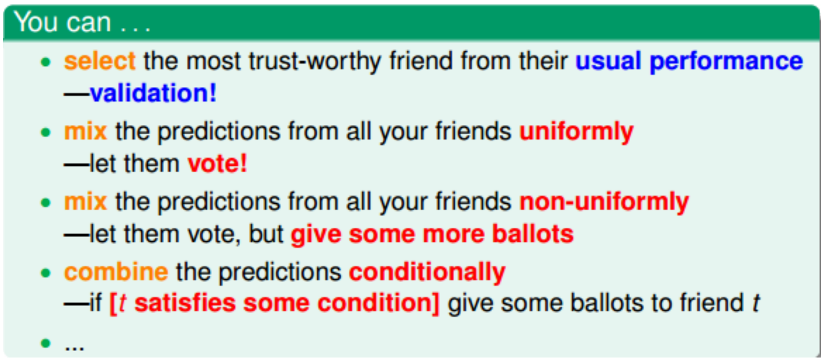
Blending and Bagging

主要内容：如何将不同的hypothesis和features结合起来，让模型更好。

1.Motivation of Aggregation



第一种方法对应的模型：

$$G(x) = g_{t_*}(x)\ with\ t_* = argmin_{t \in 1,2,\cdots,T}\ E_{val}(g_t^-)$$

第二种方法对应的模型：

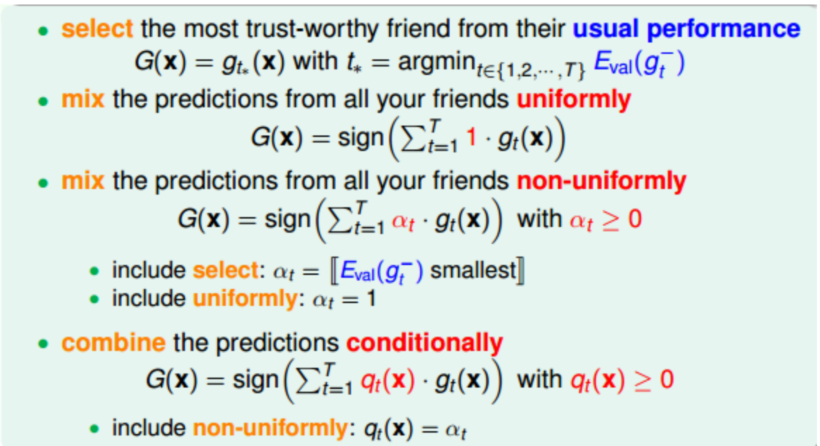
$$G(x) = sign(\sum_{t=1}^{T} 1 \cdot g_t(x))$$

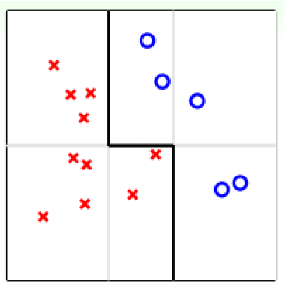
第三种方法对应的模型：

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

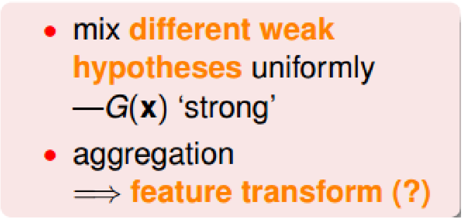
第四种方法对应的模型：

$$G(x) = sign(\sum_{t=1}^{T} q_t(x) \cdot g_t(x))\ with\ q_t(x) \geq 0$$



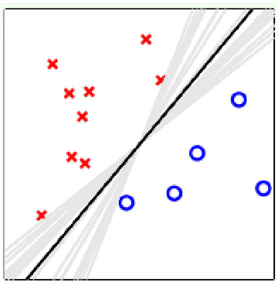


如果要求只能用一条水平的线或者垂直的线进行分类（即上述第一种方法：validation），那不论怎么选取直线，都达不到最佳的分类效果。
如果可以使用aggregate，比如一条水平线和两条垂直线组合而成的图中折线形式，就可以将所有的点完全分开，得到了最优化的预测模型。

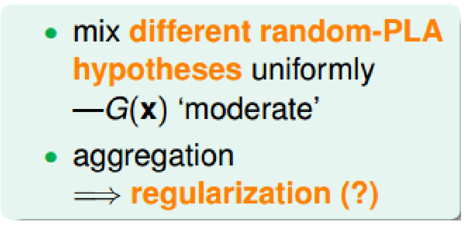


aggregation提高了预测模型的power，起到了特征转换的效果。

使用PLA算法，可以得到很多满足条件的分类线



aggregation也起到了正则化的效果，让预测模型更具有代表性。



aggregation的两个优势：feature transform和regularization。
feature transform和regularization是对立的。
如果进行feature transform，那么regularization的效果通常很差，反之亦然。也就是说，单一模型通常只能倾向于feature transform和regularization之一，在两者之间做个权衡。

aggregation能将feature transform和regularization各自的优势结合起来，，从而得到不错的预测模型。

2.Uniform Blending

uniform blending，应用于classification分类问题，做法是将每一个可能的矩赋予权重1，进行投票，得到的G(x)表示为：

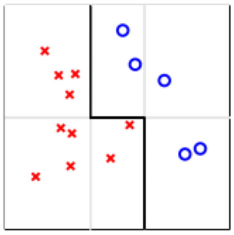
$$g(x) = sign(\sum_{t=1}^{T} 1 \cdot g_t(x))$$

这种方法对应三种情况：
1.每个候选的矩g_t都完全一样，这跟选其中任意一个g_t效果相同
2.每个候选的矩g_t都有一些差别，通过投票的形式使多数意见修正少数意见，从而得到很好的模型

3.多分类问题，选择投票数最多的那一类。

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

$$G(\mathbf{x}) = \underset{1 \le k \le K}{\operatorname{argmax}} \sum_{t=1}^{T} [\![g_t(\mathbf{x}) = k]\!]$$



uniform blending应用于regression，将所有的矩g_t求平均值：

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

uniform blending for regression对应两种情况：
1.每个候选的矩g_t都完全一样，这跟选其中任意一个g_t效果相同
2.每个候选的矩g_t都有一些差别，有的$g_t > f(x)$，有的$g_t < f(x)$，此时求平均值的操作可能会消去这种大于和小于的影响，从而得到更好的回归模型。因此，一般来说，求平均值的操作更加稳定，更加准确。

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

计算g_t的平均值可能比单一的g_t更稳定，更准确：

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

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

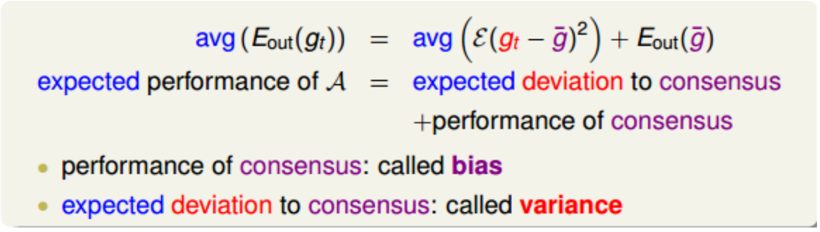
$avg(E_{out}(g_t)) \ge E_{out}(G)$，从而证明了从平均上来说，计算$g_t$的平均值G(t)要比单一的$g_t$更接近目标函数f，regression效果更好。

G是数目为T的g_t的平均值。令包含N个数据的样本D独立同分布于P^N，每次从新的D_t中学习得到新的g_t，在对g_t求平均得到G，当做无限多次，即T趋向于无穷大的时候：

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 $\mathcal{A}(\mathcal{D}_t)$

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

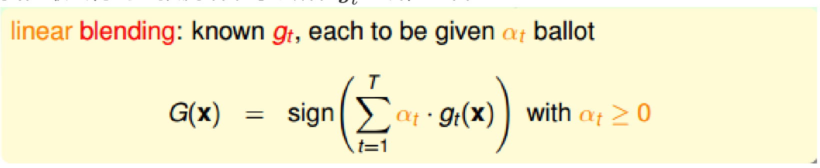
当T趋于无穷大的时候，$G = \bar{g}$，则有如下等式成立：

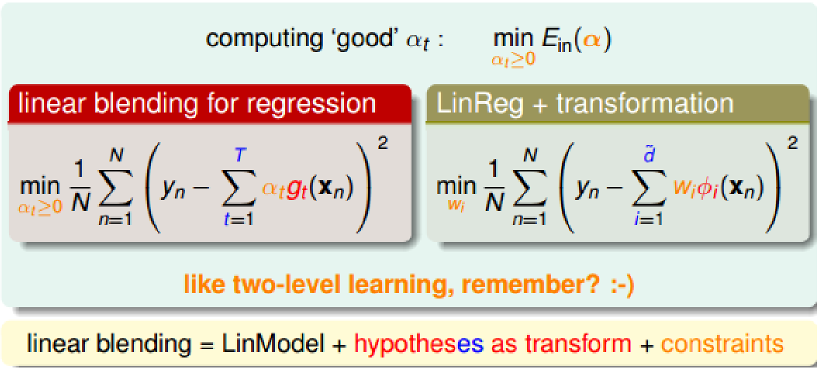


一个演算法的平均表现可以被拆成两项，一个是所有g_t的共识，一个是不同g_t之间的差距是多少，即bias和variance。
uniform blending的操作时求平均的过程，削减弱化了上式第一项variance的值，从而演算法的表现就更好了，能得到更加稳定的表现。

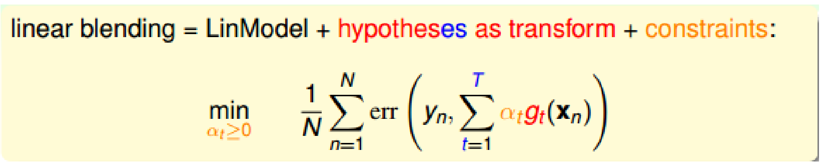
3.Linear and Any Blending

linear blending，每个g_t赋予的权重α_t不同，$\alpha_t \geq 0$。
我们最终得到的预测结果等于所有g_t的线性组合:



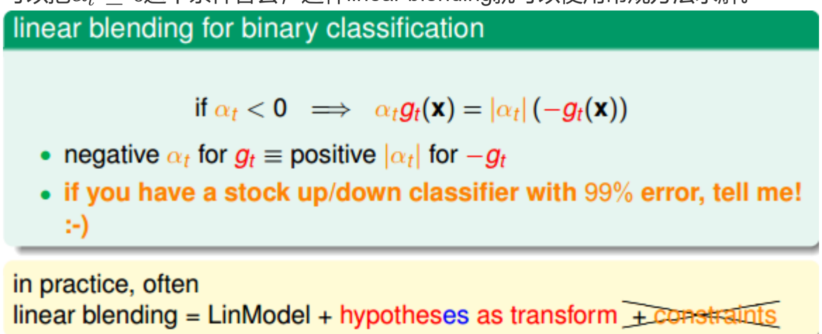


利用误差最小化的思想，找出最佳的α_t，使$E_{in}(\alpha)$取最小值。

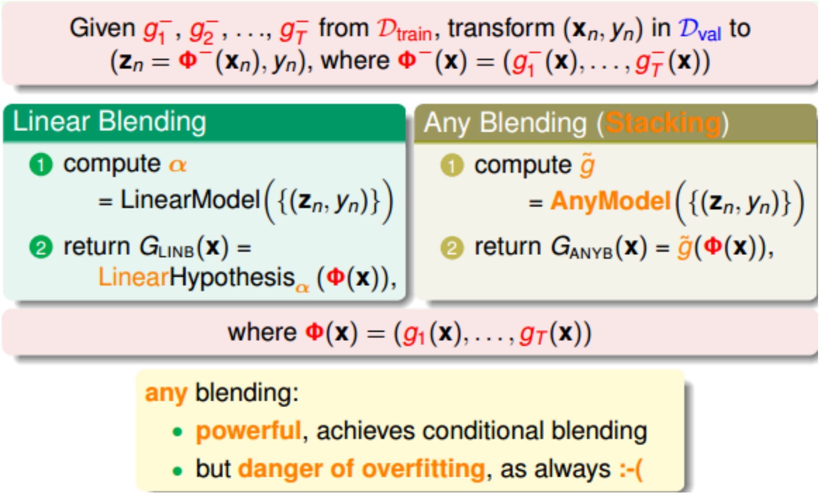


先计算$g_t(x_n)$，再进行linear regression得到α_t值。

$\alpha_t < 0$并不会影响分类效果，只需要将正类看成负类，负类当成正类即可。
可以把$\alpha_t \geq 0$这个条件舍去，这样linear blending就可以使用常规方法求解。



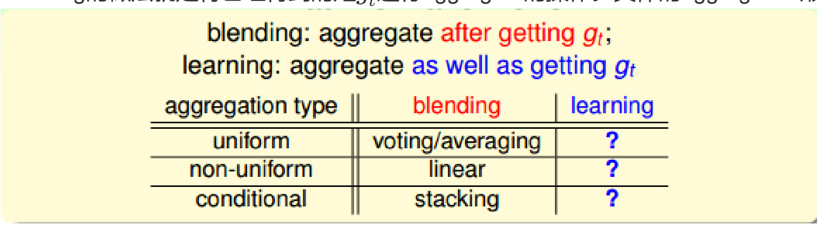
Linear Blending中使用的g_t是通过模型选择而得到的，利用validation，从D_{train}中得到$g_1^-, g_2^-, \cdots, g_T^-$。
将D_{train}中每个数据点经过各个矩的计算得到的值，代入到相应的linear blending计算公式中，迭代优化得到对应α值。
利用所有样本数据，得到新的g_t代替g_t^-，则G(t)就是g_t的线性组合而不是g_t^-，系数是α_t。



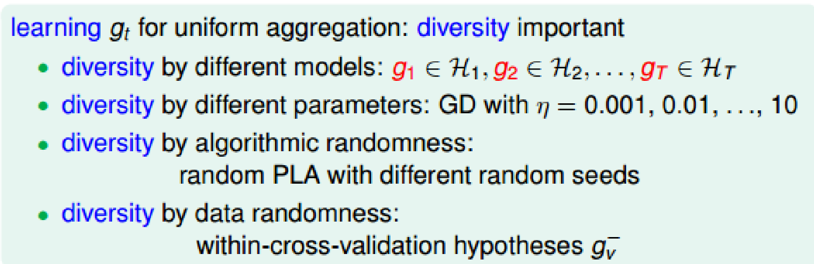
linear blending中，G(t)是g(t)的线性组合；any blending中，G(t)可以是g(t)的任何函数形式（非线性）,这种形式的blending也叫做Stacking。
any blending的优点是模型复杂度提高，更容易获得更好的预测模型；缺点是复杂模型也容易带来过拟合的危险。
在使用any blending的过程中要时刻注意避免过拟合发生，通过采用regularization的方法，让模型具有更好的泛化能力。

4.Bagging(Bootstrap Aggregation)

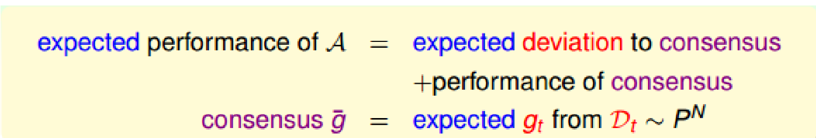
blending的做法就是将已经得到的矩g_t进行aggregate的操作。具体的aggregation形式包括：uniform，non-uniforn和conditional。


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	?

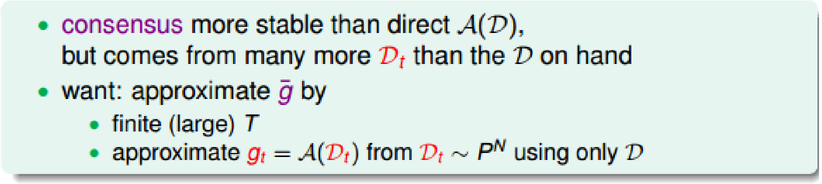


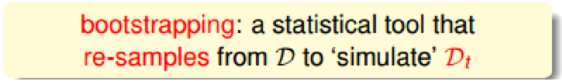
可以选取不同模型H；可以设置不同的参数，例如η、迭代次数n等；可以由算法的随机性得到，例如PLA、随机种子等；可以选择不同的数据样本等。这些方法都可能得到不同的g_t。



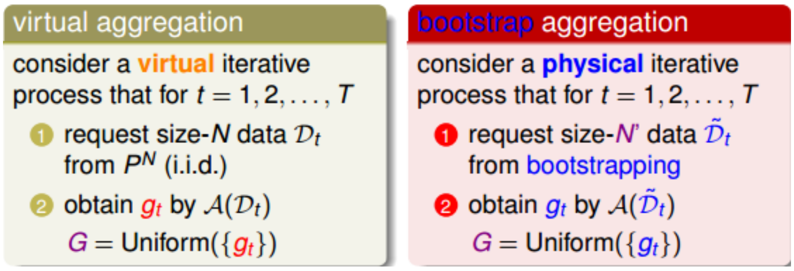
$\bar{g}$是在矩个数T趋向于无穷大的时候，不同的g_t计算平均得到的值。
这里我们为了得到$\bar{g}$，做两个近似条件：
1.有限的T；
2.由已有数据集D构造出D_t P^N，独立同分布

第一个条件没有问题，第二个近似条件的做法就是bootstrapping。
bootstrapping是统计学的一个工具，思想就是从已有数据集D中模拟出其他类似的样本D_t。

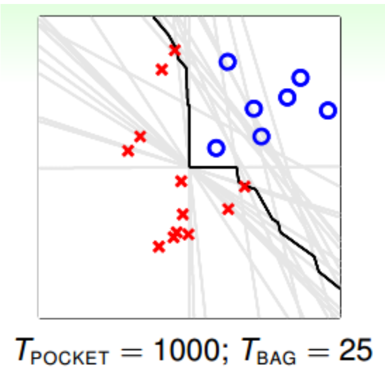




bootstrapping的做法是，假设有N笔资料，先从中选出一个样本，再放回去，再选择一个样本，再放回去，共重复N次。
这样我们就得到了一个新的N笔资料，这个新的$\breve{D}_t$中可能包含原D里的重复样本点，也可能没有原D里的某些样本，$\breve{D}_t$与D类似但又不完全相同。
抽取-放回的操作不一定非要是N，次数可以任意设定。
利用bootstrap进行aggragation的操作就被称为bagging。



eg:Bagging Pocket算法应用。
先通过bootstrapping得到25个不同样本集，再使用pocket算法得到25个不同的g_t，每个pocket算法迭代1000次。
再利用blending，将所有的g_t融合起来，得到最终的分类线。
虽然bootstrapping会得到差别很大的分类线（灰线），但是经过blending后，得到的分类线效果是不错的，则bagging通常能得到最佳的分类模型。



只有当演算法对数据样本分布比较敏感的情况下，才有比较好的表现。

5.summary

blending和bagging都属于aggregation，是将不同的g_t合并起来，利用集体的智慧得到更加优化的G(t)。
Blending通常分为三种情况：Uniform Blending，Linear Blending和Any Blending。
其中，uniform blending采样最简单的“一人一票”的方法，linear blending和any blending都采用标准的two-level learning方法，类似于特征转换的操作，来得到不同g_t的线性组合或非线性组合。
利用bagging（bootstrap aggregation），从已有数据集D中模拟出其他类似的样本D_t，而得到不同的g_t，再合并起来，优化预测模型。