Blending and Bagging

一、简介

我们像调制鸡尾酒一样来对待我们备选的模型，我们大致有几种方案：

1. 最佳原则——validation
2. uniformly vote
3. non-uniformly vote
4. conditionally

建模，假设有T个备选的g，$g_1, g_2, \ldots, g_T$

$$\begin{aligned} G(x) &= g_{t_*}(x) \\ t_* &= argmin_{t \in 1,2,..,T} E_{val^-}(g_t^-) \end{aligned} \tag{1}$$

$$G(x) = sign(\sum_{t=1}^{T} 1 * g_t(x)) \tag{2}$$

$$\begin{gathered} G(x) = sign(\sum_{t=1}^{T} \alpha_t * g_t(x)) \\ with\ \alpha_t \geq 0 \end{gathered} \tag{3}$$

$$\begin{gathered} G(x) = sign(\sum_{t=1}^{T} q_t(x) g_t(x)) \\ with\ q_t(x) \geq 0 \end{gathered} \tag{4}$$

可以发现最后一种情况包山包海，可以把前三种情况算在它的范畴里面

二、Uniform Blending

1. Classification:

$$G(x) = sign(\sum_{t=1}^{T} 1 * g_t(x)) \tag{5}$$

其实类似于uniform voting for multiclass

$$G(x) = argmax_{1 \leq k \leq K} \sum_{t=1}^{T} [[g_t(x) = k]] \tag{6}$$

2. Regression:

$$G(x) = \frac{1}{T} \sum_{t=1}^{T} g_t(x) \tag{7}$$

3. Theoretical Analysis of Uniform Blending:

Theoretical Analysis of Uniform Blending

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

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

$$\begin{aligned}
\text{avg}\left(E_{\text{out}}(g_t)\right) &= \text{avg}\left(\mathcal{E}(g_t - G)^2\right) + E_{\text{out}}(G) \\
&\geq \qquad\qquad\qquad\qquad + E_{\text{out}}(G)
\end{aligned}$$

4. 传说中的Bias Variance 分析:

Some Special g_t

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

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

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

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

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

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

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

三、Linear and Any Blending

1. Linear Blending 和 LinReg+transformation的相同之处:

我们先知道g_t然后利用这个来计算α_t因此实际上类似于一个2-level的学习过程，类似于probabilistic的SVM模型，但是这里有一个区别就是他的α_t是 > 0的。

Linear Blending

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

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

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

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

Constraint on α_t

linear blending = LinModel + hypotheses as transform + constraints:

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

linear blending for binary classification

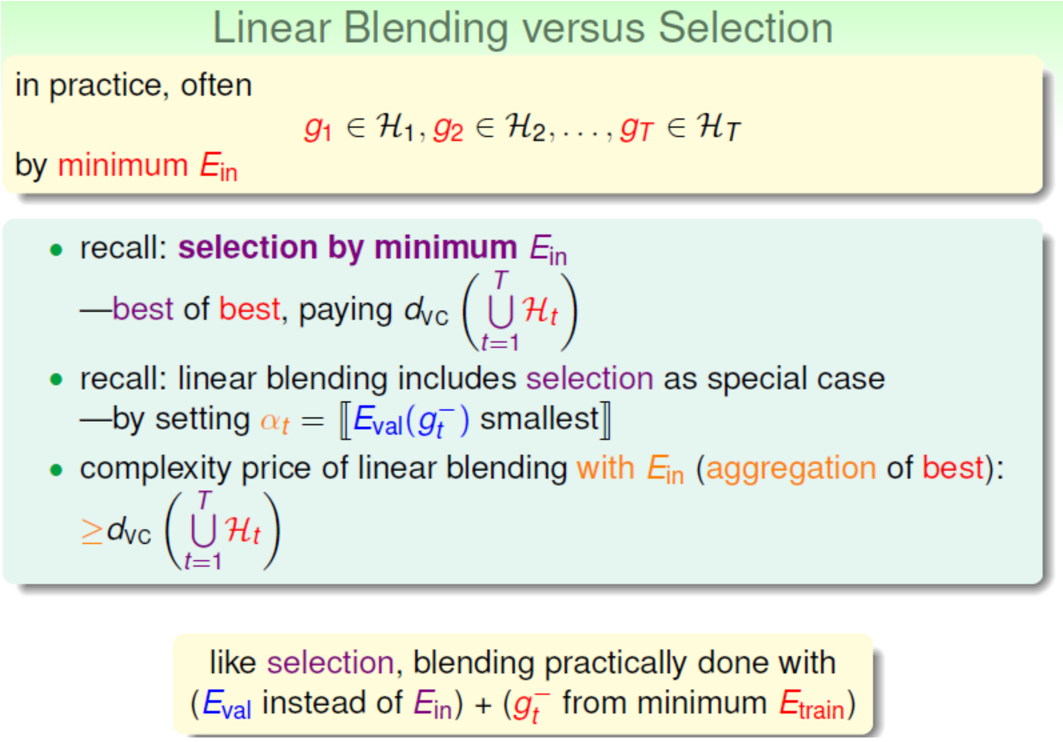
$$\text{if } \alpha_t < 0 \implies \alpha_t g_t(\mathbf{x}) = |\alpha_t| \left(-g_t(\mathbf{x})\right)$$

- negative α_t for $g_t \equiv$ positive $|\alpha_t|$ for $-g_t$
- **if you have a stock up/down classifier with 99% error, tell me! :-)**

in practice, often
linear blending = LinModel + hypotheses as transform ~~+ constraints~~

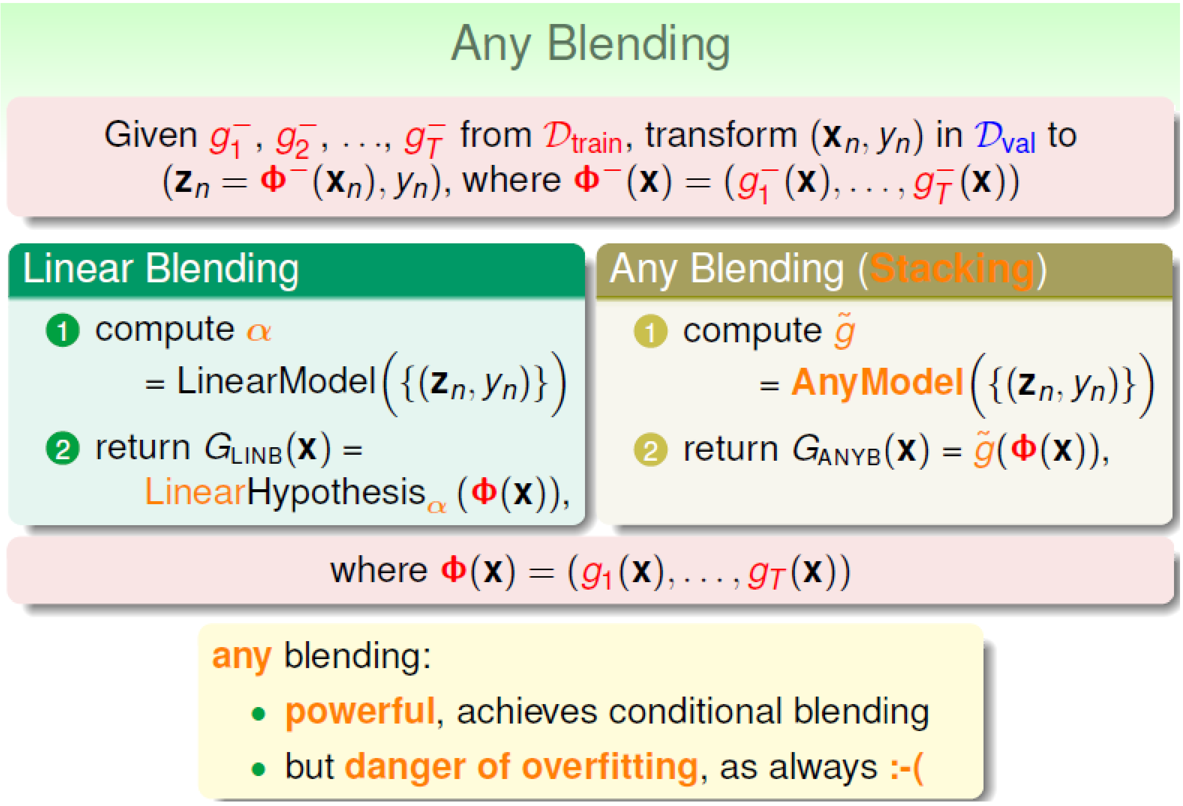
在实际过程中我们常常忽略constraints，把非正的结果想做事投反对票的思维。

2. 对比一下linear blending 和selection的模型，利用VC-dim来进行一些分析，我们发现其实这样做十分容易过拟合



我们在selection的时候要付出的VC-dim的代价已经很大了现在blending就更危险了。（因为selection其实是一种特例）

3. 其他的blending，又叫做stacking：



危险而诱人！

四、 Bagging (Bootstrap Aggregation)

1. 我们之前讲的方法都是blending，也就是混合鸡尾酒，但是事实上在learning的过程中我们希望找每个g_t的过程不需要在aggregate之前，这样的话我们可以提高效率，换句话说我们就是希望能让找g_t和blending能同步进行。
2. 还有一件特别重要的事情就是，我们尽量让g_t尽可能不同，发挥每个模型的特点的作用就是要让他们diverse这些diversity总结如下：

blending: aggregate after getting g_t;
learning: aggregate as well as getting g_t

aggregation type	blending	learning
uniform	voting/averaging	?
non-uniform	linear	?
conditional	stacking	?

learning g_t for uniform aggregation: diversity important

- diversity by different models: $g_1 \in \mathcal{H}_1, g_2 \in \mathcal{H}_2, \ldots, g_T \in \mathcal{H}_T$
- diversity by different parameters: GD with $\eta = 0.001, 0.01, \ldots, 10$
- 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^-

保持diversity的方式其实就是让data尽量保持随机性。

3. 保持随机性的方式比较好的就是使用同分布的很多数据，但是很多情况下数据是有限的，我们希望在有限的数据中发挥数据的最大效果，这就需要我们对于数据集做一点操作。于是我们就引入了bootstrapping aggregation（也就是bagging）

bootstrapping: a statistical tool that
re-samples from $\mathcal{D}$ to 'simulate' $\mathcal{D}_t$

4. 所谓bootstrapping的方法其实很简单，其实就是对于N个数据量的data做一次放回的抽样。

我们对比一下理想的学习过程和bootstrap的过程（实际情况下我们只能选择bagging而不是真的要再找规模为N的数据量）

bootstrapping

bootstrap sample $\tilde{\mathcal{D}}_t$: re-sample N examples from $\mathcal{D}$ **uniformly with replacement**—can also use arbitrary N' instead of original N

virtual aggregation

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

1. request size-N data $\mathcal{D}_t$ from 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 $t = 1, 2, \ldots, T$

1. request size-N' data $\tilde{\mathcal{D}}_t$ from bootstrapping
2. obtain g_t by $\mathcal{A}(\tilde{\mathcal{D}}_t)$

$G = \text{Uniform}(\{g_t\})$

bootstrap aggregation (BAGging):
a simple **meta algorithm**
on top of **base algorithm** $\mathcal{A}$

bagging相当于是在base algorithm A之上的meta algorithm，其实是对取样的操作。

我们bootstrap最后对于G的选择是uniform形式的。

对接下来的模型做一个总结：

1. Bagging —— uniformly vote

2. AdaBoost —— non-uniformly vote

3. Decision Tree & Random Forest —— conditionally