Note for cs229-note4-5

iSea @ Jan. 6th, 2015

Content

---

5 统计学习理论

（由于这一章过于理论抽象，所以一切从简。）

5.1 学习目标

统计学习是使泛化误差（Generalization Error）最小，而并非直接地使training set上的误差最小。不好的模型要么Overfit，要么Underfit，或者用bias和variance来表示。

通常，用ERM（Empirical Risk Minimization，经验风险最小化）来作为学习目标，也就是选取最优的参数，使训练误差$\hat{\varepsilon}(h_\theta)$最小，也就是

$$\hat{\theta} = \arg\min_{\theta} \hat{\varepsilon}(h_\theta)$$

虽然$\hat{\varepsilon}(h_\theta) \neq \varepsilon(h_\theta)$，但是ERM准则仍然认为经验误差可以收敛到泛化误差。更广义的，定义一个假设空间$\mathcal{H}$，统计学习的目标是找到假设

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \hat{\varepsilon}(h)$$

5.2 有限假设空间

根据Hoeffding不等式，可以证明

$$\varepsilon(\hat{h}) \leq (\min_{h \in \mathcal{H}} \varepsilon(h)) + 2\sqrt{\frac{1}{2m} \log \frac{2|\mathcal{H}|}{\delta}}$$

有$1-\delta$的概率成立。因此在$\mathcal{H}$有限的情况下，经验误差可以接近泛化误差。

5.3 无限假设空间

当$\mathcal{H}$无限时，上面的式子就不能证明两个误差接近了。

事实上，在计算机中，不存在数学意义上的“无限”，即使一个模型的参数是d个实数，在计算机中表示也是$k \approx 64$位bit，也就是$|\mathcal{H}| = 2^{kd}$，还是有限的。

但如果希望假设足够好，好到两个误差无限接近，解空间自然应该是无限的。

为了解决这个问题，引入VC维数（Vapnik-Chervonenkis dimension）。VC维数是用来权衡分类算法的复杂度的。如果至少存在一个大小为d的集合，可以被$\mathcal{H}$分散，称$\mathcal{H}$的VC维数为d，如果任意集合都可以分散，那么$\mathrm{VC}(\mathcal{H})=\infty$。$k$维度的线性分类器的$\mathrm{VC}(\mathcal{H})=k+1$。可以证明

$$\varepsilon(\hat{h}) \leq \varepsilon(h^*)+O(\sqrt{\frac{d}{m}\log\frac{m}{d}+\frac{1}{m}\log\frac{1}{\delta}})$$

有$1-\delta$的概率成立，其中$d=\mathrm{VC}(\mathcal{H})$。也就是说，如果一个假设的VC维度是有限的，经验误差可以接近泛化误差。而大多数假设的VC维度是和它的参数个数成正比的，也就是说，对于一个学习模型，training set的个数应该与模型参数的个数成正比。

6 规则化和模型选择

6.1 交叉验证

对于学习问题，很多时候有很多模型可以选择，记为$\mathcal{M}=\{M_1,M_2,\dots,M_d\}$，我们需要选择其一，然后进行参数学习。通常，简单的**交叉验证**（Cross Validation）可以用来解决这个问题：

1. 随机将training set S划分为S_{train}和S_{cv}（比如7:3）
2. 在S_{train}上训练每个模型M_i，得到一个假设函数h_i
3. 选取在S_{cv}上有最小泛化误差$\hat{\varepsilon}_{S_{\text{cv}}}(h_i)$的$M_i$作为结果

通常，为了避免S_{cv}的“浪费”，会在全部training set上再训练一次模型。而当训练数据很有限时，或者希望更加充分利用数据时，有一个更好的**k折叠交叉验证**（k-fold Cross Validation)：

1. 随机将S划分为k个不相交集合，记做$S_1,\dots,S_k$
2. 对每个M_i重复训练k次，每次训练集合去掉S_j，测试集合选取S_j，计算出平均的$\hat{\varepsilon}_{S_j}(h_{ij})$
3. 选取最小平均泛化误差的模型，并在全部训练集合上重新训练一次

6.2 特征选取

特征选取是模型选择中重要的一步，通常贪心地前向搜索：

1. 初始化 $\mathcal{F}=\emptyset$
2. 重复直到 $\mathcal{F}=\{1,2,\dots,n\}$
    1. $\mathcal{F}_i=\mathcal{F}\cup i \text{ if } i\notin\mathcal{F}$，交叉验证选取$\mathcal{F}_i$
    2. $\text{Set } \mathcal{F}=\mathcal{F}_i$
3. 输出表现最好的feature子集

需要训练模型的次数是$O(n^2)$的。类似的贪心方法有后向搜索。

过滤特征选择也是一个有效的方法：

1. 依据每个特征的信息量进行排序，计算方法为

$$MI(x_i,y)=\sum_{x_i}\sum_{y}p(x_i,y)\log\frac{p(x_i,y)}{p(x_i)p(y)}$$

2. $\mathcal{F}_i = \{x_{sorted_1},...,sorted_i\}$，交叉验证选取$\mathcal{F}_i$

由于进行了事先排序，训练次数减小到了$O(n)$。

6.3 贝叶斯统计和规则化

前面的章节中多次使用了极大似然估计，认为θ是未知的常量，即

$$\theta_{ML} = \arg\max_{\theta} \prod_{i=1}^{M} p(y^{(i)}|x^{(i)};\theta)$$

注意这里的θ是作为参数的。而贝叶斯统计的视角认为θ也是一个随机变量，具有自己的概率分布，根据贝叶斯定理

$$\begin{aligned} p(\theta|S) &= \frac{p(S|\theta)p(\theta)}{p(S)} \\ &= \frac{(\prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta))p(\theta)}{\int_{\theta}(\prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta))p(\theta)d\theta} \end{aligned}$$

同时可以计算出

$$p(y|x,S) = \int_{\theta} p(y|x,\theta)p(\theta|S)d\theta$$

将θ后验概率分布近似成单点估计，综合两个式子，可以得到**最大后验概率估计（MAP）**：

$$\theta_{MAP} = \arg\max_{\theta} \prod_{i=1}^{M} p(y^{(i)}|x^{(i)},\theta)p(\theta)$$

通常可以选择$\theta \sim \mathcal{N}(0,\tau^2 I)$，并且满足MAP估计的模型比ML估计更好的解决了过overfit的情形。比如，贝叶斯逻辑回归用于文本分类时，即使$n \gg m$，也有很不错的性能。