机器学习基础

--原理、方法与实践

王志鹏

- 鸣谢
  - eCafe
  - 成臣

- 四次课
  - → 1.定义、前沿成果、基础方法
  - 2. 基础方法讲解与实践
    - 回顾
    - 逻辑回归（logistic regression）
    - 凸优化的一阶方法，二阶方法
    - 正则化
    - 随机梯度下降
    - 梯度提升树(gradient boosting decision trees) 、k-means
  - 3. 神经网络原理讲解与实践
    - Multi-layer perceptron: tensorflow
      - 反向传播、激活函数、dropout及其他相关知识点
    - Convolutional neural network: keras
    - Recurrent neural network: keras
  - 4. 强化学习方法介绍与实践
    - Alpha go论文介绍
    - Policy gradient

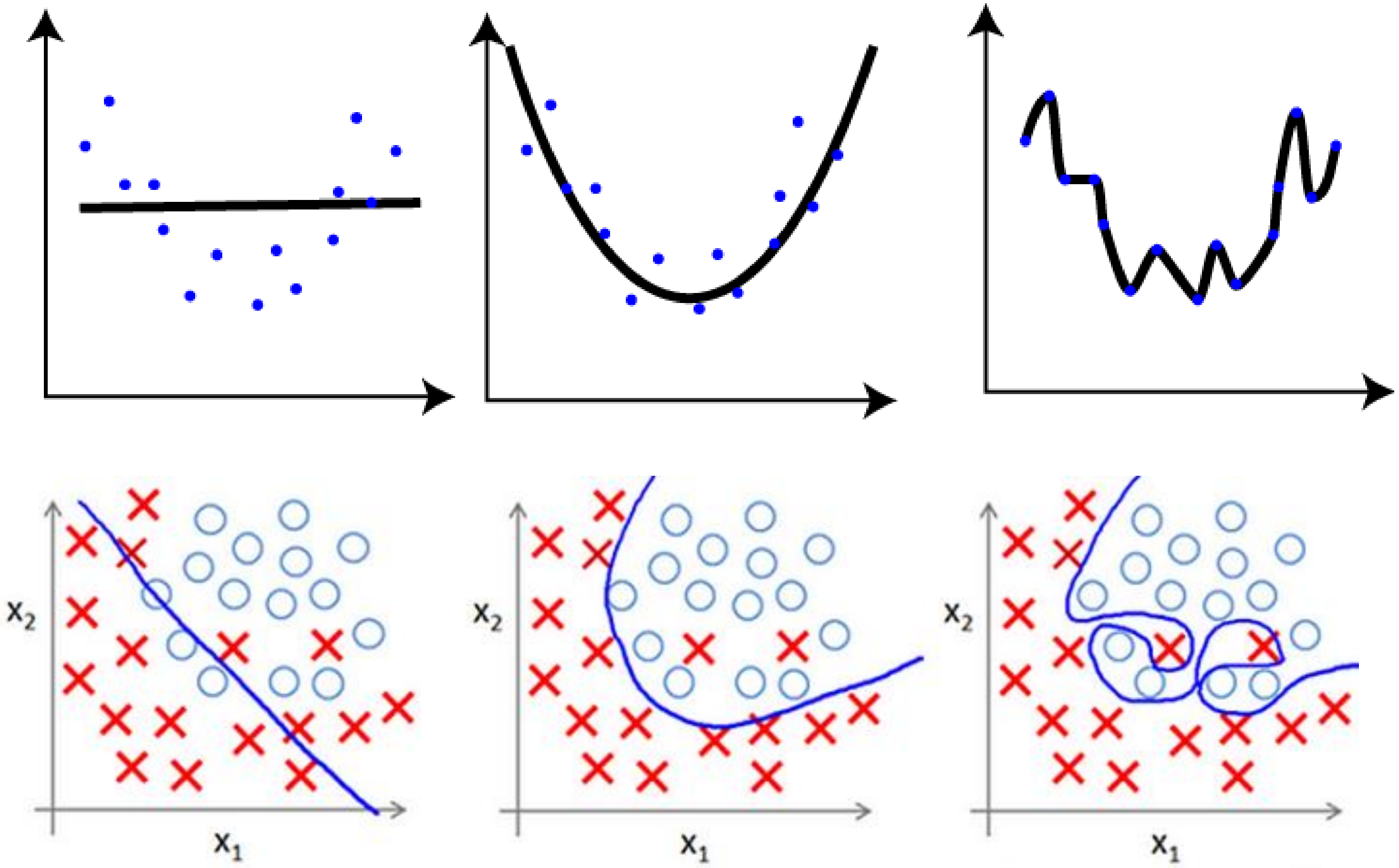
- 机器学习定义
  - 对于某类任务T和（对任务的）性能指标P，一个计算机程序能够从经验E里学习，也就是说，基于经验E，（计算机程序）在任务T上的性能指标P有所提升。 -- Tom Mitchell
    - T P E
      - 学习：从经历（历史数据）里面找到道理，来做的更好
  - 机器学习就是不直接编程而让计算机有学习（解决问题）的能力 -- Arthur Samuel
    - 自动从**数据**中发现**规律**，并使用规律**解决问题**
      - 使用**优化方法**找到**模型**基于**数据**的**最适合的参数**，使用得到的参数通过模型**完成任务**

- 对于某类任务T和性能指标P， 一个计算机程序能够从经验E里学习，也就是说，基于经验E，在任务T上的性能指标P有所提升。
    - 怎么保证提升
    - 内在规律存在并被有效发现
    - 一定程度的统计不变性

一种典型监督学习做法

- 现有数据分为训练数据集与测试数据集（严格分开，8:2或7:3)
- →选择合适模型
- 根据训练数据得到模型的合适的参数
- 在测试数据上对模型进行验证

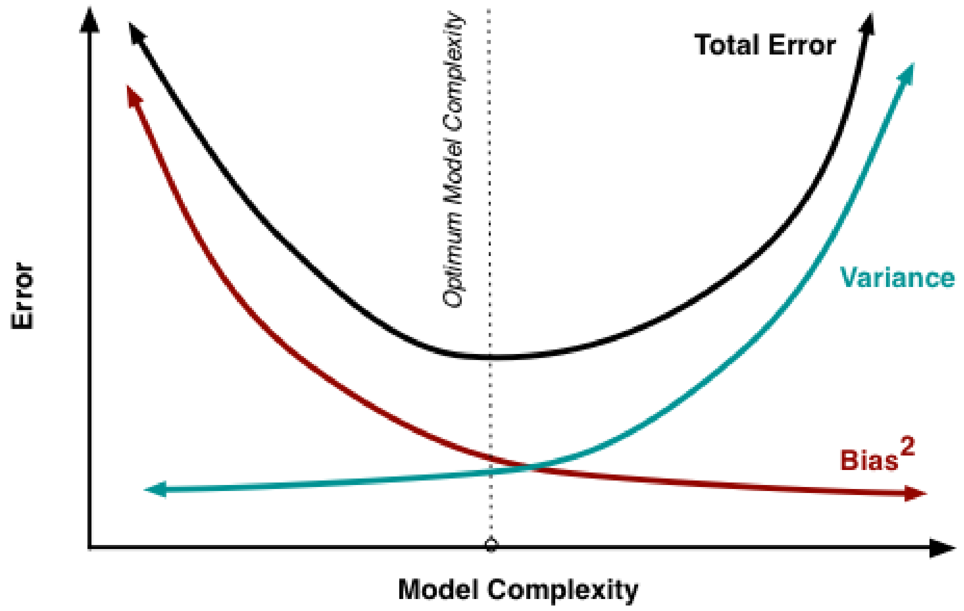
- 模型拟合能力：欠拟合与过拟合
  - 模型复杂度上升 → 模型拟合训练数据的能力越强 & 模型在训练数据上的表现与在测试数据上的表现越难一致

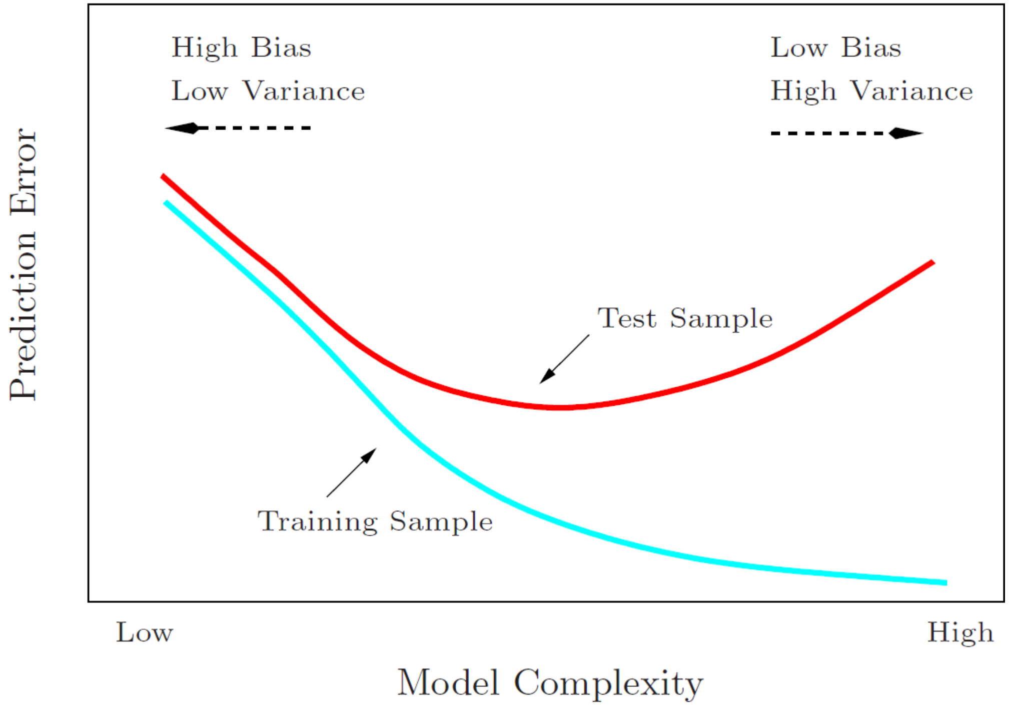


- 模型泛化：偏差（bias）与方差（variance）

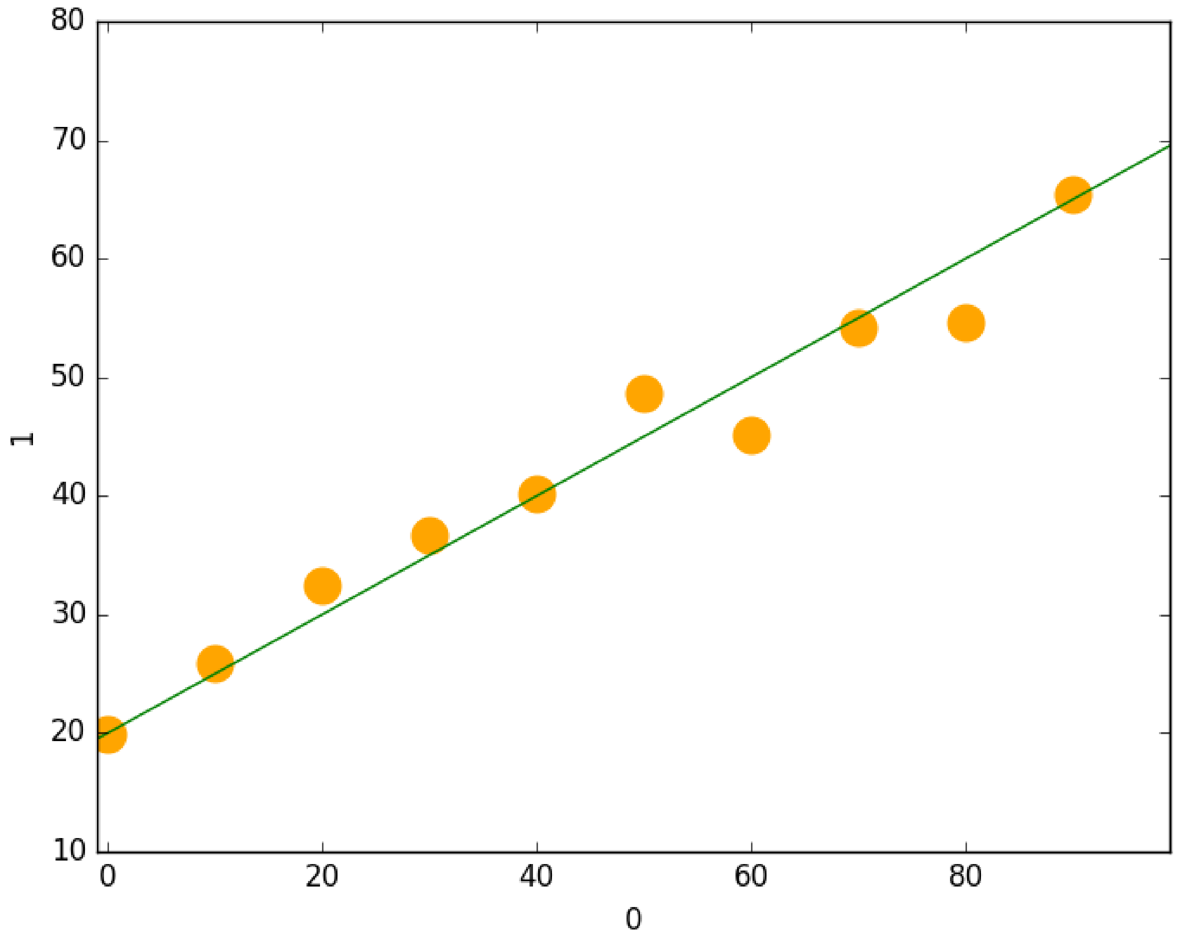
  - 偏差：在训练数据上预测与实际标签的偏差
    - 预测的准确性
  - **方差**：不同训练数据集训练出来的模型，预测间的差异
    - 预测的稳定性

图片来自网络

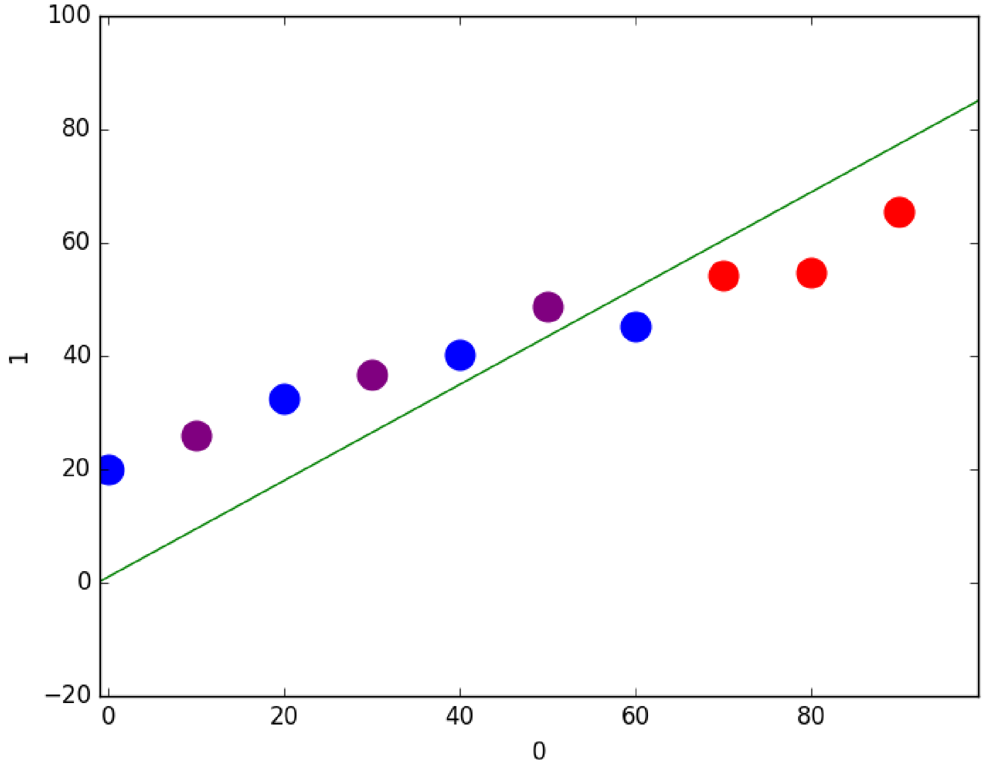


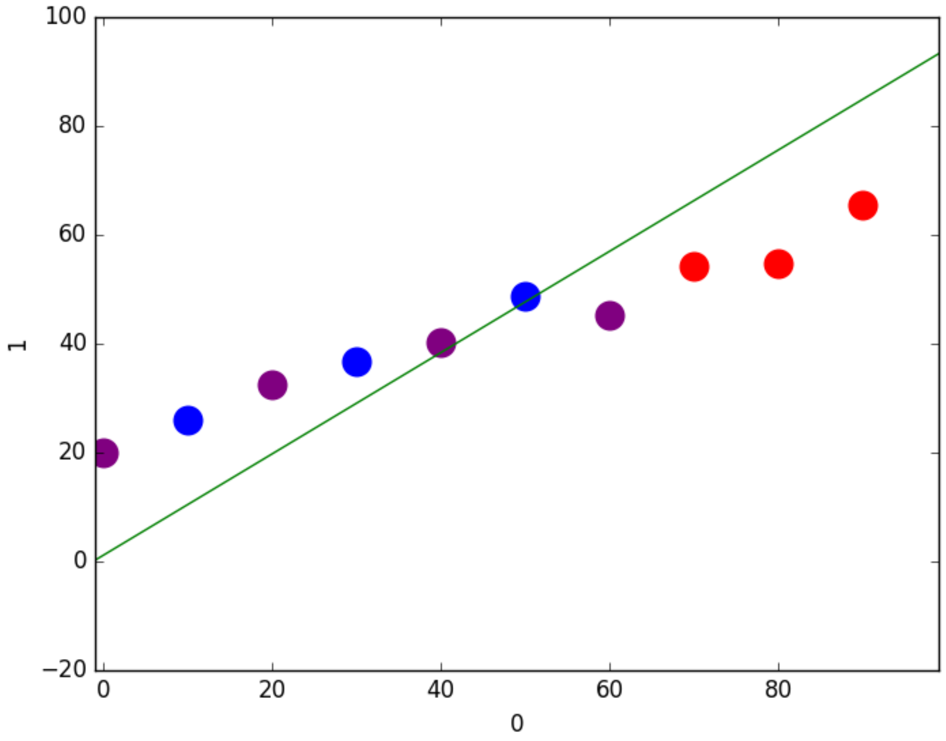


- $y = 0.5x + 20 + noise$

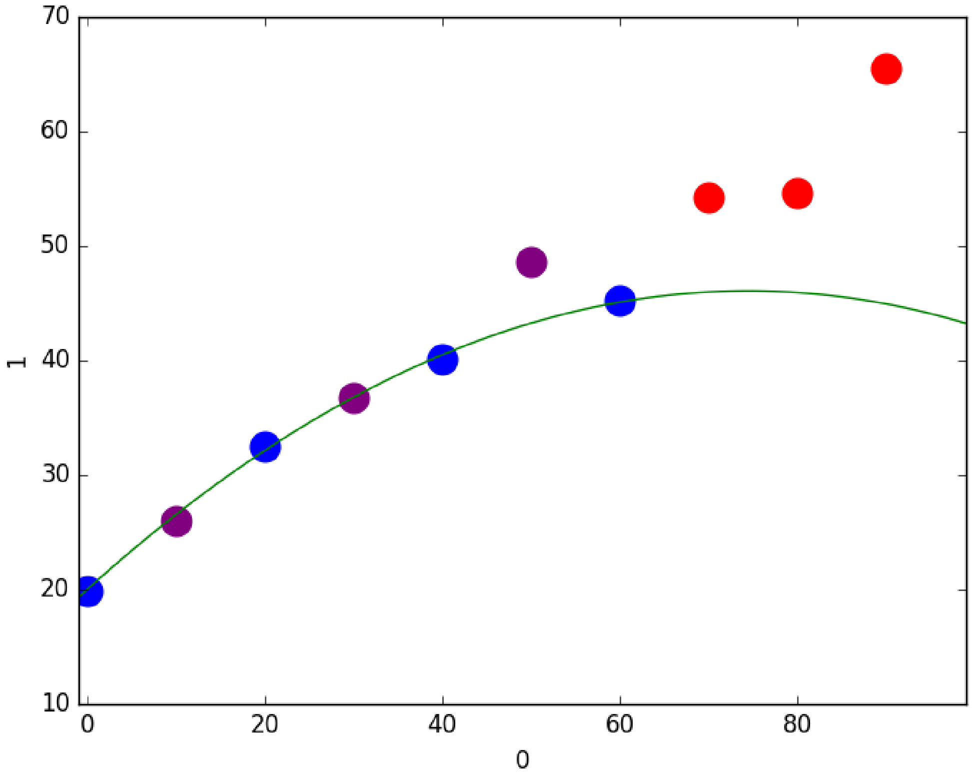


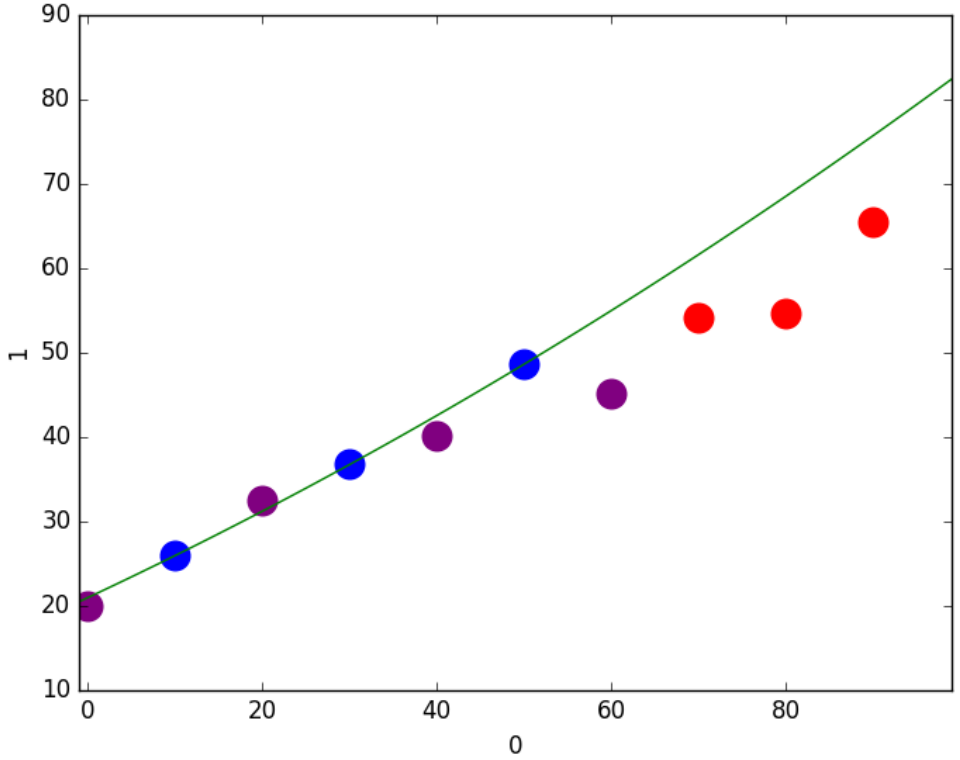
- 模型：$y = ax + b$
  - 蓝点：训练数据
  - 紫点：未使用数据
  - 红点：测试数据





- 模型：$y = ax^2 + bx + c$
  - 蓝点：训练数据
  - 紫点：未使用数据
  - 红点：测试数据

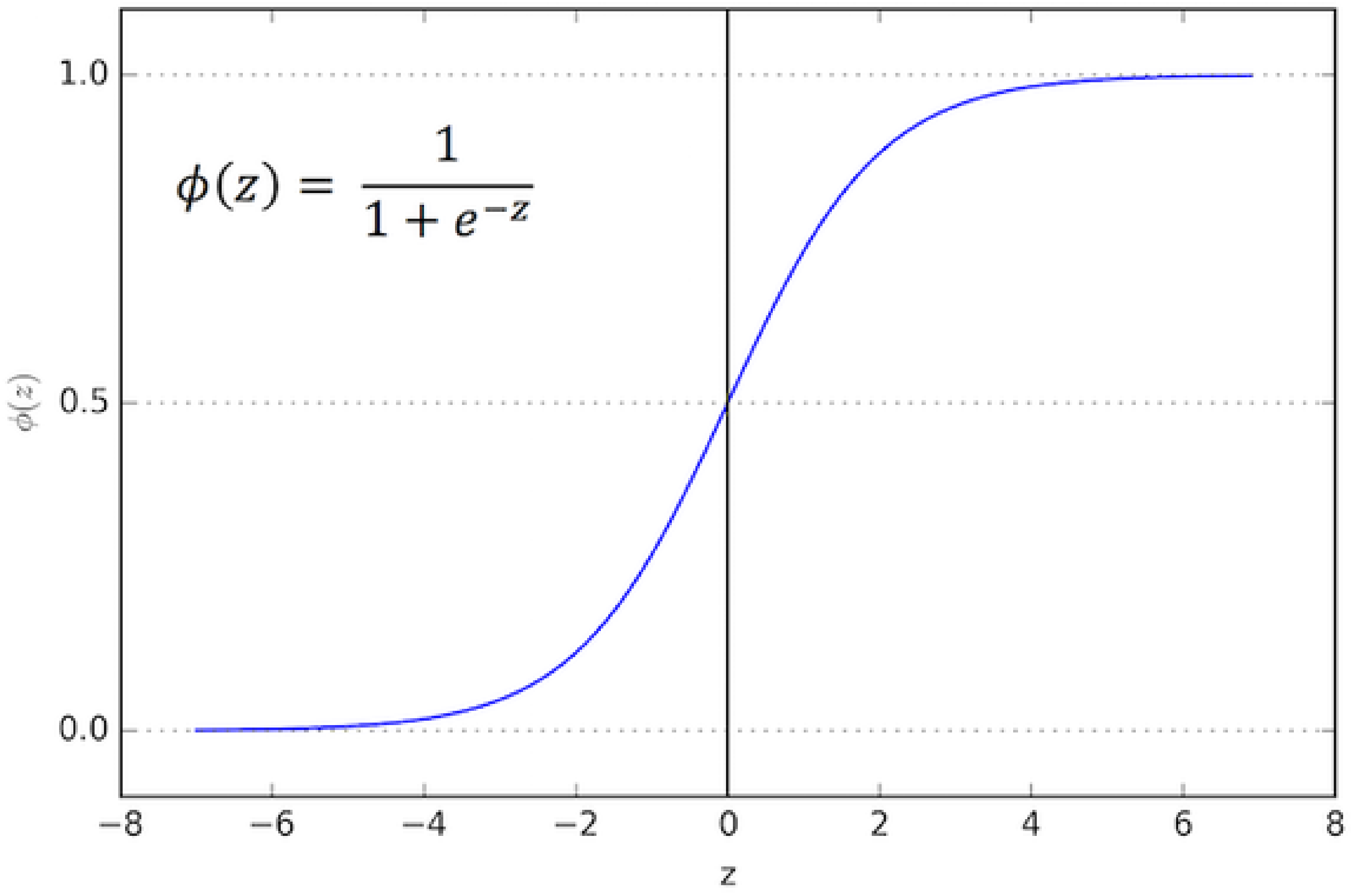




一种典型监督学习做法

- 现有数据分为训练数据集与测试数据集（严格分开，8:2或7:3)
- 选择合适模型
- →根据训练数据得到模型的合适的参数
- 在测试数据上对模型进行验证

• 逻辑回归（logistic regression）



逻辑回归的内在

点广告可能性

iphone	android	ipad
2 : 98	1 : 200	3 : 20

最简单模型 $y = \begin{cases} 1 & \text{点} \\ 0 & \text{不点} \end{cases}$

$$P(y=1|x) = \begin{cases} \frac{2}{2+98} & x = \text{iphone} \\ \frac{1}{1+200} & x = \text{android} \\ \frac{3}{3+20} & x = \text{ipad} \end{cases}$$

$$P(y=1|x)=\begin{cases}\dfrac{\frac{2}{2}}{\frac{2}{2}+\frac{98}{2}}\\[2ex]\dfrac{1}{1+200}\\[2ex]\dfrac{\frac{3}{3}}{\frac{3}{3}+\frac{20}{3}}\end{cases}$$

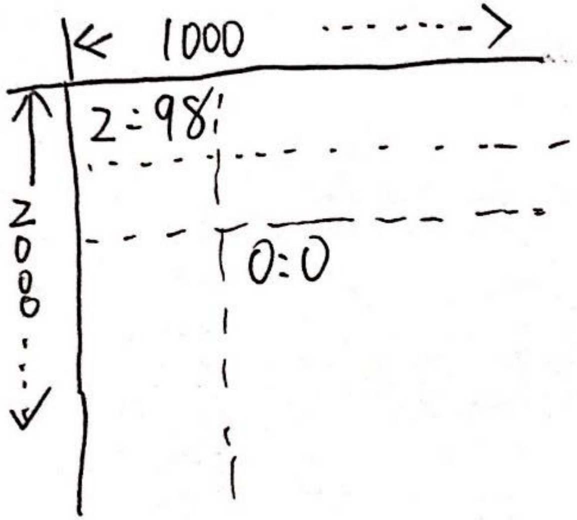
$$P(y=1)=\begin{cases}\dfrac{1}{1+W_{iphone}}\\[2ex]\dfrac{1}{1+W_{android}}\\[2ex]\dfrac{1}{1+W_{ipad}}\end{cases}$$

同时考虑城市 Tokyo, Ozaka

$$P(y=1)=\begin{cases}\dfrac{1}{1+W_{iphone\_Tokyo}}\\ \vdots\\ \dfrac{1}{1+W_{ipad\_Ozaka}}\end{cases}$$

记录 $3*2=6$ 种 W

device有1000种 city有2000个



$$P(y=1)=\begin{cases}\dfrac{1}{1+W_{device1\_city1}}\\ \vdots\\ \dfrac{1}{1+W_{device1000\_city2000}}\end{cases}$$

记录 1000*2000=200万 W

0:0 的W怎么定?

$$P(y=1) = \frac{1}{1+W_{device(i)\_city(j)}}$$

假定device与city独立起作用

$$P(y=1) = \frac{1}{1+W_{device(i)} * W_{city(j)}}$$

参数w的量200万 → 1000+2000=3000

$$P(y=1) = \frac{1}{1+e^{W_{device}+W_{city}}}$$

怎么定每个device的w?

每个city的w?

$$2^3 = 2^2 * 2^1$$

$$= 2^{(2+1)}$$

换一个维度 从每一条数据看

device	city	y
2	1	1
3	5	0

$$P(y=1)=\frac{1}{1+e^{W_{device2}+W_{city1}}}$$

$$P(y=0)=1-P(y=1)=1-\frac{1}{1+e^{W_{device3}+W_{city5}}}$$

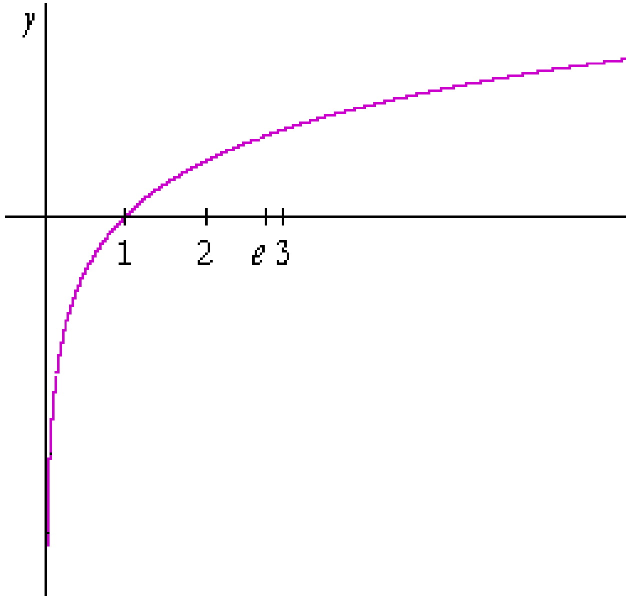
假设每条记录相互独立

$P=\prod_{i=1}^{n}P_i$ 最大化 P $\Leftrightarrow$ 最大化 $\log P$

$\log P=\sum_{i=1}^{n}\log(P_i)$ 最大化 $\log P$ $\Leftrightarrow$ 最小化 $-\log P$

目标 最小化 $-\sum_{i=1}^{n}\log(P_i)$

$$P_i=\begin{cases}\frac{1}{1+e^{W_{device\ of\ i}+W_{city\ of\ i}}} & y=1\\ 1-\frac{1}{1+e^{W_{device\ of\ i}+W_{city\ of\ i}}} & y=0\end{cases}$$



两个假设

特征相互不相关

数据相互不相关

$$L = -\sum_{i=1}^{n} \log(P_i)$$

最终任务：最小化 $L(W_{device1}, \cdots W_{device1000}, W_{city1} \cdots W_{city2000})$

$$L(\vec{w})$$

梯度

$$\nabla L = \left(\frac{\partial L}{\partial w_1} \cdots \frac{\partial L}{\partial w_{3000}}\right) \quad \vec{w} = (w_1, \cdots w_{3000})$$

$$\frac{\partial L}{\partial w_1} = \sum_{i=1}^{n} \frac{\partial -\log(P_i)}{\partial w_1}$$

在每个点对 w_1 的偏导相加

连续要素：看网页时间

$$p(y=1)=\frac{1}{1+e^{w_1x_1+w_2x_2+\cdots+b}}$$

w是参数

x是要素

凸优化一阶方法（无约束）

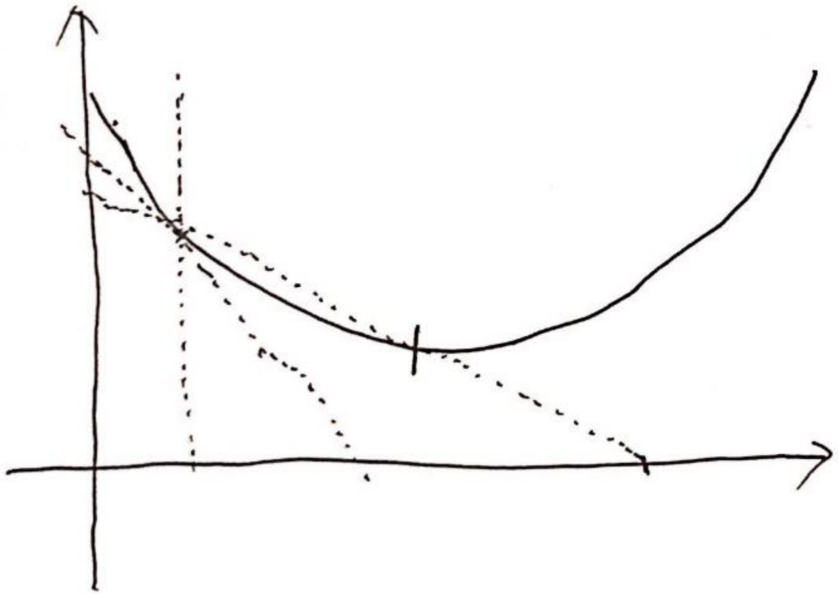
$f(x)$ $f'(x)$ 令 $\Delta x = -f'(x)$

$x \Leftarrow x + t\,\Delta x$

t：变化大小

Δx：变化方向

t怎么确定



1. 作x处关于$f(x)$的切线
2. 作x处~~关于~~
   作辅助线过$(x, f(x))$点，
   斜率是$\frac{1}{3}f'(x)$，记作$g(x)$
3. $t=1$
   重复 $t \Leftarrow 0.8 * t$
   直到 $f(x+t\Delta x)$
   $< g(x+t\Delta x)$

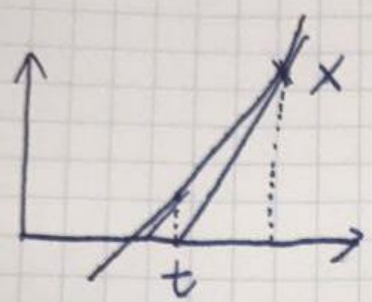
决定t的大小：回溯线搜索

$\Delta x = -\nabla f(x)$

backtracking line search (with parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$)

- starting at $t = 1$, repeat $t := \beta t$ until

$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$$



牛顿法求方程解

1、在 x 处做函数切线

2、与 x 轴交于 t 处

3. set $x=t$, goto 1

t 怎么算

$$f'(x)=\frac{f(x)}{x-t}$$

$$t=x-\frac{f(x)}{f'(x)}$$

牛顿求极值

等价于解 $f'(x)=0$

这个方程

牛顿法解 $f'(x)=0$

1、在 x 处做 $f'(x)=0$ 的切线

2、与 x 轴交于 t 处

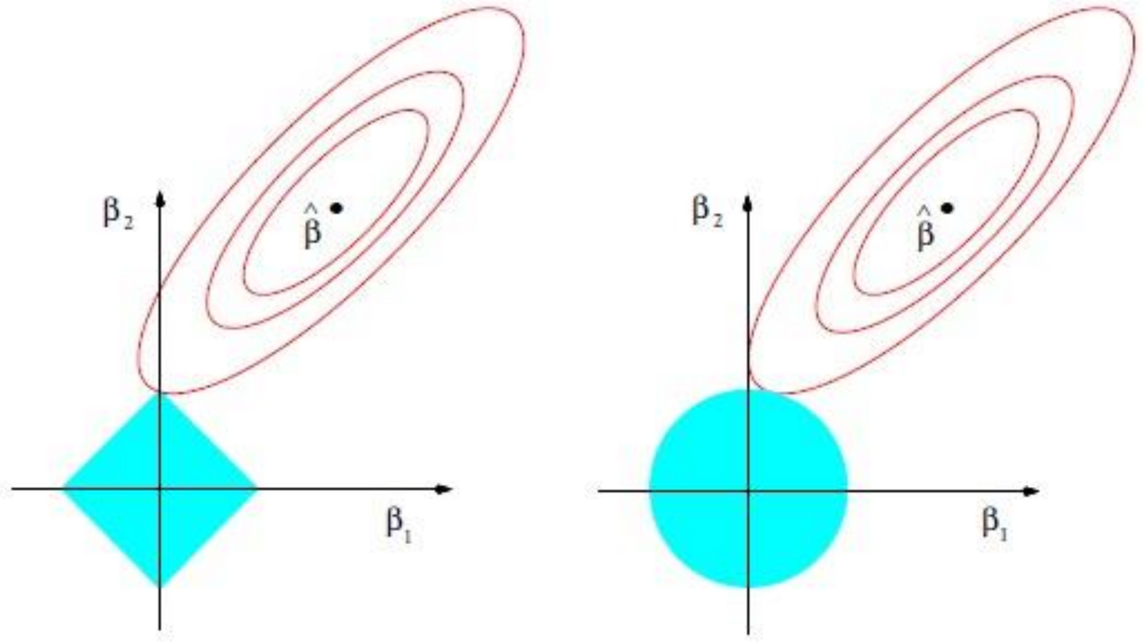
3、set $x=t$, goto 1

t 怎么算

$$f''(x)=\frac{f'(x)}{x-t} \quad t=x-\frac{f'(x)}{f''(x)}$$

- 模型泛化：正则化

$$L(\vec{w}) = \sum_{i=1}^{n} -\log(p_i) + l_2 \sum w^2 + l_1 \|w\|_1$$



- 模型泛化
  - 合适的复杂度
  - 正则化
  - Early stopping
  - Dropout

- 一边训练，一边看在训练数据与测试数据的表现

- 随机梯度下降

L ∇L

全体样本的 ΔL 经常过于耗时

部分样本求 ∇L

$$w = w - 0.1 \nabla L$$

学习率

随着学习不断下减小

- 监督学习
  - 经常是找到一个模型，根据数据找到模型的合适的参数（最小化损失函数+正则化项），使用模型进行回归或预测

  - 不全是这样的
    - 梯度提升决策树

- 梯度提升决策树(gradient boosting decision trees）
  - 参考：https://homes.cs.Washington.edu/~tqchen/pdf/BoostedTree.pdf
  - 经常是在全体样本上使用的一种方法

- 叠加训练

$$
\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}
$$

New function

Model at training round t

Keep functions added in previous round

- 逻辑回归：一个函数，改变参数，以减小损失函数

- GBDT：通过增加新的函数（树），以减小损失函数

- Model: assuming we have K trees

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$$

- Objective

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Training loss

Complexity of the Trees

- The prediction at round t is $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$

This is what we need to decide in round t

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

Goal: find f_t to minimize this

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

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

- Objective, with constants removed

$$\sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

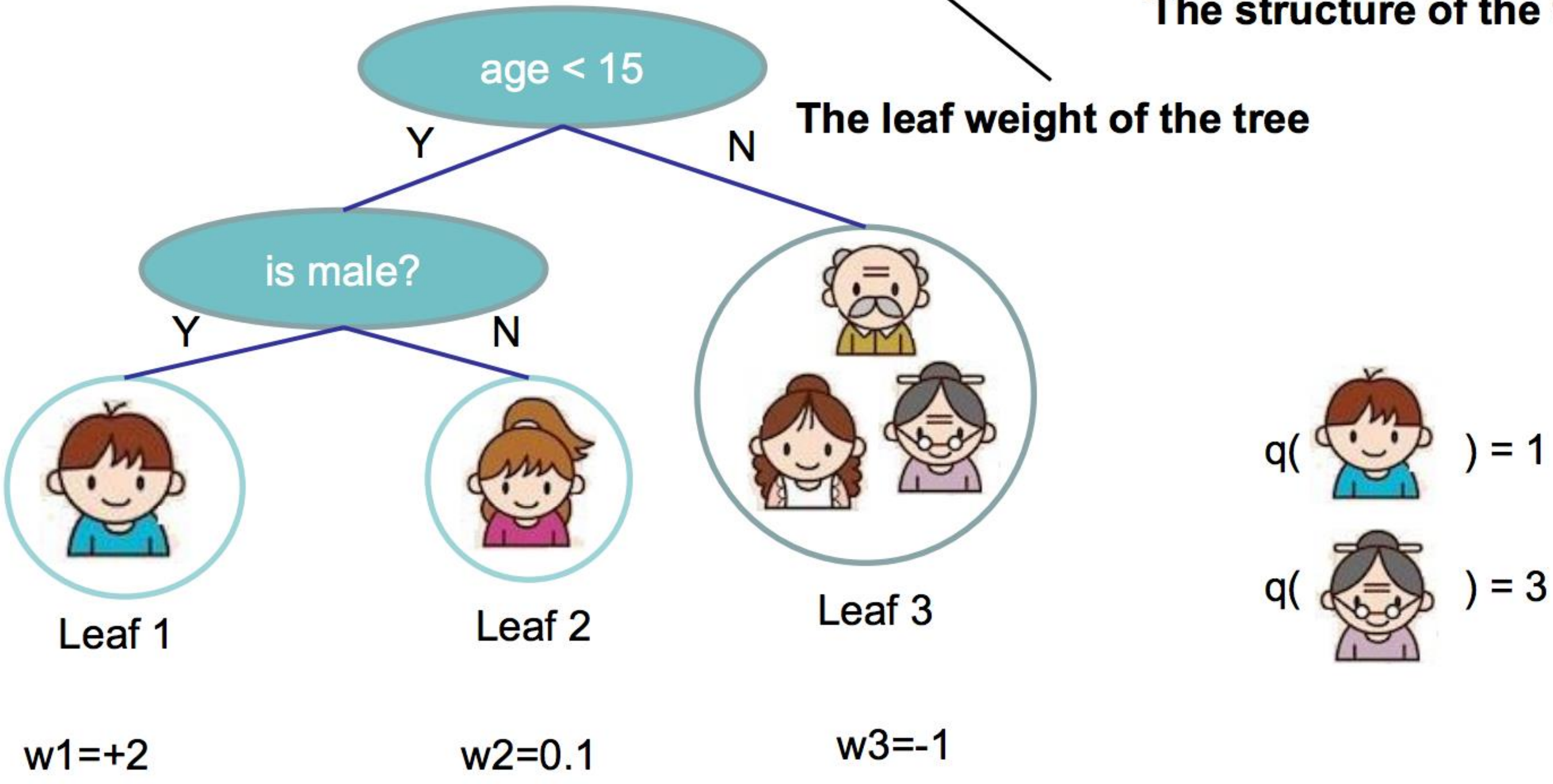
  - where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}), \quad h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$

- We define tree by a vector of scores in leafs, and a leaf index mapping function that maps an instance to a leaf

$$f_t(x) = w_{q(x)}, \quad w \in \mathbf{R}^T, q : \mathbf{R}^d \rightarrow \{1, 2, \cdots, T\}$$

The structure of the tree

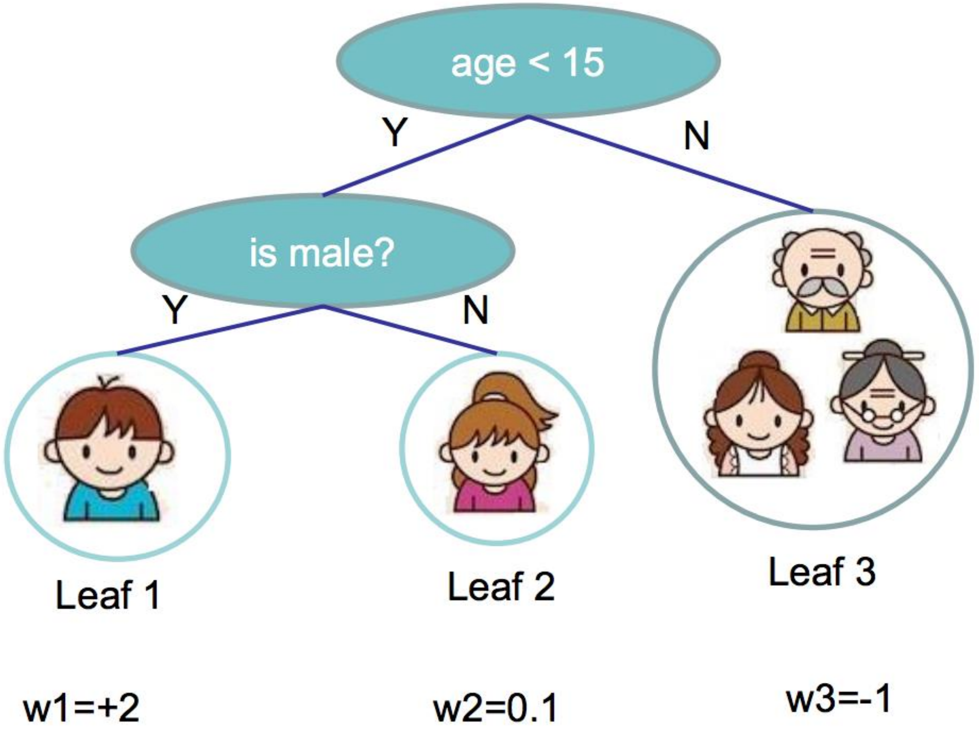
The leaf weight of the tree



- Define complexity as (this is not the only possible definition)

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

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



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

- Regroup the objective by each leaf

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

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

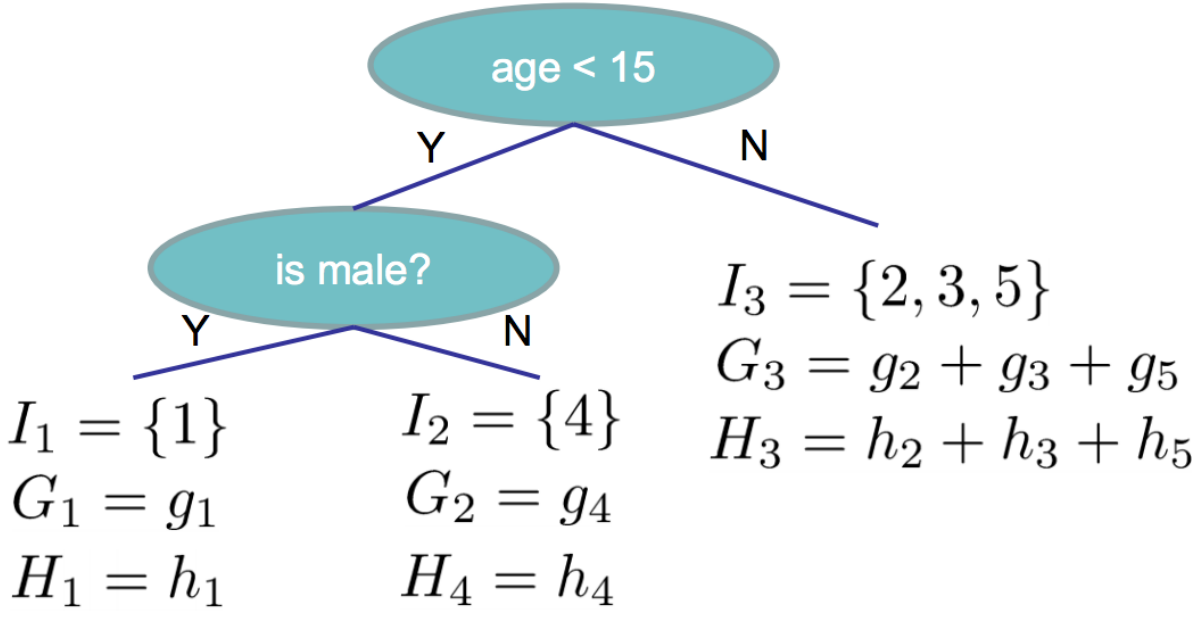
$$
\begin{aligned}
Obj^{(t)} &= \sum_{j=1}^{T}\left[(\sum_{i\in I_j} g_i)w_j + \tfrac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2\right] + \gamma T \\
&= \sum_{j=1}^{T}\left[G_j w_j + \tfrac{1}{2}(H_j + \lambda)w_j^2\right] + \gamma T
\end{aligned}
$$

- Assume the structure of tree (q(x)) is fixed, the optimal weight in each leaf, and the resulting objective value are

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

This measures how good a tree structure is!

Instance index		gradient statistics
1		g1, h1
2		g2, h2
3		g3, h3
4		g4, h4
5		g5, h5



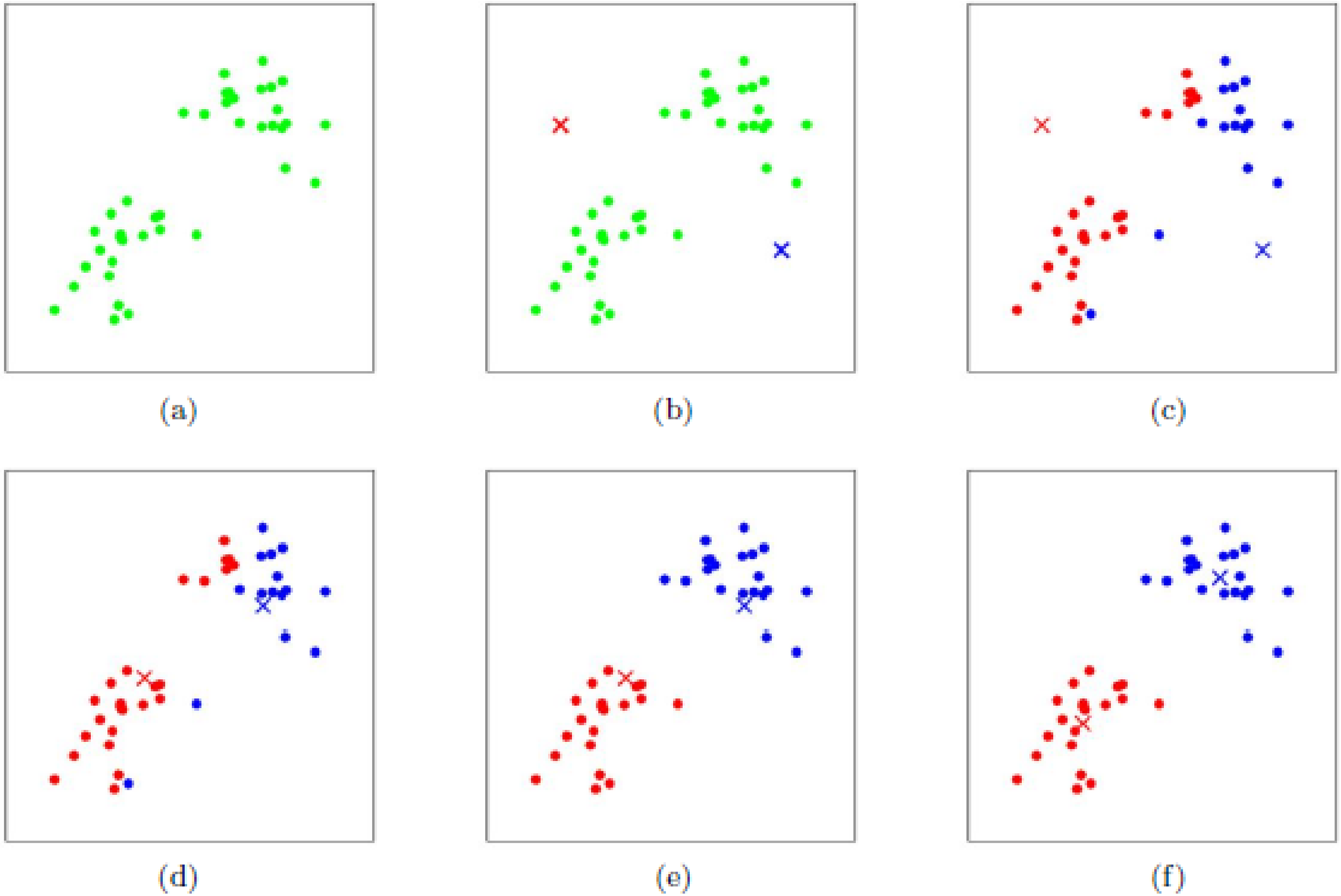
$$Obj = -\sum_j \frac{G_j^2}{H_j + \lambda} + 3\gamma$$

The smaller the score is, the better the structure is

- For each leaf node of the tree, try to add a split. The change of objective after adding the split is

$$Gain = \frac{1}{2}\left[\underbrace{\frac{G_L^2}{H_L+\lambda}}_{\text{the score of left child}} + \underbrace{\frac{G_R^2}{H_R+\lambda}}_{\text{the score of right child}} - \underbrace{\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}}_{\text{the score of if we do not split}}\right] - \underbrace{\gamma}_{\text{The complexity cost by introducing additional leaf}}$$

K-means



- 以上