Ensemble Learning
- Boosting：serial，必须逐一训练
  - AdaBoost
  - Gradient Boosting
- 可同时训练
  - Bagging
  - Random Forest
- Stacking

AdaBoost

Greedily minimize the exponential loss function.
After each base learner is trained, adjust the data distribution so that the next base learner minimizes the cumulative loss.

Model:

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

$$\min_H \ell_{exp}(H \mid D) = E_x\left[e^{-y_i \hat{y}_i}\right]$$

$$= E_x\left[e^{-f(x_i) H(x_i)}\right]$$

Derivation:

① α_t：使 $\alpha_t h_t$ minimize loss function：（greedy）

$$\ell_{exp}(\alpha_t h_t \mid D_t)$$

$$= E_x\left[\, e^{-f(x)\alpha_t h_t(x)} \,\right]$$

$$= e^{-\alpha_t} \cdot P\{f(x)=h_t(x)\} + e^{\alpha_t} \cdot P\{f(x)\neq h_t(x)\}$$

$$\frac{\partial}{\partial \alpha_t} = -e^{-\alpha_t} \cdot P\{f(x)=h_t(x)\} + e^{\alpha_t} \cdot P\{f(x)\neq h_t(x)\} = 0$$

$$\Rightarrow \quad \alpha_t = \frac{1}{2} \ln\left(\frac{P\{f(x)=h_t(x)\}}{P\{f(x)\neq h_t(x)\}}\right) = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

② h_t：希望其纠正 H_{t-1} 的错误，即 min cumulative error：

$$l_{exp}(H_{t-1} + h_t \mid D)$$

$$= E_x\left[e^{-f(x)(H_{t-1}(x)+h_t(x))}\right]$$

$$= E_x\left[e^{-f(x)H_{t-1}(x)} e^{-f(x)h_t(x)}\right]$$

$$\approx E_x\left[e^{-f(x)H_{t-1}(x)}\left(1-f(x)h_t(x)+\frac{f^2(x)h_t^2(x)}{2}\right)\right] \quad \leftarrow e^x \approx 1+x+\frac{x^2}{2}$$

$$= E_x\left[e^{-f(x)H_{t-1}(x)}\left(1-f(x)h_t(x)+\frac{1}{2}\right)\right] \quad \leftarrow f^2(x)=h_t^2(x)=1$$

$$h_t = \arg\min_h l_{exp}(H_{t-1}+h \mid D)$$

$$= \arg\max_h E_x\left[e^{-f(x)H_{t-1}(x)} \cdot f(x)h_t(x)\right]$$

$$= \arg\max_h E_{x\sim D}\left[\frac{e^{-f(x)H_{t-1}(x)}}{E_{x\sim D}[e^{-f(x)H_{t-1}(x)}]} f(x)h_t(x)\right]$$

$$D_t(x) = \frac{D(x)e^{-f(x)H_{t-1}(x)}}{E_{x\sim D}[e^{-f(x)H_{t-1}(x)}]}$$

$$= \arg\max_h E_{x\sim D_t}[f(x)h_t(x)]$$

$$= \arg\min_h E_{x\sim D_t}[\mathbb{I}\{f(x)\neq h_t(x)\}] \Rightarrow$$ 即 minimize 分类误差

③ $D_{t+1}(x) = D(x)\dfrac{e^{-f(x)H_t(x)}}{E_{x\sim D}[e^{-f(x)H_t(x)}]}$

$= D(x)\dfrac{e^{-f(x)H_{t-1}(x)} \cdot e^{-f(x)h_t(x)} \cdot E_{x\sim D}[e^{-f(x)H_{t-1}(x)}]}{E_{x\sim D}[e^{-f(x)H_{t-1}(x)}] \cdot E_{x\sim D}[e^{-f(x)H_t(x)}]}$

$= D_t(x) \cdot e^{-f(x)h_t(x)} \cdot \dfrac{E_{x\sim D}[e^{-f(x)H_{t-1}(x)}]}{E_{x\sim D}[e^{-f(x)H_t(x)}]}$

特点 : ① Each weak learner is trained greedily to minimize the exponential loss function

② After each training, data are reweighted in such a way that allows future learners to focus on improving the data points that are hard to classify.

Gradient Boosting

A generalization of AdaBoost. 特点为 any differentiable loss function can be used.

$$F(x) = \sum_{m=1}^{M} F_m(x)$$

$$F_m(x) = F_{m-1}(x) + \beta_m \cdot h(x; a_m)$$

$$(\beta_m, a_m) = \underset{\beta, a}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, F_{m-1}(x) + \beta \cdot h(x; a))$$

For binary loss, $L(y_i, \hat{y}_i) = -(y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i))$
(cross-entropy)

XGBoost

Pro: - has regularization
- flexibility: customize objective and evaluation metric
- learns the best approach to handle missing values
- make split upto the max_tree_depth, then perform pruning

lightfm:

pro: - can be 20x faster than XGBoost while having the same accuracy
- low memory usage

stochastic gradient boosting :

At each round, a random subsample of data is used as trainingset.

Bagging

通过 Bootstrapping 采样出T个含有m样本的采样集.
分别训练 base learning.
采取 voting (classification) / averaging (regression)
的方式 aggregate result.

pro:
- 适用于 binary, multi-classification, regression
- 每个 base learner 未被 sample 的 data points 可作为 out-of-bag estimate
- 降低 variance
- Time Complexity 与 base learner 相同

Random Forest

在 Bagging 基础上, 加入随机属性选择.
At every split, choose the best feature out of a random sample of k features ($k = \log_2 d$)
d: total # of features

Stacking

Train a meta-learner to map the predictions of individual base learners to a final output value.

为防止 Data leakage / Overfitting :