机器学习

机器学习算法的本质是：从具有共性的数据中，**学习机**在所有可能存在的函数中（可行域）中搜寻无法完全解释的**显式的**数学公式，以达到范化能力（学习机的嵌入）。统计学习理论

SVM

基本SVM

原始问题，最大化间隔（margin）$\frac{2}{||\boldsymbol{w}||}$，等价

$$\begin{aligned} &\min_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \\ &s.t.\ (\boldsymbol{w}^T x_i + b) y_i \geqslant 1 \qquad \forall i \end{aligned}$$

用Lagrange，转化为对偶问题$\min\ \ \max$，利用KKT条件转换为$\max\ \ \min$，极小化后者，于是变为 $\max$，最后求出Lagrange乘子 α

利用KKT条件还能得到 $\alpha_i(y_i f(\boldsymbol{x}_i) - 1) = 0$，说明支撑向量必然是使得 $y_i f(x_i) = 1$的点(x_i, y_i)

软间隔SVM

$$\begin{aligned} &\min_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + c\sum_{i=1}^n \varepsilon_i \\ &s.t.\ (\boldsymbol{w}^T x_i + b) y_i \geqslant 1 - \varepsilon_i \qquad \forall i \end{aligned}$$

加入Lagrange乘子α_i, r_i得到极值条件：

$$\begin{aligned} L(\boldsymbol{w}, b, \varepsilon) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + c\sum_{i=1}^n \varepsilon_i - \sum_{i=1}^n \alpha_i(\boldsymbol{w}^T x_i + b y_i - 1 + \varepsilon_i) - \sum_{i=1}^n r_i \varepsilon_i \\ \frac{\partial L(\boldsymbol{w}, b, \varepsilon)}{\partial w} = w - \sum_{i=1}^n \alpha_i y_i x_i \Rightarrow w = \sum_{i=1}^n \alpha_i y_i x_i \\ \frac{\partial L}{\partial b} = \sum_{i=1}^n \alpha_i y_i = 0 \\ \frac{\partial L}{\partial \varepsilon_i} = C - \alpha_i - r_i = 0 \end{aligned}$$

最后等价结果为，仍为凸的二次问题

$$
\begin{aligned}
\min_{\alpha} \quad & L=\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i^T x_j-\sum_i \alpha_i \\
s.t. \quad & \sum_i \alpha_i y_i = 0 \quad 0\leqslant \alpha_i \leqslant C \\
& \alpha_i((\boldsymbol{w}^T x_i+b)y_i-1+\varepsilon_i)=0 \\
& r_i\varepsilon_i=0
\end{aligned}
$$

进一步得到

$$
\begin{cases}
(\boldsymbol{w}^T x_i+b)y_i>1\Rightarrow \alpha_i=0 & \text{分对的点} \\
(\boldsymbol{w}^T x_i+b)y_i<1\Rightarrow \alpha_i=C & \text{分错的点，也有分对的在间隔内的点} \\
(\boldsymbol{w}^T x_i+b)y_i=1\Rightarrow 0\leqslant\alpha_i\leqslant C & \text{分对且在间隔边界上的点}
\end{cases}
$$

$$
\begin{aligned}
&(w^T x_i+b)y_i>1\Rightarrow \varepsilon_i=0 \\
&(w^T x_i+b)y_i<1\Rightarrow \varepsilon_i=1-(w^T x_i+b)y_i \\
&(w^T x_i+b)y_i=1\Rightarrow \varepsilon_i=0
\end{aligned}
$$

第三个条件，分为$\alpha_i=0$和$\neq 0$的两种情况讨论即可。

将三个式子合在一起就是Hingen Loss：

$$
\varepsilon_i=L(f_w(x_i)y_i)=(1-(w^T x_i+b)y_i)_+
$$

求解α,w,b：求解 α 通过凸优化方法，再求得 w，最后通过间隔边界上的点确定 b。

一般二分类方法：$f(x_i)y_i=(w^T x_i+b)y_i$是一个用来衡量分类标准，如果是在$(0,1)$中分类正确，二分类的损失函数均为

$$
L(f_w(x)y)
$$

的形式

核化SVM

方法1：升维 $x\leftarrow\varphi(x)$，难以找到有效的升维方法

方法2：找到高维空间的一组基（核函数）
$K(x,y)=\varphi^T(x)\varphi(y):\mathbb{R}^d\times\mathbb{R}^d\rightarrow\mathbb{R}$，需要特征是内积形式 $x^T x$：

$$
\begin{aligned}
&\min_{\alpha} \quad L=\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i^T x_j \\
\xRightarrow{\text{核化}} &\min_{\alpha} \quad L=\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j K(x_i,x_j)
\end{aligned}
$$

Gauss核函数 $K(x,x')=\exp\{-\frac{1}{2\sigma^2}||x-x'||^2\}$，于是核化方法等价于对 $(\varphi(x_i),y_i)$ 求SVM。

参数选择

核函数的参数和惩罚系数C，**验证型调参**：交叉验证 $n-fold$，$leave\ one\ out$，LOO。

预测超参数?

统计学习理论

机器学习基本要素：

- 训练数据集 $\{x_i, y_i\}_{i=1}^n$
- 表现度量：目标函数
- 学习机（备选函数集）：$H=\{f(x,\alpha)|\alpha\in\Omega\}$

如果数据满足某种分布（数据采样充分时，才可能找到最优函数的逼近），则一定存在最优的目标函数逼近

- 输入：训练数据与学习机
- 输出：学习机最近接未知目标函数的逼近

风险函数（损失函数）：

$$L(y, f(x,\alpha))$$

实际风险：

$$R(f)=\int_{X,Y} L(y,f(x,\alpha))\,dF(y|x)F(x)=\int_{X,Y} L(y,f(x,\alpha))\,dF(x,y)$$

最优分类器：

- $f^*=\arg\inf_{f\in H} R(f)=OPT(H)$
- ε 误差解：$R(f^*)\leqslant OPT(H)+\varepsilon$

  $\varepsilon-\delta$ 解：$P(R(f^*)\leqslant OPT(H)+\varepsilon)>1-\delta$
- PAC理论：依概率近似准确，$P(|R(h)-\hat{R}(h)|\leqslant\varepsilon)\geqslant 1-\delta$。

  学习目标：寻找 $\varepsilon-\delta$ 解

构造学习算法的两个基本原理：

- 经验风险：$R_{emp}(f)=\frac{1}{n}\sum_{i=1}^n L(y_i, f(x_i,\alpha))$
- 经验风险极小化就能满足PAC理论：$f^*=\arg\inf_{f\in H} R_{emp}(f)$

学习机H**容量的度量**：VC维

- n 个数据样本 $x_1,\cdots,x_n$，总共有 2^n 个二分类可能。

- 若H能将$x_1, \cdots, x_n$所有可能取值都取到（打散shatter）
- 能够被H打散的数据点的最大数目

GMM & EM 算法

GMM聚类(Gaussian Mixture Model)

将所有的变量（观测变量x, y和隐变量θ, ω）都视为服从某种概率分布，理解模型参数计算是在求$p(\theta|D)$，即模型参数θ的后验分布，D为训练数据集，所以可以通过MLE求解模型参数θ。

以概率的方式理解聚类问题，混合概率分布模型的隐变量记为y来自概率$p(y)$，聚类的本质可以视为：特征x来自于与y相关的某个分布$p(x|y)$，于是$p(x,y) = p(y)p(x|y) = \pi_k N(x|\mu_k, \Sigma_k)$其中$k$为(x,y)的类别。

$$p(x) = \sum_y p(x,y) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

考虑通过最大似然求解$\theta \in \{\mu_k, \sigma_k, \pi_k\}$即

$$\max_\theta \prod_{i=1}^{n} p(x_i|\theta)$$

分类问题可以理解为$p(x,y) = p(x)p(y|x)$，根据x求出y的概率分布。

假设存在K个混合分布，其中第i个分布为Gauss分布$N(\mu_i, \Sigma_i)$。

数据生成方法：根据$\pi_i = P(y=i)$选择一个Gauss分布，根据对应的分布$N(\mu_i, \Sigma_i)$生成特征x。

用途：根据特征预测隐变量的分布（分类）：

$$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{\pi_y N(\mu_y, \Sigma_y)}{\sum_{k=1}^{K} \pi_k N(\mu_k, \Sigma_k)}$$

K均值：硬分类；GMM：软分类，得到每个特征属于每个类的概率，且具有生成功能。

假设类别k的每个x_i都来自某个均值未知的Gauss分布，于是μ_k的似然函数为

$$\prod_{i=1}^{n} p(x_i) = \prod_{i=1}^{n} N(\mu_{x_i}, \sigma^2) = \prod_{i=1}^{n} \left(\frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{||x-\mu_{x_i}||^2}{2\sigma^2}) \right)$$

取对数似然之后，上式的MLE等价为

$$\min_{\mu_x} \sum_{i=1}^{n} ||x-\mu_k||^2$$

而这正好就是对x的K-means算法，但是每个类别需要有相同的方差。

通过$\log \frac{p(y=i|x)}{p(y=j|x)}=0$可以确定出分类面，有意思的是，每个类别的方差相同，则分类面是线性的，就是K-means；而如果σ各不相同时，分类面也就是二次函数。

假设$\theta=[\mu_1,\cdots,\mu_K,\sigma^2,\pi_1,\cdots,\pi_K]$，关于$\theta$的MLE为

$$\arg\max_\theta \prod_{j=1}^n p(x_j|\theta)=\arg\max_\theta \prod_{j=1}^n \sum_{i=1}^K P(y_j=i|\theta)p(x_j|y_j=i,\theta)$$

上式难以通过求log求出最小值，因为内部有一个求和符号，所以要引入EM算法求解上式。

EM算法

先考虑$p(x,y)$的MLE：

$$\max_\theta \prod_j p(x_j,y_j|\theta)\propto \sum_j \log p(x_j,y_j|\theta)\approx \sum_{j=1}^n\sum_{i=1}^K p(y_j=i|x_j,\theta^*)\log p(x_j,y_j=i|\theta)$$

建立Q函数，通过迭代求解θ使得Q达到极大值

$$Q(\theta^t|\theta^{t-1})=\sum_{j=1}^n\sum_{i=1}^K p(y_j=i|x_j,\theta^{t-1})\log p(x_j,y_j=i|\theta^t)$$

下面证明这样迭代是收敛的：

利用到一个求解函数极值的技术，如果要求$f(x)$的极小值，先取x_0，再找一个在$f(x_0)$处相切的函数$g(x)$并且是其上界，于是另$x_1\leftarrow \arg\min_x g(x)$（求极大值反之亦然），根据该方法进行迭代即可得到$f(x)$的极小值。

下面推到一个关于θ的MLE十分重要的结论：

$$\begin{aligned}\log p(x|\theta) &= \int_Y q(y)\log p(x|\theta)\,\mathrm{d}y=\int_Y q(y)\log\frac{p(x,y|\theta)}{p(y|x,\theta)}\frac{q(y)}{q(y)}\,\mathrm{d}y\\ &=\int_Y q(y)\log p(x,y|\theta)\,\mathrm{d}y-\int_Y q(y)\log q(y)\,\mathrm{d}y+\int_Y q(y)\log\frac{q(y)}{p(y|x,\theta)}\,\mathrm{d}y\end{aligned}$$

注意到右式第三项正好是p,q的KL散度$\mathrm{KL}(q||p)\geqslant 0$，于是前两项构成$\log p(x|\theta)$的下界，要求极大似然的极大值，第二项与$\theta$无关，所以只需对第一项求即可，也就是

$$\max_\theta \int_Y q(y)\log p(x,y|\theta)\,\mathrm{d}y$$

取$q(y)=p(y|x,\theta)$时，KL散度正好为0，极大似然对应的θ^*可通过迭代求解下式得到

$$\theta \leftarrow \arg\max_{\theta} \int_{Y} p(y|x,\theta) \log p(x,y|\theta) \, \mathrm{d}y$$

这正好就是(19)式内部迭代的内容。

进一步学习：《PRML》第430页到441页内容

变分自编码器

自编码器（Auto-encoder）

和PCA的原理基本类似，从高维降到低维，然后在重新升为会高维，使得重建后的结果和输入的特征尽可能相似。

变分自编码器分为两个网络：

- 编码网络（降维）：$q_{\phi}(z|x) = N(\mu(z;\theta), \sigma^2(z;\theta))$（变分：$q(z|x,\phi)$假的，并且尽可能像真的数据到参数的条件概率分布）
- 解码网络（升维）：$p_{\theta}(x|z) = N(\mu(x;\phi), \sigma^2(x;\phi))$

从变分方程上解释：

$$\log p(x|\theta) = \mathrm{KL}(q(z|x,\phi)||p(z|x,\theta)) + \int q(z|x,\phi) \log \frac{p(x,z|\theta)}{q(z|x,\phi)} \, \mathrm{d}z = \mathrm{KL} + L$$

这里的 $p(z|x,\theta)$ 是数据到隐参数的真实概率分布；L 称为**变分下界**，最大化变分下界，或者最小化 q 等价

$$L = \int q_{\phi} \log \frac{p_{\theta} p(z)}{q_{\phi}} \, \mathrm{d}z = \int q_{\phi} \log \frac{p(z)}{q_{\phi}} \, \mathrm{d}z + \int q_{\phi} \log p_{\theta} \, \mathrm{d}z = -\mathrm{KL}(q_{\phi}||p(z)) + \int q_{\phi}$$

其中 $p(z) \sim N(0,1)$，把前面的 KL 散度认为正则项（能够有效抽取特征的正则），最后一项则是 $\mathbb{E}_{q_{\phi}(z|x)}(\log p(x|Z,\theta))$ 生成图像的期望。

KL散度可以直接计算得到（俩Gauss分布的KL散度），最后一项通过对z的重采样，近似计算（概率重采样）

$\varepsilon_i \sim N(0,1)$，$z_i = \sigma(x,\phi) \cdot \varepsilon + \mu(x,\phi) \sim N(\mu(x,\phi), \sigma^2(x,\phi))$，于是

$$\int q_{\phi}(z|x,\phi) \log p(x|z,\theta) \, \mathrm{d}z \approx \frac{1}{J} \sum_{i=1}^{J} \log p(x|z_i,\theta)$$

于是仍然用梯度下降求解

传统的auto-encoder就是最小化

$$||x - \mu_{\theta}(z)||^2$$

Ada Boost

理论证明：数据权重更新的归一化因子和更新公式：

$$
\begin{aligned}
Z_t &= \sum_{i=1}^{m} D_t(i)\exp(-\alpha_t y_i h_t(x_i)) \\
D_{t+1}(i) &= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t} \\
&= \frac{D_{t-1}(i)\exp(-\alpha_t y_i h_t(x_i) + \alpha_{t-1} y_i h_{t-1}(x_i))}{Z_t Z_{t-1}} \\
&= \cdots \\
&= \frac{D_1(i)\exp(-y_i \sum_{j=1}^{t} \alpha_j h_j(x_i))}{\prod_{j=1}^{t} Z_j} \\
\xRightarrow{\text{对 } i \text{ 求和}} 1 &= \frac{\frac{1}{m}\sum_{i=1}^{m}\exp(-y_i F_t(x_i))}{\prod_{j=1}^{t} Z_j} \\
\Rightarrow \prod_{j=1}^{t} Z_t &= \frac{1}{m}\sum_{i=1}^{m}\exp(-y_i F_t(x_i))
\end{aligned}
$$

上文中红色部分就是 $F_t(x_i)$ 复合分类器，绿色部分就是Ada Boost的损失函数

$$
\begin{aligned}
Z_t &= \left(\sum_{x_i\text{分对}} D_t(i)\right)\exp(-\alpha_t) + \left(\sum_{x_i\text{分错}} D_t(i)\right)\exp(\alpha_t) \\
&= (1-\varepsilon_t)e^{-\alpha_t} + \varepsilon_t e^{\alpha_t} \\
&\text{当}\alpha_t = \frac{1}{2}\log\frac{1-\varepsilon_t}{\varepsilon_t}\text{时，}Z_t\text{取到最小值}
\end{aligned}
$$

其中 $\varepsilon_t = P_{i\sim D_t(i)}[h_t(x^i) \neq y^i] = \sum_{i=1}^{m} D_t(i)\delta(h_t(x_i) \neq y_i)$。