Bagging

Random Forest

@cambridgespark

CAMBRIDGE SPARK

Bagging

Intuition

CAMBRIDGE SPARK

Ensembles: Why do we care?

- Good performance
- General purpose algorithms
- Usually easier to train than other fancy techniques
- Really popular in industry and ML competitions

CAMBRIDGE SPARK

What are ensemble models?

- Combining multiple simple models into a larger one
- Popular techniques:
  - **Bagging (this module)**
  - Boosting
  - Stacking

CAMBRIDGE SPARK

Intuition

Accuracy = 75%

Accuracy = 80%

Both say you have **X**… how confident are you about the diagnosis?

CAMBRIDGE SPARK

Intuition

Basic Idea:

1. We consult multiple doctors instead of one.

Intuition

Basic Idea:

1. We consult multiple doctors instead of one.
2. We combine their opinion to get a better diagnosis

Intuition

Basic Idea:

1. We consult multiple doctors instead of one.
2. We combine their opinion to get a better diagnosis

Does consulting more doctors **always** improve the diagnosis?

CAMBRIDGE SPARK

Intuition

Basic Idea:

1. We consult multiple doctors instead of one.
2. We combine their opinion to get a better diagnosis

Does consulting more doctors **always** improve the diagnosis?

- Doctors need to be **better than random** guessing if using majority vote
- Doctors need to be making their reasoning **independently** and **differently** so that they don't make identical mistakes

Formalising ensembles (a bit)

- Set of all observable symptoms: $x = (x_1 , . . . , x_p)$
- Set of consulted doctors: $D = \{ h_1, . . . , h_T \}$
- Each doctor is a "function" returning a diagnosis: $h_i (x)$
- Final diagnosis: $H(h_1 (x), . . . , h_T (x))$

The aggregating function H can be based on a majority vote .

Weights can be applied within H to take doctors' accuracies into account.

Bagging

In Practice

CAMBRIDGE SPARK

Back to supervised learning

- **Weak Learner:** a model with accuracy better than random guess
- **Diverse learners :** models that make mistakes on different data points

Aggregating weak and diverse learners leads to a model that can significantly outperform the weak learners.

Extremely powerful and successful both in classification and regression.

Bagging in practice

Take the context of classification for now, we want to:

- Train a set of diverse base classifiers: $\{h_1(\cdot), \ldots, h_T(\cdot)\}$
    - what classifiers?
    - how to get diverse classifiers?
- Aggregate the output of the classifiers: $H(h_1(\cdot), \ldots, h_T(\cdot))$
    - how to aggregate?

CAMBRIDGE SPARK

Bagging - Bootstrap Aggregating

For a class ***h*** of models (e.g Decision Tree):

Bagging - Bootstrap Aggregating

For a class ***h*** of models (e.g Decision Tree):

1. Randomly sample T datasets with replacement from the original one
    a. Bootstrapping

Bagging - Bootstrap Aggregating

For a class ***h*** of models (e.g Decision Tree):

1. Randomly sample T datasets with replacement from the original one
   a. Bootstrapping
2. Train T models on the bootstrap samples

Bagging - Bootstrap Aggregating

For a class ***h*** of models (e.g Decision Tree):

1. Randomly sample T datasets with replacement from the original one
    a. Bootstrapping
2. Train T models on the bootstrap samples
3. Aggregate their output

Bagging - Bootstrap Aggregating

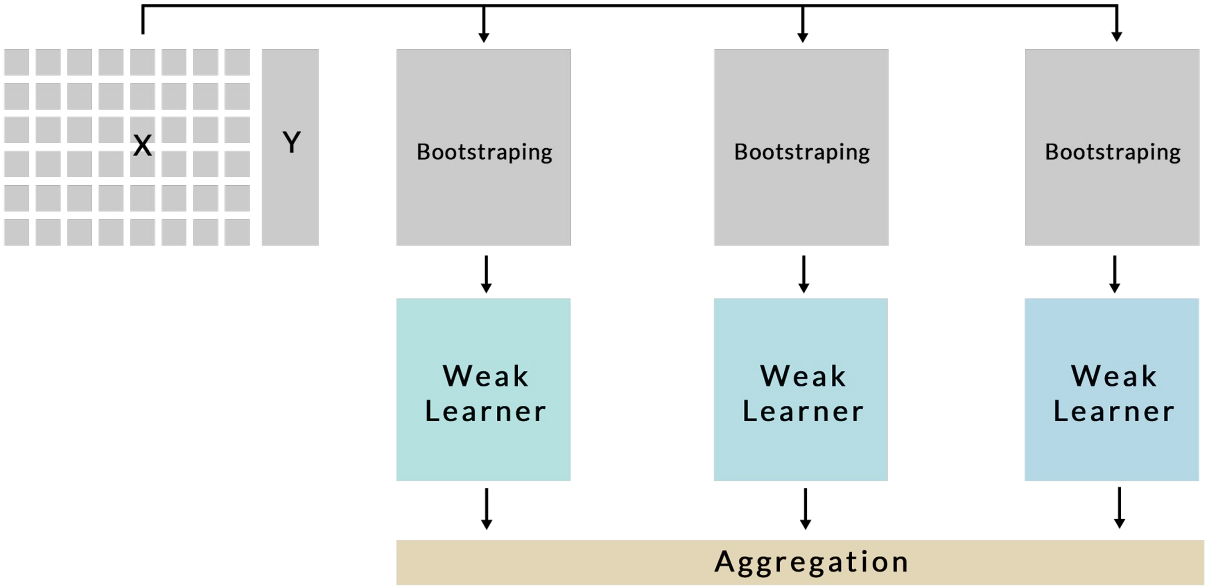
For a class ***h*** of models (e.g Decision Tree):

1. Randomly sample T datasets with replacement from the original one
   a. Bootstrapping
2. Train T models on the bootstrap samples
3. Aggregate their output

Bootstrap samples are "statistically similar". Some points may not appear, some may appear several times.

CAMBRIDGE SPARK

Bagging - Bootstrap Aggregating



CAMBRIDGE SPARK

Comments on Bagging

Works well with unstable models (significant difference in model with slightly different data)

- Decision tree = unstable model (high variance)
- Logistic regression = stable model

Bagging: pay attention to rare points

Consider a dataset with a few “rare” data-points that appear, e.g.: with 1/100. Then many models will be trained without this rare points.

Bagging: pay attention to rare points

Consider a dataset with a few “rare” data-points that appear, e.g.: with 1/100. Then many models will be trained without this rare points.

- Bagging leads to estimator that typically perform (very) well on the bulk of the population but may do quite badly on outliers

CAMBRIDGE SPARK

Bagging: pay attention to rare points

Consider a dataset with a few “rare” data-points that appear, e.g.: with 1/100. Then many models will be trained without this rare points.

- Bagging leads to estimator that typically perform (very) well on the bulk of the population but may do quite badly on outliers
- This may be a good thing but it ≈ amounts to implicitly ignoring outliers → keep this in mind.

CAMBRIDGE SPARK

Random Forest

CAMBRIDGE SPARK

How do we build **diverse** trees?

- Each one is trained on a subsample of **observations** [Bootstrapping]

CAMBRIDGE SPARK

How do we build **diverse** trees?

- Each one is trained on a subsample of **observations** [Bootstrapping]
- Each one is trained on a subsample of **features** [Features subsampling]

How do we build **diverse** trees?

- Each one is trained on a subsample of **observations** [Bootstrapping]
- Each one is trained on a subsample of **features** [Features subsampling]
- Loosen your constraints to let your trees overfit

CAMBRIDGE SPARK

How do we build **diverse** trees?

- Each one is trained on a subsample of **observations** [Bootstrapping]
- Each one is trained on a subsample of **features** [Features subsampling]
- Loosen your constraints to let your trees overfit

Don't overdo it… We still need:

- Good performance per tree (no underfitting)

CAMBRIDGE SPARK

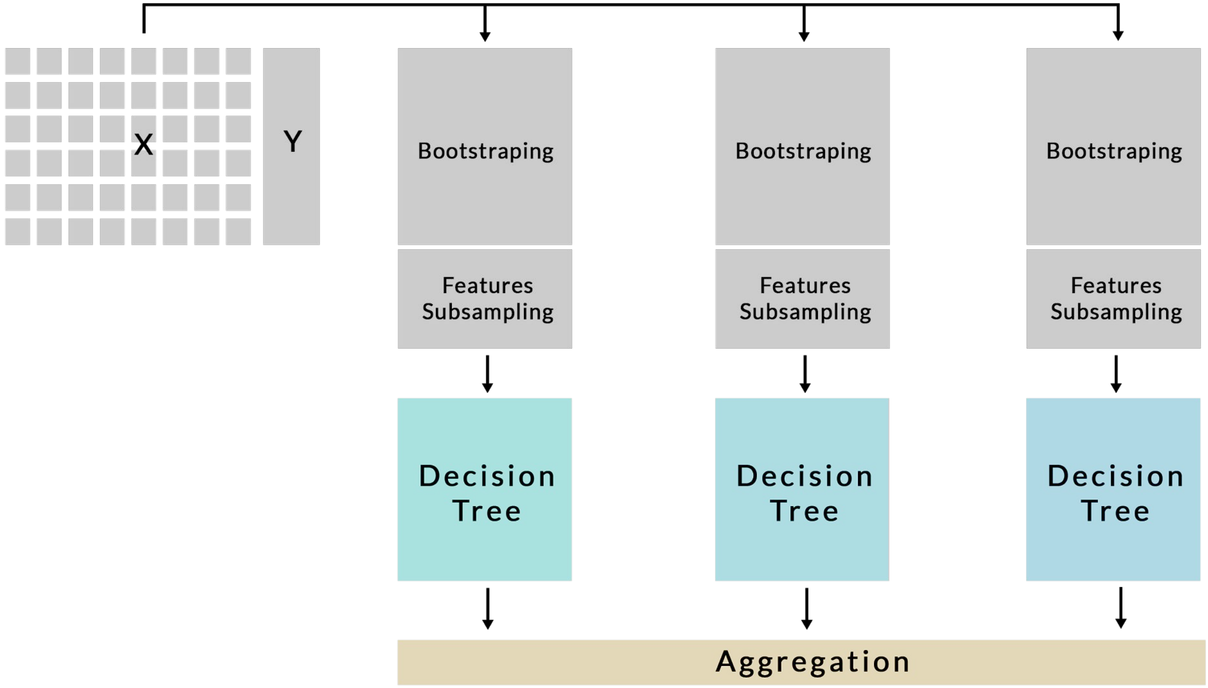
How do we build **diverse** trees?

- Each one is trained on a subsample of **observations** [Bootstrapping]
- Each one is trained on a subsample of **features** [Features subsampling]
- Loosen your constraints to let your trees overfit

Don’t overdo it… We still need:

- Good performance per tree (no underfitting)
- Able to generalise (no overfitting)

Bagging - Random Forest



CAMBRIDGE SPARK

Some pros and cons

+ Easy to run in **parallel**

CAMBRIDGE SPARK

Some pros and cons

+ Easy to run in **parallel**

+ Decision Trees = we can get **feature importance**

Some pros and cons

+ Easy to run in **parallel**
+ Decision Trees = we can get **feature importance**
− Models remain **correlated** (similar data)

Some pros and cons

+ Easy to run in **parallel**
+ Decision Trees = we can get **feature importance**
− Models remain **correlated** (similar data)
− Hard to **interpret**

CAMBRIDGE SPARK

Some pros and cons

+ Easy to run in **parallel**

+ Decision Trees = we can get **feature importance**

– Models remain **correlated** (similar data)

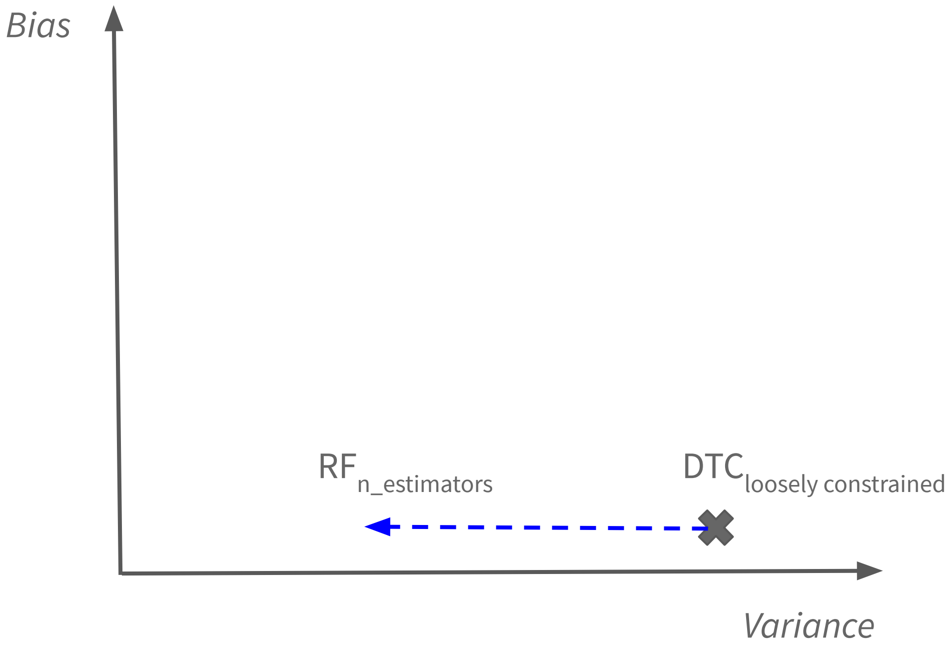
– Hard to **interpret**

? **Outliers** likely to be ignored by most weak learners

Bagging in SkLearn

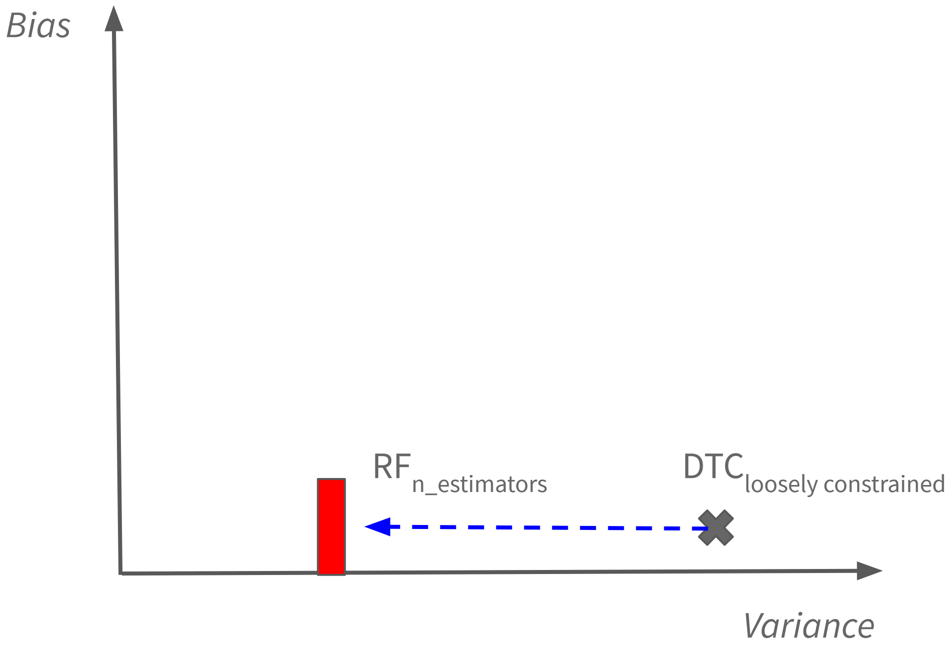
- BaggingClassifier, BaggingRegressor
- RandomForestClassifier, RandomForestRegressor
- ExtraTreesClassifier, ExtraTreesRegressor

Bias vs Variance



- Random Forest allows to reduce variance by ensembling DTs
- What happens if we keep adding trees?

Bias vs Variance



- Random Forest allows to reduce variance by ensembling DTs
- What happens if we keep adding trees?
    - We'll just reach a maximum performance and stop improving

Hands-on session

01-bagging_random_forest.ipynb