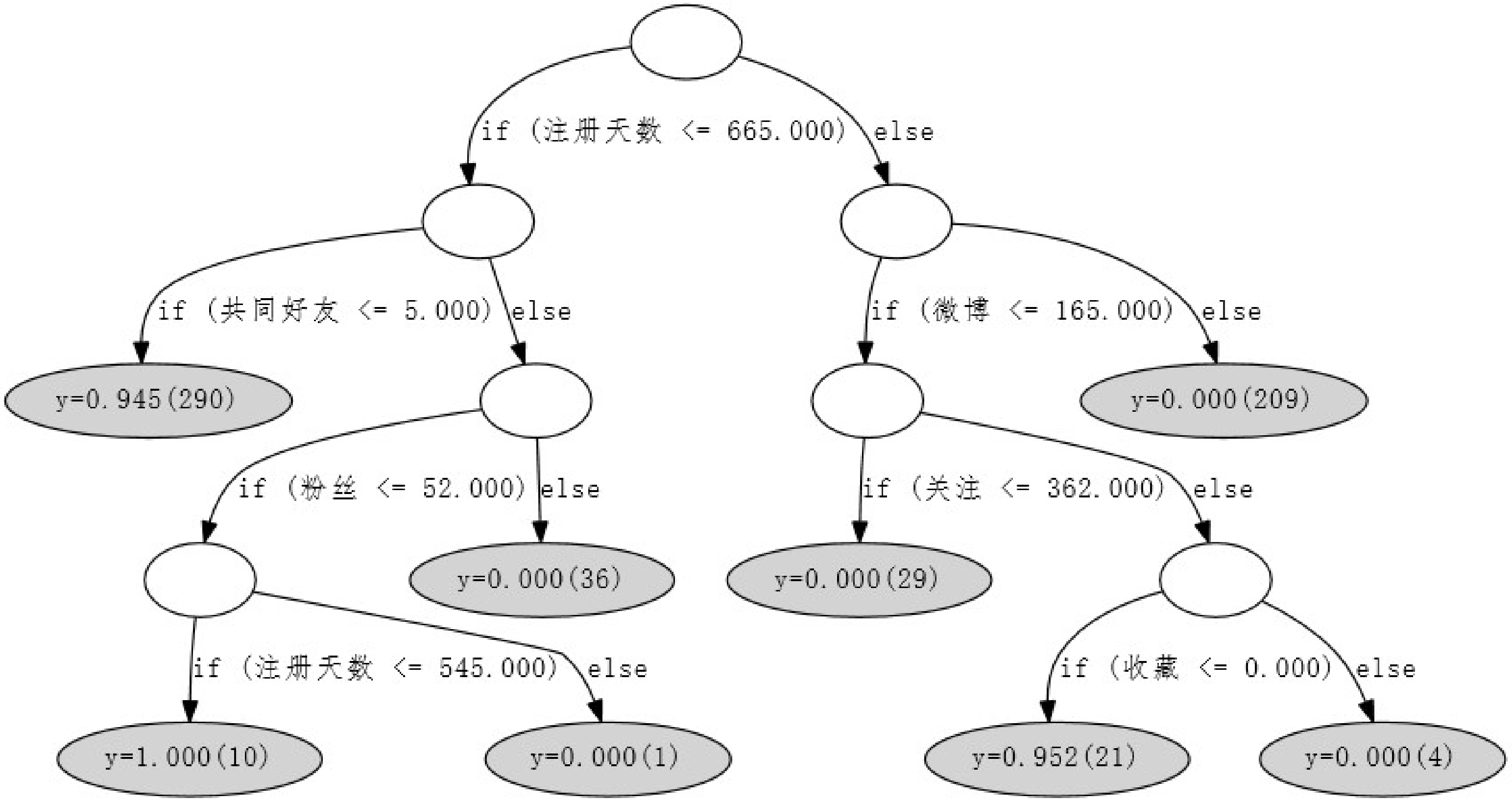
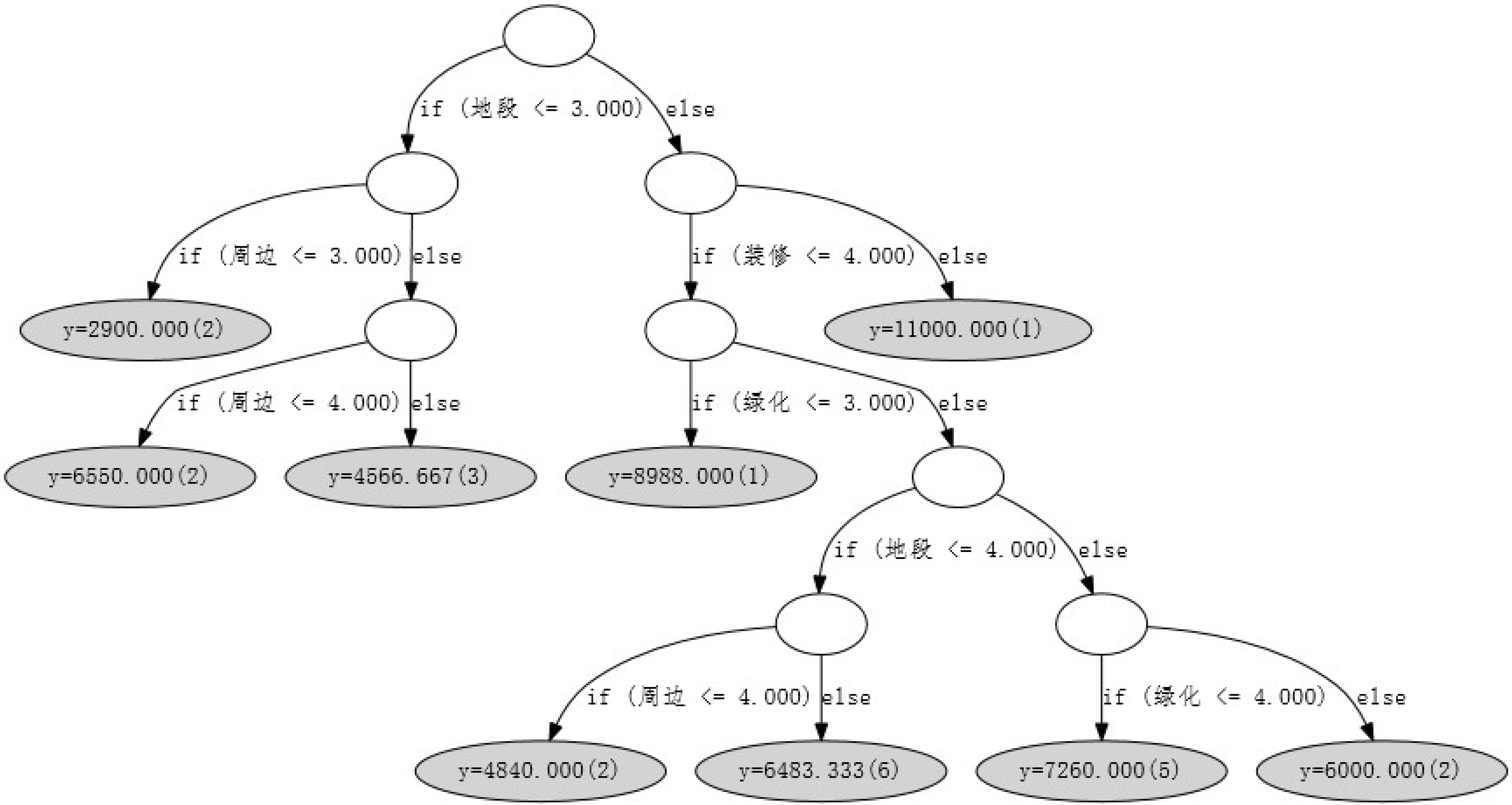
Advanced

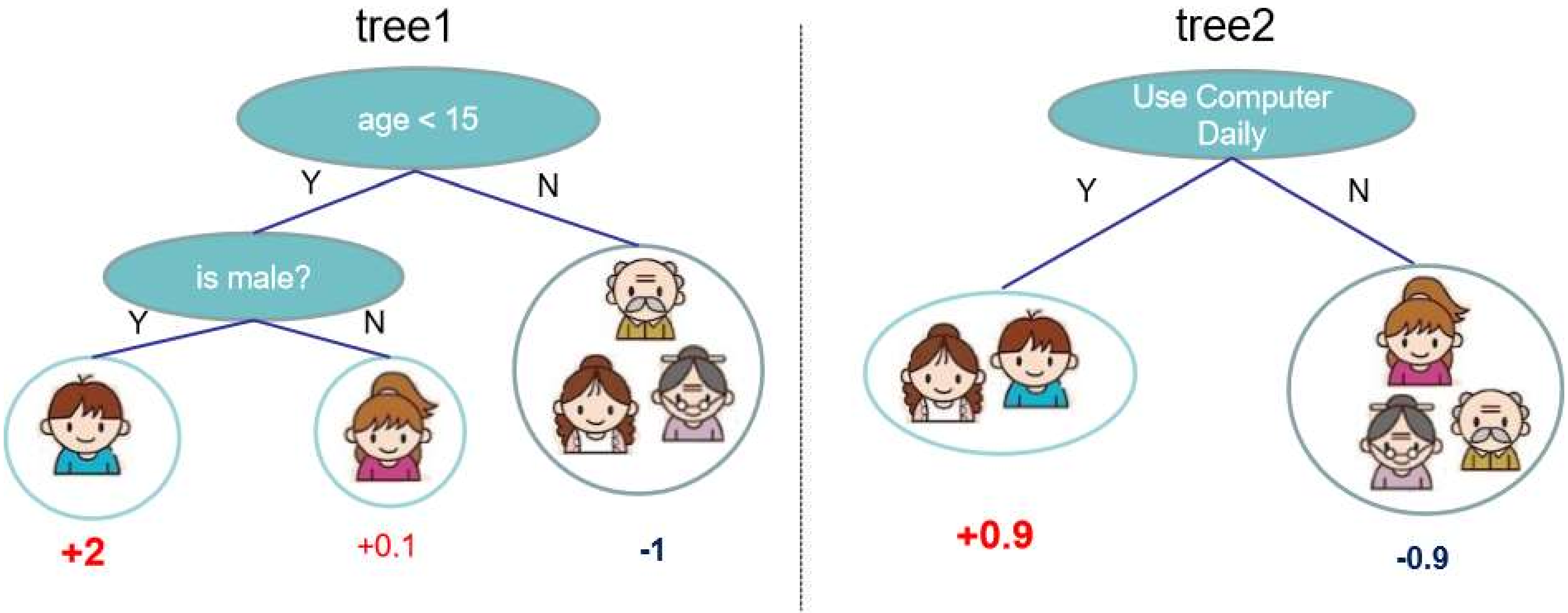
Classification Demo



Regression Demo



Regression Tree Ensemble



f() = 2 + 0.9= 2.9　　f()= -1 - 0.9= -1.9

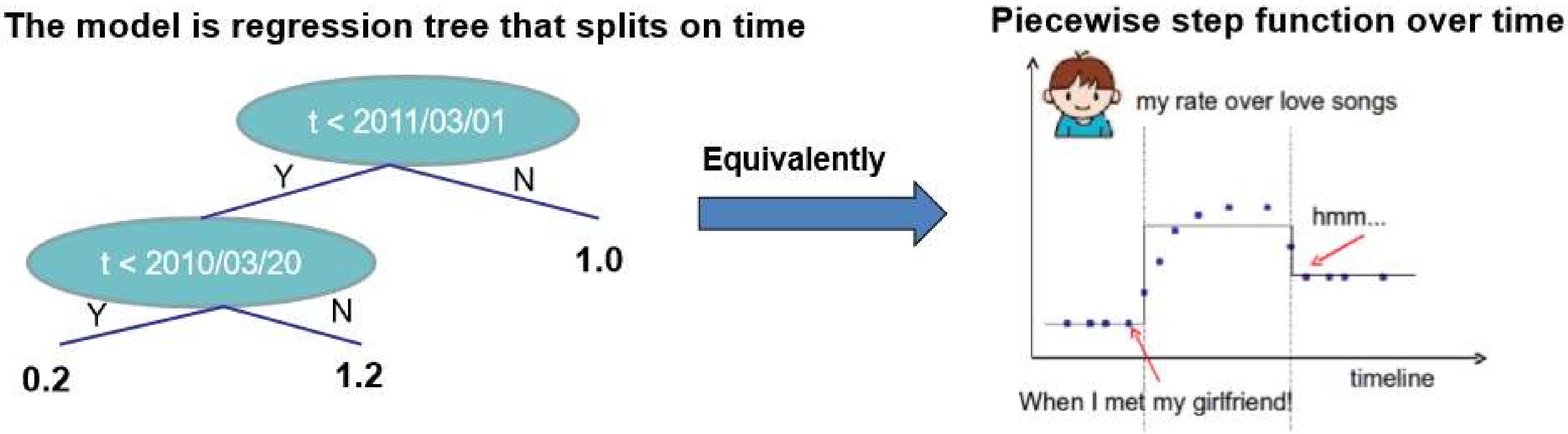
Prediction of is sum of scores predicted by each of the tree

Tree Ensemble methods

- • Very widely used, look for GBM, random forest…
  - Almost half of data mining competition are won by using some variants of tree ensemble methods
- • Invariant to scaling of inputs, so you do not need to do careful features normalization.
- • Learn higher order interaction between features.
- • Can be scalable, and are used in Industr

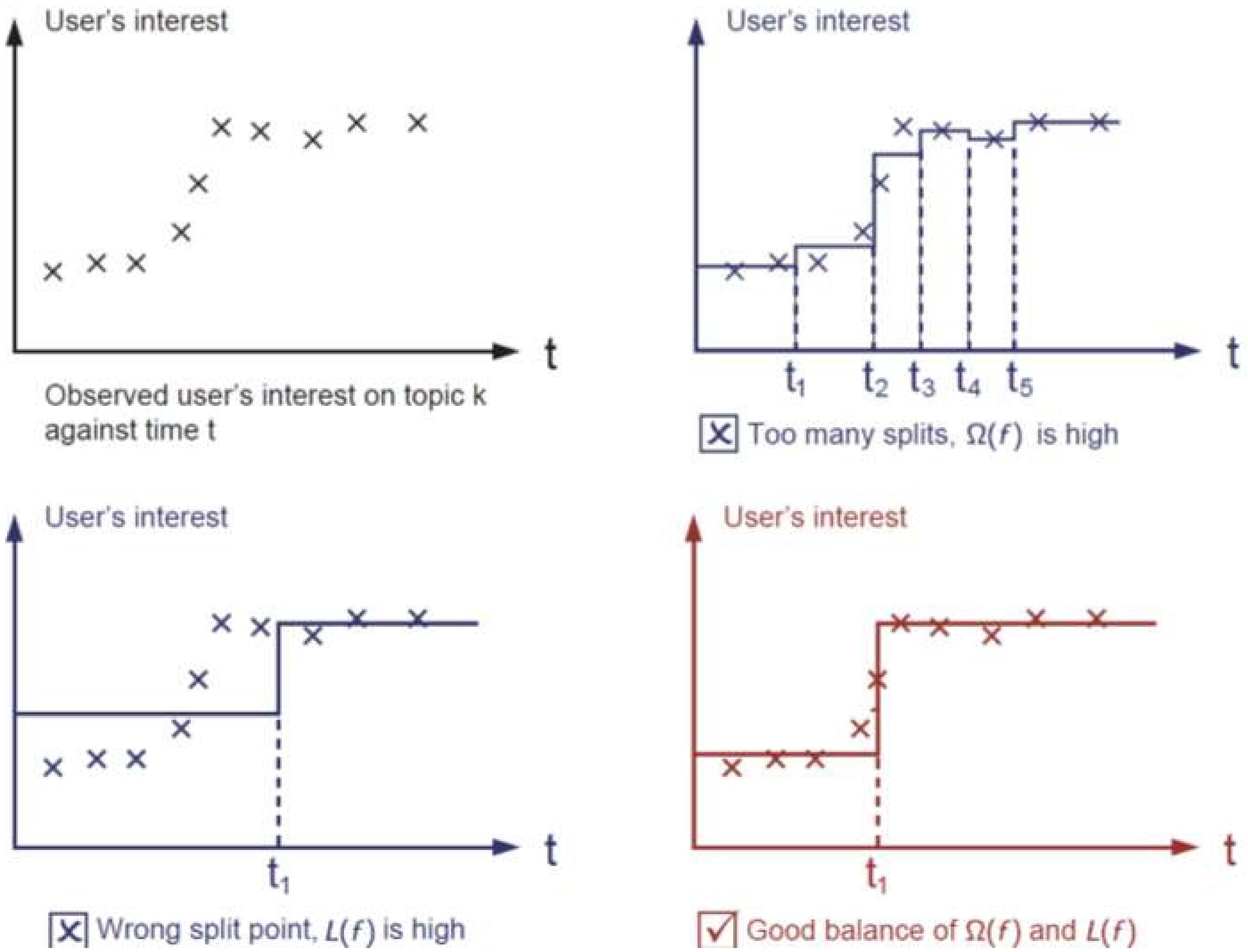
Learning a step function

- Things we need to learn



- Objective for single variable regression tree
  - Training Loss: How will the function fit on the points?
  - Regularization: How to define complexity of the function?

Learning a step function



Additive Training

- Objective: $\sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_k \Omega(f_k), f_k \in \mathcal{F}$

- We can not use methods such as SGD, to find f (since they are trees, instead of just numerical vectors)

- Solution: Start from constant prediction, add a new function each time

$$
\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i)
\end{aligned}
$$

Gradient Boosted Regression Tree

Let h be a DT, F be a tree ensemble.

Use square error loss

$$L(y, F) = \frac{(y - F)^2}{2}$$

1 choose an initial guess f_0, let $F_0 = f_0$

2 for $k = 1, 2, \ldots, K$

  2.1 $\tilde{y}_i = -\frac{\partial L(y_i, F_{k-1}(x_i))}{\partial F_{k-1}(x_i)}$, $i = 1, 2, \ldots, N$

  2.2 $w^* = \arg\min_w \sum_{i=1}^{N} [\tilde{y}_i - h_k(x_i; w)]^2$

  2.3 $\rho^* = \arg\min_\rho \sum_{i=1}^{N} L(y_i, F_{k-1}(x_i) + \rho h_k(x_i; w^*))$

  2.4 let $f_k = \rho^* h_k(x; w^*)$, $F_k = F_{k-1} + f_k$

3 output F_K

Gradient Boosted Regression Tree

- Objective:

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant$$

- Take Taylor expansion of the objective

Recall $f(x + \Delta x) \simeq f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2$

Define $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}), \quad h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$

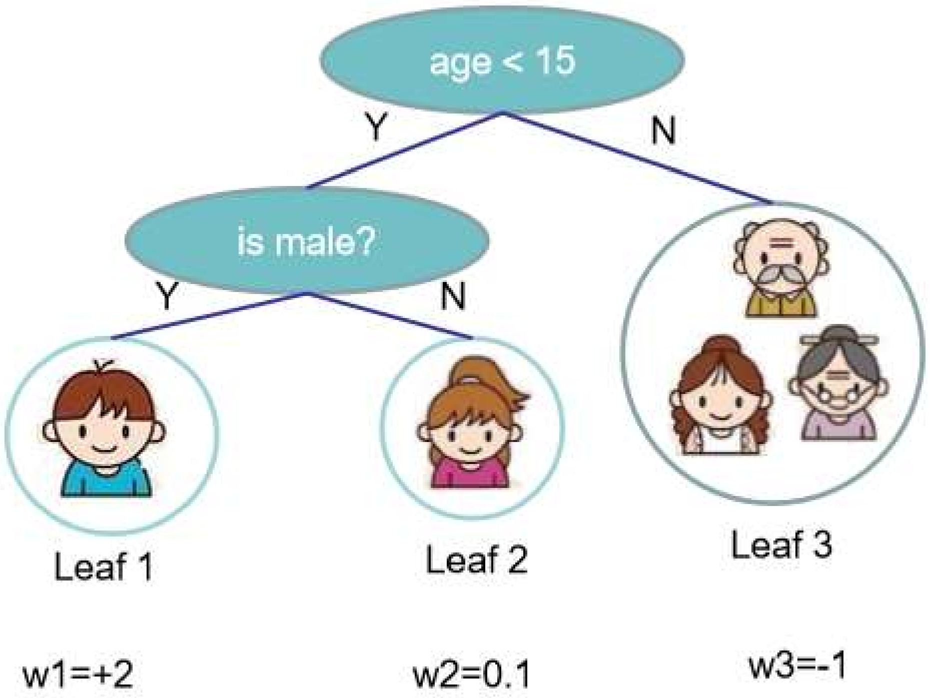
$$Obj^{(t)} \simeq \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + constant$$

Complexity of Tree

- Define complexity as

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Number of leaves **L2 norm of leaf scores**



$$\Omega = \gamma 3 + \frac{1}{2}\lambda(4 + 0.01 + 1)$$

Complexity of Tree

- Regroup the objective by each leaf

$$
\begin{aligned}
Obj^{(t)} &\simeq \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t) \\
&= \sum_{i=1}^{n}\left[g_i w_{q(x_i)} + \frac{1}{2}h_i w_{q(x_i)}^2\right] + \gamma T + \lambda\frac{1}{2}\sum_{j=1}^{T} w_j^2 \\
&= \sum_{j=1}^{T}\left[(\sum_{i\in I_j} g_i)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2\right] + \gamma T
\end{aligned}
$$

The Structure Score

- Let us define $G_j = \sum_{i \in I_j} g_i \quad H_j = \sum_{i \in I_j} h_i$

$$
\begin{aligned}
Obj^{(t)} &= \sum_{j=1}^{T} \left[(\sum_{i \in I_j} g_i) w_j + \tfrac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2 \right] + \gamma T \\
&= \sum_{j=1}^{T} \left[G_j w_j + \tfrac{1}{2} (H_j + \lambda) w_j^2 \right] + \gamma T
\end{aligned}
$$

- Then:

$$
w_j^* = -\frac{G_j}{H_j + \lambda} \qquad Obj = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T
$$

This measures how good a tree structure is!

$$
Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma
$$

Square Loss

- Square error

$$L(y, F) = \frac{(y - F)^2}{2}$$

$$g_i = F_{k-1}(x_i) - y_i = -\tilde{y}_i$$

$$h_i = 1$$

- We have

$$b_j^* = -\frac{\sum_{x_i \in R_j} g_i}{\sum_{x_i \in R_j} h_i} = \frac{\sum_{x_i \in R_j} \tilde{y}_i}{\sum_{x_i \in R_j} 1}$$

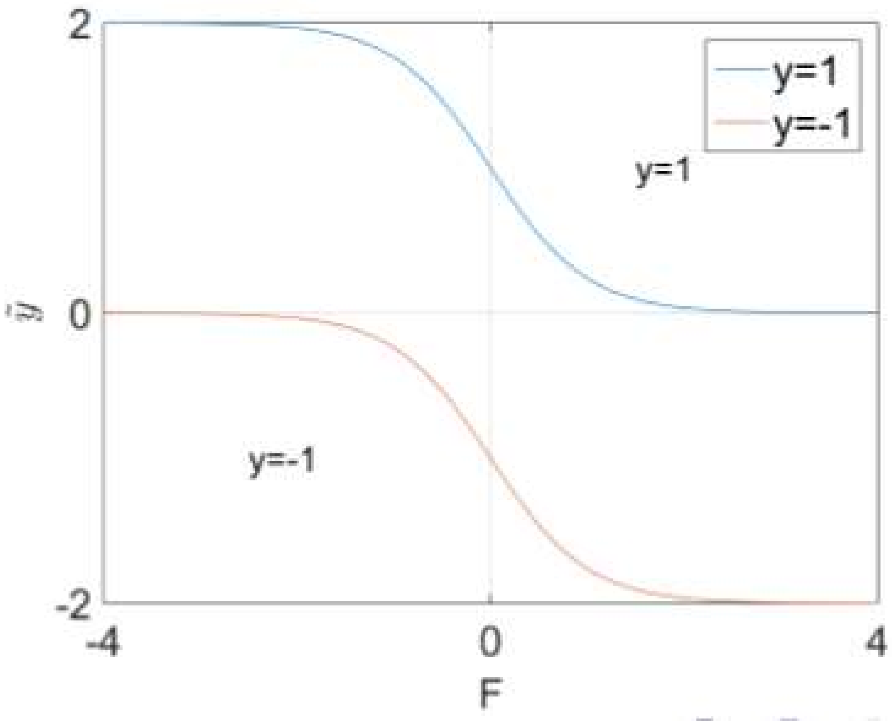
Logistic Loss

- For binary classification
  - Logistic loss

$$L(y, F) = \log(1 + \exp(-2yF))$$

$$g_i = -\frac{2y_i}{1 + \exp(2y_i F_{k-1}(x_i))}$$

$$h_i = \frac{4\exp(2y_i F_{k-1}(x_i))}{[1 + \exp(2y_i F_{k-1}(x_i))]^2} = |g_i|\,(2 - |g_i|)$$



  - We have

$$b_j^* = -\frac{\sum_{x_i \in R_j} g_i}{\sum_{x_i \in R_j} |g_i|\,(2 - |g_i|)}$$

$$p(y = 1|x) = \frac{1}{1 + \exp^{-2F(x)}}$$

Other

- Shrinkage
- Column subsampling