

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  $g_i$ ), 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  $g_i(\mathbf{x})$ .

- **select** the most trust-worthy friend from their **usual performance**  
 $G(\mathbf{x}) = g_{l_*}(\mathbf{x})$  with  $l_* = \operatorname{argmin}_{l \in \{1, 2, \dots, T\}} E_{\text{val}}(g_l)$
- **mix** the predictions from all your friends **uniformly**  
 $G(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^T \frac{1}{T} \cdot g_i(\mathbf{x})\right)$
- **mix** the predictions from all your friends **non-uniformly**  
 $G(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^T \alpha_i \cdot g_i(\mathbf{x})\right)$  with  $\alpha_i \geq 0$ 
  - include **select**:  $\alpha_{l_*} = \frac{1}{T} E_{\text{val}}(g_{l_*})$  smallest
  - include **uniformly**:  $\alpha_i = \frac{1}{T}$
- **combine** the predictions **conditionally**  
 $G(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^T \underbrace{q_i(\mathbf{x})}_{\text{weight}} \cdot \underbrace{g_i(\mathbf{x})}_{\text{prediction}}\right)$  with  $q_i(\mathbf{x}) \geq 0$

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  
—  $G(\mathbf{x})$  'strong'
- aggregation  
⇒ **feature transform (?)**

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

Aggregation includes: blending and bagging

### Uniform Blending:

一个很直观的理论是:

$g_i$  同质化严重, blending后的结果 as good as one  $g_i$ ; 差异大的  $g_i$  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

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

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

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

**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  $g_i(\mathbf{x}) < f(\mathbf{x})$   
⇒ average could be more accurate than individual

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

严谨地证明:

**Theoretical Analysis of Uniform Blending**

$$G(\mathbf{x}) = \frac{1}{T} \sum_{i=1}^T g_i(\mathbf{x})$$
$$\begin{aligned} \operatorname{avg}((g_i(\mathbf{x}) - f(\mathbf{x}))^2) &= \operatorname{avg}(g_i^2 - 2g_i f + f^2) \\ &= \operatorname{avg}(g_i^2) - 2Gf + f^2 \\ &= \operatorname{avg}(g_i^2) - G^2 + (G - f)^2 \\ &= \operatorname{avg}(g_i^2) - 2G^2 + G^2 + (G - f)^2 \\ &= \operatorname{avg}(g_i^2 - 2\alpha_i G + G^2) + (G - f)^2 \end{aligned}$$

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

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

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。这里的意思是, 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  $t = 1, 2, \dots, T$

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

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

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

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

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

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

HARVEY K. LO (NYU CS)     Statistical Learning Techniques     11/25

## Linear Blending

**Linear Blending**

linear blending: known  $g_t$ , each to be given  $\alpha_t$  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'  $\alpha_t$ :  $\min_{\alpha_t \geq 0} E_{\text{in}}(\alpha)$

**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**

$$\min_w \frac{1}{N} \sum_{n=1}^N \left( y_n - \sum_{j=1}^{\tilde{d}} w_j \phi_j(\mathbf{x}_n) \right)^2$$

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?

learning and testing     Linear mix any learning

**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  $E_{\text{in}}$

- recall: **selection by minimum  $E_{\text{in}}$**   
—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  $\alpha_t = [E_{\text{in}}(g_t)]$  smallest
- complexity price of linear blending with  $E_{\text{in}}$  (**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:  $1 * model_1 + 2 * model_2$ ). So I can't do this way.

Now I throw  $D_{train}$  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.  $\min_a \sum (y_{val,n} - a^T z_{val,n})^2$
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  $g_i$  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  $g_i^-$

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  $g_i$  from  $\mathcal{D}_i \sim \mathcal{P}^N$

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

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

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

### bootstrap aggregation

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

- 1 request size- $N$  data  $\bar{\mathcal{D}}_t$  from bootstrapping
  - 2 obtain  $g_t$  by  $\mathcal{A}(\bar{\mathcal{D}}_t)$
- $G = \text{Uniform}(g_t)$

## Bagging 的效果:

### Bagging Pocket in Action



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

- very diverse  $g_i$  from bagging
- proper non-linear boundary after aggregating binary classifiers

已使用 OneNote 创建。