Ensemble Methods in Machine Learning

LO: Be able to

Describe the difference between strong and weak learnings.

Describe the techniques of bootstrapping, gradient boosting, and Ada boosting.

Implement the random forest machine learning model.

START AUDIO 02

Boosting

- **Boosting** is
  - a machine learning ensemble meta-algorithm for reducing bias and variance in supervised learning,.
  - a family of machine learning algorithms that convert **weak learners** to **strong learners**.

Boosting methods

- Most boosting methods consist of iteratively adding the results of a weak learner to form a strong learner.
  - When adding the results of the weak learners, they are typically weighted by a factor related to the weak learner's accuracy.
  - Each time the results are added, the sample data are reweighted before the next weak learner is applied to them.

STOP

Algorithm
Gradient Boosting
AdaBoost
Tree Bagging and Random Forest

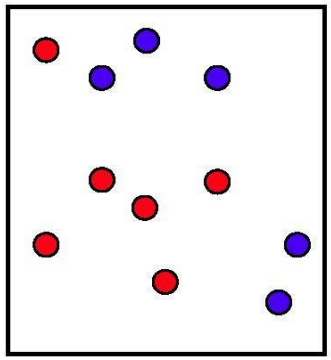
START AUDIO 03

AdaBoost (short for Adaptive Boosting)

- Formulated by Yoav Freund and Robert Schapire
- Awarded the Gödel Prize in 2003 for this work.

Initial data

$m=0$ $m=1$ $m=2$ $m=3$

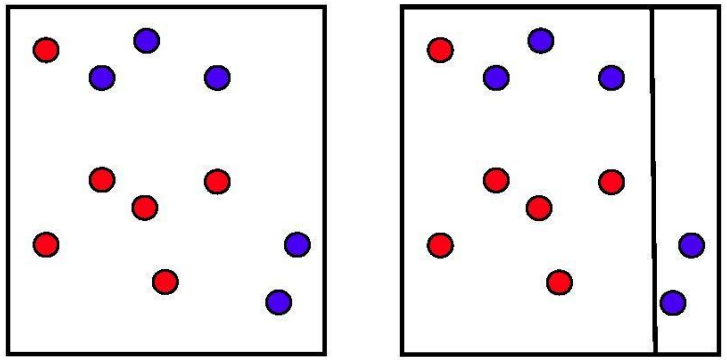


k_m

w_{m+1}

k_1

$m=0$ $m=1$ $m=2$ $m=3$

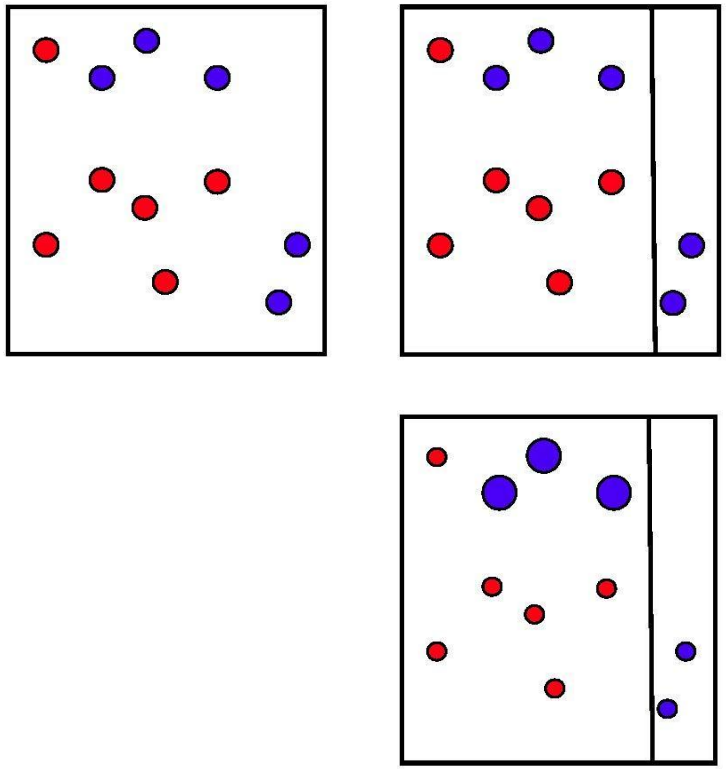


k_m

w_{m+1}

w_2

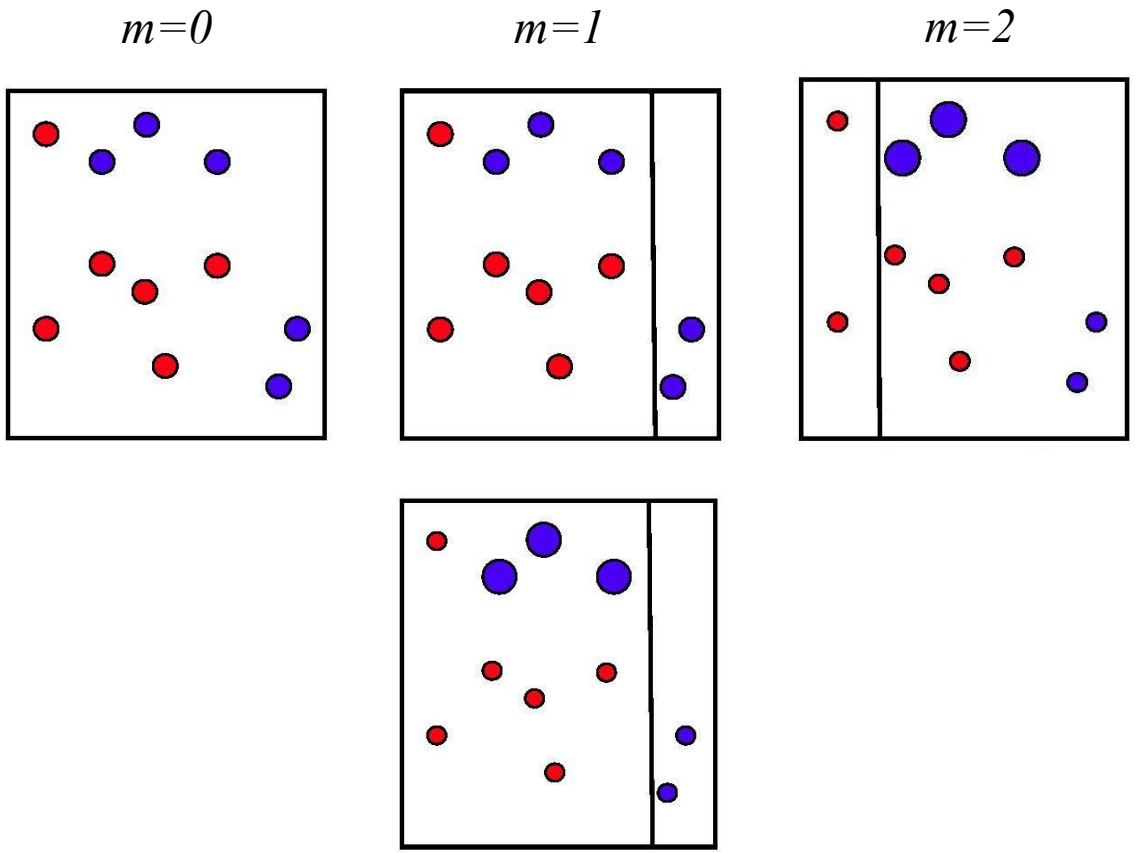
$m=0$ $m=1$ $m=2$ $m=3$



k_m

w_{m+1}

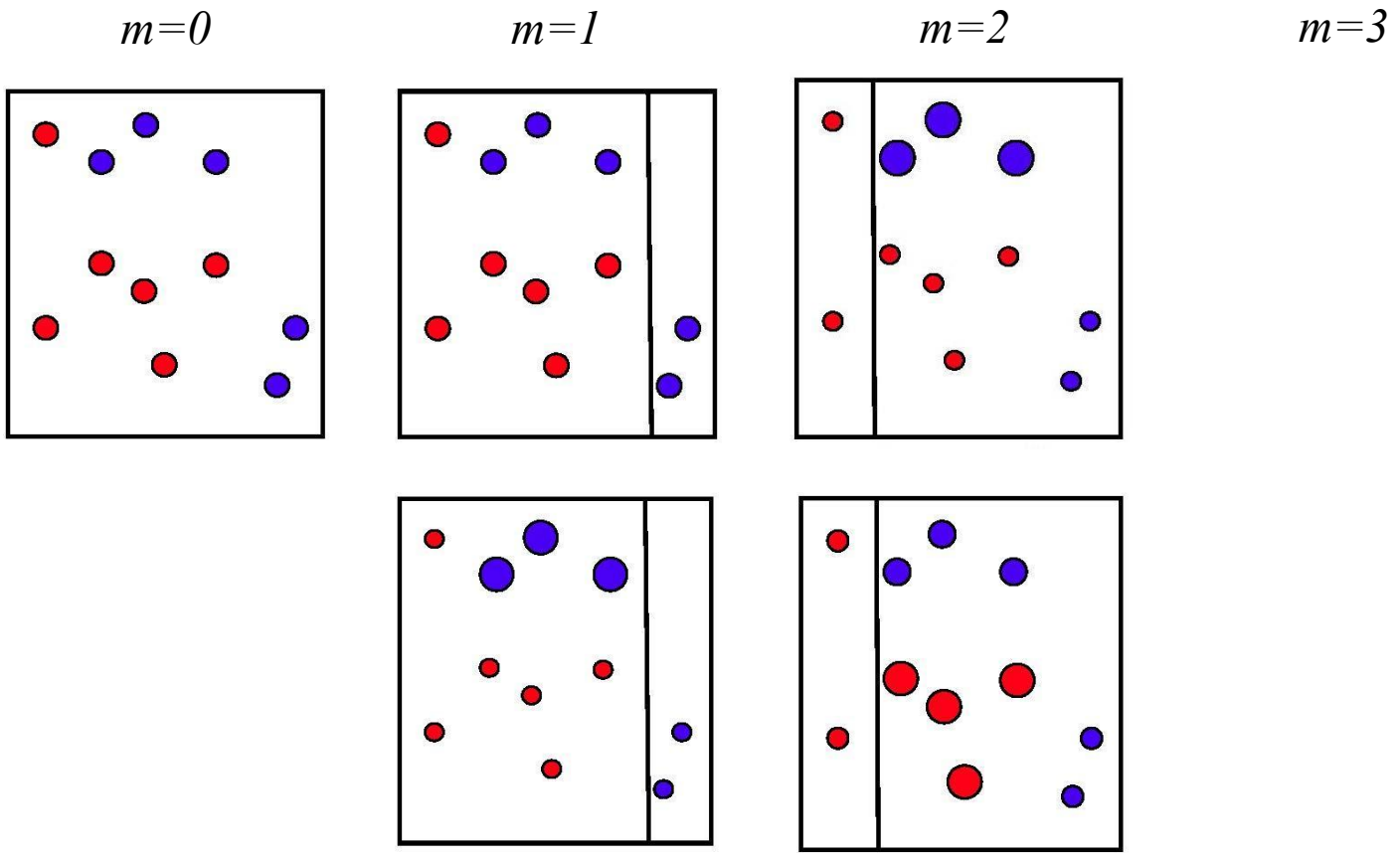
k_2



k_m

w_{m+1}

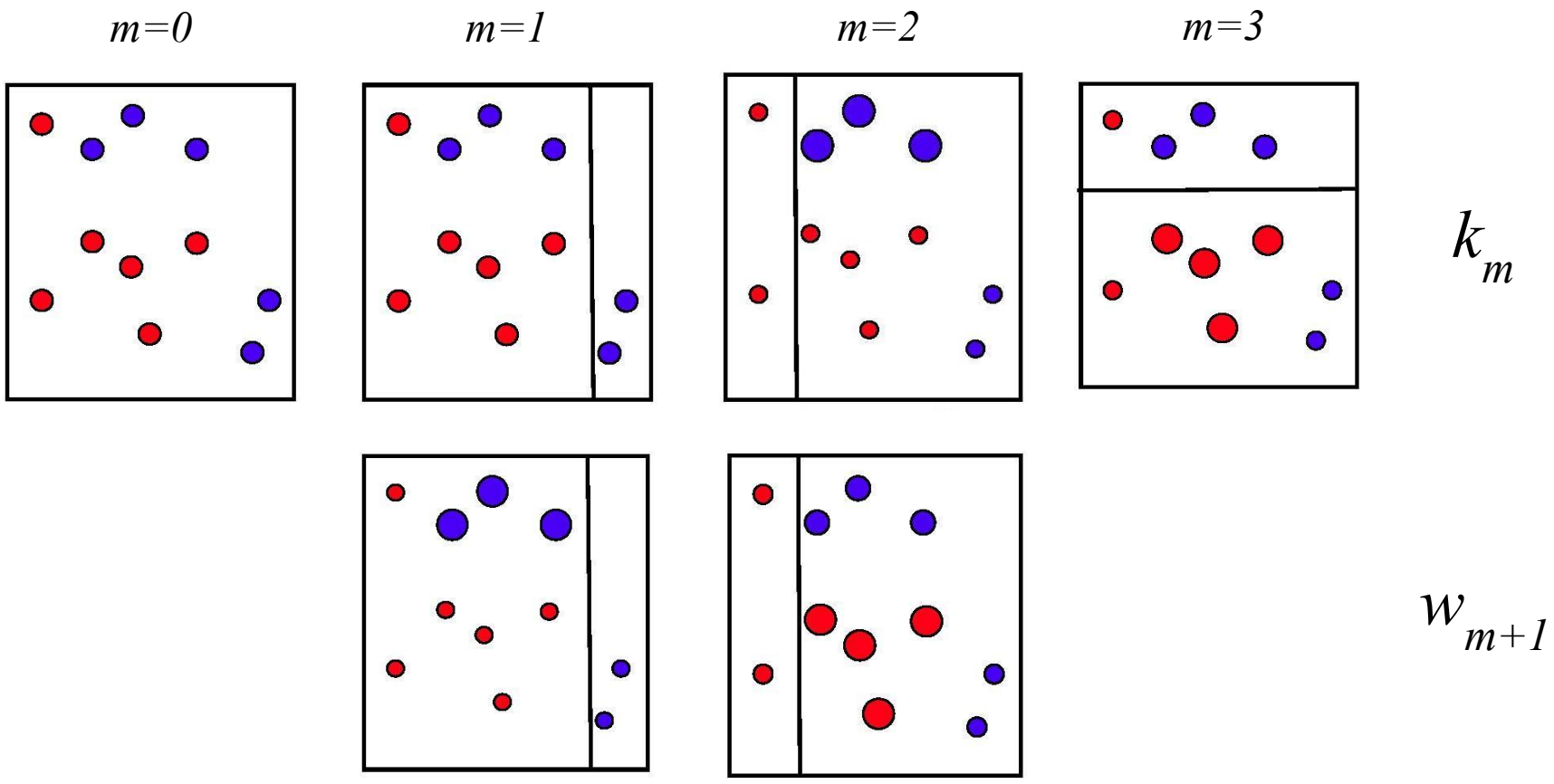
w_3



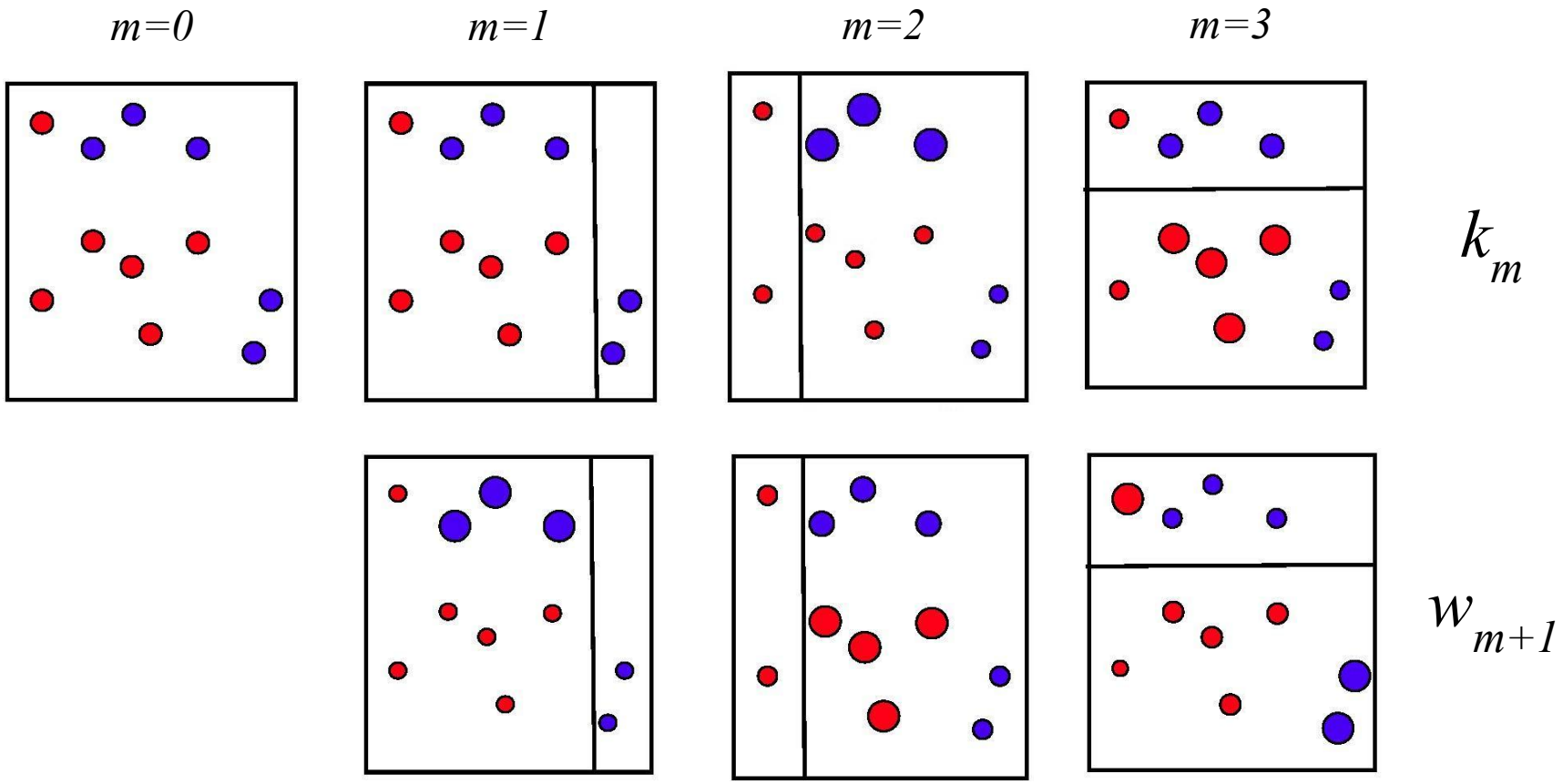
k_m

w_{m+1}

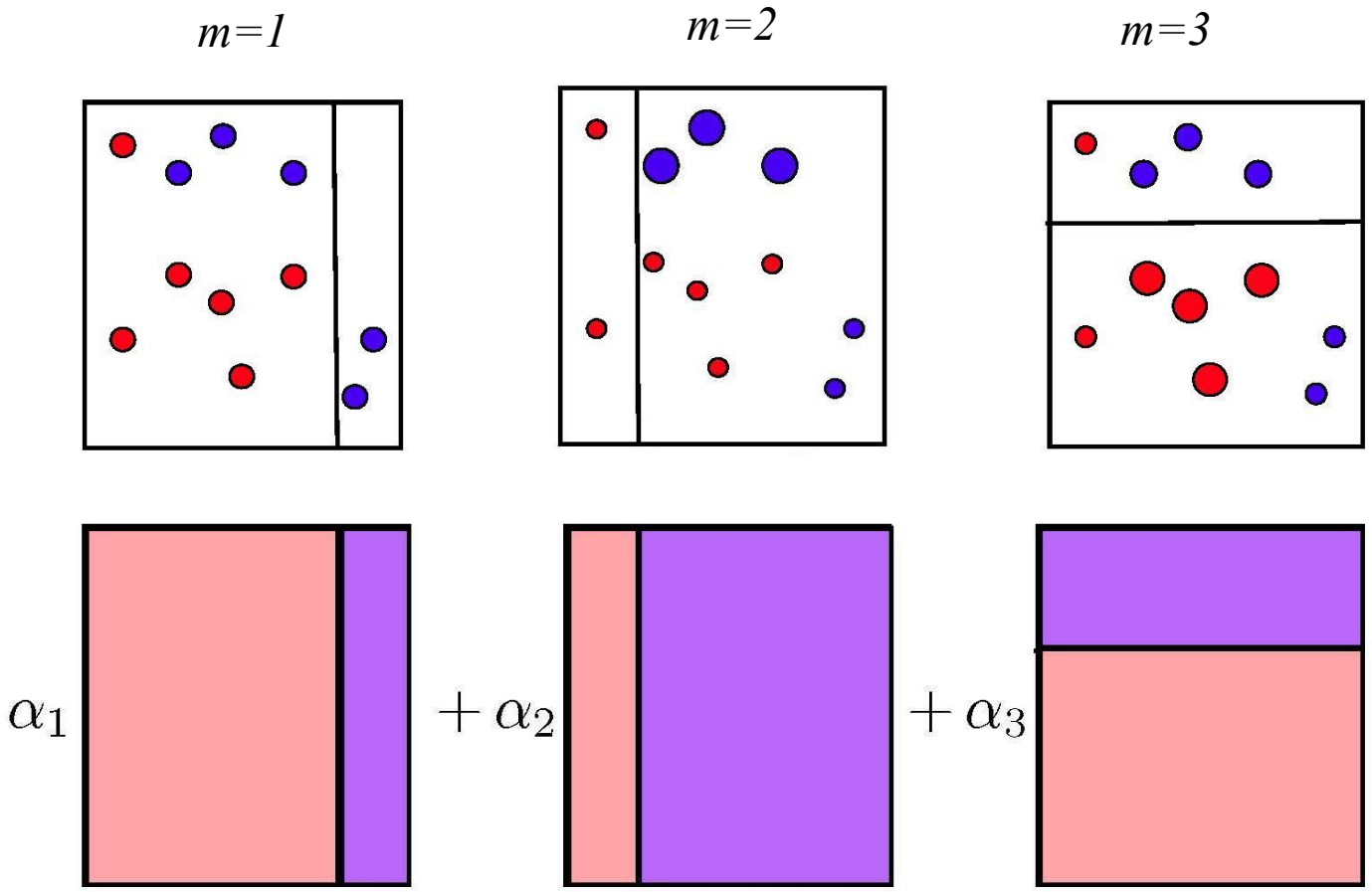
k_3



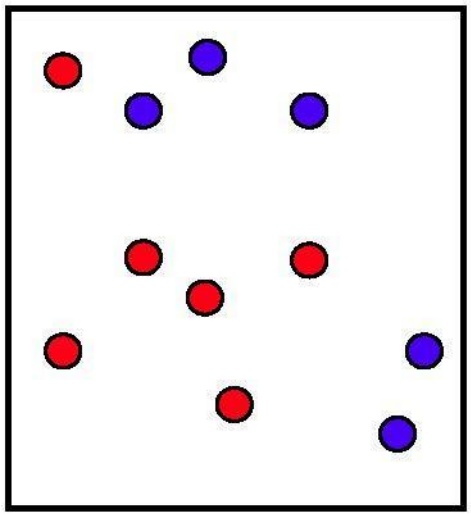
w_4

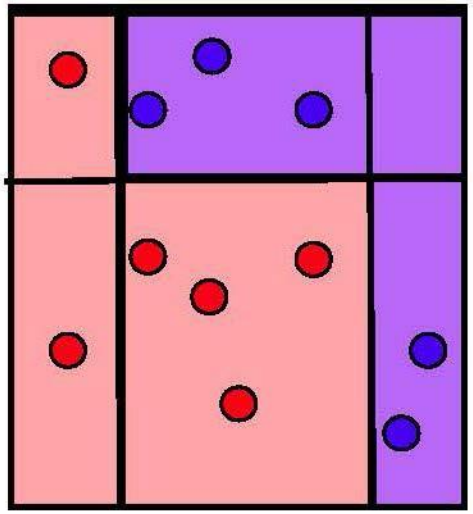


Final classifier components



Final classifier





STOP

Bootstrapping

NYU Tandon Bioinformatics
Introduction to Machine Learning
Professor Erik K. Grimmelmann, Ph.D.

START AUDIO 04

Bootstrapping

- **Bootstrapping** is any test or metric that relies on random sampling **with replacement**.
- Bootstrapping allows assigning measures of accuracy (defined in terms of bias, variance, confidence intervals, prediction error or some other such measure) to sample estimates.

Bootstrapping

- Bootstrapping can be used to generate multiple training dataset from a sample dataset.
- If a set of observations can be assumed to be from an I.I.D. population, an approximating distribution of the population can implemented by constructing a number of resamples (with replacement) of the observed dataset and **of equal size to the observed dataset**.

STOP

Algorithm
Gradient Boosting
AdaBoost
Tree Bagging and Random Forest

RISE

Tree Bagging algorithm

1) Randomly choose *n* samples from the training set with replacement to create a **bootstrap sample**.
2) Grow a decision tree from the bootstrap sample. At each node:
    1) Split the node using the feature for the best split according to the objective function (e.g., maximizing information gain).
3) Repeat steps 1) and 2) *k* times.
4) Aggregate the prediction by each tree to assign a classification by **majority vote**.

Random Forest algorithm

1) Randomly choose *n* samples from the training set with replacement to create a **bootstrap sample**.
2) Grow a decision tree from the bootstrap sample. At each node:
    1) **Randomly select *d* features without replacement**
    2) Split the node using the feature for the best split according to the objective function (e.g., maximizing information gain).
3) Repeat steps 1) and 2) *k* times.
4) Aggregate the prediction by each tree to assign a classification by **majority vote**.

What distinguishes the Random Forest algorithm from Bagging

1) **Randomly select *d* features without replacement**

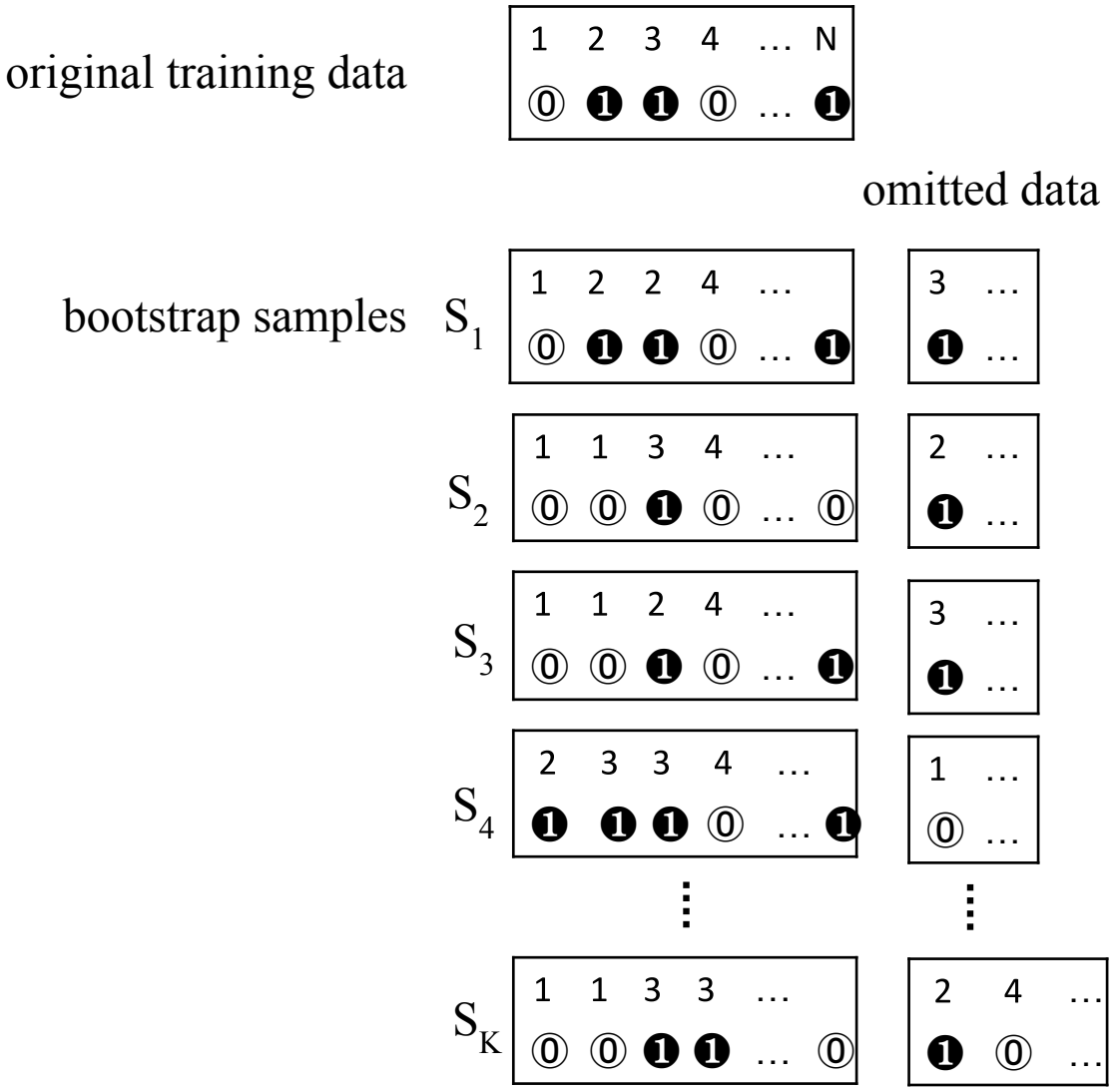
This added step is sometimes call **feature bagging.**

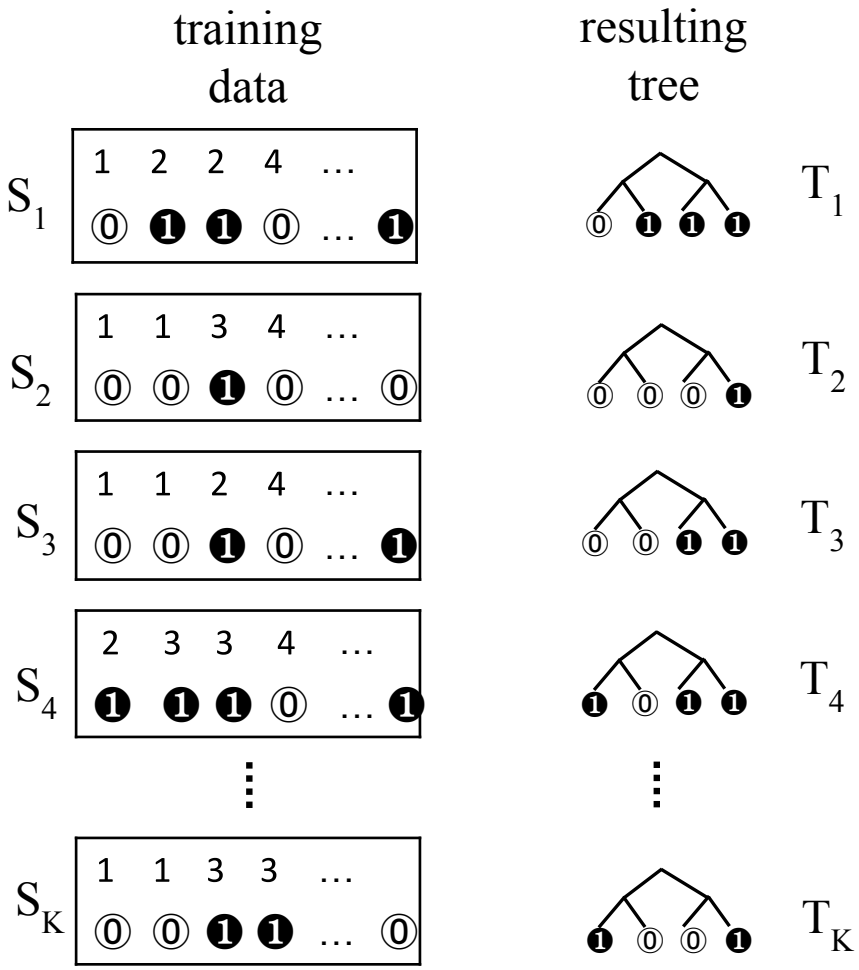
START AUDIO 05

original training data

Sample ID	1	2	3	4	…	N
Feature 1	$F_1^{(1)}$	$F_1^{(2)}$	$F_1^{(3)}$	$F_1^{(4)}$	…	$F_1^{(N)}$
Feature 2	$F_2^{(1)}$	$F_2^{(2)}$	$F_2^{(3)}$	$F_2^{(4)}$	…	$F_2^{(N)}$
⋮	⋮	⋮	⋮	⋮		⋮
Feature M	$F_M^{(1)}$	$F_M^{(2)}$	$F_M^{(3)}$	$F_M^{(4)}$	…	$F_M^{(N)}$
Class	⓪	❶	❶	⓪	…	❶

1	2	3	4	…	N
⓪	❶	❶	⓪	…	❶

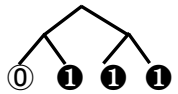

original training data
1 2 3 4 … N
omitted data
bootstrap samples
S_1
1 2 2 4 …
3 …
S_2
1 1 3 4 …
2 …
S_3
1 1 2 4 …
3 …
S_4
2 3 3 4 …
1 …
S_K
1 1 3 3 …
2 4 …


training data
resulting tree
S1
1 2 2 4 …
0 1 1 0 … 1
T1
S2
1 1 3 4 …
0 0 1 0 … 0
T2
S3
1 1 2 4 …
0 0 1 0 … 1
T3
S4
2 3 3 4 …
1 1 1 0 … 1
T4
SK
1 1 3 3 …
0 0 1 1 … 0
TK

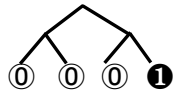
new instance

Sample ID	j
Feature 1	$F_1^{(j)}$
Feature 2	$F_2^{(j)}$
⋮	⋮
Feature M	$F_M^{(j)}$
Class	?

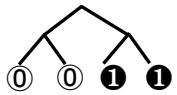
tree



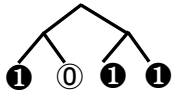
T_1



T_2

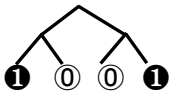


T_3



T_4

⋮



T_K

classification using tree T_i

C_1=❶

C_2=⓪

C_3=❶

C_4=❶

⋮

C_K=⓪

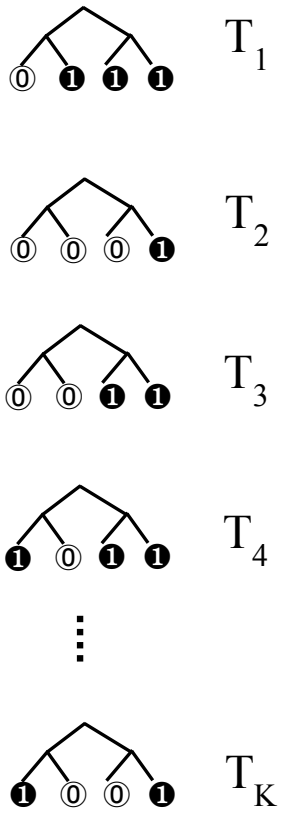
majority vote classification

C =❶

training data

Sample ID	1	2	…	N
Feature 1	$F_1^{(1)}$	$F_1^{(2)}$	…	$F_1^{(N)}$
Feature 2	$F_2^{(1)}$	$F_2^{(2)}$	…	$F_2^{(N)}$
⋮	⋮	⋮		⋮
Feature M	$F_M^{(1)}$	$F_M^{(2)}$	…	$F_M^{(N)}$
Class	⓪	❶	…	❶

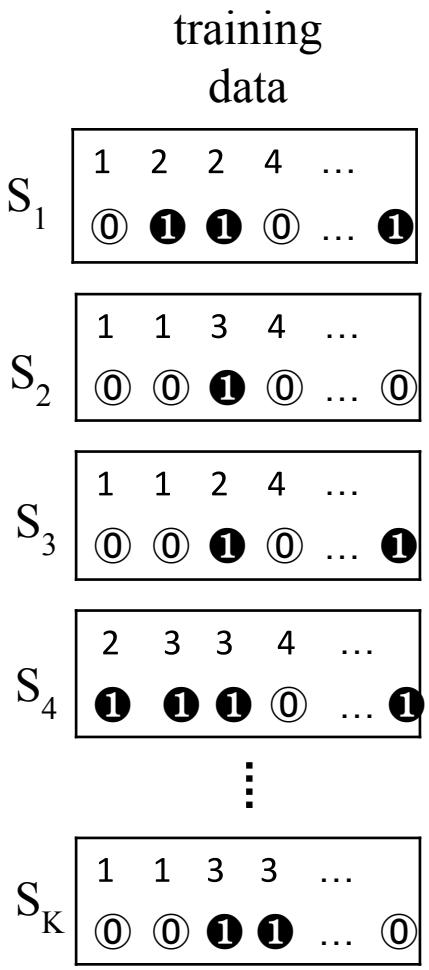
random forest

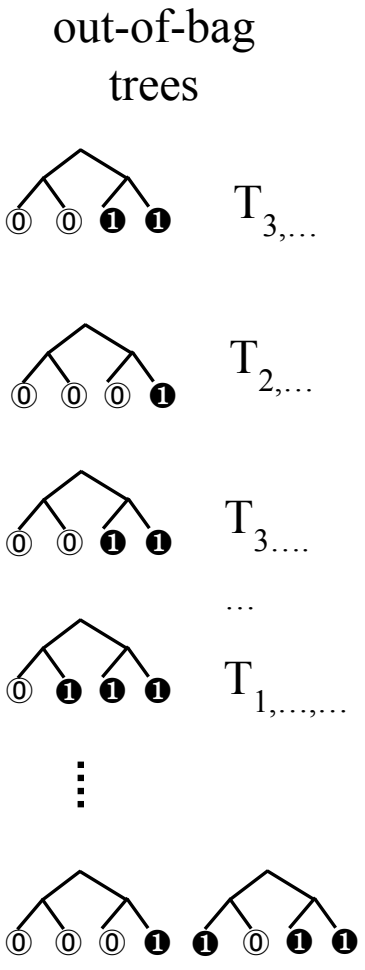


majority –vote predictions

Sample ID	1	2	…	N
Feature 1	$F_1^{(1)}$	$F_1^{(2)}$	…	$F_1^{(N)}$
Feature 2	$F_2^{(1)}$	$F_2^{(2)}$	…	$F_2^{(N)}$
⋮	⋮	⋮		⋮
Feature M	$F_M^{(1)}$	$F_M^{(2)}$	…	$F_M^{(N)}$
Class	⓪	❶	…	❶
Predicted Class	⓪	❶	…	⓪
Error	0	0	…	1

$$Error_{training} = \frac{1}{N}\sum_{i=1}^{N} Error^{(i)}$$





Classify each x_i using only the S_k that do not contain x_i to obtain $\hat{F}_{oob,k}(x_i)$.

$$Error_{oob} = \frac{1}{N}\sum_{i=1}^{N} 1_{\hat{F}_{oob}(x_i)\neq y_i}$$

STOP

Out-of-bag samples

RISE

When choosing a bootstrap sample ***S*** of size N, we select sample points without replacement; some points can (and likely will) be repeated in the bootstrap sample. Other points can (and likely will) be omitted.

The probability that a particular sample point is omitted from ***S*** is

$$\Pr(N) = \left(1 - \frac{1}{N}\right)^N .$$

$$\lim_{N\to\infty} \Pr(N) = \lim_{N\to\infty} \left(1 - \frac{1}{N}\right)^N = e^{-1} \approx 0.3678.$$ N does not have to be very large for Pr(N) to be close to it asymptotic value.

For example, $\Pr(5) \approx 0.328$ and $\Pr(50) \approx 0.364$.

Out-of-bag sample advantages

- The fact any given data point, $(\boldsymbol{x}_i, \mathrm{y}_i)$ is missing from a significant fraction ($\sim 1/3$) of bootstrap samples, $\hat{S}_k$, drawn from even a small sample set, S, creates an interesting opportunity.
- We can train our model on all of the bootstrap samples $\left\{\hat{S}_1, \hat{S}_2, \ldots \hat{S}_K\right\}$ and then test how well the model does in predicting y_i given $\boldsymbol{x}_i$ using only those bootstrap samples that do not contain $\boldsymbol{x}_i$.
- A data point is said to be **out of bag** for the bootstrap samples that do not contain it.
- Using out of bag samples for testing allows us to use all of our data for training and still have data for testing that has not been used in training.

Training a majority voting classifier using out-of-bag samples

$$\text{Let } \mathrm{F}(\boldsymbol{x}_i) = \text{MajorityVoting}\left(\mathrm{F}_1(\boldsymbol{x}_i), \mathrm{F}_2(\boldsymbol{x}_i), ..., \mathrm{F}_K(\boldsymbol{x}_i)\right),$$

The training error is given by is

$$\mathrm{E}\left[\mathrm{F}(\boldsymbol{x})\right] = \frac{1}{N}\sum_{i=1}^{N} 1_{F(\boldsymbol{x}_i) \neq y_i},$$