DEEP LEARNING

March 20, 2019

Zhihua Zhang (张志华)
Ruohua Shi (史若画)
Deap Learning Courses in Peking University

Contents

1 Overview 3
1.1 Refrences . . . 3
1.2 Machine Learning . . . 3

2 Chapter 2 Basics 4
2.1 Optimization Target . . . 4
2.2 Basic Models . . . 4
2.3 Chapter 2.4 . . . 7
2.4 Chapter 2.5 Loss Functions Design . . . 11

3 Chapter 3 Kernel 15
3.1 Chapter 3.1 Motivation . . . 15
3.2 Chapter 3.2 . . . 16
3.3 Chapter 3.3 PCA . . . 16

1 Overview

1.1 Refrences

《Numerical Linear Algebra》

《Linear Algebra and Learning from Data》 $https://math.mit.edu/\ gs/learningfromdata/$

《Numerical Optimization》 $http://www.bioinfo.org.cn/\ wangchao/maa/Numerical_Optimizat$

$https://github.com/ShiqinHuo/Numerical-Optimization-Books$

1.2 Machine Learning

1）频率派/贝叶斯派

贝叶斯派的 MCMC 是目前最常用的采样方法，但是其计算问题仍需解决

2）参数化/非参数

参数化是指参数的数量和数值不依赖于数据，是固定的，如一个高斯分布就两个参数。

非参数指参数的数量和数值依赖于数据，如最近邻算法，数据的个数就是参数的个数。

kernel 方法：这种方法可以将参数与非参数方法转化，“主问题-> 对偶问题”就是一种“参数-> 非参数”的方法。

3）生成模型/判别模型

$(x,y)\sim\mathbb{D}$，x 和 y 服从联合分布。—-生成模型

$(x,y),y\sim\mathbb{D}$，x 固定，指关注 y 的分布。—-判别模型

2 Chapter 2 Basics

2.1 Optimization Target

对于一个分类器 classification h, 它的泛化误差 The general error 定义为:

$$Pr(h(x) = y(x)) = \mathbb{E}_{x\in\mathbb{D}}(1_{(h(x)\neq y(x))}) \tag{1}$$

对分类问题 Classification：Input: $x \in R^d$. Output: $y \in \{-1, 1\}$
对回归问题 Regression: Input: $x \in R^d$. Output: $y \in R$
已知一个训练数据序列: $(x_i, y_i), i = 1...n, x_i \in R^d, y_i \in \{-1, 1\}$，它的经验误差 the experience error of h 定义为:

$$\hat{R}(h) = \mathbb{E}_{x\in\mathbb{D}}(1_{(h(x)\neq y(x))}) \tag{2}$$

$$\hat{D} = \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}() \tag{3}$$

为了最小化这个经验误差，自然的得到一个 0-1 loss，它是高维不连续的，在数据量较大的时候很难做离散的梯度下降来优化。所以需要找到一个损失函数来代替，这个损失函数需要满足两点：1）凸函数（Convex）；2）是一个上界函数（Upper bound of the original loss）。

2.2 Basic Models

SVM logistic regression

目标是 $max\{0, 1 - y_i w^T x_i\}$，多层的感知机就是一个嵌套的过程 $f() = f_1(f_2(...))$.

Linear Classification

首先做一个仿射函数 (Affine Function):

$$L_d = \{f_{w,b}, w \in R^d, b \in R\} \tag{4}$$

$$f_{w,b}(x) =< w, x > +b = \frac{1}{n}\sum_{i=1}^{n} w_i x_i + b_i =< \bar{w}, \bar{x} > \doteq < w, x > \tag{5}$$

其中 $\bar{w} = (b, w) \in R^{d+1}$; $\bar{x} = (1, x^T) \in R^{d+1}$. h 是在 L_d 上的映射 $R \to y$. 对回归问题: $h(x) = f_w(x)$，对分类问题：$h(x) = sign(f_w(x))$。

【生成模型】 Generation Model

对 p 维数据 (x,y)，$P(x,y)=(P(y)(P(x|y)$，设 y 的先验是伯努利分布，$y\in\{0.1\}$，x 服从正态分布。

$$P(y|\pi)=\pi^{y}(1-\pi)^{-y};\pi\in(0,1) \tag{6}$$

$$P(x|y=0)=\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}exp\{-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\} \tag{7}$$

$$P(x|y=1)=\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}exp\{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\} \tag{8}$$

接下来做极大似然估计 $min-\sum_{i=1}^{n}logP(y_i)P(x_i|y_i)$. 假设已知求得的所有参数极大似然估计 $\hat{\theta}_{MLE}$. 那么去求 y 属于那一类其实就是求对 $y=0/1$ 的后验概率，如果大于 1/2 则属于这一类。

$$\begin{aligned}
P(y=1|x,\theta)&=\frac{P(x|y=1)P(y=1|\pi}{P(x|y=1,\theta)P(y=1|\pi)+P(x|y=0,\theta)\cdot P(y=0|\pi)}\\
&=\frac{\pi exp\{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1))\}}{\pi exp\{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\}+(1-\pi)exp\{-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\}}\\
&=\frac{1}{1+\frac{1-\pi}{\pi}exp\{(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)-(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\}}\\
&=\frac{1}{1+exp\{-(\mu_1-\mu_0{}^T\Sigma^{-1}x+\frac{1}{2}(\mu_1-\mu_0)^T\Sigma^{-1}(\mu_1+\mu_0)-log\frac{\pi}{1-\pi}\}}\\
&=\frac{1}{1+exp\{-w^Tx+b\}}\\
w&=\Sigma^{-1}(\mu_1-\mu_0);b=log\frac{\pi}{1-\pi}
\end{aligned} \tag{9}$$

由上可知，$P(y=1|x,\theta)>\frac{1}{2}$ 与 $sign(w^Tx+b)$ 一致。但是参数量减少了很多。

【感知算法】Perception Algorithm

1. Batch Perception 批处理（Mini Batch）

Input: $\{(x_i,y_i)\}$

Initialize: $w^{(0)}=(0,...,0)$

$for \quad t=0,1,...,T$

$\quad if(\exists i,s.t.y_i<w^{(t)},x_i>\leq 0) \quad w^{(t+1)}=w^{(t)}+y_ix_i$

$else \quad output w^{(t)}$

$$\begin{aligned} y_t < w^{(t+1)}, x_t > &= y_i < w^{(t)} + y_t x_t, x_t > \\ &= y_i < w^{(t)}, x_t > + {y_t}^2 < x_t, x_t > \\ &= y_t < w^{(t)}, x_t > + \|x_t\|^2 \end{aligned} \tag{10}$$

2. Online Perception Algorithm

Initialize: $w^{(0)} = (0, ..., 0)$
$for \quad t = 0, 1, ..., T$
$Receive \quad x_t$
Predict $C_r = sign(< w^{(t)}, x_t >)$
$if(y_t < w^{(t)}, x_i > \leq 0) \quad w^{(t+1)} = w^{(t)} + \eta y_t x_t \ \eta \in [0,1]$
$else \quad w^{(t+1)} = w^{(t)}$

现在我们的损失函数 $1_{[y<w,x>\leq 0]}$(Fig.1) 替换成 $max\{0, -y < w, x >\}(max\{, -z\})$(Fig.2)。这时由一个不连续的函数变成连续但不可导的函数。这时的优化目标变成 $min\{y(w) = \frac{1}{n}\sum_{n}^{i=1} max\{0, -y_i < w_i, x_i >\}\}$. 虽然零点处不可导，但是可以用此梯度来代替，参数更新过程是:

$w^{(t+1)} = w^{(t)} + \eta \nabla_w y(w^{(t)})$
$w^{(t+1)} = w^{(t)} - \eta \nabla_w y(w^{(t)}), \quad if \quad < w^{(t)}, x_t > \neq 0$
$w^{(t+1)} = w^{(t)} + \eta y_t x, \quad if \quad < w^{(t)}, x_t >= 0 \ \nabla_w y(w) = -y_t x_t, \quad if \quad y_t(w^{(t)}) < 0$
$\nabla_w y(w) = 0, \qquad if \quad y_t(w^{(t)}) > 0$

如果将拐点右移，如 Fig.3蓝色曲线，优化过程转为 $max\{0, 1 - y < w, x >\}$，是 SVM 的优化过程，也就是到 Hinton Loss。

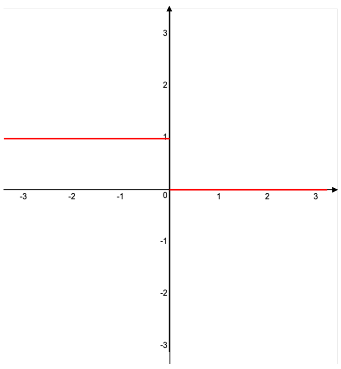

Figure 1: 0-1 Loss

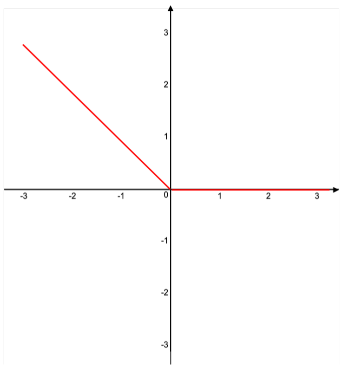

Figure 2: Replace Loss

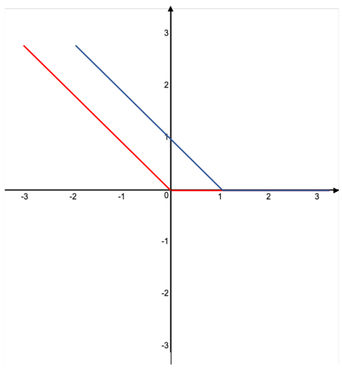

Figure 3: Hinton Loss

2.3 Chapter 2.4

1. **梯度下降法**

定义导数和次导数：

$$
\begin{aligned}
&w^{(t+1)} = w^{(t)} - \eta \nabla g(w^{(t)}) \\
&w^{(t+1)} = \begin{cases} w^{(t)} - \eta \nabla g(w^{(t)}), & if \quad < w^{(t)}, x_t > \neq 0. \\ w^{(t)} + \eta y_t x_t, & if \quad < w^{(t)}, x_t >= 0. \end{cases} \\
&\eta \nabla g(w) = \begin{cases} -yx_t, & if \quad y_t < w, x_t < 0. \\ 0, & if \quad y_t < w, x_t > 0. \end{cases}
\end{aligned} \tag{11}
$$

2. **判别模型**

对 $y \in \{0,1\}$，有：

$$
\begin{aligned}
&P(x,y) = P(y)P(x|y) \\
&P(y|\pi) = \pi^y (1-\pi)^{1-y} \qquad \pi \in (0,1) \\
&P(x|y=0) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp\{-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\} \\
&P(x|y=1) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp\{-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\}
\end{aligned} \tag{12}
$$

模型的参数为 $\pi, \mu_0, \mu_1, \Sigma$，已知数据对为 $\{(x_i, y_i)\}, i = 1, ..., N$,

$$
\begin{aligned}
&\prod_{n=1}^{N} P(x_i, y_i) = \prod_{n=1}^{N} P(y_i)P(x_i|y_i) \\
&P(y=1|x,\theta) = \frac{P(x|y=1)P(y=1|\pi}{P(x|y=1,\theta)P(y=1|\pi) + P(x|y=0,\theta) \cdot P(y=0|\pi)} \\
&min - \sum_{n=1}^{N} logP(y_i)P(x_i|y_i) \qquad \hat{\theta}_{MLE}
\end{aligned} \tag{13}
$$

$$
\begin{aligned}
&= \frac{\pi exp\{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1))\}}{\pi exp\{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\}+(1-\pi)exp\{-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\}} \\
&= \frac{1}{1+\frac{1-\pi}{\pi}exp\{(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)-(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\}} \\
&= \frac{1}{1+exp\{-(\mu_1-\mu_0{}^T\Sigma^{-1}x+\frac{1}{2}(\mu_1-\mu_0)^T\Sigma^{-1}(\mu_1+\mu_0)-log\frac{\pi}{1-\pi}\}} \\
&= \frac{1}{1+exp\{-w^Tx+b\}} \Leftrightarrow sign(w^Tx+b)
\end{aligned} \tag{14}
$$

2. **生成模型**

已知数据 $(x,y), x\in R^p, y\in\{0,1\},\{-1,1\}$

$$
\begin{aligned}
&P(x,y)=P(y)P(x|y) \\
&P(y|\tau)=\tau^\theta(1-\tau)^{1-\theta} \qquad \theta\in(0,1) \\
&P(x|y=0)=\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}exp\{-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\} \\
&P(x|y=1)=\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}exp\{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\}
\end{aligned} \tag{15}
$$

模型的参数为 $\Theta=(\tau,\mu_0,\mu_1,\Sigma)$，

$$
\begin{aligned}
&P(y=1|x)=\frac{P(x|y=1)P(y=1)}{P(x)} \\
&P(x)=\sum_y P(x,y)=P(y=0)P(x|y=0)+P(y=1)P(x|y=1)
\end{aligned} \tag{16}
$$

1) **高斯混合模型，EM 算法**

$$
\begin{aligned}
&P(y=1|x)=\frac{1}{1+exp(-w^Tx-b)} \\
&w\doteq\Sigma^{-1}(\mu_1-\mu_0) \\
&b\doteq\frac{1}{2}(\mu_1-\mu_0)^T\Sigma^{-1}(\mu_1+\mu_0)-log\frac{\tau}{1-\tau}
\end{aligned} \tag{17}
$$

与用 $sign(f_{w,b}=w^Tx+b)$ 效果相同。

2）**贝叶斯**

$$
\begin{aligned}
&P(x,y,\theta)=P(\theta)P(x,y|\theta) \\
&P(\theta|x,y)=\frac{P(\theta)P(x,y|\theta)}{P(x,y)}
\end{aligned} \tag{18}
$$

3) MAP

$$\begin{aligned} \underset{\theta}{argmax} P(\theta|x,y) &\Leftrightarrow \underset{\theta}{argmax} P(\theta)P(x,y|\theta) \\ logP(\theta|x,y) &\Leftrightarrow log(P(\theta)P(x,y|\theta)) \end{aligned} \tag{19}$$

4) Maximum Likelihood Estimation (MLE)

$$\begin{aligned} &\hat{D} = \{(x_n, y_n), n = 1, ..., N\}. \quad x_n \in R^p, \quad y_n \in \{0,1\} \\ &l(\theta, 0) = \sum_{n=1}^{N} log[P(y_n|\tau)P(x_n, y_n, \mu, \Sigma)] \\ &\qquad = \sum_{n=1}^{N} logP(y_n|\tau) \sum_{n=1}^{N} logP(x_n|y_n, \mu, \Sigma)] \\ &\hat{\tau}_{MLE} = \underset{\tau}{argmax} \sum_{n=1}^{N} y_n log\tau + (1-y_n) log(1-\tau) = \frac{1}{N} \sum_{n=1}^{N} y_n \end{aligned} \tag{20}$$

$$\begin{aligned} &\sum_{n=1}^{N} logP(x_n|y_n, \mu_0, \mu_1, \Sigma) \\ &= \sum_{n=1}^{N} log(P(x_n|y_n = 1, \mu_1, \Sigma)^{y_n} P(x_n|y_n = 0, \mu_0, \Sigma)^{1-y_n}) \\ &= \sum_{n=1}^{N} [y_n log(P(x_n|y_n = 1, \mu_1, \Sigma) + (1-y_n) logP(x_n|y_n = 0, \mu_0, \Sigma))] \\ &= \sum_{n=1}^{N} [y_n(1 - \frac{1}{2} log|\Sigma| - \frac{1}{2}(x_n - \mu_1)^T \Sigma^{-1}(x_n - \mu_1) \\ &\quad + (1-y_n)(-\frac{1}{2} log|\Sigma| - \frac{1}{2}(x_n - \mu_0)^T \Sigma^{-1}(x_n - \mu_0))] \\ &= -\frac{1}{2} \sum_{n=1}^{N} [log|\Sigma| + y_n(x_n - \mu_1)^T \Sigma^{-1}(x_n - \mu_1) + (1-y_n)(x_n - \mu_0)^T \Sigma^{-1}(x_n - \mu_0)](*) \end{aligned} \tag{21}$$

对与 μ_1 有关的项 $-\frac{1}{2}\sum_{n=1}^{N} y_n(x_n-\mu_1)^T\Sigma^{-1}(x_n-\mu_1)$ 求导，得 $\sum_{n=1}^{N} y_n\Sigma^{-1}(x_n-\mu_1) = 0 \Rightarrow \sum_{n=1}^{N} y_n(x_n-\mu_1) = 0$，故 μ_1 的估计为：$\hat{\mu_1} = \frac{\sum_{n=1}^{N} y_n x_n}{\sum_{n=1}^{N} y_n}$，同理 $\hat{\mu_2} = \frac{\sum_{n=1}^{N}(1-y_n)x_n}{\sum_{n=1}^{N}(1-y_n)}$.

(*) 式对 Σ 求导，考虑 $log|\Sigma|$ 项，$log|\Sigma| : S^{p\times p} \to R$，$\lim_{t\to 0}\frac{log|\Sigma+tA|-log|\Sigma|}{t} =< A, B >= tr(A^TB)$。则导数为 B，称为 G-导数。

几点性质：

$$\begin{aligned}
X^T\Sigma^{-1}X &= tr(\Sigma^{-1}XX^T) & \Sigma\Sigma^{-1} &= I \\
dX^T\Sigma^{-1}X &= tr(d\Sigma^{-1}XX^T) & d\Sigma\Sigma^{-1}+\Sigma d\Sigma^{-1} &= 0 \\
&= tr(\Sigma^{-1}d\Sigma\Sigma^{-1}XX^T) & d\Sigma^{-1} &= -\Sigma^{-1}d\Sigma\Sigma^{-1} \\
&= tr(\Sigma^{-1}XX^T\Sigma^{-1}d\Sigma) \\
\frac{dX^T\Sigma^{-1}X}{d\Sigma} &= \Sigma^{-1}XX^T\Sigma^{-1}
\end{aligned} \tag{22}$$

5）The Binomial and Bernoulli

对 Binomial 分布，$C_n^k\theta^k(1-\theta)^{n-k}$

对 Bernoulli 分布，$\theta^y(1-\theta)^y$.

已知 $x\in R^p, y_n\in\{1,...,k\}, y\in R^k$，eg. k=3 时，y=(1,0,0),y=(0,1,0),y=(0,0,1) —One-Hot

6）The Multinomial

$y=(y_1,...,y_k)^T$，$Mu(y)=\frac{n!}{y_i!-y_k!}\prod_{k=1}^K {\theta_k}^{y_k}$， $\sum_{k=1}^K\theta_k=1$

Multinoulli

$D(y)=\prod_{k=1}^K {\tau_k}^{y_k}, \tau_k\in(0,1)$

$$\begin{aligned}
P(x|y_k=1) &= \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}exp(-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) \\
P(y_k=1|x) &= \frac{exp({w_k}^Tx)}{\sum_{k=1}^p exp({w_k}^Tx)} \quad ---(**)softmax
\end{aligned} \tag{23}$$

softmax 即找到 $\{<w_k,x>\}$ 的最大值。

7）Gumbel-Max trick

$G-Gumbel(m)$ $P(G\leq y)=exp(-exp(y+m))$，若 G 是 iid 的 Gumbel(0) 分布，则 $(**)=\frac{exp(\phi_m)}{\sum_{n-1}^N exp(\phi_m)}=\underset{i}{argmax}\{G_i+\phi_i\}$ $--$ $softmax$.

8）The Naive Bayes Classifier

$x\in R^p$, $x=(x_1,...,x_p)^Ty$，特征值是离散值，不能用高斯分布。则 $P(x|y)=\sum_{j=1}^p P(y)P(x_j|y)$，$\Sigma=diag(\sigma_1,...,\sigma_p)$。

2.4 Chapter 2.5 Loss Functions Design

对一个 Discriminant Model，我们看 Logistic Regression，考虑输入是 $x \in R^p$，$y \in \{0,1\}$ 是类标，有如下分布：

$$P(y|x) = \mu(x)^y 1 - \mu(x)^{1-y}, \quad \mu \in (0,1) \tag{24}$$

$$\mu(x) = \frac{1}{1 + exp(-w^T x)} \tag{25}$$

现在想要训练 w，假设 $\{(x_n, y_n) : n = 1, ..., N\}$ 是训练数据，可以通过最小化它的极大似然估计来优化：

$$\min_w \quad L(w) = -\sum_{n=1}^{N} y_n log\mu_n + (1 - y_n) log(1 - \mu_n), \quad \mu_n = \mu(x_n) \tag{26}$$

其中 μ_n 对应真是分布，这就是常见的交叉熵，等价于极大似然，也等价于算经验分布与真实分布的 KL。

对于一个数据矩阵 $X = [x_1, ..., x_N]^T{}_{(N\times p)}$，$Y = (y_1, ..., y_N)^T{}_{(1\times N)}$，$\mu = \mu_1, ..., \mu_N{}^T$，设：

对角阵 $D = diag(\mu_1(1-\mu_1), ..., \mu_N(1-\mu_N))$

L 的梯度 $\nabla_w L = X^T(\mu - Y)$

海森阵 $H = \frac{\partial L}{\partial w \partial w^T} = X^T D X \geq 0$，是一个半正定的，也说明 L 关于 w 是凸的。

（这里插入一点矩阵求导的知识，可以用极限近似的方法，转化为求 log 或 trace，如 $f(X), x \in R^{(p\times q)}, f : R^{(p\times q)} \to R$，可以转化为 $\lim_{t\to 0} \frac{f(X+tY)-f(X)}{t} =< Y, C >$ 这样的内积的形式，（G 导数））

1. **一阶近似**

对于一阶近似（梯度迭代），$w^{(t+1)} = w^{(t)} - \eta_t X^T(\mu^{(t)-y})$，其中 η_t 是学习率，满足 $m_{t=0}^{\infty}\eta_t = \infty$，$\sum_{t=0}^{\infty} \eta_t{}^2 < \infty$。令 $L(w)$ 在 $w^{(t)}$ 点做一阶近似：

$$L(w) \approx L(w^{(t)}) \quad + \quad < \nabla_w L(w^{(t)}), w - w^{(t)} > \quad + \quad \frac{1}{\eta} \| w - w^{(t)} \|^2 \tag{27}$$

这是一个用线性函数近似的过程，最后一项是希望控制下一步迭代与这一步相差不要太远（近邻）。有时 w 有特殊的性质，比如当 w 非负时，用平方近邻就不太合适，这时可以考虑 $+\eta KL(w^{(t)} \| w)$

2. **二阶近似**

对于二阶近似，又称牛顿法（Newton-Raphson）

$$w^{(t+1)} = w^{(t)} - (X^T D^{(t)} X)^{-1} X^T (\mu^t - y) \tag{28}$$

$$L(w) \approx L(w^{(t)}) \quad + \quad < \nabla_w L(w^{(t)}), w - w^{(t)} > \quad + \quad \frac{1}{2}(w - w^{(t)})^T H (w - w^{(t)}) \tag{29}$$

但二阶近似方法需对 $D_{(p\times p)}$ 求逆，已知 $rank(X^T DX) \le min\{p,N\}, p \le N$，对于少量数据或不满秩的情况，要满足可逆很难，所以需要加入一个正则化项，其中 λ 为超参数，交叉验证选取，即：

$$L(w) = \sum_{n=1}^{N} (y_n log\mu_n + (1-y_n) log(1-\mu_n)) + \lambda \| w \|^2 \tag{30}$$

用贝叶斯方法来解释：

$$\begin{aligned} \underset{w}{argmin} L(w) &\Leftrightarrow \underset{w}{argmax} \sum_{n=1}^{N} (y_n log\mu_n + (1-y_n) log(1-\mu_n)) - \lambda \| w \|^2 \\ &\Leftrightarrow \underset{w}{argmax} \quad exp(\sum_{n=1}^{N} (y_n log\mu_n + (1-y_n) log(1-\mu_n))) \cdot exp(-\lambda \| w \|^2) \\ &\Leftrightarrow \underset{w}{argmax} \prod_{n=1}^{N} {\mu_n}^{y_n} (1-\mu_n)^{(1-y_n)} \frac{\lambda^{1/2}}{\sqrt{2\pi}^p} exp(-\lambda \| w \|^2) \end{aligned} \tag{31}$$

也就是等价于最大后验，$p(y|w)p(w) \propto p(w|y)$。假设给 w 一个均匀分布 $[-m,m]$，相当于一个常数，也就是无信息先验（noninformation prior）。

下面考虑其中几个矩阵的运算，

$$\begin{aligned} &H = X^T DX + \lambda I_p \\ &\nabla_w L = X^T (\mu - y) + \lambda w \\ &w^{(t+1)} = w^{(t)} - (X^T DX + \lambda I_p)^{-1} [X^T (\mu^t - y) + \lambda w^{(t)}] \end{aligned} \tag{32}$$

这时 H 就是严格大于一个数，I_p 是一个 $p \times p$ 的对称阵，现在就变成一个强凸的函数了。另外，$(X^T DX + \lambda I_p)$ 是一个 $p \times p$ 的矩阵，计算复杂度为 $O(p^3)$。假设 $p > N$，

有 $(Y^TY+\lambda I_p)^{-1}Y^T = Y^T(YY^T+\lambda I_N)^{-1}$，通过变换把 $p\times p$ 转换为 $N\times N$。

$$\begin{aligned}
&(X^TDX+\lambda I_p)^{-1}((X^TDX+\lambda I_p)w - X^T(\mu-y)-\lambda w)\\
=&(X^TDX+\lambda I_p)^{-1}(X^TDXw - X^T(\mu-y))\\
=&(X^TDX+\lambda I_p)^{-1}X^TD(Xw-D^{-1}(\mu-y))\\
=&((D^{1/2}X)^TD^{1/2}+\lambda I_p)^{-1}(D^{1/2}X)^TD^{1/2}\\
=&X^TD^{1/2}(D^{1/2}XX^TD^{1/2}+\lambda I_N)^{-1}D^{1/2}\\
=&X^T(XX^T+\lambda D^{-1})^{-1}
\end{aligned} \tag{33}$$

虽然这是一个简化参数的方法，但参数量还是过大，导致神经网络很少用二阶方法。

3. **对于一阶方法的加速**

现在考虑一般问题 $\underset{w}{min} f(w)$，使用组合的思想。

① Player's heavy ball

$$\begin{aligned}
w^{(t+1)} &= w^{(t)} - \eta\nabla f(w^{(t)}) + \beta(w^{(t)} - w^{(t-1)})\\
&= (1+\beta)w^{(t)} - \beta w^{(t-1)} - \eta\nabla f(w^{(t)})
\end{aligned} \tag{34}$$

其中第一行的 $w^{(t)} - w^{(t-1)}$ 就是常说的动量（momentum），第二行的 $(1+\beta)w^{(t)} - \beta w^{(t-1)}$ 就是外插的过程。

② Nesterous “蛙跳”

$$\begin{aligned}
&y^{(t+1)} = w^{(t)} - \eta\nabla f(w^{(t+1)})\\
&w^{(t+1)} = y^{(t+1)} + \beta(y^{(t+1)} - y^{(t)})\\
&w^{(t+1)} = (1+\beta)w^{(t)} - \beta w^{(t-1)} - \eta((1+\beta)\nabla f(w^{(t)}) - \beta\nabla f(w^{(t+1)}))
\end{aligned} \tag{35}$$

这里的 β 可以不 >0，若 $\beta<0$，则变成内插形式（非凸）。在强凸的情况下可以将时间复杂度从 $O(\frac{1}{\sqrt{T}})$ 变为 $O(\frac{1}{T})$。

4. **隐变量处理**

还是考虑开头讲的 $\mu(x) = \frac{1}{1+exp(-w^Tx)}$，这时一个可求导的 Logistic link。它的 Probit link:

$$P(y=1|x,w) = \mu(x) = \Phi(w^Tx) = \int_{-\infty}^{w^Tx} \frac{exp(1-\frac{1}{2}t^2)}{(2\pi)^{1/2}}dt \tag{36}$$

一般用隐变量处理。设 latent variable $z = w^T x + \epsilon, \epsilon \in N(0,1)$

$$P(y=1|z,w) = \begin{cases} 1, & z > 0. \\ 0, & z \le 0. \end{cases}, \quad and \quad P(y=1|z,w) = 1 - P(y=0|z,w) \tag{37}$$

这时把 z 边界掉，有：

$$\begin{aligned} P(y=1|z,w) &= P(y=1|z>0,w)\cdot P(z>0|w) + P(y=1|z<0,w)\cdot P(z<0|w) \\ &= P(z>0|w) \\ &= Pr\{\varepsilon > -w^T X\} \\ &= \int_{-w^T X}^{+\infty} \frac{1}{(2\pi)^{1/2}} exp(-\frac{1}{2}t^2)dt \\ &= \int_{-\infty}^{w^T X} \frac{1}{(2\pi)^{1/2}} exp(-\frac{1}{2}t^2)dt \\ &= \Phi(w^T X) \end{aligned} \tag{38}$$

现在回顾一下这个问题，本质上是解决 0-1 问题，想找一个凸的上界函数，已知一个变换：

$$\mu(x)^y(1-\mu(x))^{1-y} = (\frac{1}{1+exp(-w^T X)})^y (\frac{exp(-w^T X)}{1+exp(-w^T X)})^{1-y} \tag{39}$$

那么观察式子：

$$\begin{aligned} -logP(y|x) &= ylog(1+exp(-w^T X)) + (1-y)w^T X + (1-y)log(1+exp(-w^T X)) \\ &= log(1+exp(-w^T X)) + (1-y)w^T X \\ &= \begin{cases} log(1+exp(w^T X)), & y = 0. \\ log(1+exp(-w^T X)), & y = 1. \end{cases} \end{aligned} \tag{40}$$

若 $y \in \{-1,1\}$，则上式 $= log(1+exp(-yw^T X))$。由于它是大于 0 的，是 0-1 问题的一个上界函数，已知 $\beta > 0$，若：

$$\lim_{\beta\to\infty} \frac{1}{\beta} log(1+exp(-\beta y w^T X)) = \begin{cases} 0, & yw^T X = 0. \\ 0, & yw^T X > 0. \\ -yw^T X, & yw^T X < 0. \end{cases} \tag{41}$$

这时候量化为 SVM。

若

$$\lim_{\beta\to\infty}\frac{1}{\beta}log(1+exp(\beta(-yw^TX)))=\begin{cases}0, & yw^TX=0.\\ 0, & yw^TX>0.\\ yw^TX, & yw^TX<0.\end{cases} \tag{42}$$

这时候退化为 ReLU。

若记 $z=yw^TX$，回到隐变量 z：

$$\max_{w\in(0,1)} wz+\lambda(wlogw+(1-w)log(1-w)) \tag{43}$$

这就是熵（entropy），作用是把一个东西变光滑。

———————— 这里推荐两本书：Leo Breiman 《Statistical Modeling》David A. Freedman 《The Two Cultures Statistical Models Theory and Practice》————————

3 Chapter 3 Kernel

3.1 Chapter 3.1 Motivation

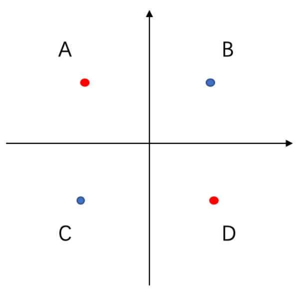

Figure 4: XOR

前面讨论了生成模型（对数据有假设）和判别模型（计算）。都是在平面内线性可分的，考虑 Fig.4中的情况：若想要把 AD 分为一类，BC 分为一类，显然在这个空间线性是不可分的饿，这时候就需要引入一个核方法（Kernel）。这种思想是用最简单的方法达到线性结果。

也就是从输入空间 $\underrightarrow{\phi}$ 特征空间做一个非线性映射，$\phi R^p \to R^r, \quad r \gg p$。让数据在跟高维的空间线性可分。$\phi(x)$ 作为特征，做一个线性模型。神经网络的做法是直接近似 ϕ，而核函数的方法是直接定义在这个特征空间里的内积形式，也就是只关心空间的内积性质，具体是什么空间，有什么其他其他性质并不关心。

定义 $<\phi(x_1),\phi(x_2)>=k(x_1,x_2)$ 是两个数据点 x_1,x_2 的核函数。这个核函数的映射过程 $k:R^p\times R^p\to R$ 必须满足两点性质：

1）$k(x,y)=k(y,x)$ 对称性 2）$k(x,y)\geq 0$ 半正定性

常用的两种核函数是：

1）多项式核：$k(x,y)=1+<x,y>^d$

2）高斯核：$k(x,y) = exp(-\frac{\|x-y\|^2}{2\sigma^2})$ （$\phi(x) \in R^{\infty}$ ，因为可以拓展到无限维所以常用）

例 1： 设 $x, y \in R^2, d = 2, x = (x_1, x_2)^T, y = (y_1, y_2)^T$。它的多项式核将 2 维扩展到 6 维：

$$
\begin{aligned}
k(x,y) &= (1 + x_1 y_1 + x_2 y_2)^2 \\
&= x_1 {y_1}^2 + x_2 {y_2}^2 + 1 + 2x_1 x_2 y_1 y_2 + 2x_1 y_1 + 2x_2 y_2 \\
&= (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, {x_1}^2, {x_2}^2)(1, \sqrt{2}y_1, \sqrt{2}y_2, \sqrt{2}y_1 y_2, {y_1}^2, {y_2}^2)^T \\
&=< \phi(x), \phi(y) >
\end{aligned} \tag{1}
$$

对于一个线性模型：$\sum_{n=1}^{N} log(1 + exp(-y_n {w_n}^T x_n)) + \lambda \| w \|^2$，

设有 $\phi(x) \Rightarrow \sum_{n=1}^{N} log(1 + exp(-y_n, \quad \phi(x_n) >)) + \lambda \| w \|^2$，但是 $\phi(x)$ 不知道，只知道 $k(x_1, x_2)$。

例 2： 设 $x_1, x_2 \in R$，对于一个高斯核

$$
\begin{aligned}
k(x_1, x_2) &= exp(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}) \\
&= exp(-\frac{\|x_1\|^2 + \|x_2\|^2 - 2 < x_1, x_2 >}{2\sigma^2}) \\
&= exp(-\frac{\|x_1\|^2}{2\sigma^2}) exp(-\frac{\|x_2\|^2}{2\sigma^2}) exp(\frac{< x_1, x_2 >}{\sigma^2})
\end{aligned} \tag{2}
$$

$$
exp(\frac{< x_1, x_2 >}{\sigma^2}) = \sum_{k=0}^{\infty} \frac{1}{k!} (\frac{< x_1, x_2 >}{\sigma^2})^k = (x_1, {x_1}^2, ...)(x_2, {x_2}^2, ...)^T \tag{3}
$$

这样高斯核就把二维数据扩展到无限维。

3.2 Chapter 3.2

3.3 Chapter 3.3 PCA