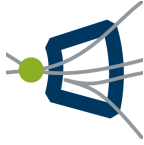


Data Analysis Block Course: Statistics and Machine learning



Exercise: AdaBoost and Gradient Boosting

Prof. Dr. Mikhail Mikhasenko

Assistants: Hendric Jonas, Christoph Tönnis



28 September 2026

[Indico schedule](#)

[RUB Moodle](#)

Learning objectives

By the end of this exercise, you should be able to:

- Explain what a **decision stump** is.
- Understand how a stump makes a simple classification.
- Explain why **AdaBoost assigns weights to training examples**.
- Calculate the error and weight of a weak learner in AdaBoost.
- Update patient weights after a weak learner makes predictions.
- Understand how several weak learners are combined into a strong classifier.
- Explain the basic idea behind **gradient boosting**.
- Understand how gradient boosting focuses subsequent models on the errors made by previous models.
- Distinguish the way **AdaBoost** and **gradient boosting** build ensembles.

Clinical scenario

A hospital wants to develop a simple machine-learning system to help diagnose **heart disease**. For each patient, four characteristics are recorded:

1. **Age** - patient's age in years.
2. **Body weight** - patient's weight in kg.
3. **Blocked arteries** - whether the patient has blocked arteries (Yes/No).
4. **Chest pain** - whether the patient experiences characteristic chest pain (Yes/No).

A cardiologist has already determined whether each patient actually has heart disease. This diagnosis will be treated as the **target variable**.

For this exercise:

- +1 = heart disease
- -1 = no heart disease

Important: This is a simplified educational dataset. The variables and values are fictional and are not intended to represent a real clinical diagnostic system.

The 10 patients

Patient	Age	Weight (kg)	Blocked arteries	Chest pain	Actual diagnosis
1	25	62	No	No	-1
2	35	72	No	Yes	+1
3	45	85	Yes	No	+1
4	50	95	No	Yes	+1
5	55	80	Yes	Yes	+1
6	60	90	Yes	No	+1
7	65	70	No	No	-1
8	40	100	No	No	-1
9	70	88	Yes	No	+1
10	30	110	No	No	-1

There are **6 patients with heart disease** and **4 without heart disease**.

Part I - Decision stumps

A **decision stump** is a decision tree with only **one split**.

For example:

If age > 47, predict heart disease; otherwise predict no heart disease.

This is a very simple model. It uses only **one feature and one threshold**.

For categorical variables, a stump might instead be:

If blocked arteries = Yes, predict heart disease; otherwise predict no heart disease.

Exercise 1.1 - Build some stumps

Consider the following possible stumps.

Stump A - Age

$$\hat{y} = \begin{cases} +1 & \text{Age} > 47 \\ -1 & \text{otherwise} \end{cases}$$

Stump B - Blocked arteries

$$\hat{y} = \begin{cases} +1 & \text{Blocked arteries} = \text{Yes} \\ -1 & \text{otherwise} \end{cases}$$

Stump C - Chest pain

$$\hat{y} = \begin{cases} +1 & \text{Chest pain} = \text{Yes} \\ -1 & \text{otherwise} \end{cases}$$

Questions

For each stump:

1. Predict the diagnosis for all 10 patients.
2. Count the number of incorrect predictions.
3. Calculate the classification error:

$$\text{Error} = \frac{\text{number of incorrect predictions}}{10}$$

Complete the table:

Stump	Feature used	Number incorrect	Error
A	Age		
B	Blocked arteries		
C	Chest pain		

Part II - Which stump should AdaBoost choose?

AdaBoost starts by giving **every training example the same weight**.

Because there are 10 patients:

$$w_i = \frac{1}{10} = 0.1$$

for every patient.

Thus:

Patient	Initial weight
1	0.10
2	0.10
3	0.10
4	0.10
5	0.10
6	0.10
7	0.10
8	0.10
9	0.10
10	0.10

AdaBoost calculates the **weighted error** of each stump:

$$\epsilon = \sum_{i=1}^{10} w_i I(y_i \neq h(x_i)) \quad (1)$$

where $I(\cdot)$ equals 1 when the prediction is wrong and 0 otherwise.

Questions

1. Calculate the weighted error of Stump A.
2. Calculate the weighted error of Stump B.
3. Calculate the weighted error of Stump C.
4. Which stump would AdaBoost select first?

Part III - AdaBoost gives the stump a weight

Once AdaBoost has selected a weak learner, it gives that learner an importance weight. The weight is

$$\alpha = \frac{1}{2} \ln \left(\frac{1 - \epsilon}{\epsilon} \right) \quad (2)$$

where ϵ is the weighted error of the stump.
Suppose the first stump has weighted error

$$\epsilon = 0.2$$

Questions

1. Calculate α .
2. Is α positive or negative?
3. What does a larger value of α mean?
4. What would happen to α if the stump had an error of exactly 50%?

Part IV - Updating the patient weights

This is the key idea behind AdaBoost.
After the first stump has been trained:

- Patients that were classified **correctly** receive less weight.
- Patients that were classified **incorrectly** receive more weight.

The update can be written as

$$w'_i = w_i \exp(-\alpha y_i h(x_i)) \quad (3)$$

where:

- y_i is the true label,
- $h(x_i)$ is the stump's prediction,
- α is the stump's weight.

A correct prediction has $y_i h(x_i) = +1$, so its weight is multiplied by $e^{-\alpha}$. An incorrect prediction has $y_i h(x_i) = -1$, so its weight is multiplied by e^{α} .

The weights are then **normalized** so that they add up to 1.

Exercise 4.1

Suppose the first stump has

$$\epsilon = 0.2$$

and therefore

$$\alpha \approx 0.693$$

Initially every patient has weight 0.1.

Calculate the new, unnormalized weight for:

- a correctly classified patient;
- an incorrectly classified patient.

Use

$$w'_i = w_i e^{-\alpha y_i h(x_i)} \quad (4)$$

Then normalize all weights.

Questions

1. Which patients receive the largest weights?
2. Why does AdaBoost deliberately give these patients larger weights?
3. After this update, is every patient still equally important to the next stump?

Part V - The second stump

The second stump is trained using the **new patient weights**.

This means that making a mistake on a heavily weighted patient is more costly than making a mistake on a patient with a small weight.

Consider two possible second stumps:

Stump D

$$\hat{y} = \begin{cases} +1 & \text{Chest pain} = \text{Yes} \\ -1 & \text{otherwise} \end{cases}$$

Stump E

$$\hat{y} = \begin{cases} +1 & \text{Weight} > 77.5 \text{ kg} \\ -1 & \text{otherwise} \end{cases}$$

Questions

Using the updated patient weights from Part IV:

1. Determine which patients Stump D classifies incorrectly.
2. Calculate its **weighted error**.

3. Determine which patients Stump E classifies incorrectly.
4. Calculate its **weighted error**.
5. Which stump should AdaBoost choose?
6. Why might the stump with the smallest *number* of mistakes not necessarily be the stump with the smallest **weighted error**?

Part VI - Combining the stumps

AdaBoost does not simply take a majority vote where every stump has equal importance. Instead, it uses the weighted vote

$$F(x) = \alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x) + \dots$$

The final prediction is

$$\hat{y} = \text{sign}(F(x))$$

where:

- $h_1, h_2, h_3, \dots$ are the individual stumps;
- $\alpha_1, \alpha_2, \alpha_3, \dots$ are their weights.

Example

Suppose three stumps produce the following predictions for a new patient:

Stump	Prediction	Stump weight
h_1	+1	0.8
h_2	-1	0.4
h_3	+1	0.3

Calculate

$$F(x) = (0.8)(+1) + (0.4)(-1) + (0.3)(+1)$$

Questions

1. Calculate $F(x)$.
2. What is the final AdaBoost prediction?
3. Why does h_1 have more influence than h_2 ?

Part VII - Understanding AdaBoost

Explain the following training sequence:

1. Initialize all example weights equally.
2. Train a stump using the current weights.
3. Calculate its weighted error and its learner weight α .
4. Update the example weights and normalize them.
5. Repeat steps 2-4, then combine the stumps by weighted vote.

Questions

1. Why does AdaBoost focus increasingly on difficult patients?
2. What happens if a patient is repeatedly misclassified?
3. Why are decision stumps useful as weak learners?
4. Why can many simple stumps produce a much more powerful model when combined?

Part VIII - From AdaBoost to Gradient Boosting

AdaBoost and gradient boosting are both **boosting methods**, but their mechanisms are different.

The central idea of gradient boosting is:

Build a model, examine its errors, and train the next model to improve those errors.

Instead of explicitly changing patient weights as AdaBoost does, gradient boosting can work with the **residuals** or **negative gradients of a loss function**.

A simplified regression example

To understand the mechanism, temporarily imagine that we are predicting a continuous **heart-disease risk score** rather than a yes/no diagnosis.

Suppose the initial model predicts the same risk for every patient:

$$F_0(x) = 0.5$$

For five patients, suppose the observed target values are:

Patient	Actual target y	Initial prediction $F_0(x)$
1	0.0	0.5
2	1.0	0.5
3	1.0	0.5
4	0.0	0.5
5	1.0	0.5

For squared-error loss, the residual is

$$r_i = y_i - F_0(x_i)$$

Questions

1. Calculate the residual for each patient.
2. Which patients does the current model underestimate?
3. Which patients does it overestimate?
4. What should the next decision tree try to predict?

Part IX - Gradient boosting with a decision stump

Suppose the next stump learns the following rule:

$$\text{If Age} > 50 \quad r = +0.3$$

otherwise:

$$r = -0.2$$

The new model is obtained by **adding the correction** to the previous model:

$$F_1(x) = F_0(x) + \eta h_1(x) \tag{5}$$

where η is the **learning rate**.

For this exercise, use $\eta = 1$.

Questions

1. Calculate the new prediction for a patient aged 60.
2. Calculate the new prediction for a patient aged 40.
3. Explain why the second tree is called a **correction** to the first model.
4. What would happen if we continued adding more trees?

Part X - AdaBoost vs. Gradient Boosting

Complete the comparison table.

	AdaBoost	Gradient Boosting
Basic building block		
What happens after an error?		
How is the next learner guided?		
How are learners combined?		
Main idea		

Final questions

1. In your own words, explain the difference between a **stump** and an **ensemble of stumps**.
2. Explain why AdaBoost changes the weights of training examples.
3. Explain how the stump's α determines its influence.
4. Explain how gradient boosting uses the errors of the current model.
5. What is the main conceptual similarity between AdaBoost and gradient boosting?
6. What is the main conceptual difference?

Key concepts

The important ideas to take away are:

Decision stump

A very simple decision tree containing only **one decision**.

AdaBoost

AdaBoost starts with equal patient weights once, then repeats the training and reweighting steps:

1. Uses the current patient weights.
2. Trains a weak learner.
3. Measures its weighted error.
4. Gives the learner a weight α .
5. Increases the weights of incorrectly classified patients.
6. Trains another learner that focuses more on difficult cases.
7. Combines the learners using their weights.

Gradient boosting

Gradient boosting repeatedly:

1. Starts with a simple prediction.
2. Calculates how the current model is wrong according to a loss function.
3. Trains a new tree to predict a **correction** to the current model.
4. Adds that correction to the existing model.
5. Repeats the process.

Thus, both methods turn many **weak learners** into a stronger model, but they determine what the next learner should focus on in different ways.