#ODSC

Ensemble Models Demystified

@KevinLemagnen

CAMBRIDGE SPARK

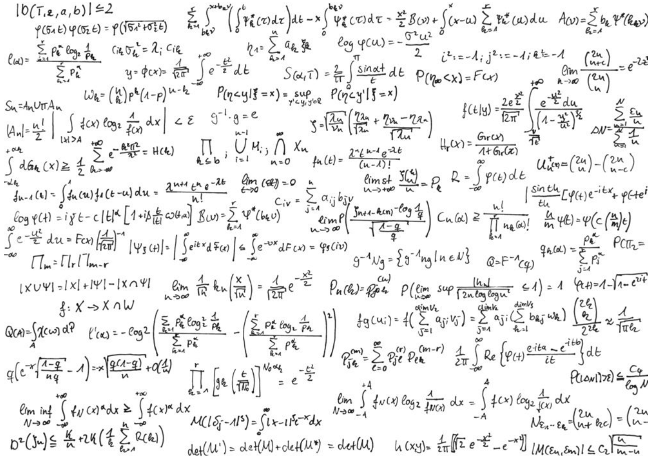
Ensembles: Why do we care?

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

CAMBRIDGE SPARK

Why this talk?

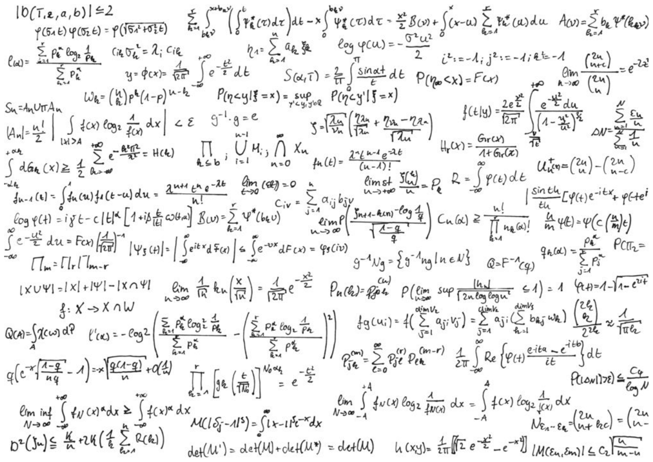
Theory



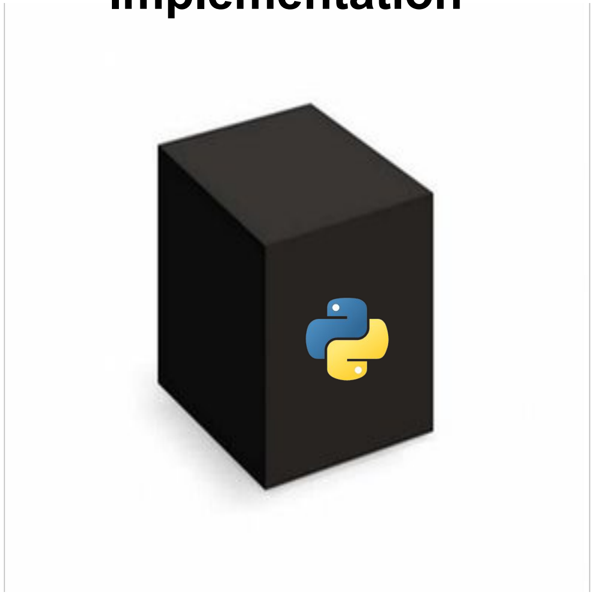
CAMBRIDGE SPARK

Why this talk?

Theory

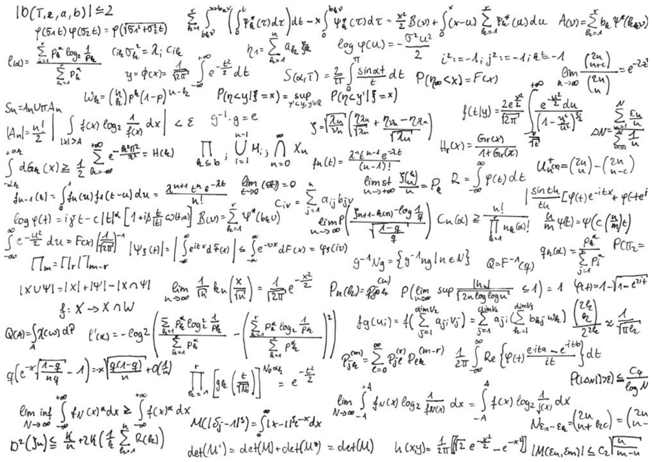


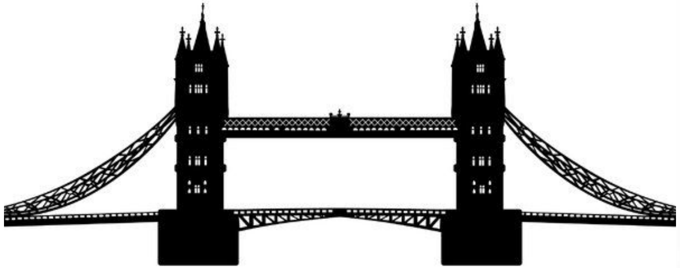
Implementation



Why this talk?

Theory





Implementation

Agenda

1. Intuition
2. Weak learner (Decision Tree)
3. Bagging (Random Forest)
4. Boosting (Gradient Boosting)
5. Other boosting libraries

CAMBRIDGE SPARK

1. Intuition

CAMBRIDGE SPARK

What are ensemble models?

- Combining multiple simple models (weak learners) into a larger one (ensemble)
- Two popular techniques:
    - Bagging
    - Boosting
    - Usually with decision trees as weak learner

CAMBRIDGE SPARK

Intuition

Accuracy = 60%

Accuracy = 75%

Both say you have **X**… what's the likelihood that you really have **X**?

CAMBRIDGE SPARK

Two big assumptions

- **Weak Learner:** “Experts” need to be more right than wrong on average

Two big assumptions

- **Weak Learner:** “Experts” need to be more right than wrong on average
- **Diversity**: “Experts” need to make different errors

2. Weak Learners

Decision Trees

CAMBRIDGE SPARK

Why are they good?

- Can capture complex relationships in the data
  - We’ll often be able to get our > 50% accuracy!
- Overfits easily
  - We can use that to create diverse models!

CAMBRIDGE SPARK

How do we control them?

No constraints = one leaf per sample = **massive overfitting**

Some good constraints:

- Pick a **maximum depth**
- Pick a **minimum number of samples** needed in a new node/leaf

3. Bagging

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**

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**
- 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**
- 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?

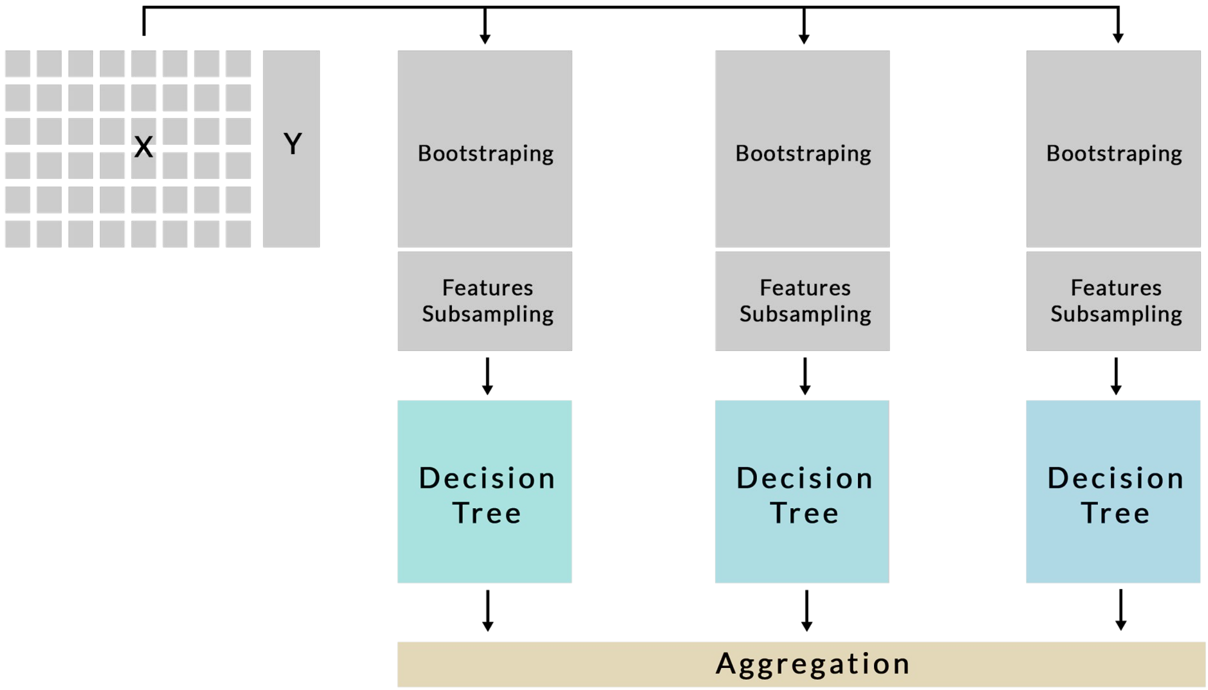
- Each one is trained on a subsample of **observations** [Bootstrapping]
- Each one is trained on a subsample of **features**
- Loosen your constraints to let your trees overfit

Don't overdo it… We still need:

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

CAMBRIDGE SPARK

Bagging - Random Forest



CAMBRIDGE SPARK

Some pros and cons

+ Easy to run in **parallel**

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)

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**

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

4. Boosting

Gradient Boosting

CAMBRIDGE SPARK

Boosting - Intuition

We want to build weak learners that **actively compensate** each others' errors.

Boosting - Intuition

We want to build weak learners that **actively compensate** each others' errors.

Let's focus on one sample: **(X, y)** with $y = 100$

Boosting - Intuition

We want to build weak learners that **actively compensate** each others' errors.

Let's focus on one sample: **(X, y)** with $y = 100$

1. Train DT1 on **(X**, **y)** $DT1(X) = 95$

CAMBRIDGE SPARK

Boosting - Intuition

We want to build weak learners that **actively compensate** each others' errors.

Let's focus on one sample: **(X, y)** with $y = 100$

1. Train DT1 on **(X**, **y)** $DT1(X) = 95$
2. Compute residuals $r = 100 - 95 = 5$

CAMBRIDGE SPARK

Boosting - Intuition

We want to build weak learners that **actively compensate** each others' errors.

Let's focus on one sample: **(X, y)** with $y = 100$

1. Train DT1 on **(X**, **y)** — $DT1(X) = 95$
2. Compute residuals — $r = 100 - 95 = 5$
3. Train DT2 on **(X, 5)** — $DT2(X) = 4$

Boosting - Intuition

We want to build weak learners that **actively compensate** each others' errors.

Let's focus on one sample: **(X, y)** with $y = 100$

1. Train DT1 on **(X, y)** — $DT1(X) = 95$
2. Compute residuals — $r = 100 - 95 = 5$
3. Train DT2 on **(X, 5)** — $DT2(X) = 4$
4. Aggregate DT1 and DT2 — $DT1(X) + DT2(X) = 95 + 4 = 99$

CAMBRIDGE SPARK

Boosting - Intuition

We want to build weak learners that **actively compensate** each others' errors.

Let's focus on one sample: **(X, y)** with $y = 100$

1. Train DT1 on **(X, y)** — $DT1(X) = 95$
2. Compute residuals — $r = 100 - 95 = 5$
3. Train DT2 on **(X, 5)** — $DT2(X) = 4$
4. Aggregate DT1 and DT2 — $DT1(X) + DT2(X) = 95 + 4 = 99$
5. Repeat

CAMBRIDGE SPARK

Boosting - Intuition

We want to build weak learners that **actively compensate** each others' errors.

Let's focus on one sample: **(X, y)** with $y = 100$

1. Train DT1 on **(X, y)** — $DT1(X) = 95$
2. Compute residuals — $r = 100 - 95 = 5$
3. Train DT2 on **(X, 5)** — $DT2(X) = 4$
4. Aggregate DT1 and DT2 — $DT1(X) + DT2(X) = 95 + 4 = 99$
5. Repeat

Weak learners increasingly focus on "hard points"

CAMBRIDGE SPARK

Boosting - Gradient Boosting

too many stages **OR** too complex trees = overfit to noise

CAMBRIDGE SPARK

Boosting - Gradient Boosting

too many stages **OR** too complex trees = overfit to noise

- Getting the number of stages right is **extremely** important

CAMBRIDGE SPARK

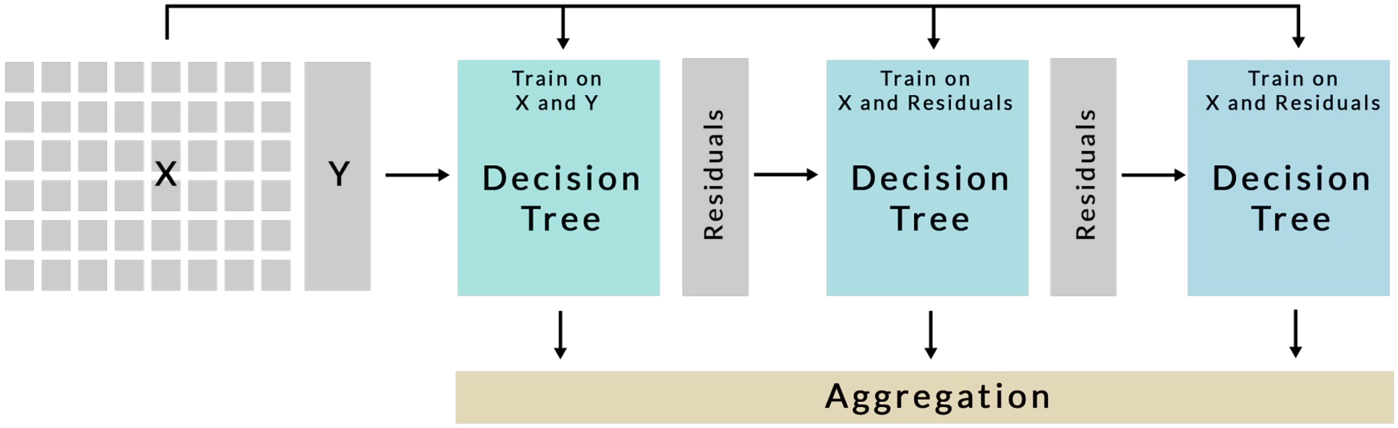
Boosting - Gradient Boosting

too many stages **OR** too complex trees = overfit to noise

- Getting the number of stages right is **extremely** important
- We need to build **small**, **constrained** trees

CAMBRIDGE SPARK

Boosting - Gradient Boosting



CAMBRIDGE SPARK

Some pros and cons

+ Great **performance** (usually)

Some pros and cons

+ Great **performance** (usually)

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

Some pros and cons

- \+ Great **performance** (usually)
- \+ Decision Trees = we can get **feature importance**
- − Hard to run in **parallel**

Some pros and cons

+ Great **performance** (usually)

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

– Hard to run in **parallel**

– Hard to **interpret**

Some pros and cons

- \+ Great **performance** (usually)
- \+ Decision Trees = we can get **feature importance**
- − Hard to run in **parallel**
- − Hard to **interpret**
- − Can easily **overfit**

CAMBRIDGE SPARK

Any questions?

CAMBRIDGE SPARK