AdaBoost

Xabier Etxezarreta

Mondragon
Unibertsitatea

Faculty of
Engineering

AdaBoost

1. Internals
2. Scikit-Learn
3. Example (Jupyter Notebook)
4. References

1

AdaBoost
Internals

Internals - What is AdaBoost?

- **AdaBoost** is a machine learning **meta-algorithm**

- **Boosting** originated from the question of whether a set of **weak classifiers** could be converted **to a strong classifier**.

- AdaBoost stands for **'Adaptive Boosting'** which transforms weak learners or predictors to **strong predictors** in order to solve problems of classification.

- Improves the **performance** of machine learning algorithms.
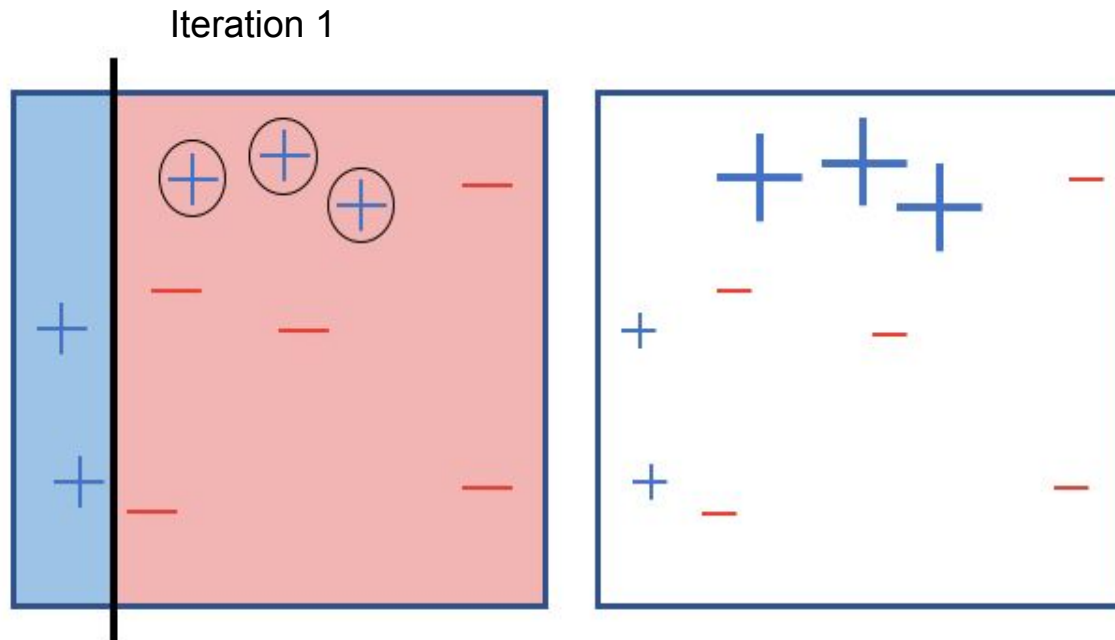
Internals - How AdaBoost works?
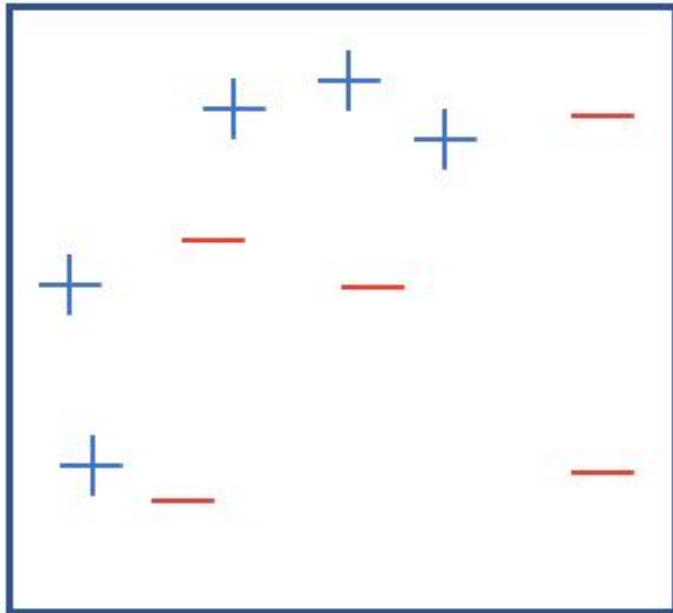
- Classification problem

- 10 instances
  - 5 positive
  - 5 negative

Internals - How AdaBoost works?

AdaBoost first iteration

- Blue zone (positive zone)
- Red zone (negative zone)

Incorrectly classified instances will have a **greater importance** in the next iteration.

Iteration 1

Internals - How AdaBoost works?

AdaBoost second iteration

- Blue zone (positive zone)
- Red zone (negative zone)

Incorrectly classified instances will have a **greater importance** in the next iteration.
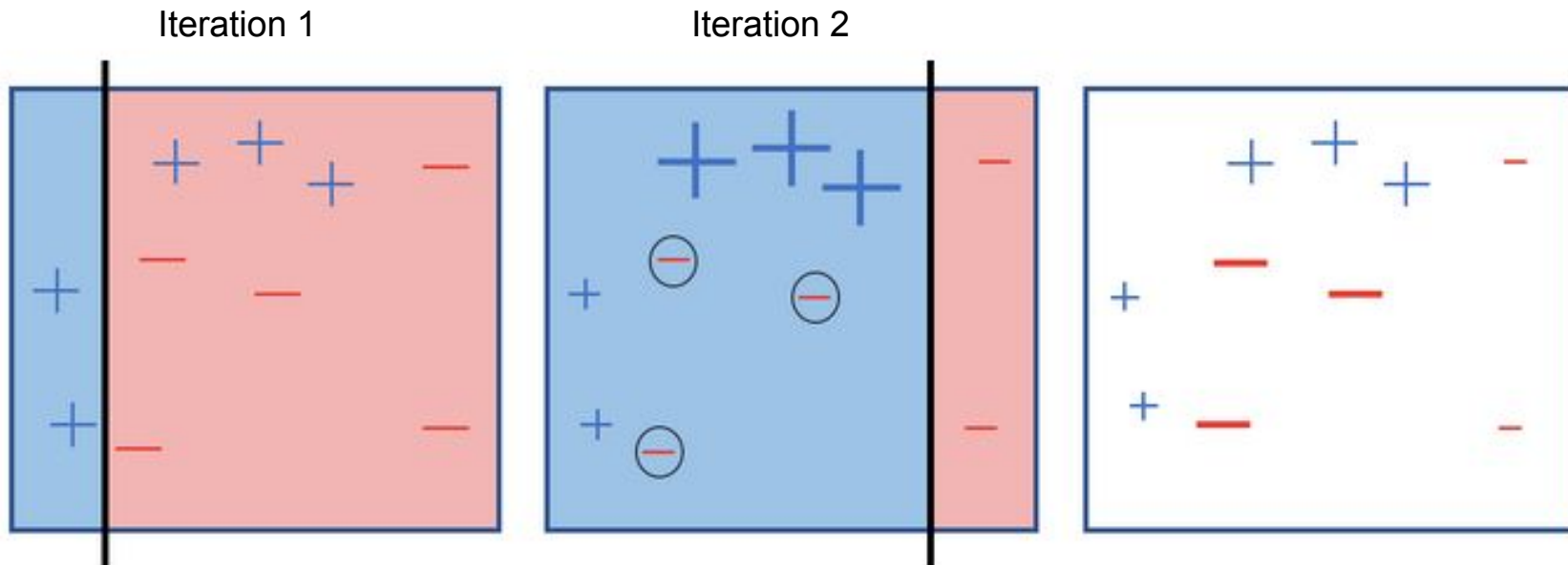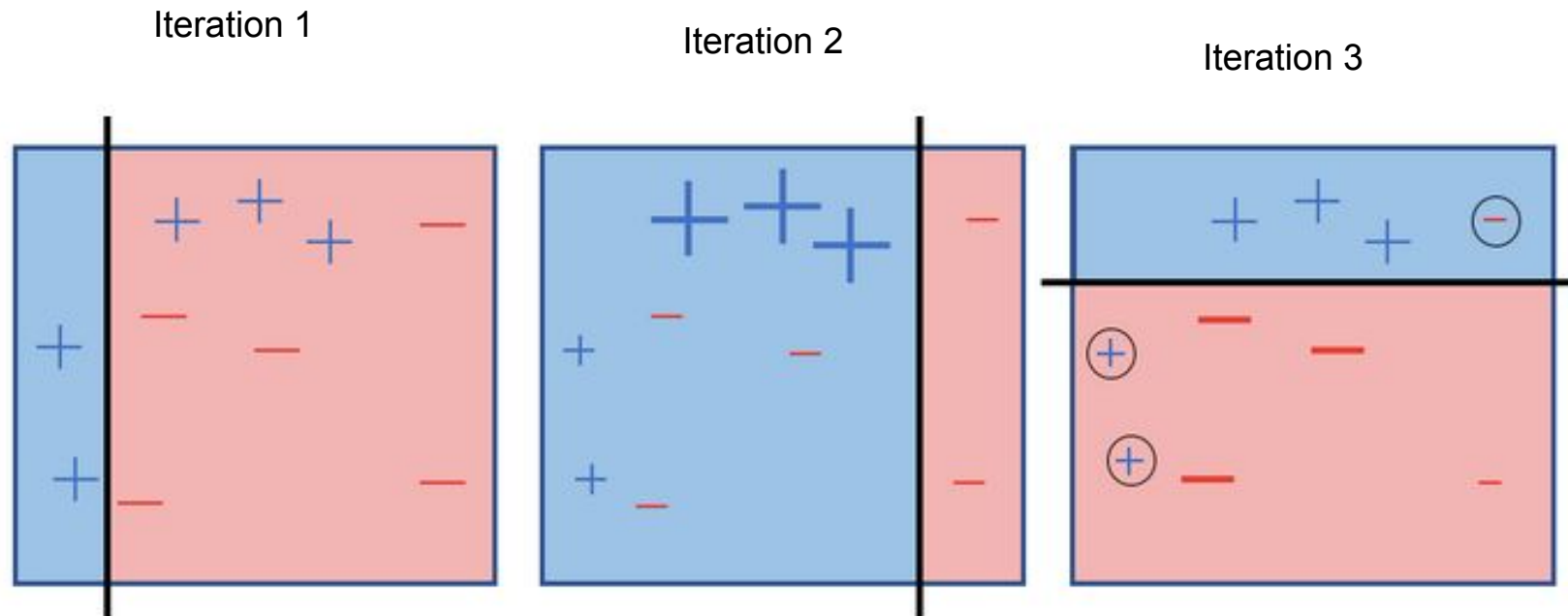

Iteration 1   Iteration 2
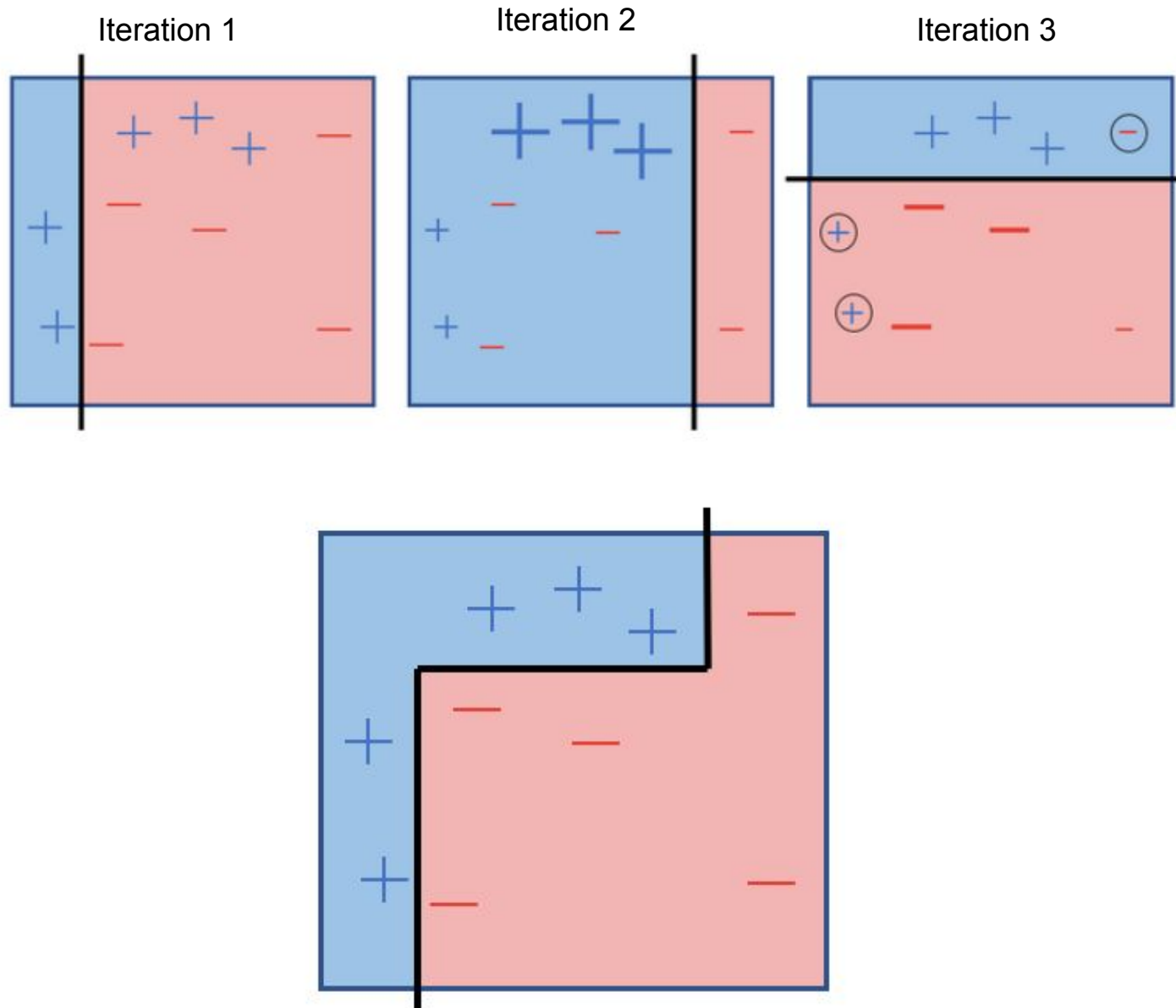
Internals - How AdaBoost works?

AdaBoost third iteration

- Blue zone (positive zone)
- Red zone (negative zone)

Iteration 1

Iteration 2

Iteration 3

Internals - How AdaBoost works?

Iteration 1

Iteration 2

Iteration 3

Internals - AdaBoost Pros and Cons

- **Pros**
  - Fast, simple and easy to program.
  - Flexibility to be combined with any algorithm.

- **Cons**
  - Boosting increases the overall complexity of your system
  - Weak classifiers being too weak can lead to low margins and overfitting.

2

AdaBoost
Scikit-Learn

Internals - AdaBoost Pros and Cons

sklearn.ensemble.AdaBoostClassifier

class `sklearn.ensemble.` **AdaBoostClassifier** (*base_estimator=None, n_estimators=50, learning_rate=1.0, algorithm='SAMME.R', random_state=None*)   [source]

base_estimator
  - The base estimator from which the boosted ensemble is built.
  - Base estimator → *DecisionTreeClassifier(max_depth=1)*

n_estimators
  - The maximum number of estimators at which boosting is terminated.
  - The number of weak learners is controlled by this parameter.
  - In case of perfect fit, the learning procedure is stopped early.

The **main parameters to tune** to obtain good results are **n_estimators** and the **complexity of the base estimators** (e.g., its depth max_depth or minimum required number of samples to consider a split min_samples_split).

3

AdaBoost
Example

Example - AdaBoost

4

References

References

- https://www.educba.com/adaboost-algorithm/

- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

- https://medium.com/soldai/ensamble-de-clasificadores-usando-adaboost-50bc2ca47640

- https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/

- http://rob.schapire.net/papers/explaining-adaboost.pdf

Eskerrik asko
Muchas gracias
Thank you

Xabier Etxezarreta
xabier.etxezarreta@alumni.mondragon.edu