法律声明

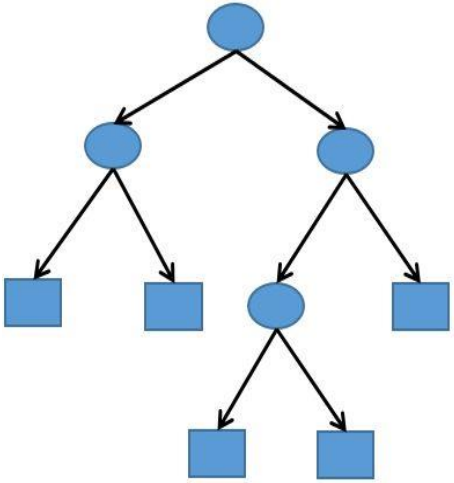
□ 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

关注 小象学院

Machine Learning

Part 4: Classical Machine Learning Models

Zengchang Qin (Ph.D.)



Tree Ensembles

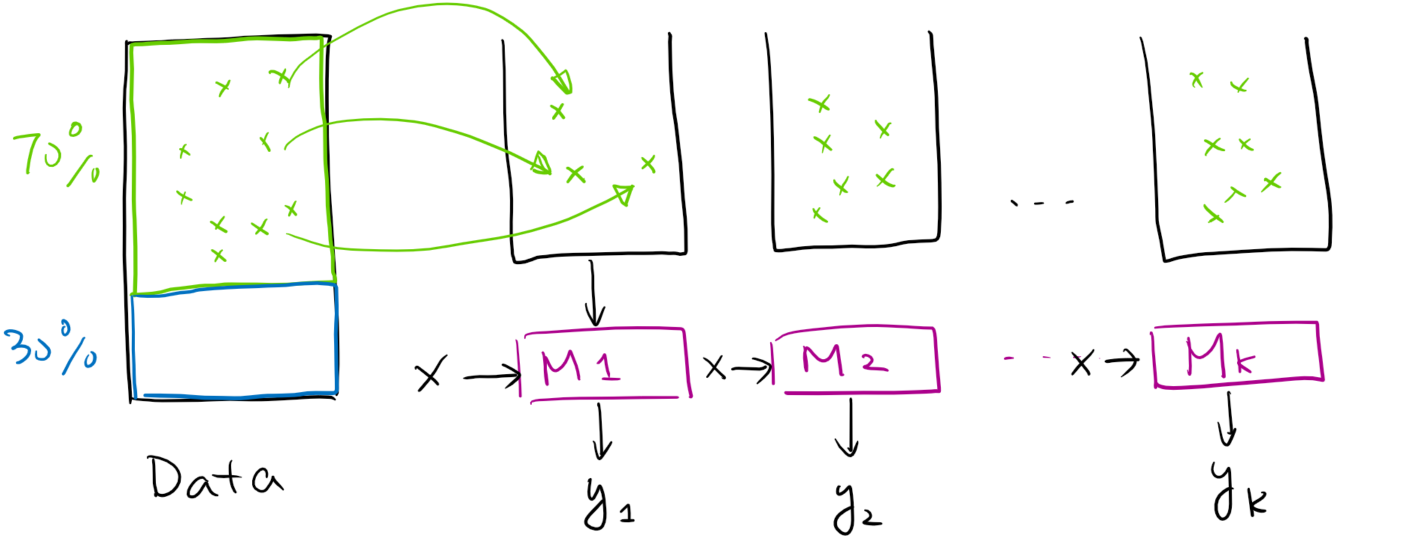
CART

Please Check out the note.

Bagging

Bagging is the abbreviation of Bootstrap Aggregating.

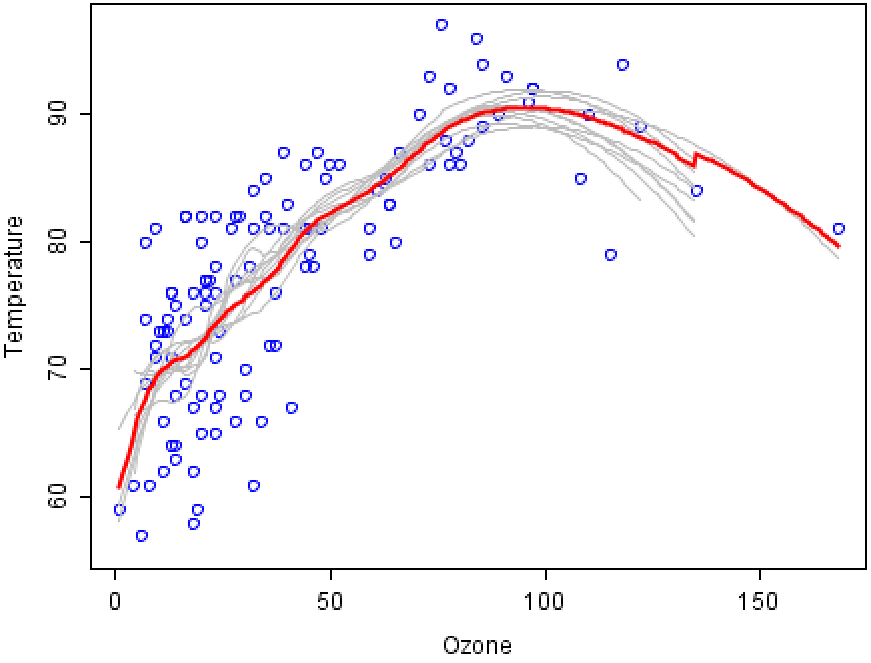
Bootstrapping is any test or metric that relies on random sampling with replacement.



$$X \rightarrow y^*, \quad y^* = \text{mean}(y_1, y_2 \cdots y_k)$$

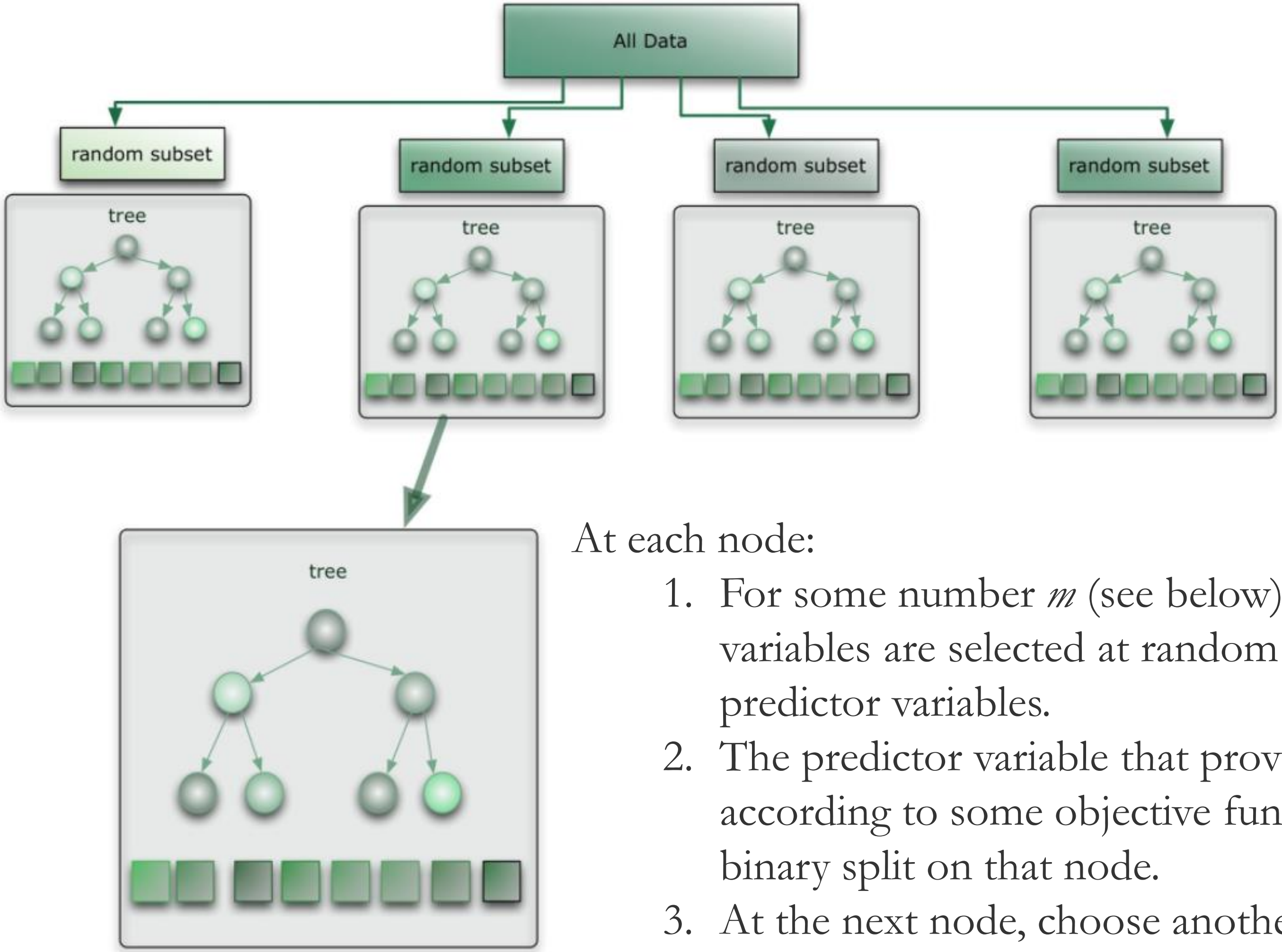
Bagging

proposed by Leo Breiman in 1994 to improve classification by combining classifications of randomly generated training sets.



Given a standard training set D of size *n*, bagging generates *m* new training sets, each of size n' , by sampling from D uniformly with replacement (bootstrapping). By sampling with replacement, some observations may be repeated in each round. If n' =*n*, then for large *n* the sampled set is expected to have the fraction (1 - 1/e) of original data (≈63.2%)

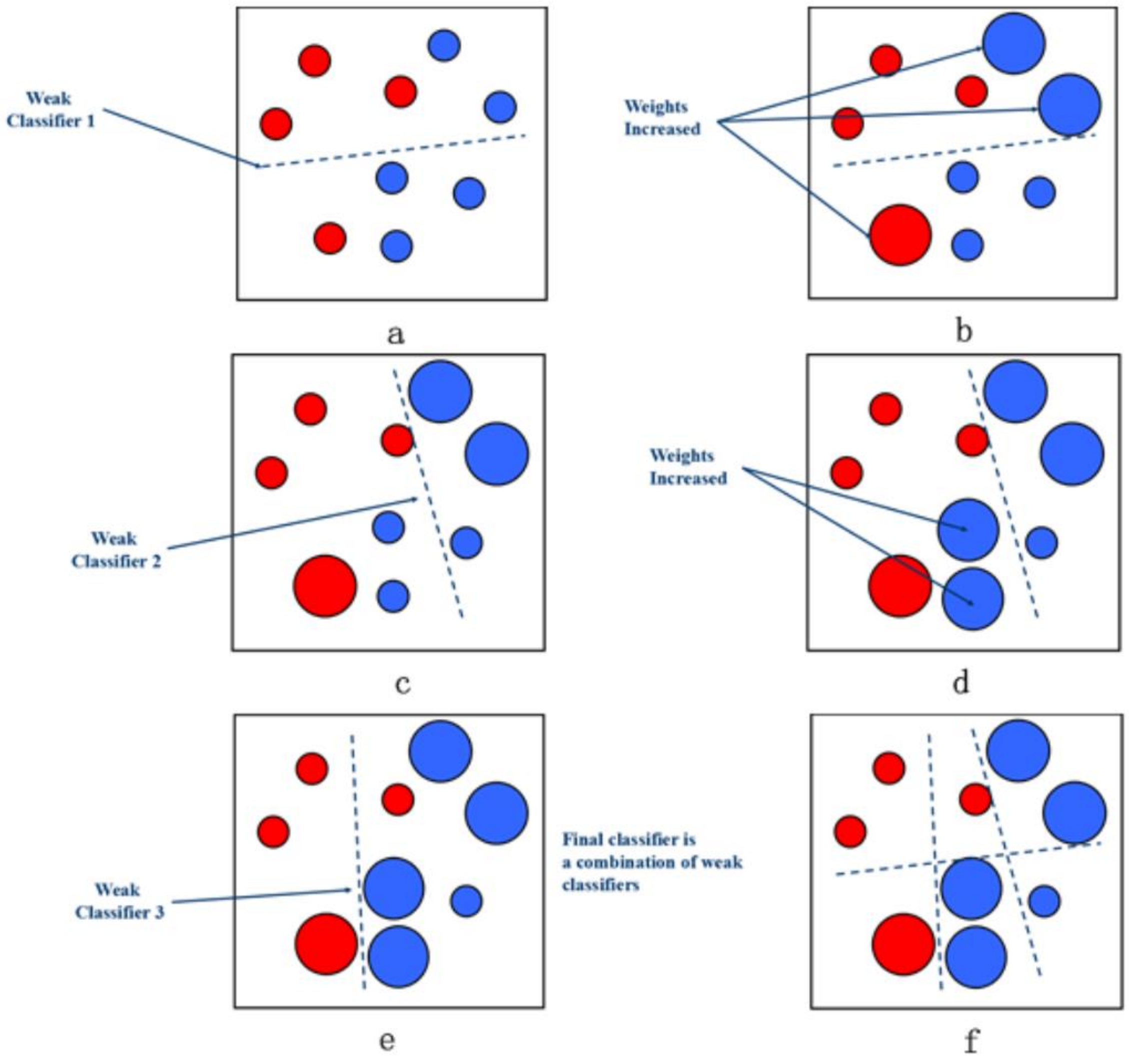
Random Forest



At each node:

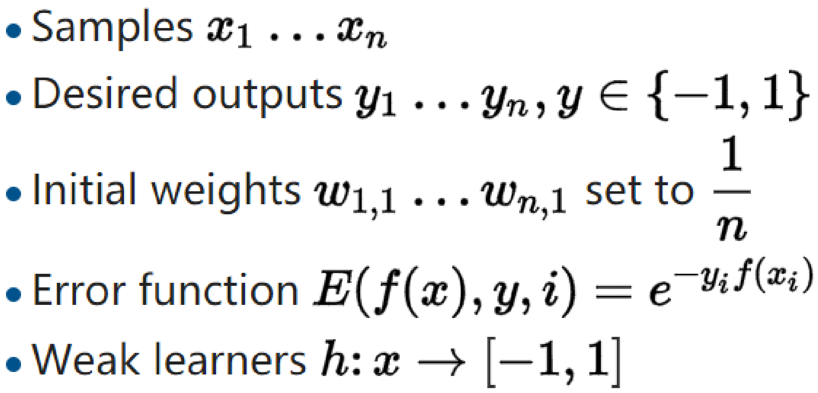
1. For some number *m* (see below), *m* predictor variables are selected at random from all the predictor variables.
2. The predictor variable that provides the best split, according to some objective function, is used to do a binary split on that node.
3. At the next node, choose another *m* variables at random from all predictor variables and do the same.

Partition


Weak Classifier 1
a
Weights Increased
b
Weak Classifier 2
c
Weights Increased
d
Weak Classifier 3
e
Final classifier is a combination of weak classifiers
f

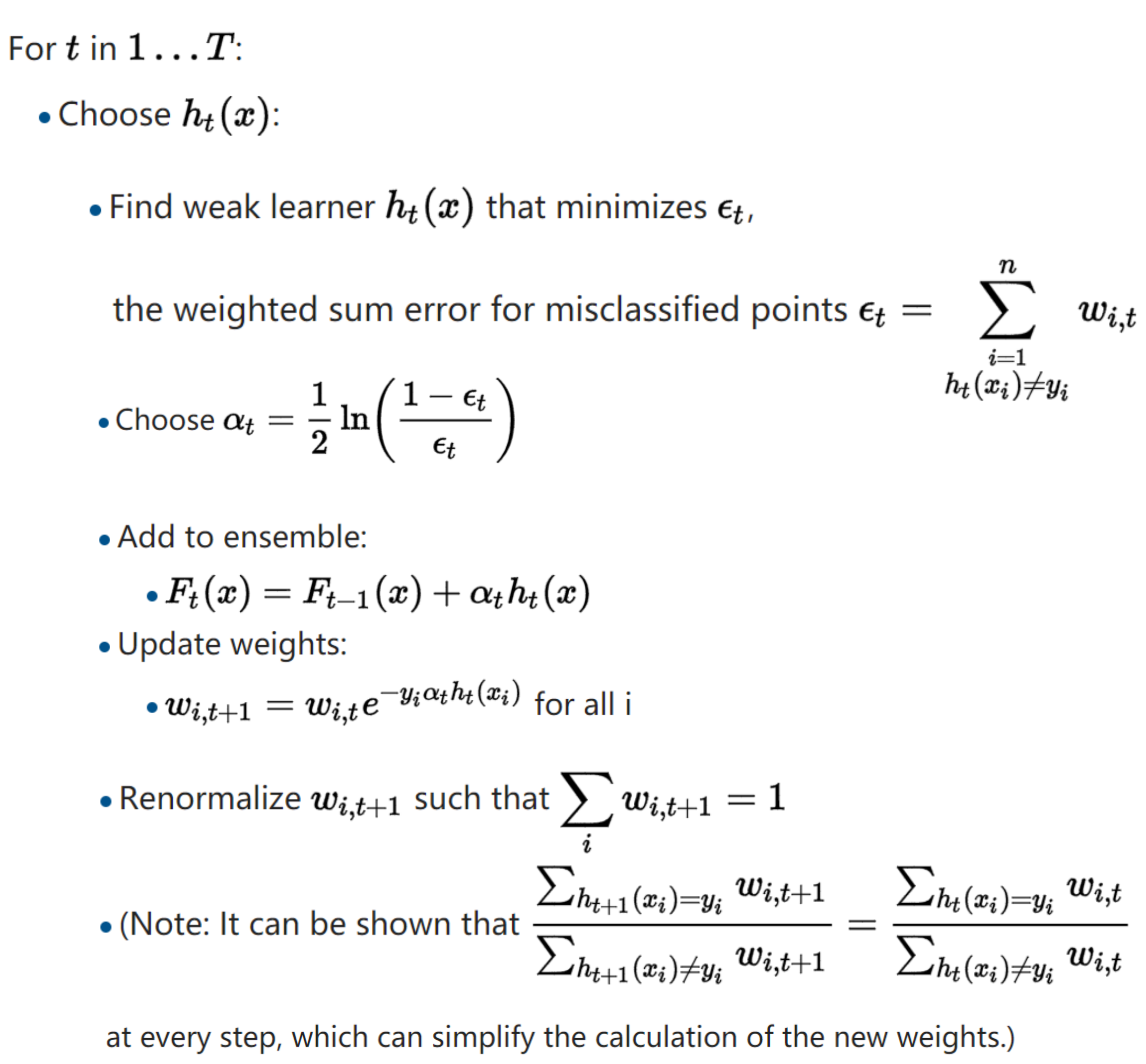
Adaptive Boost

The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers.



- Samples $x_1 \ldots x_n$
- Desired outputs $y_1 \ldots y_n, y \in \{-1, 1\}$
- Initial weights $w_{1,1} \ldots w_{n,1}$ set to $\frac{1}{n}$
- Error function $E(f(x), y, i) = e^{-y_i f(x_i)}$
- Weak learners $h \colon x \rightarrow [-1, 1]$

Information Gain



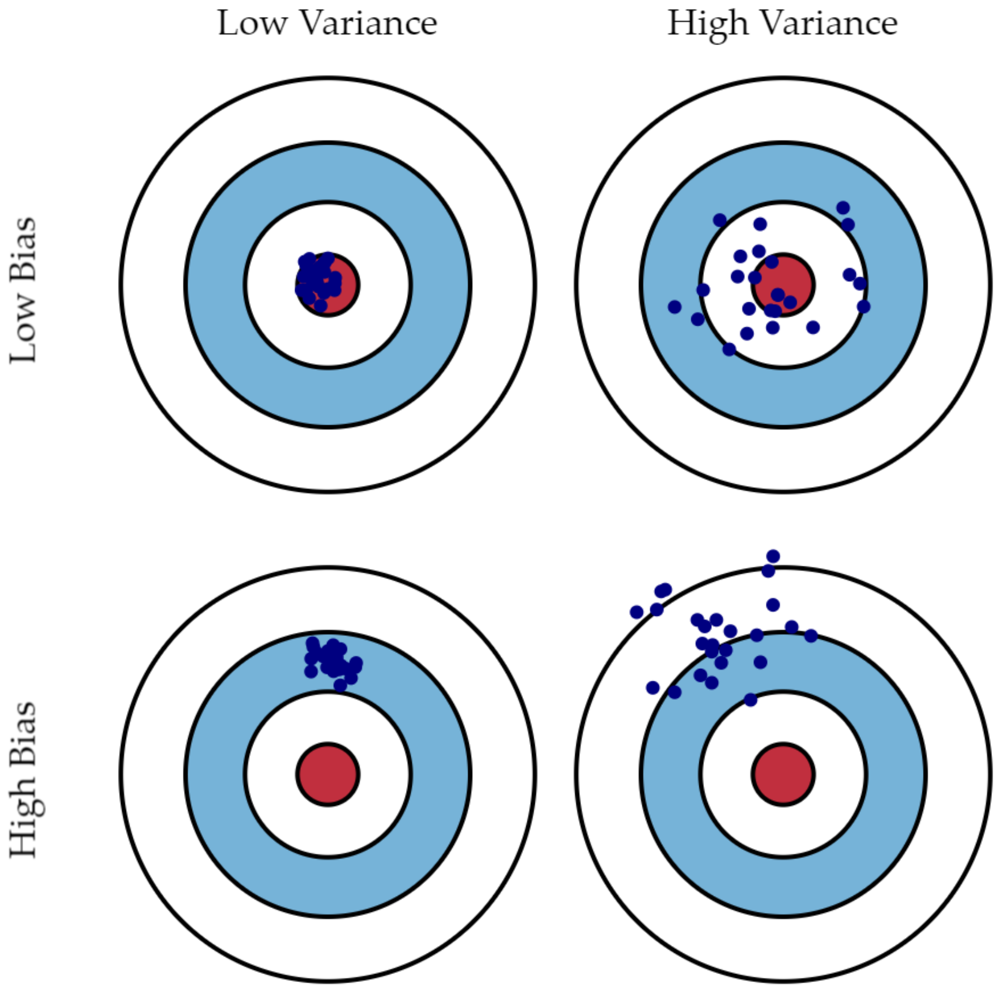
For t in $1 \ldots T$:

- Choose $h_t(x)$:
  - Find weak learner $h_t(x)$ that minimizes ϵ_t,

    the weighted sum error for misclassified points $\epsilon_t = \sum_{\substack{i=1 \\ h_t(x_i) \neq y_i}}^{n} w_{i,t}$
  - Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$
  - Add to ensemble:
    - $F_t(x) = F_{t-1}(x) + \alpha_t h_t(x)$
  - Update weights:
    - $w_{i,t+1} = w_{i,t} e^{-y_i \alpha_t h_t(x_i)}$ for all i
  - Renormalize $w_{i,t+1}$ such that $\sum_i w_{i,t+1} = 1$
  - (Note: It can be shown that $\frac{\sum_{h_{t+1}(x_i)=y_i} w_{i,t+1}}{\sum_{h_{t+1}(x_i)\neq y_i} w_{i,t+1}} = \frac{\sum_{h_t(x_i)=y_i} w_{i,t}}{\sum_{h_t(x_i)\neq y_i} w_{i,t}}$

    at every step, which can simplify the calculation of the new weights.)

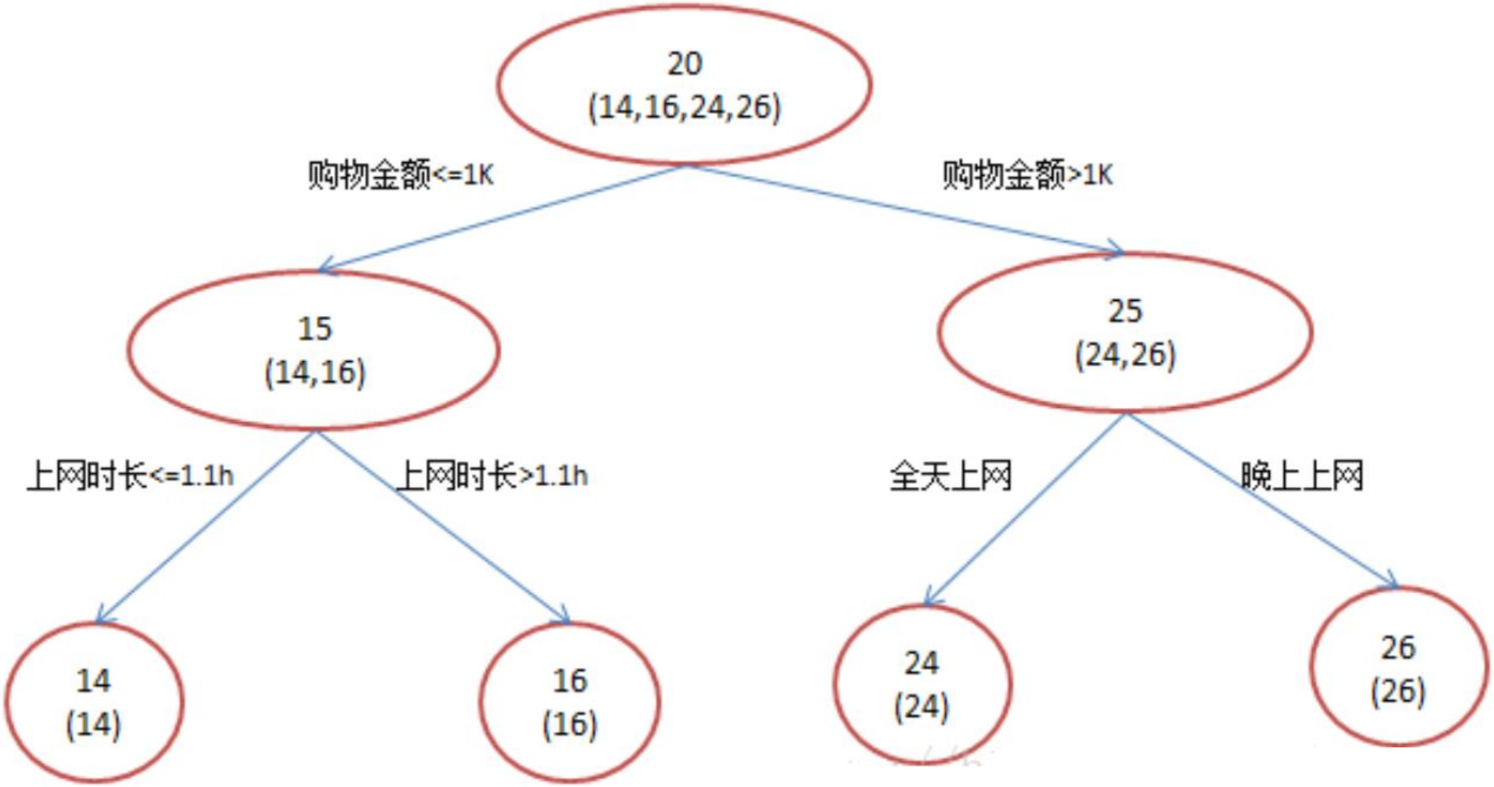
Bias-Variance



GBDT

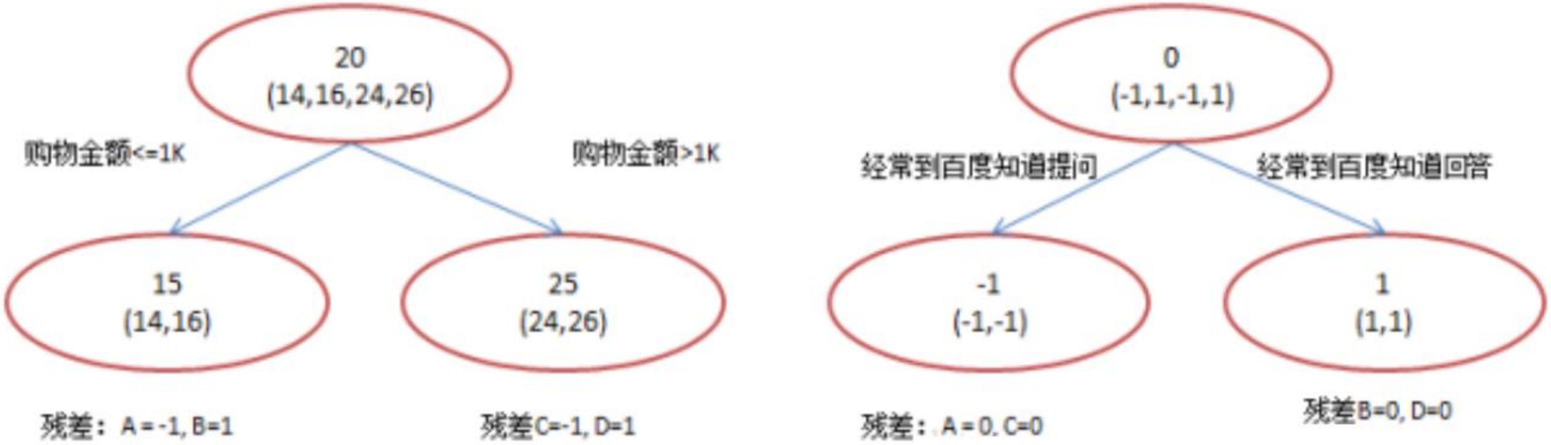
Please Check out the note.

GBDT Example



https://toutiao.io/posts/u52t61/preview

Discretization



XGBoost

What is XGBoost?

XGBoost stands for e**X**treme **G**radient **B**oosting.

The name xgboost, though, actually refers to the engineering goal to push the limit of computations resources for boosted tree algorithms. Which is the reason why many people use xgboost.

https://homes.cs.washington.edu/~tqchen/2016/03/10/story-and-lessons-behind-the-evolution-of-xgboost.html

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：**小象学院**