发件人: Yulin Wu yw4923@nyu.edu

日期: 2020年9月9日 下午5:10

收件人:



Aggregation

2020年8月28日 星期五 上午11:55

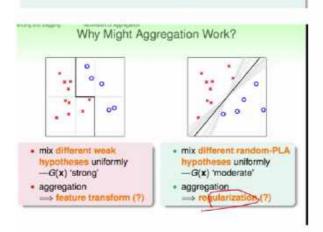
15个朋友给了你15个意见(model g_t), 你如何去运用你朋友的意见呢?

Aggregation with Math Notations Your T friends g_1, \dots, g_T predicts whether stock will go up as $g_t(\mathbf{x})$. - select the most trust-worthy friend from their usual performance $G(\mathbf{x}) = g_{t_r}(\mathbf{x})$ with $t_r = \operatorname{argmin}_{t \in \{1,2,\dots,T\}} E_{\text{val}}(g_t^-)$ - mix the predictions from all your friends uniformly $G(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \mathbf{1} \cdot g_t(\mathbf{x})\right)$ · mix the predictions from all your friends non-uniformly $G(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot g_t(\mathbf{x})\right) \text{ with } \alpha_t \ge 0$ • include select: $\alpha_t = [E_{\rm sal}(g_t) \text{ smallest}]$ • include uniformly: $\alpha_t = 1$ · combine the predictions conditionally $G(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} g_t(\mathbf{x}) \cdot g_t(\mathbf{x})\right) \text{ with } q_t(\mathbf{x}) \ge 0$

第一种情形:

选出一个最强的(model selection)。但若你的朋友都是弱弱的,就gg

而aggreation真正想做的是: 三个臭皮匠、胜过一个诸葛亮



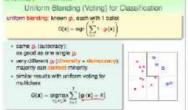
Aggregation includes: blending and bagging

Uniform Blending:

一个很直观的理论是:

g_t同质化严重,blending后的结果as good as one g; 差异大的g blending后才能有更好的效果

先直观地理解这个理论:



Uniform Blending for Regression $G(x) = \frac{1}{7} \sum_i g_i(x)$



严谨地证明:

Theoretical Analysis of Uniform Blending

$$G(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} g_t(\mathbf{x})$$

$$avg ((g_t(x) - f(x))^2) = avg (g_t^2 - 2g_t f + f^2)$$

$$= avg (g_t^2) - 2Gf + f^2$$

$$= avg (g_t^2) - G^2 + (G - f)^2$$

$$= avg (g_t^2) - 2G^2 + G^2 + (G - f)^2$$

$$= avg (g_t^2 - 2g_t G + G^2) + (G - f)^2$$

$$= \text{ avg}((g_f - G)^2) + (G - f)^2$$

$$avg(E_{out}(g_t)) = avg(E(g_t - G)^2) + E_{out}(G)$$

对每一个x来看, $g_t(x)$ - f(x) 的平方代表了 g_t 这个model在预测x上的error,可以用 E_t Out(g_t)来代替。 $avg(E_out(g_t))$ 就是,所有model预测x的偏差的平均。这里的意义是,如果你每次只选择一个g,那么你 能期待的表现是avg(E_out(g_t))。

(G-f)的平方代表了G这一个model在预测x上的error。而G是谁呢? 就是uniform Blending后的产物。3个 臭皮匠结合后的产物。(G-f)的平方就是3个臭皮匠结合后的预测。

等式说明:

blending后的预测(3个臭皮匠的结合)比每次只选择一个g来预测(臭皮匠单独预测)要好!而且要好 avg((g_t-G)^2)那么多。

而且I avg($I(g_t - G)^2$) 代表了g之间的方差。

Some Special gt

consider a virtual iterative process that for t = 1, 2, ..., T

- o request size-N data Dr from PN (i.i.d.)
- \odot obtain g_t by $A(\mathcal{D}_t)$

$$\tilde{g} = \lim_{T \to \infty} G = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} g_t = \frac{\mathcal{E}}{\mathcal{D}} \mathcal{A}(\mathcal{D})$$

$$avg(E_{out}(g_t)) = avg(E(g_t - \bar{g})^2) + E_{out}(\bar{g})$$

expected performance of A = expected deviation to consumsus

+performance of consensus

- performance of consensus: called bias
- · expected deviation to consensus: called variance

uniform blending:

reduces variance for more stable performance

Linear Blending

Linear Blending

linear blending: known gt, each to be given of ballot

$$G(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{T} \alpha_i \cdot g_i(\mathbf{x})\right) \text{ with } \alpha_i \ge 0$$

computing 'good' α_t : min $E_{in}(\alpha)$

like two-level learning, remember? :-)

linear blending = LinModel + hypotheses as transform + constra

其实这里的 $\alpha \ge 0$ 的限期可以去掉。

ok,now, how to 训练这个model呢?

首先、关于如何解这个目标函数、要么用linear regression的解析解、要么用gradien descent.

然后,这个blending的input和output如何构造呢?还可以用原本训练集的数据吗?

Linear Blending versus Selection

in practice, often

$$g_1 \in \mathcal{H}_1, g_2 \in \mathcal{H}_2, \dots, g_T \in \mathcal{H}_T$$

by minimum Ein

- · recall: selection by minimum Ein
- —best of best, paying $d_{VG} \left(\bigcup \mathcal{H}_{f} \right)$
- · recall: linear blending includes selection as special case
- —by setting $\mathbf{o}_t = [E_{\text{val}}(g_t^-) \text{ smallest}]$
- complexity price of linear blending with E_n (aggregation of best); ≥d_{vo} (UH)

1. D分为D train & D val, use D train to train g7, g7, ..., g7 这一步可以看作feature transform.

If use $g = [g_1^-, g_2^-, ..., g_T^-]$ directly as input, y_T train as output to train the blending model (如何 将朋友的意见加权平均),what would happen?

注: model selection is a special blending, how does it work? It selects the best model based on validation error instead of in-sample error. Essentially, $[g_1^-(Dval),...,g_T^-(Dval)]$ as input, and y_val as output,演算法就是手动选出最接近的那个g-

Similarly, if I use g as input and y_train as output, I will definitely choose the best model or construct a even more complicated model, (ex: 1*model_1 + 2 * model_2). So I can't do this way.

Now I throw Dtrain out. 不再使用训练集了,只剩下验证集。

- 2. 模仿model selection,用g格验证集预测出来,so I have $z_{val} = [g_1(Dval), g_2(Dval), ..., g_1(Dval)]$ As input, and y_val as output. 在验证集上训练s.t. $\min_{\alpha} \sum (y_{val_n} - \alpha^T z_{val_n})^2$
- 3. Finally, obviously, 选出最优的hypothesis后(ex: the second model is uniform blending, and the first model is XGB, SVM, LASSO), 再用所有的D fit一次, 选出最优的w。

最后,如何构造g的diversity

learning g_t for uniform aggregation: diversity important • diversity by different models: $g_1 \in \mathcal{H}_1, g_2 \in \mathcal{H}_2, \dots, g_7 \in \mathcal{H}_7$ diversity by different parameters: GD with y = 0.001, 0.01, ..., 10 . diversity by algorithmic randomness: random PLA with different random seeds diversity by data randomness: within-cross-validation hypotheses g_r

next: diversity by data randomness without g

Bagging:

Revisit of Bias-Variance expected performance of A = expected deviation to consensus +performance of consensus consensus $\bar{g} = \text{expected } g_t \text{ from } \mathcal{D}_t \sim P^N$

最理想的情况是:

- 1. 有无限多个g 2. 每个can 每个g都有新的数据来训练

但是,太理想了,现实中要妥协

- finite but large g
 use bootstrapping (resample N examples with replacement) 去生成新的数据



Bagging 的效果:

