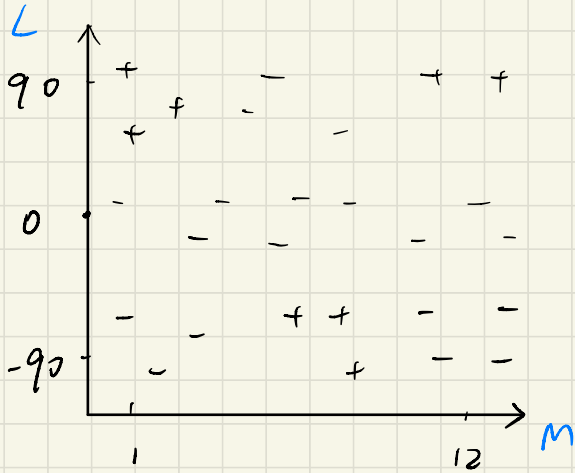
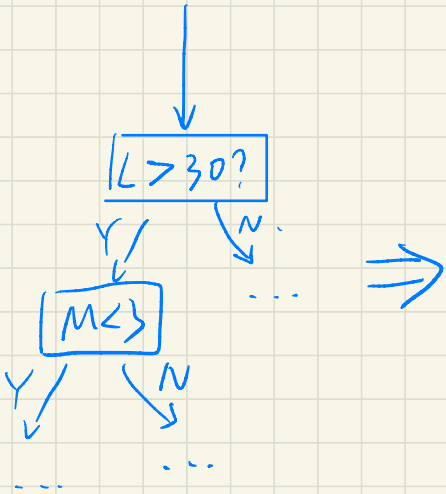
eg. ski.



intuition: non-linear.

Greedy top-Down Recursive partition.



Region: R_p.

Looking for split S_p.

S.t. $S_p(j,t) = (\{x \mid x_j < t, x \in R_p\})$ R_1

$(\{x \mid x_j \geq t, x \in R_p\})$ R_2

feature 门限

$\Downarrow$

how to split?

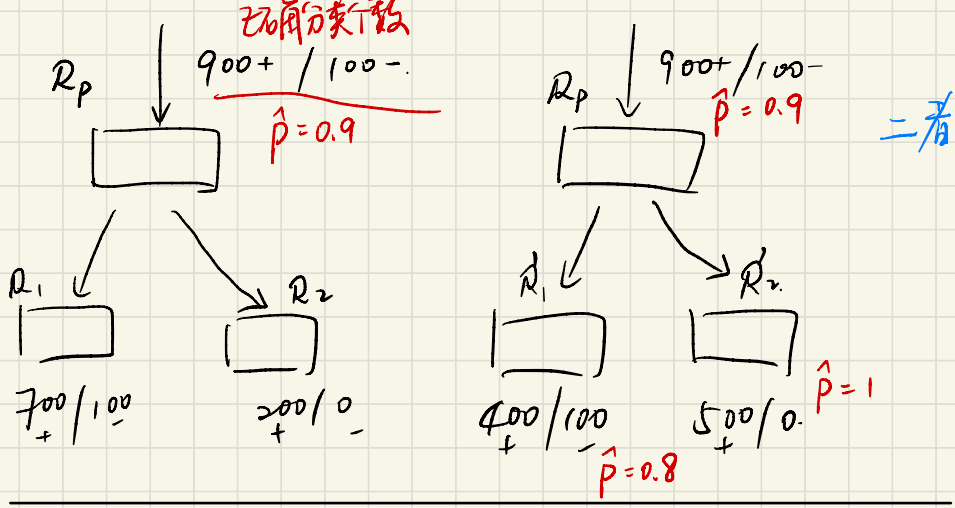
Define Loss(R): loss for partition

$\Rightarrow$ Given C class, define $\hat{p}_c$ to be to proportion of the sample in R that are class c.

$\Rightarrow$ $\mathcal{L}oss = 1 - \max_c \hat{p}_c \Rightarrow$ 不够准确

$\max_{j,t} \underbrace{\mathcal{L}(R_p)}_{父} - (\underbrace{\mathcal{L}(R_1) + \mathcal{L}(R_2)}_{子})$

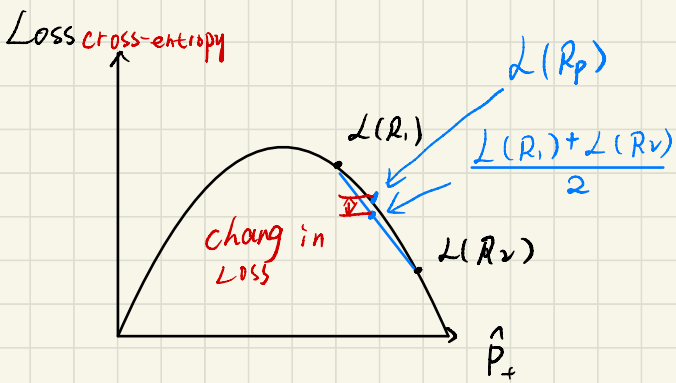
Misclassification Loss



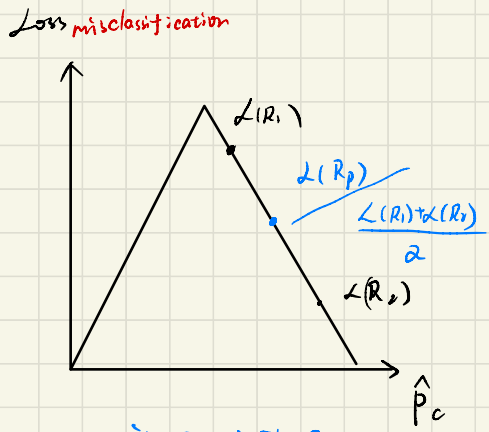
二者 Loss 一样

Cross entropy loss — Loss 标准

$$\mathcal{L}_{cross} = -\sum_c \hat{p}_c \log_2 \hat{p}_c$$

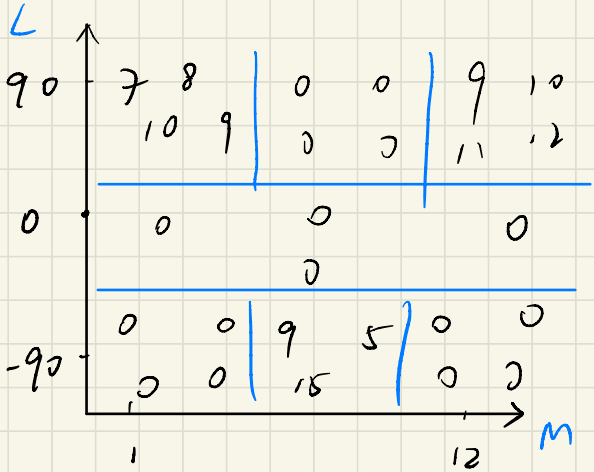


better.
Sensitive.



in-sensitive

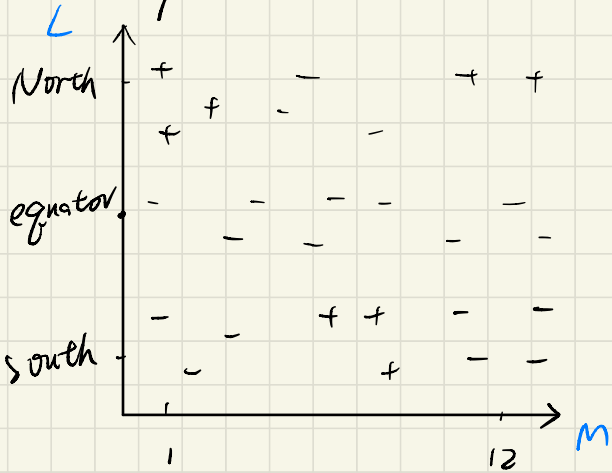
$$Loss_{Gini} = \sum_c \hat{p}_c (1-\hat{p}_c) \longrightarrow$$ 与 cross entropy 曲线差不多



predict $\hat{y}_m = \dfrac{\sum_{i \in R_m} y_i}{|R_m|}$

$$L_{square} = \frac{\sum_{i \in R_m} (y_i - \hat{y}_m)^2}{|R_m|}$$

Catergorical



$\boxed{Loc \in \{N\}}$

一共有8个类，会有2^8个分类

如果一直回分 最终会将所有点分开 → high var

Regulation

1) 叶节点个数

2) 最大深度

3) 最大节点个数

4) 最小减少 Loss — 可能要先问些不好的问题，才能会有最优

5) 长完整个树.

Run time.

n : example
f : feature
d : depth

Test time.

$O(d)$
$d = \log_2 n$

Train time.

Each point part of $O(d)$ nodes

Cost of partition : $O(f)$

$\Rightarrow O(nfd)$

缺点

① No additive structure

② high variance.

③ Low predictive accrecy

优点

① 易于计算，结果可以 interpret

② 快

③ 处理类别

} ensamble method

Ensembling

- X_i 为独立同分布的随机变量

$Var(X_i) = \sigma^2$, $Var(\bar{X}) = \frac{\sigma^2}{n}$

- 不独立，但同分布？ ← 不同方法结合

ρ 为相关系数：$Var(\bar{X}) = \rho\sigma^2 + \frac{1-\rho}{n}\sigma^2$

1) 用不同算法，结合到一起

2) 不同训练集

3) bagging (Random forest)

4) boosting (Adaboost, xgboost)

Bagging

· bootstrap aggregation.

有一真实分布 P. Training Set : S

认为 S = P.

再从S中取样. 得到 M个 子集 : $Z_1, Z_2, \cdots, Z_m$

后分别训练, 再综合输出

$\Rightarrow$ $G_m \longrightarrow Z_m$.

$$G(m) = \frac{\sum G_m(x)}{M}$$

· Bias - Variaus 分析

bootstraping : $\rho\downarrow$ $m\uparrow$

更显著 ← · $$Var(\bar{X}) = \rho\sigma^2 + \frac{1-\rho}{m}\sigma^2$$

· 但增加了 Bias $\longrightarrow$ 在 less data 上训练

DTs + Bagging :

Bias ↓ + Bias ↑ $\Longrightarrow$ Good !

Var ↑ var ↓

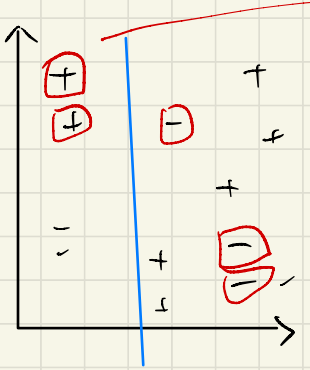
Random Forest.

- 每次分的时候只考虑 feature 部分. ⇒ de-correlating
- 避免 DTs 在第一步总是采取很好的方法

Boosting

- Bias↓ ⟶ 加 DTs depth 限制在 1
- Additive.

下一次计算时增加 weight



$$\Rightarrow \log \frac{1 - err_m}{err_m}$$

$$G(m) = \sum \alpha_m G_m$$

↓

在个 Reweighted 的 set