ML note

mstyoda 骆轩源

Contents

1 Rademacher Complexity and VC-Dimension

1.1 Rademacher complexity

Rademacher Complexity是用来衡量一个函数族$G : X \to \mathbb{R}$的复杂程度的指标，它考验的是G对于随机噪声的拟合能力。比如给定样本$S = (z_1, z_2 ... z_m)$，其中$z_i = (x_i, y_i)$，那么先随机生成一个序列$\sigma = (\sigma_1, \sigma_2 ... \sigma_m)$，然后在G中找到一个函数$g$，使得$\mathbf{g}(S) \cdot \sigma$最大，把这个值对于$\sigma$求期望，就得到了Empirical Rademacher complexity。

Definition 1. 给定函数族G，其中的函数将Z映射到实数区间$[a, b]$，并且给定一个大小为m的S，σ_i独立**均匀**分布在$\{-1, 1\}$内，则定义G对于S的*Empirical Rademacher complexity* 为：

$$\hat{\mathcal{R}}_S(G) = E_\sigma[\sup_{g \in G} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i)] \tag{1}$$

很自然地，如果我们不固定S，只固定m，让$z_1, z_2 ... z_m$独立同分布于D，则可以定义G对于样本大小为m的Rademacher complexity为：

Definition 2. 给定函数族G，其中的函数将Z映射到实数区间$[a, b]$，并且给定m，z_i独立同分布D，σ_i独立均匀分布在$\{-1, 1\}$内，则定义G 对于S 的*Rademacher complexity* 为：

$$\begin{aligned} \mathcal{R}_m(G) &= E_{S,\sigma}[\sup_{g \in G} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i)] \\ &= E_{S \sim D^m}[\hat{\mathcal{R}}_S(G)] \end{aligned} \tag{2}$$

那么我们研究函数族的复杂性有什么作用呢？它能够提供如下一个bound:

Theorem 1. 假如G是一个函数族，其中的函数是从Z到$[0, 1]$区间的映射，那么对于任意$\delta > 0$，至少有$1 - \delta$，使得对于任意$g \in G$满足：

$$E[g(z)] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_i) + 2\mathcal{R}_m(G) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}} \tag{3}$$

$$and \quad E[g(z)] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_i) + 2\hat{\mathcal{R}}_m(G) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}} \tag{4}$$

在证明之前，我们先来看看这个定理想要表达的意思，根据PAC 那一章节的定义，Generalization error和Empirical error分别为：

$$R(h) = \Pr_{x\sim D}[h(x) \neq c(x)] = E_{x\sim D}[1_{h(x)\neq c(x)}] \tag{5}$$

$$\hat{R}(h) = \frac{1}{m}\sum_{i=1}^{m} 1_{h(x_i)\neq c(x_i)} \tag{6}$$

$R(h)$说的假设h的错误率，$\hat{R}(h)$说的是假设h在这m 个测试样本上的错误率。那么回到我们刚才的定理，给定一个假设空间H（是一个函数族），我们可以将其转换到令一个函数族G，对于任意$g \in G$，其对应于某一个$h \in H$，有$g(z) = g(x, y) = L(h(x), y)$，$L$是损失函数loss function。在这里我们认为L为0-1 loss，也即:

$$L(y', y) = \begin{cases} 1 & y' \neq y \\ 0 & y' = y \end{cases} \tag{7}$$

那么定理中左边的$E[g(z)]$就对应于$R(h)$，右边的$\frac{1}{m}\sum_{i=1}^{m} g(z_i)$对应于$\hat{R}(h)$。所以定理说的其实是：

$$R(h) \le \hat{R}(h) + 2\mathcal{R}_m(G) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

由于$R(h)$是很难说明的，但是$\hat{R}(h)$是可以实验得到的，该bound就能利用实验得出的结果来估算$R(h)$的一个上界。注意到G仅仅跟L和H有关，所以我们将$2\mathcal{R}_m(G)$写成H 的形式：(中间的推导基于σ_i是均匀分布在$\{-1, 1\}$的随机变量，所以$E[\sigma_i] = 0$)

$$\begin{aligned}
\hat{\mathcal{R}}_S(G) &= E_\sigma[\sup_{g\in G}\frac{1}{m}\sum_{i=1}^{m} g(z_i)\sigma_i] \\
&= E_\sigma[\sup_{g\in G}\frac{1}{m}\sum_{i=1}^{m}\frac{1-h(x_i)y_i}{2}\sigma_i] \\
&= E_\sigma[\sup_{g\in G}\frac{1}{m}\sum_{i=1}^{m}\frac{-h(x_i)y_i}{2}\sigma_i] \\
&= \frac{1}{2}E_\sigma[\sup_{g\in G}\frac{1}{m}\sum_{i=1}^{m} h(x_i)\sigma_i] = \frac{1}{2}\hat{\mathcal{R}}_{SX}(H)
\end{aligned} \tag{8}$$

最后一步推导是因为，当y_i只能是1或-1，所以$-y_i\sigma_i$也是在$\{-1, 1\}$之间的均匀分布，和σ_i同分布，所以可以替换。由如上结论可以得到：

$$\mathcal{R}_m(G) = E_{S\sim D^m}[\hat{\mathcal{R}}_S(G)] = \frac{1}{2}\mathcal{R}_m(H) \tag{9}$$

所以到这一步，之前的定理可以转化为：

$$R(h) \leq \hat{R}(h) + \mathcal{R}_m(H) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}} \tag{10}$$

$$\text{and} \quad R(h) \leq \hat{R}(h) + \hat{\mathcal{R}}_S(H) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}} \tag{11}$$

下面给出Theory1的证明：

Proof. 令$\Phi(S) = \sup_{g\in G} E[g(z)] - \frac{1}{m}\sum_{i=1}^{m} g(z_i)$，它描述的是，相减的两个东西的差距的上界，也即求出$\Phi(S)$就可以完成证明，则对于任意S'，如果S' 与S 仅有1 个z_k 不同，那么有:

$$\Phi(S) - \Phi(S') = \sup_{g\in G} \frac{1}{m} g(z'_k) - g(z_k) \leq \frac{1}{m} \tag{12}$$

由对称性可以知道，$\Phi(S') - \Phi(S) \leq \frac{1}{m}$，这时候使用McDiarmid’s inequality(之后证明)，可以得到，对于任意$\delta > 0$，至少有$1 - \delta/2$ 的概率满足：

$$\Phi(S) \leq E_S[\Phi(S)] + \sqrt{\frac{\log \frac{2}{\delta}}{2m}} \tag{13}$$

如果$E_S[\Phi(S)]$是一个比较稳定的值，那么该不等式也即说明了随着样本数目m的增加，二者的差距在$1 - \delta/2$的信心下的误差上界以根号的速度减小，接下来研究$E_S[\Phi(S)]$的上界:

$$\begin{aligned} E_S[\Phi(S)] &= E_S[\sup_{g\in G} E[g(z)] - \frac{1}{m}\sum_{i=1}^{m} g(z_i)] \\ &= E_S[\sup_{g\in G} E_{S'}[\hat{E}_{S'}[g(z)]] - \hat{E}_S[g(z)]] \\ &= E_S[\sup_{g\in G} E_{S'}[\hat{E}_{S'}[g(z)] - \hat{E}_S[g(z)]]] \\ &\leq E_{S,S'}[\sup_{g\in G}(\hat{E}_{S'}[g(z)] - \hat{E}_S[g(z)])] \end{aligned} \tag{14}$$

最后一步用到了$\sup_x E_y[f(x,y)] \leq E_y[\sup_x f(x,y)]$。

考虑到σ_i等概率取自1或-1，由于S和S'均为随机变量，当σ_i给定时，根据对称性有$E_{S,S'}\sigma_i(g(z_i) - g(z'_i)) = E_{S,S'}(g(z_i) - g(z'_i))$，那么也就有：

$$\begin{aligned} \text{上式} &= E_{\sigma,S,S'}[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m} \sigma_i(g(z_i) - g(z'_i))] \\ &= E_{\sigma,S}[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m} \sigma_i(g(z_i))] + E_{\sigma,S'}[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m} \sigma_i(g(z'_i))] \\ &= 2\mathcal{R}_m(G) \end{aligned} \tag{15}$$

综上我们有，至少有$1-\delta/2$的信心满足：

$$\begin{aligned}\Phi(S) &\leq E_S[\Phi(S)]+\sqrt{\frac{\log\frac{2}{\delta}}{2m}}\\ &\leq 2\mathcal{R}_m(G)+\sqrt{\frac{\log\frac{2}{\delta}}{2m}}\end{aligned} \tag{16}$$

又因为对于任意两个仅相差一个元素的S'和S，$\hat{\mathcal{R}}_S(G)-\hat{\mathcal{R}}_{S'}(G)\leq\frac{1}{m}$，再一次使用McDiarmid.s inequality，可以知道至少有$1-\delta/2$的信心满足：

$$E_S[\hat{\mathcal{R}}_S(G)]\leq\hat{\mathcal{R}}_S(G)+\sqrt{\frac{\log\frac{2}{\delta}}{2m}} \tag{17}$$

也即：

$$\mathcal{R}_m(G)\leq\hat{\mathcal{R}}_S(G)+\sqrt{\frac{\log\frac{2}{\delta}}{2m}} \tag{18}$$

使用union bound合并式上两个式子，可以至少有$1-\delta$的概率满足：

$$\Phi(S)\leq 2\hat{\mathcal{R}}_S(G)+3\sqrt{\frac{\log\frac{2}{\delta}}{2m}} \tag{19}$$

到此为止，该定理的两个不等式都得到了证明。 □

由于$\mathcal{R}_m(H)$的计算涉及到随机变量和上确界，其计算非常困难，接下来我们引入其一个上界growth function，它与随机变量无关，可计算性更强。

1.2 Growth function

首先引入growth function的定义，

Definition 3. 给定假设空间H，和正整数m，定义*growth function* $\Pi_H:\mathbb{N}\rightarrow\mathbb{N}$为：

$$\Pi_H(m)=\max_{(x_1,x_2...x_m)\in\mathcal{X}}|\{(h(x_1),h(x_2),...,h(x_m))|h\in H\}| \tag{20}$$

该函数定义的也是H的一个复杂性，给定m个点，用H中的函数去映射，能产生不超过$|H|$种结果，找到m个点，使得该结果数最多，此时的结果数就为$\Pi_H(m)$。

接下来引入Massart’s lemma，它为Rademacher complexity和growth function搭了一个重要的桥梁：

Theorem 2. 设有限集合$A \subseteq R^m$，令$r = \max_{x \in A} \|x\|_2$，则有：

$$E_\sigma[\frac{1}{m} \sup_{x \in A} \sum_{i=1}^{m} \sigma_i x_i] \leq \frac{r\sqrt{2 \log |A|}}{m} \tag{21}$$

其中σ_i独立均匀取自$\{-1, 1\}$。

Proof. 定义随机变量$Y = \sup_{x \in A} \sum_{i=1}^{m} \sigma_i x_i$，函数$f(y) = \exp(t \cdot y), t > 0$，所以有：

$$f(E[Y]) \leq E[f(y)] \tag{22}$$

这是因为对于任意$\alpha_1, \alpha_2, ..., \alpha_n > 0$且$\sum_i \alpha_i = 1$时，有：(下凸函数性质)

$$f(\alpha_1 y_1 + \alpha_2 y_2 + ... + \alpha_n y_n) \leq \alpha_1 f(y_1) + \alpha_2 f(y_2) + ... + \alpha_n f(y_n) \tag{23}$$

又因为，Y的取值最多只有2^m种，故令$n = 2^m$，$y_1 ... y_n$分别对应每一种取值，则有：

$$f(E[Y]) = f(\sum_{i=1}^{n} y_i \Pr[Y = y_i]) \leq \sum_{i=1}^{n} \Pr[Y = y_i] f(y_i) = E[f(y)] \tag{24}$$

所以有，

$$\begin{aligned}
\exp(t \cdot E[\sup_{x \in A} \sum_{i=1}^{m} \sigma_i x_i]) &\leq E[\exp(t \cdot \sup_{x \in A} \sum_{i=1}^{m} \sigma_i x_i)] \\
&= E[\sup_{x \in A} \exp(\sum_{i=1}^{m} t \cdot \sigma_i x_i)] \\
&\leq \sum_{x \in A} E[\exp(\sum_{i=1}^{m} t \cdot \sigma_i x_i)] \\
&= \sum_{x \in A} \prod_{i=1}^{m} E[\exp(t \sigma_i x_i)]
\end{aligned} \tag{25}$$

上述推导用了求和来放缩sup，并且σ_i之间相互独立，继续放缩右边的式子:

$$E[\exp(t \sigma_i x_i)] \leq \exp(\frac{t^2 (2r)^2}{8}) \tag{26}$$

这是因为$\sigma_i x_i \in [-x_i, x_i]$，令$a = -x_i, b = x_i$，由凸函数性质有：

$$\exp(t \sigma_i x_i) \leq \frac{b - \sigma_i x_i}{b - a} \exp(ta) + \frac{\sigma_i x_i - a}{b - a} \exp(tb) \tag{27}$$

两边取期望得到：

$$E[e^{t \sigma_i x_i}] \leq e^{ta} E[\frac{b - \sigma_i x_i}{b - a}] + e^{tb} E[\frac{\sigma_i x_i - a}{b - a}] \tag{28}$$

由于$E[\sigma_i x_i]=0$ 所以上式子可以写成：

$$\begin{aligned} E[e^{t\sigma_i x_i}] &\leq e^{ta}\frac{b}{b-a}+e^{tb}\frac{-a}{b-a} \\ &\leq e^{\phi(t)} \end{aligned} \tag{29}$$

$$\begin{aligned} \phi(t) &= \log(e^{ta}\frac{b}{b-a}+e^{tb}\frac{-a}{b-a}) \\ &= ta+\log(\frac{b}{b-a}+e^{tb-ta}\frac{-a}{b-a}) \\ &= \phi(0)+t\phi'(0)+\frac{t^2}{2}\phi''(\theta) \\ &= t^2\frac{(b-a)^2}{8} \end{aligned} \tag{30}$$

最后一步是暴力泰勒展开得到的，所以得到：

$$E[e^{t\sigma_i x_i}] \leq \exp(t^2\frac{(b-a)^2}{8}) = \exp(t^2\frac{x_i^2}{2}) \tag{31}$$

$$\begin{aligned} \exp(t\cdot E[\sup_{x\in A}\sum_{i=1}^m \sigma_i x_i]) &\leq \sum_{x\in A}\exp(t^2r^2/2) \\ &\leq |A|\exp(t^2r^2/2) \end{aligned} \tag{32}$$

$$(t\cdot E[\sup_{x\in A}\sum_{i=1}^m \sigma_i x_i]) \leq (\frac{t^2r^2}{2})+\log|A| \tag{33}$$

$$E[\sup_{x\in A}\sum_{i=1}^m \sigma_i x_i] \leq (\frac{tr^2}{2})+\frac{\log|A|}{t} \tag{34}$$

取$t=\sqrt{\frac{2\log|A|}{r^2}}$可以得到右边最小值为$\sqrt{2\log|A|r^2}$，此时得到：

$$E[\sup_{x\in A}\sum_{i=1}^m \sigma_i x_i] \leq \sqrt{2\log|A|r^2} \tag{35}$$

$$\frac{1}{m}E[\sup_{x\in A}\sum_{i=1}^m \sigma_i x_i] \leq \frac{r\sqrt{2\log|A|}}{m} \tag{36}$$

定理得证。

□

我们尝试把$\mathcal{R}_m(H)$和$\Pi_H(m)$建立联系，首先假设H中的函数h将点映射到$\{-1,1\}$，则：

$$\mathcal{R}_m(H) = E_{S\sim D^m}[E_\sigma[\sup_{h\in H}\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(z_i)]] \tag{37}$$

使用Theorem2，把H_S看成A可以得到：

$$\begin{aligned}\mathcal{R}_m(H) &\leq E_{S\sim D^m}[\frac{r\sqrt{2\log|H_S|}}{m}] \\ &\leq \frac{r\sqrt{2\log\Pi_H(m)}}{m}\end{aligned} \tag{38}$$

其中当S给定时$r = \max_{h\in H}\{\|(h(x_1), h(x_2), ..., h(x_m))\|_2\}$，我们假设$H$中的函数映射到$\{-1,1\}$，那么有$r \leq \sqrt{m}$对任意$S \sim D^m$成立。

所以在这种假定下，上式可以写成：

$$\mathcal{R}_m(H) \leq \sqrt{\frac{2\log\Pi_H(m)}{m}} \tag{39}$$

所以前一小节的bound可以被写为：

$$\begin{aligned}R(h) &\leq \hat{R}(h) + \mathcal{R}_m(H) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}} \\ &\leq \hat{R}(h) + \sqrt{\frac{2\log\Pi_H(m)}{m}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}\end{aligned} \tag{40}$$

1.3 VC-dimension

前面提到的growth function虽然说不依赖于随机变量，但是计算仍然相当困难，接下来引入另外一个用来衡量假设空间H的复杂性的指标，VC-dimension。它的定义如下：

Definition 4. 一个假设空间H的*VC-dimension*被定义为，最大的可能被H**完全打散**的数据的大小，也即：

$$VCdim(H) = \max\{m : \Pi_H(m) = 2^m\} \tag{41}$$

它的定义蕴含了两个意思：

1. 对于任意$m \leq VCdim(H)$，存在一个$S = (x_1, x_2, ..., x_m)$，使得$|H_{|S}| = 2^m$。
2. 对于任意$m > VCdim(H)$，不存在$S = (x_1, x_2, ..., x_m)$，使得$|H_{|S}| = 2^m$。

下面我们将$VCdim(H)$和$\Pi_H(m)$建立联系，这样我们就可以将之前的bound用$VCdim(H)$来表示。

首先引入一个定理：

Theorem 3. 设假设空间H的$VCdim(H)=d$，那么对于任意$m \in \mathbb{N}$，有如下不等式成立：

$$\Pi_H(m) \leq \sum_{i=0}^{d} \binom{m}{i} \tag{42}$$

Proof. 对$m+d$的大小归纳，这么归纳的目的是为了用如下性质：

$$\binom{m}{i} = \binom{m-1}{i} + \binom{m-1}{i-1} \tag{43}$$

利用该性质我们可以得到：

$$\sum_{i=0}^{d} \binom{m-1}{i} + \sum_{i=0}^{d-1} \binom{m-1}{i} = \sum_{i=0}^{d} \binom{m}{i} \tag{44}$$

接下来我们按照两重循环求组合数的顺序(先枚举m，再枚举d) 来归纳证明：

1. 基础：$m=1$时，$d=0$，$d=1$都有结论成立。
2. 归纳：$m \geq 2$时，若$d=0$则结论成立，否则令S为满足$|H_{|S}| = \Pi_H(m)$ 的一个Sample。令G表示将H约束到S上的函数集合(将H定义域改成S)，有$|G| = \Pi_H(m)$。
   下面我们将G分割成两个假设空间G_1和G_2，使得$VCdim(G_1) \leq d$，$VCdim(G_2) \leq d-1$，且有$|G_1| \leq \Pi_{G_1}(m-1)$，$|G_2| \leq \Pi_{G_2}(m-1)$，就能完成归纳：
   先约定$H : \mathcal{X} \to \{0,1\}$，$S' = (x_1, x_2, ..., x_{m-1})$，
   那么令G_1为将H约束到S'上的函数集合，也就是在G中只看前$m-1$ 个点的分类结果来去重。
   我们在G中找到两个函数g_1,g_2，使得它们约束到S'上都是一样的，那么有它们对x_m的分类就一个是0，一个是1。我们把分类是0的那个函数扔到G_2'里。
   我们将G中的函数，按照其在S'的取值作为key，G中每个key恰好在G_1中出现一次，G 中每个出现2次的key都恰好在G_2'出现一次。
   所以有$|G_1| + |G_2'| = |G|$，再让G_2为将G_2'约束到S'的结果，一定有$|G_2| = |G_2'|$，故替换后有$|G_1| + |G_2| = |G|$。

由于G_1, G_2，考虑到G_1，G_2大小都等于自己作用在S'上的大小，由growth function定义有：

$$|G_1| \leq \Pi_{G_1}(m-1) \tag{45}$$

$$|G_2| \leq \Pi_{G_2}(m-1) \tag{46}$$

由于$G_1 \subseteq H$，肯定有$VCdim(G_1) \leq VCdim(H) = d$，又因为$G_2$的定义域是$S'$，我们假设其VCdim为$k$，那么$G_2$的极限也就是能把$S_k \subseteq S'$中的元素全部打散，往$S_k$中加入$x_m$，这时$H$就可以打散，但是$G_2$不可以打散。所以有$VCdim(G_2) \leq VCdim(H) - 1 = d - 1$。
所以有:

$$|G_1| \leq \Pi_{G_1}(m-1) \leq \sum_{i=0}^{d} \binom{m-1}{i} \tag{47}$$

$$|G_2| \leq \Pi_{G_2}(m-1) \leq \sum_{i=0}^{d-1} \binom{m-1}{i} \tag{48}$$

$$\Pi_H(m) = |G| = |G_1| + |G_2| \leq \sum_{i=0}^{d} \binom{m}{i} \tag{49}$$

□

然后我们把组合数求和变成一个容易求的上界，如果$(m \geq d)$，

$$\begin{aligned}
\Pi_H(m) &\leq \sum_{i=0}^{d} \binom{m}{i} \\
&\leq \sum_{i=0}^{d} \binom{m}{i} (\frac{m}{d})^{d-i} \\
&\leq \sum_{i=0}^{m} \binom{m}{i} (\frac{m}{d})^{d-i} \\
&= (\frac{m}{d})^d \sum_{i=0}^{m} \binom{m}{i} (\frac{d}{m})^{i} \\
&= (\frac{m}{d})^d (1 + \frac{d}{m})^m \\
&= (\frac{m}{d})^d (1 + \frac{d}{m})^{m/d*d} \\
&\leq (\frac{em}{d})^d
\end{aligned} \tag{50}$$

推导中比较巧妙的地方在于，凭空加入$(\frac{m}{d})^{d-i}$这一项，凑出来一个二项式求和，反向使用二项式定理，然后用自然对数e的展开就很自然了。

到了这一步，我们可以将之前的bound改成，如果$m \geq d$，对于某VCdim为d的假设空间H，有$1-\delta$的信心满足：

$$\begin{aligned} R(h) &\leq \hat{R}(h) + \sqrt{\frac{2\log \Pi_H(m)}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}} \\ &\leq \hat{R}(h) + \sqrt{\frac{2d\log(\frac{em}{d})}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}} \\ &\leq \hat{R}(h) + O\left(\sqrt{\frac{\log(m/d)}{(m/d)}}\right) \end{aligned} \tag{51}$$

接下来介绍一个结论：

Theorem 4. *所有n维的超平面分类函数构成的集合H_n的VCdim为$n+1$。*

一个n维超平面可以用一个n维向量$w = (w_1, w_2...w_n)^T$和一个实数b表示，该平面由$w \cdot x = b$ 确定。所以对于该超平面分类器对点x的分类为$sgn(w \cdot x - b)$，下面给出上述定理的证明：

Proof. 令$m = n+1$，构造$S_X = (x_0, x_1, ..., x_n)$，其中$x_0$为原点，$x_i$ 为n维one−hot向量，第i维为1。则对于任意$S_Y = (y_0, y_1, ..., y_n)$，$y_i \in \{-1, 1\}$，则令$w = (y_1, y_2, ..., y_n)$，$b = y_0/2$，则有：

$$sgn(w \cdot x_i - b) = sgn(y_i - y_0/2) = y_i \tag{52}$$

对所有$0 \leq i \leq n$成立，所以证明了存在一个包含$n+1$ 个点的sample使得其能被H_n打散，接下来证明H_n无法打散任意一个大小为$n+2$的sample。

这需要利用到一个性质，对于任意一个$S = (x_1, x_2, ..., x_{n+2})$，一定存在其一个划分$S_1$和$S_2$，使得$S_1$的凸壳与$S_2$的凸壳相交。有这条性质的话，我们假设可以找到一个超平面分类器能分类S_1和S_2，那么该平面肯定分开了该凸壳，得到这两个凸壳不可能相交，于是产生矛盾，就证明了结论。

那么下面来证明上述性质，我们考虑方程组：

$$\sum_{i=1}^{d+2} \alpha_i x_i = 0 \quad \text{and} \quad \sum_{i=1}^{d+2} \alpha_i = 0 \tag{53}$$

该方程组有$d+2$个未知数$\alpha_{1...d+2}$，和$d+1$个方程，所以肯定有非零解$\beta_1,...,\beta_{d+2}$，那么令

$$I_1 = \{i \in [1, d+2] : \beta_i > 0\} \tag{54}$$

$$I_2 = \{i \in [1, d+2] : \beta_i < 0\} \tag{55}$$

$$\beta = \sum_{i \in I_1} \beta_i \tag{56}$$

则有：

$$\sum_{i \in I_1} \frac{\beta_i}{\beta} x_i = -\sum_{i \in I_2} \frac{\beta_i}{\beta} x_i \tag{57}$$

由凸壳的定义(书上B.4)可以知道，点$\sum_{i\in I_1} \frac{\beta_i}{\beta} x_i$即在$I_1$下标里的点构成的凸壳里，也在$I_2$下标里的点构成的凸壳里。 □

2 Boosting

2.1 Introduction

这一章节讲述的Boosting是一种将多个弱的分类器合成出一个强的分类器的方法。PAC-learnable的条件对我们来说太过苛刻，我们不妨放低一点标准，所以引入一个新的概念：

Definition 5 (Weak learning)**.** 如果一个*Concept Class* C，满足存在一个算法A，和一个常数$\gamma > 0$，一个固定的多项式$poly(.,.,.,.)$使得对于任意$\epsilon > 0$和$\delta > 0$，以及任意分布D，和任意给定$c \in C$，当$m \geq poly(1/\epsilon, 1/\delta, n, size(c))$时：

$$\Pr_{S\sim D^m}\left[R(h_s) \leq \frac{1}{2} - \gamma\right] \geq 1 - \delta \tag{58}$$

简单来说，存在一个学习concept class C的算法A，使得当训练数据越来越多的时候，算法A返回的分类器错误率小于$\frac{1}{2}$的概率趋近于1。这样的算法被称为weak learning algorithm，其返回的分类器(也就是$h \in H$)成为base classifiers。

Boost的中心思想就是运用weak learning algorithm去构造一个strong learner，接下来就来介绍AdaBoost。

2.1.1 AdaBoost

AdaBoost算法如下所示：

Algorithm 1 AdaBoost($S = ((x_1, y_1), (x_2, y_2), ..., (x_m, y_m))$)

for $i = 1 \to m$ **do**
  $D_1(i) \leftarrow \frac{1}{m}$
end for
for $t = 1 \to T$ **do**
  $h_t \leftarrow$ base classfier in H with small error $\epsilon_t = \Pr_{i \sim D_t}[h(x_i) \neq y_i]$
  $\alpha_t \leftarrow \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
  $Z_t \leftarrow 2[\epsilon_t(1-\epsilon_t)]^{\frac{1}{2}}$
  for $i = 1 \to m$ **do**
    $D_{t+1}(i) \leftarrow \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
  end for
end for
$g \leftarrow \sum_{t=1}^{T} \alpha_t h_t$
return $h = sgn(g)$

算法简单来看其实就是迭代T次，第t次迭代找到在分布D_t下错误率ϵ_t**最小**的$h_t \in H$，根据ϵ_t得到h_t在g中的比例α_t，并计算D_{t+1} 开始下一次迭代。

首先来看$\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$，由于$h_t \in H$ 是base classfier，所以$\epsilon_t < \frac{1}{2}$，也就有$\alpha_t > 0$，而且错误率ϵ_t越小，α_t越大，也符合直觉。

再来看分布D_t如何计算，一开始$D_1(i) = \frac{1}{m}$为均匀分布。之后更新D_t的策略是减少$h_t(x_i) = y_i$的分布，而增加$h_t(x_i) \neq y_i$的分布，也就是多“练习”错误的“题”才有进步的空间。

Z_t是一个归一化因子，也即：

$$2[\epsilon_t(1-\epsilon_t)]^{\frac{1}{2}} = \sum_{i=1}^{m} D_t(i)\exp(-\alpha_t y_i h_t(x_i)) \tag{59}$$

接下来说明上式的正确性，由于：

$$\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t} \tag{60}$$

将该结果代入要证明的结论的右边得到：

$$\begin{aligned}
\text{右边} &= \sum_{i=1}^{m} D_t(i)\exp(-\alpha_t y_i h_t(x_i)) \\
&= \sum_{i=1}^{m} D_t(i)\left(\frac{1-\epsilon_t}{\epsilon_t}\right)^{\frac{-1}{2}y_i h_t(x_i)} \\
&= \sum_{y_i\neq h_t(x_i)} D_t(i)\left(\frac{1-\epsilon_t}{\epsilon_t}\right)^{1/2} + \sum_{y_i=h_t(x_i)} D_t(i)\left(\frac{1-\epsilon_t}{\epsilon_t}\right)^{-1/2} \\
&= \epsilon_t\left(\frac{1-\epsilon_t}{\epsilon_t}\right)^{1/2} + (1-\epsilon_t)\left(\frac{1-\epsilon_t}{\epsilon_t}\right)^{-1/2} \\
&= 2(\epsilon_t(1-\epsilon_t))^{1/2} = \text{左边}
\end{aligned} \tag{61}$$

接下来我们给出$\hat{R}(g)$的一个上界：

Theorem 5. *AdaBoost*得到的g的*empirical error* 满足：

$$\hat{R}(g) \leq \exp\left[-2\sum_{t=1}^{T}(\frac{1}{2}-\epsilon_t)^2\right] \tag{62}$$

由该定理可以知道，ϵ_t越小的话，上界就会越紧，所以我们是要选择ϵ_t**尽量小**的h_t，在证明该定理之前，需要一个结论来辅助：

$$D_{t+1}(i) = \frac{e^{-y_i g_t(x_i)}}{m\prod_{s=1}^{t} Z_s} \tag{63}$$

其中$g_t = \sum_{s=1}^{t} g_s\alpha_s$，可以用归纳法证明该结论：

1. 基础：$t=1$时，$D_2(i) = \frac{D_1(i)\exp(-\alpha_1 y_i h_1(x_i))}{Z_1} = \frac{\exp(-y_i g_1(x_i))}{mZ_1}$。
2. 归纳：假设结论对$1,2,...,t-1$成立，由算法的定义有：

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t} \tag{64}$$

由归纳假设可以知道：

$$D_t(i) = \frac{\exp\left(-y_i g_{t-1}(x_i)\right)}{m\prod_{s=1}^{t-1} Z_s} \tag{65}$$

代入上式可以得到：

$$\begin{aligned}
D_{t+1}(i) &= \frac{\exp(-(\alpha_t y_i h_t(x_i) + y_i g_{t-1}(x_i)))}{m\prod_{s=1}^{t} Z_s} \\
&= \frac{\exp(-y_i g_t(x_i))}{m\prod_{s=1}^{t} Z_s}
\end{aligned} \tag{66}$$

所以归纳成立。

有了这个结论，现在证明上述定理：

Proof. 由empirical error 的定义可以知道，$\hat{R}(g) = \frac{1}{m}\sum_{i=1}^{m} 1_{g(x_i)\neq y_i}$，所以有：

$$\begin{aligned}\hat{R}(g) &= \frac{1}{m}\sum_{i=1}^{m} 1_{g(x_i)\neq y_i} \\ &\leq \frac{1}{m}\sum_{i=1}^{m} e^{-g(x_i)y_i}\end{aligned} \tag{67}$$

由于$\sum_{i=1}^{m} D_{T+1}(i) = 1 = \sum_{i=1}^{m} \frac{\exp(-y_i g_T(x_i))}{m\prod_{s=1}^{T} Z_s}$，所以有：

$$\sum_{i=1}^{m} \frac{1}{m}\exp(-y_i g_T(x_i)) = \prod_{s=1}^{T} Z_s = \prod_{s=1}^{T} 2[\epsilon_s(1-\epsilon_s)]^{\frac{1}{2}} \tag{68}$$

代入上式可以得到：

$$\begin{aligned}\hat{R}(g) &\leq \frac{1}{m}\sum_{i=1}^{m} e^{-g(x_i)y_i} \\ &\leq \prod_{s=1}^{T} 2[\epsilon_s(1-\epsilon_s)]^{\frac{1}{2}} \\ &= \prod_{s=1}^{T} 2\sqrt{-\left(\epsilon_s - \frac{1}{2}\right)^2 + \frac{1}{4}} \\ &= \prod_{s=1}^{T} \sqrt{1 - 4\left(\epsilon_s - \frac{1}{2}\right)^2} \\ &\leq \prod_{s=1}^{T} e^{-2\left(\epsilon_s - \frac{1}{2}\right)^2} \\ &= \exp\left[\sum_{s=1}^{T} -2\left(\epsilon_s - \frac{1}{2}\right)^2\right]\end{aligned} \tag{69}$$

□

所以从这个结论我们可以看到，随着迭代次数T的增加，$\hat{R}(g)$会越来越小，但不意味着$R(g)$就会越来越小，也即我们不仅要考虑$\hat{R}(g)$还需要考虑它和$R(g)$的差距，我们先得到g的假设空间$\mathcal{F}_T$：

$$\mathcal{F}_T = \left\{ sgn\left(\sum_{t=1}^{T} \alpha_t h_t\right) : \alpha_t \in \mathbb{R}, h_t \in H, t \in [1, T] \right\} \tag{70}$$

由书上结论，可以知道

$$VCdim(\mathcal{F}_T) \leq 2(d+1)(T+1)\log_2((T+1)e) \tag{71}$$

其中$d = VCdim(H)$。从这个式子可以看出，$VCdim(\mathcal{F}_T)$按$O(T\log T)$增长，由前面VCdim 的性质可以知道，$\hat{R}(g)$和$R(g)$的差距在不断增大，也就是说随着迭代次数的增多，可能导致Overfit。

3 On-Line Learning

3.1 Introduction

On-Line Learning指的是训练样例不是一下子全部给出，而是分成T轮，每一轮给出一个数据(x_t, y_t)，我们需要利用这个数据修正我们的模型，这时候我们就不能假设训练数据满足某种分布，而且我们还希望学会某种concept之前犯尽量少的错误。接下来就介绍几种在线学习的思路，以及评价标准。

3.2 Prediction with expert advice

这是一类在线学习的思路，先给N个expert，其实也就是一个大小为假设空间H，每遇到一个数据x_t，会综合H中expert给出预测$\hat{y}_t$，然后得到$label\ y_t$，根据预测是否正确来调整expert的“发言权”，最后得到一个模型。

引入一个衡量标准regret(我们希望最小化regret)，它的定义式为：

$$R_T = \sum_{t=1}^{T} L(\hat{y}_t, y_t) - \min_{i=1}^{N} \sum_{t=1}^{T} L(y_{t,i}, y_t) \tag{72}$$

$L(\hat{y}_t, y_t)$是Loss-function，R_T表示T轮下来总的Loss，和犯错误最少的那个expert的Loss的差。这就是为什么叫做regret，如果当时听了那个最聪明的expert，就不会多犯这么多错误了。

3.2.1 Mistake bounds and Halving algorithm

在这里我们仅考虑realizable-case，也就是$c \in H$。Halving algorithm 就是一个非常简单的想法，遇到一个x_t，让H中的所有expert都做出预测，然后和人数多的（一定过半）一方做一样的预测$\hat{y}_t$，如果错了，就把这一堆expert全部从H中踢出，这样既保证了正确性也让H 减少了至少一半。

这里引入另外一个衡量标准mistake bound model，它衡量的是一个在线学习算法A的能力，首先固定concept c，令：

$$M_A(c) = \max_{x_1,x_2...x_T} |mistakes(A,c)| \tag{73}$$

下标$x_1, x_2...x_T$表示任取一个正整数T，且T 轮给出的x_t也是任取，综合起来就是任取一个在线序列。

然后再扩展到concept class C：

$$M_A(C) = \max_{c \in C} M_A(c) \tag{74}$$

接下来我们给出Halving algorithm的mistake bound：

Theorem 6. 对于有限大小的假设空间H，考虑*realizable-case*时有：

$$M_{Halving}(H) \leq \log_2 |H| \tag{75}$$

Proof. 根据Halving algorithm算法的性质，每一次犯错，会导致H减少一半，所以犯错达到$\log_2 |H|$之后，H中就只剩下一个expert了，由于是realizable-case，所以剩下的这一个expert就是c，而c是不会犯错的。 □

如此我们就找到了其上界，我们还可以找到其下界：

Theorem 7. 对于有限大小的假设空间H，考虑*realizable-case*时有：

$$VCdim(H) \leq M_{Halving}(H) \leq \log_2 |H| \tag{76}$$

Proof. 令$d = VCdim(H)$，也即存在$S = (x_1, x_2...x_d)$满足$|H_S| = 2^d$，我们就按照对手策略构造$y_1, y_2...y_d$，也即第$t(1 \leq t \leq d)$轮，如果$\hat{y}_t = 1$，就让$y_t = -1$，可以知道在这d轮，算法永远都在犯错而且H不会为空，所以找到一组S，使得其犯错至少为d次，证明了结论。 □

3.2.2 Weighted majority algorithm

Halving algorithm之所以敢把犯错的都**去掉**，就是因为其考虑的是realizable-case，但是现实中，这个假设很难实现，我们不能保证H中一定有一个永远不会错的expert，只能够让算法最后能达到准确率$p(p \leq 1)$，这时候去掉犯错的expert就不妥了，可能到最后H就被删光了。一个较为缓和的方法，就是根据一个expert的犯错次数的多少来决定它的决定在最终决定中的比重。

这就是Weighted majority algorithm，它赋予每个H中h_i一个权重$w_{t,i}$，该权重会随着算法的进行更新，所以与t有关，算法流程大致如下：

1. 初始化$w_{1,i}=1$，对于$1\leq i\leq N$。
2. 第t轮得到x_t，如果$\sum_{h_i(x_t)=1} w_{t,i} > \sum_{h_i(x_t)=-1} w_{t,i}$，那么$\hat{y}_t=1$，否则$\hat{y}_t=-1$。
3. 更新$w_{t+1,i}$：

$$w_{t+1,i}=\begin{cases} w_{t,i} & h_i(x_t)=y_t \\ \beta w_{t,i} & h_i(x_t)\neq y_t \end{cases} \tag{77}$$

4. 令$t+=1$，goto 步骤2，直到$t>T$。

由于现在是non-realizable-case，像之前算法那样用$M_A(C)$来评估其性能就不妥，因为极端错误可能会随着T 的增加而增加，于是我们固定轮数T，再来比较，也即用m_T表示T轮数据下来，算法一共出错次数，用m_T^*表示T轮数据下来，出错最少的expert的出错次数，那么可以提供m_T的上界：

Theorem 8. 假如$\beta\in(0,1)$，且$|H|=N$，假设T轮的数据固定，m_T表示这T轮算法出错个数，m_T^*表示这T轮中犯错最少的*expert*的出错次数，则有：

$$m_T\leq\frac{\log N+m_T^*\log\frac{1}{\beta}}{\log\frac{2}{1+\beta}} \tag{78}$$

Proof. 用势能分析可以简单地证明这个结论，定义势能函数$W_t=\sum_{i=1}^N w_{t,i}$，它满足如下性质：

1. W_t随着t的增大单调不上升。
2. $W_t\geq w_{t,i}\geq 0$ 对于$1\leq i\leq N$都成立。

考虑第t轮算法做了错误的预测，那么一定有做出错误决定的一方的权值和大于$1/2W_t$，根据算法描述，这些expert都会受到β的惩罚，所以一定有：

$$W_{t+1}\leq\frac{1}{2}\beta W_t+\frac{1}{2}W_t=\frac{1+\beta}{2}W_t \tag{79}$$

其中$\frac{1+\beta}{2}<1$，所以有：

$$W_T\leq\left(\frac{1+\beta}{2}\right)^{m_T}N \tag{80}$$

假设在这T轮中最厉害的expert为h_i，有性质2可以知道：

$$W_T\geq w_{T,i}=\beta^{m_T^*} \tag{81}$$

联立如上两式子，可以得到：

$$\beta^{m_T^*} \leq \left(\frac{1+\beta}{2}\right)^{m_T} N \tag{82}$$

$$m_T^* \log(\beta) \leq m_T \log\left(\frac{1+\beta}{2}\right) + \log N \tag{83}$$

$$-m_T^* \log(\beta) \geq -m_T \log\left(\frac{1+\beta}{2}\right) - \log N \tag{84}$$

$$\frac{m_T^* \log\left(\frac{1}{\beta}\right) + \log N}{\log\left(\frac{2}{1+\beta}\right)} \geq m_T \tag{85}$$

$$\tag{86}$$

□

这就证明了结论。

3.2.3 Randomized weighted majority algorithm

我们还可以将上述算法改成一个随机算法，只需要把$w_{t,i}$ 变成概率就行：

1. 初始化$w_{1,i} = 1$，$p_{1,i} = 1/N$，对于$1 \leq i \leq N$。
2. 第t轮得到x_t，算法按照$p_{t,i}$的概率决定选择h_i的决定。
3. 更新$p_{t+1,i}$，$w_{t+1,i} = 1$：

$$w_{t+1,i} = \begin{cases} w_{t,i} & h_i(x_t) = y_t \\ \beta w_{t,i} & h_i(x_t) \neq y_t \end{cases} \tag{87}$$

$$p_{t+1,i} = w_{t+1,i} / \sum_{i=1}^{N} w_{t+1,i}(\text{归一化}) \tag{88}$$

4. 令$t+=1$，goto 步骤2，直到$t > T$。

为什么要使用随机算法，这是因为任意一个**确定性算法**A都无法做到让$R_T = o(N)$($O(N)$但不是$\Theta(n)$)，因为可能会碰到很极端的x_t和N，比方说$N = 2$，而两个expert 一个只返回1，另一个只返回-1。而且每一轮，y_t都和$\hat{y}_t$相反（这是可行的，因为算法是确定的），那么$m_T = T$，$m_T^* \leq T/2$，所以：

$$R_T = m_T - m_T^* \geq T/2 = \Omega(N) \tag{89}$$

对于随机算法，我们需要修改R_T的定义，这是因为对于给定的x_t，算法返回的结果并不确定，可以用其期望来代替：

$$R_T = \mathcal{L}_T - \mathcal{L}_T^{\min} \tag{90}$$

其中$\mathcal{L}_T = \sum_{t=1}^T E\left[l(y_t, \hat{y}_t)\right] = \sum_{t=1}^T \sum_{i=1}^N p_{t,i} l_{t,i}$，表示$T$轮下来的期望错误。

$\mathcal{L}_T^{\min} = \min_{i=1}^N \sum_{t=1}^T l_{t,i}$。表示犯错误最少的expert的犯错数，这一个量是不涉及随机变量的。

这时候我们可以给出$\mathcal{L}_T$和$\mathcal{L}_T^{\min}$差距的一个bound，进而给出R_T 的bound：

Theorem 9. 假定$\beta \in [1/2, 1)$，那么对于$T > 1$，有如下不等式成立：

$$\mathcal{L}_T \leq \frac{\log N}{1-\beta} + (2-\beta)\mathcal{L}_T^{min} \tag{91}$$

实际中，令$\beta = \max\left\{1/2, 1 - \sqrt{(\log N)/T}\right\}$，那么有如下*bound*成立：

$$\mathcal{L}_T \leq \mathcal{L}_T^{\min} + 2\sqrt{T \log N} \tag{92}$$

Proof. 令$L_t = \sum_{i=1}^N p_{t,i} l_{t,i}$，$W_t = \sum_{i=1}^N w_{t,i}$，所以有$p_{t,i} = w_{t,i}/W_t$。类似Weighted majority algorithm，我们采用势能分析的方法，将W_t作为势能函数。

$$\begin{aligned} W_{t+1} &= \sum_{l_{t,i}=1} \beta w_{t,i} + \sum_{l_{t,i}=0} w_{t,i} \\ &= \sum_{i=1}^N w_{t,i} + \sum_{l_{t,i}=1} (\beta - 1) w_{t,i} \\ &= W_t + W_t \sum_{l_{t,i}=1} (\beta - 1) p_{t,i} \\ &= W_t + W_t(\beta - 1) L_t \\ &= W_t(1 + (\beta - 1) L_t) \end{aligned} \tag{93}$$

所以可以得到：$W_{T+1} = N \prod_{t=1}^T (1 + (\beta - 1) L_t)$，又因为$W_{T+1} \geq w_{T+1,i}$，所以$W_{T+1} \geq \beta^{\mathcal{L}_T^{min}}$，联立可以得到：

$$N \prod_{t=1}^T (1 + (\beta - 1) L_t) \geq \beta^{\mathcal{L}_T^{min}} \tag{94}$$

两边取Log得到：

$$\log N + \sum_{t=1}^{T} \log(1-(1-\beta)L_t) \geq \mathcal{L}_T^{min} \log \beta \tag{95}$$

当$0 < x < 1$时，$\log(1-x) \leq -x$，所以可以放缩上式：

$$\begin{aligned}\mathcal{L}_T^{min} \log \beta &\leq \log N - \sum_{t=1}^{T} \log(1-(1-\beta)L_t) \\ &\leq \log N - \sum_{t=1}^{T} (1-\beta)L_t \\ &= \log N - (1-\beta)\mathcal{L}_T\end{aligned} \tag{96}$$

两边同时乘以-1，并带进log内部可以得到：

$$(1-\beta)\mathcal{L}_T \leq \mathcal{L}_T^{min} \log \frac{1}{\beta} + \log N \tag{97}$$

$$\begin{aligned}\mathcal{L}_T &\leq -\frac{\mathcal{L}_T^{min} \log \beta}{(1-\beta)} + \frac{\log N}{(1-\beta)} \\ &\leq -\frac{\log(1-(1-\beta))}{(1-\beta)} \mathcal{L}_T^{min} + \frac{\log N}{(1-\beta)}\end{aligned} \tag{98}$$

由于$-\log(1-x) \leq x + x^2$，所以：

$$\mathcal{L}_T \leq (2-\beta)\mathcal{L}_T^{min} + \frac{\log N}{(1-\beta)} \tag{99}$$

到此为止证明了第一部分，让右边的式子对β求导并令其为0，就可以找到一个最紧的bound，此时$\beta_0 = 1 - \sqrt{(\log N)/T}$，如果$\beta_0 > 1/2$（定理的前提），则最优解在$\beta_0$取到，否则在边界1/2取到。这就证明了定理的第二部分，并做适当变形可以得到：

$$\mathcal{R}_T \leq 2\sqrt{T \log N} \tag{100}$$

□

这个定理说明了，如上随机算法RWM的期望regret的上界为$\mathcal{R}_T \leq 2\sqrt{T \log N}$，接下来的定理说明如果仅用**期望regret**作为指标的话，RWM 已经达到了极限:

Theorem 10. 当$N = 2$，存在一个*loss*的随机*loss vector*序列，使得对于任意在线学习算法都有：$E[R_T] \geq \sqrt{T/8}$。

Proof. 令l_t等概率从$(0,1)^T$和$(1,0)^T$中取得，不论是随机算法还是确定算法，都可以用$p_{t,i}$的形式来描述：

$$\begin{aligned} E[R_t] &= E[\mathcal{L}_T] - E[\mathcal{L}_T^{min}] \\ &= \sum_{t=1}^{T} E[p_{t,1}l_{t,1} + p_{t,2}l_{t,2}] - E[\mathcal{L}_T^{min}] \\ &= T/2 - E[\mathcal{L}_T^{min}] \end{aligned} \tag{101}$$

又因为：

$$\begin{aligned} \mathcal{L}_T^{min} &= \min\{\mathcal{L}_{T,1}, \mathcal{L}_{T,2}\} \\ &= \frac{1}{2}(\mathcal{L}_{T,1} + \mathcal{L}_{T,2} - |\mathcal{L}_{T,1} - \mathcal{L}_{T,2}|) \\ &= T/2 - |L_{T,1} - T/2| \end{aligned} \tag{102}$$

所以：

$$\begin{aligned} E[R_t] &= E[|L_{T,1} - T/2|] \\ &= E[|\sum_{t=1}^{T} \frac{1+\sigma_t}{2} - T/2|] \\ &= E[|\sum_{t=1}^{T} \frac{\sigma_t}{2}|] \end{aligned} \tag{103}$$

使用书上的定理D.4，并且令$x_t = 1/2$得到：

$$E[R_t] = E[|\sum_{t=1}^{T} \sigma_t x_t|] \geq \sqrt{\frac{1}{2}\sum_{t=1}^{T} x_t^2} = \sqrt{T/8} \tag{104}$$

□

如何使用这个定理去证明$R_t = \Omega(\sqrt{T\log N})$，**没有想出来！！！**。

4 Dimensionality Reduction

4.1 Principal Component Analysis

我们经常用一个n维的向量来描述一个事物，比如说一个单词，一张图片。那么有时候n太大导致处理太困难。希望能将数据重新表示成一个k维（k 远远小于n）的向量。比如有100个2维向量，恰好都落在一条直线上，那么只需要一个Pd，加

上100个实数就可以表示，但是一般情况下数据不可能恰好落在一条直线上。那么想把这100 个点从二维降到一维，就必须要有所损失。那么一个直观的做法就是，找到一条直线，使得这100个点离这条直线的偏离最小，然后对于某个点，它的降维后的结果就是在这个直线上的投影。

4.1.1 奇异值分解(Singular Value Decomposition

首先需要约定一下奇异值分解SVD的形式（与线性代数(下)有点不同），对于一个n行m列的矩阵A，设其SVD分解为：

$$A = U\Sigma V^T \tag{105}$$

其中$U = (u_1, u_2, ..., u_r)$是一个n行r列的矩阵，r为A的秩(rank)，Σ为r行r列的对角方阵。V为m行r 列的矩阵。

满足U中r个列向量两两正交，也即有$U^TU = I_r$，类似地有$V^TV = I_r$。

4.1.2 正交投影矩阵(orthogonal projection matrix)：

给定k个相互正交的n维向量$(a_1, a_2...a_k)$构成一组基A，则对于给定任意n维向量b，分别求解b在a_i上的投影长度x_i，再合成得到其投影向量$p = \sum_{i=1}^k a_ix_i$。由于$x_i = \frac{a_i^Tb}{a_i^Ta_i}$，故可以得到：

$$p = (\sum_{i=1}^{k} \frac{a_ia_i^T}{a_i^Ta_i})b \tag{106}$$

令$P = \sum_{i=1}^k \frac{a_ia_i^T}{a_i^Ta_i}$，$U_k = (\frac{a_1}{\sqrt{a_1^Ta_1}}, ...\frac{a_k}{\sqrt{a_k^Ta_k}})$，则有:

$$P = U_kU_k^T \tag{107}$$

P就是正交投影矩阵，正交表示其对应的基A是正交的，投影矩阵的意思就是用P左乘某个向量b就可以得到其在A上的投影p。那么一个正交投影矩阵满足以下性质：

1. $P^T = P$：由公式(2)可以得到。
2. $P^2 = P$：一个向量投影到A上之后，再投影还是这个向量。

4.1.3 PCA

假定输入数据为$X = (x_1, x_2...x_m)$，X是一个$n \times m$的矩阵，我们希望将其降至k维，并且使得受到的损失最小，也即找到一个正交投影矩阵P^*（P^* 对应k 个n

维正交基），满足：

$$P^* = \arg\min_P \|PX - X\|_F \tag{108}$$

也就是投影后，两个矩阵的差距最小，由于：

$$\begin{aligned}\|PX - X\|_F^2 &= Tr((PX - X)^T(PX - X)) \\ &= Tr(X^TP^TPX - X^TPX - X^TP^TX - X^TX) \\ &= Tr(-X^TPX - X^TX) \\ &= Tr(-X^TPX) - Tr(X^TX)\end{aligned} \tag{109}$$

上述推导用到了$P^T = P$，$P^2 = P$，和矩阵**迹Tr**算符的线性性质。由于X是给定的，所以：(假设$P = U_kU_k^T$，U_k为$n \times k$矩阵，且列向量互相正交)。

$$\begin{aligned}P^* &= \arg\min_P \|PX - X\|_F \\ &= \arg\max_P Tr(X^TPX) \\ &= \arg\max_{U_k} Tr((X^TU_k)((X^TU_k)^T)) \\ &= \arg\max_{U_k} Tr(U_k^T(XX^T)U_k) \\ &= \arg\max_{U_k} \sum_{i=1}^{k} u_i^T(XX^T)u_i\end{aligned} \tag{110}$$

令$C = XX^T$，取u_i为将C奇异值分解后，第i个左奇异值向量(left singular vector)。则可以得到最优解U_k，所以$PX = U_kU_k^TX$，令$Y = U_k^TX$，就得到降维后的向量。(后面可以看到，其实左右奇异值向量是一样）

下面解释原理，假设X的奇异值分解为：

$$X = U\Sigma V^T \tag{111}$$

那么有：

$$\begin{aligned}XX^T &= U\Sigma V^TV\Sigma^TU^T \\ &= U\Sigma\Sigma^TU^T \\ &= \sum_{i=1}^{r} \sigma_i^2 u_iu_i^T\end{aligned} \tag{112}$$

所以有$u_1XX^Tu_1^T = \sigma_1^2$最大，$u_2XX^Tu_2^T = \sigma_2^2$次大.....

4.2 Kernel Principal Component Analysis

有时候在n维下，线性分类无法区分一类概念，但是将其映射到更高的维度就可以了。本来我们是直接把数据从n维降到k 维，现在我们是将数据先映射到一个更高的维度的Hilbert space（定义了内积的空间），然后再降到k维。我们设这样的映射为$\Phi(x)$，令映射后的数据为$X = (\Phi(x_1), \Phi(x_2), ..., \Phi(x_m))$，然后对$X'$套用PCA 即可。

不过我们有时候并不需要知道映射$\Phi(x)$是什么，我们只需要知道一个Kernel Matrix K，它表示两两元素的在高维空间的内积。假设我们已经把数据映射后得到了X，注意这时候X 就不是n行，m列了（因为维度更高）。那么我们可以定义$K = X^T X$（注意这里的矩阵乘法，内积由其所在的高维空间定义，但是由于内积的性质，我们依然可以套用原来的写法）。

直接对X进行PCA，也就是对XX^T进行SVD，我们假设X的SVD为$X = U\Sigma V^T$，那么有$XX^T = U\Sigma^2 U^T$，$K = V\Sigma^2 V^T$。令$\Lambda = \Sigma^2$，λ_i表示第i个对角元素。

我们的目标是，将降维结果Y能用K表示，而不是X的形式。由之前的结论:

$$Y = U_k^T X \tag{113}$$

那么我们先把U_k试着用K来表示，我们把$X = U\Sigma V^T$ 两边同时右乘$V\Sigma^{-1}$得到$U = XV\Sigma^{-1}$（Σ 是可逆的），改写一下：

$$\begin{aligned} U &= XV\Lambda^{-1/2} \\ &= X(\frac{v_1}{\sqrt{\lambda_1}}, \frac{v_2}{\sqrt{\lambda_2}}, ..., \frac{v_r}{\sqrt{\lambda_r}}) \end{aligned} \tag{114}$$

所以有:

$$U_k = (X\frac{v_1}{\sqrt{\lambda_1}}, X\frac{v_2}{\sqrt{\lambda_2}}, ..., X\frac{v_k}{\sqrt{\lambda_k}}) \tag{115}$$

带入得到：

$$\begin{aligned} Y &= U_k^T X \\ &= (X\frac{v_1}{\sqrt{\lambda_1}}, X\frac{v_2}{\sqrt{\lambda_2}}, ..., X\frac{v_k}{\sqrt{\lambda_k}})^T X \\ &= (X^T X\frac{v_1}{\sqrt{\lambda_1}}, X^T X\frac{v_2}{\sqrt{\lambda_2}}, ..., X^T X\frac{v_k}{\sqrt{\lambda_k}})^T \\ &= (K\frac{v_1}{\sqrt{\lambda_1}}, K\frac{v_2}{\sqrt{\lambda_2}}, ..., K\frac{v_k}{\sqrt{\lambda_k}})^T \end{aligned} \tag{116}$$

由特征向量性质$Kv_i = \lambda_i v_i$，代入得到：

$$Y = (\sqrt{\lambda_1}v_1, \sqrt{\lambda_2}v_2, ..., \sqrt{\lambda_k}v_k)^T \tag{117}$$

也就是给定核矩阵K，就能够求出原m个数据的降维表示。

4.3 Johnson-Lindenstrauss lemma

该引理说的是，对于m个n维的点，可以用一个映射将其降维至$k(k \geq O(\frac{\log m}{\epsilon^2}))$，同时满足任意两点之间的距离比原来不超过$(1 \pm \epsilon)$ 倍。下面开始证明：

Lemma 11. 假定Q服从自由度为k的卡方分布，则对于任意$0 < \epsilon < 1/2$，有如下不等式成立：

$$\Pr[(1-\epsilon)k \leq Q \leq (1+\epsilon)k] \geq 1 - 2e^{-(\epsilon^2-\epsilon^3)k/4} \tag{118}$$

Proof. 正难则反，先计算Q在取值范围外的概率，再减去。

$$\begin{aligned}\Pr[Q \geq (1+\epsilon)k] &= \Pr[\exp(\lambda Q) \geq \exp(\lambda(1+\epsilon)k)] \\ &\leq \frac{E[\exp(\lambda Q)]}{\exp(\lambda(1+\epsilon)k)}\end{aligned} \tag{119}$$

$$E[\exp(\lambda Q)] = \prod_{i=1}^{k} E[e^{\lambda X_i^2}] \tag{120}$$

$$\begin{aligned}E[e^{\lambda X_i^2}] &= \int_{-\infty}^{\infty} \frac{e^{\lambda t^2}}{\sqrt{2\pi}} e^{-t^2/2} \mathrm{d}t \\ &= \frac{1}{\sqrt{1-2\lambda}}\end{aligned} \tag{121}$$

式(17)要求$Re(\lambda) < 1/2$。所以(15)可以继续写成：

$$\Pr[Q \geq (1+\epsilon)k] \leq \frac{(1-2\lambda)^{-k/2}}{\exp(\lambda(1+\epsilon)k)} \tag{122}$$

将等式右边对λ 求导，并令导数等于0，得到:

$$\lambda^* = \frac{\epsilon}{2(\epsilon+1)} < 1/2 \tag{123}$$

带入λ^*得到：

$$\Pr[Q \geq (1+\epsilon)k] \leq (\frac{1+\epsilon}{\exp(\epsilon)})^{k/2} \tag{124}$$

考虑到

$$\begin{aligned}\exp(\epsilon - (\epsilon^2-\epsilon^3)/2) &\geq 1 + [\epsilon - (\epsilon^2-\epsilon^3)/2] + [\epsilon - (\epsilon^2-\epsilon^3)/2]^2/2 \\ &= (1 + \epsilon + (5\epsilon^4)/8 - \epsilon^5/4 + \epsilon^6/8) \\ &\geq 1 + \epsilon\end{aligned} \tag{125}$$

故有：

$$\Pr[Q \geq (1+\epsilon)k] \leq \exp(-k(\epsilon^2-\epsilon^3)/4) \tag{126}$$

类似可以证明:

$$\Pr[Q \leq (1-\epsilon)k] \leq \exp(-k(\epsilon^2-\epsilon^3)/4) \tag{127}$$

由Union bound可以得到引理成立。 □

Lemma 12. 给定n维向量x，和一个k行n列的矩阵A，保证A中的元素均独立同$N(0,1)$分布，那么对于任意$0<\epsilon<1/2$有：

$$\Pr[(1-\epsilon)\left\|x\right\|^2 \leq \left\|\frac{1}{\sqrt{k}}Ax\right\|^2 \leq (1+\epsilon)\left\|x\right\|^2] \geq 1-2e^{-(\epsilon^2-\epsilon^3)k/4} \tag{128}$$

Proof.

$$\begin{aligned} &\text{上式左边} \\ &= \Pr[(1-\epsilon)k \leq \left\|Ax\right\|^2/\left\|x\right\|^2 \leq (1+\epsilon)k] \end{aligned} \tag{129}$$

令$\hat{x}=Ax$，$T_j=\hat{x}_j/\left\|x\right\|$，有$T_j$服从$N(0,1)$，故得到$Q=\sum_{i=1}^{k}T_j^2$服从自由度为$k$的卡方分布。由Lemma11 可以得到结论成立。 □

Lemma 13 (Johnson-Lindenstrauss)**.** 对于任意$0<\epsilon<1/2$，和任意整数$m>4$，令$k=\frac{20\log m}{\epsilon^2}$。则对于任意$n$为空间的$m$个点构成的集合$V$，存在一个映射$f: R^n \to R^k$，使得对于任意$u,v\in V$，

$$(1-\epsilon)\left\|u-v\right\|^2 \leq \left\|f(u)-f(v)\right\|^2 \leq (1+\epsilon)\left\|u-v\right\|^2 \tag{130}$$

Proof. 我们将V中的点，标号为$v_1,v_2,...,v_m$，令事件A_{ij}表示$x=v_i-v_j$满足Lemma12的不等式。则只要能说明：

$$\Pr[\bigcap_{1\leq i<j\leq m} A_{ij}] > c \tag{131}$$

其中c为一给定大于0的常数。由Lemma2可知，

$$\Pr[A_{ij}^C] = 1-\Pr[A_{ij}] \leq 2e^{-(\epsilon^2-\epsilon^3)k/4} \tag{132}$$

所以可以得到：

$$\begin{aligned} \Pr[\bigcap_{1\leq i<j\leq m} A_{ij}] &= 1-\Pr[\bigcup_{1\leq i<j\leq m} A_{ij}^C] \\ &\geq 1-\sum_{1\leq i<j\leq m}\Pr[A_{ij}^C] \\ &\geq 1-(m-1)m/2 * 2e^{-(\epsilon^2-\epsilon^3)k/4} \end{aligned} \tag{133}$$

取$\epsilon = 1/2$上式概率取到最小，当$k = \frac{20\log m}{\epsilon^2}$时，有：

$$
\begin{aligned}
\Pr[\bigcap_{1\le i<j\le m} A_{ij}] &\ge 1-(m-1)m/2 * 2m^{-(5-2.5)} \\
&\ge 1-m^2 * 2m^{-2.5} \\
&\ge 1-2m^{-0.5} \\
&> 0
\end{aligned}
\tag{134}
$$

□