# Lecture 15　Validation

整理者: LobbyBoy*　2020年2月28日

## 1. Model Selection Problem

至此，我们已经学习了许多Model(=Hypothesis set+Algorithm)。仅仅对二元分类问题，我们可以对下面的各项进行自由组合，得到一个完整的Model：

- $A \in$ PLA, Pocket, Linear regression, Logistic regression, ...
- $T \in$ 100, 1000, 10000, ...
- $\eta \in$ 1, 0.01, 0.0001, ...
- $\Phi \in$ Linear, Quadratic, Ploy-10, Legendre-poly-10, ...
- $\Omega(\vec{w}) \in$ L2, L1, Symmetry, ...

所以，我们不得不在这么多组合(combination)中进行选择(selection)。也就是说，我们现在有$M$个可用的models：

$$(H_1, A_1), (H_2, A_2), ..., (H_M, A_M)$$

我们的目标是，选到$(H_{m^*}, A_{m^*})$，使得：$g_{m^*} = A_{m^*}(D)$的$E_{out}(g_{m^*})$很小。然而，我们之前就说过，因为数据的分布未知，所以$E_{out}$无法计算，我们需要找到某个proxy进行判断。

可以用$E_{in}$来进行选择吗？即：

$$m^* = \underset{1 \le m \le M}{argmin} \Big(E_m = E_{in}(A_m(D))\Big)$$

不可以！因为：

- $\Phi_{1126}$的$E_{in}$一定比$\Phi_1$的$E_{in}$小，$\lambda = 0$永远比$\lambda = 0.1$的$E_{in}$小，但是明显会overfitting；
- 用$A_1$从$H_1$中挑到$g_1$，用$A_2$从$H_2$中挑到$g_2$，再比较$g_1$与$g_2$，看哪个$E_{in}$小。这个过程等价于从$H_1 \cup H_2$中挑$g^*$。也就是说，我们用的是一个$d_{VC}(H_1 \cup H_2)$更大的hypothesis set，所以会导致bad generalization。

*本笔记根据台湾大学林轩田教授于线上教育平台Coursera开设的“机器学习基石”课程整理而成(课程内容见: https://www.coursera.org/learn/ntumlone-mathematicalfoundations/home/welcome)。笔记内的大多数图片来自于林老师的课程slides。感谢林老师能够将如此精彩的课程通过线上平台同所有人分享，thanks!

可以通过$E_{test}$选择吗？即：

$$m^* = \underset{1\le m\le M}{argmin}\left(E_m = E_{test}(A_m(D))\right)$$

理论上可以，是因为有finite-bin Hoeffding的保证(因为用test set进行选择本质上是用了一份全新的资料)：

$$E_{out}(g_m^*) \le E_{test}(g_m^*) + O(\sqrt{\frac{logM}{N_{test}}})$$

但最大的问题是，你拿不到$D_{test}$，锁在老板的柜子里呢——Selecting by $E_{test}$ is infeasible and cheating。但我们可以从上面的叙述中得到一点点启示：我们需要用一组“干净的”数据！我们将其称为Validation Set，然后计算各个Model最优解的$E_{val}$，取具有最小$E_{val}$的最优解，即为“最最优解”。

## 2. Validation

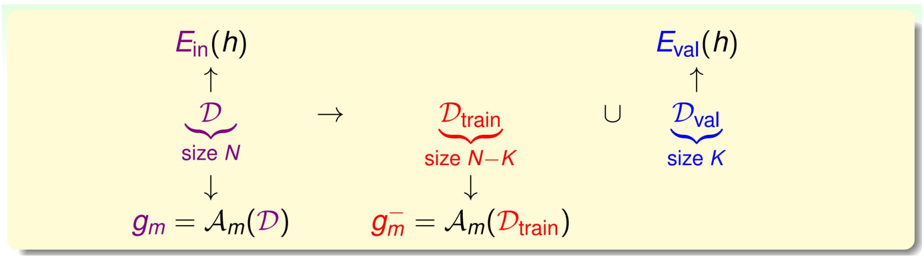


图 1: Sample = Training set+Validation set

我们将从原来的Training Set中拿出$K$个样本组成验证集，即Validation Set，剩下的Training Examples再组成训练集，记号如下：

- $D$：原训练集——$size = N$
- $D_{train}$：现训练集，数量变少——$size = N - K$
- $D_{val}$：验证集——$size = K$，这$K$个样本是random抽取的，即i.i.d

对于某个Model $m$，我们之前拿到的$g_m = A_m(D)$。但现在，我们拿到的是$g_m^- = A_m(D_{train})$——减号表示我们用的训练数据比原来的训练数据要小。再给所有的$g^-$喂进去$D_{val}$，拿到具有最小的$E_{val}$的$g_{m^*}^-$，则$m^*$即为最佳模型。这里最关键的一点是，有VC Bound能够为每一个$g_m^-$作保证：

$$E_{out}(g_m^-) \le E_{val}(g_m^-) + O\left(\sqrt{\frac{logM}{K}}\right)$$

完整的流程如下图所示。需要注意，拿到最好的Model后，我们需要再次训练一遍——这次训练我们不再用样本量少的$D_{train}$，而用整个数据集$D$：

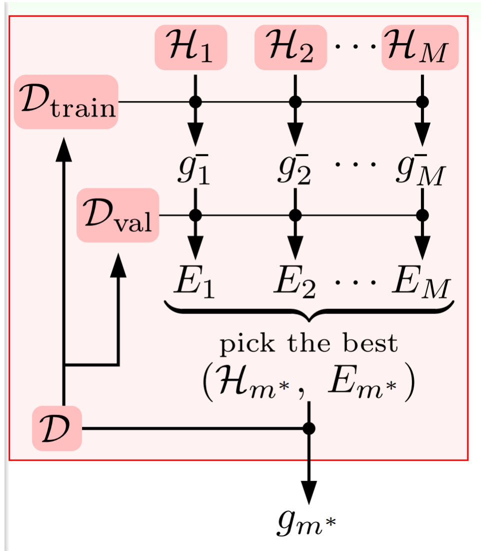


图 2: Model selection by validation set

为什么还要再用整个数据集训练一次呢？因为模型固定时，样本量越多，训练效果当然越好，这点很直觉，也可以从learning curve中看出来。即：

$$E_{out}\left(g_{m^*}\right) \le E_{out}\left(g^-_{m^*}\right)$$

结合之前的VC bound，有：

$$E_{out}\left(g_{m^*}\right) \le E_{out}(g^-_{m^*}) \le E_{val}(g^-_{m^*}) + O\left(\sqrt{\frac{logM}{K}}\right)$$

虽然上式为我们提供了某种程度上的保证，但≤毕竟和≈还是有所区别，这里会牵扯到验证集大小$K$的选择：

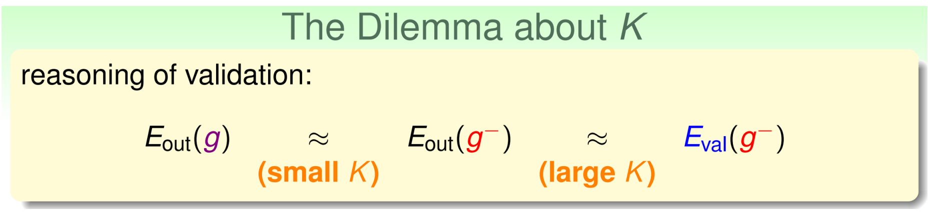


图 3: Small $K$ vs. Large $K$

如果$K$太大，那么根据validation error选到的$g^-$，其out-of-sample error也会是最小的；但是由于$K$太大了，$g^-$的out-of-sample error并不是$g$的out-of-sample error的一个紧的上界，因此最终回传的$g$的out-of-sample error不一定是所有模型中最好的。

如果$K$太小，$g^-$的out-of-sample error的确是$g$的out-of-sample error的一个非常好的反映；但是根据validation error选到的$g^-$，其out-of-sample error不一定是最好的。经验上：

$$K = \frac{N}{5}$$

## 3. Leave-One-Out Cross Validation

考虑一种极端情况：$K=1$，即只拿一个样本集中的样本作为validation set，其他的样本都用来训练。我们知道，当$K=1$时，$E_{out}(g)$和$E_{out}(g^-)$很接近，但$E_{out}(g^-)$和$E_{val}(g^-)$会有一定差距。下面我们引入一些记号：

- $D_{val}^{(n)} = \{(\mathbf{x_n}, y_n)\}$，表示被选到的那一个作为验证集的样本，序号记作$n$；
- $E_{val}^{(n)}(g_n^-) = err\Big(g_n^-(\mathbf{x_n}, y_n)\Big) = e_n$，表示某模型的验证误差，即用除$n$号样本外的样本训练出的$g_n^-$，在$n$号样本上的误差。

对于某个模型，单一一个$e_n$当然会与$E_{out}(g)$有差距，但是如果把每个样本都当作验证集，计算$N$次$e_n$并取平均，那么可能得到$\overline{e_n} \approx E_{out}(g)$。我们把$N$次平均的验证误差记作：

$$E_{loocv}(H, A) = \frac{1}{N}\sum_{n=1}^{N} e_n = \frac{1}{N}\sum_{n=1}^{N} err(g_n^-(\mathbf{x_n}), y_n)$$

我们希望：

$$E_{loocv}(H, A) \approx E_{out}(g)$$

证明如下：

$$\begin{aligned}
\boxed{\mathop{\mathcal{E}}_{\mathcal{D}} E_{\text{loocv}}(\mathcal{H}, \mathcal{A})} = \mathop{\mathcal{E}}_{\mathcal{D}} \frac{1}{N}\sum_{n=1}^{N} e_n \quad &= \frac{1}{N}\sum_{n=1}^{N} \mathop{\mathcal{E}}_{\mathcal{D}} e_n \\
&= \frac{1}{N}\sum_{n=1}^{N} \mathop{\mathcal{E}}_{\mathcal{D}_n} \mathop{\mathcal{E}}_{(\mathbf{x}_n, y_n)} \text{err}(g_n^-(\mathbf{x}_n), y_n) \\
&= \frac{1}{N}\sum_{n=1}^{N} \mathop{\mathcal{E}}_{\mathcal{D}_n} E_{\text{out}}(g_n^-) \\
&= \frac{1}{N}\sum_{n=1}^{N} \overline{E_{\text{out}}}(N-1) = \boxed{\overline{E_{\text{out}}}(N-1)}
\end{aligned}$$

图 4: 留一误差的期望等于$E_{out}(g^-)$的期望

上述推导说明，$E_{loocv}(H, A)$是$E_{out}(g^-)$的很好的近似。而$E_{out}(g^-)$又很近似于$E_{out}(g)$(因为$g^-$是用只去掉一个样本的训练数据训练得到的)，因此$E_{loocv}(H, A)$是$E_{out}(g)$的很好反映，

那么我们就可以安全地根据$E_{loocv}(H,A)$来选择model了。

## 4. V-Fold Cross Validation

留一交叉验证的缺点有两个。第一，Computation：对每一个Model，带来$N$个“额外”的训练过程，不太可行，费时费力；第二，Stability：单个点的方差（波动）较大——例如二元分类，单个点的err可能是0或1，即使我们取多次值的平均，也可能造成较大的波动。

思考LOOCV的实质。LOOCV是将资料切成了$N$份，然后每次取$N-1$份训练，将训练结果在剩下的1份上做验证，一共这样做$N$次。那我们可不可以不切成$N$份？比如只随机切成$V$份，这样，我们拿$V-1$份做训练，剩下1份做验证，这样只用做$V$次。例如：我们将数据集切成10份：

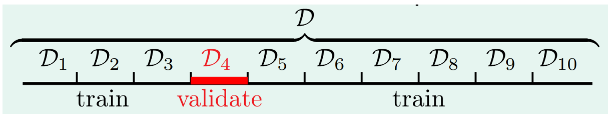


图 5: Cross Validation

这样的验证误差我们称为$E_{cv}$：

$$E_{cv}(H,A)=\frac{1}{V}\sum_{v=1}^{V}E_{val}^{(v)}(g_v^-)$$

此后，我们对不同的Model计算他们的$E_{cv}$，取到最小的$E_{cv}$对应的Model：

$$m^*=\underset{1\le m\le M}{argmin}\Big(E_m=E_{cv}(H_m,A_m)\Big)$$

经验法则：

$$V=10$$

最后，关于Validation的总结：

- 通常，交叉验证(Cross Validation)比单一验证(Single Validation)要好；
- 通常5-Fold或10-Fold就足够了，不需要LOOCV。

关于训练(Training)、验证(Validation)、测试(Testing)：

- 训练：在大大的Hypothesis Set中做选择；
- 验证：“复赛”做选择；
- 测试：并不是在选择，只是在评估。