

Perceptrons and Decision Trees

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Pattern Recognition and Machine Learning

Lecture 05 Perceptrons and Decision Trees



Chizhi Chris ZHANG

zhangchizhi@ciomp.ac.cn

Advanced Computing and Digital Technology Research Center

University of Chinese Academy of Sciences

Autumn 2026

Lecture plan and time allocation

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Topic	Minutes
Perceptron	45
Decision trees	67.5
Integrated reasoning	22.5

Correct a score or choose a question

Follow complete perceptron updates; grow a tree using weighted impurity.

The numerical steps are shown on the slides.



From a score to a decision

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For labels $y \in \{-1, +1\}$, a score $a = w^T x + b$ gives the predicted side.

$$y \in \{-1, +1\} \quad a = w^T x + b$$

An additive spam score suits several weak clues. A rule such as “urgent AND unfamiliar sender” expresses a conditional combination.

$a(x) = 0$ is the boundary. The vector w is perpendicular to it. The bias b shifts it.

$$a(x) = w_1 x_1 + w_2 x_2 + b = w^T x + b, \quad \hat{y} = \begin{cases} +1, & a(x) > 0 \\ -1, & a(x) \leq 0. \end{cases}$$

$$a(x) = 0 \quad w = (-2, 0), \quad b = 1$$

$$a(0, 1) = 1 \quad a(1, 0) = -1$$

$$x_1 = 0.5 \quad \tilde{x} = (x_1, x_2, 1)^T$$

$$\tilde{w} = (w_1, w_2, b)^T \quad a(x) = \tilde{w}^T \tilde{x}$$



The signed score identifies a mistake

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Key idea

The signed score is $m = y(w^T x + b)$. A negative value means the score points to the wrong side. The loss is zero at $m = 0$, but our training convention still updates boundary cases.

$m_i = y_i a(x_i), \quad y_i \in \{-1, +1\}.$			
y_i	$a(x_i)$	m_i	Training interpretation
+1	2	2	Correct side
-1	-3	3	Correct side
-1	2	-2	Mistake
+1	0	0	On the boundary: update

$$J(w) = \sum_{i:m_i \leq 0} (-m_i)$$

A mistake gives an update direction

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

$m = y(w^T x + b) \leq 0$. We want to increase m . Its derivatives are $\partial m / \partial w = yx$ and $\partial m / \partial b = y$.

$$m = y(w^T x + b) \leq 0 \quad \partial m / \partial w = yx$$

$$\partial m / \partial b = y$$

$$w' = w + \eta yx, \quad b' = b + \eta y.$$

$$m' = y(w'^T x + b') = m + \eta y^2 (x^T x + 1) = m + \eta (\|x\|^2 + 1).$$

$$\eta > 0$$

$$\ell_i = -y_i \tilde{w}^T \tilde{x}_i \text{ and } \nabla_{\tilde{w}} \ell_i = -y_i \tilde{x}_i.$$

$$\ell_i = -y_i \tilde{w}^T \tilde{x}_i \quad \nabla_{\tilde{w}} \ell_i = -y_i \tilde{x}_i$$

$$\tilde{w}_{\text{new}} = \tilde{w} - \eta \nabla \ell_i = \tilde{w} + \eta y_i \tilde{x}_i, \quad \eta > 0.$$

$$x = (1, 0), y = -1, \eta = 1$$

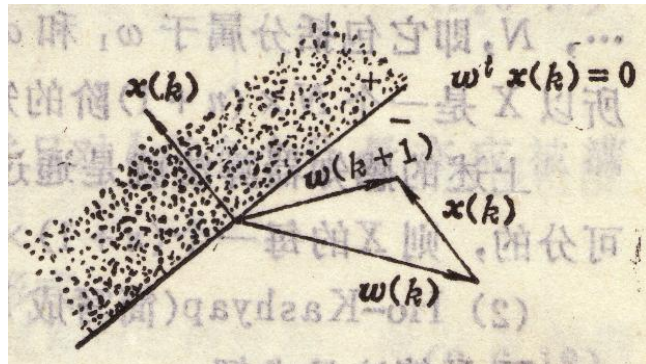
$$\Delta w = (-1, 0) \quad \Delta b = -1$$



Read the perceptron update as vector addition

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For a mistaken positive example, add ηx to the weights. For a mistaken negative example, subtract it. Updating the bias follows the same signed direction.



Why the corrected sample improves

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

An update increases the current signed score. It can still leave that example on the wrong side.

Substitute the new weights

$$\begin{aligned} y_i \tilde{w}_{\text{new}}^T \tilde{x}_i &= y_i (\tilde{w} + \eta y_i \tilde{x}_i)^T \tilde{x}_i \\ &= y_i \tilde{w}^T \tilde{x}_i + \eta \| \tilde{x}_i \|^2. \end{aligned}$$

Labels satisfy $y_i^2 = 1$. The added quantity is positive.

For another example j , the change is $\eta y_i y_j \tilde{x}_i^T \tilde{x}_j$, which may help or hurt it. Even a raw zero input can update the bias: its augmented vector is $(0, 0, 1)$.

Improvement is not completion

For $\tilde{x} = (1, 0, 1)$ and $\eta = 1$,

$$\eta \| \tilde{x} \|^2 = 1^2 + 0^2 + 1^2 = 2.$$

$$m_{\text{old}} = -5 \Rightarrow m_{\text{new}} = -3.$$

The example is still wrong.

Two updates in the first pass, with every substitution

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

$\tilde{w}^{(0)} = (0, 0, 0)$, $\eta = 1$, visit order A, B, C, D. Update whenever $y\tilde{w}^\top \tilde{x} \leq 0$.

Case	(x_1, x_2)	$\tilde{x}$	y
A	(0, 0)	(0, 0, 1)	+1
B	(0, 1)	(0, 1, 1)	+1
C	(1, 0)	(1, 0, 1)	-1
D	(1, 1)	(1, 1, 1)	-1

$$\tilde{w}^{(0)} = (0, 0, 0) \quad \eta = 1$$

Use the newest w, b immediately. The following pass table includes B and D as well.

$$x = (0, 0), y = +1 \quad w = (0, 0), b = 0$$

$$m = 1(0 + 0 + 0) = 0$$

$$w' = (0, 0) + (0, 0) = (0, 0), \quad b' = 0 + 1 = 1.$$

$$x = (1, 0), y = -1 \quad w = (0, 0), b = 1$$

$$m = -1[0(1) + 0(0) + 1] = -1,$$

$$w' = (0, 0) - (1, 0) = (-1, 0), \quad b' = 1 - 1 = 0.$$

Epoch 1 changes the bias and the first weight

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A first creates a positive bias. C then lowers its own score by changing w_1 and b .

Visit	Weights before	Signed score	update	Weights after
A	(0, 0, 0)	0	$+(0, 0, 1)$	(0, 0, 1)
B	(0, 0, 1)	1	none	(0, 0, 1)
C	(0, 0, 1)	-1	$-(1, 0, 1)$	(-1, 0, 0)
D	(-1, 0, 0)	1	none	(-1, 0, 0)

Check. A and B now lie on score zero. This epoch contained two updates, so training continues.

Later epochs produce a verified solution

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A complete pass has zero updates. The solution is $a(x) = -2x_1 + 1$, with boundary $x_1 = 0.5$.

Round	Updates during this pass	Final weights
2	A: $(-1, 0, 0) \rightarrow (-1, 0, 1)$; C: $(-1, 0, 1) \rightarrow (-2, 0, 0)$	$(-2, 0, 0)$
3	A: $(-2, 0, 0) \rightarrow (-2, 0, 1)$; B, C and D are already correct	$(-2, 0, 1)$
4	No update on A, B, C, D	$(-2, 0, 1)$

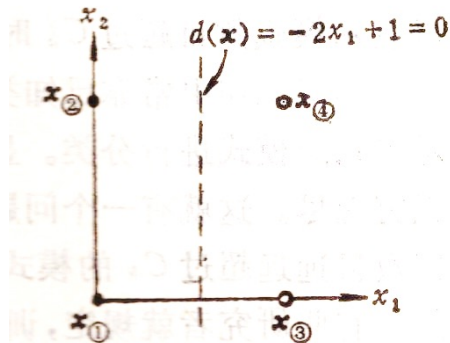
$$a(A) = a(B) = 1, \quad a(C) = a(D) = -1.$$

$$a(x) = -2x_1 + 1 \quad x_1 = 0.5$$



The learned score defines a vertical boundary

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER



x_1 is horizontal and x_2 is vertical. The dashed line is $-2x_1 + 1 = 0$.

$$-2x_1 + 1 = 0 \quad x = (0.25, 0.8)$$

$$a = 0.5 \quad x_1 = 0.75$$

$$w_2 = 0$$



The implementation matches the displayed calculation

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

```
w = np.zeros(X.shape[1])
b = 0.0
for epoch in range(max_epochs):
    mistakes = 0
    for x, y in zip(X, labels):
        if y * (w @ x + b) <= 0:
            w += eta * y * x
            b += eta * y
            mistakes += 1
    if mistakes == 0:
        break
```

Labels are $-1, +1$. Both w and b update. A whole zero-update epoch triggers stopping. Keep a maximum epoch limit for nonseparable data.



Convergence requires a separable training set

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Earlier updates change the scores seen by later examples. A different visit order may reach a different separating hyperplane.

There exists a unit vector u and $\gamma > 0$ such that $y_i u^T \tilde{x}_i \geq \gamma$ for every training example.

$\gamma > 0$

Condition to check

Generalization on new samples, probability calibration and the size of the learned margin do not follow from this theorem.

A larger margin permits fewer mistakes

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Conditions.

Start at zero, use step size 1, and update only if $y_i \tilde{w}^T \tilde{x}_i \leq 0$. Assume a unit separator u satisfies $y_i u^T \tilde{x}_i \geq \gamma > 0$, and $\|\tilde{x}_i\| \leq R$.

Progress toward u

$$u^T \tilde{w}_{t+1} = u^T \tilde{w}_t + y_i u^T \tilde{x}_i \geq u^T \tilde{w}_t + \gamma.$$

After M updates, $u^T \tilde{w}_M \geq M\gamma$.

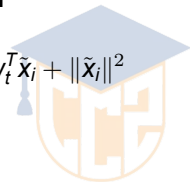
Growth of the weight length

$$\begin{aligned} \|\tilde{w}_{t+1}\|^2 &= \|\tilde{w}_t\|^2 + 2y_i \tilde{w}_t^T \tilde{x}_i + \|\tilde{x}_i\|^2 \\ &\leq \|\tilde{w}_t\|^2 + R^2. \end{aligned}$$

Thus $\|\tilde{w}_M\|^2 \leq MR^2$.

$$M\gamma \leq u^T \tilde{w}_M \leq \|\tilde{w}_M\| \leq \sqrt{MR} \quad \Rightarrow \quad M \leq (R/\gamma)^2.$$

The unit-vector projection cannot exceed vector length. The radius includes the augmented bias coordinate.



XOR reveals a limitation of one linear score

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Here the coordinates are 0, 1, unlike the $-1, +1$ corners used in Lecture 4. Require the sign of $g = w_1x_1 + w_2x_2 + b$ to match y .

x_1	x_2	y
0	0	-1
0	1	+1
1	0	+1
1	1	-1

$$(0, 0) : b < 0,$$

$$(0, 1) : w_2 + b > 0,$$

$$(1, 0) : w_1 + b > 0,$$

$$(1, 1) : w_1 + w_2 + b < 0.$$



The two positive cases give $w_1 > -b$ and $w_2 > -b$. Therefore

$$w_1 + w_2 + b > -b > 0,$$

contradicting the required negative score at $(1, 1)$. A feature such as $z = x_1 + x_2 - 2x_1x_2$ changes the representation.

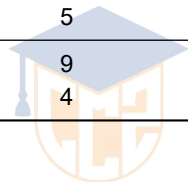
Nonseparable data needs a practical stopping rule

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Set a maximum number of epochs. Count training mistakes after each candidate update or epoch.

Check. Use validation data to choose model settings and report test performance once. Pocket search does not prove optimality or repair label noise.

Epoch	1	2	3	4	5
Training errors	12	8	10	7	9
Retained candidate	1	2	2	4	4



Multiclass perceptron corrects two class scores

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For multiple classes, keep one augmented weight row per class. Predict the largest score. If prediction $\hat{y}$ is wrong, raise the true row and lower the wrongly chosen row:

$$\hat{y} = \arg \max_c \tilde{w}_c^T \tilde{x}.$$

$$\tilde{w}_y \leftarrow \tilde{w}_y + \eta \tilde{x}, \quad \tilde{w}_{\hat{y}} \leftarrow \tilde{w}_{\hat{y}} - \eta \tilde{x}.$$

Take $\tilde{x} = (1, 1, 1)$, true class B and $\eta = .5$. The last coordinate is the bias input.

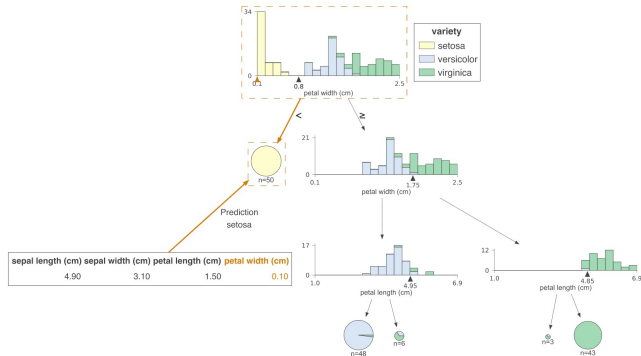
Class	Old weights	Old score	New weights	New score
A	(1, 0, 1)	2	(.5, -.5, .5)	.5
B	(0, 1, 0)	1	(.5, 1.5, .5)	2.5
C	(0, 0, -1)	-1	unchanged	-1

Initially A wins incorrectly. After this update B wins. Other class rows stay unchanged; more training observations may still require updates.



A tree prediction is a path of questions

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER



The perceptron changes one global score. A tree chooses conditional tests.

Petal width 0.10 cm is below 0.8 cm.
The left leaf predicts setosa.



Paths create conditional regions

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The root divides the full space. A later split divides only the region inherited from its ancestors. Leaves cover disjoint regions.

Path	Region and prediction
$x_1 \leq 2$	Predict class A.
$x_1 > 2$ versus $x_2 \leq 5$	Predict class B.
$x_1 > 2$ versus $x_2 > 5$	Predict class C.

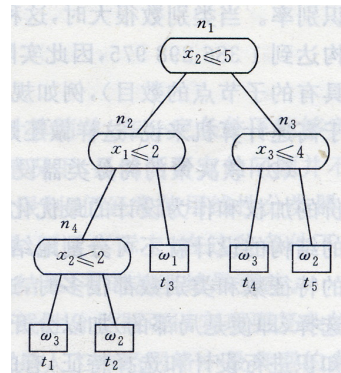


Check. A rule learned in a small region is not evidence of a global causal relationship.

Trace a complete root-to-leaf decision

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Start at the root, apply each inequality, and stop only when a class leaf is reached.



A leaf estimates a label distribution

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A classification leaf estimates label proportions from the training records that reach it. A regression leaf predicts a number. Squared loss selects the mean of its targets.

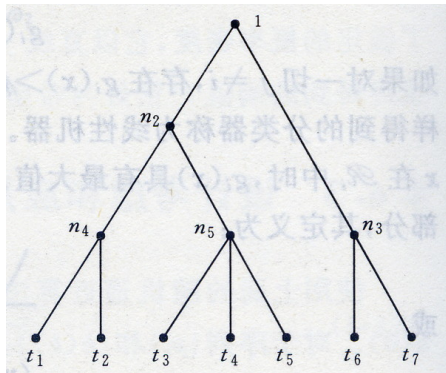
$$\hat{p} = (6/10, 3/10, 1/10) = (0.6, 0.3, 0.1), \quad \hat{y} = \text{Category 1}.$$

For responses 4, 1 and 9, the constant minimizing squared error is their mean $(4 + 1 + 9)/3 = 14/3$. Classification predicts a class; regression predicts a number.



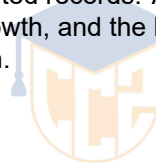
Recursive growth repeats one local decision

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER



$x_j \leq t$ sends a sample left. Sort unique values and test midpoints between adjacent values.

Every child repeats the same local procedure on its inherited records. A stopping rule ends growth, and the leaf supplies the prediction.



Six fruits determine the admissible split candidates

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Six measured fruits provide three oranges and three lemons. The two features supply seven admissible thresholds. Their memberships must be known before computing impurity.

width	height	Class
7.0	9.5	Orange
6.9	6.0	Orange
6.6	6.6	Orange
6.5	6.6	Lemon
6.0	6.6	Lemon
6.9	9.6	Lemon



Entropy measures a node's label uncertainty

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Key idea

An outcome with probability p has surprise $-\log_2 p$. Entropy averages that surprise using the label proportions within the node.

$$H = - \sum_k p_k \log_2 p_k.$$

$$H = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = \frac{1}{2} + \frac{1}{2} = 1.$$
$$-1 \log_2 1 = 0$$

A pure node leaves no label uncertainty. A balanced binary node has maximum uncertainty. The base changes units, not the ordering of split gains.

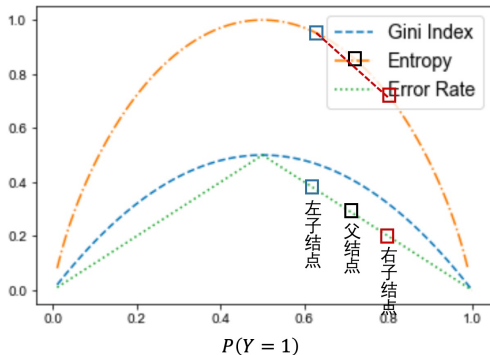
$$H(D) = - \sum_{c=1}^C p_c \log_2 p_c,$$

$$p_c = n_c/n, \quad 0 \log_2 0 := 0.$$

Binary counts	Class probabilities	Entropy in bits
(10, 0)	(1, 0)	0
(5, 5)	(0.5, 0.5)	1
(7, 3)	(0.7, 0.3)	0.8813

Entropy is largest at a balanced binary node

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER



The orange entropy curve reaches zero at pure nodes and 1 at a balanced binary node. The blue curve will provide a second impurity measure.



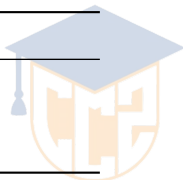
Gini gives another measure of mixture

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Two independently drawn labels from the empirical node distribution disagree with probability G .

$$G(D) = 1 - \sum_c p_c^2. \quad \text{Binary: } G = 2p(1 - p).$$

Counts	Gini	Majority-class error
(10, 0)	0	0
(5, 5)	0.5	0.5
(7, 3)	0.42	0.3



The best split lowers weighted child impurity

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

To isolate weighting, this page first uses a separate ten-row example. We then return to the six fruits and their candidate thresholds.

Use the same sample weighting for more than two children. Class weights replace raw counts with consistent weighted counts.

$$I_{\text{after}} = \frac{n_L}{n} I(D_L) + \frac{n_R}{n} I(D_R), \quad \Delta I = I(D) - I_{\text{after}}.$$

$$I_{\text{after}} = \frac{2}{10} \cdot 0 + \frac{8}{10} \cdot 0.5 = 0.4, \quad \Delta I = 0.08.$$

Check. A simple average gives 0.25 and exaggerates the split's benefit. A tiny pure leaf cannot represent the other 80% of examples.



Width 6.55: membership before impurity

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Both width-6.9 fruits go right. A threshold cannot separate equal feature values.

width	Class	Width ≤ 6.55
7.0	Orange	No
6.9	Orange	No
6.6	Orange	No
6.5	Lemon	Yes
6.0	Lemon	Yes
6.9	Lemon	No

$$G_L = 0 \quad G_R = 1 - (3/4)^2 - (1/4)^2 = 3/8$$

$$G_{\text{after}} = (2/6)(0) + (4/6)(3/8) = 1/4.$$



Width 6.55 wins the threshold comparison

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Width ≤ 6.55 : (0 orange, 2 lemon). Width > 6.55 : (3 orange, 1 lemon).

Worked calculation

Feature	Threshold and weighted child Gini			
width	6.25 : 0.400	6.55 : 0.250	6.75 : 0.444	6.95 : 0.400
height	6.30 : 0.400	8.05 : 0.500	9.55 : 0.400	—

$$G_{\text{after}} = \frac{2}{6}(0) + \frac{4}{6} [1 - (3/4)^2 - (1/4)^2] = 0.25.$$

$$0.5 - 0.25 = 0.25$$

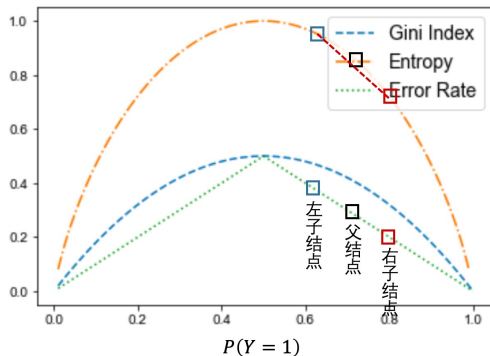
Error rate and impurity can favor different splits

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A parent with 7 positives and 3 negatives splits into children (3, 2) and (4, 1). Both still predict positive. Error stays .30, but Gini falls from .42 to .40: class proportions contain information beyond the majority label.

$$E_{\text{after}} = \frac{0.4 + 0.2}{2} = 0.30.$$

$$G_{\text{after}} = \frac{0.48 + 0.32}{2} = 0.40.$$



Categorical splits ask membership questions

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A categorical split asks whether a value belongs to a subset, such as ward A or C. Coding categories as 1, 2 and 3 would impose an arbitrary order.

$$\text{GainRatio}(X) = \frac{H(Y) - H(Y|X)}{-\sum_k (n_k/n) \log_2(n_k/n)}.$$

The numerator is label-entropy reduction. The denominator is the entropy of branch membership, adjusting the preference for many branches. A zero denominator needs special handling.

Check. A unique ID can isolate every training case but teach nothing about a new ID. Gain ratio also needs care when its denominator is small.

The denominator is the entropy of child membership: many tiny children receive a larger split-entropy penalty. A zero denominator means no effective split.

Regression splits reduce squared error

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

$\bar{y}_R$ minimizes $\sum_{i \in R} (y_i - c)^2$. A split compares the sum of both children's squared errors.

$$\sum_{i \in R} (y_i - c)^2 \quad x = (1, 2, 3, 4)$$

$$x \leq 2.5 : \quad \bar{y}_L = 1.5, \bar{y}_R = 8.5, \quad \text{SSE}_L + \text{SSE}_R = 0.5 + 0.5 = 1.$$

Check. A basic regression tree cannot extrapolate a trend beyond observed regions. Its leaf value remains a training-derived constant.



A depth-two tree represents XOR

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Two sequential questions create four pure leaves. One linear score in the original two features cannot do this.

Path	Predicted
$x_1 \leq 0.5, x_2 \leq 0.5$	-1
$x_1 \leq 0.5, x_2 > 0.5$	+1
$x_1 > 0.5, x_2 \leq 0.5$	+1
$x_1 > 0.5, x_2 > 0.5$	-1



Condition to check

Either root split leaves both children 50% positive, so its immediate gain is zero. A learner that stops at zero gain will not discover this exact tree.

Postpruning trades impurity for fewer leaves

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A tree can keep splitting until each leaf holds one example. Training impurity then reaches zero even when labels contain errors.

With minimum leaf size 4, a split into 2 and 8 samples is inadmissible even if the small child is pure.

$R(T)$ is sample-weighted leaf impurity. $|L(T)|$ is the number of leaves. $\alpha \geq 0$ charges for each leaf.

$$R_{\alpha}(T) = R(T) + \alpha|L(T)|, \quad R(T) = \sum_{t \in L(T)} \frac{n_t}{n} I(D_t).$$

$$R = 0.10 \quad R = 0.16$$

$$\alpha < 0.02 \quad \alpha > 0.02$$

$$\alpha = 0.02$$



A branch has a calculable pruning threshold

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A costs $.10 + 6\alpha$. B costs $.16 + 3\alpha$. Ask when A is cheaper.

$$.10 + 6\alpha < .16 + 3\alpha \Rightarrow 3\alpha < .06 \Rightarrow \alpha < .02.$$

α	A cost	B cost	Choose
.01	.16	.19	A
.02	.22	.22	Tie
.03	.28	.25	B



Replace subtree T_t with a single leaf t . Costs outside that subtree cancel.

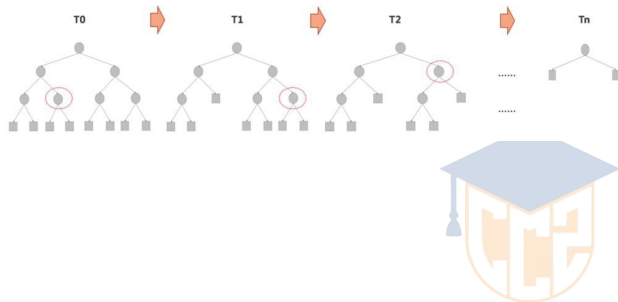
$$R(T_t) + \alpha|L(T_t)| = R(t) + \alpha \Rightarrow \alpha_{\text{eff}} = \frac{R(t) - R(T_t)}{|L(T_t)| - 1}.$$

$$(0.14 - 0.06)/(3 - 1) = 0.04$$

A pruning path supplies nested candidates

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

$T_0, T_1, T_2, \dots$ have fewer leaves. Each tree keeps part of the earlier structure. The training data supply effective alpha values.



Validation chooses complexity on unseen examples

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The medium candidate wins on these validation observations. Its lower training accuracy is the price of removing unstable details.

Candidate	Leaves	Train accuracy	Validation accuracy
Deep	24	1.00	0.84
Medium	10	0.97	0.89
Shallow	3	0.88	0.85

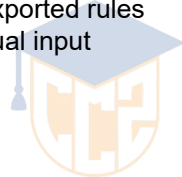
Check. This table is a teaching example, not a measured result from the course code. Small score differences need variability estimates.

Tree training exposes interpretable parameters

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

```
tree = DecisionTreeClassifier(  
    max_depth=3, min_samples_leaf=5,  
    criterion="gini", random_state=0)  
tree.fit(X_train, y_train)  
pred = tree.predict(X_valid)  
print(export_text(tree, feature_names=names))  
path = tree.cost_complexity_pruning_path(  
    X_train, y_train)  
alphas = path.ccp_alphas
```

Depth 3 and leaf size 5 restrict growth. The pruning path uses training examples. Candidate alphas still need validation comparison. Exported rules refer to the actual input columns.



Tiny leaves produce fragile probabilities

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A leaf with 2 positive and 0 negative examples reports $\hat{p} = 1$. A leaf with 90 positive and 10 negative examples reports $\hat{p} = 0.9$.

$$\hat{p} = 1$$

$$\tilde{p}_c = (n_c + \lambda) / (n + C\lambda) \quad \lambda = 1$$

$$\tilde{p}_+ = \frac{2 + 1}{2 + 2} = \frac{3}{4}.$$



Importance and stability need different evidence

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Feature importance and prediction stability answer different questions.

Evidence	What it describes	What still needs checking
Split importance	Weighted impurity reduction in this fitted tree	A large score does not establish a causal effect.
Refitting	Changes after small training-sample perturbations	Compare the selected feature, threshold and prediction.
Interpretation	The rule followed by this tree	Another plausible tree may use a different rule.



A readable tree can still be unstable. Validate the predictions and qualify the explanation.

The same split supports both model pipelines

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The same evaluation rules apply to both learners. We can now compare a completed perceptron update with a completed tree split.

Reserve identical test observations. Fit preprocessing only on training data. Choose settings through validation. Compare suitable metrics and inspect the same mistakes.

Both models see identical rows. Only the linear pipeline learns scaling here. Tune settings using training and validation data, then compare final test predictions.



```
linear = make_pipeline(  
    StandardScaler(),  
    Perceptron(max_iter=1000, random_state=0))  
tree = DecisionTreeClassifier(  
    max_depth=3, min_samples_leaf=5, random_state=0)  
for model in [linear, tree]:  
    model.fit(X_train, y_train)  
    print(balanced_accuracy_score(  
        y_valid, model.predict(X_valid)))
```

The update raises this score but reaches the boundary

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The feature update alone does not give the complete new classifier. Both w' and b' are used in the next score.

Worked calculation

$$w = (1, -1) \quad b = -.5$$

$$x = (0, 2) \quad y = +1$$

$$\eta = .5$$

$$a = w^T x + b = 1(0) - 1(2) - .5 = -2.5,$$

$$ya = -2.5 \leq 0 \Rightarrow \text{update},$$

$$w' = w + \eta y x = (1, -1) + .5(1)(0, 2) = (1, 0),$$

$$b' = b + \eta y = -.5 + .5 = 0.$$

The training rule updates when $ya \leq 0$. This row has improved, but it is still a boundary case, so one update is not proof of convergence.

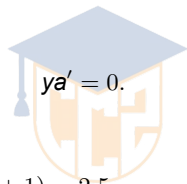
$$w' = (1, 0) \quad b' = 0$$

$$x = (0, 2) \quad y = +1$$

$$a' = 1(0) + 0(2) + 0 = 0, \quad ya' = 0.$$

$$a' - a = 0 - (-2.5) = 2.5$$

$$\eta(\|x\|^2 + 1) = .5(0^2 + 2^2 + 1) = 2.5.$$



The lower weighted impurity gives the larger gain

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The parent has counts (4, 4). Weight each child impurity by its share of the parent.

$$G_P = 1 - (4/8)^2 - (4/8)^2 = 1/2$$

Candidate	Child counts	Sizes	Child Gini
1 left	(3, 1)	4	$1 - (3/4)^2 - (1/4)^2 = 3/8$
1 right	(1, 3)	4	$3/8$
2 left	(2, 0)	2	0
2 right	(2, 4)	6	$1 - (2/6)^2 - (4/6)^2 = 4/9$

$$G_1 = \frac{4}{8} \frac{3}{8} + \frac{4}{8} \frac{3}{8} = \frac{3}{8},$$

$$G_2 = \frac{2}{8} (0) + \frac{6}{8} \frac{4}{9} = \frac{1}{3},$$

$$\Delta G_1 = \frac{1}{2} - \frac{3}{8} = \frac{1}{8},$$

$$\Delta G_2 = \frac{1}{2} - \frac{1}{3} = \frac{1}{6}.$$

$$1/3 < 3/8 \quad 1/6 > 1/8$$

Check. It selects the better local training split. Future performance and stopping still require the complexity controls and validation discussed earlier.

$1/3 < 3/8$, equivalently $1/6 > 1/8$: candidate 2 wins. The parent is identical, so minimizing post-split impurity and maximizing the gain give the same ranking.



Can a model represent a rule that training fails to find?

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Question

A depth-two tree can represent XOR. Will greedy tree training necessarily find it?

Evidence and reasoning

No. On balanced XOR, either one-feature root split has zero gain. A rule requiring strictly positive immediate gain can stop before the useful second split. Representation, search procedure and stopping rule are separate parts of the explanation.

Check. The perceptron convergence argument also requires separability; repeated local updates alone do not establish it.