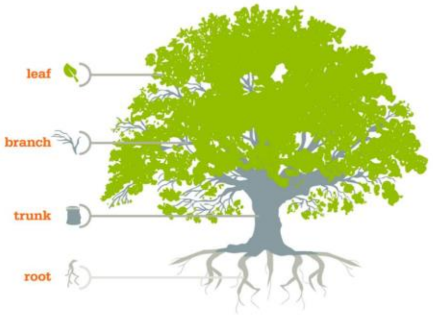
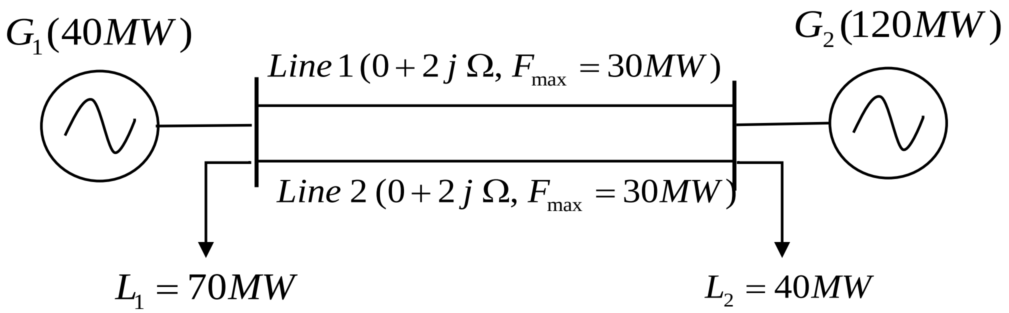
Big Data Technology and its Applications



Decision Tree

张宁 ningzhang@tsinghua.edu.cn

A problem in power system

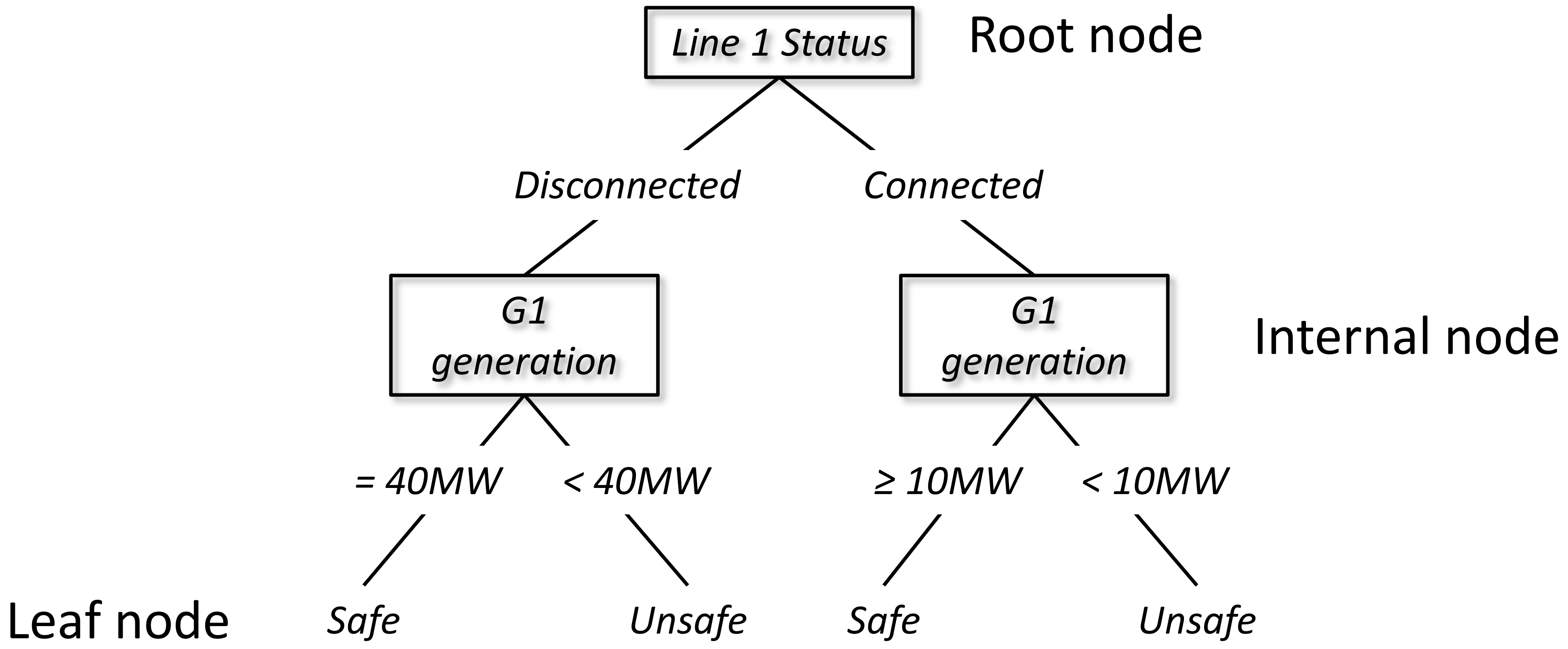


- Given a set of operation state and assuming loads are constant, how to judge whether the power system is safe?

ID	G1 generation	G2 generation	Line 1 status	Safe or not
1	0	110	Connected	**N**
2	20	90	Connected	**Y**
3	40	70	Connected	**Y**
4	0	110	Disconnected	**N**
5	20	90	Disconnected	**N**
6	40	70	Disconnected	**Y**

- Lots of similar problems in power systems.

Decision Tree for power system safety



Each internal node: test one feature X_i
Each branch from a node: select a division for X_i
Each leaf node: predict Y (or $P(Y|X \in \text{leaf})$)

From Tom Mitchell, Machine Learning, 1997.

Appropriate problems for decision tree

- Instances are represented by feature-value pairs
- The target function has discrete output values
- The training data may contain errors
- The training data may contain missing feature values

From Tom Mitchell, Machine Learning, 1997.

Decision Tree Problem Setting

- Set of possible instances X
  - each instance is a feature vector
  - e.g., *<Line 1 Status=Connected, G1 generation=40MW>*
- Unknown target function $f : X \longrightarrow Y$
  - *Y* is discrete valued
- Set of function hypotheses $H = \{h | \ h : X \rightarrow Y\}$
  - each hypothesis *h* is a decision tree
  - trees sorts instance *x* to leaf, which assigns $y \in Y$

Input: Training examples $\{ \ x_i, y_i \ \ | \ x_i \in X, y_i \in Y\}$

Output: Hypothesis $h \in H$ that best approximates target function

From Tom Mitchell, Machine Learning, 1997.

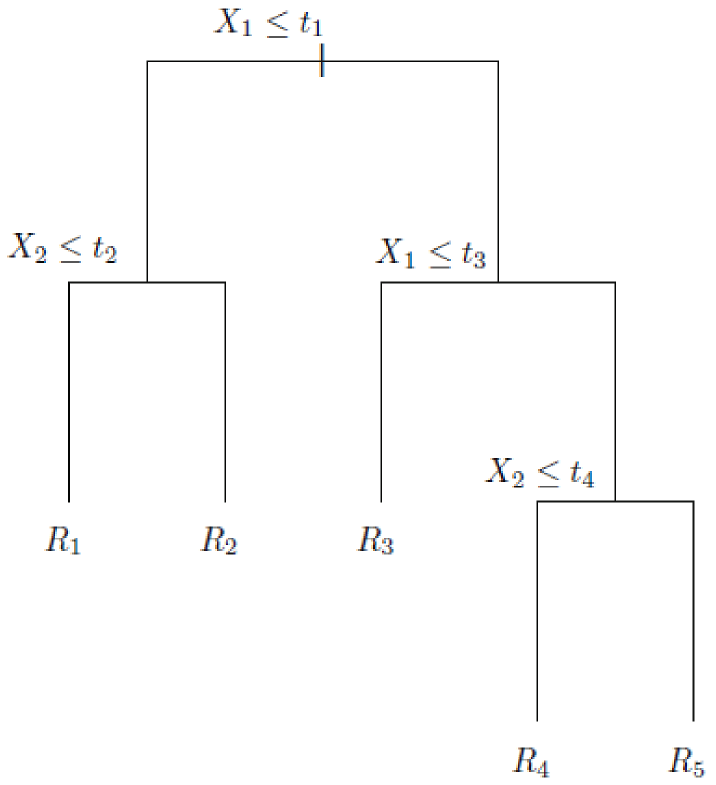
Type of decision tree

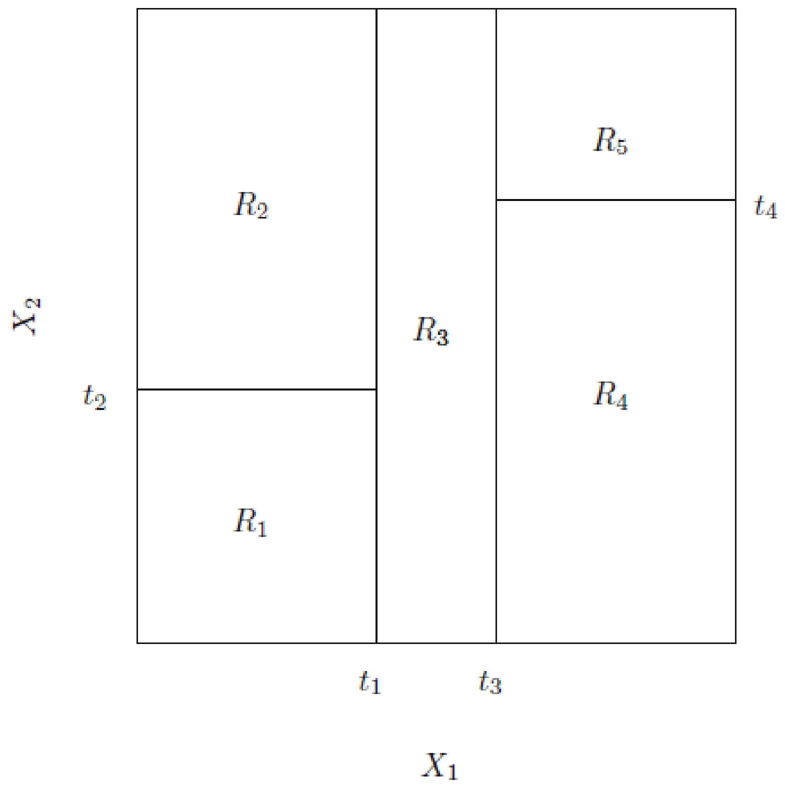
- Classification tree
  - Classification trees are designed for dependent variables that take a finite number of unordered values.
  - Prediction error measured in terms of misclassification cost.

- Regression tree
  - Regression trees are for dependent variables that take continuous or ordered discrete values.
  - Prediction error typically measured by the squared difference between the observed and predicted values.

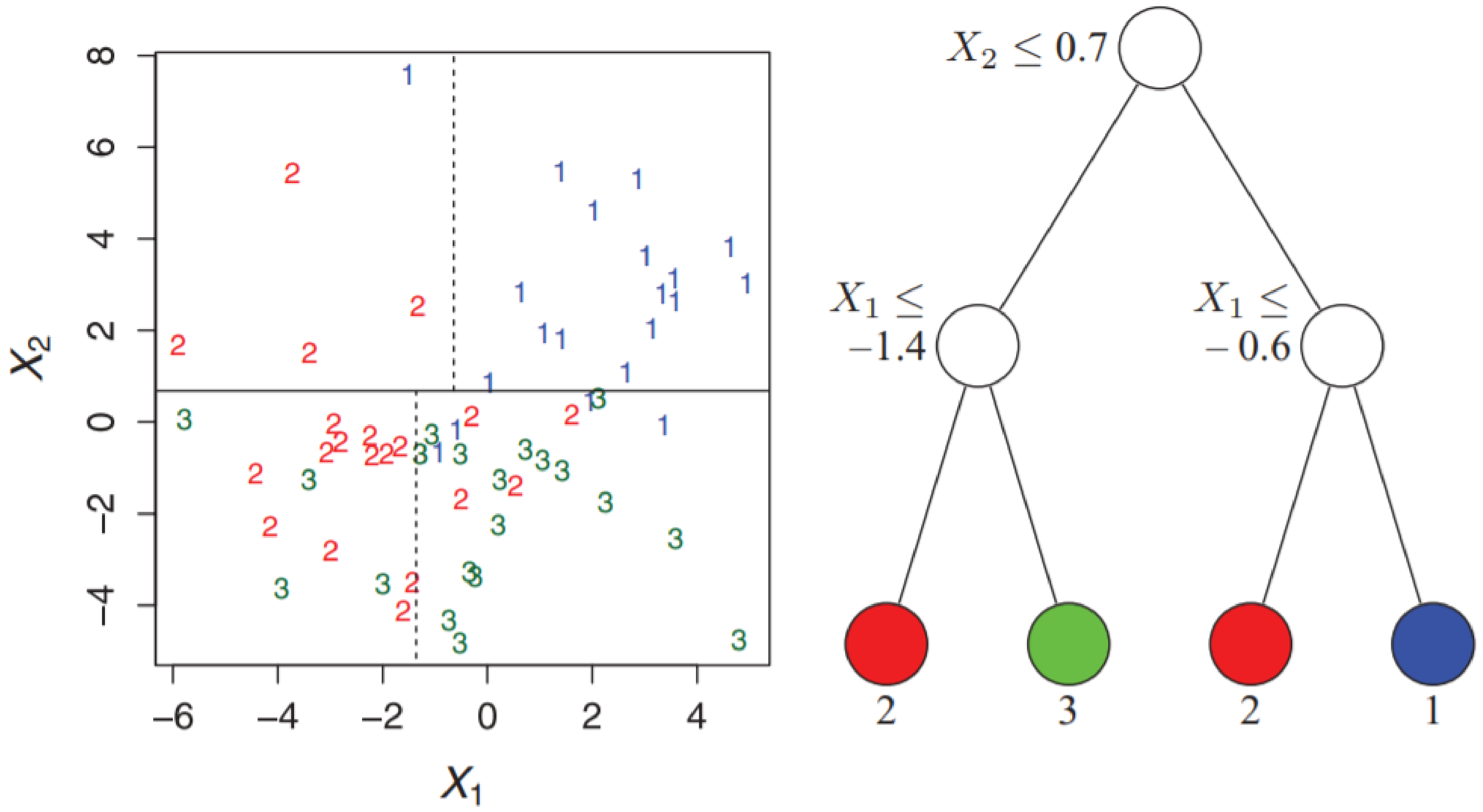
Binary Decision Tree

- Binary Decision Tree on $X_1, X_2 \mid X_1, X_2 \in \mathbf{R}$
- If *R1-R5* learn ordered or continuous value, it is regression tree
- If *R1-R5* learn unordered discrete value, it is classification tree





Classification Tree Example



From Loh W Y. Classification and regression trees, 2011.

Basic algorithm of decision tree

Principle: Divide and Conquer （分而治之）

Start: *node*=Root

Main Loop:

1. Choosing “best” decision feature A for next *node*
2. For each value of A, create new descendant *node*
3. Sort Training examples to leaf *nodes*
4. If reaching one of the following criteria:
   - training examples perfectly classified
   - reaching the maximum depth
   - no examples in the current leaf
   - all examples have same feature value or no candidate feature

   Then stop, Else iterate over new leaf *nodes*

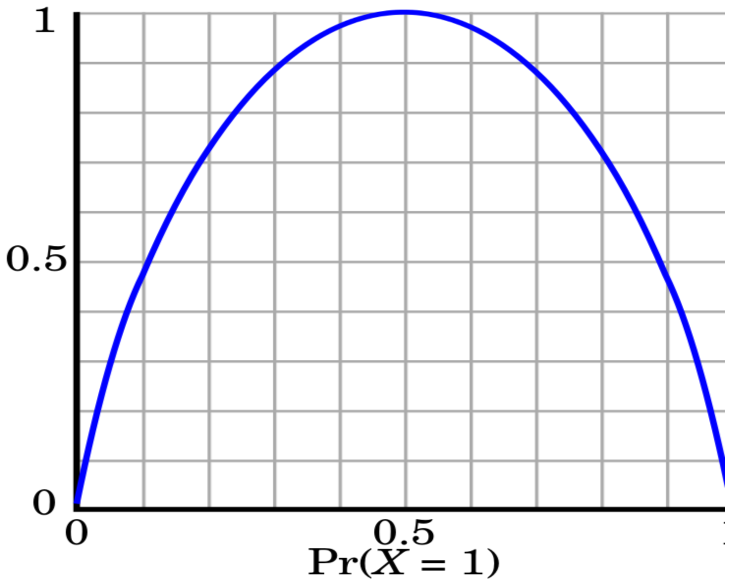
Key question： which is the best splitting feature for next node?

Information Entropy

- Principle: the best splitting feature make the child nodes have highest “purity”
- Decision tree uses purity index like **information entropy** to choose best splitting feature.

$$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$$

Small information entropy means high purity



Perspective from information theory: Entropy is the expected number of bits needed to encode a value of random variable

Other impurity indices

- Different decision tree algorithms use different impurity indices

Decision Tree	Impurity index
CART (1984)	Gini index
ID3(1986)	Information Gain
C4.5(1993)	Information Ratio

ID3: Information gain

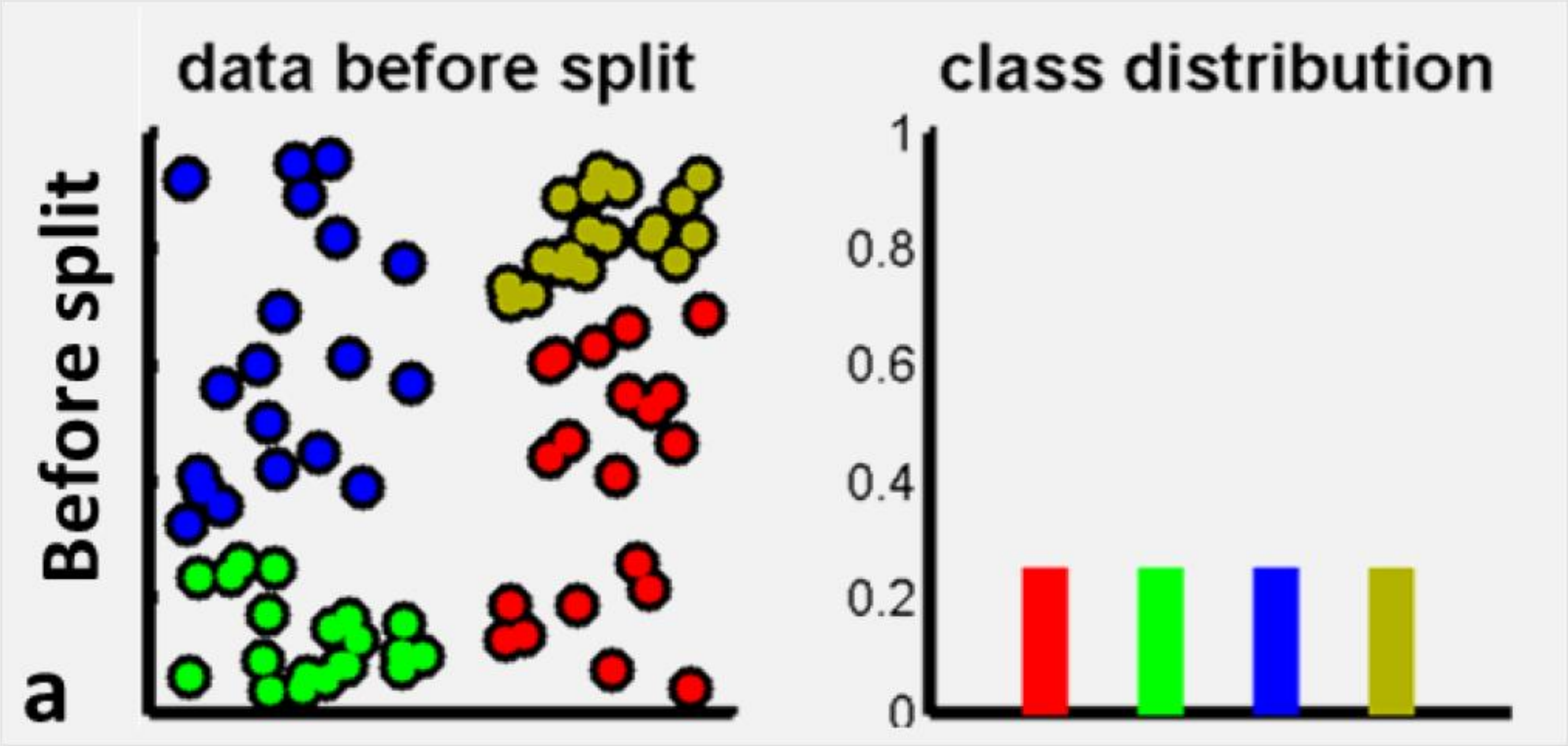
- ID3 algorithm [Quilan 1986] uses information gain to choose splitting feature
- Information gain is the entropy increase after splitting a feature

$$\mathrm{Gain}(D,a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{\left|D^v\right|}{|\ D\ |} \mathrm{Ent}\ \ D^v$$

- Choosing feature with largest information gain

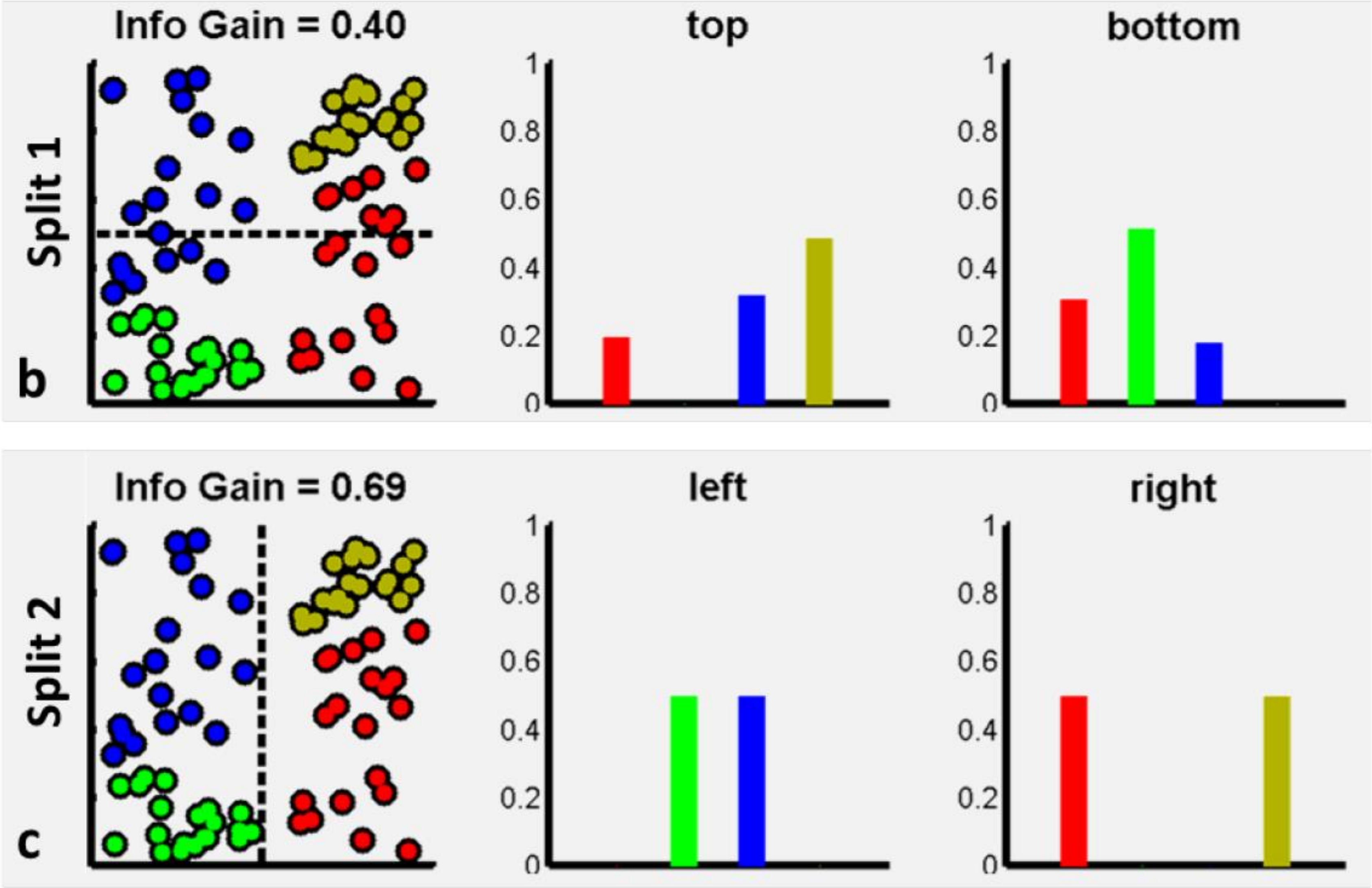
$$a_* = \underset{a \in A}{\arg\max}\, \mathrm{Gain}(D,a)$$

ID3: Information gain



From Criminisi et al. MSR-TR-2011-114, 28 October 2011.

ID3: Information gain



From Criminisi et al. MSR-TR-2011-114, 28 October 2011.

Information gain example with discrete feature

Example: Considering the discrete feature Line1 status

- Calculate the entropy on whole dataset

$$\mathrm{Ent}(D) = -P(\mathrm{safe})\log_2 P(\mathrm{safe}) - P(\mathrm{unsafe})\log_2 P(\mathrm{unsafe})$$
$$= -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1$$

- Calculate the entropy on split dataset

$$\mathrm{Ent}\left(D^{\mathrm{dis}}\right) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$$
$$\mathrm{Ent}\left(D^{\mathrm{con}}\right) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$$

- Calculate information gain

$$\mathrm{Gain}(D, \mathrm{line1}) = \mathrm{Ent}(D) - \frac{\left|D^{\mathrm{dis}}\right|}{|D|}\mathrm{Ent}\left(D^{\mathrm{dis}}\right) - \frac{\left|D^{\mathrm{con}}\right|}{|D|}\mathrm{Ent}\left(D^{\mathrm{con}}\right)$$
$$= 1 - \frac{3}{6}\times 0.918 - \frac{3}{6}\times 0.918$$
$$= 0.082$$

ID	Line 1	Security
1	Connected	N
2	Connected	Y
3	Connected	Y
4	Disconnected	N
5	Disconnected	N
6	Disconnected	Y

Decision Tree with continuous feature

- Decision tree uses bi-partition method to discretize the continuous feature
- Given a continuous feature a with ascending values in dataset *D.* $a^1, a^2, \dots, a^n$
- Consider the following n-1 candidates t to split the whole dataset D as

$$D^-, D^+ \quad \text{where } a \in D^- < t \;\&\; a \in D^+ > t$$

$$t_i = \left\{ \frac{a^i + a^{i+1}}{2} | 1 \leqslant i \leqslant n-1 \right\}$$

- Then the information gain with continuous feature a can be calculated as:

$$\begin{aligned} \mathrm{Gain}(D, a) &= \max_{t \in T_a} \mathrm{Gain}(D, a, t) \\ &= \max_{t \in T_a} \mathrm{Ent}(D) - \sum_{\lambda \in \{-,+\}} \frac{\left|D_t^\lambda\right|}{|\, D \,|} \mathrm{Ent}\ \ D_t^\lambda \end{aligned}$$

Information gain example with continuous feature

Example: Considering the continuous feature G1 generation

- Determine split candidates

$$t \in \{10, 30\}$$

- Calculate the information gain on every candidates

$$\mathrm{Gain}(D, \mathrm{G1{,}10}) = 1 - \frac{2}{6} \times 0 - \frac{4}{6} \times 0.811 = 0.460$$

$$\mathrm{Gain}(D, \mathrm{G1{,}30}) = 1 - \frac{4}{6} \times 0.811 - \frac{2}{6} \times 0 = 0.460$$

- Determine the maximum information gain

$$\begin{aligned}\mathrm{Gain}(D, \mathrm{G1}) &= \max\{\mathrm{Gain}(D, \mathrm{G1{,}10}), \mathrm{Gain}(D, \mathrm{G1{,}30})\} \\ &= 0.460\end{aligned}$$

ID	G1	Security
1	0	N
2	20	Y
3	40	Y
4	0	N
5	20	N
6	40	Y

Compared with Line 1 status, G1 generation is better feature for splitting

Question： What if we choose both candidates？

Information gain example with continuous feature

Example: Considering the continuous feature G1 generation

- Determine split candidates

$$t \in \{10,\, 30\}$$

- Calculate the information gain if choose both candidates

$$\mathrm{Gain}(D, \mathrm{G1}, \{10,\, 30\})$$
$$= 1 - \frac{2}{6} \times 0 - \frac{2}{6} \times 0 - \frac{2}{6} \times \left(-\frac{1}{2}\log_2 \frac{1}{2} - \frac{1}{2}\log_2 \frac{1}{2} \right) = 0.67$$

- Which feature has the maximum information gain?

ID	G1	Security
1	0	N
2	20	Y
3	40	Y
4	0	N
5	20	N
6	40	Y

Compared with $\mathrm{Gain}(D, \mathrm{G1}, 10), \mathrm{Gain}(D, \mathrm{G1}, 30)$, $\mathrm{Gain}(D, \mathrm{G1}, \{10,\, 30\})$ is better feature for splitting.

Question： What is the limitation of Information gain?

C4.5： Gain ratio

- Information gain prefers the feature with more candidate values
- C4.5 algorithm [Quinlan 1993] uses the Gain Ratio to choose splitting feature to mitigate the limitation

$$\text{Gain\ \ ratio}(D,a)=\frac{\text{Gain}(D,a)}{\text{IV}(a)}$$

- IV is intrinsic value of a feature. The feature with more candidate values has larger intrinsic value

$$\text{IV}(a)=-\sum_{v=1}^{V}\frac{\left|D^v\right|}{|\ D\ |}\log_2\frac{\left|D^v\right|}{|\ D\ |}$$

- Choose the feature with maximum Gain ratio

Gain ratio example

Example: Considering the discrete feature Line1 status

- Calculate information gain

$$\mathrm{Gain}(D,\mathrm{line1}) = \mathrm{Ent}(D) - \frac{\left|D^{\mathrm{dis}}\right|}{|\,D\,|}\mathrm{Ent}\ \ D^{\mathrm{dis}}\ \ - \frac{\left|D^{\mathrm{con}}\right|}{|\,D\,|}\mathrm{Ent}\ \ D^{\mathrm{con}}$$
$$= 1 - \frac{3}{6}\times 0.918 - \frac{3}{6}\times 0.918$$
$$= 0.082$$

- Calculate intrinsic value

$$\mathrm{IV}(\mathrm{line1}) = -\frac{\left|D^{con}\right|}{|\,D\,|}\log_2\frac{\left|D^{con}\right|}{|\,D\,|} - \frac{\left|D^{dis}\right|}{|\,D\,|}\log_2\frac{\left|D^{dis}\right|}{|\,D\,|}$$
$$= -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1$$

- Calculate gain ratio

$$\mathrm{Gain\ \ ratio}(D,\mathrm{line1}) = \frac{\mathrm{Gain}(D,\mathrm{line1})}{\mathrm{IV}(\mathrm{line1})} = 0.082$$

ID	Line 1	Security
1	Connected	N
2	Connected	Y
3	Connected	Y
4	Disconnected	N
5	Disconnected	N
6	Disconnected	Y

Gain ratio example with continuous feature

Example: Considering the continuous feature G1 generation

- Determine split candidates

$$t \in \{10,\ 30\}$$

- Calculate the gain ratio on every candidates

$$\text{Gain\_ratio}(D,\text{G1},10) = \frac{1 - \frac{2}{6}\times 0 - \frac{4}{6}\times 0.811}{-\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6}} = 0.500$$

$$\text{Gain\_ratio}(D,\text{G1},30) = \frac{1 - \frac{4}{6}\times 0.811 - \frac{2}{6}\times 0}{-\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6}} = 0.500$$

- Determine the maximum gain ratio

$$\text{Gain\_ratio}(D,\text{G1}) = \max\{\ \text{Gain\_ratio}(D,\text{G1},10),\ \text{Gain\_ratio}(D,\text{G1},30)\} = 0.500$$

Compared with Line 1 status, G1 generation is better feature for splitting

ID	G1	Security
1	0	N
2	20	Y
3	40	Y
4	0	N
5	20	N
6	40	Y

Gain ratio example with continuous feature

Example: Considering the continuous feature G1 generation

- Determine split candidates

$$t \in \{10,\ 30\}$$

- Calculate the gain ratio on every candidates

$$\text{Gain\_ratio}(D,\text{G1},\{10,30\}) = \frac{1 - \frac{2}{6}\times 0 - \frac{2}{6}\times 0 - \frac{2}{6}\times\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right)}{-\frac{2}{6}\log_2\frac{2}{6} - \frac{2}{6}\log_2\frac{2}{6} - \frac{2}{6}\log_2\frac{2}{6}} = 0.42$$

- Determine the maximum gain ratio

ID	G1	Security
1	0	N
2	20	Y
3	40	Y
4	0	N
5	20	N
6	40	Y

$$\text{Gain\_ratio}(D,\text{G1}) = \max\{\ \text{Gain\_ratio}(D,\text{G1},10),\ \text{Gain\_ratio}(D,\text{G1},30),\ \text{Gain\_ratio}(D,\text{G1},\{10,30\})\} = 0.50$$

Compared with $\text{Gain}(D,\text{G1},\{10,\ 30\})$, $\text{Gain}(D,\text{G1},10)$ & $\text{Gain}(D,\text{G1},30)$ are better features for splitting

CART: Gini index

- CART algorithm [Breiman, 1984] uses gini index to choose splitting feature

$$\mathrm{Gini}(D) = \sum_{k=1}^{|\mathcal{Y}|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2$$

- Gini index represents the probability of randomly choosing two samples with different labels. Small Gini index means large purity.

$$a_* = \arg\min_{a \in A} \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Gini}\ D^v$$

Gini index example

Example: Considering the discrete feature Line1 status

- Calculate the Gini index on split dataset

$$\mathrm{Gini}(D^{con}) = 1 - P(\mathrm{safe})^2 - P(\mathrm{unsafe})^2$$

$$= 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = 0.444$$

$$\mathrm{Gini}(D^{dis}) = 1 - P(\mathrm{safe})^2 - P(\mathrm{unsafe})^2$$

$$= 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.444$$

- Calculate weighted Gini index

$$\mathrm{Gini}(D, \mathrm{line1}) = \frac{3}{6}\mathrm{Gini}(D^{con}) + \frac{3}{6}\mathrm{Gini}(D^{dis}) = 0.444$$

- What is the Gini index of feature G1 generation?

ID	Line 1	Security
1	Connected	N
2	Connected	Y
3	Connected	Y
4	Disconnected	N
5	Disconnected	N
6	Disconnected	Y

Gini index example

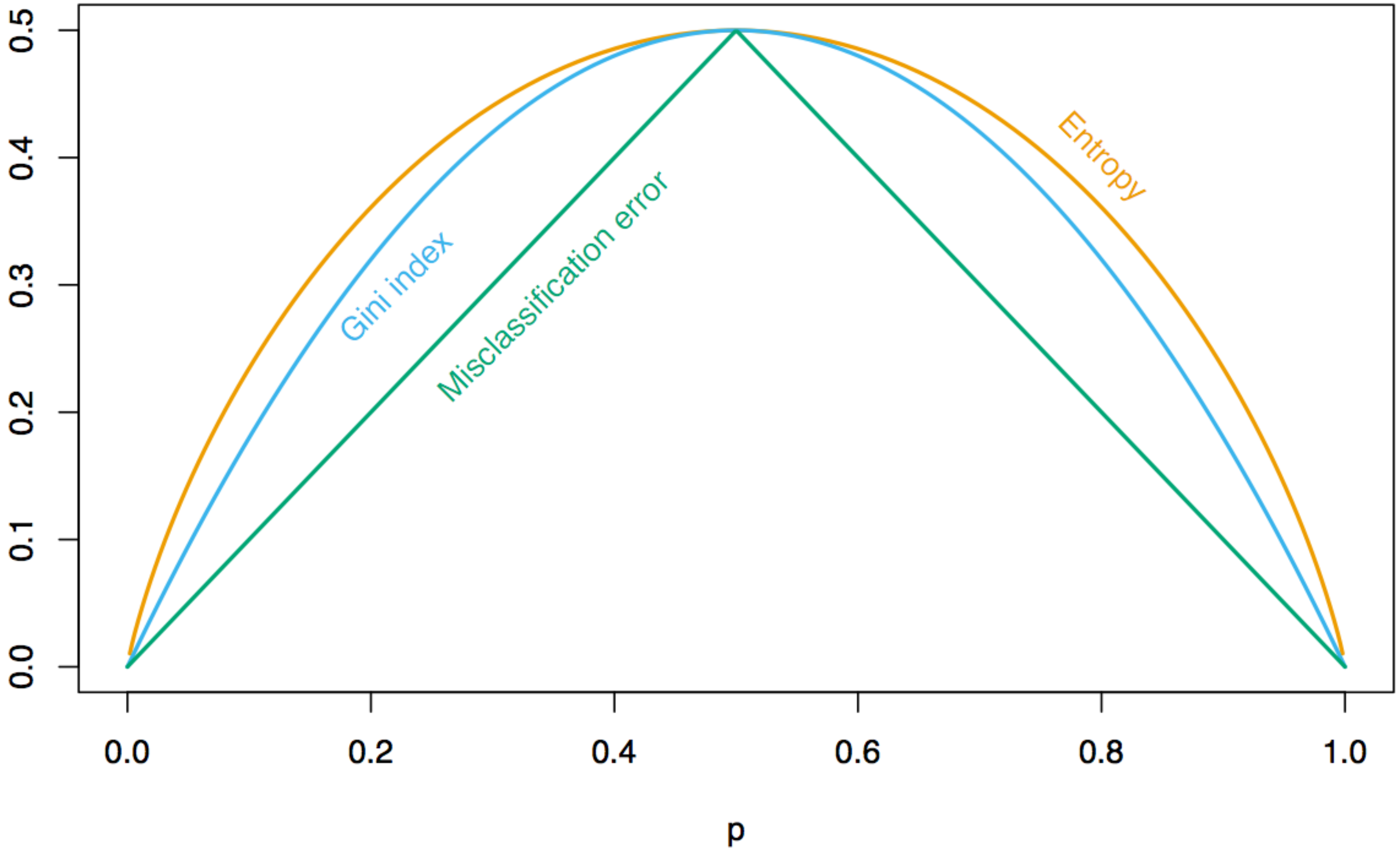
$$\mathrm{Gini}(D, \mathrm{G1}, 10)$$

$$\mathrm{Gini}(D, \mathrm{G1}, \{10,\ 30\})$$

ID	Line 1	Security
1	Connected	**N**
2	Connected	**Y**
3	Connected	**Y**
4	Disconnected	**N**
5	Disconnected	**N**
6	Disconnected	**Y**

Gini index vs information entropy

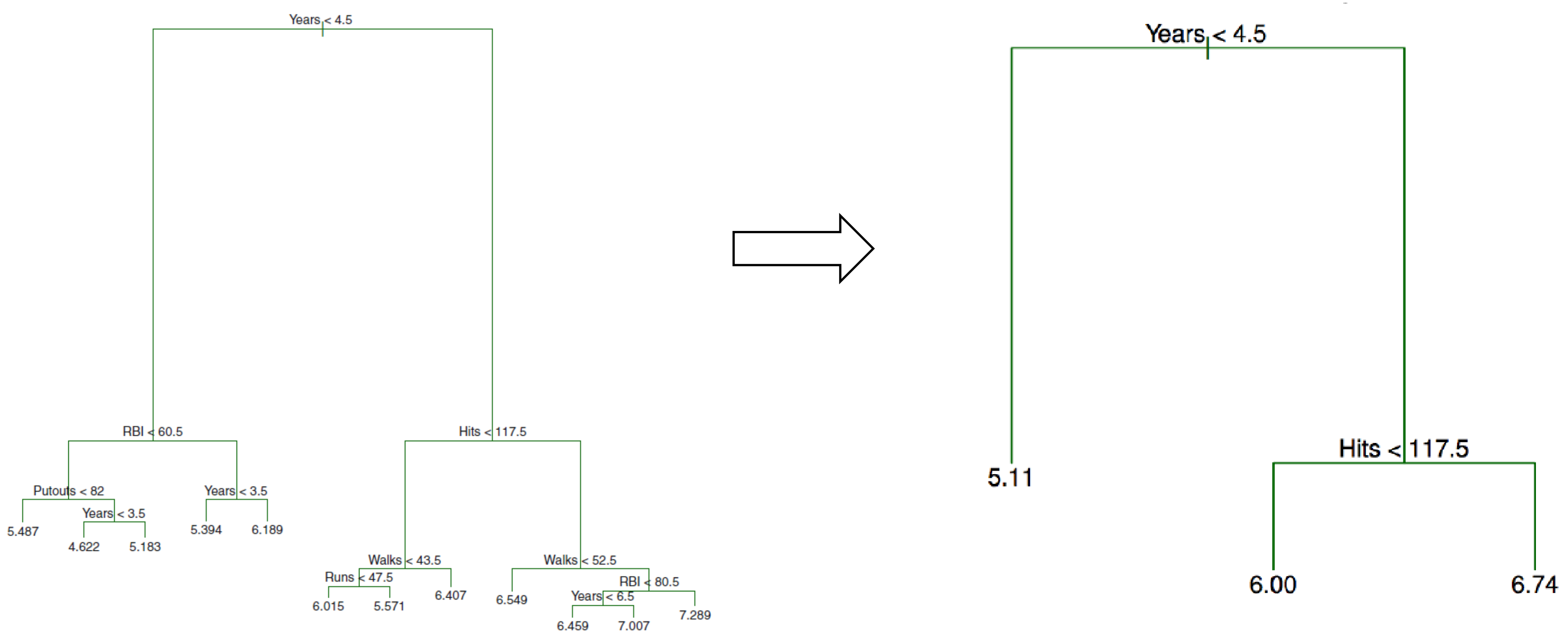
- Gini index decreases faster than entropy



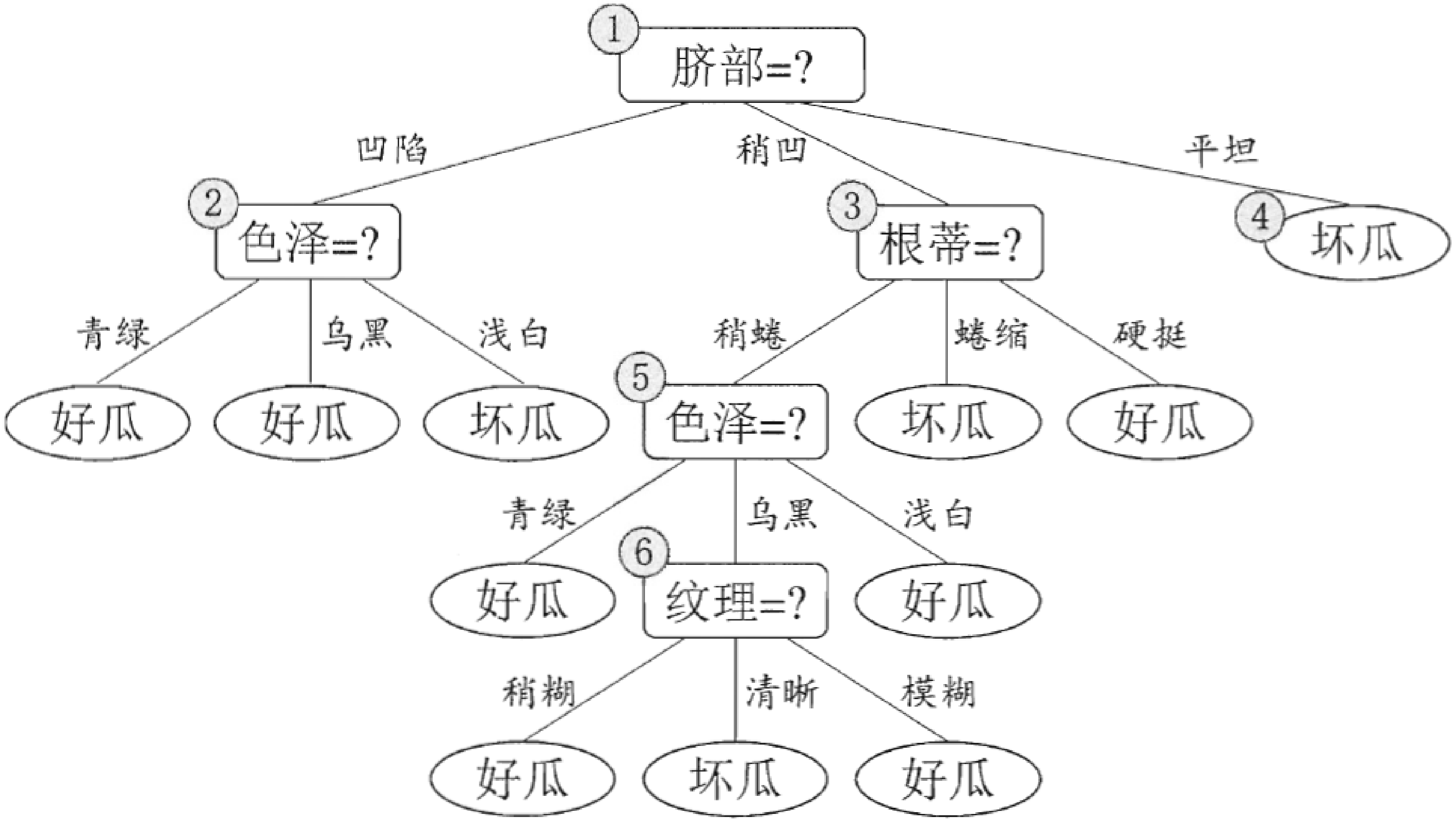
From DS-GA 1003

Pruning

- Pruning is to prevent overfitting and reduce the generalization error of decision tree
- How to measure generalization error?
  - Test decision tree on validation set
- Two greedy strategies: Prepruning and Postpruning

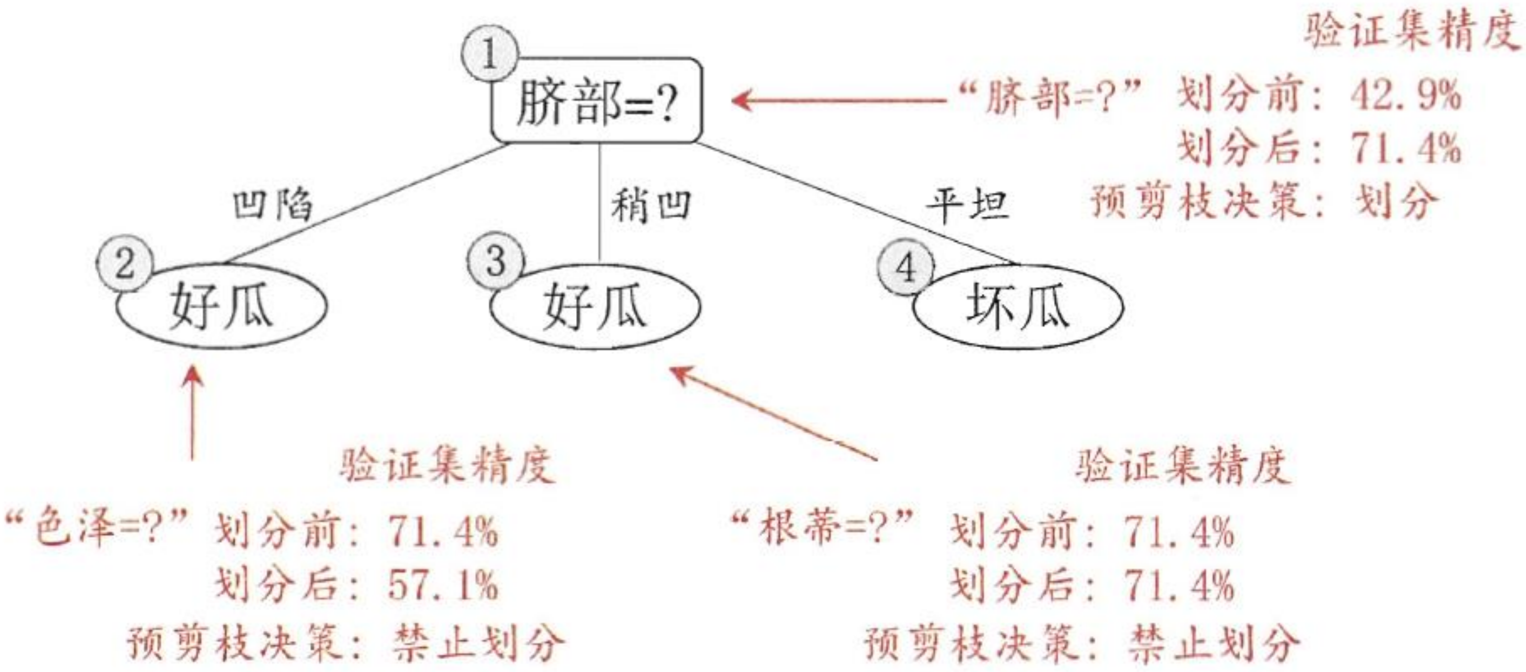


Before pruning



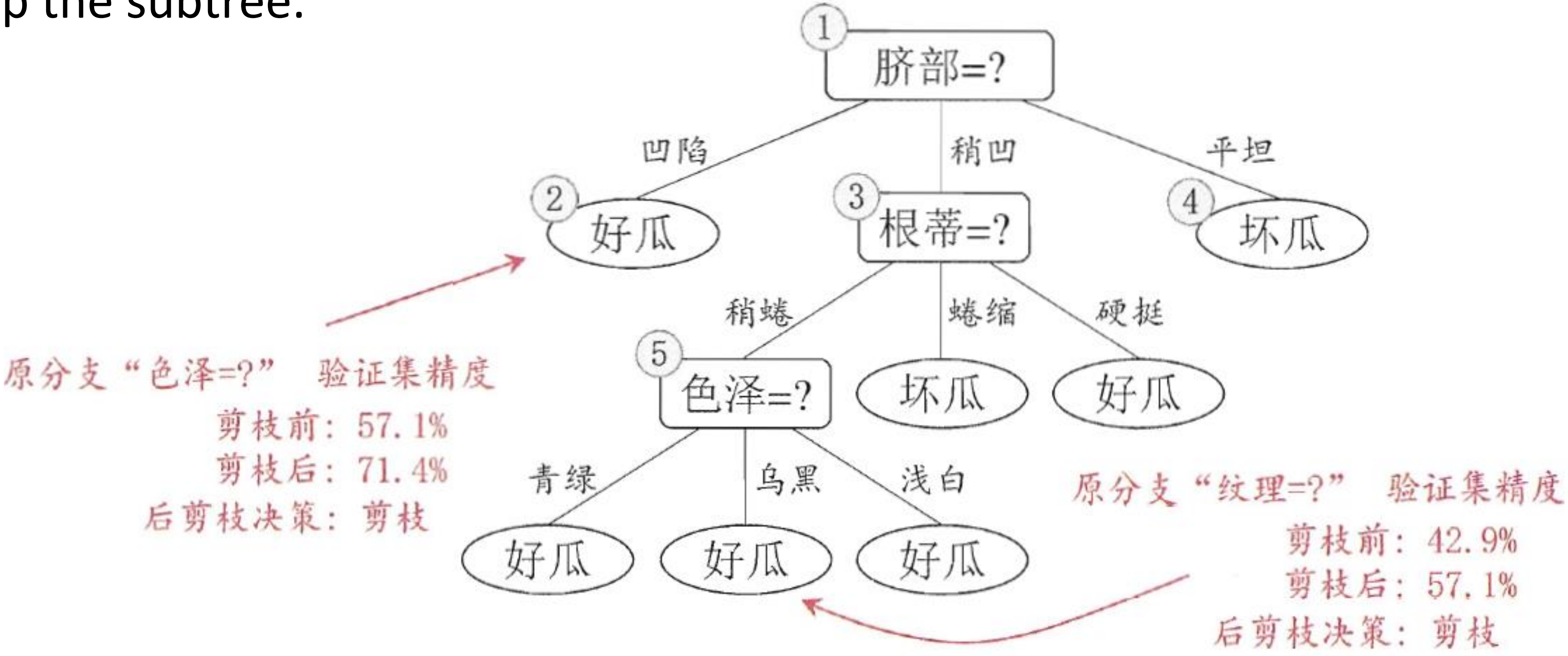
Prepruning

- Use the impurity index to choose split feature
- Test the generalization error before and after splitting current node
- If the splitting current node reduces the generalization precision, then stop split, else continue.



Postpruning

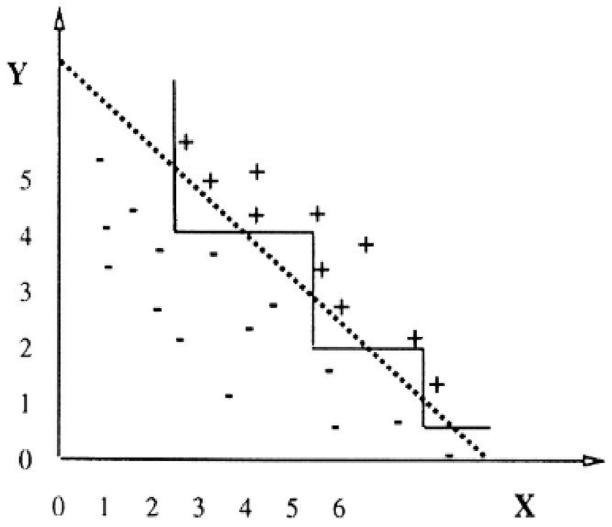
- Train a large tree
- Testing the generalization error before and after replace a subtree with leaf node.
- If the replacement reduces the generalization error, Then remove the subtree, else keep the subtree.



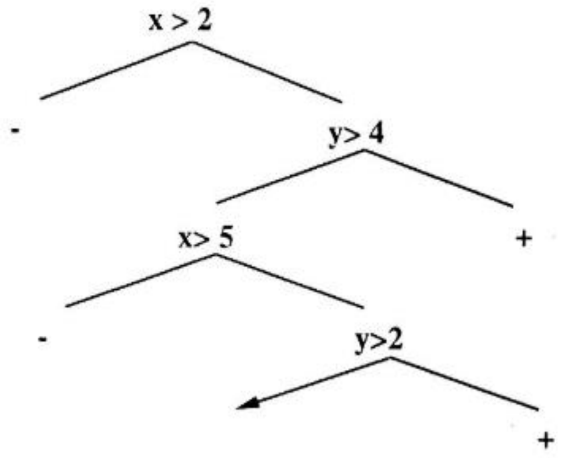
Prepruning vs Postpruning

Pruning	Advantages	Disadvantages
Prepruning	• Avoid overfitting • Less computation time	• High underfitting risk
Postpruning	• Avoid overfitting • Less underfitting risk	• More computation time

Oblique decision tree



$n \cdot q$

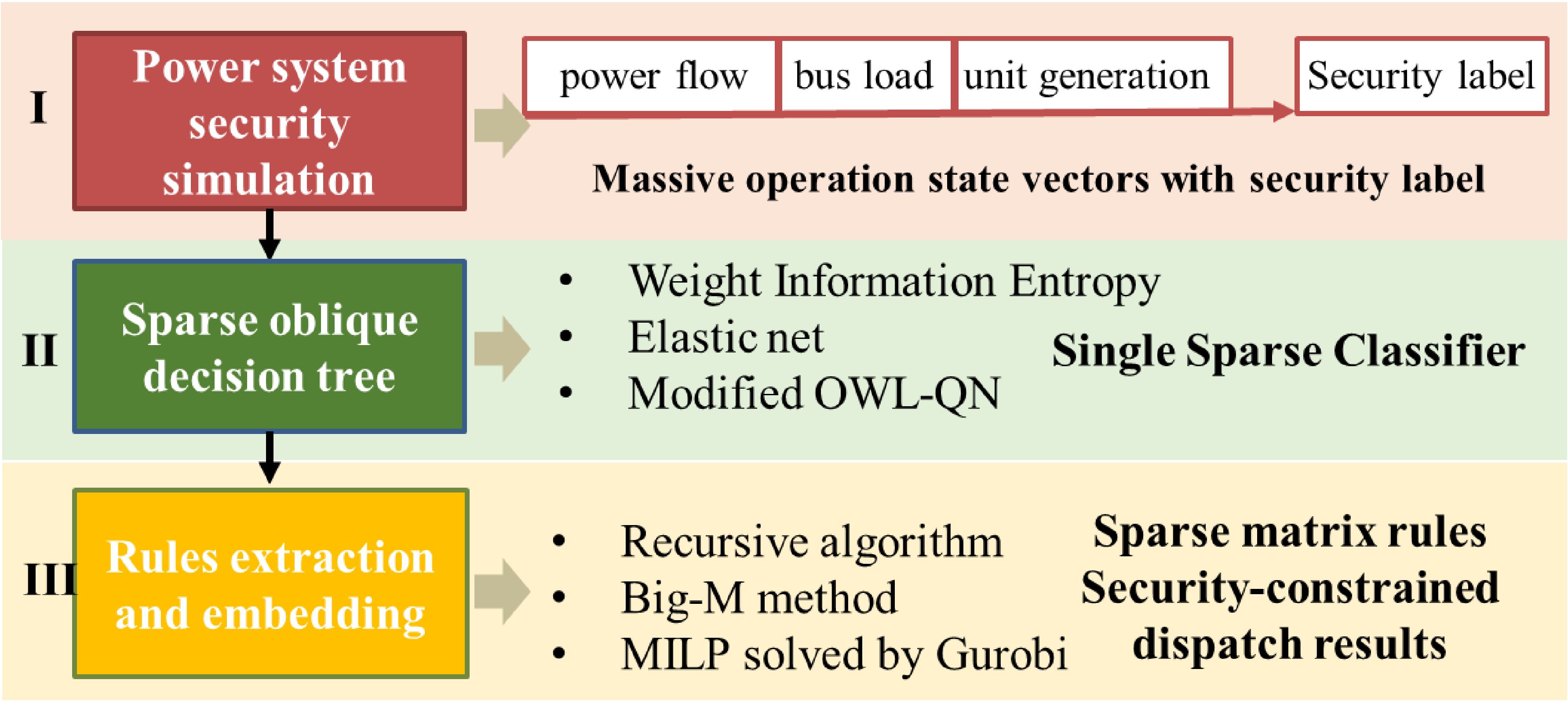


$2^q \cdot \binom{n}{q}$

$x + y \leqslant 8$?

	Univariate Decision Tree	**Oblique Decision Tree**
Advantage	• Easy to understand • Easy to train (fast)	• Simple rule (lower depth) • Strong representation • Higher accuracy
Disadvantage	• Lower accuracy • Complex structure • Easy to affected by samples	• Harder to understand • Harder to train-NP hard

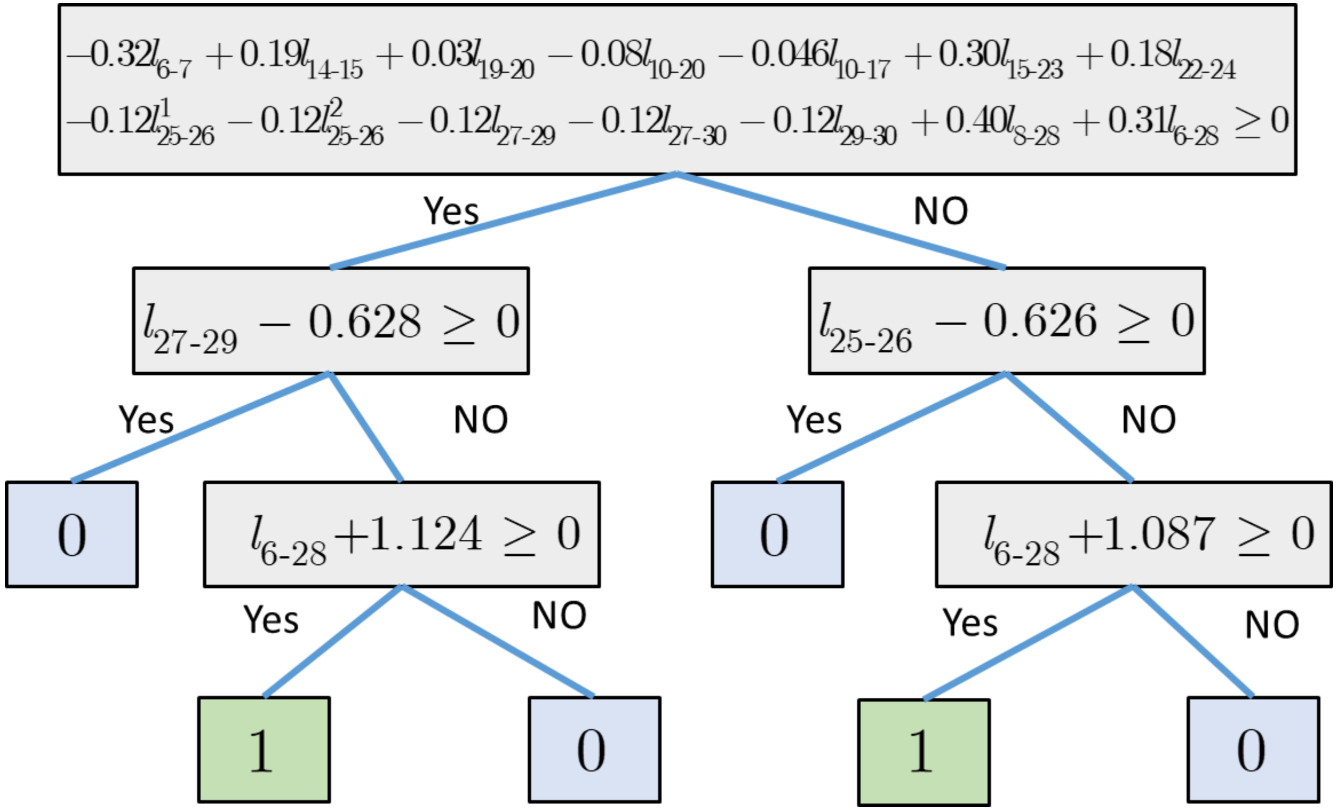
Oblique decision tree application on power system security rules extraction



Q. Hou, N. Zhang, D. S. Kirschen, E. Du, Y. Cheng, and C. Kang, "Sparse Oblique Decision Tree for Power System Security Rules Extraction and Embedding," *IEEE Trans. Power Syst.*, pp. 1–1, 2020.

N-1 security classification results

- 97% testing accuracy on IEEE-30 bus system with high renewable energy penetration



Q. Hou, N. Zhang, D. S. Kirschen, E. Du, Y. Cheng, and C. Kang, "Sparse Oblique Decision Tree for Power System Security Rules Extraction and Embedding," *IEEE Trans. Power Syst.*, pp. 1–1, 2020.

Decision Trees in Scikit-learn

- Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(criterion='entropy', max_depth=5)
clf.fit(X, y)
```

- Decision Tree Regressor

```
from sklearn.tree import DecisionTreeRegressor
clf = DecisionTreeRegressor(max_depth=5)
clf.fit(X, y)
```

References

- Zhihua Zhou, Machine learning, 2016
- Tom Mitchell, Machine learning, 1997

http://www.cs.cmu.edu/~tom/mlbook.html

- Gareth James, et al. An introduction to statistical learning, 2013

https://faculty.marshall.usc.edu/gareth-james/ISL/ISLR%20Seventh%20Printing.pdf

- A. Criminisi, et al. Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning, 2016

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/decisionForests_MSR_TR_2011_114.pdf

Homework 4

(1) Based on the dataset *hw4_data.csv*, which includes the power system operation states (e.g. active and reactive generation(PV_P, PV_Q), power load(Pl, Ql), bus voltage(Va, Vm), line power flow(Line_Ps, Line_Qs), line power loss(Line_Pl, Line_Ql).) and the small-signal stability states (named '*SSSA*', 1 for safe, 0 for unsafe) of an IEEE 118-bus test system. Please use classification methods to fit *SSSA* by the operation states.

- Basic requirements: Try to find the best classification results (by SVM, DT, and so on).
- Further thinking: The precise awareness of insecurity power system state is usually critical for power system operation, which results in different tolerances of false-negative and false-positive samples. Try to consider this situation in your model.

References on the IEEE 118-bus test system (not on the dataset):
http://labs.ece.uw.edu/pstca/pf118/pg_tca118bus.htm
https://matpower.org/docs/ref/matpower5.0/case118.html

Q&A