Institute for Web Science & Technologies

UNIVERSITÄT KOBLENZ · LANDAU

Machine Learning and Data Mining WS21/22

“7 Decision Trees & Random Forest”

Dr. Zeyd Boukhers
@ZBoukhers

Institute for Web Science and Technologies
University of Koblenz-Landau

December 8, 2021

From previous lecture

- KNN
- K-D Tree
- Bayes Theorem
- Naïve Bayes

Agenda

- Decision Trees
- Overfitting/Underfitting
- Random Forest

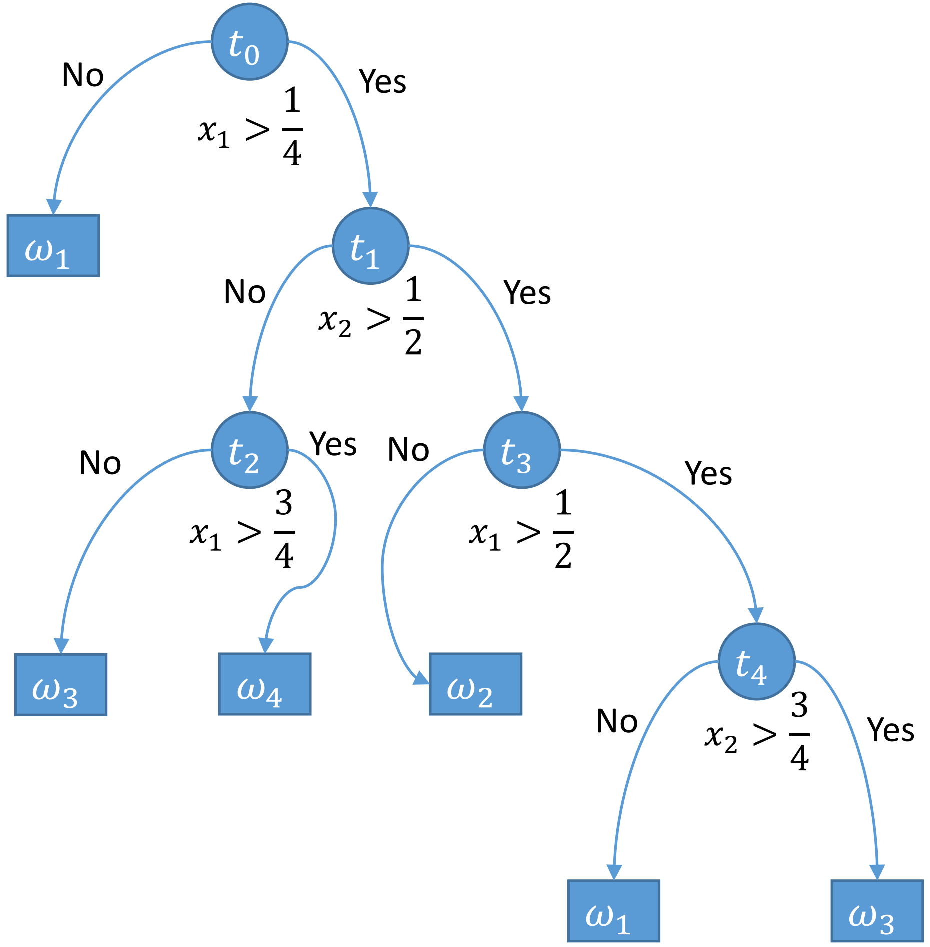
Decision Trees

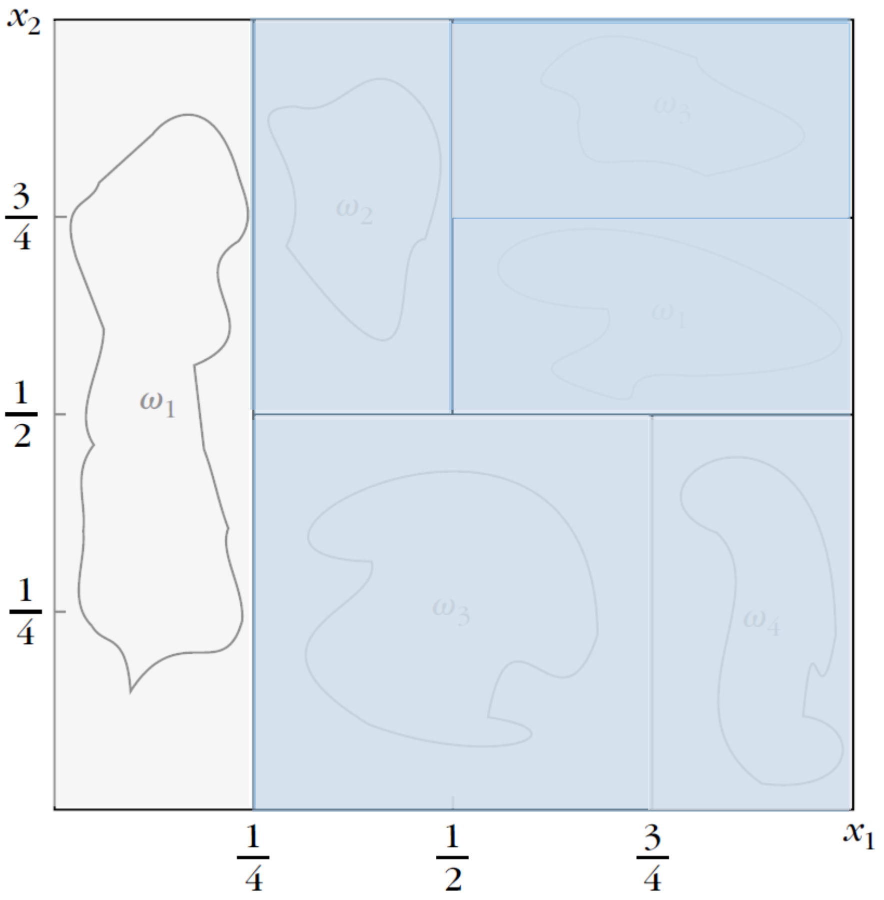
Decision Trees

- What are they?
  - Non-linear classifiers
  - Multistage decision
    - the feature space is split into unique regions, corresponding to the classes, *in a sequential manner*.
      - Hyper-rectangles
      - Convex polyhedral cells
      - Pieces of spheres
      - Etc.
    - At each stage, the decision is made based on the answer of the question: is $x_j \leq \alpha$, where α is a threshold value.

  - Work well when a large number of classes are involved.

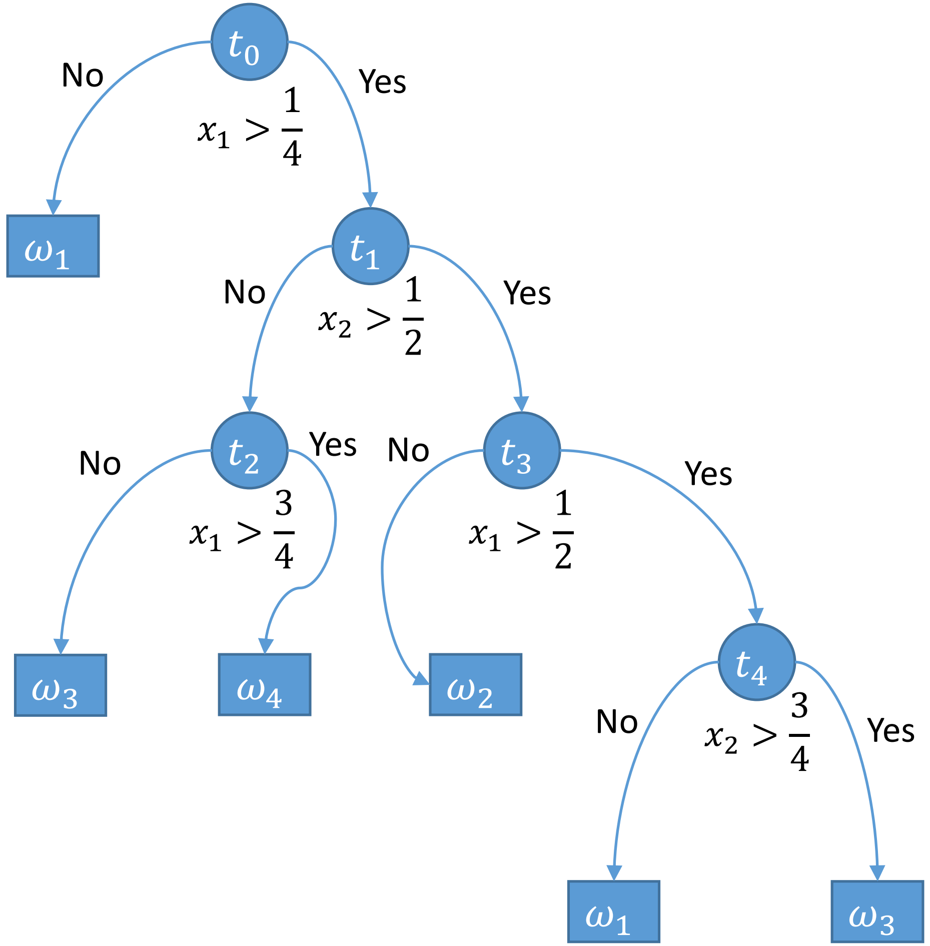
Decision Trees

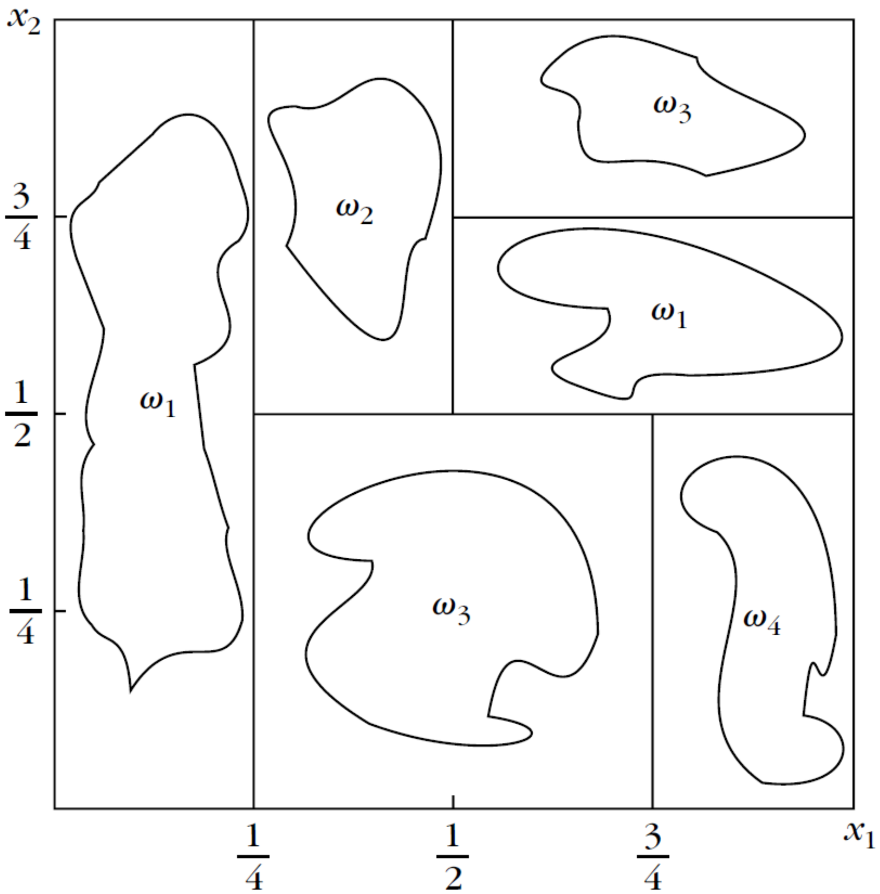




Source: Theodoridis, S., & Koutroumbas, K. “Pattern recognition. ” Fourth Edition, 9781597492720, 2008

Decision Trees





Source: Theodoridis, S., & Koutroumbas, K. “Pattern recognition. ” Fourth Edition, 9781597492720, 2008

Notes

- Some observations from the example:
  - Two-dimensional space (remember, we use two dimensions in almost all our example only for an intuitive visualization).
  - The thresholds are obtained by a simple observation of the geometry of the feature space.
  - The tree started with x_1. Why?

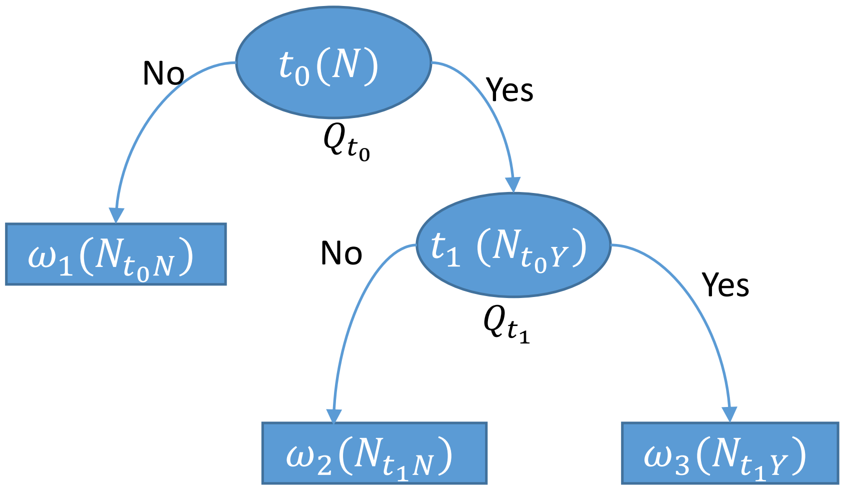
Decision Trees

- Given:
  - A k-class classification problem,
  - A data set $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N]$
- Each node t is associated with
  - a subset X_t, where X is associated with the root (t_0).
  - a question Q_t.
- At each node t,
  - The subset X_t is split into two *disjoint descendant subsets* X_{tY} and X_{tN}.
    - X_{tY} consists of feature vectors corresponding to the answer "Yes" of the question.
  - The following is true:
    - $X_{tY} \cap X_{tN} = \emptyset$,
    - $X_{tY} \cup X_{tN} = X_t$

- The *splitting criterion* must be adopted according to the best candidate question.
- The growth of the tree is controlled with *a stop-splitting rule,* where the terminal node is called "*leaf*".
- For each attribute x_j, α_j can take any possible value.
  - There is an infinite set of questions has to be asked if $\alpha_j \in \mathbb{R}$
  - In practice, only a finite set can be considered.
  - At a node t, any attribute x_j can take at most $N_t < N$ different values.
  - At a node t, the total number of candidate questions is $\sum_{u=1}^{l} N_{t,u}$
  - Only one question has to be chosen.
    - It must be the one leading to the best split.

Splitting criterion

- Every split must generate subsets that are more "class homogenous" compared to the ancestor.
  - The instances in each subset show a higher preference for the corresponding class.
  - E.g. N_{t_0N} and N_{t_0Y} have to be more homogenous (or purer) than N.
  - How can we measure this?



Impurity measure (Entropy)

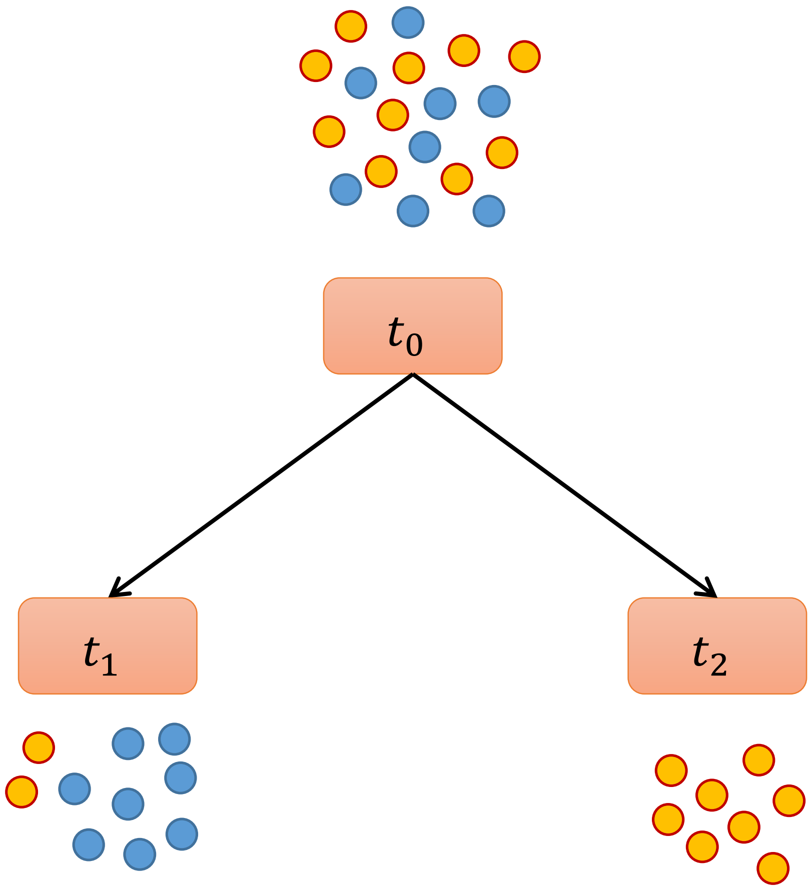
- **Goal:** quantifying node impurity and split the node such as the overall impurity of the descendant nodes decreases w.r.t the impurity of the ancestor node.
- Let $P(\omega_j|t)$ be the probability that an instance associated with a node t, belongs to class $\omega_j, j = 1, 2, \ldots, k.$, the node impurity $I(t)$ is written as:

$$I(t) = -\sum_{j=1}^{k} P(\omega_j|t) \log_2 P(\omega_j|t)$$

- This is nothing else than the entropy associated with the subset X_t
- $P(\omega_j|t) = \frac{N_t^j}{N_t}$, where N_t^j is the number of points in X_t that belong to class ω_j.

Impurity measure (Entropy)

- Example
  - Parent node (t_0):
    - 10 orange, 8 blue objects
    - Impurity $I = 0.991$
  - Child Node t_1:
    - 2 orange, 8 blue objects
    - Impurity $I = 0.722$
  - Child Node t_2:
    - 8 orange, 0 blue objects
    - Impurity $I = 0$



Source: MLDM Slides by Prof. Steffen Staab

Impurity measure (Entropy)

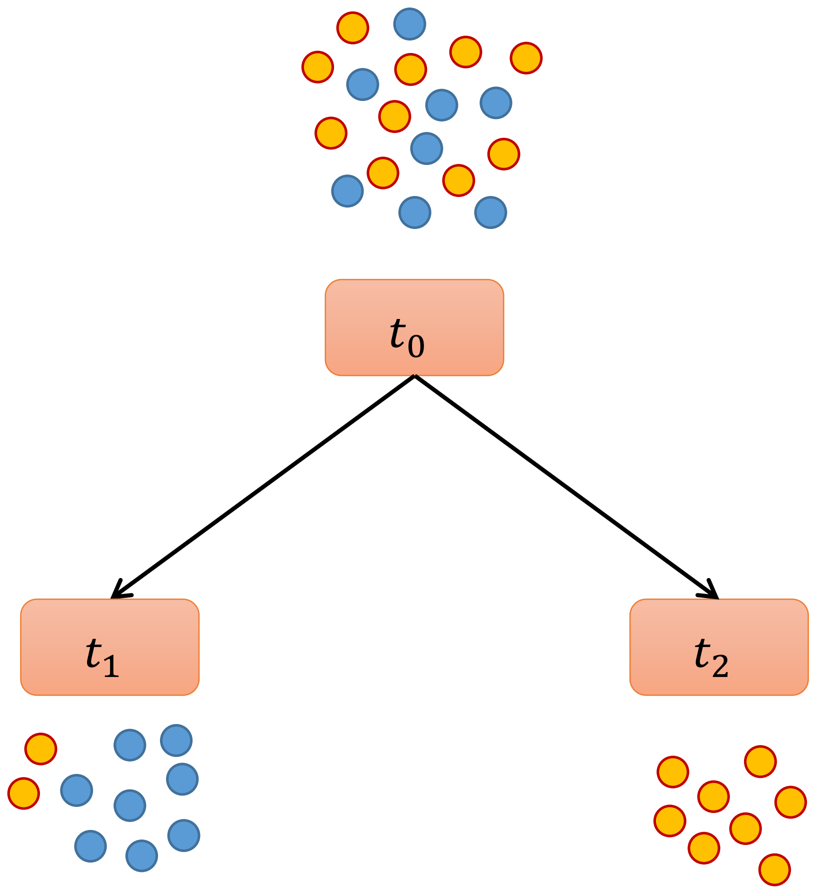
- After performing a split, N_{tY} points are sent into the "Yes" node (X_{tY}) and N_{tN} into the "No" node (X_{tN}). The *decrease in node impurity* is defined as:

$$\Delta I(t) = I(t) - \sum_{o} \frac{N_{to}}{N_t} I(t_o) = I(t) - \frac{N_{tY}}{N_t} I(t_Y) - \frac{N_{tN}}{N_t} I(t_N)$$

The goal now becomes to adopt, from the set of candidate questions, the one that performs the split leading to the highest decrease of impurity.

Impurity measure (Entropy)

- Example
  - Parent node (t_0):
    - 10 orange, 8 blue objects
    - Impurity $I = 0.991$
  - Child Node t_1:
    - 2 orange, 8 blue objects
    - Impurity $I = 0.722$
  - Child Node t_2:
    - 8 orange, 0 blue objects
    - Impurity $I = 0$



$$\Delta I(t) = 0.991 - \left(\frac{10}{18} \cdot 0.722 + \frac{8}{18} \cdot 0\right) = 0.59$$

Source: MLDM Slides by Prof. Steffen Staab

Stop-Splitting rule and class assignment rule

- when to decide to stop splitting a node and declares it as a leaf of the tree.
  - If the maximum value of $\Delta I(t)$, over all possible splits, is less than T, which is a predefined a threshold.
  - If N_t is small enough.
  - If X_t is pure (zero impurity).
    - All samples in it belong to a single class.

- Every leaf in the tree has to be assigned to a class. the leaf is labelled as $\omega_{\hat{j}}$ where:

$$\hat{j} = \arg\max_{j} P\left(\omega_j \middle| t\right).$$

Binary Decision Tree: algorithm

- The root node $X_t = X$
- For each new node t
  - For every feature $x_u, u = 1, \ldots, l$
    - For every value $\alpha_{u,n}, n = 1, \ldots, N_{t,u}$
      - Generate X_{tY} and X_{tN} according to: $x_{u,i} \leq \alpha_{u,n}, i = 1, \ldots, N_t$
      - Compute $\Delta I(t)$.
    - Choose $\alpha_{u,n0}$ leading to the maximum of $\Delta I(t)$.
  - Choose x_{u0} and associated $\alpha_{u0,n0}$ leading to the overall maximum of $\Delta I(t)$.
  - If the stop-splitting rule is met, declare node t as a leaf and designate it with a class label
  - If not, generate two descendant nodes t_Y and t_N with associated subsets X_{tY} and X_{tN}, depending on: is $x_{u0} \leq \alpha_{u0,n0}$

Non-Binary Decision Tree: algorithm

- The root node $X_t = X$
- For each new node t
  - For every feature $x_u, u = 1, \dots, l$
    - Generate $X_{u,i}, i = 1, \dots, N_{t,u}$ according to the question
    - Compute $\Delta I(t)$.
  - Choose x_{u0} leading to the maximum of $\Delta I(t)$.

Non-Binary vs. Binary Decision Tree

- Non-binary:
  - One child node per value
  - Problem:
    - High fan-out of tree
    - Not applicable to all attribute types
- Binary:
  - Binary splits of the value set
  - Advantage: binary tree
  - Disadvantage: Many possible splits
    - t values → $2^t - 2$ possible splits

Notes

- the size of a tree must be large enough but not too large; otherwise, it tends to learn the particular details of the training set → Overfitting
- The best threshold value for the impurity decreases is very hard to define and does not lead to trees of the right size.
  - One solution is to grow a tree up to a large size first and then prune nodes according to a ***pruning*** criterion.

Example

???

- Entropy root node
  - 100 instances
  - 21 poisonous
  - 79 edible
  - Impurity: $\mathrm{I}(t_0) = 0.741$
- Attribute: Cap-shape

value	P	E	total	impurity
b	0	29	29	0
f	1	13	14	0.371
s	0	3	3	0
x	20	34	54	0.951

- Impurity: 0.565

$$\Delta I(t_0) = 0.176$$

Source: MLDM Slides by Prof. Steffen Staab

Example

???

- Attribute: Cap-surface

$$\Delta I(t_0) = 0.053$$

- Attribute: Cap-color

$$\Delta I(t_0) = 0.236$$

- ...
- Attribute: Habitat

$$\Delta I(t_0) = 0.279$$

value	p	e	total	Impurity
f	0	14	14	0
s	8	29	37	0.753
y	13	36	49	0.835

Source: MLDM Slides by Prof. Steffen Staab

Example

???

- Attribute: Cap-surface

$$\Delta I(t_0) = 0.053$$

value	p	e	total	Impurity
f	0	14	14	0
s	8	29	37	0.753
y	13	36	49	0.835

- Attribute: Cap-color

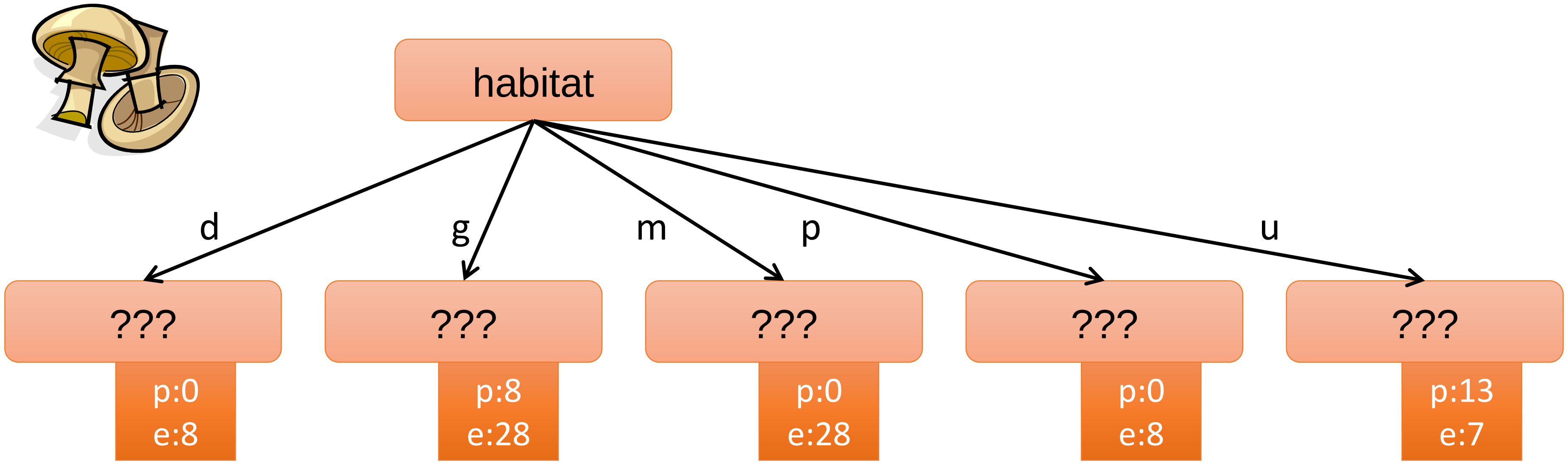
$$\Delta I(t_0) = 0.236$$

- …
- Attribute: Habitat

$$\Delta I(t_0) = 0.279$$

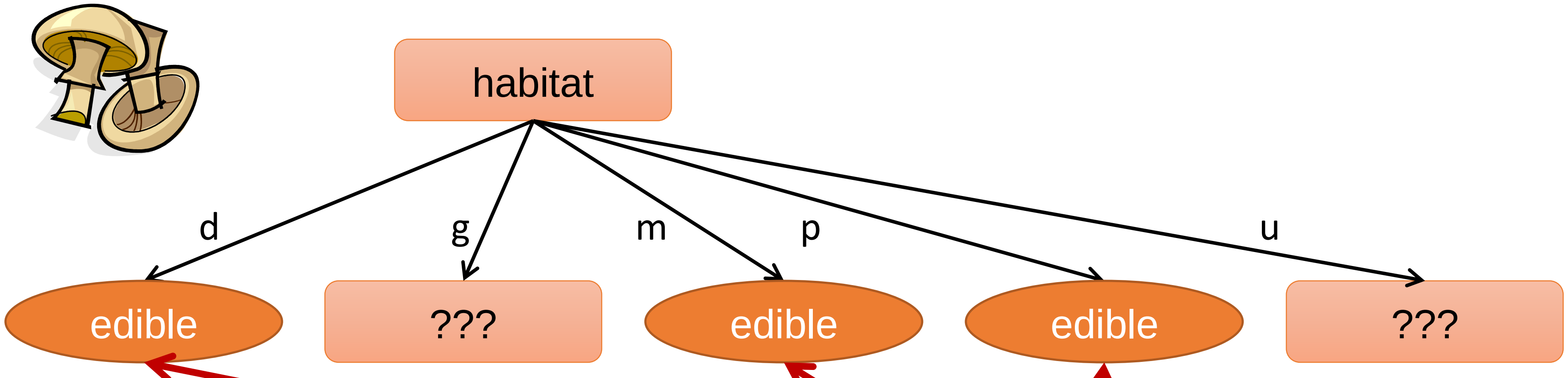
value	p	e	total	Impurity
d	0	8	8	0
g	8	28	36	0.764
m	0	28	28	0
p	0	8	8	0
u	13	7	20	0.934

Example



value	p	e	total	Impurity
d	0	8	8	0
g	8	28	36	0.764
m	0	28	28	0
p	0	8	8	0
u	13	7	20	0.934

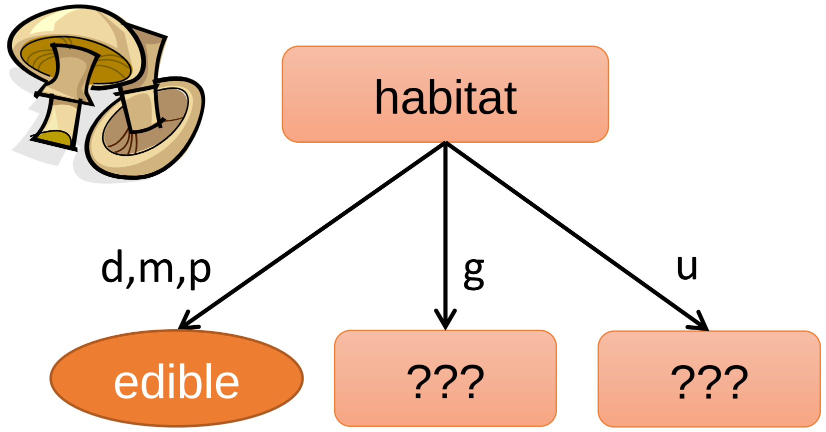
Example



value	p	e	total	Impurity
d	0	8	8	0
g	8	28	36	0.764
m	0	28	28	0
p	0	8	8	0
u	13	7	20	0.934

Merge nodes (typically: post processing)

Example



value	p	e	total	Impurity
d	0	8	8	0
g	8	28	36	0.764
m	0	28	28	0
p	0	8	8	0
u	13	7	20	0.934

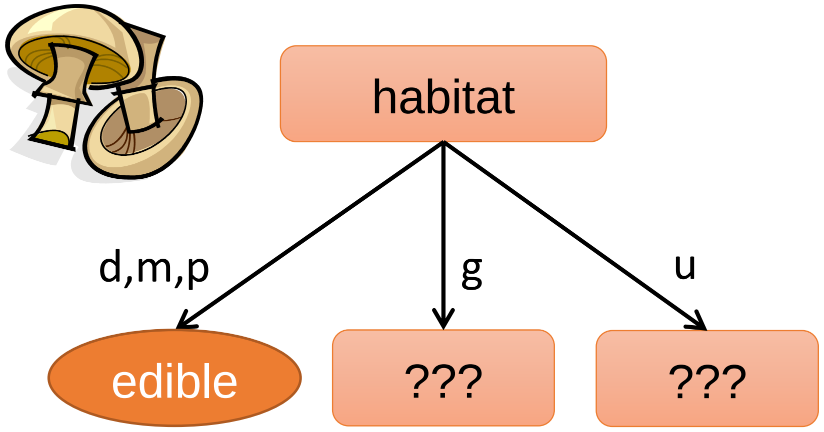
- Next node
  - 36 samples
  - 8 poisonous
  - 28 edible
  - Impurity: $I(t_1) = 0.764$
- Attribute: Cap-shape

value	p	e	total	Impurity
b	0	8	8	0
f	1	6	7	0.592
s	0	0	0	0
x	7	14	21	0.918

- Impurity: 0.651

$$\Delta I(t_1) = 0.113$$

Example



- Attribute: Cap-surface

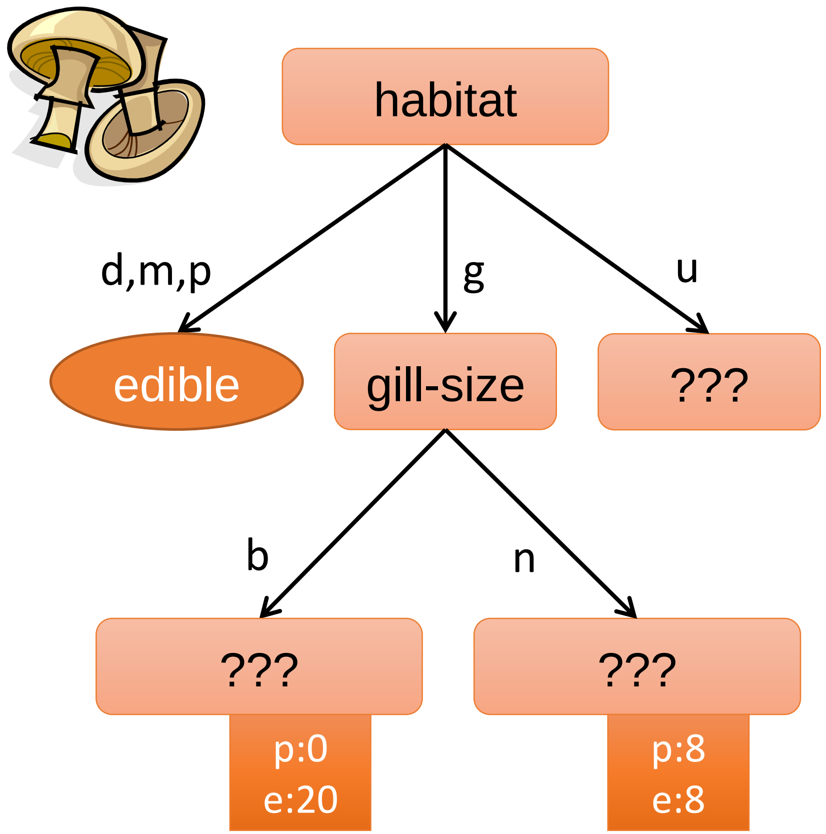
$$\Delta I(t_1) = 0.113$$

- …
- Attribute: Gill-size

$$\Delta I(t_1) = 0.32$$

value	p	e	total	Impurity
b	0	20	20	0
n	8	8	16	1

Example



- Attribute: Cap-surface

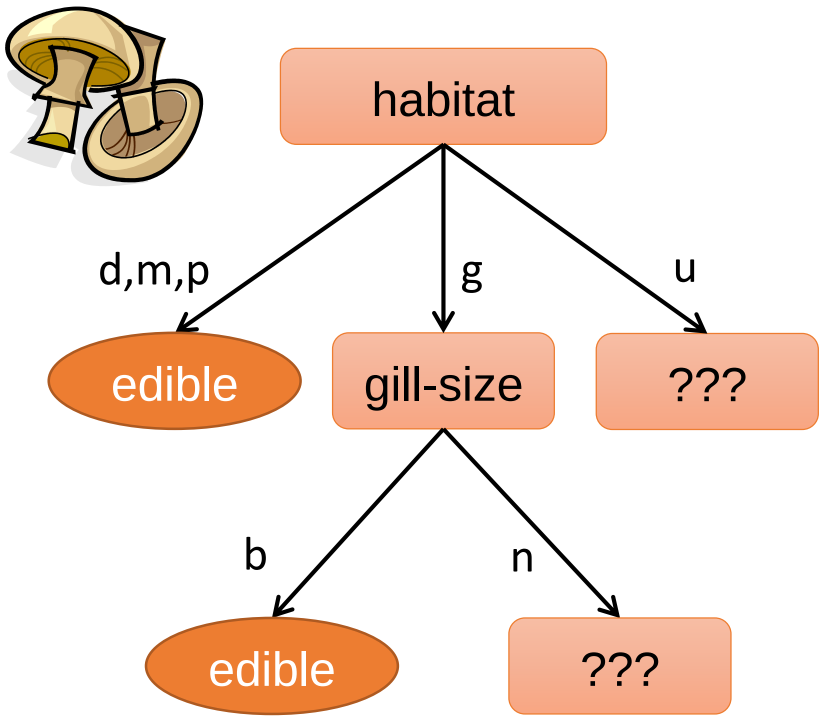
$$\Delta I(t_1) = 0.113$$

- ...
- Attribute: Gill-size

$$\Delta I(t_1) = 0.32$$

value	p	e	total	Impurity
b	0	20	20	0
n	8	8	16	1

Example



- Attribute: Cap-surface

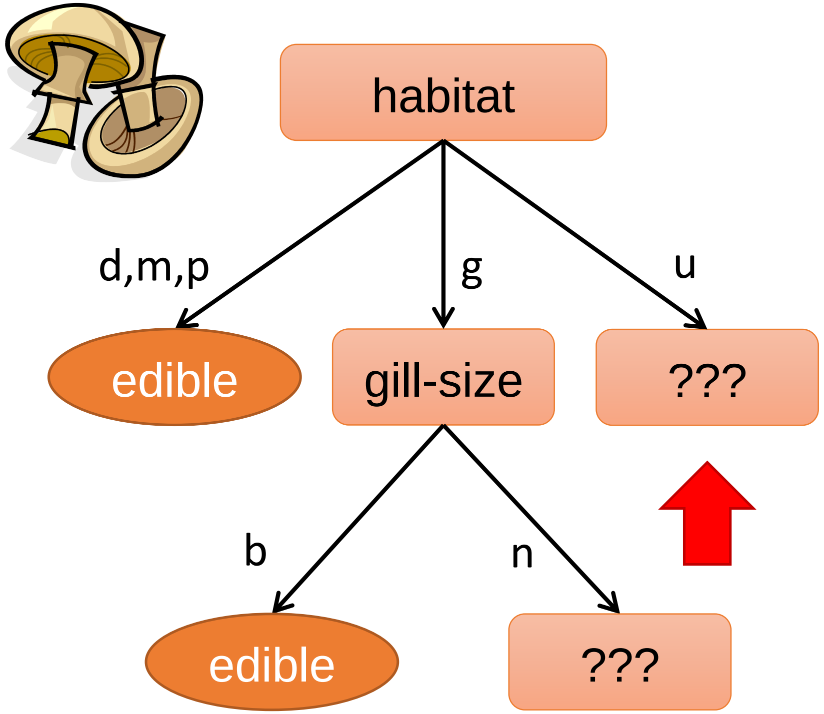
$$\Delta I(t_1) = 0.113$$

- ...
- Attribute: Gill-size

$$\Delta I(t_1) = 0.32$$

value	p	e	total	Impurity
b	0	20	20	0
n	8	8	16	1

Example



- Next node
  - 20 samples
  - 13 poisonous
  - 7 edible
  - Entropy: $I(t_2) = 0.934$
- …

More about Decision Trees

Observations

- What if there is an attribute that has a large number of values?
  - The parent set is split into a large number of children subsets.
  - Only a few samples in each child subset.
  - The children subsets are more likely to be pure.
  - ➢ This attribute has a higher chance to be chosen first because *impurity decrease* (Stop-splitting criterion) is biased towards choosing attributes leading to pure children $\Rightarrow$ attributes with a large number of values.
- Consequences:
  - Overfitting.
  - Too many children subsets.

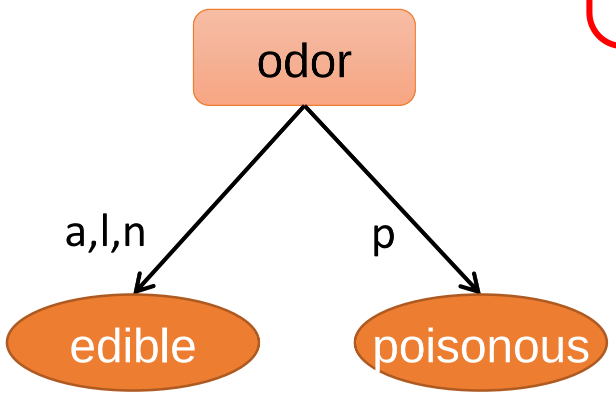
Observations

- What if there is an attribute that perfectly distinguishes between the two classes?
    - $\Delta I(t_0) = I(t_0) \Rightarrow$ The maximum possible Impurity decrease.

cap-shape	cap-surface	cap-color	bruises	gill-spacing	gill-size	gill-color	habitat	odor
bell=b convex=x flat=f sunken=s	fibrous=f scaly=y smooth=s	brown=n gray=g white=w yellow=y	bruises=t no=f	close=c crowded=w	broad=b narrow=n	black=k brown=n gray=g pink=p white=w	grasses=g meadows=m paths=p urban=u woods=d	almond=a, anise=l none=n pungent=p

new

value	p	e	total	entropy
a	0	31	31	0
l	0	35	35	0
n	0	13	13	0
p	21	0	21	0



$$\Delta I(t_0) = 0.741$$

Normalized Impurity Decrease

- Use *Normalized Impurity Decrease* (Gain ratio) instead of *Impurity Decrease* (Information Gain). The *Normalized Impurity Decrease* at a node t is then:
  - $\hat{\Delta} I(t) = \frac{\Delta I(t)}{E_d(t)}$, where $E_d(t)$ is the *entropy of distribution* (Intrinsic Information).
  - $E_d(t) = -\sum_j P(t_j|t) \log_2 P(t_j|t)$,
  - $P(t_j|t) = \frac{N_{t_j}}{N_t}$.

Example

???

- Root node
  - 100 samples
  - 21 poisonous
  - 79 edible
  - Impurity: $I(t_0) = 0.741$
- Attribute: Cap-shape
  - Impurity: 0.565
  - Entropy of distribution: 1.547

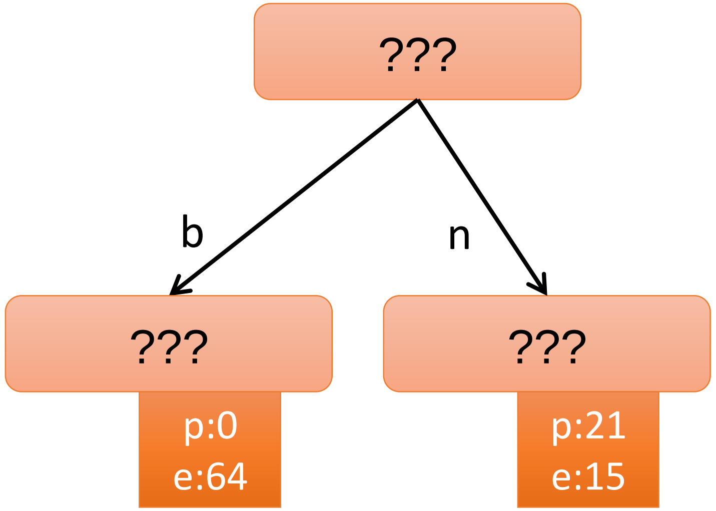
$$\Delta I(t_0) = 0.176$$

$$\hat{\Delta} I(t_0) = 0.114$$

value	p	e	total	Impurity
b	0	29	29	0
f	1	13	14	0.371
s	0	3	3	0
x	20	34	54	0.951

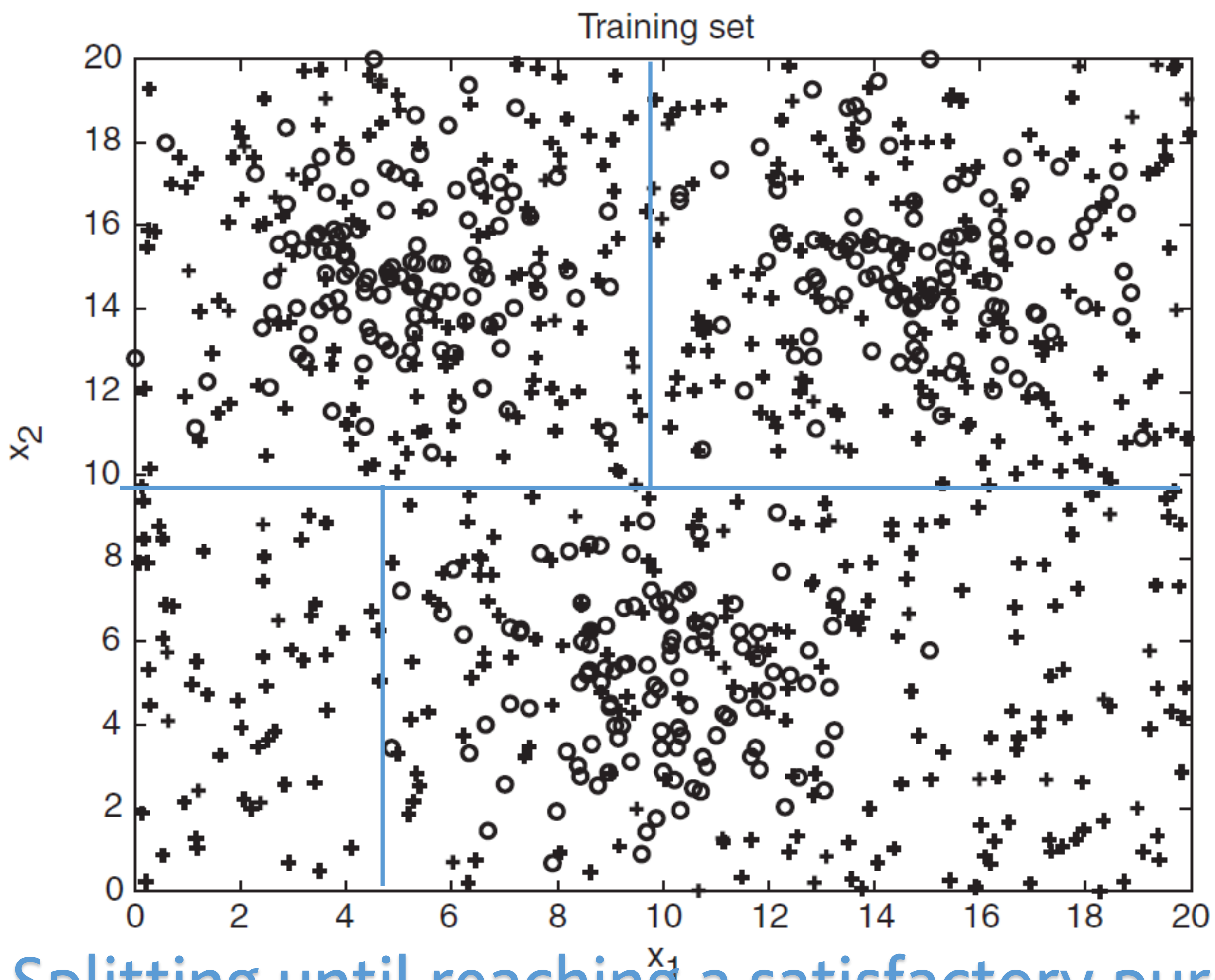
Example from [1]

- Attribute: Habitat
  - $\hat{\Delta}I(t_0) = \ 0.134$
- Attribute: Gill-size
  - Entropy of distribution: 0.943
  - $\hat{\Delta}I(t_0) = \ 0.412$



value	p	e	total	Impurity
b	0	64	64	0
n	21	15	36	1

Overfitting



1200 o
1800 +

Splitting until reaching a satisfactory pureness withing the training set.

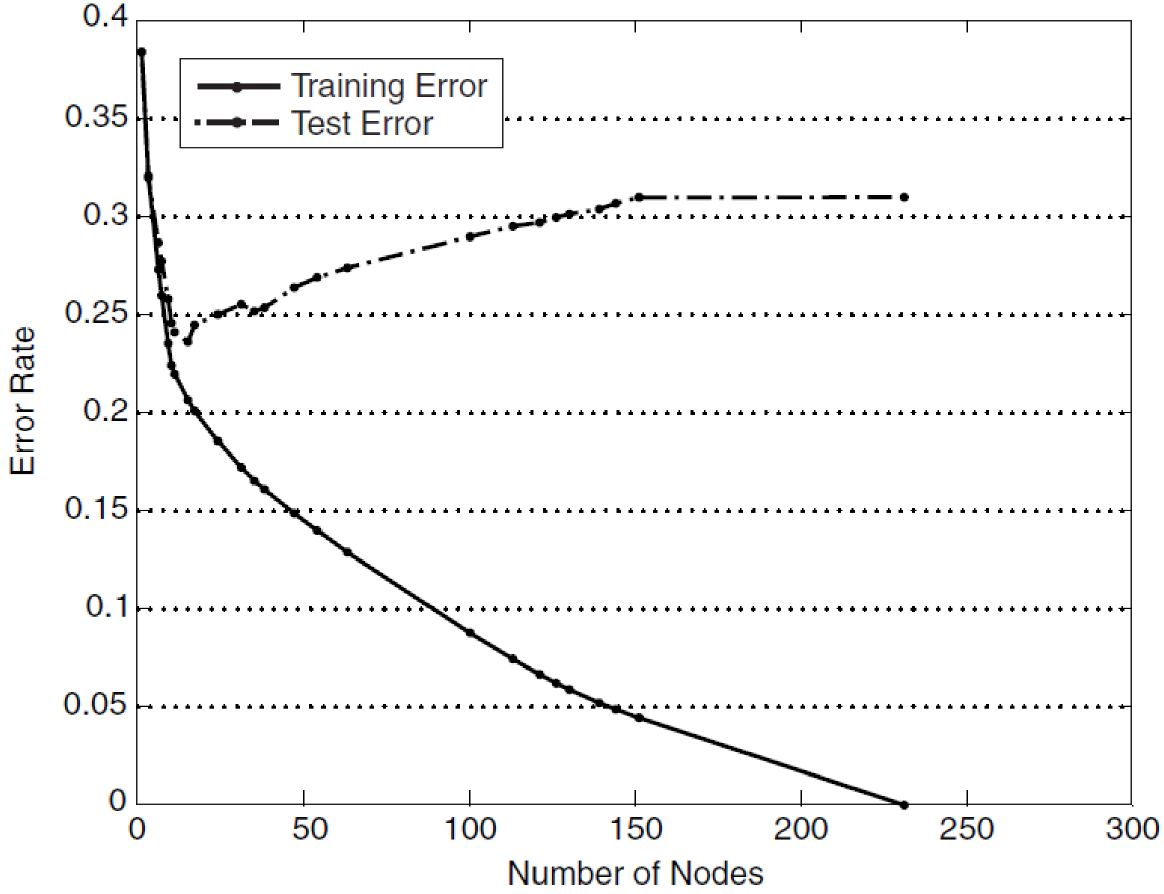
Source: « MLDM slides » from Prof. Steffen Staab lecture.

Overfitting

- Remember the Stop-Splitting rules
  - If the maximum value of $\Delta I(t)$, over all possible splits, is less than T, which is a predefined a threshold. $\Rightarrow$ ***Pre pruning.***
  - If N_t is small enough.
  - If X_t is pure (zero impurity).
    - All samples in it belong to a single class.

Overfitting

Data split: 30% training, 70% evaluation



Source: « MLDM slides » from Prof. Steffen Staab lecture.

Tree Post Pruning

- Grow a tree up to a large size.
- Prune (remove) nodes according to a pruning criterion [*].
  - E.g. Expected Generalisation Error (Applied on the training set but expected for the testing set),
  - Reduce Error Pruning (Applied on a validation set).

- At a node t with J children nodes $t_{j=1,\dots,J}$,

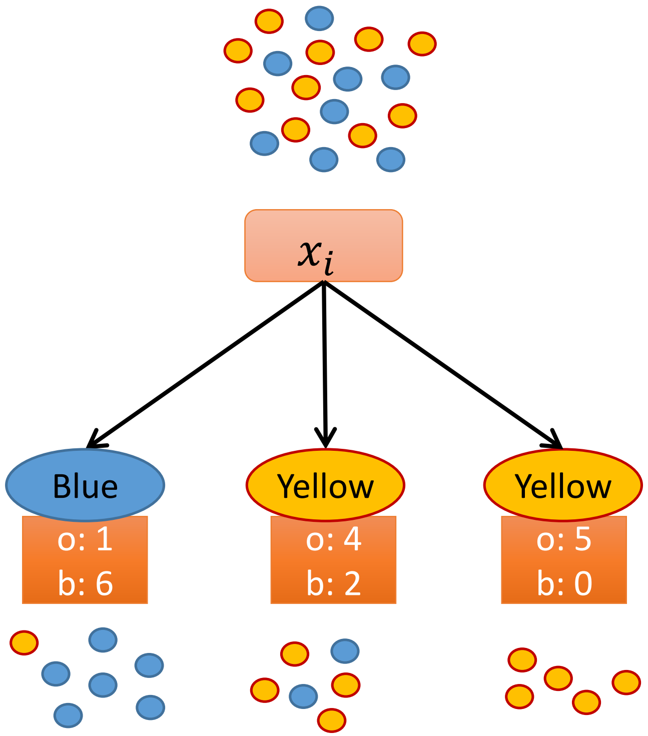
the overall error rate (optimistic) is:

  - $\frac{1}{N_t}\sum_j |\gamma(N_{t_j}) \neq \omega(t_j)|$

Example: 0.167

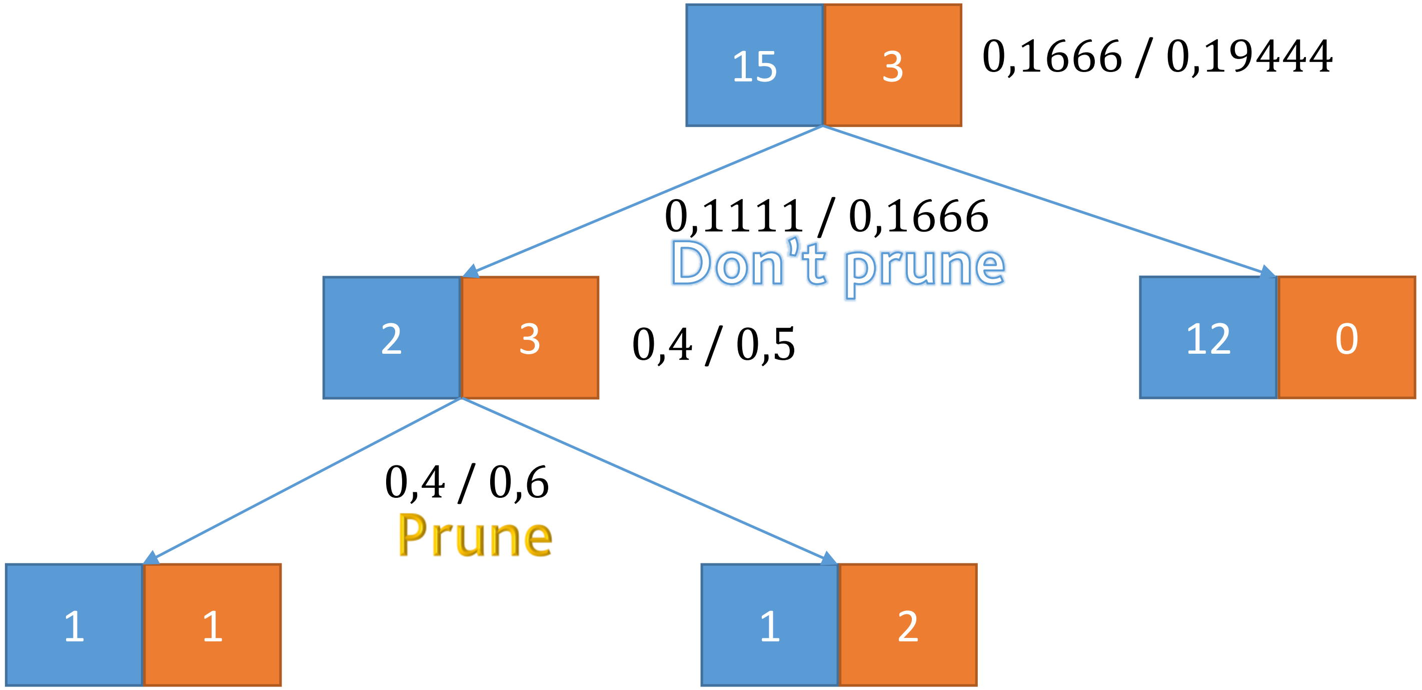
- Use the pessimistic error:
  - $\frac{1}{N_t}\sum_j |\gamma(N_{t_j}) \neq \omega(t_j)| + \lambda$

Example (λ=0.5): 0.25



[*] Patel, N., & Upadhyay, S. (2012). Study of various decision tree pruning methods with their empirical comparison in WEKA. *International journal of computer applications*, *60*(12).

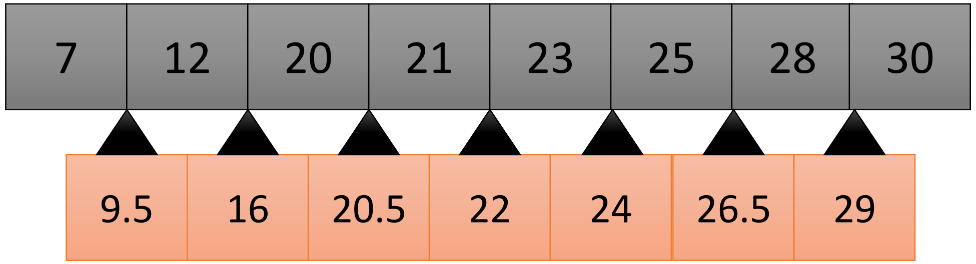
Example



Optimistic / Pessimistic

Type of attributes

- Categorical values
  - Split w.r.t the values belonging to different classes.
- Ordinal or continuous values
  - Split w.r.t a threshold (higher or lower)
  - For continuous values, we can use the middle value between two observed values for better separation boundaries.



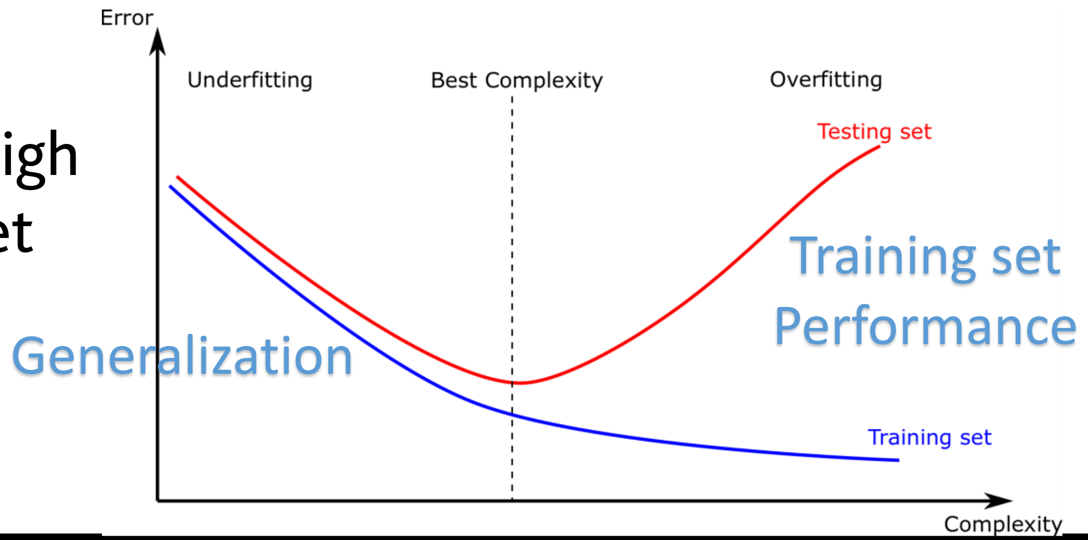
Pros and Cons

- Advantages
  - Interpretable.
  - Non-parametric.
  - Can handle missing data.
  - Low complexity (prediction) $O(l)$.
  - Invariant to feature scaling.
    - Does not require data normalization.
  - Can handle heterogeneous data.
    - Attributes of different types.
- Disadvantages
  - Splits are aligned w.r.t axes.
    - It might cause overfitting because the tree goes far more complex than needed.

Overfitting & Underfitting

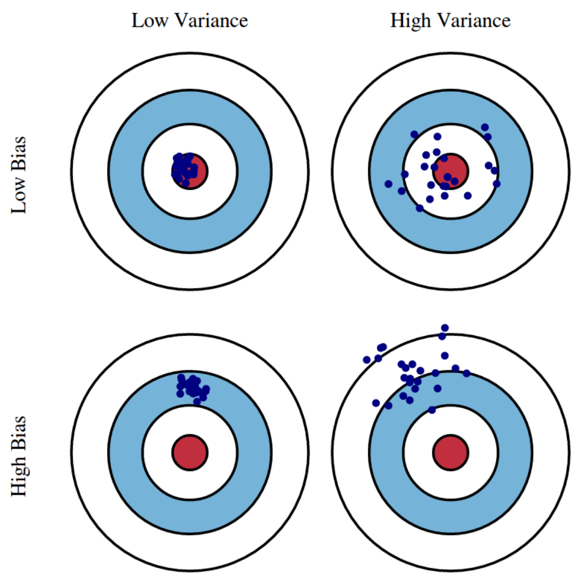
- **Overfitting**: the model is too complex and captures all the details in the training set, *but* it does not achieve the same accuracy on the testing set (not generalized).
- **Underfitting**: The model is too simple and can achieve a good accuracy on neither the training set nor the testing set.
- **Generalization**: When the model achieves similar accuracy on unseen data as it does on the training data.

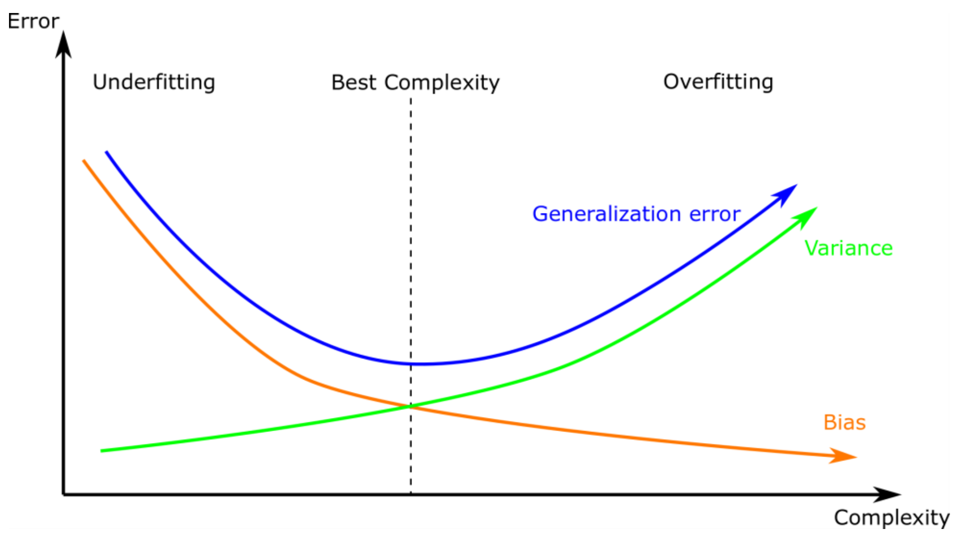
  - **Good model**: it achieves high accuracy on the training set and generalizes well on unseen data.



Overfitting & Underfitting

- Bias error: Expected (or assumed) error in the result given by the model.
- Variance error: Variability in the result given by the model when the dataset is changed.





Source: Lecture of Kilian Weinberger.
http://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html

Generalization Error

- Training error:

$$E_{\text{train}} = \frac{1}{N}\sum_{i=1}^{N} \text{error}\big(\gamma(\boldsymbol{x}_i), \omega(\boldsymbol{x}_i)\big).$$

- Generalization error:

$$E_{\text{gen}} = \int \text{error}\big(\gamma(\boldsymbol{x}), \omega(\boldsymbol{x})\big) p(\omega, \boldsymbol{x}) d\boldsymbol{x}.$$

  - It cannot be computed but it can be estimated:

- Testing error:

$$E_{\text{test}} = \frac{1}{N'}\sum_{i=1}^{N'} \text{error}\big(\gamma(\boldsymbol{x}_i), \omega(\boldsymbol{x}_i)\big).$$

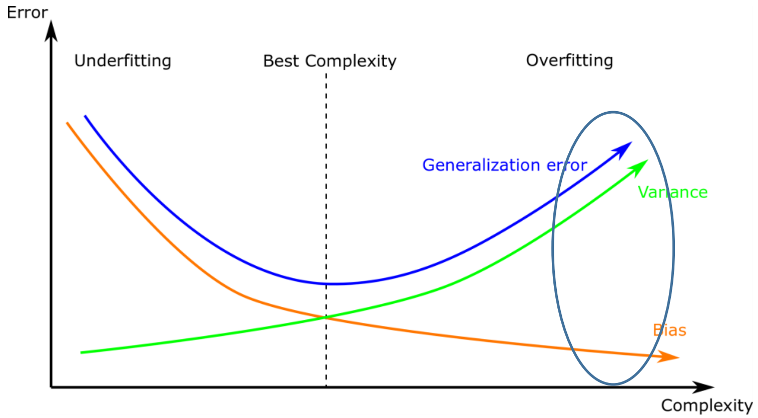
  - $E_{\text{gen}} = \lim_{N' \to \infty} E_{\text{test}}$

Generalization Error of a Decision Tree

- For squared error loss, the generalization error can be decomposed as:
  - $E_{\text{gen}}\big(\gamma(X)\big) = \text{noise}(X) + \text{bias}(X)^2 + \text{var}(X)$
- The generalization error is minimized when finding a good balance between bias and variance.

Generalization Error of a Decision Tree

- Decision trees usually have
  - Low bias
  - High variance

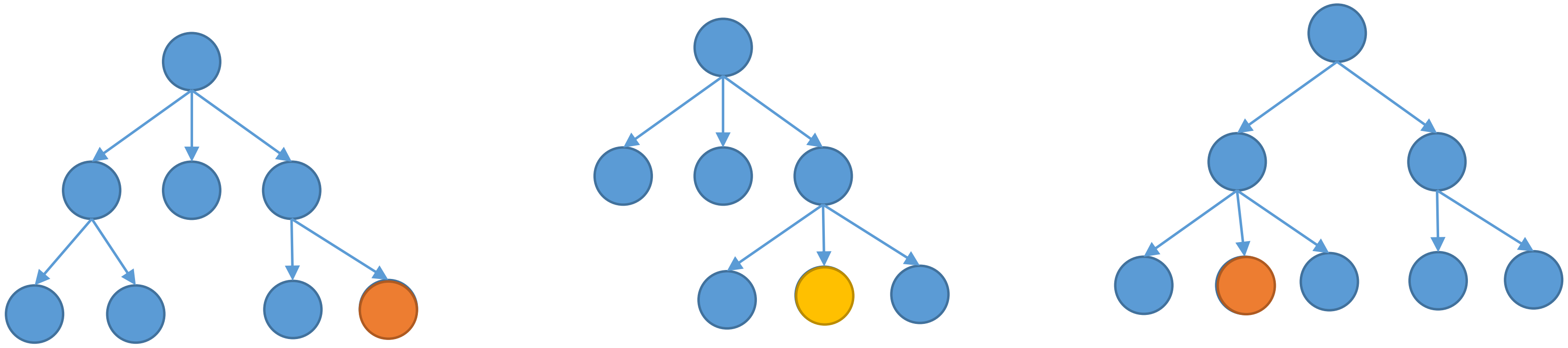


✓Combine the predictions of several trees into a single model.

Random Forest

Random forest

- It constructs multiple *decision trees*
- The final decision is made based on the majority votes of all trees.



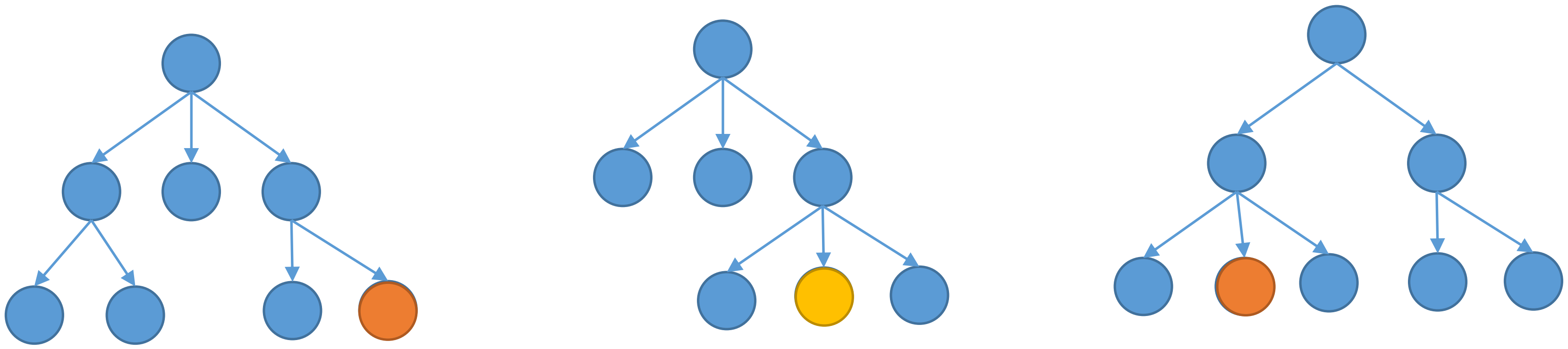
Random forest

- Define the number of trees B
- For $b = 1$ to B
  - Draw a bootstrap sample N^* from the original data set N.
    - *Bootstrap: random sampling with replacement.
  - Grow a *decision* tree using the N^* samples. For each node, repeat:
    - Select l^* attributes at random from the l attribute.
    - Using *Normalized* Impurity Decrease, split the node into children nodes.
- Output the ensemble of trees.

Using random forest for classification

- Given an unseen sample that is represented by $\boldsymbol{x}$
- Let $\gamma_b(\boldsymbol{x})$ is the class prediction of $\boldsymbol{x}$ by the b tree,

$$\gamma(\boldsymbol{x}) = \underset{j}{\operatorname{argmax}} \left| \gamma_b(\boldsymbol{x}) = \omega_j \right|_{b=1}^{B}$$



Generalization Error of an ensemble of Decision Trees

- For squared error loss, the generalization error can be decomposed as:
  - $E_{\text{gen}}\big(\gamma(X)\big) = \text{noise}(X) + \text{bias}(X)^2 + \text{var}(X)$
- Notes:
  - The bias of the ensemble is identical to the bias of a randomized tree but higher than the bias of a non-randomized tree.
  - Stronger randomization: $\text{var}(X) \to 0$
  - Weaker randomization: $\text{var}(X) \to$ the variance of a non-randomized tree.

✓Randomization increases bias but decreases the variance of the ensemble of trees.

✓Find the right bias-variance trade-off.

Advantages

- Better accuracy than a decision tree.
- Robustness to outliers and missing data.
- Robustness to irrelevant attributes.
- Non-parametric (completely random)
- Invariant to feature scaling and types.
- Reduces overfitting.
- High accuracy, especially for large data sets.

- Interpretability ???

Attribute importance scores

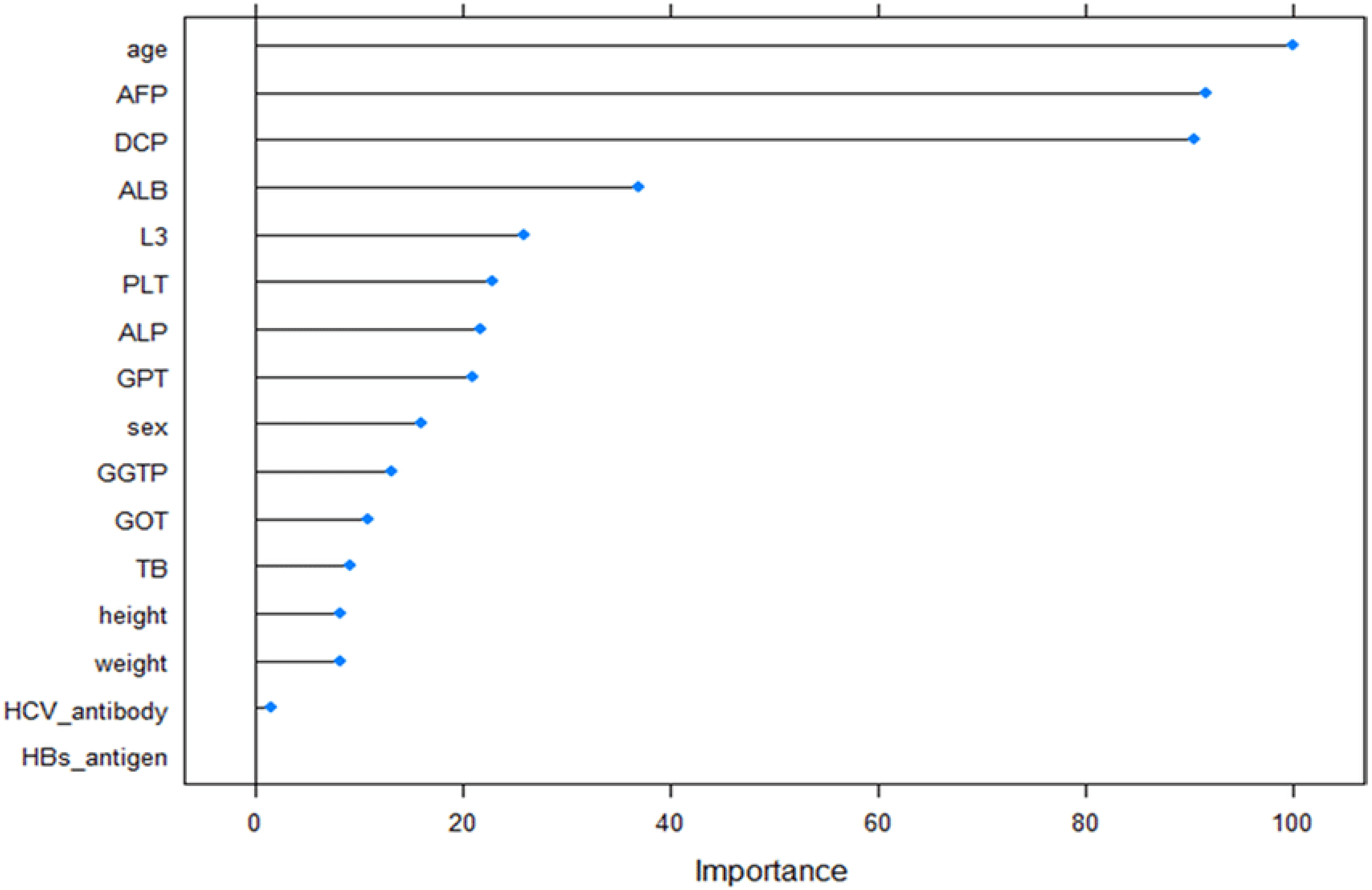
- Mean Decrease of Accuracy (MDA)
    - Consider the out-of-bag instances (which were not sampled in the bootstrapped set). Of course, we may use a separate set.
    - Consider the corresponding trees.
    - Permute the value of the attribute (to be assessed) with random noise.
    - Consider an evaluation metric (e.g. accuracy)
    - Compute the mean decrease accuracy over all corresponding trees.

➢ The attribute is important when the MDA is high

Attribute importance scores

- Mean Decrease in Impurity (MDI):
  - The importance of an attribute x_u is measured as:
    - $Imp(x_j) = \frac{1}{B}\sum_{b=1}^{B}\sum_{t:b(t)=x_u} P(t) * \Delta I(t)$,
    - Where $P(t) = \frac{N_t}{N}$
  - The intuition is that an attribute is important when:
    - It decreases a lot of impurities
    - It is used to split nodes with many instances.
    - It is used many times.
- Compared to MDA, MDI is widely used because:
  - It is faster and easier to compute.
  - Experiences showed that it correlates well with MDA.

Example of MDI



Source: Sato, M., Morimoto, K., Kajihara, S., Tateishi, R., Shiina, S., Koike, K., & Yatomi, Y. (2019). Machine-learning Approach for the Development of a Novel predictive Model for the Diagnosis of Hepatocellular Carcinoma. *Scientific reports*, *9*.

Computational complexity

	Training	Prediction
Decision Tree	$O(l * N\log(N))$	$O(l)$
Random Forest	$O(T * \hat{l} * \widehat{N}\log(\widehat{N}))$	$O(T * \hat{l})$
Extra Tree	$O(T * \hat{l} * N\log(N))$	$O(T * \hat{l})$

- $\hat{l}$: the number of variables randomly drawn at each node.
- T: the number of trees
- $\widehat{N} = 0.632 \times N$

Summary

Summary

- Decision Trees
  - Impurity
  - Impurity decrease
  - Normalized impurity decrease
  - Pruning
- Generalizibility
  - Overfitting
  - Underfitting
- Mean Decrease in Impurity
- Random Forest

Thank you!

Zeyd Boukhers

E-mail: Boukhers@uni-koblenz.de
Phone: +49 (0) 261 287-2765
Web: Zeyd.Boukhers.com

University of Koblenz-Landau
Universitätsstr. 1
56070 Koblenz