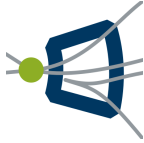


Data Analysis Block Course: Statistics and Machine learning



AdaBoost and Gradient Boosting - Solutions

Prof. Dr. Mikhail Mikhasenko

Assistants: Hendric Jonas, Christoph Tönnis



28 September 2026

[Indico schedule](#)

[RUB Moodle](#)

Part I - Decision stumps

Solution 1.1 - Build some stumps

Each stump predicts +1 when its condition holds and −1 otherwise. The complete predictions are:

Patient	Actual y	Stump A	Stump B	Stump C
1	-1	-1	-1	-1
2	+1	-1	-1	+1
3	+1	-1	+1	-1
4	+1	+1	-1	+1
5	+1	+1	+1	+1
6	+1	+1	+1	-1
7	-1	+1	-1	-1
8	-1	-1	-1	-1
9	+1	+1	+1	-1
10	-1	-1	-1	-1

Stump	Incorrect patients	Number incorrect	Error
A: age > 47	2, 3, 7	3	0.30
B: blocked arteries	2, 4	2	0.20
C: chest pain	3, 6, 9	3	0.30

Part II - Which stump should AdaBoost choose?

With equal weights $w_i = 0.1$, the weighted error is the fraction of incorrectly classified patients:

$$\epsilon_A = 0.30 \quad \epsilon_B = 0.20 \quad \epsilon_C = 0.30$$

Among the three candidates, **Stump B** has the smallest weighted error and is selected first.

Part III - AdaBoost gives the stump a weight

1. For the selected stump, $\epsilon = 0.2$, giving

$$\alpha = \frac{1}{2} \ln \left(\frac{1 - 0.2}{0.2} \right) = \frac{1}{2} \ln 4 = \ln 2 \approx 0.693147$$

2. The weight is positive because $\epsilon < 0.5$.
3. For $0 < \epsilon < 0.5$, a smaller weighted error gives a larger positive α and a stronger contribution to the ensemble vote.
4. At $\epsilon = 0.5$, $\alpha = \frac{1}{2} \ln 1 = 0$. This stump has no influence on the vote.

Part IV - Updating the patient weights

Solution 4.1 - Update and normalize

Stump B misclassifies patients 2 and 4. Using the exact value $\alpha = \ln 2$:

$$\begin{aligned}w'_{\text{correct}} &= 0.1e^{-\ln 2} = 0.05 \\w'_{\text{incorrect}} &= 0.1e^{\ln 2} = 0.20 \\Z &= 8(0.05) + 2(0.20) = 0.80 \\w_{\text{correct,new}} &= 0.05/0.80 = 0.0625 \\w_{\text{incorrect,new}} &= 0.20/0.80 = 0.25\end{aligned}$$

Patient	Correct?	Unnormalized w'_i	Normalized weight
1	Yes	0.05	0.0625
2	No	0.20	0.2500
3	Yes	0.05	0.0625
4	No	0.20	0.2500
5	Yes	0.05	0.0625
6	Yes	0.05	0.0625
7	Yes	0.05	0.0625
8	Yes	0.05	0.0625
9	Yes	0.05	0.0625
10	Yes	0.05	0.0625

The normalized weights sum to $8(0.0625) + 2(0.25) = 1$.

1. Patients 2 and 4 receive the largest weights.
2. Their mistakes contribute more to the next weighted error, encouraging the next stump to classify them correctly.
3. The patients are no longer equally important: each previously misclassified patient has four times the weight of each correctly classified patient.

Part V - The second stump

1. Stump D predicts positive for patients 2, 4 and 5. Its errors are patients **3, 6 and 9**.

2. Its weighted error is

$$\epsilon_D = 3(0.0625) = 0.1875$$

3. Stump E predicts positive for patients **3, 4, 5, 6, 8, 9 and 10**. It predicts negative for patients 1, 2 and 7. Its errors are patients **2, 8 and 10**.
4. Its weighted error is

$$\epsilon_E = 0.25 + 0.0625 + 0.0625 = 0.375$$

5. AdaBoost selects **Stump D**, since $0.1875 < 0.375$.
6. Both candidates make three mistakes, but Stump E misclassifies the heavily weighted patient 2. The identities and weights of the errors matter, not only their number.

Part VI - Combining the stumps

1. The supplied example gives

$$F(x) = (0.8)(+1) + (0.4)(-1) + (0.3)(+1) = 0.7$$

2. The score is positive, so $\hat{y} = +1$: the model predicts heart disease.
3. Stump h_1 has twice the influence of h_2 because its weight is 0.8 rather than 0.4.

Part VII - Understanding AdaBoost

Start with equal example weights **once**. Then repeat: train a stump, calculate its weighted error and α , update and normalize the example weights, and train the next stump. Combine the stumps by their weighted vote.

1. Misclassified patients receive larger relative weights, making their errors more important to subsequent learners.
2. Repeated misclassification can make a patient's weight dominate. This can also emphasize noisy or incorrectly labelled examples.
3. Stumps are simple to train and interpret. Each captures only a limited pattern, making it a weak learner.
4. Different stumps can capture different useful patterns. Their weighted combination allows a more complex decision rule than one split.

Part VIII - From AdaBoost to Gradient Boosting

With the supplied initial prediction $F_0(x) = 0.5$, the residuals are:

Patient	Target y_i	$F_0(x_i)$	Residual r_i
1	0.0	0.5	-0.5
2	1.0	0.5	+0.5
3	1.0	0.5	+0.5
4	0.0	0.5	-0.5
5	1.0	0.5	+0.5

Patients 2, 3 and 5 are **underestimated**; patients 1 and 4 are **overestimated**. The next tree should predict these residuals. For the loss $\ell_i = \frac{1}{2}(y_i - F(x_i))^2$, they are exactly the negative gradients:

$$-\frac{\partial \ell_i}{\partial F(x_i)} = y_i - F(x_i)$$

For squared error without the factor $1/2$, the negative gradient is twice the residual; this constant can be absorbed into the step size. The value $F_0 = 0.5$ is given for this exercise. The optimal constant under squared error for these five targets would be their mean, 0.6 .

Part IX - Gradient boosting with a decision stump

Using $F_0 = 0.5$ and $\eta = 1$:

1. At age 60, the correction is $+0.3$, hence $F_1(60) = 0.5 + 0.3 = 0.8$.
2. At age 40, the correction is -0.2 , hence $F_1(40) = 0.5 - 0.2 = 0.3$.
3. The tree adds a correction to the current prediction. It is trained to address the current errors rather than to replace the entire model.
4. Further iterations fit new corrections to the updated residuals or negative gradients:

$$F_m(x) = F_{m-1}(x) + \eta h_m(x)$$

This can improve the fit, but too many trees can overfit. The learning rate and stopping point control the accumulated corrections.

Part X - AdaBoost vs. Gradient Boosting

Aspect	AdaBoost	Gradient boosting
Building block	Weak classifier, here a stump	Regression tree fitting a correction
After an error	Increase the example's relative weight	Recompute residuals or negative gradients
Next learner	Minimize weighted classification error	Fit the negative loss gradients
Combination	Weighted sum followed by a sign	Add scaled corrections to the current model
Main idea	Focus on difficult training examples	Reduce the chosen loss in stages

Final questions

1. A stump has one split. An ensemble combines several such rules into a more expressive model.
2. AdaBoost changes example weights to make previous mistakes more important to subsequent learners.
3. The factor α scales a stump's contribution to the vote; for errors below 0.5 , lower error gives a larger positive contribution.

4. Gradient boosting fits a new learner to the negative gradient of the current loss. Under squared error, this is proportional to the residual.
5. Both methods build an ensemble sequentially, with each new learner addressing weaknesses of the current model.
6. AdaBoost explicitly reweights examples in this classification formulation; gradient boosting fits corrections based on a chosen loss. AdaBoost can also be viewed as stagewise minimization of exponential loss, so the two ideas are related.

Key concepts

Concept	Meaning
Stump	A tree with one split
Example weight	Importance of an observation during training
Weighted error	Sum of weights of misclassified observations
Learner weight α	Contribution of a weak classifier to the vote
Normalization	Rescaling example weights to sum to one
Residual	Target minus the current prediction
Negative gradient	Direction that locally reduces the loss
Learning rate η	Scale of an added correction
Ensemble	A combination of several models