Fundamentals of Artificial Intelligence

COMP307/AIML420

Ensembles

Dr. Heitor Murilo Gomes
heitor.gomes@vuw.ac.nz
http://www.heitorgomes.com

Information

- Check Assignment 1 (due on week 5 - 27 March 2024)

- Helpdesks as available daily (Monday to Friday, 3pm to 4pm) on **CO242B**

- **Online helpdesks** (reach out to organize one)

- **Office hours** (send me an email to arrange a time)

Outline

- Introduction
- Diversity, Combination and Base learners
- Algorithms
  - Bagging
  - Random Forest
- Other considerations

Introduction

Why ensembles?

Instead of using a single learner, we train several learners and combine them… **why?**

Intuition

3 classifiers, each one of them is **correct 6 out of 10**
What happens when we combine their votes?

Ground-truth	c1	c2	c3
X	X	O	X
X	X	O	X
X	X	O	X
X	X	X	O
X	X	X	O
O	O	O	X
O	X	X	X
O	X	O	O
O	X	O	O
O	X	O	O

Intuition

3 classifiers, each one of them is **correct 6 out of 10**
What happens when we combine their votes?

Ground-truth	c1	c2	c3	Maj Vote
X	X	O	X	X
X	X	O	X	X
X	X	O	X	X
X	X	X	O	X
X	X	X	O	X
O	O	O	X	O
O	X	X	X	X
O	X	O	O	O
O	X	O	O	O
O	X	O	O	O

Why ensembles?

Better predictive performance

Intuitively, an ensemble **should outperform** any of its members individually in terms of **predictive performance** (e.g. accuracy)

There are three main [Kuncheva 2014] reasons to use ensembles: **Statistical**, **Computational** and **Representational**

Kuncheva, Ludmila I. Combining pattern classifiers: methods and algorithms. John Wiley & Sons, 2014.

Statistical

The **generalization capabilities of each classifier may be different** when they are applied to a test set, so their accuracy will vary

It is **safer** to use the **mean of individual predictions** from these classifiers **instead of using only one of them**, since the chance of selecting the classifier with the worst generalization capabilities is eliminated

*There is a chance that the **ensemble accuracy is worst than** that of the **best of its members***

Kuncheva, Ludmila I. Combining pattern classifiers: methods and algorithms. John Wiley & Sons, 2014.

Computational

(a) Imperfect training algorithm

Some classifiers may converge to a local minimum. Suppose that the local minima of *L* classifiers are close to the absolute minimum.

Such that, there is a way of combining them in a model even closer to the absolute minimum (optimal classifier) than any of them is capable individually.

Kuncheva, Ludmila I. Combining pattern classifiers: methods and algorithms. John Wiley & Sons, 2014.

Computational

(b) Too much data

Train many classifiers on small “subsets” of the data and aggregate their outputs.

(c) Too little data

There is just not enough data to build a “strong” classifier.

We can resort to **bootstrapping** to generate several subsets of data, train a classifier on each of them and obtain a better (combined) learner

Kuncheva, Ludmila I. Combining pattern classifiers: methods and algorithms. John Wiley & Sons, 2014.

Computational

(d) Divide and conquer

The classification problem might be easier to solve if the data is split into smaller and easier-to-handle problems

A classifier is trained on each of these subsets and used for predictions if the instance falls on its "domain"

This is related to ***model selection***

Kuncheva, Ludmila I. Combining pattern classifiers: methods and algorithms. John Wiley & Sons, 2014.

Computational

(e) Data fusion

The data may come from several different sources.

Example: to provide an improved product recommendation the system might have access to items that you previously bought, ads that you clicked, your friends' political views, places that you visited, audio recordings of conversations with your mom, and so on.

This is related to ***multi-view learning***

Kuncheva, Ludmila I. Combining pattern classifiers: methods and algorithms. John Wiley & Sons, 2014.

Representational

Several simple classifiers can approximate complex classification boundaries.

Kuncheva, Ludmila I. Combining pattern classifiers: methods and algorithms. John Wiley & Sons, 2014.

Another example…

What happens when we combine their votes this time?

Ground-truth	c1	c2	c3
X	O	O	O
X	O	O	O
X	O	O	O
X	X	X	X
X	X	X	X
O	O	O	O
O	X	X	X
O	O	O	O
O	O	O	O
O	O	O	O

Another example…

What happens when we combine their votes this time?

Not a very interesting result…

Ground-truth	c1	c2	c3	Maj Vote
X	O	O	O	O
X	O	O	O	O
X	O	O	O	O
X	X	X	X	X
X	X	X	X	X
O	O	O	O	O
O	X	X	X	X
O	O	O	O	O
O	O	O	O	O
O	O	O	O	O

Diversity, Combination & Base learner

Diversity, Combination & Base learner

Diversity: learners have to be “diverse” w.r.t their predictions

Combination: must obfuscate incorrect predictions and highlight correct predictions

Base learner: must be somewhat accurate

Inducing Diversity

Train on different instances: Bagging [1]

Train on different features: Random Subspaces [2]

Both: Random Forest [3] and Random Patches [4]

Heterogeneous learners

Combine different learners like decision trees, kNN, perceptrons, …

[1] Breiman, Leo. "Bagging predictors." *Machine learning*, Springer, 1996.

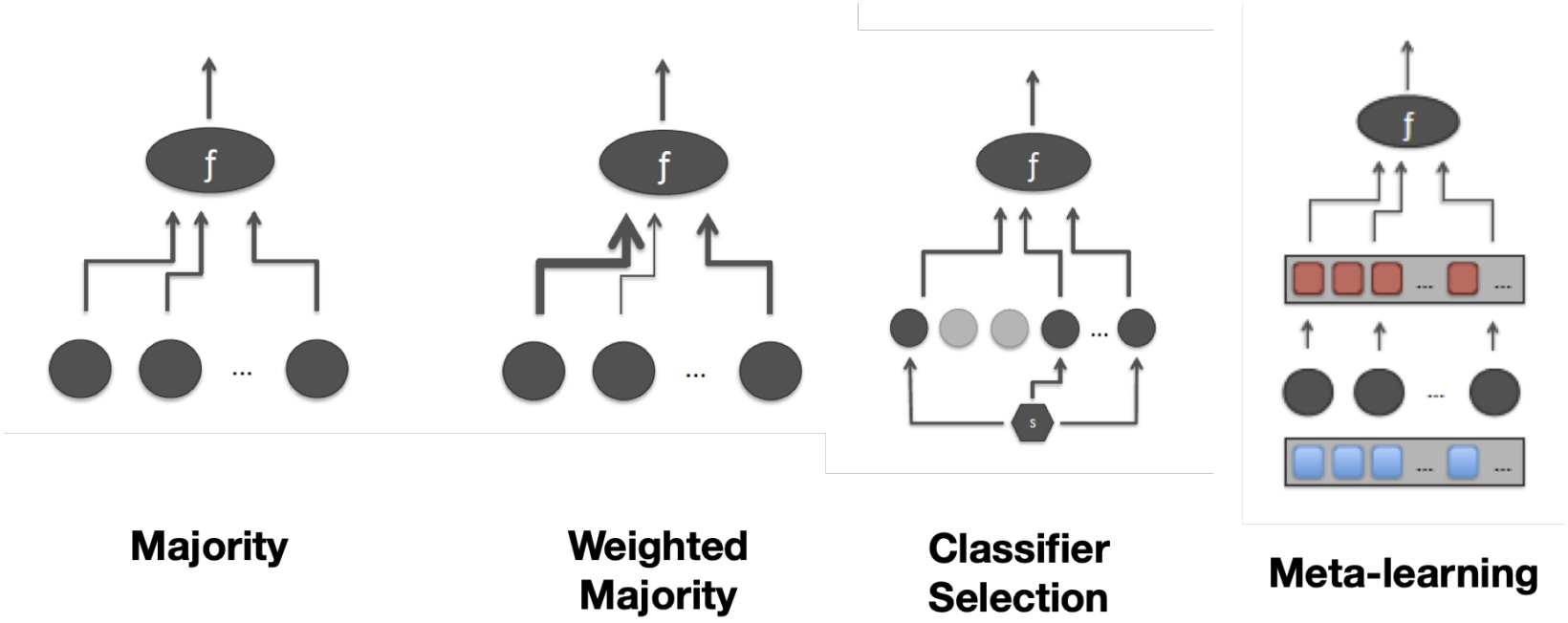
[2] Ho, Tin Kam. "The random subspace method for constructing decision forests." *IEEE TPAMI*, 1998

[3] Breiman, Leo. "Random forests." *Machine learning*, Springer, 2001

[4] Louppe, Gilles, and Pierre Geurts. "Ensembles on random patches." *ECML-PKDD 2012*

Combination

Defines how individual predictions are combined to form the ensemble prediction



Adapted from [5]

[5] *Gomes, Heitor M., et al. "A survey on ensemble learning for data stream classification." ACM Computing Surveys (CSUR), 2017*

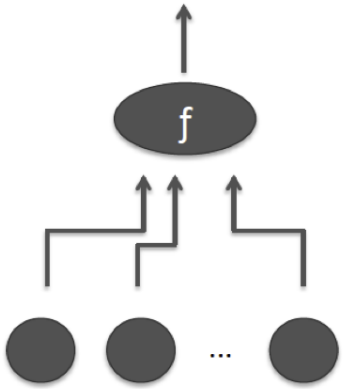
Combination

Majority vote: choose whichever label is predicted by more than half of the learners

Weighted majority vote: adds more weight on learners according to a weighting function (usually based on the predictive performance)

Classifier selection: a function S maps input data to specific learners (others are not allowed to vote)

Meta-Learner: Train a second-layer learner using the predictions of the first-layer learners (stacking)



Majority

Adapted from [5]

[5] *Gomes, Heitor M., et al. "A survey on ensemble learning for data stream classification." ACM Computing Surveys (CSUR), 2017*

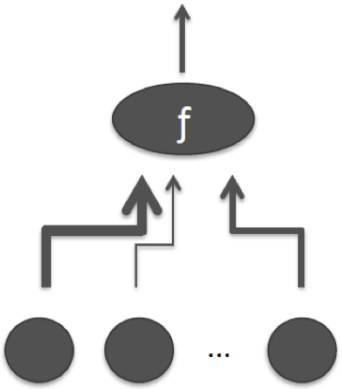
Combination

Majority vote: choose whichever label is predicted by more than half of the learners

Weighted majority vote: adds more weight on learners according to a weighting function (usually based on the predictive performance)

Classifier selection: a function S maps input data to specific learners (others are not allowed to vote)

Meta-Learner: Train a second-layer learner using the predictions of the first-layer learners (stacking)



Weighted Majority

Adapted from [5]

[5] *Gomes, Heitor M., et al. "A survey on ensemble learning for data stream classification." ACM Computing Surveys (CSUR), 2017*

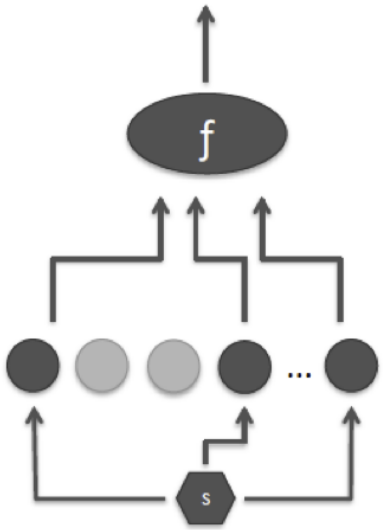
Combination

Majority vote: choose whichever label is predicted by more than half of the learners

Weighted majority vote: adds more weight on learners according to a weighting function (usually based on the predictive performance)

Classifier selection: a function S maps input data to specific learners (others are not allowed to vote)

Meta-Learner: Train a second-layer learner using the predictions of the first-layer learners (stacking)



Classifier Selection

Adapted from [5]

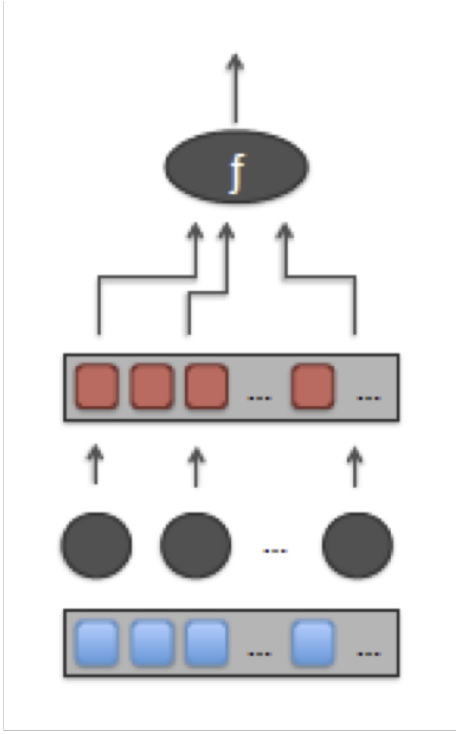
[5] *Gomes, Heitor M., et al. "A survey on ensemble learning for data stream classification." ACM Computing Surveys (CSUR), 2017*

Combination

Majority vote: choose whichever label is predicted by more than half of the learners

Weighted majority vote: adds more weight on learners according to a weighting function (usually based on the predictive performance)

Classifier selection: a function S maps input data to specific learners (others are not allowed to vote)



Meta-learning

Meta-Learner: Train a second-layer learner using the predictions of the first-layer learners (stacking)

Adapted from [5]

[5] *Gomes, Heitor M., et al. "A survey on ensemble learning for data stream classification." ACM Computing Surveys (CSUR), 2017*

Base learner

Individual accuracy: at least better than random

Dependent / Independent: the training of one learner depend on the other learners

“Unstable”: given small perturbations the learning algorithm must generate different models

Algorithms

Bagging

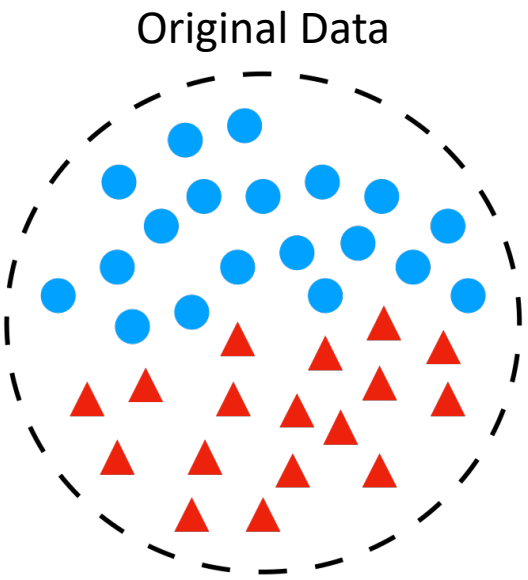
Bootstrap **Agg**regat**ing**

Bagging trains each model of the ensemble with a bootstrap sample from the original dataset

Every bootstrap contains each original sample **K** times, where the probability **P(K=k)** follows a binomial distribution

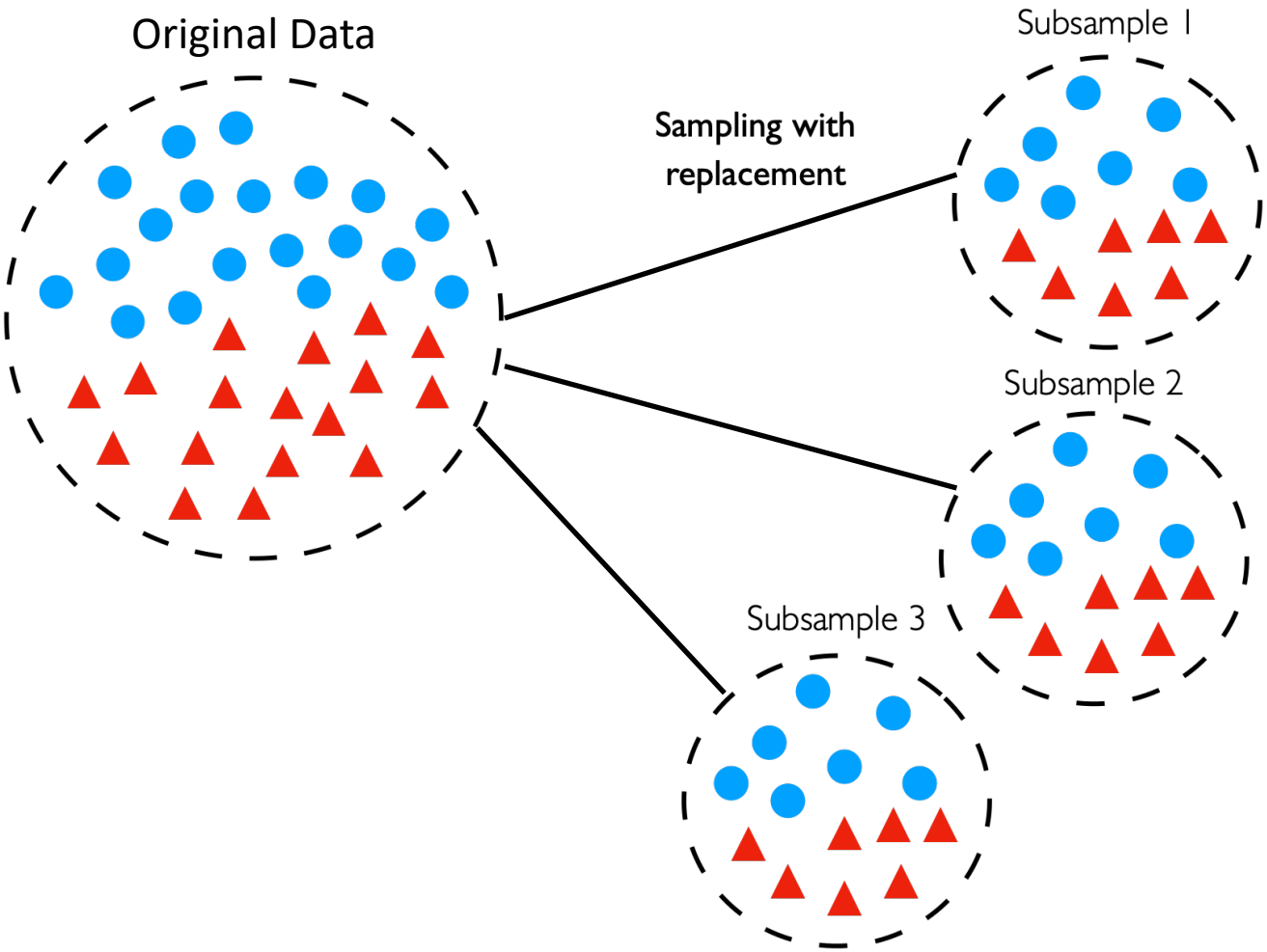
Breiman, Leo. "Bagging predictors." *Machine learning*, Springer, 1996.

Bagging



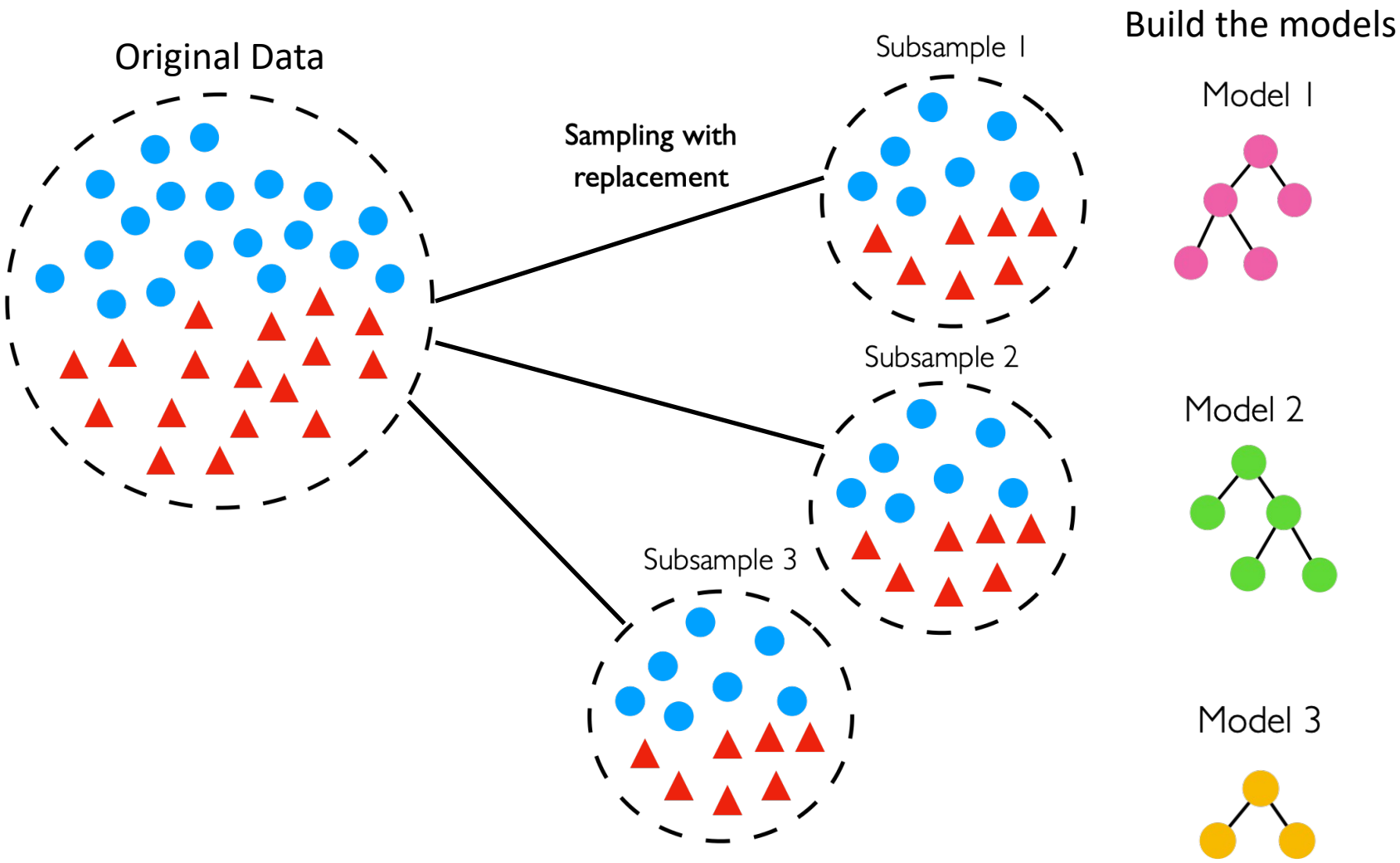
Breiman, Leo. "Bagging predictors." *Machine learning*, Springer, 1996.

Bagging



Breiman, Leo. "Bagging predictors." *Machine learning*, Springer, 1996.

Bagging



Breiman, Leo. "Bagging predictors." *Machine learning*, Springer, 1996.

Bagging: subsamples

On average for each subsample:

~63.2% of the instances are from the original dataset

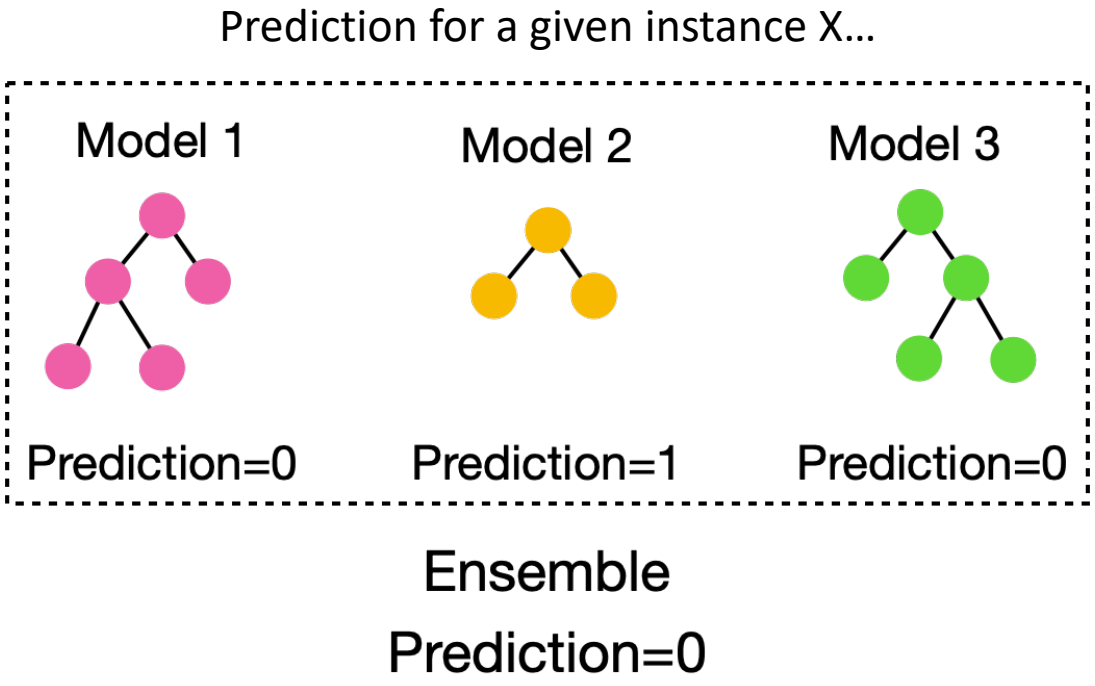
~36.8% are repeated instances

~36.8% of the original instances are not present*

* Out-Of-Bag (OOB)

Bagging: predicting

The predictions of each learner are aggregated using **majority vote** to obtain the final prediction.

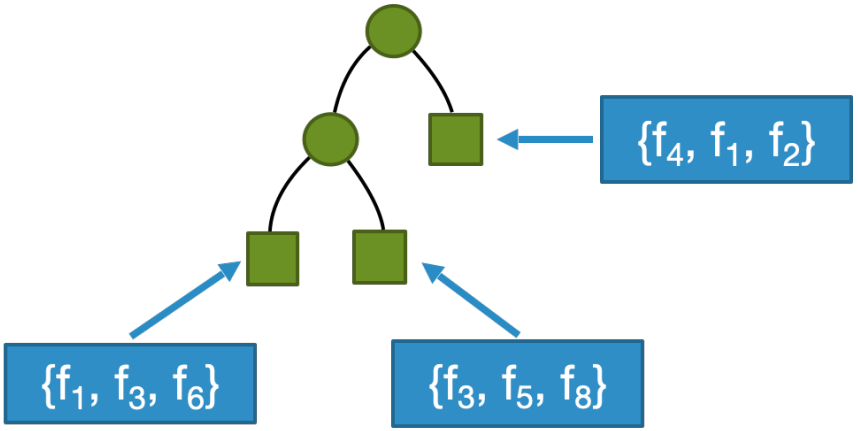


Randomizing the feature set

Bagging train base learners on random subsets of instances, but we can achieve higher diversity if we also randomize the feature set

Local randomization

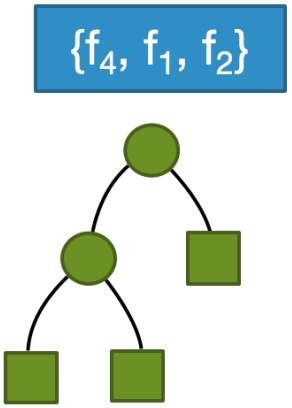
Random Forest



Global randomization

Random Subspaces

Random Patches

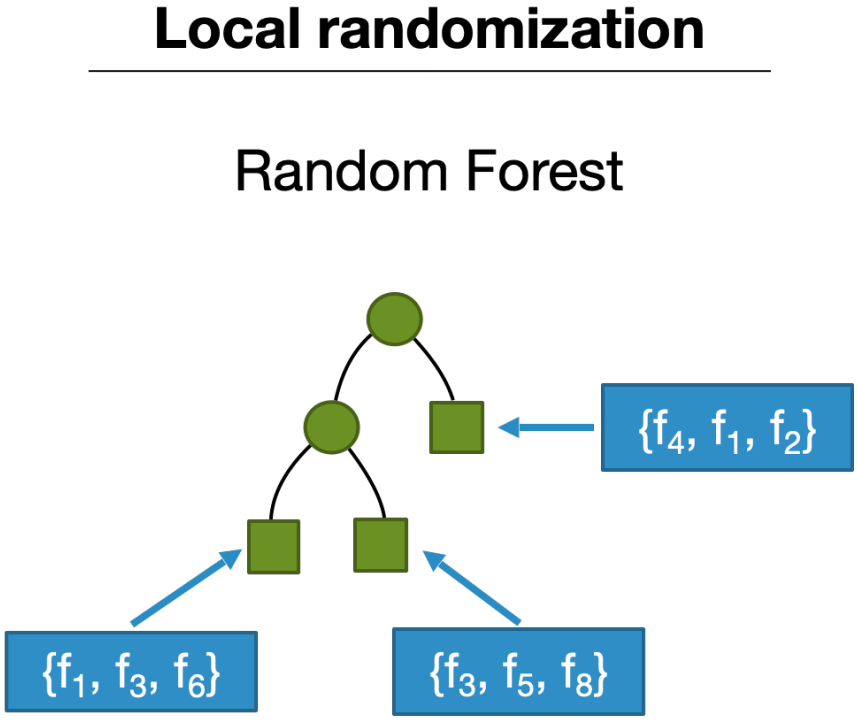


Random Forest

Adds **extra randomization** on top of **Bagging**

Base learner **must be** a **decision tree**

Split decisions are made using a **random subset of features** selected for each leaf (terminal tree node)



Random Forest

Number of trees & Subspace size. There are several hyperparameters on a RF, but the number of trees and subspace size are the most influential ones

Bias and Variance reduction. RFs are effective at reducing variance by introducing randomness during training and averaging predictions (combination), which also reduce the bias compared to single trees (see statistical reason)

OOB error estimation. Same as in Bagging, we can estimate the generalization error without a separate set

Liu, Brian, and Rahul Mazumder. "Randomization Can Reduce Both Bias and Variance: A Case Study in Random Forests." arXiv preprint arXiv:2402.12668 (February 2024)

Other Considerations

- **For Most ensembles**
  - training is costly, but predicting can be costly as well
  - We can alleviate that by leveraging parallel implementations
- **Random Subspaces and Random Patches**
  - Share some characteristics with RF, but are not limited to decision trees
- **Boosting and Gradient Boosting Machines (GBMs)**
  - Sequential training: we train weak learners in sequence
  - Focus on misclassified instances: assign higher weigths to misclassified instance to ensure subsequent models focus more on correcting these
  - Gradient descent optimization: (GBMs) use it to minimize a loss function iteratively
  - GBMs are the basis for XGBoosting and other efficient and accurate algorithms

Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of computer and system sciences*, 1997

Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." *Annals of statistics*, 2001

Chen, T., & Guestrin, C. “Xgboost: A scalable tree boosting system”. ACM KDD, 2016

Wrap up

- **Ensembles** are powerful methods to improve predictive performance using several “weak learners”
- **Diversity** is a must, but the **base learner** and **combination** are important too
- **Bagging and RF** are simple, yet powerful ensemble methods
- See chapter 19.8 from our textbook for more

Coming up next…

- Clustering (next lecture)
- Ensemble and clustering examples (Tutorial this week)

Russell, Stuart J., and Peter Norvig. Artificial intelligence a modern approach. 4th edition