

发件人: Yulin Wu yw4923@nyu.edu
 主题:
 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**
 $G(\mathbf{x}) = g_{t_*}(\mathbf{x})$ with $t_* = \operatorname{argmin}_{t \in \{1, 2, \dots, T\}} E_{\text{val}}(g_t)$
- **mix** the predictions from all your friends **uniformly**
 $G(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^T 1 \cdot g_t(\mathbf{x})\right)$
- **mix** the predictions from all your friends **non-uniformly**
 $G(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^T \alpha_t \cdot g_t(\mathbf{x})\right)$ with $\alpha_t \geq 0$
 - include **select**: $\alpha_t = \mathbb{I}[E_{\text{val}}(g_t) \text{ smallest}]$
 - include **uniformly**: $\alpha_t = 1$
- **combine** the predictions **conditionally**
 $G(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^T q_t(\mathbf{x}) \cdot g_t(\mathbf{x})\right)$ with $q_t(\mathbf{x}) \geq 0$

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aggregation真正想做的是:

三个臭皮匠, 胜过一个诸葛亮

Why Might Aggregation Work?

- mix **different weak hypotheses** uniformly
 $\rightarrow G(\mathbf{x})$ 'strong'
- aggregation
 \Rightarrow **feature transform (?)**

- mix **different random-PLA hypotheses** uniformly
 $\rightarrow G(\mathbf{x})$ 'moderate'
- aggregation
 \Rightarrow **regularization (?)**

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 (Voting) for Classification

uniform blending: known g_i , each with 1 ballot

$$G(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^T 1 \cdot g_i(\mathbf{x})\right)$$

- same g_i (autocracy):
as good as one single g_i
- very different g_i (diversity + democracy):
majority can **correct** minority
- similar results with uniform voting for multiclass

$$G(\mathbf{x}) = \operatorname{argmax}_{1 \leq k \leq K} \sum_{i=1}^T \mathbb{I}[g_i(\mathbf{x}) = k]$$

Uniform Blending for Regression

$$G(\mathbf{x}) = \frac{1}{T} \sum_{i=1}^T g_i(\mathbf{x})$$

- same g_i (autocracy):
as good as one single g_i
- very different g_i (diversity + democracy):
some $g_i(\mathbf{x}) > f(\mathbf{x})$, some $g_i(\mathbf{x}) < f(\mathbf{x})$
 \Rightarrow average **could be** more accurate than individual

diverse hypotheses:
 even simple uniform blending
 can be better than any **single hypothesis**



严谨地证明:

Theoretical Analysis of Uniform Blending

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

$$\begin{aligned} \operatorname{avg}((g_t(\mathbf{x}) - f(\mathbf{x}))^2) &= \operatorname{avg}(g_t^2 - 2g_t f + f^2) \\ &= \operatorname{avg}(g_t^2) - 2Gf + f^2 \\ &= \operatorname{avg}(g_t^2) - G^2 + (G - f)^2 \\ &= \operatorname{avg}(g_t^2) - 2G^2 + G^2 + (G - f)^2 \\ &= \operatorname{avg}(g_t^2 - 2\alpha_t G + G^2) + (G - f)^2 \end{aligned}$$

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

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

等式的左边:

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 $E_{\text{out}}(g_t)$ 来代替。 $\text{avg}(E_{\text{out}}(g_t))$ 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 $\text{avg}(E_{\text{out}}(g_t))$ 。

等式的右边:

$(G-f)$ 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 $\text{avg}((g_t - G)^2)$ 那么多。

而且! $\text{avg}((g_t - G)^2)$ 代表了g之间的方差。

Some Special g_t

consider a **virtual** iterative process that for $t = 1, 2, \dots, T$

- request size- N data \mathcal{D}_t from P^N (i.i.d.)
- obtain g_t by $\mathcal{A}(\mathcal{D}_t)$

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

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

expected performance of $\mathcal{A} = \text{expected deviation to consensus} + \text{performance of consensus}$

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

uniform blending:
reduces **variance** for more stable performance

Hsuan-Tien Li (NTU CSIE) Machine Learning Techniques 10/29

Linear Blending

Linear Blending

linear blending: known g_t , each to be given α_t ballot

$$G(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^T \alpha_t \cdot g_t(\mathbf{x})\right) \text{ with } \alpha_t \geq 0$$

computing 'good' α_t : $\min_{\alpha_t \geq 0} E_{\text{in}}(\alpha)$

linear blending for regression

$$\min_{\alpha_t \geq 0} \frac{1}{N} \sum_{n=1}^N \left(y_n - \sum_{t=1}^T \alpha_t g_t(\mathbf{x}_n) \right)^2$$

LinReg + transformation

$$\min_{w_i} \frac{1}{N} \sum_{n=1}^N \left(y_n - \sum_{i=1}^{\tilde{d}} w_i \phi_i(\mathbf{x}_n) \right)^2$$

like two-level learning, remember? :-)

linear blending = LinModel + hypotheses as transform + constraints

其实这里的 $\alpha \geq 0$ 的限制可以去掉。

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gradient 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?

blending and bagging Linear and Any Blending

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 E_{in}

- recall: **selection by minimum E_{in}**
—best of best, paying $d_{\text{vc}}\left(\bigcup_{t=1}^T \mathcal{H}_t\right)$
- recall: linear blending includes **selection** as special case
—by setting $\alpha_t = \llbracket E_{\text{val}}(g_t) \rrbracket$ smallest
- complexity price of linear blending with E_{in} (aggregation of best):
 $\geq d_{\text{vc}}\left(\bigcup_{t=1}^T \mathcal{H}_t\right)$

1. D分为D_train & D_val, use D_train to train $g_1^-, g_2^-, \dots, g_T^-$ 这一步可以看作feature transform.

If use $g = [g_1^-, g_2^-, \dots, g_T^-]$ directly as input, y_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, $[g_1^-(D_{val}), \dots, g_T^-(D_{val})]$ 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: $1 * \text{model_1} + 2 * \text{model_2}$). 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将验证集预测出来, so I have $z_{val} = [g_1^-(D_{val}), g_2^-(D_{val}), \dots, g_T^-(D_{val})]$ 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: $g_1 \in \mathcal{H}_1, g_2 \in \mathcal{H}_2, \dots, g_T \in \mathcal{H}_T$
- diversity by different parameters: GD with $\eta = 0.001, 0.01, \dots, 10$
- diversity by algorithmic randomness: random PLA with different random seeds
- diversity by data randomness: within-cross-validation hypotheses g_v^-

next: diversity by data randomness without g^-

Bagging:

Revisit of Bias-Variance

expected performance of \mathcal{A} = expected deviation to consensus
+ performance of consensus

consensus \bar{g} = expected g_t from $\mathcal{D}_t \sim P^N$

最理想的情况是:

1. 有无限多个g
2. 每个g都有新的数据来训练

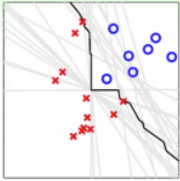
但是, 太理想了, 现实中要妥协

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

virtual aggregation	bootstrap aggregation
consider a virtual iterative process that for $t = 1, 2, \dots, T$	consider a physical iterative process that for $t = 1, 2, \dots, T$
① request size-N data \mathcal{D}_t from P^N (i.i.d.)	① request size- N' data $\tilde{\mathcal{D}}_t$ from bootstrapping
② obtain g_t by $\mathcal{A}(\mathcal{D}_t)$	② obtain g_t by $\mathcal{A}(\tilde{\mathcal{D}}_t)$
$G = \text{Uniform}(g_t)$	$G = \text{Uniform}(g_t)$

Bagging 的效果:

Bagging Pocket in Action



$T_{\text{POCKET}} = 1000; T_{\text{BAG}} = 25$

- very diverse g_t from bagging
- proper **non-linear** boundary after aggregating binary classifiers

已使用 OneNote 创建。