Applied Machine Learning

Random Forests

Kevyn Collins-Thompson

Associate Professor of Information & Computer Science
University of Michigan

Random Forests

- **An ensemble of trees, not just one tree.**
- **Widely used, very good results on many problems.**
- **`sklearn.ensemble` module:**
  - *Classification:* `RandomForestClassifier`
  - *Regression:* `RandomForestRegressor`
- **One decision tree → Prone to overfitting.**
- **Many decision trees → More stable, better generalization**
- **Ensemble of trees should be diverse: introduce random variation into tree-building.**

Random Forest Process

Original dataset

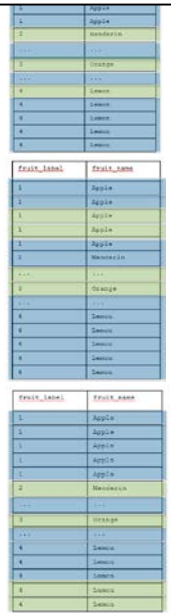
Randomized bootstrap copies

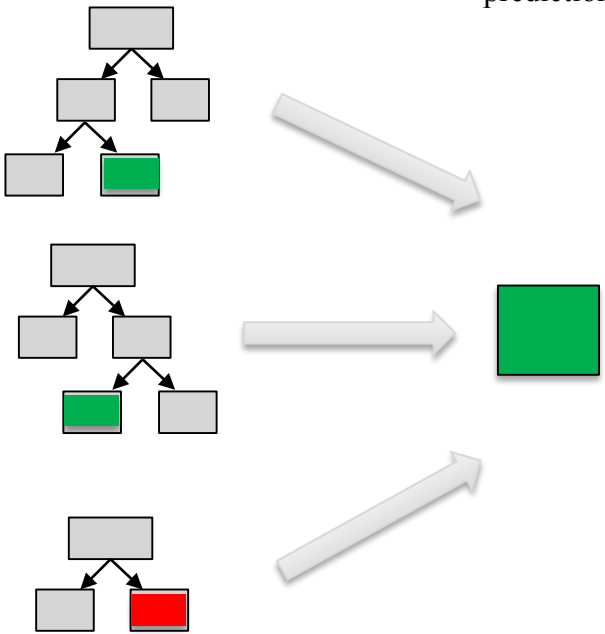
n_estimator

Randomized feature splits

Ensemble prediction

fruit_label	fruit_name
1	Apple
1	Apple
1	Apple
1	Apple
1	Apple
2	Mandarin
...	...
3	Orange
...	...
4	Lemon
4	Lemon
4	Lemon
4	Lemon
4	Lemon





Random Forest Process: Bootstrap Samples

Bootstrap sample 1

fruit_label	fruit_name
1	Apple
1	Apple
1	Apple
1	Apple
1	Apple
2	Mandarin
...	...
3	Orange
...	...
4	Lemon
4	Lemon
4	Lemon
4	Lemon
4	Lemon

Bootstrap sample 2

fruit_label	fruit_name
1	Apple
1	Apple
1	Apple
1	Apple
1	Apple
2	Mandarin
...	...
3	Orange
...	...
4	Lemon
4	Lemon
4	Lemon
4	Lemon
4	Lemon

Bootstrap sample 3

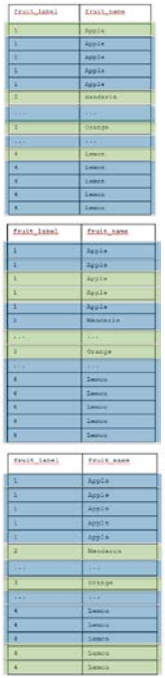
fruit_label	fruit_name
1	Apple
1	Apple
1	Apple
1	Apple
1	Apple
2	Mandarin
...	...
3	Orange
...	...
4	Lemon
4	Lemon
4	Lemon
4	Lemon
4	Lemon

Random Forest Process

Original dataset

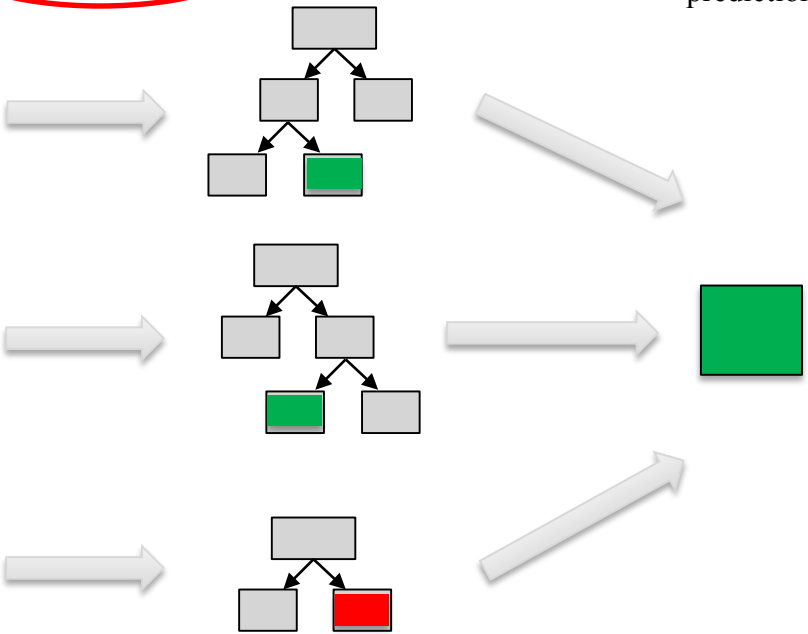
fruit_label	fruit_name
1	Apple
1	Apple
1	Apple
1	Apple
1	Apple
2	Mandarin
...	...
3	Orange
...	...
4	Lemon
4	Lemon
4	Lemon
4	Lemon
4	Lemon

Randomized bootstrap copies



Randomized feature splits

max_features



Ensemble prediction

Random Forest `max_features` Parameter

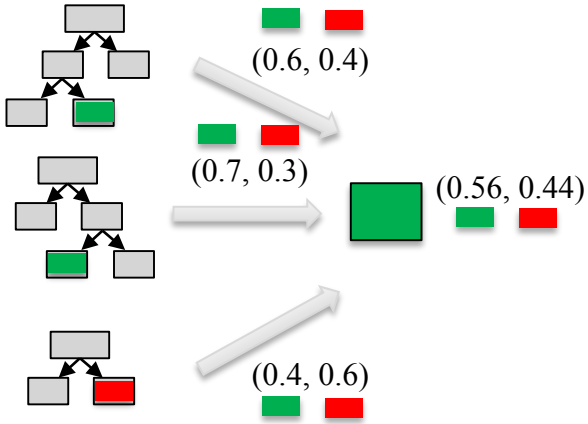
- Learning is quite sensitive to `max_features`.
- Setting `max_features = 1` leads to forests with diverse, more complex trees.
- Setting `max_features = <close to number of features>` will lead to similar forests with simpler trees.

Prediction Using Random Forests

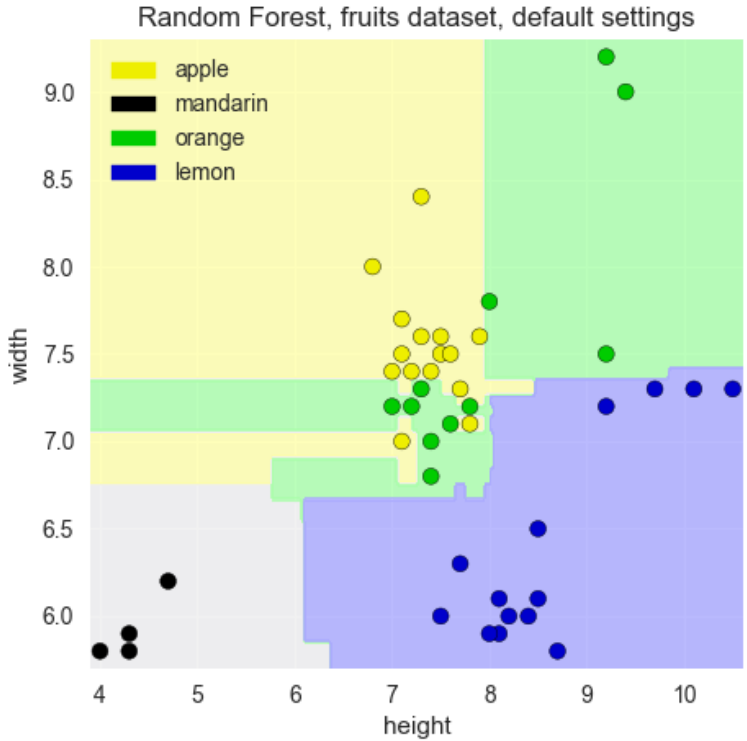
1. Make a prediction for every tree in the forest.

2. Combine individual predictions

- *Regression: mean of individual tree predictions.*
- *Classification:*
  - *Each tree gives probability for each class.*
  - *Probabilities averaged across trees.*
  - *Predict the class with highest probability.*



Random Forest: Fruit Dataset



Random Forest: Pros and Cons

Pros:

- Widely used, excellent prediction performance on many problems.
- Doesn't require careful normalization of features or extensive parameter tuning.
- Like decision trees, handles a mixture of feature types.
- Easily parallelized across multiple CPUs.

Cons:

- The resulting models are often difficult for humans to interpret.
- Like decision trees, random forests may not be a good choice for very high-dimensional tasks (e.g. text classifiers) compared to fast, accurate linear models.

Random Forests: RandomForestClassifier Key Parameters

- `n_estimators`: number of trees to use in ensemble (default: 10).
  - *Should be larger for larger datasets to reduce overfitting (but uses more computation).*
- `max_features`: has a strong effect on performance. Influences the diversity of trees in the forest.
  - *Default works well in practice, but adjusting may lead to some further gains.*
- `max_depth`: controls the depth of each tree (default: None. Splits until all leaves are pure).
- `n_jobs`: How many cores to use in parallel during training.

- Choose a fixed setting for the random_state parameter if you need reproducible results.