Chunqi SHI & Jonah BU

66天写的逻辑回归

The Logistic Regression Written in 66 Days

June 8, 2016

SYZYGY

Preface

逻辑回归号称实业界用的最广的机器学习算法之一。本文试图分成两大部分进行表述:

1. 知篇：知为本，知则有亲，有亲可久。
   - 第I部分: 逻辑回归简介
   - 第II部分: 广义线性模型
   - 第III部分: 最小结构风险
   - 第IV部分: Softmax回归
   - 第V部分: 最大熵模型
   - 第VI部分: 利用熵来解释GLM,MaxEnt,MLE
   - 第VII部分: 逻辑PCA扩展
2. 用篇：用为体，用而有功，有功则大。
   - 第??部分: 对比常见算法(SVM,NB,LDA)
   - 第??部分: 优化求解
   - 第??部分: 并行架构

我们计划先发布知篇，在收集反馈修改后再发布用篇。从计划动笔到现在我们一起伏案二个多月，因此这个版本我们命名为《66天写の逻辑回归》。

chunqi.shi@hotmail.com; Jonah.Bu@outlook.com

请您给出意见和建议：

- 修改建议：WIKI
- 问题汇报：ISSUES

Contents

主要符号表

符号	含义
x	标量
$\boldsymbol{x}$	向量
$\mathbf{x}$	变量集合
$\mathbf{A}$	矩阵
$\mathcal{X}$	样本空间或状态空间
$\mathcal{D}$	概率分布
σ^2	方差
Σ	协方差
D	数据样本(数据集)
$\mathcal{H}$	假设空间
H	假设集
$\mathfrak{L}$	学习算法
$P(\cdot)$, $P(\cdot\|\cdot)$	概率质量函数，条件概率质量函数
$p(\cdot)$, $p(\cdot\|\cdot)$	概率密度函数，条件概率密度函数
$\mathbb{E}_{\cdot\sim\mathcal{D}}[f(\cdot)]$	函数$f(\cdot)$在分布$\mathcal{D}$下的数学期望
$Var_{\cdot\sim\mathcal{D}}[f(\cdot)]$	函数$f(\cdot)$在分布$\mathcal{D}$下的方差
$L(\boldsymbol{\theta})$	损失函数
$\mathcal{L}(\boldsymbol{\theta}\|\boldsymbol{x}) = P(\boldsymbol{x}\|\boldsymbol{\theta})$	似然(Likelihood)函数
$\ell(\boldsymbol{\theta})$	Log似然函数
$\mathbb{I}(A)$ or $\mathbb{I}_A(\boldsymbol{x})$	指示函数
$\|\cdot\|_p$	L_p范数，p缺省时为L_2范数
∇	向量的一次导数
∇^2	二次导数的Hessian矩阵
$\mathbf{H} \succeq 0$	矩阵$\mathbf{H}$半正定(PSD)
$\mathbf{A} \succ 0$	矩阵$\mathbf{A}$正定(PD)
$\mathbb{H}(X)$ or $\mathbb{H}(p)$	数据集X概率分布$p(X)$的熵
$\mathbb{I}(X;Y)$	数据集X与Y的互信息
$\mathbb{KL}(p\|q)$	概率分布p与q的KL散度
$\arg\min$	$\arg\min_x f(x) = \{x \mid f(x) = \min_{x'} f(x')\}$
$\arg\max$	$\arg\max_x f(x) = \{x \mid f(x) = \max_{x'} f(x')\}$

Part I
逻辑回归(LR)简介

Chapter 1
Logisitic回归

1.1 对二值条件概率建模

首先简单的回忆一下每个人都熟知的线性回归。线性回归做了一件什么事情呢？线性回归估算的是一个连续变量的条件期望(Conditional Expectation):

$$\mathbb{E}(Y) = \boldsymbol{\theta}^T \boldsymbol{x} + b$$

那如果我们感兴趣的对象不再是一个连续的数值，且只有二值化的输出时，例如气象中心预测明天是否下雨，医生预测患者会不会发病，大学生评估自己是否会挂科等等，我们该如何对其进行建模和分析呢?

这种问题通常被称为分类问题(Classification)。一种解决方案是制定一条条规则(Rule)，例如决策树或者知识库，然后把输入数据与规则进行比对，经过若干次判断之后得到一个确定的结果。然而现实世界的经验告诉我们，对于大多数情况，完全相同的条件并不一定能够导致完全相同的结果，也许是因为噪声的存在，也许是因为条件的描述还不够准确，也许是因为事情本身就是随机的。因此，我们希望在给出预测结果的同时还能给出一个该结果发生的概率，比如你挂科的概率达到了90%。

现在我们期望的输出是，给定输入X之后Y发生的概率$P(Y|X)$。现在我们约定，Y的取值有两类(Bi-Classe)，“1”和“0”。而$P(Y = 1) = E(Y)$。定义了输入和输出，剩下的就是寻找一个回归函数(Regression Function)来进行拟合了。拟合出来的结果就是我们想要的条件概率函数(Conditional Probability Function)。

令$P(Y = 1|X = \boldsymbol{x}) = P(\boldsymbol{x}; \boldsymbol{\theta})$，其中$P$是以$\boldsymbol{\theta}$为参数的函数。假设样本$\boldsymbol{x}_i$之间相互独立，则似然函数(*likelihood function*)为：

$$\prod_{i=1}^{n} P(Y = y_i|X = \boldsymbol{x}_i) = \prod_{i=1}^{n} P(\boldsymbol{x}_i; \boldsymbol{\theta})^{y_i}(1 - P(\boldsymbol{x}_i; \boldsymbol{\theta}))^{1-y_i} \tag{1.1}$$

观察上式会发现，如果所有的样本$\boldsymbol{x}_i$都相同，即每次试验的条件都相同时，每次的概率质量函数$P(\boldsymbol{x}; \boldsymbol{\theta})$都为同一个值$p$，则likelihood 变为：

$$\prod_{i=1}^{n} p^{y_i}(1-p)^{1-y_i}$$

这是一个伯努利试验(Bernoulli Trial)结果的似然函数。

回过头来观察$P(\boldsymbol{x};\boldsymbol{\theta})$，直观上我们对它有如下要求：相同的样本$\boldsymbol{x}_i$下$Y$的概率应该相同；相近的样本$\boldsymbol{x}_i$ 下Y的概率应该相近。找到一个合适的P函数之后我们可以通过极大似然估计(*maximum likelihood estimation*)来估计参数$\boldsymbol{\theta}$。

1.2 Logistic回归

现在我们已经定义了二值输出Y，我们希望建立一个关于观察样本$\boldsymbol{x}$的函数，该函数的输出是Y的条件概率$P(Y=1|X=\boldsymbol{x})$，而建模的方法是**线性回归**。

样本$\boldsymbol{x}$的线性函数(linear function)的值域是$(-\infty,+\infty)$，而我们期望的输出p是一个概率，其值域是$[0,1]$，二者不匹配，因此我们需要对p进行一下变换，才能继续使用线性回归。那$\log P$呢？log函数在$[0,1]$上的值域是$[-\infty,0]$，还是有一半不匹配。log函数在$[0,+\infty)$上的值域才是我们需要的$(-\infty,+\infty)$，那我们考虑对P进行一点点变换，使得变换之后的函数值域为$[0,+\infty)$，而我们发现$\frac{P}{1-P}$ 满足这个要求。好吧，最终我们找到的变换是$\log\frac{P}{1-P}$(该变换称为logistic或logit变换)，其值域是$(-\infty,+\infty)$。现在可以使用我们熟悉的线性回归进行拟合了。这种对线性回归的输出进行logistic变换的回归称为**logistic回归**。

于是有：

$$\log\frac{P(\boldsymbol{x};\boldsymbol{\theta})}{1-P(\boldsymbol{x};\boldsymbol{\theta})}=\boldsymbol{\theta}\boldsymbol{x} \tag{1.2}$$

求解$P(\boldsymbol{x};\boldsymbol{\theta})$可得：

$$P(\boldsymbol{x};\boldsymbol{\theta})=\frac{1}{1+e^{-(\boldsymbol{\theta}\boldsymbol{x})}} \tag{1.3}$$

这就是我们所熟知的logistic回归的形式。

注意，$\frac{P}{1-P}$正好是odds的定义，而$\log\frac{P}{1-P}$就是log odds。odds的意义我们将在附录中讨论。

下面的任务就是使用上一节最后提到的极大似然估计进行求解。

利用(1.1)式得到likelihood：

$$L(\theta)=\prod_{i=1}^{n} P(\boldsymbol{x}_i;\boldsymbol{\theta})^{y_i}(1-P(\boldsymbol{x}_i;\boldsymbol{\theta}))^{1-y_i} \tag{1.4}$$

对likelihood取log，把乘积变成求和，则log-likelihood为：

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} y_i \log P(\boldsymbol{x}_i;\boldsymbol{\theta}) + (1-y_i)\log(1-P(\boldsymbol{x}_i;\boldsymbol{\theta})) \tag{1.5}$$

$$= \sum_{i=1}^{n} \log(1-P(\boldsymbol{x}_i;\boldsymbol{\theta})) + \sum_{i=1}^{n} y_i \log \frac{P(\boldsymbol{x}_i;\boldsymbol{\theta})}{1-P(\boldsymbol{x}_i;\boldsymbol{\theta})} \tag{1.6}$$

$$= \sum_{i=1}^{n} \log(1-P(\boldsymbol{x}_i;\boldsymbol{\theta})) + \sum_{i=1}^{n} y_i \boldsymbol{\theta}\boldsymbol{x}_i \tag{1.7}$$

$$= \sum_{i=1}^{n} -\log(1+e^{\boldsymbol{\theta}\boldsymbol{x}_i}) + \sum_{i=1}^{n} y_i \boldsymbol{\theta}\boldsymbol{x}_i \tag{1.8}$$

(1.6)到(1.7)使用了公式(1.2)$\log \frac{P(\boldsymbol{x};\boldsymbol{\theta})}{1-P(\boldsymbol{x};\boldsymbol{\theta})} = \boldsymbol{\theta}\boldsymbol{x}$，(1.7)到(1.8)使用了公式(1.3)$P(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{1+e^{-(\boldsymbol{\theta}\boldsymbol{x})}}$。

我们希望求得likelihood的最大值，因此将log-likelihood对参数$\boldsymbol{\theta}$的每一个分量θ_i求导，并令其等于0:

$$\frac{\partial \ell}{\partial \theta_i} = -\sum_{i=1}^{n} \frac{1}{1+e^{\boldsymbol{\theta}\boldsymbol{x}_i}} e^{\boldsymbol{\theta}\boldsymbol{x}_i} \boldsymbol{x}_{ij} + \sum_{i=1}^{n} y_i \boldsymbol{x}_{ij} \tag{1.9}$$

$$= \sum_{i=1}^{n} (y_i - P(\boldsymbol{x}_i;\boldsymbol{\theta}))\boldsymbol{x}_{ij} \tag{1.10}$$

其中$\boldsymbol{x}_{ij}$表示样本$\boldsymbol{x}_i$的第j个分量。

令(1.10)等于0得到的方程没有解析解(闭合解，closed-form solution)，我们可以使用梯度下降(*Gradient Descent*)或牛顿法(*Newton Method*)等算法进行数值求解，该部分内容将在以后进行介绍。

Appendices

Appendix A
附录

A.1 Sigmoid函数, Logistic函数和Logit函数

*A.1.1 Sigmoid*函数

Sigmoid函数又称为S函数，是一种最常用的激发函数(Activation function)。其他常见的激发函数包括阶跃函数(Step Function)，TanH函数，修正线性单元(Rectified Linear Unit, ReLU)。而Sigmoid函数应用之广，又被称为软阶跃函数(Soft Step Function)

$$S(t) = \frac{1}{1 + e^{-t}}.$$

它一个很大优点是连续可导，并且导数形式非常好，可以用它本身的二次函数来表示。

$$S'(t) = S(t)(1 - S(t)).$$

*A.1.2 Logistic*函数

逻辑函数(Logistic Function)，有称为逻辑曲线值域$[0, L]$上，k为陡度(Steepness)，以x_0为中值点(Midpoint) 的S(Sigmoid)曲线。所以直观上来说，逻辑函数是Sigmoid函数在值域，陡度和中值点上的一个更为一般的形式。

$$f(x) = \frac{L}{1 + \mathrm{e}^{-k(x-x_0)}}$$

而把$k = 1, x_0 = 0, L = 1$的逻辑函数称为标准逻辑函数。所以标准逻辑函数(Standard Logistic Function)就是Sigmoid函数。

*A.1.3 Logit*函数

Logit函数是对数赔率(Log-odds), 是赔率(odds)的自然对数(Natural Logarithm)形式。直观看上去和逻辑函数没有联系。

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \log(p) - \log(1-p) = -\log\left(\frac{1}{p} - 1\right).$$

但是，其实他们关系非常，Logit函数的反函数就是标准Logistic函数。

$$\text{logistic}(\alpha) = \text{logit}^{-1}(\alpha) = \frac{1}{1+\exp(-\alpha)}$$

在之后一章，我们还会通过Logit函数来解释Logistic回归的由来。

Part II
从广义线性模型(GLM)出发

Chapter 2
广义线性模型

2.1 引入广义线性模型

首先我们利用上一章中从线性回归推出Logisitc回归的思想类比出广义线性回归的思想，然后给出正式的解释。

从线性回归到Logistic回归，我们主要做了一件事情：由于我们希望的输出y的值域与线性回归拟合出来的$\boldsymbol{\theta}^T\boldsymbol{x}$的值域不匹配，于是对$y$进行了一次变换得到关于$y$的函数(记为$g(y)$)，使得$g(y)$的值域为$(-\infty,+\infty)$，这样就可以使用线性回归得到$g(y)=\boldsymbol{\theta}^T\boldsymbol{x}$。

在Logistic回归中，我们面对的是一个两类问题，希望拟合的输出y是给定输入$\boldsymbol{x}$ 后事件发生的概率，即$P(Y=1|X=\boldsymbol{x})$，因此我们选择的变换函数$g(y)$为$\log\frac{y}{1-y}$。那么一个自然的想法是，如果面对的问题不同，希望的输出不同，例如可以是计数(某件事情发生了多少次)，可以是可靠性等等，是否可以选择不同的变换函数$g(y)$？答案是：可以，你需要广义线性模型。

广义线性模型(*Generalized Linear Model*, *GLM*)由三部分组成:

1. 随机成分(*Random Component*)：定义了输出变量y在给定输入变量$\boldsymbol{x}$时的条件分布(Conditional Distribution)。一般情况下，我们接触到的分布都属于指数分布族(*Exponential Families*)，例如高斯分布(*Gaussian* or *Normal*)，二项分布(*Binomial*)，泊松分布(*Poisson*)，伽玛分布(*Gamma*)等。现在GLM已经拓展至多变量指数分布族(*Multivariate Exponential Families*)，非指数分布族(*Non-exponential Families*)，甚至y_i的分布未完全定义的情况。这部分内容将在附录中进行讨论。
2. 线型预测器(*Linear Predictor*)：即线型回归部分

$$\eta_i=\boldsymbol{\theta}^T\boldsymbol{x}_i=\theta_0+\theta_1\boldsymbol{x}_{i1}+\theta_2\boldsymbol{x}_{i2}+\cdots+\theta_m\boldsymbol{x}_{im}$$

   这就是GLM中的线性部分。
3. 链接函数(*Link Function*)：一个光滑(Smooth)且可逆(Invertible)的函数$g(\cdot)$，对期望的输出$\mathbb{E}(Y)$进行变换，使得变换后的$g(\mathbb{E}(Y))$与现行预测器相匹配。记$\mu_i=\mathbb{E}(y_i)$，则

$$g(\mu_i) = \eta_i = \theta_0 + \theta_1 \boldsymbol{x}_{i1} + \theta_2 \boldsymbol{x}_{i2} + \cdots + \theta_m \boldsymbol{x}_{im}$$

在Logistic回归中，我们选取的链接函数即Logistic变换。

所以广义线性模型可以看作是**期望输出变换后的线性模型(Linear Model)**，或者是**期望输出的非线性模型(Non-linear Model)**。

现在回过头来再看Logistic回归。LR中的线性预测器就是线性回归部分$\boldsymbol{\theta}^T\boldsymbol{x}$，链接函数即Logistic函数$\log\frac{y}{1-y}$，那LR中的随机成分是什么呢？我们知道GLM中的随机成分指的是给定输入$\boldsymbol{x}$之后期望输出μ所服从的条件分布。LR的输出μ的意义是给定输入$\boldsymbol{x}$后事件发生的概率，即n次独立重复实验中事件发生的概率，就是二项分布。

现在的问题是，我们已经知道目标期望输出服从二项分布，那使用什么东西来把线型预测器$\boldsymbol{\theta}^T\boldsymbol{x}$拟合出来的结果约束到二项分布中呢？答案就是链接函数。在Logistic回归中，当我们选择了Logistic函数作为链接函数对期望输出进行变换时(或者说选择了Sigmoid函数对线型预测器拟合出来的结果进行变换时)，最终得到的期望输出就会“自然而然”地服从二项分布。所以在GLM看来，正确的顺序是：确定期望输出服从二项分布$\longrightarrow$选择Logistic函数作为链接函数$\longrightarrow$使用线性预测器进行拟合。

到这里，只剩下一个问题了：为什么链接函数可以影响GLM中的随机成分，进而决定了期望输出的条件分布？我们将在下一节中进行推导和解答。

2.2 深入理解广义线性模型

上一节中在介绍GLM三个组成部分中的随机成分时有提到，一般情况下，我们遇到分布大属于指数分布族，例如Logistic回归里的二项分布。下面我们首先正式介绍指数分布族。

统计学里许多重要的分布都属于指数分布族，它们都可以表示成如下通式:

$$p(y;\theta,\phi) = \exp[\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)] \tag{2.1}$$

其中:

- $p(y;\theta,\phi)$是随机变量Y的概率函数(Y离散)或概率密度函数(Y连续)。
- $a(\cdot)$，$b(\cdot)$和$c(\cdot)$是三个函数，不同的分布中这三个函数也不同。
- θ是一个未知参数，称为标准参数(*Canonical Parameter*)，该参数具有关系$\theta = g_c(\mu)$，其中$g_c(\cdot)$是标准链接函数(*Canonical Link Function*)，μ表示随机变量Y的期望$\mathbb{E}(Y)$。
- ϕ称为分散参数(*Dispersion Parameter*)，该参数具有关系$\phi > 0$，在某些分布中已知且固定，在其他分布中未知，需要与标准参数θ一起进行参数估计。

下面介绍指数分布族表示为通式(2.1)后的两个重要性质。一般情况下面对一个分布，我们最关心的两个量是服从该分布的随机变量Y的均值μ和方差$Var(Y)$。关于均值和方差，指数分布族的通式表示具有如下性质:

$$\mu = b'(\theta) \tag{2.2}$$
$$Var(Y) = a(\phi)b''(\theta) \tag{2.3}$$

接下来我们首先简要地证明这两个性质，然后回答上一节所遗留的最后一个问题：**为什么链接函数可以影响GLM中的随机成分，进而决定了期望输出的条件分布?**

首先引入两个概念：动差生成函数(*Moment Generating Function*)和累积量生成函数(*Cumulant Generating Function*)。动差，moment，就是“矩估计”中的“矩”。动差生成函数的定义为：

$$M_X(t) = \mathbb{E}e^{tX} \tag{2.4}$$

将$M_X(t)$对t求导(并交换期望和微分)可以得到：

$$\begin{aligned}
\frac{d}{dt}M_X(t) &= \mathbb{E}Xe^{tX} \\
\frac{d}{dt^2}M_X(t) &= \mathbb{E}X^2e^{tX} \\
&\vdots \\
\frac{d}{dt^k}M_X(t) &= \mathbb{E}X^ke^{tX}
\end{aligned}$$

上面几个式子在$t = 0$点的值分别为：

$$\begin{aligned}
\frac{d}{dt}M_X(0) &= \mathbb{E}X \\
\frac{d}{dt^2}M_X(0) &= \mathbb{E}X^2 \\
&\vdots \\
\frac{d}{dt^k}M_X(0) &= \mathbb{E}X^k
\end{aligned}$$

因此通过(2.4)式可以得到不同阶的矩(动差)，所以称其为动差生成函数。紧接着继续介绍累积量生成函数，该函数定义为动差生成函数取log：

$$K_X(t) = \log M_X(t) \tag{2.5}$$

类似地，令$K_X(t)$对t求导可得：

$$\begin{aligned}
\frac{d}{dt}K_X(t) &= \frac{M'_X(t)}{M_X(t)} \\
\frac{d}{dt^2}K_X(t) &= \frac{M_X(t)M''_X(t) - M'_X(t)^2}{M_X(t)^2}
\end{aligned}$$

其中$M'_X(t) = \frac{d}{dt}M_X(t)$。同样地，令$t = 0$可得：

$$\frac{d}{dt}K_X(0) = \mathbb{E}X = \mu \tag{2.6}$$

$$\frac{d}{dt^2}K_X(0) = \mathbb{E}X^2 - \mathbb{E}^2X = Var(X) \tag{2.7}$$

其中μ为X的均值，$Var(X)$为X的方差。事实上，如果对$K_X(t)$在$t=0$点进行泰勒展开(Taylor Series Expansion)可得：

$$K_X(0) = \sum_{k=0}^{\infty}\frac{1}{k!}\frac{d^k}{dt^k}K_X(0)$$

其中$\frac{d^k}{dt^k}K_X(0)$即为随机变量X的k阶累积量。

有了累积量生成函数之后，结合指数分布族的通式(2.1)以及累积量生成函数定义式(2.5)可以得到指数分布族的累积量生成函数：

$$K(t) = \frac{b(\theta + a(\phi)t) - b(\theta)}{a(\phi)} \tag{2.8}$$

上式对t求导并令$t=0$得到：

$$\begin{aligned}
K'(t) &= \frac{b'(\theta + a(\phi))a(\phi)t}{a(\phi)} = b'(\theta + a(\phi)t)a(\phi) = b'(\theta)\\
K''(t) &= b''(\theta + a(\phi)t)a^2(\phi) = b'(\theta)a(\phi)\\
&\vdots\\
K^{(k)} &= b^{(k)}(\theta)a^k(\phi)
\end{aligned}$$

其中$b^{(k)}(\theta) = \frac{d^k}{d\theta^k}b(\theta)$。再结合式子(2.6)和(2.7) 可以证明式子(2.2)和(2.3)。

由(2.2)式，以及标准链接函数与标准参数θ的关系$\theta = g_c(\mu)$：

$$b'(\theta) = \mu = g_c^{-1}(\theta) \tag{2.9}$$

上式表明$b'(\cdot)$即为标准链接函数$g_c(\cdot)$的逆函数。**该结果表明，链接函数的选择将会直接影响期望输出所服从的分布**。

最后再回过头来看看Logistic回归。我们已经知道期望输出Y服从二项分布，而二项分布写成指数分布族通式为

$$p(y;\theta,\phi) = \exp[\frac{y\theta - \log(1+e^{\theta})}{1/n} + \log\binom{n}{ny}]$$

其中$b(\theta) = \log(1+e^{\theta})$，则标准链接函数$g_c(\mu)$的逆函数为：

$$g_c^{-1}(\theta) = b'(\theta) = \frac{1}{1+e^{-\theta}} = \mu$$

于是二项分布的标准链接函数(即上式的逆函数)为

$$g_c(\mu) = \log \frac{\theta}{1-\theta}$$

该式就是logisitic函数。

所以，从广义线性模型的角度来看，因为期望输出服从二项分布，所以我们选择logisitic函数作为标准链接函数，而这就意味着我们选择了logistic回归。

Appendices

Appendix B
附录

B.1 指数分布族补充

前面我们提到了的概率分布的指数分布族是指(*Natural Form*, *NF*)。还有没有其他形式的描述呢？其实还有更为一般化的形式(*General Form*, *GF*)，和均值自然形式(*Mean Value Natural Form*, *MVNF*)。这里稍微补充一下他们之间的关系。

B.1.1 一般形式*(GF)*

一般指数分布族,

$$p(y;\theta) = \exp[d(\theta)e(y) + g(\theta) + h(y)]$$

在一般形式里面，所有的参数表达式都放在指数函数中，包括三部分

1. 自变量y的函数部分$h(y)$
2. 参数θ的函数部分$g(\theta)$
3. 自变量y与参数θ的函数的乘积$d(\theta)e(y)$

B.1.2 自然形式*(NF)*

指数分布族的自然形式在前面我们提过:

$$p(y;\theta,\phi) = \exp[\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)]$$

这种形式又称为分散指数分布族(Dispersion Exponential Family, DEF)：

1. 累积生成函数(Cumulant Generating Function) $b(\theta)$
2. 标准参数(Canonical Parameter) θ
3. 分散参数(Dispersion Parameter) ϕ

分散指数函数也可以写成λ的形式, λ就称为精确参数(Precision Parameter)：
令

$$\lambda = \frac{1}{a(\phi)} \Rightarrow p(y;\theta,\lambda) = \exp[\lambda\{y\theta - b(\theta)\} + c(y,\lambda)]$$

指数分布族有一些特征前面我们也提到了：

1. 均值: $\mu \equiv \mathbb{E}[Y] = b'(\theta)$
2. 方差: $Var[Y] = b''(\theta)\phi = Var(\mu)\phi$
3. 标准链接函数： $g(x)$函数满足$\theta \equiv g(\mu)$

B.1.3 均值自然形式*(MVNF)*

均值自然形式, 我们把标准参数(θ) 换成数学期望μ的函数$\mu = \tau(\theta)$ 那么$\theta = \tau^{-1}(\mu)$并且$\tau^{-1}(x) = g(x)$:

$$p(y;\mu,\phi) = \exp[\frac{\tau^{-1}(\mu)y - b(\tau^{-1}(\mu))}{a(\phi)} + c(y,\phi)]$$

Part III
结构风险最小(SRM)的视角

Chapter 3
结构风险最小(SRM)

3.1 什么是经验风险最小(ERM)

在统计机器学习框架下，很多有监督学习都可以纳入到经验风险最小(*Empirical Risk Minimization*)。经验风险(ER)一般是由一组样本的损失函数(*Loss Function*)之和或者均值来定义的。经验风险是由Vladimir Vapnik提出来的，他同时也是SVM算法和VC理论的提出者，开创性的把统计机器学习提高到一个新的高度。

$$ER(X, Y, \boldsymbol{\theta}) = \frac{1}{n}\sum_{1}^{n} Loss(\boldsymbol{x}_i, y_i; \boldsymbol{\theta})$$

其中$X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T, Y = (y_1, \ldots, y_n)^T$，是样本集合。而$\boldsymbol{\theta}$是学习模型的参数。

通常，每个样本的损失函数可以有两种定义角度去定义：

- 误差函数(*Error Function*, *ERF*)：

$$\begin{aligned} Loss(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}) &= ERF(f(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i) \\ &= \begin{cases} (f(\boldsymbol{x}_i; \boldsymbol{\theta}) - y_i)^2 & \text{if ERF is SE} \\ |f(\boldsymbol{x}_i; \boldsymbol{\theta}) - y_i| & \text{if ERF is AE} \end{cases} \end{aligned}$$

  常见的ERF有平方误差(Squared Error, SE)和绝对值误差(Absolute Error, AE)。这样对应的经验风险就可以是均方差(MSE)，或者均绝对值差(MAE)。
- 负的Log似然函数(*Negative Log Likelihood*, *NLL*)：

$$Loss(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}) = -\ell(\boldsymbol{\theta}; \boldsymbol{x}_i, y_i) = -\log P(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}).$$

  当然如果对于连续情况下，也可以直接利用概率密度函数来计算似然函数。

$$Loss(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}) = -\ell(\boldsymbol{\theta}; \boldsymbol{x}_i, y_i) = -\log p(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}).$$

那么ERM就可以看成是样本集合的NLL。

$$
\begin{aligned}
ER(X,Y;\boldsymbol{\theta}) &= \frac{1}{n}\sum_{1}^{n} -\ell(\boldsymbol{\theta};\boldsymbol{x}_i,y_i) = \frac{1}{n}\sum_{1}^{n} -\log p(\boldsymbol{x}_i,y_i;\boldsymbol{\theta}) \\
&= -\frac{1}{n}\log\prod_{1}^{n} p(\boldsymbol{x}_i,y_i;\boldsymbol{\theta}) \\
&= -\frac{1}{n}\log p(X,Y;\boldsymbol{\theta})
\end{aligned}
$$

对于这两种方式，内在是由一定的联系的。如果我们定义残差(Residual)为学习模型预测值与真实值之差。

$$
r(\boldsymbol{x}_i,y_i;\boldsymbol{\theta}) = f(\boldsymbol{x}_i|\boldsymbol{\theta}) - y_i
$$

那么，最小平方误差等价于残差符合高斯分布(正态分布)下的最小NLL。而最小绝对值误差等价于残差符合拉普拉斯分布下的最小NNL。具体推理过程就省略了。

$$
\begin{aligned}
& r(\boldsymbol{x}_i,y_i;\boldsymbol{\theta}) \sim N(0,\sigma^2) \Longrightarrow \\
& \arg\min_{\boldsymbol{\theta}} (f(\boldsymbol{x}_i;\boldsymbol{\theta}) - y_i)^2 \Leftrightarrow \arg\min_{\boldsymbol{\theta}} -\log\frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(f(\boldsymbol{x}_i;\boldsymbol{\theta})-y_i)^2}{2\sigma^2}} \\
& r(\boldsymbol{x}_i,y_i;\boldsymbol{\theta}) \sim Laplace(0,b) \Longrightarrow \\
& \arg\min_{\boldsymbol{\theta}} |f(\boldsymbol{x}_i;\boldsymbol{\theta}) - y_i| \Leftrightarrow \arg\min_{\boldsymbol{\theta}} -\log\frac{1}{2\,b} e^{-\frac{|f(\boldsymbol{x}_i;\boldsymbol{\theta})-y_i|}{b}}
\end{aligned}
$$

我们来比较一下损失函数是平方误差和绝对值误差这两种不同情况:

- 平方误差(SE)
  1. 等价于残差符合高斯分布的NLL
  2. 残差较小(<1)的数据分配的权重较小，而残差较大(>1)数据分配权重较大，因此对残差较大项的有抑制效果(参见图3.1)。
  3. 连续光滑，容易求梯度(导数)
- 绝对值误差(AE)
  1. 等价于残差符合拉普拉斯分布的NLL
  2. 对残差较小(<1)和较大(>1)的数据分配权重平均，因此对不同的残差项同等看待(参见图3.1)。
  3. 不光滑，不容易求梯度(因此有人提出了光滑绝对值误差(Smoothed Absolute Error)，是一个分段函数，称为Huber函数)。

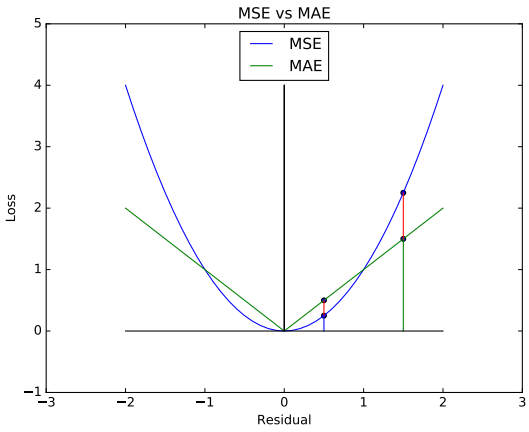


Fig. 3.1 比较平方误差和绝对值误差

3.2 从经验风险到结构风险

我们知道，经验风险就是对训练误差的一个估算，但是我们训练的学习模型，最后是要用来做预测。所以我们更加关注测试误差。更一般的说，我们需要知道的是我们的学习模型对整个输入空间$\mathcal{X} \to \mathcal{Y}$的误差的期望，即真实风险(True Risk)。显然地，真实风险和训练学习模型密切相关。所有学习模型的真实风险最小值称为贝叶斯风险(Bayes Risk)。

一般把训练学习模型的过程称为拟合(Fitting)，拟合过程中，我们根据经验风险来训练模型，但最终目标是希望真实风险最小。通常在拟合完成之后我们会遇到下面两种情况:

- 经验风险不小，且真实风险不小，那么可能是欠拟合(Under Fitting)。
  1. 一般学习模型不够复杂。
  2. VC定理就是用来度量学习模型的拟合能力的一种尺度。
- 经验风险很小，但真实风险不小，那么可能是发生了过拟合(Over Fitting)。
  1. 选用的学习模型过于复杂。
  2. 使用交叉验证(Cross Validation)来进行确认是否过拟合了。
  3. 使用正则化(Reguralization)来限制模型参数。

为了防止过拟合的情况，一般选用拟合能力适合的学习模型，而学习模型的拟合能力可以通过VC定理进行考察。问题是选定了某个具体的学习模型之后，如何防止过拟合的发生呢? 一个常用的方法是模型的参数进行一些限制，即正则化。所以现在我们在拟合的过程中有两个方面需要考虑：**经验风险**和**正则化**。而经验风险和正则化这种联合训练方法称为结构风险最小(*Structure Risk Minimization*, *SRM*)。 因此结构风险的定义为:

Table 3.1 防止过拟合

方法	依据
选择合适拟合能力的学习模型	学习模型的VC维
选择合适参数、结构	正则化
评估是否过拟合	交叉验证

$$SR(X,Y;\boldsymbol{\theta}) = ER(X,Y;\boldsymbol{\theta}) + \lambda \cdot Reguralization(\boldsymbol{\theta})$$

而结构风险最小就是:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \ SR(X,Y;\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}}(ER(X,Y;\boldsymbol{\theta}) + \lambda \cdot Reguralization(\boldsymbol{\theta}))$$

其中λ是正则化系数。如果采用更为一般的描述，把不同参数的函数看成正则化对象，那么对于一个函数簇$f \in \mathcal{F}$。

$$f^* = \arg\min_{f\in\mathcal{F}} \ SR(X,Y;f) = \arg\min_{f\in\mathcal{F}}(ER(X,Y;f) + \lambda \cdot Reguralization(f))$$

这样正则化就可以包含一些非参数(Non-parametric)模型，比如决策树(CART, C4.5)的剪枝(Pruning)等。

那么问题来了，如何确定正则化限制形式，以及如何确定正则化比例系数λ呢?

最常见的正则化项是L_p正则化项，它是L_p空间的模(Norm)，假设$\mathbf{x} = (x_1, x_2, \ldots, x_n)$，那么

$$\|\mathbf{x}\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{\frac{1}{p}}.$$

在通常情况下，不是直接用$L_p = \|\boldsymbol{\theta}\|_p$模，而是用$L_p^p = \|\boldsymbol{\theta}\|_p^p$，即

$$Reguralization(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_p^p = \sum_{k=1}^{|\boldsymbol{\theta}|} |\theta_k|_p$$

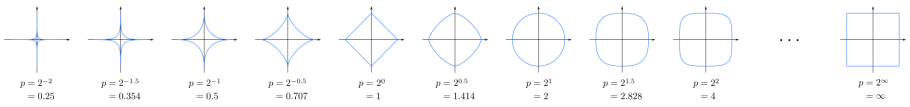


Fig. 3.2 不同p取值的L_p模为1的单位球

本节我们提出了SRM的定义，但并没有深入的解释，下一节我们将给出一个直观的理解。

3.2.1 直观地理解*SRM(*从空间角度*)*

上一节中，我们在ERM上加入了正则化，并最终提出了SRM。本节我们将在此基础上进行一点数学推导，深入理解SRM，特别是正则化项的意义。注意ERM中的损失函数有两种定义方式——误差函数和NLL。本节中所出现的损失函数均是以误差函数来定义的。

首先从拉格朗日乘数法出发：

$$\begin{cases}\min f(x)\\ g(x)\le 0\end{cases}\quad\Leftrightarrow \min_x\max_\lambda \mathcal{L}(x,\lambda)=\min_x\max_\lambda f(x)+\lambda\cdot g(x)$$

根据KKT条件之一: $\lambda\cdot g(x)=0$，因此当$\lambda\neq 0$的时候，有$g(x)=0$。设

$$\lambda^*=\arg\max_\lambda \mathcal{L}(x,\lambda)\neq 0$$

则

$$\begin{cases}\min f(x)\\ g(x)\le 0\end{cases}\quad\Leftrightarrow \min_x f(x)+\lambda^*\cdot g(x)$$

令其中的$f(\boldsymbol{\theta})$和$g(\boldsymbol{\theta})$分别为$ERM(X,Y;\boldsymbol{\theta})$和$R_{L_p}(\boldsymbol{\theta})-C$，且$\lambda^*\neq 0$，即

$$\begin{cases}f(\boldsymbol{\theta})=ERM(X,Y;\boldsymbol{\theta})\\ g(\boldsymbol{\theta})=R_{L_p}(\boldsymbol{\theta})-C\end{cases}$$

于是有

$$\begin{aligned}\arg\min_{\boldsymbol{\theta}} SRM(X,Y;\boldsymbol{\theta})&\Leftrightarrow \arg\min_{\boldsymbol{\theta}}\left(ERM(X,Y;\boldsymbol{\theta})+\lambda^*(R_{L_p}(\boldsymbol{\theta})-C)\right)\\ &\Leftrightarrow\begin{cases}\min_{\boldsymbol{\theta}} ERM(X,Y;\boldsymbol{\theta})\\ R_{L_p}(\boldsymbol{\theta})\le C\end{cases}\end{aligned}$$

观察上式我们发现，正则化相当于存在某个常数C，当$g(\boldsymbol{\theta})=R_{L_p}(\boldsymbol{\theta})-C=0$的时候，恰好$\lambda^*\neq 0$且

$$\lambda^*=\arg\max_\lambda\left(ERM(X,Y;\boldsymbol{\theta})+\lambda(R_{L_p}(\boldsymbol{\theta})-C)\right)$$

直观来说，正则化项的数学意义就是我们限制了$R_{L_p}(\boldsymbol{\theta})=C$，而这个$C$是由正则化系数$\lambda^*$来决定的。因此SRM的意义就是，**在满足正则化项对参数$\boldsymbol{\theta}$的限制条件时，求经验风险$ERM(X,Y;\boldsymbol{\theta})$最小的模型$f(\boldsymbol{\theta})$**。

图3.3中，在$g(x,y)=c$的限制条件下，求$f(x,y)$的最值，拉格朗日数乘法就是找到以$f(x,y)=d_n$定义的等高线上，找到与$g(x,y)=c$相切的点。 那么

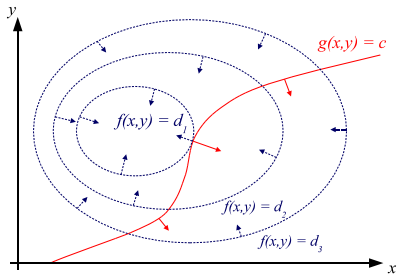


Fig. 3.3 拉格朗日图示求最值，$f(x,y)=d_n$的等高线与$g(x,y)=c$相切

类比到SRM，就是在$R_{L_p}(\boldsymbol{\theta})=C$的线上求与$ERM(X,Y;\boldsymbol{\theta})=d_n$的等高线相切的点(可参见图3.4)。

3.2.2 直观地理解*SRM*(从贝叶斯观点)

前面我们把SRM中的损失函数以误差函数来定义，并从空间的角度对SRM进行了一个直观理解。现在我们从贝叶斯概率分布的观点给出SRM的另一个直观解释。

根据前面对损失函数的两种方式的定义，我们知道损失函数同时可以定义为NLL:

$$\begin{aligned} SRM(X,Y;\boldsymbol{\theta}) &= ERM(X,Y;\boldsymbol{\theta})+\lambda R_{L_p}(\boldsymbol{\theta}) \\ &= -\frac{1}{n}\log p(X,Y;\boldsymbol{\theta})+\lambda R_{L_p}(\boldsymbol{\theta}) \\ &= -\frac{1}{n}\log p(X,Y;\boldsymbol{\theta})+\left(-\frac{1}{n}\log e^{-n\lambda R_{L_p}(\boldsymbol{\theta})}\right) \\ &= -\frac{1}{n}\log p(X,Y;\boldsymbol{\theta})\cdot e^{-n\lambda R_{L_p}(\boldsymbol{\theta})} \end{aligned}$$

其中$0\le e^{-n\lambda R_{L_p}(\boldsymbol{\theta})}\le e^0=1$，令

$$prior(\boldsymbol{\theta})=e^{-n\lambda R_{L_p}(\boldsymbol{\theta})}\in[0,1]$$

则$prior(\boldsymbol{\theta})$看成是参数$\boldsymbol{\theta}$的先验概率(*Prior Probability*)则ERM等价于极大似然估计:

$$\arg\min_{\boldsymbol{\theta}} ERM(X,Y;\boldsymbol{\theta}) \Leftrightarrow \arg\max_{\boldsymbol{\theta}} p(X,Y;\boldsymbol{\theta}) \Leftrightarrow \boldsymbol{\theta}_{MLE(X,Y;\boldsymbol{\theta})}$$

而SRM等价最大后验概率(*Maximum A Posteriori Probability, MAP*):

$$\begin{aligned}\arg\min_{\boldsymbol{\theta}} SRM(X,Y;\boldsymbol{\theta}) &\Leftrightarrow \arg\max_{\boldsymbol{\theta}} p(X,Y;\boldsymbol{\theta})\cdot e^{-n\lambda R_{L_p}(\boldsymbol{\theta})} \\ &= \arg\max_{\boldsymbol{\theta}} p(X,Y;\boldsymbol{\theta})\cdot prior(\boldsymbol{\theta}) \\ &\Leftrightarrow \boldsymbol{\theta}_{MAP(X,Y;\boldsymbol{\theta})}\end{aligned}$$

所以从贝叶斯观点来看，正则化是给模型的参数加了个先验分布的限制条件

$$prior(\boldsymbol{\theta}) = \left(e^{-\lambda R_{L_p}(\boldsymbol{\theta})}\right)^n = \left(e^{-\frac{R_{L_p}(\boldsymbol{\theta})}{\lambda^{-1}}}\right)^n$$

相当于每个样本对应一个$e^{-\lambda R_{L_p}(\boldsymbol{\theta})}$的先验概率。

3.2.3 L_2正则化与L_1正则化

现在我们有了SRM的空间和贝叶斯两种解释，接下来在此基础上我们来更加深入的理解两种最常用的正则化方法——L_1正则化和L_2正则化。

首先从空间的角度来分别讨论L_2和L_1正则化，同样地，在空间解释中ERM的损失函数为误差函数。

$$R_{L_2}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2 = \|\boldsymbol{\theta}\|^2 = \theta_1^2 + \theta_2^2 + \ldots + \theta_m^2$$

假设是两维问题，对于参数$\boldsymbol{\beta} = (\beta_1, \beta_2)$的情况下，那么$L_2$正则化相当于把参数$\boldsymbol{\beta}$的取值范围限定在了以原点为圆心半径为$C$的圆中(图3.4)：

$$R_{L_2}(\boldsymbol{\beta}) = \beta_1^2 + \beta_2^2 = C \Leftrightarrow \beta_1^2 + \beta_2^2 = C \geq 0$$

假设ERM是MSE，那么$ERM(X,Y;\boldsymbol{\beta})$ 在线性回归$Y = \boldsymbol{\beta}^T \cdot X$情况下，也是二次曲线。这种情况称为岭回归(*Ridge Regression*)。

$$ERM(X,Y,\boldsymbol{\beta}) = \frac{1}{n}\sum_1^n (\boldsymbol{\beta}^T \cdot x_i - y_i)^2$$

$$\hat{\boldsymbol{\beta}} = (X^T X + \lambda I)^{-1} X^T Y$$

岭回归有哪些好处呢?

1. 那么根据结果公式，很明显的一个好处是有共线性(Multicollinearity)情况的时候，$X^T X$虽然是半正定的的，却是奇异的(Singular)。而在加上λI之后，结果变得可以求逆了。
2. 根据图3.4, 岭回归会把$\boldsymbol{\beta}$限制在一定的范围呢，使得我们对$\boldsymbol{\beta}$的估计从最佳线性无偏估计(*Best Linear Unbiased Estimate*，*BLUE*) 变成了最小方差估计(*Minimum Variance Unbiased Estimator*，*MVUE*)。从BLUE到MVUE体现了参数估计中牺牲偏差(Bias)来换取更小方差(Variance)

的偏差方差权衡(Bias Variance Tradeoff)的思想。为什么更小的方差有好处呢？方差越小的话，模型对数据较小的扰动能够更稳定，否则数据引入很小的突变点(Outlier)之后，学习的模型会迅速退化，与原模型差异较大。这不是我们想要的结果。但是如果偏差较大，使得模型在较小的方差情况下难以覆盖未训练数据，从而导致泛化误差变大，有点类似欠拟合。我们知道正则化系数λ越大，会导致方差越小，偏差越大。如何找到合适的λ达到偏差和方差平衡目前还没有特别好的办法。

$$\begin{aligned}\mathbb{E}\big[(y-\hat{f})^2\big] &= \mathbb{E}[y^2+\hat{f}^2-2y\hat{f}] \\ &= \mathbb{E}[y^2]+\mathbb{E}[\hat{f}^2]-\mathbb{E}[2y\hat{f}] \\ &= \mathrm{Var}[y]+\mathbb{E}[y]^2+\mathrm{Var}[\hat{f}]+\mathbb{E}[\hat{f}]^2-2f\mathbb{E}[\hat{f}] \\ &= \mathrm{Var}[y]+\mathrm{Var}[\hat{f}]+(f-\mathbb{E}[\hat{f}])^2 \\ &= \mathrm{Var}[y]+\mathrm{Var}[\hat{f}]+\mathbb{E}[f-\hat{f}]^2 \\ &= \sigma^2+\mathrm{Var}[\hat{f}]+\mathrm{Bias}[\hat{f}]^2\end{aligned}$$

其中 $\hat{f}=\mathbf{x}\cdot\hat{\boldsymbol{\beta}}$

3. L_2正则化还有一个好处就是，正则项是二次的，因此连续且二次可导，和MSE同次同构，所以不会增加计算的复杂度。求解原问题可以使用的最优化方法(例如要求二次可导的牛顿法)正则化后都可以继续使用。

接下来是L_1正则化：

$$R_{L_1}(\boldsymbol{\theta})=\|\boldsymbol{\theta}\|_1=|\theta_1|+|\theta_2|+\cdots+|\theta_m|.$$

当损失函数是MSE，而正则化是L_1模的时候，这时候称为Lasso回归(*Least Absolute Shrinkage and Selection Operator Regression*，*Lasso*)最小绝对值缩选算符。

同样假设是两维问题，对于参数$\boldsymbol{\beta}=(\beta_1,\beta_2)$的情况下，那么正则化曲线为第一象限直线，其他象限轴对称，合起来是一个正方形(图3.4)：

$$R_{L_1}(\boldsymbol{\beta})=|\beta_1|+|\beta_2|=C \Leftrightarrow \begin{cases}\beta_1+\beta_2=C \text{ if } \beta_1>0\&\&\beta_2>0\\ \text{轴对称图形 if 除了第一象限}\end{cases}$$

Lasso回归有哪些特点呢?

1. Lasso最大的特征就是，由于$R_{L_1}(\boldsymbol{\beta})=C$是方体，所以在各个轴上的点比较突出，于是$ERM(X,Y;\boldsymbol{\beta})$很容易与顶点相切。于是乎使得这些轴上的$\boldsymbol{\beta}$被优先选中。而这些轴上的点具有的特征在其他方向上是0。对于立体的情况，两个轴上连线的顶点也由于突出容易被切到。这样使得存在一个优先级的切到：从各个轴的顶点，较少的轴的连线，然后再到更多轴的连接面。在这样的优先性选择下使得大部分轴为0。也就是说m个特征向量中尽可能多的维度变成了0，因此在对稀疏性有要求的情况下特别适用。

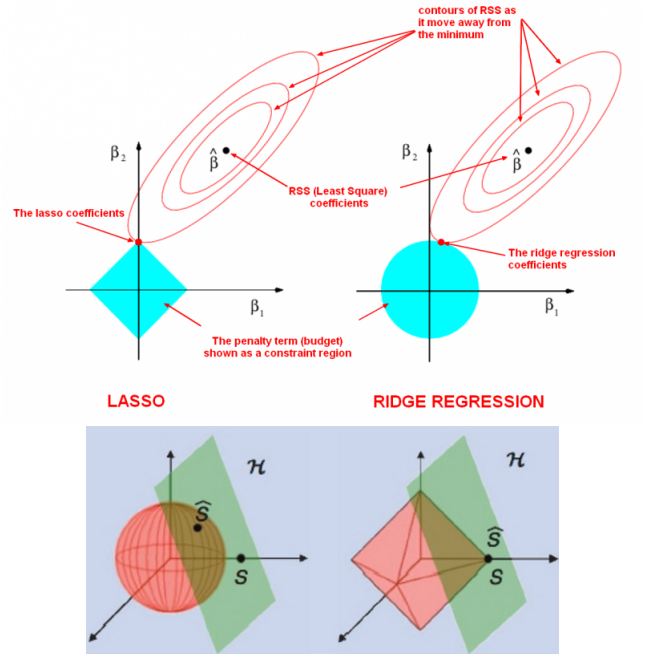


Fig. 3.4 比较L1和L2正则化

2. 同时也正是Lasso的这种内在稀疏性，应此称为特征选择的主要方法之一。在有监督学习的特种选择(Feature Selection)中，相对于过滤(Filter)，包裹(Wrapper)[1]，属于内嵌(Embeded)方法的正则化(主要是Lasso)有着集中了Filter在计算性方面的优势，同时又有Wrapper在自动化好方面的优势。
3. Lasso由于是绝对值函数，所以不能二次求导(牛顿法和共轭梯度法(*Conjugate Gradient*)需要二次求导)，这样在最优化方法的选择上就有了限制，需要利用坐标下坡(Corrdinate Descent)或者近端梯度法(Proximal Gradient)进行求解。

从贝叶斯角度来看，$\boldsymbol{\theta}$的先验分布$e^{-\lambda R_{L_p}(\boldsymbol{\theta})}$在$L_2$和$L_1$的情况下分别对应了高斯分布和拉普拉斯分布。

$$e^{-\lambda\|\boldsymbol{\theta}\|_2} \Leftrightarrow \boldsymbol{\theta} \sim N(0, \sqrt{\frac{1}{2\lambda}})$$
$$e^{-\lambda\|\boldsymbol{\theta}\|_1} \Leftrightarrow \boldsymbol{\theta} \sim Laplace(0, \frac{1}{\lambda})$$

从图3.5上我们可以看到相比较高斯分布，拉普拉斯分布更加尖锐一些。

[1] Wrapper翻译成包裹，跟随周志华的《机器学习》

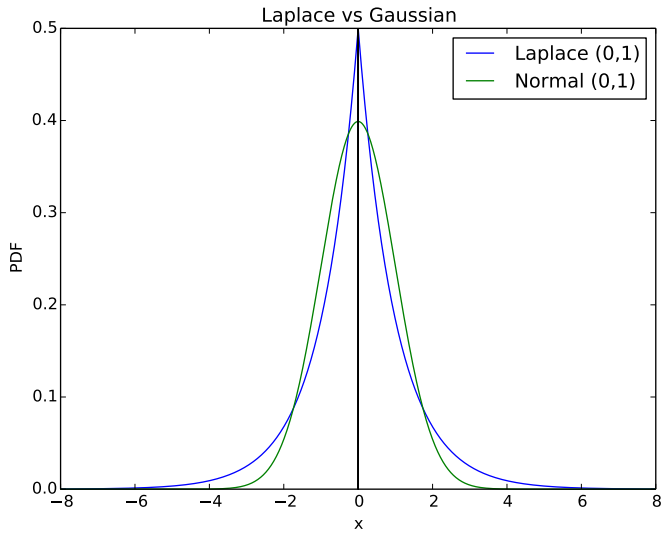


Fig. 3.5 Laplace vs Normal Distribution

既然L_2和L_1分别具有不同的优势，那么把两者结合起来是不是会更好呢？这就是ElasticNet的想法。从它的图形上来看，它既有比较突出的顶点在坐标轴上，又接近圆形限制。这样对共线性和稀疏化都有很大帮助。虽然ElasticNet的效果与Lasso非常相似，但是Lasso对共线性是比较敏感的，会随机选择其中一个特征并把其他特征的系数置0。而ElasticNet则会在两者之间选择一个平衡的点进行加权。

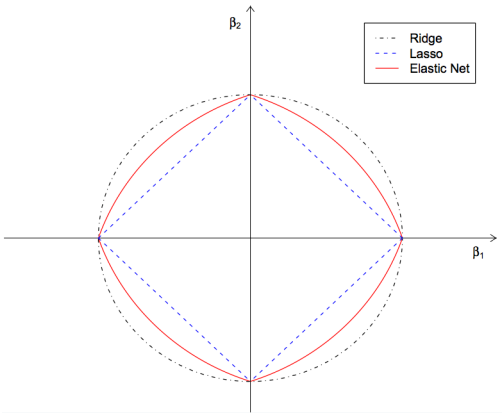


Fig. 3.6 Elastic Net

$$
\begin{aligned}
\lambda R_{EN}(\boldsymbol{\theta}) &= \alpha\lambda_1 R_{L_1}(\boldsymbol{\theta}) + (1-\alpha)\lambda_2 R_{L_2}(\boldsymbol{\theta}) \Leftrightarrow \\
R_{EN}(\boldsymbol{\theta}) &= \alpha\frac{\lambda_1}{\lambda}\|\boldsymbol{\theta}\|_1 + (1-\alpha)\frac{\lambda_2}{\lambda}\|\boldsymbol{\theta}\|_2 \\
&= \frac{\lambda_1}{\lambda}\left(\alpha\|\boldsymbol{\theta}\|_1 + (1-\alpha)\frac{\lambda_2}{\lambda_1}\|\boldsymbol{\theta}\|_2\right) \\
&= \frac{\lambda_1}{\lambda}\left(\alpha\|\boldsymbol{\theta}\|_1 + (1-\alpha)\frac{\lambda_2}{\lambda_1}\|\boldsymbol{\theta}\|_2\right) \\
&= \alpha\|\boldsymbol{\theta}\|_1 + (1-\alpha)\frac{1}{2}\|\boldsymbol{\theta}\|_2
\end{aligned}
$$

当$\alpha = 1$时$R_{EN}(\boldsymbol{\theta}) \sim R_{L_1}(\boldsymbol{\theta})$ 而当$\alpha = 0$时$R_{EN}(\boldsymbol{\theta}) \sim \frac{1}{2}R_{L_2}(\boldsymbol{\theta})$ 所以α又被称为L_1比率。

但是到目前为止还是有个很明显的问题没有解决——λ如何选择。另外ElasticNet里面的L_1比率α又该如何选择?

到目前为止还没有特别完美的办法解决这个问题。当没有好办法的时候只能是做Try-and-Fail。这时候需要利用交叉验证来通过测试误差判断正则化系数的好坏。而对一组λ和一组α进行交叉验证比较然后选择最优的λ和α技术称为Grid Search。是暴力查找优化参数的办法。另外的办法就随机查找(Random Search)。

Try and fail, but never fail to try!

在这里不得不提一下ν-SVM，在SVM里面对松弛变量的L_1正则化如何设置$C(\lambda)$也是一个很大的问题。ν-SVM通过对C在最大Margin中意义探讨，用一个$\nu \in [0,1]$来替代了C，这样需要根据支持向量的个数来设置C的大小，变成了根据支持向量数量的占比ν来设置。但是ν本身也没有好的最优化设置。

Chapter 4
正则化(Regularization)

根据SRM，最小化的目标分成了损失函数和正则化项了。而两者之间是通过比例系数λ来控制的。如果更为泛化的眼光来看，其实哪一项是损失函数，哪一项是正则化也并不需要那么明确。

4.1 分类算法的损失函数

下面我们试图在SRM框架下统一给出损失函数和正则化。

前面我们已经介绍了两种回归的损失函数SE和AE。本材料讨论的LR其实是个分类算法，那么分类算法一般如何计算损失呢?

假设一个两类问题$y_i \in \{0, 1\}$，最直接的方法就是数一下样本分错的数目，即0-1指示函数(*Indicator Function*)。

$$Err = \sum_{1}^{n} \mathbb{I}(y_i \neq f(\boldsymbol{x}_i; \boldsymbol{\theta}))$$

到这里之前，本材料中LR的y_i取值是0或1，这样在把损失函数定义为NLL的时候形式可以很简洁。然而在机器学习领域，分类问题y_i取值通常是-1或1，因为这样一个对称的形式具有优势。这样定义之后，y_i与$f(\boldsymbol{x}_i|\boldsymbol{\theta})$同号即表示样本分类分对了，定义:

$$\mathbb{I}(y_i = f(\boldsymbol{x}_i; \boldsymbol{\theta})) \Leftrightarrow y_i * f(\boldsymbol{x}_i; \boldsymbol{\theta}) = 1 \Leftrightarrow -y_i * f(\boldsymbol{x}_i; \boldsymbol{\theta}) = -1$$
$$\mathbb{I}(y_i \neq f(\boldsymbol{x}_i; \boldsymbol{\theta})) \Leftrightarrow y_i * f(\boldsymbol{x}_i; \boldsymbol{\theta}) = -1 \Leftrightarrow -y_i * f(\boldsymbol{x}_i; \boldsymbol{\theta}) = 1$$

这样损失函数便可以写成单位阶跃函数(*Heaviside Step Function*)的形式:

$$Loss = \mathbb{H}(-y_i * f(\boldsymbol{x}_i; \boldsymbol{\theta})) \Longleftrightarrow \mathbb{H}(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0, \end{cases}$$

一般地，分类问题的损失函数通常表示为$y_i * f(\boldsymbol{x}_i;\boldsymbol{\theta})$形式的函数。

$$Loss(f(\boldsymbol{x}_i;\boldsymbol{\theta}),y_i)=\phi(y_i f(\boldsymbol{x}_i;\boldsymbol{\theta}))\Longleftrightarrow\phi(x)=\begin{cases}0, & x>0,\\ 1, & x\leq 0,\end{cases}=\mathbb{I}(x\leq 0)$$

前面几个章节中LR中的损失函数以NLL的方式来定义，现在我们把y_i的取值由$\{0,1\}$变成了$\{-1,1\}$，损失函数需要相应地发生变化(依然以NLL的方式来定义)。对于样本$(\boldsymbol{x}_i,y_i)$，Logistic函数同样可以理解为$y_i=1$的时候的概率:

$$P(\boldsymbol{x}_i)=\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}_i}}$$

此时的损失函数为

$$\begin{aligned}Loss(\boldsymbol{x}_i,y_i,\boldsymbol{\theta})&=\begin{cases}-\log P(\boldsymbol{x}_i)\text{ if }y_i=1\\ -\log 1-P(\boldsymbol{x}_i)\text{ if }y_i=-1\end{cases}\\&=\begin{cases}-\log\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}_i}}\text{ if }y_i=1\\ -\log 1-\frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}_i}}\text{ if }y_i=-1\end{cases}\\&=\begin{cases}\log 1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}_i}\text{ if }y_i=1\\ -\log\frac{e^{-\boldsymbol{\theta}^T\boldsymbol{x}_i}}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}_i}}\text{ if }y_i=-1\end{cases}\\&=\begin{cases}\log 1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}_i}\text{ if }y_i=1\\ \log\frac{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}_i}}{e^{-\boldsymbol{\theta}^T\boldsymbol{x}_i}}\text{ if }y_i=-1\end{cases}\\&=\begin{cases}\log 1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}_i}\text{ if }y_i=1\\ \log e^{\boldsymbol{\theta}^T\boldsymbol{x}_i}+1\text{ if }y_i=-1\end{cases}\\&=\begin{cases}\log 1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}_i}\text{ if }y_i=1\\ \log 1+e^{\boldsymbol{\theta}^T\boldsymbol{x}_i}\text{ if }y_i=-1\end{cases}\\&=\log 1+e^{-y_i(\boldsymbol{\theta}^T\boldsymbol{x}_i)}\end{aligned}$$

则经验风险为

$$ER(X,Y;\boldsymbol{\theta})=\frac{1}{n}\sum_{1}^{n}\log 1+e^{-y_i(\boldsymbol{\theta}^T\boldsymbol{x}_i)}$$

4.2 分类界限(CM)

事实上，$y_i f(\boldsymbol{x}_i)$(其中$y_i \in \{-1, 1\}$)被称为它被称为分类界限(*Classification Margin*, *CM*)。这个定义来自支持向量机。对于一个二分问题，在线性可分的情况下(图4.1)，分属于两个类别的数据$(\boldsymbol{x}_i, y_i = 1)$ 和$(\boldsymbol{x}_j, y_j = -1)$有如下关系成立:

$$\begin{cases} f(\boldsymbol{x}_i) = \boldsymbol{\theta}^T \boldsymbol{x}_i + b \geq 1 \ \text{ where } y_i = 1 \\ f(\boldsymbol{x}_j) = \boldsymbol{\theta}^T \boldsymbol{x}_j + b \leq -1 \ \text{ where } y_j = -1 \end{cases} \Leftrightarrow yf(\boldsymbol{x}) \geq 1$$

这样就把分类边界定义如下:

$$CM(\boldsymbol{x}, y; \boldsymbol{\theta}) = yf(\boldsymbol{x}; \boldsymbol{\theta})$$

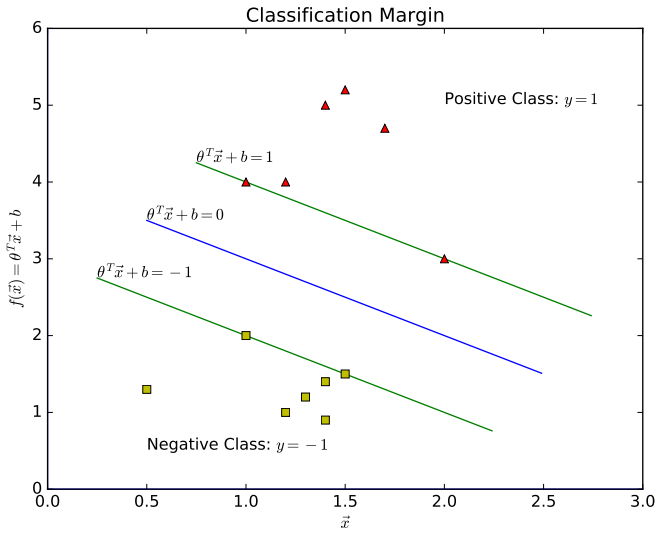


Fig. 4.1 分类界限

从图上可以看出，两条边界(支持向量所在直线)之间$f(\boldsymbol{x})$值相差了2。而$2 = 2yf(\boldsymbol{x})$，所以有些地方也把分类界限定义成$2yf(\boldsymbol{x})$。

SVM的损失函数，以CM来写的话，是铰链损失函数(*Hinge Loss*):

$$Loss(\boldsymbol{x}, y|\boldsymbol{\theta}) = \max(0, 1 - yf(\boldsymbol{x}|\boldsymbol{\theta})) \Leftrightarrow \phi(x) = \max(0, 1 - x)$$

为了让这些不同的损失函数在$CM = 0$的时候有相同的取值，对于Logistic Loss通常做一个归一化处理:

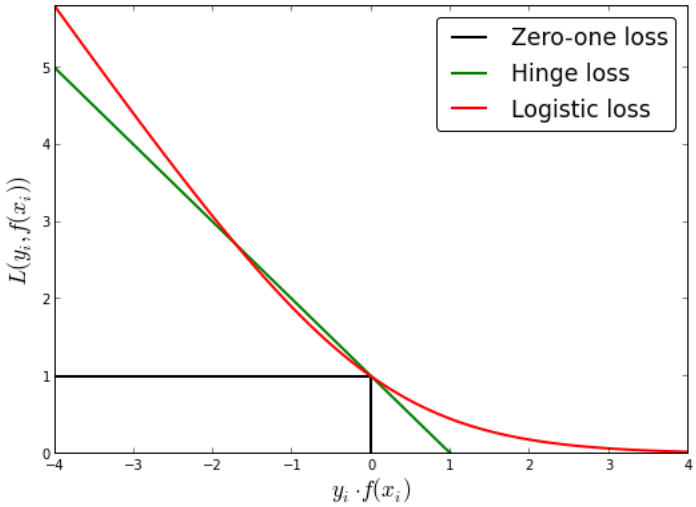


Fig. 4.2 CM and Loss Function(Zero-One Loss, Hinge Loss and Logistic Loss)

$$\phi(x) = \frac{1}{\log 2} \log 1 + e^{-x}$$

于是我们可以把Logistic Loss和Hinge Loss可以看成是0-1 Loss的一个上限(参见图4.2)。我们知道0-1 Loss的另外一个常用的上限是指数损失(*Exponential Loss*):

$$\mathbb{I}(x \leq 0) \leq e^{-x} \Leftrightarrow \phi(x) = e^{-x}$$

而这个对数损失对应的这是Adaboost算法的损失函数(参见图4.3)。这个上限在Adaboost误差收敛性证明中也会用到。

Table 4.1 损失函数

损失函数	函数形式	对应算法
0-1损失(Zero-One Loss)	$\mathbb{I}(x \leq 0)$	Linear Binary Classification
铰链损失(Hinge Loss)	$\phi(x) = \max(0, 1-x)$	SVM
逻辑损失(Logistic Loss)	$\phi(x) = \frac{1}{\log 2} \log 1 + e^{-x}$	Logistic Regression
指数损失(Exponential Loss)	$\phi(x) = e^{-x}$	AdaBoost

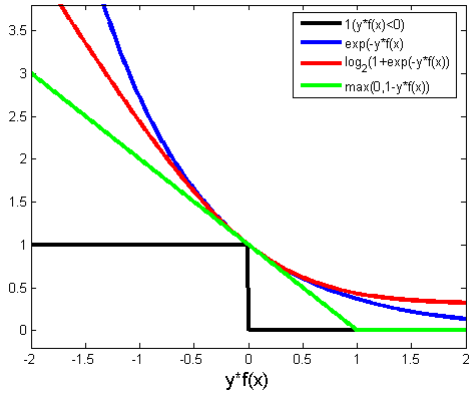


Fig. 4.3 CM and Loss Function (Zero-One Loss, Hinge Loss, Logistic Loss, and Exponential Loss)

4.3 逻辑回归的过拟合与正则化

逻辑回归也是很容易导致过拟合的。尤其在样本数据比较稀疏，或者属性维度特别大的情况下。避免过拟合一般有三类方法:

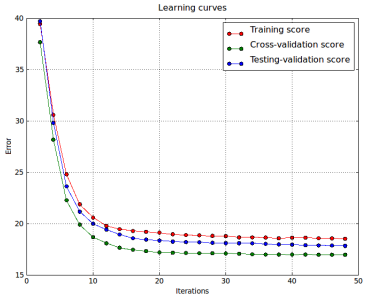
1. 增加样本数量和压缩特征属性数量：特征属性很多，且特种数目相对于训练样本较大的时候，训练数据变得极为稀疏，此时逻辑回归训练结果不稳定，且很容易陷入过拟合。可以考虑合理的增加样本数量，以及进行特征选择(Feature Selection, FS)和特征抽取(Feature Extraction, FE)：
   - 特征选择：常用过滤(Filter)方法，根据相关度(Correlation)，互信息(Mutual Information)等。也可以利用包裹(Wrapper)方法，暴力筛选特征。还有内嵌(Embeded)方法根据其他对数据稀疏或者高维特征属性空间不敏感的算法模型(SVM,KNN等)进行特征选择。
   - 特征提取：常用投影的方法，对于没有目标属性(Unsupervised)的情况下的主成分分析(Principle Component analysis, PCA)，或者自组织神经网络(Self-Organizing Mapping, SOM)。在使用目标属性(Supervised)参考的情况下，线性判别分析(Linear Discriminant Analysis, LDA)，或者投影寻踪(Projection Pursuit, PP)。
2. 提前退出训练(Early Stopping)：在发现测试误差有增大趋势的时候，停止训练。
3. 正则化(L_1 或者L_2):

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} SR(X, Y|\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{1}^{n} \log 1 + e^{-y_i \boldsymbol{\theta}^T \boldsymbol{x}_i} + \lambda \|\boldsymbol{\theta}\|_p^p$$
$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{1}^{n} \log 1 + e^{-y_i \boldsymbol{\theta}^T \boldsymbol{x}_i} + \lambda \sum_{k=0}^{|\boldsymbol{\theta}|} |\theta_k|^p$$

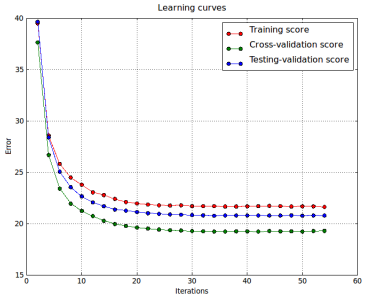
在最优化求解方面，L_2具有多次可导的特性，可以使用衍生的共轭梯度(Preconditioned Conjugate Gradient, PCG)，衍生的牛顿算法(Limited Memory BFGS, L-BFGS)。而L_1由于二次不可导，就没有那么幸运了，通常使用一些近似的替代算法(Surrogate)来逼近不可导的部分。

Table 4.2 L_2和L_1的优化方法

Method	L_2	L_1
Gradient Descent	Adam	Adam
Coordinate Descent		Cyclic Coordinate Descent (CDN)
Conjugate Gradient	Preconditioned CG	
quasi-Newton	L-BFGS	Orthant-Wise Limited-memory Quasi-Newton (OWL-QN)
Proximal Gradient		Composite Objective GD (COGD)



(a) L_1训练曲线



(b) L_2训练曲线

Fig. 4.4 L_1和L_2训练曲线比较，在相同的参数设置情况下，训练相同步数，L_1能较快的获得较低的错误率

Part IV
从两类(Bi-Class)到多类(Multi-Class)

前面的部分中，我们分别从广义线性模型和结构风险最小两个角度对逻辑回归进行了推导和解释。逻辑回归是个两类问题的分类算法，如果面对的是多类问题的分类算法，应该怎么办呢？接下来我们要从解决两类问题的逻辑回归推广到解决多类问题的软最大回归($Softmax\ Regression$)

Chapter 5
从二项(Binomial)分布到多项(Multinominal)分布

5.1 抛硬币(Coin Toss)和掷骰子(Dice)

5.1.1 伯努力*(Bernoulli)*分布

对伯努力(Bernoulli)分布$Bernoulli(p)$进行说明最常见的例子是抛硬币。假设抛一次硬币正面的概率p，那么反面的概率就为$1-p$，把两者统一起来，所以Bernoulli的概率表达式是：

$$f(x;p) = p^x(1-p)^{1-x} \qquad x \in \{0,1\}$$

5.1.2 二项*(Binomial)*分布

如果我们连续的抛1个硬币n次，那么我们就得到了二项分布$B(n,p)$。所以二项分布是一组伯努力(Bernoulli)分布变量之和。

$$X_k \sim Bernoulli(p) \Rightarrow Y = \sum_{k=1}^{n} X_k \sim \mathrm{B}(n,p)$$

$$f(x;n,p) = \binom{n}{x} p^x(1-p)^{n-x} \qquad x \in \{0,1,\ldots,n\}$$

可以看出伯努力分布是二项分布的一个特例：

$$Bernoulli(p) = B(1,p)$$

5.1.3 多项(Multinominal)分布

假设我们把抛硬币改成掷骰子，若这个骰子有K个面，并且掷n次，二项分布就变成了多项分布$M(x_1, \cdots, x_K|n, p_1, \cdots, p_K)$。

$$f(x_1, \ldots, x_K; n, p_1, \ldots, p_K) = \frac{n!}{x_1! \cdots x_K!} p_1^{x_1} \cdots p_K^{x_K}, \quad \text{where } \sum_{i=1}^{K} x_i = n$$
$$= \frac{\Gamma(\sum_i x_i + 1)}{\prod_i \Gamma(x_i + 1)} \prod_{i=1}^{K} p_i^{x_i}. \quad \Gamma \text{ is 伽玛函数}$$

因此二项分布又是多项分布的一个特例:

$$B(n, p) = M(x, n - x|n, p, 1 - p)$$

这里需要说明的，如果改成好几块(T块)不同概率的硬币(q_t是第t块硬币正面的概率)一起抛的话。那么这相当于$K = 2^T$的多项分布。假设$k - 1 = b_T \cdots b_2 b_1$, where $b_t \in \{0, 1\}$是k的二进制表示, 那么

$$p_k = \prod_1^T q_t^{b_t} (1 - q_t)^{1 - b_t}$$

$$M(x_1, \cdots, x_K|n, p_1, \cdots, p_K) = M(x_1, \cdots, x_{2^T}|n, \prod_{t=1}^{T}(1 - q_t), \cdots, \prod_{t=1}^{T} q_t)$$

Chapter 6
从二项回归(Binomial Regression)到多项回归(Multinominal Regression)

6.1 GLM模型的解释

根据广义线性模型，二项回归是对应到$\mathbb{E}(Y) \sim B(n,p)$，而多项回归对应到$\mathbb{E}(Y) \sim M(x_1,\cdots,x_K|n,p_1,\cdots,p_K)$。既然已经有了期望输出所服从的分布，那么通过对应的链接函数很容易便可以推出多项回归:

$$
\begin{aligned}
&Y \sim M(c_1,\cdots,c_K|n,p_1,\cdots,p_K) \Leftrightarrow \mu_{ik} = \mathbb{E}(y_i = c_k) \Leftrightarrow \\
&\mathcal{L}(\boldsymbol{\mu} \mid Y) = \prod_{i=1}^{n}\sum_{k=1}^{K}(\mu_{ik}\mathbb{I}(y_i = c_k))
\end{aligned}
$$

那么链接函数:

$$
\begin{aligned}
&g(\mu_{ik}) = \eta_{ik} = \boldsymbol{\theta}_k^T\boldsymbol{x}_i + \theta_{0k} = \boldsymbol{\theta}_k'^T\boldsymbol{x}_i' \Leftrightarrow \\
&\mathbf{g} = (g(\mu_{i1}),\cdots,g(\mu_{iK}))^T = (\eta_{i1},\cdots,\eta_{iK})^T = \boldsymbol{\eta}
\end{aligned}
$$

我们可以得到两种不用形式的链接函数:

$$
\boldsymbol{\eta} = \begin{bmatrix} \ln p_1 + C \\ \vdots \\ \ln p_K + C \end{bmatrix} \Leftrightarrow \mathbf{g^{-1}} = \begin{bmatrix} \frac{1}{C}e^{\eta_1} \\ \vdots \\ \frac{1}{C}e^{\eta_K} \end{bmatrix} = \begin{bmatrix} \frac{e^{\eta_1}}{\sum_{k=1}^{K}e^{\eta_k}} \\ \vdots \\ \frac{e^{\eta_K}}{\sum_{k=1}^{K}e^{\eta_k}} \end{bmatrix} \text{ where } \sum_{k=1}^{K}e^{\eta_k} = C
$$

$$\boldsymbol{\eta} = \begin{bmatrix} \ln \frac{p_1}{p_K} \\ \vdots \\ \ln \frac{p_{K-1}}{p_K} \\ 0 \end{bmatrix} = \begin{bmatrix} \ln \frac{p_1}{1-\sum_{k=1}^{K-1} p_k} \\ \vdots \\ \ln \frac{p_{K-1}}{1-\sum_{k=1}^{K-1} p_k} \\ 0 \end{bmatrix} \Leftrightarrow \mathbf{g^{-1}} = \begin{bmatrix} \frac{e^{\eta_1}}{\sum_{k=1}^{K} e^{\eta_k}} \\ \vdots \\ \frac{e^{\eta_K}}{\sum_{k=1}^{K} e^{\eta_k}} \end{bmatrix} = \begin{bmatrix} \frac{e^{\eta_1}}{1+\sum_{k=1}^{K-1} e^{\eta_k}} \\ \vdots \\ \frac{e^{\eta_{K-1}}}{1+\sum_{k=1}^{K-1} e^{\eta_k}} \\ \frac{1}{1+\sum_{k=1}^{K-1} e^{\eta_k}} \end{bmatrix}$$

6.2 软最大(Softmax)回归

6.2.1 软最大函数

软最大函数刚好是广义线性模型中的多项分布的链接函数。因此，多项回归又称为软最大(Softmax)回归。

$$\sigma(\mathbf{z})_i = \sigma((z_1, \ldots, z_i, \ldots, z_K))_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \; i \in \{1, \ldots, K\}$$

6.2.2 软最大回归的解释

软最大回归一般也称为多类逻辑回归(multiclass LR), 可以看成是适合二类问题的逻辑回归扩展到多类问题的逻辑回归。前面章节中我们指出对多项分布有两种不同形式的$\boldsymbol{\eta}$，因此我们可以基于不同的$\boldsymbol{\eta}$给出两种解释。注意，本质上两个链接函数是等价的。

1. K-1个独立二元逻辑回归: 在这种解释下，分别把前K-1个类别和第K个类别进行对比。

$$\ln \frac{\Pr(Y_i = 1)}{\Pr(Y_i = K)} = \boldsymbol{\beta}_1 \cdot \mathbf{X}_i$$
$$\ln \frac{\Pr(Y_i = 2)}{\Pr(Y_i = K)} = \boldsymbol{\beta}_2 \cdot \mathbf{X}_i$$
$$\cdots\cdots$$
$$\ln \frac{\Pr(Y_i = K-1)}{\Pr(Y_i = K)} = \boldsymbol{\beta}_{K-1} \cdot \mathbf{X}_i$$

   由此推出:

$$
\begin{aligned}
\Pr(Y_i = K) &= \frac{1}{1 + \sum_{k=1}^{K-1} e^{\boldsymbol{\beta}_k \cdot \mathbf{X}_i}} \\
\Pr(Y_i = 1) &= \frac{e^{\boldsymbol{\beta}_1 \cdot \mathbf{X}_i}}{1 + \sum_{k=1}^{K-1} e^{\boldsymbol{\beta}_k \cdot \mathbf{X}_i}} \\
\Pr(Y_i = 2) &= \frac{e^{\boldsymbol{\beta}_2 \cdot \mathbf{X}_i}}{1 + \sum_{k=1}^{K-1} e^{\boldsymbol{\beta}_k \cdot \mathbf{X}_i}} \\
&\cdots\cdots \\
\Pr(Y_i = K-1) &= \frac{e^{\boldsymbol{\beta}_{K-1} \cdot \mathbf{X}_i}}{1 + \sum_{k=1}^{K-1} e^{\boldsymbol{\beta}_k \cdot \mathbf{X}_i}}
\end{aligned}
$$

2. Log线性(Log-Linear)模型：在这种解释下，这K个类别对等看待，这样就要引入一个归一化因子Z。由于指数里面是一个线性函数，所以该模型又被称为Log线性模型。

$$
\begin{aligned}
\Pr(Y_i = 1) &= \frac{1}{Z} e^{\boldsymbol{\beta}_1 \cdot \mathbf{X}_i} \\
\Pr(Y_i = 2) &= \frac{1}{Z} e^{\boldsymbol{\beta}_2 \cdot \mathbf{X}_i} \\
&\cdots\cdots \\
\Pr(Y_i = K) &= \frac{1}{Z} e^{\boldsymbol{\beta}_K \cdot \mathbf{X}_i}
\end{aligned}
$$

$$
\begin{aligned}
1 = \sum_{k=1}^{K} \Pr(Y_i = k) = \sum_{k=1}^{K} \frac{1}{Z} e^{\boldsymbol{\beta}_k \cdot \mathbf{X}_i} = \frac{1}{Z} \sum_{k=1}^{K} e^{\boldsymbol{\beta}_k \cdot \mathbf{X}_i} \Leftrightarrow \\
Z = \sum_{k=1}^{K} e^{\boldsymbol{\beta}_k \cdot \mathbf{X}_i}
\end{aligned}
$$

Part V
最大熵(MaxEnt)

Chapter 7
最大熵原理和逻辑回归

在前面几个章节里，我们分别从广义线性模型和结构风险最小两个方面对Logistic回归进行了推导和解释，下面我们从另一个角度——熵(Entropy)来观察Logistic回归。最后我们会看到，Logistic回归可以从最大熵原理直接推导出来，或者说二者具有等价关系。

7.1 最大熵原理(Principle of Maximum Entropy)

最大熵原理是一个用于选择随机变量统计特性的原则，其主要思想是，在只掌握关于未知分布的部分知识时，应该选取符合这些知识但熵值最大的概率分布。

怎么理解这个原理呢？我们首先引入概率(Probabilities)、熵(Entropy)、限制条件(Constraints) 三个概念。

首先是概率。设A_i表示状态i，A_i发生的概率为$p(A_i)$。假设状态的总数是有限的，即$i < +\infty$。则概率分布$p(A_i)$满足：

$$p(A_i) \geq 0 \tag{7.1}$$

$$\sum_i p(A_i) = 1 \tag{7.2}$$

接下来是熵。前面的章节有给出熵的定义：

$$S = -\sum_i p(A_i) \log_2 p(A_i) \tag{7.3}$$

熵衡量的是分布$p(A_i)$的不确定性(Uncertainty)，所谓不确定性可以理解为，对确定状态所需要的信息量(Information)的一个量化指标。从熵的定义式(8.3)可以看出，当各状态的概率$p(A_i)$相等时，熵的值最大。在没有更多的信息之前，“各状态的概率相等”这个假设是相对合理的。可以这样理解：降低或升高某些状态的概率就相当于引入了新的额外的假设，而在这些

众多的假设中似乎并没有哪一个是特别合适的，因为我们不能在没有任何信息的情况下主观地认为某些状态的发生概率大于另外一些，而且从熵的定义。

最后是限制条件。所谓限制条件其实就是上面所提到的额外的信息。限制条件的存在会打破“各状态发生概率相等”的平衡，使得某些状态的概率发生变化。从熵的定义来看，这些额外的信息降低了我们对分布$p(A_i)$的不确定性。限制条件可以有很多种形式，例如某个值的期望：假设每个状态A_i对应了一个值$g(A_i)$，那么其在分布$p(A_i)$上的期望限制为G，则

$$\sum_i p(A_i)g(A_i) = G \tag{7.4}$$

现在通过以上三个概念来理解最大熵原理就直观多了：在满足**限制条件**的前提下，不引入额外的假设以免造成不确定性的下降，反映在数学上，就是分布的**熵**最大，即每个状态所分配的**概率**尽可能平均。

可以看出，最大熵原理的求解可以转化为在限制条件下的求极值(熵的最大值)问题，可以使用拉格朗日乘数法(Lagrangian)进行求解。下面一节我们会从最大熵原理来推导出Logistic回归。

7.2 最大熵原理与Logistic回归的等价性

设$\sigma(\boldsymbol{x})_v$表示$\boldsymbol{x}$属于第v个分类的概率，假设一共有K个分类。现在我们不对$\sigma(\boldsymbol{x})_v$的形式做任何假设，它可以是一个任意复杂的函数。现在我们要根据已知的确定的信息，列出对$\sigma(\boldsymbol{x})_v$ 的限制条件。首先我们知道$\sigma(\boldsymbol{x})_v$是一个概率分布，所以每个分量大于0且求和为1：

$$\sigma(\boldsymbol{x})_v \geq 0 \tag{7.5}$$

$$\sum_{v=1}^{K} \sigma(\boldsymbol{x})_v = 1 \tag{7.6}$$

还有，我们希望分布$\sigma(\boldsymbol{x})_v$满足训练集(Training Set)中数据的要求，即$\sigma(\boldsymbol{x})_v$与训练集的数据分布一致：

$$\sum_{i=1}^{n} \sigma(\boldsymbol{x}(i))_v \boldsymbol{x}(i)_j = \sum_{i=1}^{n} \mathbb{I}_v(y(i))\boldsymbol{x}(i)_j \tag{7.7}$$

其中$\mathbb{I}_u(y(i))$为指示函数，当$y(i) = u$时值为1，当$y(i) \neq u$时值为0。(8.7)式的意思是，在每一个分类u里，任意一个特征j在属于该分类的训练数据$\boldsymbol{x}(i)$上的求和，等于所训练的模型分配给特征j的概率质量(Probability Mass)之和($\boldsymbol{x}(i)$属于分类u的概率，在全部训练数据上求和)。这表明$\sigma(\boldsymbol{x}(i))_v$是对训练集的指标函数$\mathbb{I}_u(y(i))$的一个很好的近似。

$\sigma(\boldsymbol{x}(i))_v$的熵定义为：

$$-\sum_{v=1}^{K}\sum_{i=1}^{n}\sigma(\boldsymbol{x}(i))_v\log(\sigma(\boldsymbol{x}(i))_v) \tag{7.8}$$

现在我们有了概率、熵、限制条件，根据最大熵原理，我们希望在满足(8.5)(8.6)(8.7)三式时(8.8)式取到最大值。这是一个具有限制条件的最优化问题，可以使用拉格朗日乘数法求解:

$$\begin{aligned} L = & \sum_{j=1}^{m}\sum_{v=1}^{K}\lambda_{v,j}(\sum_{i=1}^{n}\sigma(\boldsymbol{x}(i))_v\boldsymbol{x}(i)_j - \sum_{i=1}^{n}\mathbb{I}_v(y(i))\boldsymbol{x}(i)_j) \\ & + \sum_{i=1}^{n}\beta_i(\sum_{v=1}^{K}\sigma(\boldsymbol{x})_v - 1) \\ & - \sum_{v=1}^{K}\sum_{i=1}^{n}\sigma(\boldsymbol{x}(i))_v\log(\sigma(\boldsymbol{x}(i))_v) \end{aligned} \tag{7.9}$$

在这里我们不打算讨论Lagrangian的对偶(Dual)问题和Karush-Kuhn-Tucker(KKT)条件，(8.9) 式中的L对$\sigma(\boldsymbol{x}(i))_v$求偏导得到:

$$\frac{\partial L}{\partial\sigma(\boldsymbol{x}(i))_v} = \lambda_v\boldsymbol{x}(i) + \beta_i - \log(\sigma(\boldsymbol{x}(i))_v) - 1 \tag{7.10}$$

令上式(8.10)等于0:

$$\lambda_v\boldsymbol{x}(i) + \beta_i - \log(\sigma(\boldsymbol{x}(i))_v) - 1 = 0 \tag{7.11}$$

解方程得到:

$$\sigma(\boldsymbol{x}(i))_v = e^{\lambda_v\boldsymbol{x}(i)+\beta_i-1} \tag{7.12}$$

根据限制条件(8.6)，概率$\sigma(\boldsymbol{x}(i))_v$求和等于1:

$$\sum_{v=1}^{k}e^{\lambda_v\boldsymbol{x}(i)+\beta_i-1} = 1 \tag{7.13}$$

于是得到:

$$e^{\beta_i-1} = \frac{1}{\sum_{v=1}^{k}e^{\lambda_v\boldsymbol{x}(i)-1}} \tag{7.14}$$

把上式中的e^{β_i-1}代入(8.12):

$$\sigma(\boldsymbol{x}(i)) = \frac{e^{\lambda_u\boldsymbol{x}(i)}}{\sum_{v=1}^{k}e^{\lambda_v\boldsymbol{x}(i)}} \tag{7.15}$$

即

$$\sigma(\boldsymbol{x}) = \frac{e^{\lambda_u\boldsymbol{x}}}{\sum_{v=1}^{k}e^{\lambda_v\boldsymbol{x}}} \tag{7.16}$$

(8.16)式即为我们在章节7.2中从广义线性模型推出来的Softmax回归。

7.3 Log-Linear模型

假设我们对Softmax函数的每个分量做线性扩展，那么我们就得到Log-Linear模型

$$p(y|x;\mathbf{w}) = \frac{\exp\left(c + \sum_{j=1}^{|\mathbf{w}|} w_j f_j(x,y)\right)}{Z(x,\mathbf{w})},$$
$$Z(x,\mathbf{w}) = \sum_{y'\in Y} \exp\left(c + \sum_{j=1}^{|\mathbf{w}|} w_j f_j(x,y')\right)$$

如果写成向量的形式就变成了：

$$p(y|x;\mathbf{w}) = \frac{\exp\left(\mathbf{w}^T\mathbf{f}(x,y)\right)}{Z(x,\mathbf{w})}$$

对比一下Softmax函数，我们把Softmax的每个输入自变量变成了线性：

$$v(x,y) = \mathbf{w}^T\mathbf{f}(x,y) \Rightarrow$$
$$p(y|x;\mathbf{w}) = \sigma(v(x,y))_{y\in Y} = \frac{e^{v(x,y)}}{\sum_{y'\in Y} e^{v(x,y')}}$$

这样，有了LogLinear的假设，我们根据训练数据X, Y, 来优化参数w。根据最大似然估计，我们的目标是找到Likelihood $\ell(\mathbf{w})$ 对w_k的导数形式：

$$\ell(\mathbf{w}) = \log P(X,Y|\mathbf{w}) \tag{7.17}$$
$$= \sum_{i=1}^{N} \log p(x_i,y_i|\mathbf{w}) \tag{7.18}$$
$$= \sum_{i=1}^{N} \log p(y_i|x_i;\mathbf{w})p(x_i) \tag{7.19}$$
$$= \sum_{i=1}^{N} \log p(y_i|x_i;\mathbf{w}) + \sum_{i=1}^{N} \log p(x_i) \tag{7.20}$$

$$\frac{\partial \ell(\mathbf{w})}{\partial w_k} = \sum_{i=1}^{N} \frac{\partial \log p(y_i|x_i;\mathbf{w})}{\partial w_k} \tag{7.21}$$

其中，我们省略与参数无关的$\sum_{i=1}^{N} \log p(x_i)$,

$$\ell(\mathbf{w}) = \sum_{i=1}^{n} \log p(y_i|x_i;\mathbf{w}) \tag{7.22}$$

对于上式求和项中的一项$\log p(y_i|x_i;\mathbf{w})$有

$$\log p(y_i|x_i;\mathbf{w}) = \log \frac{\exp(\mathbf{w}^T\mathbf{f}(x_i,y_i))}{\sum_{y'\in\mathcal{Y}}\exp(\mathbf{w}^T\mathbf{f}(x_i,y'))} \tag{7.23}$$

$$= \mathbf{w}^T\mathbf{f}(x_i,y_i) - \log\sum_{y'\in\mathcal{Y}}\exp(\mathbf{w}^T\mathbf{f}(x_i,y')) \tag{7.24}$$

上式右边第一项对w_k求导结果比较简单：

$$\frac{\partial}{\partial w_k}\mathbf{w}^T\mathbf{f}(x_i,y_i) = \frac{\partial}{\partial w_k}(\sum_k w_k f_k(x_i,y_i)) = f_k(x_i,y_i) \tag{7.25}$$

记$g(\mathbf{w}) = \sum_{y'\in\mathcal{Y}}\exp(\mathbf{w}^T\mathbf{f}(x_i,y'))$，则右边第二项为：

$$\log g(\mathbf{w}) \tag{7.26}$$

上式对w_k求导为

$$\frac{\partial}{\partial w_k}\log g(\mathbf{w}) = \frac{1}{g(\mathbf{w})}\frac{\partial}{\partial w_k}g(\mathbf{w}) \tag{7.27}$$

其中

$$\frac{\partial}{\partial w_k}g(\mathbf{w}) = \sum_{y'\in\mathcal{Y}} f_k(x_i,y')\exp(\mathbf{w}^T\mathbf{f}(x_i,y')) \tag{7.28}$$

所以我们有

$$\frac{\partial}{\partial w_k}\log g(\mathbf{w}) = \frac{\sum_{y'\in\mathcal{Y}} f_k(x_i,y')\exp(\mathbf{w}^T\mathbf{f}(x_i,y'))}{\sum_{y'\in\mathcal{Y}}\exp(\mathbf{w}^T\mathbf{f}(x_i,y'))} \tag{7.29}$$

$$= \sum_{y'\in\mathcal{Y}} f_k(x_i,y') \times \frac{\exp(\mathbf{w}^T\mathbf{f}(x_i,y'))}{\sum_{y'\in\mathcal{Y}}\exp(\mathbf{w}^T\mathbf{f}(x_i,y'))}) \tag{7.30}$$

$$= \sum_{y'\in\mathcal{Y}} f_k(x_i,y')p(y'|x;\mathbf{w}) \tag{7.31}$$

最终，把(7.25)和(7.31)结合起来得到：

$$\frac{dL(v)}{dv_k} = \sum_{i=1}^{n} f_k(x^{(i)},y^{(i)})) = f_k(x^{(i)},y^{(i)}) - \sum_{i=1}^{n}\sum_{y'\in\mathcal{Y}} p(y|x^{(i)};v)f_k(x^{(i)},y) \tag{7.32}$$

Part VI
从熵(Entropy)解释GLM, MaxEnt, MLE

Chapter 8
来自熵(Entropy)的解释

在广义线性模型中，我们提到了最大似然估计，指数分布族以及其中的累计量生成函数等概念，这里我们将从信息熵出发来解释这些它们，并最终证明这些概念都是可以从熵推导出来。

8.1 对信息熵的理解

香农信息熵是对数据分布的不确定性进行度量：假设X是来自分布$\mathcal{X}$的随机样本，则$P(x_i) = P(X = x_i) = p_{x_i}$，那么：

$$\mathbb{H}(X) = \mathbb{E}[\mathrm{I}(X)] = \mathbb{E}[-\ln(\mathrm{P}(X))].$$
$$\mathbb{H}(X) = \sum_{i=1}^{n} \mathrm{P}(x_i)\,\mathrm{I}(x_i) = -\sum_{i=1}^{n} \mathrm{P}(x_i)\log_2 \mathrm{P}(x_i),$$

一般来说香农信息熵的含义有两种解释：

- 不确定性公理化(Aximatic)解释：上述公式是满足不确定性公理化假设的唯一数学形式。
- 期望编码(Coding)长度解释：上述公式是对概率分布信息的二进制编码的期望。

8.1.1 不确定性公理化解释

对于分布概率的不确定性$\mathbb{H}(X) \equiv \mathbb{H}(p_1, \ldots, p_N)$，必须满足不确定性公理的四大假设：

1. 非负假设：$\mathbb{H}(p_1, \ldots, p_N) \geq 0$
2. 连续性假设：$\mathbb{H}(p_1, \ldots, p_N)$按全部自变量$p_i$是连续的。

3. 单调性假设：如果所有$p_i = 1/N$，那么$\mathbb{H}(p_1, \ldots, p_N)$, 必须随着$N$的增加，而不确定性增大。
4. 叠加性假设：不确定性是随着概率分布生成过程来进行叠加的。

- 比如A变量是通过X, Y, Z来生成的，$X = x_1, x_2$，当x_1 满足是$P(Y)$，当x_2 不满足是$P(Z)$，那么就有如下关系成立:

$$\mathbb{H}(A) = \mathbb{H}(X) + P(x_1)\mathbb{H}(Y) + P(x_2)\mathbb{H}(Z)$$

- 其中一种特殊情况是$A = X_1, \ldots, X_n$，那么

$$\mathbb{H}(A) = \sum_{i=1}^{n} \mathbb{H}(X_i)$$

- 另外一种特殊情况，有相互独立的两个事件$Y = y_1, \ldots, y_n$和$Z = z_1, \ldots, z_m$：那么：

$$\mathbb{H}(YZ) = \sum_{i=1}^{n}\sum_{j=1}^{m} \phi(y_i z_j) = \sum_{i=1}^{n} \phi(y_i) + \sum_{j=1}^{m} \phi(z_j) = \mathbb{H}(Y) + \mathbb{H}(Z) \tag{8.1}$$

而满足上面四个公理条件的唯一形式是:

$$\mathbb{H}(X) = K \sum_{i=1}^{n} \mathrm{P}(x_i) \log \frac{1}{\mathrm{P}(x_i)}$$

根据连续性假设，我们对公式8.1，分别对y_k 和y_t求导:

$$\sum_{j=1}^{m} z_j \phi'(y_k z_j) = \phi'(y_k)\,, \ \sum_{j=1}^{m} z_t \phi'(y_t z_j) = \phi'(y_t)$$

$$\sum_{j=1}^{m} z_j [\phi'(y_k z_j) - \phi'(y_t z_j)] = \phi'(y_k) - \phi'(y_t)$$

而上述表达式的右手边(Right Hand Side, r.h.s)是跟z_j没有关系的。

$$\sum_{j=1}^{m} z_j [\phi'(y_k z_j) - \phi'(y_t z_j) - (\phi'(y_k) - \phi'(y_t))] = 0$$

对任意独立的Y, Z，上述公式都成立的情况下，可以进一步推导到:

$$\phi'(y_k z_j) - \phi'(y_t z_j) - (\phi'(y_k) - \phi'(y_t)) = 0$$
$$\phi'(y_k z_j) - \phi'(y_t z_j) = \phi'(y_k) - \phi'(y_t)$$

如果设$y_t = 1$，则

$$\begin{aligned}\phi'(y_k z_j) - \phi'(y_t z_j) &= \phi'(y_k) - \phi'(y_t)\\ \phi'(y_k z_j) - \phi'(z_j) &= \phi'(y_k) - \phi'(1)\\ \phi'(y_k z_j) &= \phi'(y_k) + \phi'(z_j) - \phi'(1)\end{aligned}$$

当然这也可以从公式8.1进行证明。

$$\begin{aligned}\phi'(y_k z_j) &= \phi'(y_k) + \phi'(z_j) - \phi'(1)\\ \phi'(y_k z_j) &= \phi'(y_k) + \phi'(z_j)\end{aligned}$$

由此，我们转化成柯西函数公式(Cauchy's Function Equation)的$f(x) = Ax + B$:

$$\begin{aligned}f(x+y) &= f(x) + f(y)\\ f(\log y_k + \log z_j) &= f(\log y_k) + f(\log z_j)\\ \phi'(x) &= f(\log x)\\ \phi'(x) &= K\log x + B\\ \phi(x) &= Kx\log x + (B-K)x + C\end{aligned}$$

另外根据不确定性定义，概率为0和概率为1都是确定的情况，那么有$\phi(0) = 0$，$\phi(1) = 0$，于是得到$C = 0, B - K = 0$，由此可得

$$\begin{aligned}\phi(x) &= Kx\log x\\ \mathbb{H}(X) &= \phi(P(X)) = \phi(\{P(x_1), \ldots, P(x_n)\})\\ &= \sum_{i=1}^{n} K\mathrm{P}(x_i)\log\frac{1}{\mathrm{P}(x_i)}\\ &= K\sum_{i=1}^{n}\mathrm{P}(x_i)\log\frac{1}{\mathrm{P}(x_i)}\end{aligned}$$

虽然通过公理化的假设可以推出信息熵的一般形式，但是这种形式晦涩难懂，解释起来比较困难，用来不会得心应手。所以下面我们从另一个角度给出一个直观的解释。

8.1.2 期望编码长度解释

前面提到信息熵来源于香农推导出的一个度量不确定性的公式。后来在冯诺依曼的建议下信息熵和玻尔兹曼在热力学中提出的的H熵联系了起

来。而薛定谔提出了对H熵基于状态数的解释。把H熵的状态数解释类比过来便是信息熵的期望编码长度解释。假设有N个状态数，那么每个状态的概率$p_i = 1/N$，则$N = 1/p_i$。如果对N个状态数进行二进制编码，前缀码(Prefix Codes)的种类数为N，那么二进制码的长度必须要为：

$$Len(p_i) = \log N = \log \frac{1}{p_i}$$

假设有一组概率值$\{p_1, \ldots, p_n\}$，那么我们要求平均编码长度：

$$\mathbb{E}(Len(p_i)) = \sum_{i=1}^{n} p_i Len(p_i) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i}$$

期望编码长度的解释比较直观。首先通过概率倒数来解释成状态数，然后通过编码长度来解释log的作用。

8.1.2.1 相对熵

在香农信息熵的基础上，可以很容易地得到相对熵(Relative Entropy, RE)的定义：

$$\begin{aligned} RE(P\|Q) &= -\sum_i P(i)\ \log \frac{P(i)}{Q(i)} \\ &= \sum_i P(i)\,(\log \frac{1}{P(i)} - \log \frac{1}{Q(i)}) \end{aligned}$$

上式可以这样理解：给定Q分布，想看一下在P分布情况，于是就用P的编码长度减去Q的编码长度在P分布下的期望作为一种衡量。

8.1.2.2 KL散度

从相对熵的概念我们可以定义出两个分布的散度。由于相对熵恒小于0，且散度定义要求其必须非负，所以在相对熵的前面加一个负号，就得到了我们需要的散度，即KL散度(Kullback-Leibler Divergence)：

$$\begin{aligned} D_{\mathrm{KL}}(P\|Q) &= \sum_i P(i)\ \log \frac{P(i)}{Q(i)} \\ &= \sum_i P(i)\,(\log \frac{1}{Q(i)} - \log \frac{1}{P(i)}) \end{aligned}$$

给定Q分布，P分布与Q分布的KL散度即为，对于Q的编码长度与P的编码

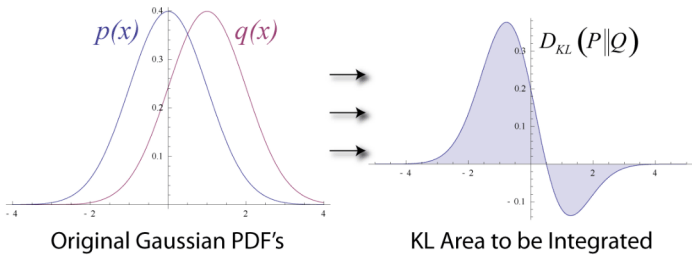


Fig. 8.1 KL散度(Kullback-Leibler Divergence)

长度$\log\frac{1}{Q(i)}-\log\frac{1}{P(i)}$之差在P上面的期望。从图8.1可以看到编码长度之差可能有正有负，然后按P的概率密度积分就是编码长度之差的期望了。

8.1.2.3 互信息

互信息(Mutual Information, MI)的定义如下:

$$\begin{aligned}\mathbb{I}(X;Y)&=\sum_{x,y}p(x,y)\log\frac{p(x,y)}{p(x)p(y)}\\&=\sum_{x,y}p(x,y)(\log\frac{1}{p(x)p(y)}-\log\frac{1}{p(x,y)})\end{aligned}$$

假设X与Y相互独立的那么$p(x,y)=p(x)p(y)$，于是互信息的直观意义就是X，Y在假设在独立情况下，和真实的非独立情况下的编码长度之差在X和Y联合分布上的期望。我们对这个式子进一步化解:

$$\begin{aligned}I(X;Y)&=\sum_{x,y}p(x,y)\log\frac{p(x,y)}{p(x)p(y)}\\&=\sum_{y}p(y)\sum_{x}p(x|y)\log\frac{p(x|y)}{p(x)}\\&=\sum_{y}p(y)\ D_{\mathrm{KL}}(p(x|y)\|p(x))\\&=\mathbb{E}_{Y}\{D_{\mathrm{KL}}(p(x|y)\|p(x))\}.\end{aligned}$$

因此互信息也可以看成条件分布$p(x|y)$到分布$p(x)$的KL散度在Y上的期望。

此外，互信息还和条件熵有着极大关系——互信息可以看成是熵和条件熵之差(参见图8.2):

$$
\begin{aligned}
\mathbb{I}(X;Y) &= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\
&= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)} - \sum_{x,y} p(x,y) \log p(y) \\
&= \sum_{x,y} p(x)p(y|x) \log p(y|x) - \sum_{x,y} p(x,y) \log p(y) \\
&= \sum_{x} p(x) \left(\sum_{y} p(y|x) \log p(y|x) \right) - \sum_{y} \log p(y) \left(\sum_{x} p(x,y) \right) \\
&= -\sum_{x} p(x) H(Y|X=x) - \sum_{y} \log p(y) p(y) \\
&= -H(Y|X) + H(Y) \\
&= H(Y) - H(Y|X).
\end{aligned}
$$

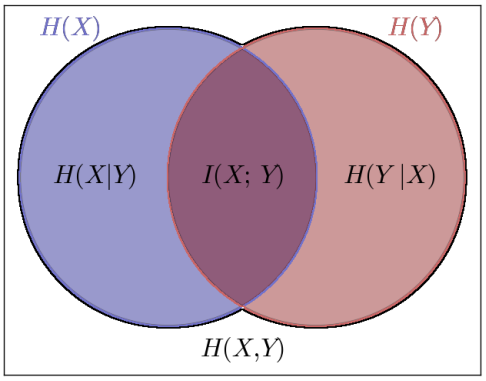


Fig. 8.2 互信息(Mutual Information)

8.2 最大信息熵原理

8.2.1 最大熵解释原理的直观理解

我们在上一部分中通过最大熵原理导出了逻辑回归的形式，事实上最大熵原理也是可以推导出来的。假设总有N的定额配量(quanta)分到M个状态中去，若每个状态中分到n_i，那么处在那个状态的概率为

$$p_i = \frac{n_i}{N}$$

现在的问题是如何配置p_i，使得分配N个球到M个状态的状态数最大。这有点类似于N次掷M面骰子的多项式分布，那么会掷出多少中不同的状态数呢？根据多项式组合：

$$W = \frac{N!}{n_1!\, n_2! \cdots n_m!}$$

我们也知道每次掷骰子可能有M种情况，那么总数为m^N次，由此概率为

$$P(\mathbf{p}) = \frac{W}{m^N}$$

那么，这个概率越大，或者说W越大(或者说哪种状态分配下，可以选择的排列数越多)，就表示该种状态分配的不确定性越大，即熵越大。则**最大不确定性**就是要**最大W**。假设我们固定状态数M，让实验数$N \to \infty$，根据玻尔兹曼的定义形式：

$$\begin{aligned}
\frac{1}{N}\log W &= \frac{1}{N}\log\frac{N!}{n_1!\, n_2! \cdots n_m!} \\
&= \frac{1}{N}\log\frac{N!}{(Np_1)!\,(Np_2)! \cdots (Np_m)!} \\
&= \frac{1}{N}\left(\log N! - \sum_{i=1}^{m}\log((Np_i)!)\right).
\end{aligned}$$

当N趋于无穷大的时候，我们根据斯特林Stirling近似

$$\ln n! = n\ln n - n + O(\ln n)$$

可以得到

$$\begin{aligned}\lim_{N\to\infty}\left(\frac{1}{N}\log W\right) &= \frac{1}{N}\left(N\log N - N - \sum_{i=1}^{m}(Np_i\log(Np_i) - Np_i)\right)\\ &= \log N - \sum_{i=1}^{m} p_i\log(Np_i) - N(1-\sum_{i=1}^{m}p_i)\\ &= \log N - \log N\sum_{i=1}^{m}p_i - \sum_{i=1}^{m}p_i\log p_i\\ &= \left(1-\sum_{i=1}^{m}p_i\right)\log N - \sum_{i=1}^{m}p_i\log p_i\\ &= -\sum_{i=1}^{m}p_i\log p_i\\ &= \mathbb{H}(\mathbf{p}).\end{aligned}$$

上面的推导直观地说明了最大熵原理的含义，即找一个分布使得在这个分布上，不确定性最大。

8.2.2 自然指数分布族的最大熵解释

我们在GLM那部分中用到的自然指数分布族所具有的通式，可以从最大熵直接导出。首先给出三个假设和一个目标:

1. 初始观察分布$m(x)$：这个可以是随意的观察情况，不一定必须是一个分布函数。
2. $\sum_{x\in S} f(x) = 1,\ f(x) \geq 0$：满足分布函数的情况，这里仅仅考虑离散的情况。
3. $\sum_{x\in S} t_j(x)f(x) = \mathbb{E}\{t_j(x)\} = \mu_j$，其中$j \in J$：$t_j(x)$是在结合上的一个测量函数，并且我们知道测量值的期望是μ_j。
4. 目标：$\arg\max_{f(x)} RE(f(x)\|m(x))$：我们希望找一个函数满足上述的限制条件，并且尽可能与初始观察分布的相对熵最大。

$$RE(f(x)\|m(x)) = -\sum_{x\in S} f(x)\log\frac{f(x)}{m(x)}$$

应用拉格朗日乘数法:

$$L(f) = -\sum_{x\in S} f(x)\log\frac{f(x)}{m(x)} + \lambda(\sum_{x\in S} f(x) - 1) + \sum_{j\in J}\theta_j(\sum_{x\in S} t_j(x)f(x) - \mu_j)$$

令$L(f)$对f的导数为0

$$0 = \frac{\partial L(f)}{\partial f} = -(\log \frac{f(x)}{m(x)} + 1) + \lambda + \sum_{j \in J} \theta_j t_j(x)$$
$$= -\log f(x) + \log m(x) - 1 + \lambda + \sum_{j \in J} \theta_j t_j(x)$$

从而得到

$$f(x) = m(x) \exp[\lambda - 1 + \sum_{j \in J} \theta_j t_j(x)] \tag{8.2}$$

接下来应用概率求和为1的限制条件：

$$1 = \sum_{x \in S} f(x) = \sum_{x \in S} m(x) \exp\{\lambda - 1 + \sum_{j \in J} \theta_j t_j(x)\}$$
$$= e^{\lambda - 1} \sum_{x \in S} m(x) \exp\{\sum_{j \in J} \theta_j t_j(x)\}$$

于是有

$$1 - \lambda = \log \left(\sum_{x \in S} m(x) \exp[\sum_{j \in J} \theta_j t_j(x)] \right)$$

我们将上式的右边定义为$b(\boldsymbol{\theta})$，其中$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{|J|})$，$\mathbf{t}(x) = (t_1(x), \ldots, t_{|J|}(x))$：

$$b(\boldsymbol{\theta}) = \log \left(\sum_{x \in S} m(x) \exp[\sum_{j \in J} \theta_j t_j(x)] \right)$$
$$= \log \left(\mathbb{E}(e^{\mathbf{t}(x)^T \boldsymbol{\theta}}) \right)$$
$$= 1 - \lambda$$

在(8.2)式中替换$1 - \lambda$得到$f(x)$的如下形式：

$$f(x) = m(x) \exp[-b(\boldsymbol{\theta}) + \textstyle\sum_{j \in J} \theta_j t_j(x)] \tag{8.3}$$
$$= m(x) \exp[\mathbf{t}(x)^T \boldsymbol{\theta} - b(\boldsymbol{\theta})] \tag{8.4}$$

(8.4)式即为自然指数分布族的形式，其中$b(\boldsymbol{\theta})$是累积量生成函数。由于最大相对熵就是最小KL散度，由此我们可以得出看出，自然指数分布族的形式，其实就是和初始观察分布最相似的满足对不同定义的测量值的期望在固定情况下的概率密度函数。

8.2.3 最大似然估计的最大熵解释

最大相对熵不仅仅可以用来推导出自然指数分布，而且可以作为最大似然估计的理论基础。

假设我们观察到N个样本，那么根据样本的估算概率，或者说根据频率来计算一个经验分布$\tilde{p}(x)$定义如下：

$$\tilde{p}(x) = \frac{1}{N}\sum_{n=1}^{N}\delta(x, x_n)$$

其中$\delta(x, x_n)$是狄拉克测量(Dirac Measure)，在这里是和指示函数等价的。

根据大数定理我们知道当抽样$n \to \infty$的时候，$\tilde{p}(x) \to p(x)$。我们考虑利用最大相对熵计算根据样本的估算概率$\tilde{p}(x)$与给定参数的条件分布$p(x;\theta)$。首先有相对熵：

$$RE\left(\tilde{p}(x)\|p(x|\theta)\right) = -\sum_{x}\tilde{p}(x)\log\frac{\tilde{p}(x)}{p(x|\theta)} \tag{8.5}$$

$$= -\sum_{x}\tilde{p}(x)\log\tilde{p}(x) + \sum_{x}\tilde{p}(x)\log p(x|\theta) \tag{8.6}$$

根据最大相对熵：

$$\begin{aligned}\max RE\left(\tilde{p}(x)\|p(x|\theta)\right) &\Leftrightarrow \min D_{KL}\left(\tilde{p}(x)\|p(x|\theta)\right)\\ &\Rightarrow \theta^* = \arg\max_{\theta} RE\left(\tilde{p}(x)\|p(x|\theta)\right)\\ &\Rightarrow \theta^* = \arg\max_{\theta}\sum_{x}\tilde{p}(x)\log p(x|\theta)\end{aligned}$$

将前面的经验分布代入：

$$\begin{aligned}\sum_{x}\tilde{p}(x)\log p(x|\theta) &= \sum_{x}\frac{1}{N}\sum_{n=1}^{N}\delta(x, x_n)\log p(x|\theta)\\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{x}\delta(x, x_n)\log p(x|\theta)\\ &= \frac{1}{N}\sum_{n=1}^{N}\log p(x_n|\theta)\end{aligned}$$

而同时我们知道对数似然函数(Log Likelihood, LL)的表达式如下：

$$\ell(\theta) = \log P(X|\theta) = \sum_{n=1}^{N} \log p(x_n|\theta)$$

结合起来最终得到:

$$\begin{aligned}
\theta^* &= \arg\max_{\theta} RE\left(\tilde{p}(x) \| p(x|\theta)\right) \\
&= \arg\max_{\theta} \sum_{x} \tilde{p}(x) \log p(x|\theta) \\
&= \arg\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log p(x_n|\theta) \\
&= \arg\max_{\theta} \frac{1}{N} \ell(\theta) \\
&= \arg\max_{\theta} \ell(\theta)
\end{aligned}$$

由此根据最大熵，我们可以得出最大似然估计。此外根据式子(8.6)，我们可以得到如下公式:

$$RE\left(\tilde{p}(x) \| p(x|\theta)\right) = -D_{KL}\left(\tilde{p}(x) \| p(x|\theta)\right) = \mathbb{H}(\tilde{p}(x)) + \frac{1}{N}\ell(\theta)$$

这个表达式在变分分析(Variational Inference)可能会来构建逼近下限。

Appendices

Appendix C
附录

C.1 信息熵的由来

鲁道夫.克劳修斯Rudolf.Clausius，这位来自科沙林的大神，学文学出生，从能量守恒中重新认识少年天才尼古拉.卡诺Nicolas.Carnot的卡诺热机和循环的卡诺原理，建立热力学第二定律，命名了熵Entropy。法国天才卡诺，36岁就霍乱而死，匠人，写下《论火的动力》，成为热力学之父！死前孤独凄凉，著作无人问津。克劳修斯把熵记成S，是卡诺的中间名Sadi。克劳修斯，15年研究，出了一篇论文，修神功成！

Fig. C.1 尼古拉.卡诺Nicolas.Carnot

约西亚.吉布斯Josiah.Gibbs，全才，出生耶鲁的他，把统计引入热力学。在克劳修斯基础上，提出能量变化的计算。统计热力学的另外两位剑客，詹姆斯.麦克斯韦James.Maxwell和路德维希.波尔兹曼Ludwig.Boltzman，一个师出剑桥，一个师出维也纳大学。

波尔兹曼第一次定义熵为乱序的统计。并且引入对数形式。吉布斯对此进行改进，成为统计热力学的基石。并且标记熵为H函数。

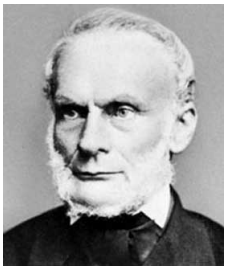


Fig. C.2 鲁道夫.克劳修斯Rudolf.Clausius

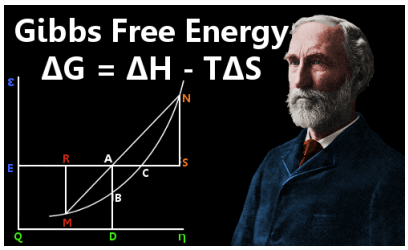


Fig. C.3 约西亚.吉布斯Josiah.Gibbs

Fig. C.4 艾尔文.薛定谔Erwin.Schrodinger

艾尔文.薛定谔Erwin.Schrodinger，同样出生维也纳大学，量子力学奠基人（量子力学量力学）第一次把概率倒数解释称状态数量，这样对数形式就可以解释成编码长度。并且引入了负号。

年轻的克劳德.香农Claude.Shannon，密西根大学的天才，将薛定谔的解释正式引入信号处理中，定义为信息损失。从此建立了信息熵，开创了信息理论。在和计算机之神约翰.冯.诺伊曼John.von.Neumann，来自苏黎世联邦理工大学的少年天才，讨论后正式把信息熵定义成不确信的测度。并且给出了H函数的定义。

Fig. C.5 克劳德.香农Claude.Shannon

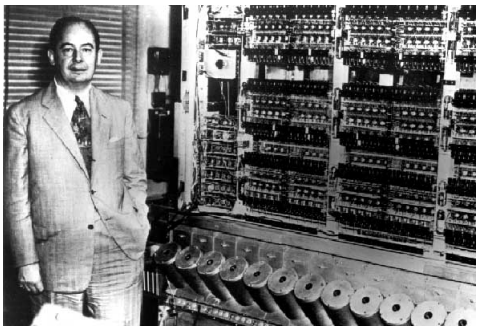


Fig. C.6 约翰.冯.诺伊曼John.von.Neumann

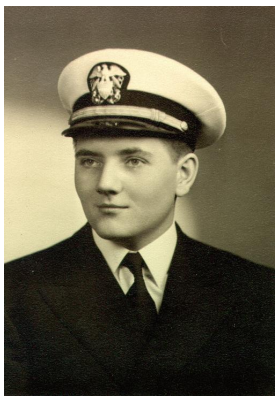


Fig. C.7 埃德温.杰恩斯Edwin.Jaynes

埃德温.杰恩斯Edwin. T. Jaynes, 普林斯顿的杰出统计学家，开创性的通过逻辑来解释统计，通过最大熵来解释统计，由此开创了统计分析的贝叶斯学派。

至此，熵一发而不可收拾的成为诸多信息处理理论基石！

Part VII
Logistic PCA

Chapter 9
Logistic PCA和PCA在指数分布族上的推广

我们在第2章介绍了广义线性模型，并通过广义线性模型对线性回归进行了拓展，使得我们可以对满足不同分布(指数分布族)的期望输出进行回归拟合。这种利用指数分布族处理不同形式数据(连续、多类、二值等)的思想同样可以用于其它场景，比如降维(Dimensionality Reduction) 中的主成分分析(Principal Component Analysis, PCA)。我们知道经典的PCA只能用于连续变量，而接下来我们会利用指数分布族对其进行推广。这跟我们利用GLM对只能用于期望输出是连续变量的线性回归进行推广是非常相似的。

9.1 主成份分析(PCA)

设$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathbb{R}^D$，并假设数据已经经过中心化(Centralization)即$\sum_i \boldsymbol{x}_i = 0$。PCA 要解决的问题是：在一个维度更低的空间$\mathbb{R}^d$中表示这些数据点($d < D$)。

下面我们分别从三个方向证明和解释PCA。

9.1.1 最大方差投影

考虑把数据点向一个通过原点的超平面(Hyperplane)投影(Projection)，设该超平面的单位法向量为$\mathbf{w} \in \mathbb{R}^D$。数据点$\boldsymbol{x}$向$\mathbf{w}$的投影为

$$\mathbf{w}^T \boldsymbol{x} \tag{9.1}$$

则投影后的数据集方差(Variance)为

$$\frac{1}{n} \sum_{i=1}^{n} (\mathbf{w}^T \boldsymbol{x}_i)^2 = \mathbf{w}^T S \mathbf{w} \tag{9.2}$$

其中

$$S = \frac{1}{n}\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^T \tag{9.3}$$

因为已经假设所有数据点已经过中心化处理，所以S即为数据集的协方差矩阵(*Covariance Matrix*)。

PCA所做的事情是寻找一个$\mathbf{w}$使得数据集在其上的投影方差最大，其背后的思想是最大程度地保留数据点之间的差异。可以看出这是一个有限制条件的最优化问题:

$$\max_{\mathbf{w}} \quad \mathbf{w}^T S \mathbf{w} \tag{9.4}$$

$$s.t. \quad \|\mathbf{w}\| = 1 \tag{9.5}$$

于是就可以使用拉格朗日乘数法(*Lagrangian*)进行求解

$$L = \mathbf{w}^T S \mathbf{w} + \lambda(1 - \mathbf{w}^T \mathbf{w}) \tag{9.6}$$

L对$\mathbf{w}$求偏导得到

$$\frac{\partial L}{\partial \mathbf{w}} = 2S\mathbf{w} - 2\lambda\mathbf{w} \tag{9.7}$$

令上式等于0可得

$$S\mathbf{w} = \lambda\mathbf{w} \tag{9.8}$$

而上式恰好是矩阵的特征值和特征向量的定义式。$\mathbf{w}$是协方差矩阵S的特征向量，而λ是其对应的特征值。现在的问题是，哪一个特征向量$\mathbf{w}$所确定的超平面才是我们所寻找的方差最大投影的那个超平面呢？根据(9.4)式数据集投影方差为

$$\mathbf{w}^T S \mathbf{w} = \mathbf{w}^T \lambda \mathbf{w} = \lambda \tag{9.9}$$

因此，最大的特征值对应的特征向量就是我们寻找的超平面$\mathbf{w}$。

9.1.2 最小重建误差

现在我们从另外一个角度——重建(Reconstruction)来解释PCA。设任意一组正交基$\mathbf{u}_1, \cdots, \mathbf{u}_D$，那么任意一个中心化之后的数据点$\boldsymbol{x}_i$可以表示成这组正交基的线性组合

$$\boldsymbol{x}_i = \sum_{j=1}^{D} \alpha_{ij} \mathbf{u}_j \tag{9.10}$$

其中

$$\alpha_{ij} = \mathbf{u}_j^T \boldsymbol{x}_i \tag{9.11}$$

既然我们的最终目标是降低维度，那么考虑选取d个正交基上的分量进行重建，得到原始数据点$\boldsymbol{x}_i$的近似:

$$\hat{\boldsymbol{x}}_i = \sum_{j=1}^{d} \alpha_{ij} \mathbf{u}_j \tag{9.12}$$

我们希望重建出来的$\hat{\boldsymbol{x}}_i$与原始数据点$\boldsymbol{x}_i$ 的误差(Error)尽可能的小:

$$\frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2 = \frac{1}{n}\sum_{i=1}^{n} \|\sum_{j=1}^{D} \alpha_{ij}\mathbf{u}_j - \sum_{j=1}^{d} \alpha_{ij}\mathbf{u}_j\|^2 \tag{9.13}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \|\sum_{j=d+1}^{D} \alpha_{ij}\mathbf{u}_j\|^2 \tag{9.14}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=d+1}^{D} \alpha_{ij}^2 \tag{9.15}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=d+1}^{D} \mathbf{u}_j^T \boldsymbol{x}_i \boldsymbol{x}_i^T \mathbf{u}_j \tag{9.16}$$

$$= \sum_{j=d+1}^{D} \mathbf{u}_j^T S \mathbf{u}_j \tag{9.17}$$

其中(9.15)因为$\mathbf{u}$相互正交，(9.16)因为直接代入(9.11)，(9.17)由直接带入协方差矩阵的定义式(9.3)得到。现在的问题是，如何选择那些去掉的$D-d$个正交基呢？考虑$d = D-1$ 的情况，此时我们需要去掉一个维度，很容易看出我们应该去掉协方差矩阵S最小特征值λ_D 对应的那个特征向量$\mathbf{w}_D$，因为在该超平面上，数据集投影的方差$\mathbf{w}_D^T S \mathbf{w}_D = \lambda_D$ 最小。类似地，我们逐步去掉特征值第2小、第3小.....第(D-d)小的方向(特征向量)，就得到了我们希望的结果。

9.1.3 高斯随机采样

在PCA的最小重建误差解释中我们的目标是重建出来的近似$\hat{\boldsymbol{x}}_i$ 与原始数据点$\boldsymbol{x}_i$的误差(Error)尽可能的小，即

$$\frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2 \tag{9.18}$$

现在我们在此基础上从概率分布的角度来解释PCA。每一个数据点$\boldsymbol{x}_i$看作是从一个未知的分布$P_{\boldsymbol{\theta}_i}$的随机采样，其中$P_{\boldsymbol{\theta}}$表示一个均值为$\boldsymbol{\theta} \in \mathbb{R}^D$的高斯分布。而PCA的目的是找到一组对应的参数$\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_n$使得原始数据的似然(Likelihood)最大，且这些参数都存在于一个更低维度的空间中。从这个角度看，真实数据点$\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_n$ 被噪声(Noise)污染后分别成为了原始数据点$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$，而PCA所做的事情是在假设噪声服从高斯分布的条件下

找到这些真实数据点。如果我们对高斯模型进行极大似然估计(MLE)，并将MLE等价地转化为负Log似然函数(*Negative Log Likelihood, NLL*)，忽略掉常数项之后将会得到

$$\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i-\boldsymbol{\theta}_i\|^2 \tag{9.19}$$

上式(9.19)其实就是(9.18)式，二者等价。

对于数据点$\boldsymbol{x}_i$是连续变量的情况，我们可以使用经典PCA，并通过上述三种方式进行解释。然而当数据是二值，整型，或者非负等时候，高斯假设就不再合适了。如果在这些情况下还想继续使用PCA，那就要对经典PCA算法进行推广了。

9.2 PCA在指数分布族上的推广

我们现在面临的问题是，在数据$\boldsymbol{x}_i$是非连续变量的情况时如何进行PCA。这与本材料第二部分开头我们所面对的问题——如何对非连续的期望输出进行线性回归，是非常相似的。我们利用指数分布族，成功地把只能用于连续变量的线性回归推广到了可用于期望输出服从任意指数分布的广义线性模型(GLM)。现在让我们对PCA做同样的事情。

在开始推导之前，为了简化运算，让我们对指数分布族进行一些限定。在第二部分GLM中，我们用到了指数分布族的Natural Form通式(2.1)，现在给出指数分布族中专门针对GLM的一个子集——自然指数分布族(*Natural Exponential Family*)。

9.2.1 自然指数分布族

一般地，指数分布族可以表示成如下形式：

$$P(x;\theta)=P_0(x)\exp(\eta(\theta)T(x)-G(\theta)) \tag{9.20}$$

其中θ为参数，$P_0(x)$和$G(\theta)$为各自的已知函数，$\eta(\theta)T(x)$为参数和变量的函数的乘积。而在其子集——自然指数分布族中，$\eta(\theta)$和$T(x)$皆为恒等函数(*Identity Function*)，则自然指数分布族可以表示为

$$P(x;\theta)=P_0(x)\exp(\theta x-G(\theta)) \tag{9.21}$$

对(9.21)进行一点点变换：

$$P(x;\theta)=\frac{P_0(x)e^{\theta x}}{e^{G(\theta)}}$$

上式两边对所有的x求和

$$\begin{aligned}\sum_x P(x;\theta) &= \sum_x \frac{P_0(x)e^{\theta x}}{e^{G(\theta)}} \\ 1 &= \frac{\sum_x P_0(x)e^{\theta x}}{e^{G(\theta)}} \\ e^{G(\theta)} &= \sum_x P_0(x)e^{\theta x} \\ G(\theta) &= \log \sum_x P_0(x)e^{\theta x} \end{aligned} \tag{9.22}$$

因此可以看出，$G(\theta)$的作用是保证$P(x;\theta)$在x域上求和等于1。如果还记得我们在第二部分GLM中提到过的累积量生产函数(参见(2.5))，则$G(\theta)$又可以看作是分布$P_0(x)$的累积量生成函数。除了以上两点之外，$G(\theta)$ 还有一个对于我们推广PCA最重要的性质。如果对$G(\theta)$进行求导：

$$G'(\theta) = \frac{\sum_x P_0(x)e^{\theta x}\cdot x}{\sum_x P_0(x)e^{\theta x}} \tag{9.23}$$

$$= \frac{\sum_x P_0(x)e^{\theta x}\cdot x}{e^{G(\theta)}} \tag{9.24}$$

$$= \sum_x P_0(x)e^{\theta x - G(\theta)}\cdot x \tag{9.25}$$

其中(9.23)到(9.24)是代入$G(\theta)$(9.22)。(9.25) 表明$G(\theta)$的一阶导数为x在分布$P(x;\theta)$上的期望$\mathbb{E}(x|\theta)$。我们把$G(\theta)$的一阶导数记为$g(\theta)$。

在经典PCA的第三个解释“高斯随机采样”那一节，我们把数据$\boldsymbol{x}_i$看作是服从高斯分布的采样，并且使用欧式距离衡量误差。同时我们也已经指出，之所以使用欧式距离是因为高斯模型的NLL等价于欧式距离(参见(9.19))。上一个小节里我们已经把高斯分布推广到了自然指数分布族，那么对应的误差衡量也将不再是欧式距离，误差衡量同样需要推广。所以在介绍广义PCA之前，我们还需要介绍另外一个概念——Bregman散度(*Bregman Divergence*)。

*9.2.2 Bregman*散度

首先直接给出Bregman散度的定义。
令$F:\Delta\to\mathbb{R}$是一个定义在完备的(Closed)凸(Convex)集$\Delta\in\mathbb{R}$上的可导(Differentiable)且严格凸(Strictly Convex)的函数。则F函数的Bregman散度定义为：

$$B_F(p\|q) = F(p) - F(q) - f(q)(p-q) \tag{9.26}$$

其中$p,q\in\Delta$，$f(x)=F'(x)$为F函数的一阶导。如何直观地理解(9.26) 式呢？对函数$F(p)$在q点进行泰勒展开就清楚了：

$$F(p) = \frac{F(q)}{0!} + \frac{F'(q)}{1!}(p-q) + R_n(p) \tag{9.27}$$

$$R_n(p) = F(p) - F(q) - f(q)(p-q) \tag{9.28}$$

由(9.28)式可以看出，**Bregman散度就是函数$F(p)$在q点进行一阶泰勒展开的余项$R_n(p)$，即函数$F(p)$与该函数自己的线性近似(一阶泰勒展开)之间的差**。

不同的函数具有不同的Bregman散度。例如当$F(x) = \|x\|^2$时，其对应的Bregman散度为$B_F(x\|y) = \|x-y\|^2$ 欧式距离；而当$F(p) = \sum_i p_i \log p_i$负的熵时，其对应的Bregman散度为$B_F(p\|q) = \sum_i p_i \log \frac{p_i}{q_i}$ 即KL散度(*KL-Divergence*)。

现在我们有了自然指数分布族及每一个分布对应的误差衡量(Bregman散度)，可以进行PCA在指数分布族上的推导了。

9.2.3 PCA在指数分布族上的推广(PCA for the Exponential Family)

到这里，最后要解决的问题是：给定一组数据$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ 和某个指数族分布$P(x;\theta)$，找到一组参数$\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_n$，使得数据集的似然函数(Likelihood)最大，即

$$\Theta = \arg\max_{\Theta} \mathcal{L}(P, \mathbf{x}, \Theta) \tag{9.29}$$

$$= \arg\max_{\Theta} \prod_i P(\boldsymbol{x}_i; \boldsymbol{\theta}_i) \tag{9.30}$$

对于$P(\boldsymbol{x};\boldsymbol{\theta})$，取log可得

$$\log P(\boldsymbol{x};\boldsymbol{\theta}) = \log P_0(\boldsymbol{x}) + \boldsymbol{x}\boldsymbol{\theta} - G(\boldsymbol{\theta}) \tag{9.31}$$

此处我们需要新定义一个函数：令$g(\theta) = \mu$，那么定义$G(\theta)$ 的对偶(Dual)函数$F(\mu)$为

$$F(\mu) = \theta\mu - G(\theta) \tag{9.32}$$

于是有：

$$F(x) - B_F(x\|\mu) = F(x) - [F(x) - F(\mu) - F'(\mu)(x-\mu)] \tag{9.33}$$

$$= F(\mu) + \theta(x-\mu) \tag{9.34}$$

$$= x\theta + F(\mu) - \theta\mu \tag{9.35}$$

$$= x\theta - G(\theta) \tag{9.36}$$

其中(9.33)到(9.34)是因为$F'(\mu) = \theta$($F(\mu)$对μ求导)，(9.35)到(9.36)是代入$F(\mu)$的定义(9.32)式。

把$x\theta - G(\theta) = F(x) - B_F(x\|\mu)$ 代入(9.31)得到

$$\log P(\boldsymbol{x};\boldsymbol{\theta}) = \log P_0(\boldsymbol{x}) + F(\boldsymbol{x}) - B_F(\boldsymbol{x}\|\boldsymbol{\mu}) \tag{9.37}$$

于是在$\boldsymbol{\Theta}$的NLL((9.30)取负的log)中代入(9.37) 得到

$$L(\boldsymbol{\Theta}) = -\sum_i \sum_j \log P(\boldsymbol{x}_{ij};\boldsymbol{\theta}_{ij}) \tag{9.38}$$

$$= C(\boldsymbol{x}) + \sum_i \sum_j B_F(\boldsymbol{x}_{ij}\|g(\boldsymbol{\theta}_{ij})) \tag{9.39}$$

$$= C(\boldsymbol{x}) + B_F(\boldsymbol{x}\|g(\boldsymbol{\Theta})) \tag{9.40}$$

到此，我们的目标函数为

$$\boldsymbol{\Theta}_{pca} = \arg\min_{\boldsymbol{\Theta}} B_F(\boldsymbol{x}\|g(\boldsymbol{\Theta})) \tag{9.41}$$

不要忘记，我们的目标是降维。所以最终的问题是，希望找到$\boldsymbol{\theta}_i$，其尽可能的接近$\boldsymbol{x}_i$，且存在于一个更低维度的空间中。根据本章开始部分介绍的第二种PCA解释“最小重构误差”的思想(参见(9.12))，降维的方法是找到一组基$\mathbf{v}_1,\ldots,\mathbf{v}_d \in \mathbb{R}^D$，并将每一个$\boldsymbol{\theta}_i$表示称为这组集的线性组合$\boldsymbol{\theta}_i = \sum_k a_{ik}\mathbf{v}_k$，且最接近$\boldsymbol{x}_i$。

令$\mathbf{X}$表示$n \times D$的矩阵，其行向量为$\boldsymbol{x}_i$；令$\mathbf{V}$ 表示$d \times D$的矩阵，其行向量为$\mathbf{v}_k$；令$\mathbf{A}$表示$n \times d$的矩阵，其每一个分量为a_{ik}。则$\boldsymbol{\Theta} = \mathbf{AV}$是一个$n \times D$的矩阵，其第$i$行为$\boldsymbol{\theta}_i$。$\boldsymbol{\Theta}$的每一个分量$\boldsymbol{\theta}_{ij}$与$\mathbf{X}$ 的每一个分量$\boldsymbol{x}_{ij}$一一对应，$\boldsymbol{\theta}_{ij}$定义了$\boldsymbol{x}_{ij}$ 的概率。在把$\boldsymbol{\Theta}$矩阵分解成$\mathbf{A}$和$\mathbf{V}$两个矩阵的乘积之后，(9.38) 便表示为

$$L(\mathbf{V},\mathbf{A}) = -\log P(\mathbf{X};\mathbf{A},\mathbf{V}) \tag{9.42}$$

$$= -\sum_i \sum_j \log P(\boldsymbol{x}_{ij};\boldsymbol{\theta}_{ij}) \tag{9.43}$$

$$= C(\boldsymbol{x}) + \sum_i \sum_j (-\boldsymbol{x}_{ij} + G(\boldsymbol{\theta}_{ij})) \tag{9.44}$$

去掉与$\boldsymbol{\theta}$无关的常数项$C(\boldsymbol{x})$并写成Bregman散度(代入(9.36))的形式得到

$$L(\mathbf{V},\mathbf{A}) = \sum_i \sum_j B_F(\boldsymbol{x}_{ij}\|g(\boldsymbol{\theta}_{ij})) \tag{9.45}$$

使用类似Coordinate Descent的思想，固定$\mathbf{A}$求$\mathbf{V}$，之后固定$\mathbf{V}$求$\mathbf{A}$，不断迭代，最终求得$\mathbf{A}$和$\mathbf{V}$。

最后我们再来简单的总结一下：原始数据为$\boldsymbol{x}_i$，我们认为$\boldsymbol{x}_i$服从某个参数未知但形式已知的指数族分布$P(\boldsymbol{x};\boldsymbol{\theta})$，该指数族分布对应一个具体的Bregman散度$B_F(p\|q)$；通过Bregman散度便可以对原始数据做Bregman

投影(*Bregman Projection*)，$\boldsymbol{x}_i$的Bregman投影为$g(\boldsymbol{\theta}_i)$；在投影之前，我们把$\boldsymbol{\theta}_i$表示成一个更低维度空间的基$\boldsymbol{v}_1,\ldots,\boldsymbol{v}_l$的线性组合，其组合系数为$a_{i,1},\ldots,a_{i,l}$，即$\boldsymbol{\theta}_i = \sum_{k=1}^{l} a_{ik}\boldsymbol{v}_k$；最终的目标是找到$\mathbf{A}$和$\mathbf{V}$，使得$\boldsymbol{x}_i$与其Bregman投影的Bregman散度最小，即：

$$(\mathbf{V},\mathbf{A}) = \arg\min_{\mathbf{V},\mathbf{A}} \sum_{i=1}^{n} B_F(\boldsymbol{x}_i \| g(\sum_{k=1}^{l} a_{i,k}\boldsymbol{v}_k)) \tag{9.46}$$

从$\mathbf{V}$的角度看，指数分布族上的PCA是在一个低维空间里寻找一组基(矩阵$\mathbf{V}$)，该组基定义了一个尽可能接近原始数据$\boldsymbol{x}_i$的面(Surface)。设由基$\mathbf{V}$定义的面为$\mathcal{S}(\mathbf{V}) = \{g(\boldsymbol{a}\mathbf{V})|\boldsymbol{a} \in \mathbb{R}^l\}$，则我们寻找的$\mathbf{V}$为：

$$\mathbf{V} = \arg\min_{\mathbf{V}} \sum_{i} \min_{\boldsymbol{q}\in\mathcal{S}(\mathbf{V})} B_F(\boldsymbol{x}_i \| \boldsymbol{q}) \tag{9.47}$$