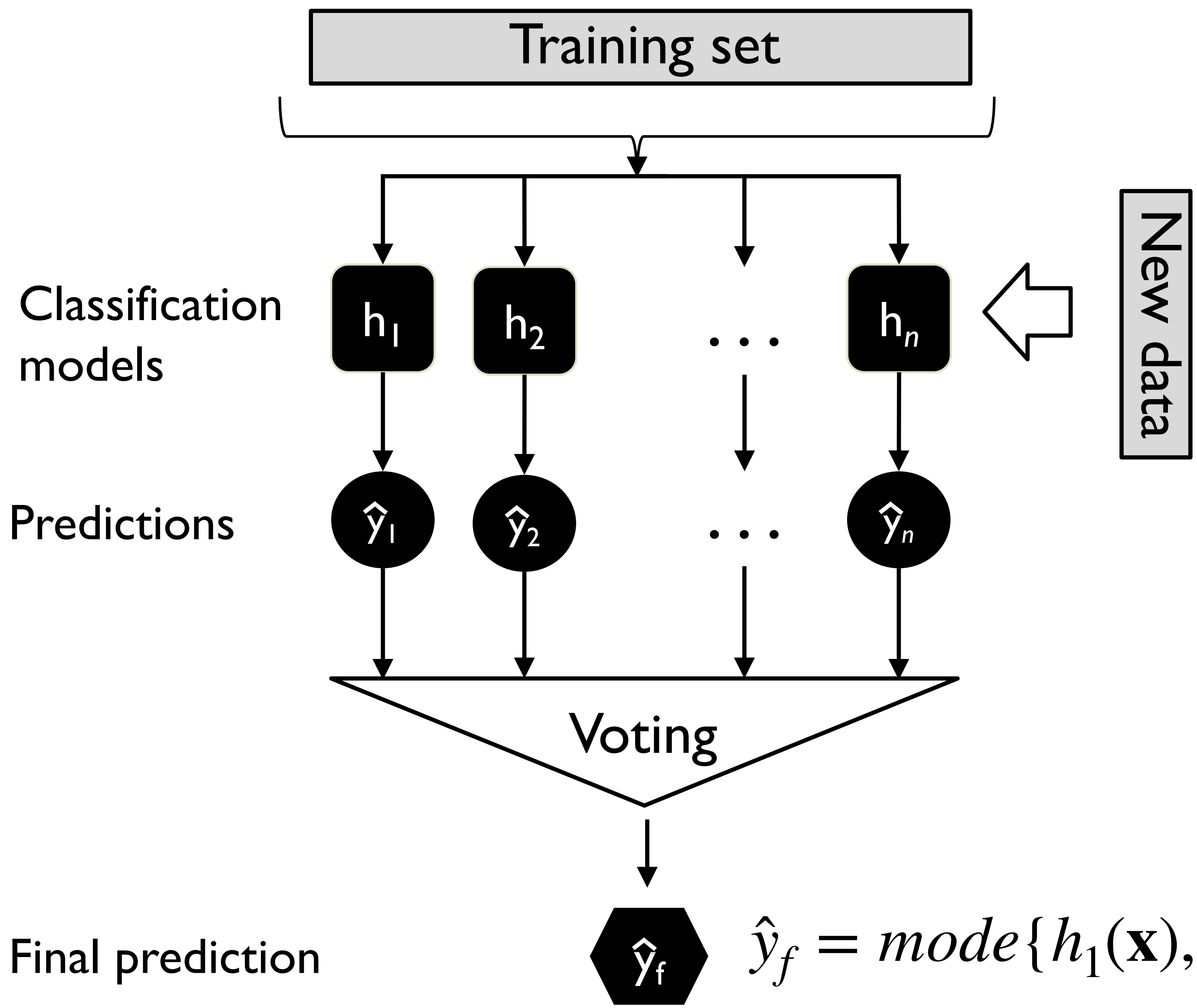
Overview


Ensemble Methods
Majority Voting
Bagging
Boosting
Random Forests
Stacking

Majority Voting


Unanimity
Majority
Plurality

Majority Vote Classifier



$$\hat{y}_f = mode\{h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots h_n(\mathbf{x})\}$$

where $h_i(\mathbf{x}) = \hat{y}_i$

Why Majority Vote?

- assume *n* independent classifiers with a base error rate ϵ
- here, independent means that the errors are uncorrelated
- assume a binary classification task
- assume the error rate is better than random guessing (i.e., lower than 0.5 for binary classification)

$$\forall \epsilon_i \in \{\epsilon_1, \epsilon_2, \ldots, \epsilon_n\}, \epsilon_i < 0.5$$

Why Majority Vote?

- assume *n* independent classifiers with a base error rate ϵ
- here, independent means that the errors are uncorrelated
- assume a binary classification task
- assume the error rate is better than random guessing (i.e., lower than 0.5 for binary classification)

$$\forall \epsilon_i \in \{\epsilon_1, \epsilon_2, \ldots, \epsilon_n\}, \epsilon_i < 0.5$$

The probability that we make a wrong prediction via the ensemble if *k* classifiers predict the same class label

$$P(k) = \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k} \qquad k > \lceil n/2 \rceil$$

Why Majority Vote?

The probability that we make a wrong prediction via the ensemble if *k* classifiers predict the same class label

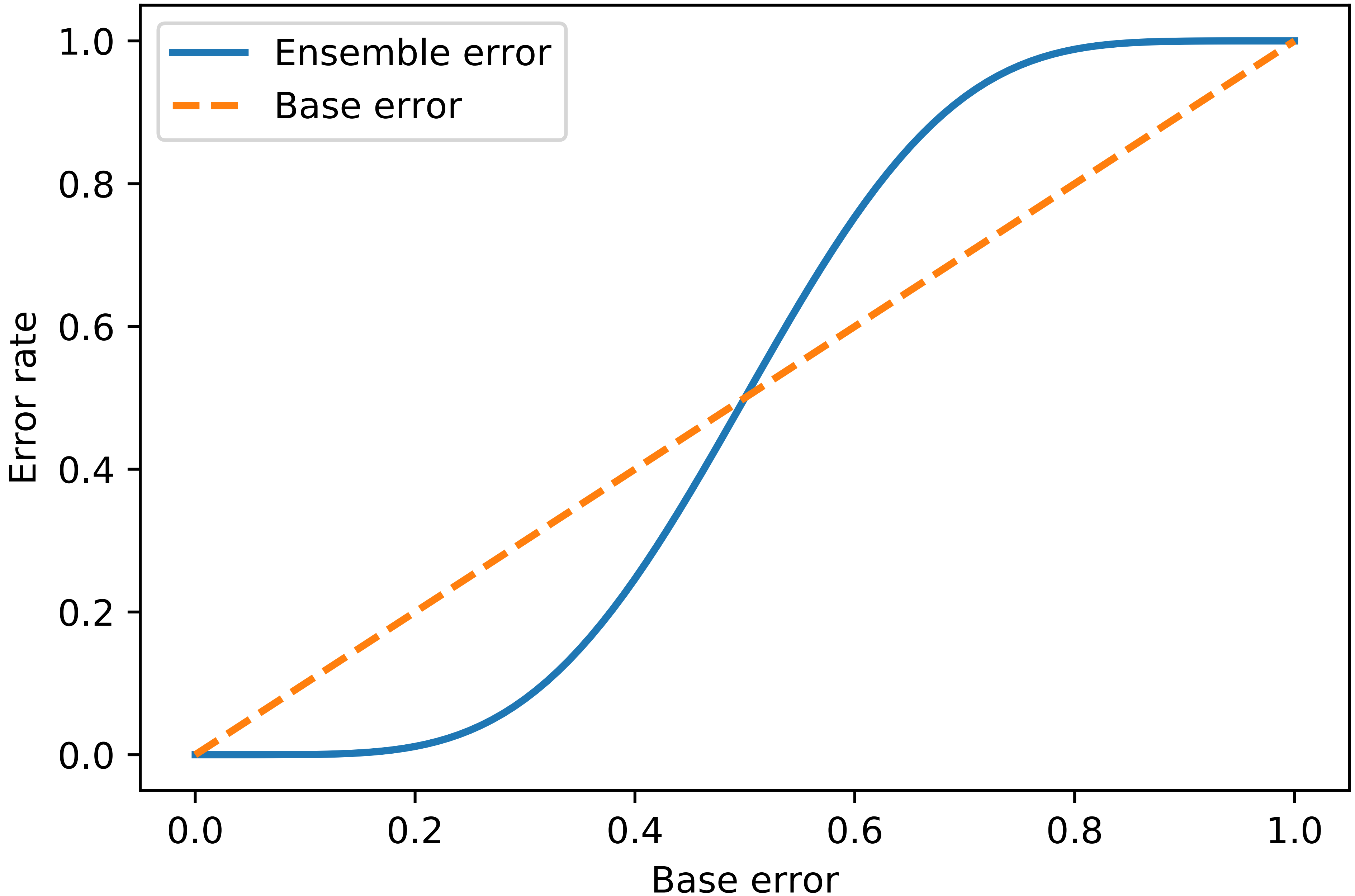
$$P(k) = \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k} \qquad k > \lceil n/2 \rceil$$

Ensemble error:

$$\epsilon_{ens} = \sum_{k}^{n} \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k}$$

$$\epsilon_{ens} = \sum_{k=6}^{11} \binom{11}{k} 0.25^k (1-0.25)^{11-k} = 0.034$$

$$\epsilon_{ens} = \sum_{k}^{n} \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k}$$



"Soft" Voting

$$\hat{y} = \arg\max_j \sum_{i=1}^{n} w_i p_{i,j}$$

$p_{i,j}$: predicted class membership probability of the *i*th classifier for class label *j*

w_j: optional weighting parameter, default $w_i = 1/n, \forall w_i \in \{w_1, \ldots, w_n\}$

"Soft" Voting

Use only for well-calibrated classifiers!

$$\hat{y} = \arg\max_j \sum_{i=1}^{n} w_i p_{i,j}$$

$p_{i,j}$: predicted class membership probability of the *i*th classifier for class label *j*

w_j: optional weighting parameter, default $w_i = 1/n, \forall w_i \in \{w_1, \ldots, w_n\}$

"Soft" Voting

$$\hat{y} = \arg\max_{j} \sum_{i=1}^{n} w_i p_{i,j}$$

Binary classification example

$j \in \{0,1\}$ $\quad$ $h_i (i \in \{1,2,3\})$

$$h_1(\mathbf{x}) \rightarrow [0.9, 0.1]$$

$$h_2(\mathbf{x}) \rightarrow [0.8, 0.2]$$

$$h_3(\mathbf{x}) \rightarrow [0.4, 0.6]$$

"Soft" Voting $\hat{y} = \arg\max_j \sum_{i=1}^{n} w_i p_{i,j}$

Binary classification example

$j \in \{0,1\} \qquad h_i(i \in \{1,2,3\})$

$$h_1(\mathbf{x}) \to [0.9, 0.1]$$
$$h_2(\mathbf{x}) \to [0.8, 0.2]$$
$$h_3(\mathbf{x}) \to [0.4, 0.6]$$

$$p(i = 0 \mid \mathbf{x}) = 0.2 \cdot 0.9 + 0.2 \cdot 0.8 + 0.6 \cdot 0.4 = 0.58$$
$$p(i = 1 \mid \mathbf{x}) = 0.2 \cdot 0.1 + 0.2 \cdot 0.2 + 0.6 \cdot 0.6 = 0.42$$

$$\hat{y} = \arg\max_j \left\{ p(i = 0 \mid \mathbf{x}), p(i = 1 \mid \mathbf{x}) \right\}$$

Bagging

(Bootstrap Aggregating)

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.

Algorithm 1 Bagging

```
Let $n$ be the number of bootstrap samples

for i=1 to $n$ do
    Draw bootstrap sample of size $m$, $\mathcal{D}_i$
    Train base classifier $h_i$ on $\mathcal{D}_i$
$\hat{y} = mode\{h_1(\mathbf{x}), ..., h_n(\mathbf{x})\}$
```

Bootstrap Sampling

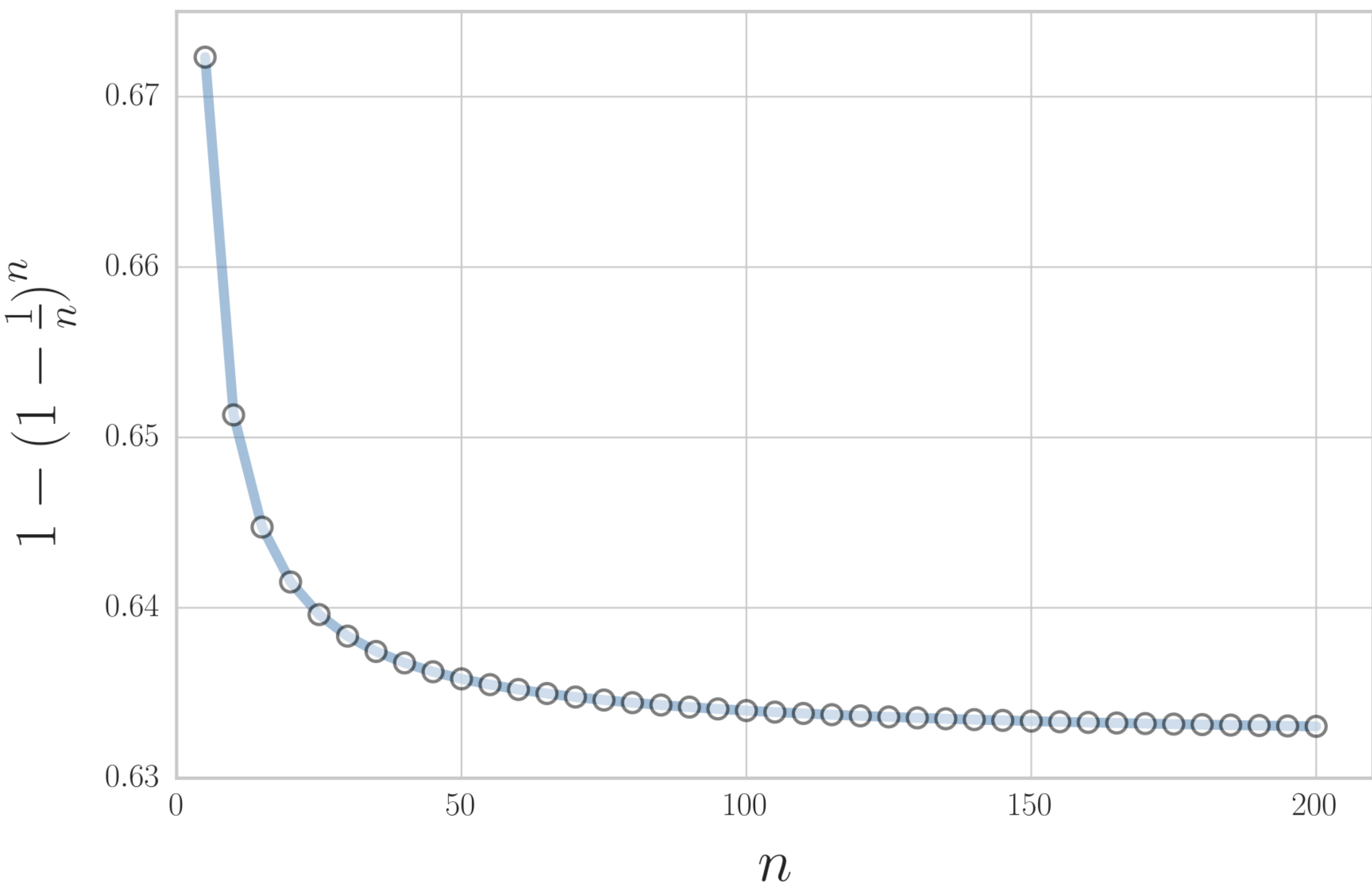
$$P(\textbf{not chosen}) = \left(1 - \frac{1}{n}\right)^n,$$

$$\frac{1}{e} \approx 0.368, \quad n \to \infty\,.$$

$$P(\textbf{not chosen}) = \left(1 - \frac{1}{n}\right)^n,$$

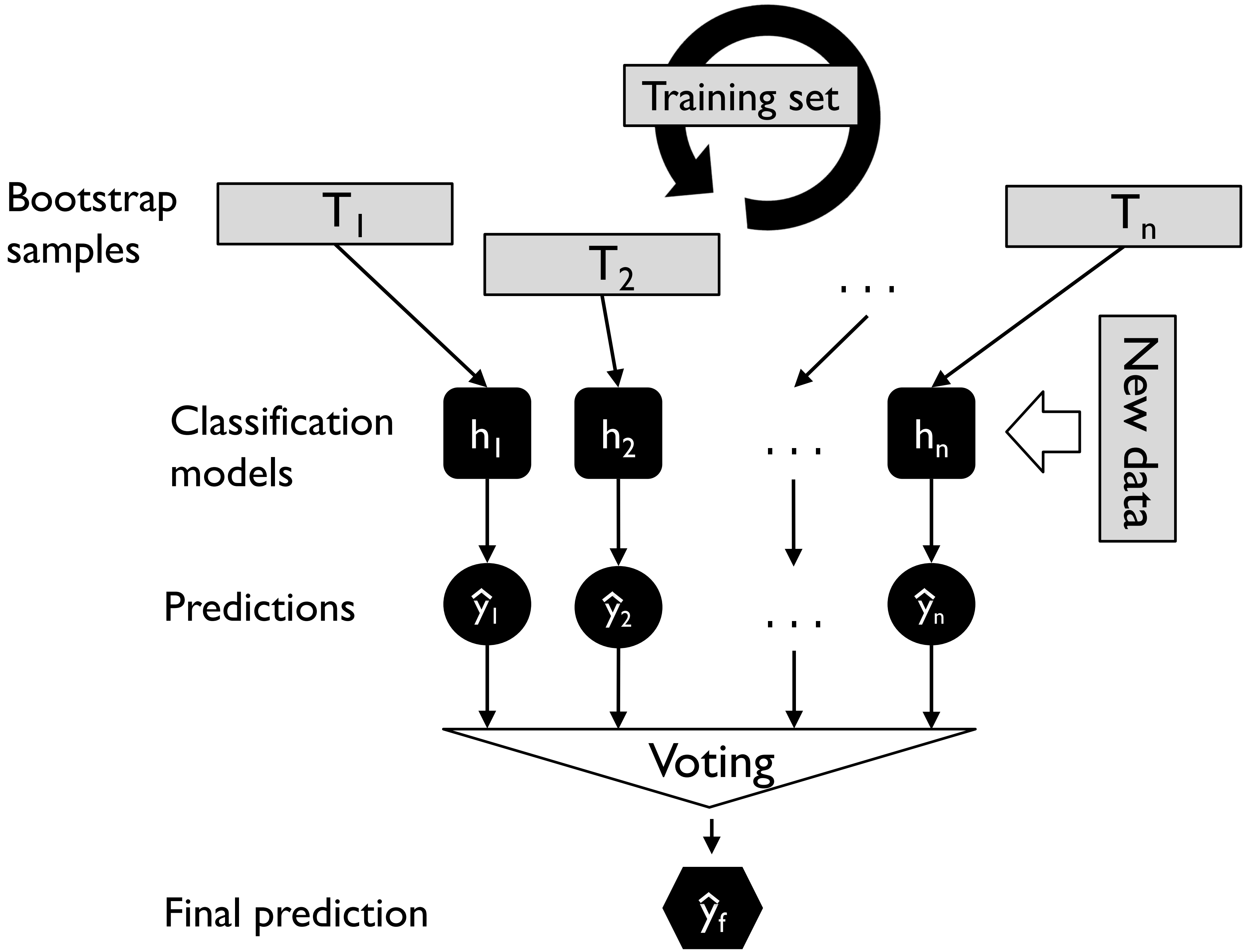
$$\frac{1}{e} \approx 0.368, \quad n \to \infty\,.$$

$$P(\textbf{chosen}) = 1 - \left(1 - \frac{1}{n}\right)^n \approx 0.632$$



Training example indices	Bagging round 1	Bagging round 2	…
1	2	7	…
2	2	3	…
3	1	2	…
4	3	1	…
5	7	1	…
6	2	7	…
7	4	7	…

h_1 h_2 h_n



$$\hat{y}_f = mode\{h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots h_n(\mathbf{x})\}$$

where $h_i(\mathbf{x}) = \hat{y}_i$

To be continued ...

This file will be updated with

- Boosting
- Random Forests
- Stacking