CH06 逻辑斯谛回归与最大熵模型

前言

章节目录

1. 逻辑斯谛回归模型
    a. 逻辑斯谛分布
    b. 二项逻辑斯谛回归模型
    c. 模型参数估计
    d. 多项逻辑斯蒂回归**模型**
2. 最大熵模型
    a. 最大熵原理
    b. 最大熵模型定义
    c. 最大熵模型学习
    d. 极大似然估计
3. 模型学习的最优化算法
    a. 改进的迭代尺度法
    b. 拟牛顿法

导读

> *在最大熵的通用迭代算法GIS中, E过程就是跟着现有的模型计算每一个特征的数学期望值, M过程就是根据这些特征的数学期望值和实际观测值的比值, 调整模型参数. 这里最大化的目标函数是**熵函数**.*
>
> *--吴军, 数学之美 P_{244}*

这里的**熵函数**是条件熵.

- 这一章放在决策树后面，可能就因为熵的概念，对比前面CH05部分理解对应的损失函数，发现其中的区别和联系。
- LR做二分类的时候， $\mathcal{Y}=\{0,1\}$，在涉及到综合考虑各种模型损失函数作为0-1损失上界存在的情况中， $\mathcal{Y}=\{-1,+1\}$，这时方便使用函数间隔$yf(x)$
- 本章从逻辑斯谛分布开始，在CH04的时候，应该熟悉狄利克雷分布和高斯分布，对于离散和连续型的情况应该熟悉这两个分布，这样在这一章看到逻辑斯谛分布的时候会更适应。在书上有这样一句

  > *二项逻辑斯谛回归模型是一种分类模型，由条件概率分布 $P(Y|X)$表示，形式为参数化的逻辑斯谛分布。*

  ~~这一句是这两小节唯一的联系，可能不是很好理解。~~ 分类问题，可以表示成one-hot的形式，而one-hot可以认为是概率的一种表达，只是很确定的一种概率表达。而最大熵模型，是一种不确定的概率表达，其中这个概率，是一个条件概率，是构建的特征函数生成的概率。他们之间的关系有点像hard 和 soft，类似的思想还有kmeans和GMM之间的关系。
  因为书中第四章并没有讲到高斯朴素贝叶斯(GNB)，有GNB做类比，这里可能更容易理解一点，这里重新推荐一下第四章的参考文献 1[1]，配合理解NB和LR的关系。
- 在模型参数估计的部分用到了π，这个应该联想到狄利克雷分布
- 关于NB和LR的对比，Ng也有一篇文章[2]
- 平方误差经过Sigmoid之后得到的是非凸函数

- 书中LR篇幅不大， 注意这样一句， 在逻辑斯谛回归模型中，输出Y=1的对数几率是输入x的线性函数。或者说，输出Y=1的对数几率是由输入x的线性函数表示的模型，及逻辑斯谛回归模型。
- LR 和 Maxent什么关系？有人说明了这两个是等价的。另外也有说在NLP里LR叫做Maxent。Maxent更多的是从信息的角度来分析这个算法。
- 书中没有直接给出LR的损失函数，在CH12中有提到LR的损失函数是**逻辑斯谛损失函数**。如果采用损失函数上界的那类分析方法，定义$\mathcal{Y}=\{+1,-1\}$， 有$\log_2(1+\exp(yf(x)))=\log_2(1+\exp(y(w\cdot x)))$
- 概率的表示方法之一是采用[0,1]之间的数字，也可以使用odds，概率较低的时候用赔率。书中逻辑斯谛分布的定义给的是CDF。
- LR和CRF也可以对比着看，在CRF中， 势函数是严格正的，通常定义为指数函数，概率无向图模型的联合概率分布$P(Y)$可以表示为如下形式：

$$P(Y)=\frac{1}{Z}\prod_C \Psi_C(Y_C)$$
$$Z=\sum_Y\prod_C\Psi_C(Y_C)$$

其实LR 也可以看成这样的形式，对应的Z有如下的表示形式

$$Z=1+\sum_{k=1}^{K-1}\exp(w_k\cdot x)$$

注意， 定义中的$x\in\mathbf{R}^{n+1},w_k\in\mathbf{R}^{n+1}$，这样，上面常数1是考虑到偏置项归一化了
实际上可以写成下面的形式

$$P(Y=k|x)=\frac{\exp(w_k\cdot x)}{\sum_{k=1}^K\exp(w_k\cdot x)}$$

其中第K项对应了偏置，对应的x=1，所以是一个常数w_0，将分子归一化就得到了书中的表达方式，这就是出现个1的原因。

- 对于问题，什么情况下需要归一化，可以考虑下模型是不是要求这个特征要构成一个概率分布。
- 单纯形法是求解线性规划问题的有效方法，最初是为了解决空军军事规划问题，之后成为了解决线性规划问题的有效方法。这个在运筹学中有介绍，比较经典的参考是胡运权的《运筹学》。

模型

> *Logistic regression is a special case of maximum entropy with two labels +1 and −1.*

逻辑斯谛回归模型

这一章的这个部分，可以认为是对第四章的一个补充与延续，只是第四章最后没有说高斯朴素贝叶斯。在《机器学习，周志华》上把这个叫做对数几率回归。

逻辑斯谛分布

注意分布函数中关于位置参数，形状参数的说明，可以大致的和高斯对应理解。

$$F(x) = P(X \leqslant x) = \frac{1}{1 + \exp(-(x - \mu)/\gamma)}$$

关于逻辑斯谛， 更常见的一种表达是Logistic function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

这个函数把实数域映射到(0, 1)区间，这个范围正好是概率的范围， 而且可导，对于0输入， 得到的是0.5，可以用来表示等可能性。

二项逻辑斯谛回归模型

书中给出的定义：

二项逻辑斯谛回归模型是如下的条件概率分布：

$$\begin{aligned} P(Y = 1|x) &= \frac{\exp(w \cdot x)}{1 + \exp(w \cdot x)} \\ &= \frac{\exp(w \cdot x) / \exp(w \cdot x)}{(1 + \exp(w \cdot x)) / (\exp(w \cdot x))} \\ &= \frac{1}{e^{-(w \cdot x)} + 1} \\ P(Y = 0|x) &= \frac{1}{1 + \exp(w \cdot x)} \\ &= 1 - \frac{1}{1 + e^{-(w \cdot x)}} \\ &= \frac{e^{-(w \cdot x)}}{1 + e^{-(w \cdot x)}} \end{aligned}$$

这部分提到了几率，但是怎么就想到几率了。

之前一直不清楚为什么就联想到几率了, 从哪里建立了这种联系. 直到看了Think Bayes[3].

> *One way to represent a probability is with a number between 0 and 1,* ***but that's not the only way****. If you have ever bet on a* ***football game or a horse race****, you have probably encountered another representation of probability,*
> *called odds*

这本书有中文版，希望这部分内容的补充能增加一些博彩业的直觉...

写到这里，突然想到一个人: 吴军博士。不记得数学之美中关于LR是如何描述的, 但是觉得能外延阐述几率和概率的这种联系的内容也许会出现在他的某部作品里。于是翻了数学之美。但，并没有。

数学之美中有这样一个公式

$$f(z)=\frac{e^z}{e^z+1}=\frac{1}{1+e^{-z}}$$

然后几率和概率之间的关系有这样一种表达

$$o=\frac{p}{1-p}$$
$$p=\frac{o}{1+o}$$

看上面红色部分, **逻辑斯谛分布**对应了一种**概率**, **几率**为指数形式e^z, z为**对数几率**$logit$.

$$logit(p)=\log(o)=\log\frac{p}{1-p}$$

上面是对数几率的定义， 这里对应了事件， 要么发生， 要么不发生。所以逻辑斯谛回归模型就表示成

$$\log\frac{P(Y=1|x)}{1-P(Y=1|x)}=\log\frac{P(Y=1|x)}{P(Y=0|x)}=w\cdot x$$

上面红色部分留一下，后面推广到多类时候用到。

多项逻辑斯谛回归

假设离散型随机变量Y的取值集合是$1,2,\ldots,K$, 多项逻辑斯谛回归模型是

$$P(Y=k|x)=\frac{\exp(w_k\cdot x)}{1+\sum_{k=1}^{K-1}\exp(w_k\cdot x)},k=1,2,\ldots,K-1$$
$$P(Y=K|x)=\frac{1}{1+\sum_{k=1}^{K-1}\exp(w_k\cdot x)}$$

下面看这个多分类模型怎么来的[4]。

二元推广

计算$K-1$种可能的取值发生的概率相对取值K发生的概率的比值， 假设其取对数的结果是x的线性模型， 有

$$\ln\frac{P(Y=1|x)}{P(Y=K|x)}=w_1\cdot x$$
$$\ln\frac{P(Y=2|x)}{P(Y=K|x)}=w_2\cdot x$$
$$\cdots$$
$$\ln\frac{P(Y=K-1|x)}{P(Y=K|x)}=w_{K-1}\cdot x$$

得到取值$1,2,\ldots,K-1$的概率表示

$$P(Y=1|x)=P(Y=K|x)\exp(w_1\cdot x)$$
$$P(Y=2|x)=P(Y=K|x)\exp(w_2\cdot x)$$
$$\cdots$$
$$P(Y=K-1|x)=P(Y=K|x)\exp(w_{K-1}\cdot x)$$
$$P(Y=k|x)=P(Y=K|x)\exp(w_k\cdot x),k=1,2,\ldots,K-1$$

上面红色部分有点像书上的(6.7)， 又有K种可能取值概率和为1，可以得到下面推导

$$
\begin{aligned}
P(Y=K|x) &= 1-\sum_{j=1}^{K-1}P(Y=j|x) \\
&= 1-P(Y=K|x)\sum_{j=1}^{K-1}\exp(w_j\cdot x) \\
&= \frac{1}{1+\sum_{j=1}^{K-1}\exp(w_j\cdot x)}
\end{aligned}
$$

所以之前红色部分的表达可以表示为

$$
\begin{aligned}
P(Y=k|x) &= P(Y=K|x)\exp(w_k\cdot x), k=1,2,\dots,K-1 \\
&= \frac{1}{1+\sum_{j=1}^{K-1}\exp(w_j\cdot x)}\exp(w_k\cdot x), k=1,2,\dots,K-1 \\
&= \frac{\exp(w_k\cdot x)}{1+\sum_{j=1}^{K-1}\exp(w_j\cdot x)}, k=1,2,\dots,K-1
\end{aligned}
$$

这里公式和书上有点区别， 求和的用了j表示， 感觉不太容易造成误解。

对数线性模型

假设归一化因子Z， 有如下关系

$$
\begin{aligned}
\ln(ZP(Y=k|x)) &= w_k\cdot x, k=1,2,\dots,K \\
P(Y=k|x) &= \frac{1}{Z}\exp(w_k\cdot x), k=1,2,\dots,K
\end{aligned}
$$

又对所有的$P(Y=k|x)$可以形成概率分布， 有

$$
\begin{aligned}
\sum_{k=1}^{K}P(Y=k|x) &= 1 \\
&= \sum_{k=1}^{K}\frac{1}{Z}\exp(w_k\cdot x) \\
&= \frac{1}{Z}\sum_{k=1}^{K}\exp(w_k\cdot x)
\end{aligned}
$$

得到

$$
Z=\sum_{k=1}^{K}\exp(w_k\cdot x)
$$

所以

$$
P(Y=k|x)=\frac{1}{Z}\exp(w_k\cdot x)=\frac{\exp(w_k\cdot x)}{\sum_{k=1}^{K}\exp(w_k\cdot x)}, k=1,2,\dots,K
$$

上面这个叫Softmax，针对多项的情况也叫Softmax Regression。

模型参数估计

通过监督学习的方法来估计模型参数[这部分不完整]。

Logistic Regression

参数估计这里， 似然函数书中的表达

$$\prod_{i=1}^{N}[\pi(x_i)]^{y_i}[1-\pi(x_i)]^{1-y_i}$$

这里利用了$y_i \in \{0,1\}$这个特点

更一般的表达

$$\prod_{i=1}^{N}P(y_i|x_i,W)$$

使用对数似然会更简单， 会将上面表达式的连乘形式会转换成求和形式。对数函数为单调递增函数， 最大化对数似然等价于最大化似然函数。

$$\begin{aligned}\log\prod_{i=1}^{N}[\pi(x_i)]^{y_i}[1-\pi(x_i)]^{1-y_i} &= \sum_{i=1}^{N}[y_i\log(\pi(x_i))+(1-y_i)\log(1-\pi(x_i))] \\ &= \sum_{i=1}^{N}[y_i\log(\frac{\pi(x_i)}{1-\pi(x_i)})+\log(1-\pi(x_i))] \\ &= \sum_{i=1}^{N}[y_i(w\cdot x_i)-\log(1+\exp(w\cdot x_i))]\end{aligned}$$

好像不用这样麻烦，似然函数表示为

$$\prod_{i=1}^{N}P(y_i|x_i,W)=\prod_{i=1}^{N}\frac{(\exp(w\cdot x_i))^{y_i}}{1+\exp(w\cdot x_i)}$$

使用对数技巧

$$\sum_{i=1}^{N}\log\frac{\exp(w\cdot x_i)}{1+\exp(w\cdot x_i)}=\sum_{i=1}^{N}[y_i(w\cdot x_i)-\log(1+\exp(w\cdot x_i))]$$

Softmax Regression

多类分类的情况这个表达式是什么样的？感觉不能用0，1这样的技巧了。

$$\prod_{i=1}^{N}P(y_i|x_i,W)=\prod_{i=1}^{N}\prod_{l=1}^{K}\left(\frac{\exp(w_k\cdot x_i)}{\sum_{k=1}^{K}\exp(w_k\cdot x_i)}\right)^{I(y_i=l)}$$

但是可以用指示函数。

最大熵模型

概念

逻辑斯谛回归模型和最大熵模型，既可以看作是概率模型，又可以看作是非概率模型。

信息量

信息量是对信息的度量, PRML中有关于信息量的讨论, 信息是概率的单调函数.

$h(x) = -\log_2 p(x)$, 符号保证了非负性. 低概率事件对应了高的信息量. 对数底选择是任意的, 信息论里面常用2, 单位是比特.

- 信息和概率的关系参考PRML中1.6节信息论部分的描述.

> *如果我们知道某件事件一定会发生, 那么我们就不会接收到信息.*
> *于是, 我们对于信息内容的度量将依赖于概率分布$p(x)$*
>
> *如果我们有两个不相关的事件x, y, 那么我们观察到两个事件同时发生时获得的信息应该等于观察到事件各自发生时获得的信息之和, 即*
> *$h(x,y) = h(x) + h(y)$, 这两个不相关的事件是独立的, 因此*
> *$p(x,y) = p(x)p(y)$*
>
> ***根据这两个关系**, 很容易看出$h(x)$一定与$p(x)$的对数有关. 所以有*
>
> $h(x) = -\log_2 p(x) = \log_2 \frac{1}{p(x)}$
>
> - *负号确保了信息非负*
> - *低概率事件x对应了高的信息.*

熵和概率

熵可以从随机变量状态需要的平均信息量角度理解, 也可以从描述统计力学中无序程度的度量角度理解.

关于熵, 条件熵, 互信息, 这些内容在第五章5.2节有对应的描述.

> *下面看下信息熵在PRML中的表达*
>
> *假设一个发送者想传输一个随机变量x的值给接受者. 在这个过程中, 他们传输的平均信息量可以通过求**信息**$h(x)$**关于概率分布**$p(x)$**的期望**得到.*
>
> *这个重要的量叫做随机变量x的**熵***

Venn图辅助理解和记忆, 这个暂时不画, 下面考虑下为什么Venn图能帮助理解和记忆?

因为熵的定义把连乘变成了求和, 对数的贡献. 这样可以通过集合的交并来实现熵之间关系的理解.

1. **概率** $\sum_{i=1}^{n} p_i = 1 \; p \in [0,1]$
2. **熵** $Ent(D) \in [0, \log_2 |\mathcal{Y}|]$, 熵可以大于1。熵是传输一个随机变量状态值所需的比特位**下界**(信息论角度的理解)，也叫香农下界。
3. **信息熵**是度量样本集合纯度最常用的一种指标。

   $$Ent(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$$

   - if $p = 0$，then $p \log_2 p = 0$

- $Ent(D)$越小，D的纯度越高。非均匀分布比均匀分布熵要小。
- 熵衡量的是不确定性，概率描述的是确定性，其实确定性和不确定性差不多。

4. **联合熵(相当于并集)**

   $H(X,Y)=H(X)+H(Y|X)=H(Y)+H(X|Y)=H(X|Y)+H(Y|X)+I(X;Y)$

   这个通过Venn应该是相对容易记忆，是不是容易理解这个。

   如果X和Y独立同分布，联合概率分布$P(X,Y)=P(X)P(Y)$

5. **条件熵**

   条件熵是最大熵原理提出的基础，最大的是条件熵，这个在书中有写(定义6.3)

   条件熵衡量了条件概率分布的均匀性

   最大熵，就是最大这个**条件熵**

   find

   $$\begin{aligned} p^* &= \arg\max_{p\in\mathcal{C}} H(p) \\ &= \arg\max_{p\in\mathcal{C}}(-\sum_{x,y}\tilde{p}(x)p(y|x)\log p(y|x)) \end{aligned}$$

   接下来的概念，把熵的思想应用在模式识别问题中。

6. **互信息**

   互信息(mutual information)，对应熵里面的交集，常用来描述差异性

   一般的，熵$H(Y)$与条件熵$H(Y|X)$之差称为互信息。注意一下，这里第五章中用到了$H(D,A)$ 可以对应理解下。

   a. Feature Selection
   b. Feature Correlation，刻画的是相互之间的关系。 **相关性主要刻画线性，互信息刻画非线性**

7. **信息增益**

   这个对应的是第五章的内容，决策树学习应用信息增益准则选择特征。

   $$g(D,A)=H(D)-H(D|A)$$

   信息增益表示得知X的信息而使类Y的信息的不确定性减少的程度。

   在决策树学习中，信息增益等价于训练数据集中类与特征的互信息。

8. **相对熵 (KL 散度)**

   相对熵(Relative Entropy)描述差异性，从分布的角度描述差异性，可用于度量两个概率分布之间的差异。

   KL散度不是一个度量，度量要满足交换性。

   KL散度满足非负性。

   > *考虑由$p(x,y)$给出的两个变量x和y组成的数据集。如果变量的集合是独立的，那么他们的联合分布可以分解为边缘分布的乘积$p(x,y)=p(x)p(y)$*
   >
   > *如果变量不是独立的，那么我们可以通过考察**联合分布**与**边缘分布乘积**之间的KL散度来判断他们是否"接近"于相互独立。*
   >
   > $I(x,y)=KL(p(x,y)|p(x)p(y))=-\iint p(x,y)\ln\left(\frac{p(x)p(y)}{p(x,y)}\right)$
   >
   > *这被称为变量x和变量y之间的互信息.*

> *--PRML 1.6.1*

注意这里，参考下第五章中关于互信息的描述

> *决策树学习中的信息增益等价于训练数据集中类与特征的互信息*

注意这里面类Y， 特征X。

互信息和条件熵之间的关系

$$I(x,y) = H(X) - H(x|y) = H(y) - H(y|x)$$

可以把互信息看成由于知道y值而造成的x的不确定性的减小(反之亦然)。*这个就是信息增益那部分的解释。*

9. **交叉熵**

   刻画两个分布之间的差异

$$\begin{aligned} CH(p,q) &= -\sum_{i=1}^{n} p(x_i)\log q(x_i) \\ &= -\sum_{i=1}^{n} p(x_i)\log p(x_i) + \sum_{i=1}^{n} p(x_i)\log p(x_i) - \sum_{i=1}^{n} p(x_i)\log q(x_i) \\ &= H(p) + \sum_{i=1}^{n} p(x_i)\log\frac{p(x_i)}{q(x_i)} \\ &= H(p) + KL(p||q) \end{aligned}$$

   CNN时候常用

对于各种熵的理解，是构建后面的目标函数的基础。

最大熵原理

最大熵原理(Maxent principle)是**概率模型**学习的一个准则。

书中通过一个例子来介绍最大熵原理，下面引用一下文献中关于这个例子的总结。

> *Model all that is known and assume nothing about that which is unknown. In other words, given a collection of facts, choose a model which is consistent with all the facts, but otherwise as uniform as possible.*
>
> *-- Berger, 1996*

书中关于这部分的总结如下：**满足约束条件下求等概率的方法估计概率分布**

关于最大熵原理有很多直观容易理解的解释，比如Berger的例子，比如吴军老师数学之美中的例子。

最大熵原理很常**见**，很多原理我们都一直在用，只是没有上升到理论的高度。

等概率表示了对事实的无知，因为没有更多的信息，这种判断是合理的。

最大熵原理认为要选择的概率模型首先必须满足**已有的事实**，即**约束条件**

最大熵原理根据已有的信息（**约束条件**），选择适当的概率模型。

最大熵原理认为不确定的部分都是等可能的，通过熵的最大化来表示**等可能性**。

最大熵的原则，承认已有的，且对未知无偏

最大熵原理并不直接关心特征选择，但是特征选择是非常重要的，因为约束可能是成千上万的。

最大熵原理几何解释

这部分书中只描述了模型空间$\mathcal{P}$，两个约束C_1和C_2是**一致性**约束的情况。

在Berger 1996里面有展开这部分，分了四个图，分别讨论了

1. 概率模型空间$\mathcal{P}$
2. 单一约束C_1
3. 一致性(consistent)约束C_1和C_2，这种情况下模型唯一确定$p = C_1 \bigcap C_2$
4. 非一致性(inconsistent)约束C_1和C_3，这种情况下没有满足约束条件的模型。

特征与约束条件

关于特征和约束，Berger有他的阐述

> *指示函数*
>
> $$f(x,y) = \begin{cases} 1 & if\ y = en\ and\ April\ follows\ in \\ 0 & otherwise \end{cases}$$

上面这个f直接引用自Berger的说明，原来的例子是英语in到法语的翻译。

这里面f就是**特征函数**，或者**特征**。

定义一个期望，如果是二值函数的话，就相当于计数。通过样本得到的这个统计。但是样本是有限的，并不是一个真实的分布， 所以叫经验分布，如果我们拿到的这个模型能够表示实际的分布，那么就可以假设经验分布和真实分布是相等的。这个，就是**约束方程**，或者**约束**。

一般模型的**特征**是关于x的函数，最大熵模型中的特征函数，是关于x和y的函数。注意理解$f(x)$与$f(x,y)$的区别。 关于特征函数可以参考条件随机场中例11.1关于特征部分的内容增强理解。

模型

假设分类模型是一个条件概率分布，$P(Y|X), X \in \mathcal{X} \subseteq \mathbf{R}^n$

给定一个训练集 $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$

N是训练样本容量，$x \in \mathbf{R}^n$

联合分布$P(X,Y)$与边缘分布P(X)的经验分布分别为$\tilde{P}(X,Y)$和 $\tilde{P}(X)$

$$\widetilde{P}(X=x,Y=y)=\frac{\nu(X=x,Y=y)}{N}$$
$$\widetilde{P}(X=x)=\frac{\nu(X=x)}{N}$$

上面两个就是不同的数据样本，在训练数据集中的比例。

如果增加n个**特征函数**,就可以增加n个**约束条件**，特征也对应增加了一列。

假设满足所有约束条件的模型集合为

$\mathcal{C}\equiv\ \{P\in\mathcal{P}|E_P(f_i)=E_{\widetilde{P}}(f_i),i=1,2,\dots,n\}$

定义在条件概率分布$P(Y|X)$上的条件熵为

$$H(P)=-\sum_{x,y}\widetilde{P}(x)P(y|x)\log P(y|x)$$

则模型集合$\mathcal{C}$中条件熵$H(P)$最大的模型称为最大熵模型，上式中对数为自然对数。

特征函数$f(x,y)$关于经验分布$\widetilde{P}(X,Y)$的期望值用$E_{\widetilde{P}}(f)$表示

$$E_{\widetilde{P}}(f)=\sum_{x,y}\widetilde{P}(x,y)f(x,y)$$

特征函数$f(x,y)$关于模型$P(Y|X)$与经验分布$\widetilde{P}(X)$的期望值,用$E_P(f)$表示

$$E_P(f)=\sum_{x,y}\widetilde{P}(x)P(y|x)f(x,y)$$

如果模型能够获取训练数据中的信息，那么就有

$$\widetilde{P}(x,y)=P(y|x)\widetilde{P}(x)$$

就可以假设这两个期望值相等，即

$$E_P(f)=E_{\widetilde{P}}(f)$$

上面这个也是约束方程

算法实现

特征提取原理

通过对已知训练集数据的分析，能够拿到联合分布的经验分布和边缘分布的经验分布。

特征函数用来描述$f(x,y)$描述输入x和输出y之间的某一事实。

$$f(x,y)=\begin{cases}1 & x\text{与}y\text{满足某一事实}\\0 & \text{否则}\end{cases}$$

这里，满足的事实，可以是in，显然，特征函数可以自己定义，可以定义多个，~~这些就是约束~~

之前理解的不对，看前面有描述特征和约束的关系。

预测分类原理

这里面重复一下书中的过程，在$L(P,w)$对P求导并令其为零的情况下解方程能拿到下面公式

$$P(y|x)=\exp\left(\sum_{i=1}^{n}w_i f_i(x,y)+w_0-1\right)=\frac{\exp\left(\sum_{i=1}^{n}w_i f_i(x,y)\right)}{\exp\left(1-w_0\right)}$$

书中有提到因为$\sum_y P(y|x)=1$，然后得到模型

$$P_w(y|x)=\frac{1}{Z_w(x)}\exp\sum_{i=1}^{n}w_i f_i(x,y)$$

$$Z_w(x)=\sum_y\exp\sum_{i=1}^{n}w_i f_i(x,y)$$

注意这里面Z_w是归一化因子。

这里面并不是因为概率为1推导出了Z_w的表达式，而是因为Z_w的位置在分母，然后对应位置$\exp(1-w_0)$也在分母，凑出来这样一个表达式，意思就是遍历y的所有取值，求分子表达式的占比。

综上，如果$f_i(x,y)$只检测是不是存在这种组合，那么概率就是*归一化的*出现过的特征，系数求和再取e指数。

最大熵模型的学习

最大熵模型的学习过程就是求解最大熵模型的过程。

最大熵模型的学习可以形式化为约束最优化问题。

$$\min_{P\in\mathcal{C}}-H(P)=\sum_{x,y}\tilde{P}(x)P(y|x)\log P(y|x) \tag{6.14}$$

$$s.t.\ E_P(f_i)-E_{\tilde{P}}(f_i)=0,i=1,2,\ldots,n \tag{6.15}$$

$$\sum_y P(y|x)=1 \tag{6.16}$$

可以通过例6.2 来理解最大熵模型学习的过程，例6.2 考虑了两种约束条件，这部分内容可以通过python符号推导实现，下面代码整理整个求解过程。

例6.2

一个约束条件

```
from sympy import *

# 1 constrains
P1, P2, P3, P4, P5, w0, w1, w2 = symbols("P1, P2, P3, P4, P5, w0, w1, w2", real=True)
L = P1 * log(P1) + P2 * log(P2) + P3 * log(P3) + P4 * log(P4) + P5 * log(P5) \
```

```
    + w0 * (P1 + P2 + P3 + P4 + P5 - 1)
P1_e = (solve(diff(L, P1), P1))[0]
P2_e = (solve(diff(L, P2), P2))[0]
P3_e = (solve(diff(L, P3), P3))[0]
P4_e = (solve(diff(L, P4), P4))[0]
P5_e = (solve(diff(L, P5), P5))[0]
L = L.subs({P1: P1_e, P2: P2_e, P3: P3_e, P4: P4_e,
P5: P5_e})
w = (solve([diff(L, w0)], [w0]))[0]
P = [P1_e.subs({w0: w[0]}),
     P2_e.subs({w0: w[0]}),
     P3_e.subs({w0: w[0]}),
     P4_e.subs({w0: w[0]}),
     P5_e.subs({w0: w[0]})]
P
```

两个约束条件

```
# 2 constrains
P1, P2, P3, P4, P5, w0, w1, w2 = symbols("P1, P2, P3,
P4, P5, w0, w1, w2",real=True)
L = P1*log(P1) +
P2*log(P2)+P3*log(P3)+P4*log(P4)+P5*log(P5)\
    +w1*(P1+P2-3/10)\
    +w0*(P1+P2+P3+P4+P5-1)
P1_e = (solve(diff(L,P1),P1))[0]
P2_e = (solve(diff(L,P2),P2))[0]
P3_e = (solve(diff(L,P3),P3))[0]
P4_e = (solve(diff(L,P4),P4))[0]
P5_e = (solve(diff(L,P5),P5))[0]
L = L.subs({P1:P1_e, P2:P2_e, P3:P3_e, P4:P4_e,
P5:P5_e})
w = (solve([diff(L,w1),diff(L,w0)],[w0,w1]))[0]
P = [P1_e.subs({w0:w[0], w1:w[1]}),
     P2_e.subs({w0:w[0], w1:w[1]}),
     P3_e.subs({w0:w[0], w1:w[1]}),
     P4_e.subs({w0:w[0], w1:w[1]}),
     P5_e.subs({w0:w[0], w1:w[1]})]
P
```

三个约束条件

```
# 3 constrains
P1, P2, P3, P4, P5, w0, w1, w2 = symbols("P1, P2, P3,
P4, P5, w0, w1, w2",real=True)
L = P1*log(P1) +
P2*log(P2)+P3*log(P3)+P4*log(P4)+P5*log(P5)\
    +w2*(P1+P3-1/2)\
    +w1*(P1+P2-3/10)\
    +w0*(P1+P2+P3+P4+P5-1)
```

```
P1_e = (solve(diff(L,P1),P1))[0]
P2_e = (solve(diff(L,P2),P2))[0]
P3_e = (solve(diff(L,P3),P3))[0]
P4_e = (solve(diff(L,P4),P4))[0]
P5_e = (solve(diff(L,P5),P5))[0]
L = L.subs({P1:P1_e, P2:P2_e, P3:P3_e, P4:P4_e,
P5:P5_e})
w = (solve([diff(L,w2),diff(L,w1),diff(L,w0)],
[w0,w1,w2]))[0]
P = [P1_e.subs({w0:w[0], w1:w[1],w2:w[2]}),
     P2_e.subs({w0:w[0], w1:w[1],w2:w[2]}),
     P3_e.subs({w0:w[0], w1:w[1],w2:w[2]}),
     P4_e.subs({w0:w[0], w1:w[1],w2:w[2]}),
     P5_e.subs({w0:w[0], w1:w[1],w2:w[2]})]
P
```

模型学习

逻辑斯谛回归模型和最大熵模型学习归结为以**似然函数**为**目标函数**的最优化问题，通常通过迭代算法求解。

目标函数

逻辑斯谛回归模型

$$
\begin{aligned}
L(w) &= \sum_{i=1}^{N}[y_i \log \pi(x_i) + (1-y_i)\log(1-\pi(x_i))] \\
&= \sum_{i=1}^{N}[y_i \log \frac{\pi(x_i)}{1-\pi(x_i)} + \log(1-\pi(x_i))] \\
&= \sum_{i=1}^{N}[y_i(w\cdot x_i) - \log(1+\exp(w\cdot x_i)]
\end{aligned}
$$

最大熵模型

$$
\begin{aligned}
L_{\widetilde{P}}(P_w) &= \sum_{x,y}\widetilde{P}(x,y)\log P(y|x) \\
&= \sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^{n}w_i f_i(x,y) - \sum_{x,y}\widetilde{P}(x,y)\log\left(Z_w(x)\right) \\
&= \sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^{n}w_i f_i(x,y) - \sum_{x,y}\widetilde{P}(x)P(y|x)\log\left(Z_w(x)\right) \\
&= \sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^{n}w_i f_i(x,y) - \sum_{x}\widetilde{P}(x)\log\left(Z_w(x)\right)\sum_{y}P(y|x) \\
&= \sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^{n}w_i f_i(x,y) - \sum_{x}\widetilde{P}(x)\log\left(Z_w(x)\right)
\end{aligned}
$$

以上推导用到了$\sum\limits_{y}P(y|x)=1$

1. 逻辑斯谛回归模型与朴素贝叶斯的关系

   这部分内容书中引用了参考文献[4]，这是Mitchell的那本《机器学习》，应该说的是第六章中相关的部分推导。

   注意对应的部分还写了在神经网络中梯度搜索实现似然最大化，给出了结果与BP结果的差异，这部分可以看看。

2. 逻辑斯谛回归模型与AdaBoost的关系

3. 逻辑斯谛回归模型与核函数的关系

其他

课后习题的第一个题目提到了**指数族**(Exponential family)分布，这个概念在PRML中有单独的章节进行阐述。

扩展一下指数族分布：

正态分布
两点分布
二项分布
泊松分布
伽马分布

大部分的算法实现，其实都是在刷权重，记得有个同学的昵称就是“一切源于wx+b”。另外，**其实算法能跑起来，能预测，并不能说明算法实现的正确的，或者说并不一定是最优的。** 但是其实这种情况也比较常见， 有的时候没有条件去创造一个专门的工具来解决问题的时候，或者没有更好的工具解决问题的时候， 我们会选择能解决部分问题，或者能解决问题的工具来**对付**

代码实现

关于代码实现，网上看似众多的版本，应该基本上都源自最早15年的一份GIS的程序。

无论怎样，这些代码的实现，都会有助于对Maxent的理解。推荐后面参考文献[1]

李航老师在本章给出的参考文献中[1, 2]是Berger的文章。

Demo

这部分代码没有LR的说明。

> *代码来源*: *https://vimsky.com/article/776.html*
> *相关公式*: *https://vimsky.com/article/714.html*

提几点:

1. 代码参考文献可以看berger的文章，公式编号基本对应。
2. 这份代码用defaultdict实现了稀疏存储。
3. 如果$f(x,y)$只是判断(x,y)在特征中出现的指示函数，那么特征可以简单的表示为(x,y)，这样给定一份含有标签的数据集，特征的数量就是$m \times n$ 其中m是标签的数量，n是词表大小。注意书中注释过，$f(x,y)$可以是任意实值函数。
4. 这份代码思路很清晰，$(E_{\tilde{p}}, Z_x \Rightarrow P(y|x) \Rightarrow E_p) \Rightarrow \delta$，具体参考书中公式6.22，6.23，6.34
5. 体会一下在做直方图的时候，对于同一个样本，同样的特征出现多次的贡献是一样的。
6. 在未知的情况下，输出结果等概率。

Maxent

> *参考链接*: *https://github.com/WenDesi/lihang_book_algorithm/tree/master/maxENT*

本来是想在这个代码的基础上更改，但是代码分解的不是非常容易理解。改来改去基本上面目全非了。保留链接以示感谢。博主写了一系列代码，至少有个成体系的参考。

提几点:

1. 没有用稀疏存储，所以，矩阵中会有很多零。需要除零错误处理

```
with np.errstate(divide='ignore',
invalid='ignore'):
    tmp = np.true_divide(self.EPxy, self.EPx)
    tmp[tmp == np.inf] = 0
    tmp = np.nan_to_num(tmp)
```

   分子分母都是0对应nan，分母为0对应inf

2. 尝试了三种数据，可以通过命令行参数实现数据选择。
   - Demo中用到的data
   - train_binary.csv 这份数据的来源是参考链接中的，只考虑了0，1两种数据，标签少。
   - sklearn中的digits，标签全，但是8x8大小，比mnist少。其实8x8也不是一个非常小的图了，因为数字相对简单一点，用OpenCV做级联分类器的训练的时候，建议的图片大小是20x20，或者40x40，或者60x60不要太大
3. 书中有一个地方还是不理解的，提到了$f^{\#}$是不是常数的问题。
4. 没有采用字典的方式实现稀疏存储，但是numpy的数组操作还是很便捷的，后面有空评估一下存储和计算资源的消耗情况。
5. 大多数算法都是在刷权重，考虑哪些量(特征)可以用，哪些方法(算法)可以让权重刷的更合理，哪些方法(优化方法)能刷的更快。

Mnist

有同学问LR实现中的GD，才发现那段代码不是很好读。而且，用到的train.csv已不在。

加了一个mnist_sample.py从Lecun那里下载数据，并按照类别采样300条。用来完成LR的Demo。

有些程序的问题，配合数据来理解。通常用到label乘法都是利用了label的符号，或者one-hot之后为了取到对应的类别的值。

代码更新了下，建议运行logistic_regression.py的时候在注释的位置断点，看下各个数据的shape，希望对理解代码有帮助。

参考

1. Berger,1995, A Brief Maxent Tutorial
2. [数学之美:信息的度量和作用]
3. [数学之美:不要把鸡蛋放在一个篮子里 谈谈最大熵模型]
4. 李航·统计学习方法笔记·第6章 logistic regression与最大熵模型（2）·最大熵模型
5. 最大熵模型与GIS ,IIS算法
6. 关于最大熵模型的严重困惑：为什么没有解析解？
7. 最大熵模型介绍 这个是Berger的文章的翻译.
8. 理论简介 代码实现
9. 另外一份代码
10. 如何理解最大熵模型里面的特征？
11. Iterative Scaling and Coordinate Descent Methods for Maximum Entropy Models
12.
13.
14.
15.
16. https://blog.csdn.net/u012328159/article/details/72155874

↑ top

---

1. Generative and discriminative classifiers: Naive Bayes and logistic regression↩

2. On Discriminative vs. Generative Classifiers: A comparison of Logistic Regression and Naive Bayes↩

3. ThinkBayes↩

4. Multinomial logistic regression↩