人工智能的数学基础

华东师范大学 数学科学学院 黎芳(教授) 2019年10月12日

Chapter 6 逻辑斯谛回归与最大熵模型

Logistic regression and maximum entropy model

Table of Contents

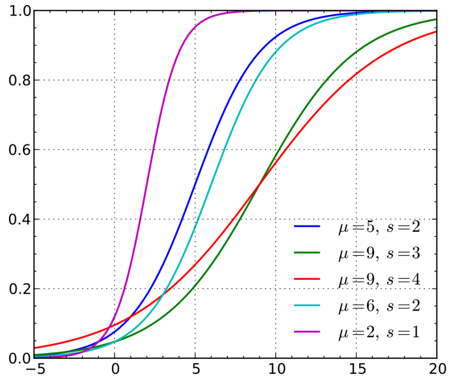
6.1 逻辑斯谛回归模型

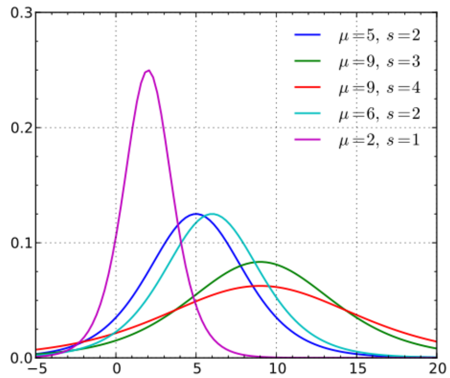
6.1.1 逻辑斯谛分布

逻辑斯谛分布：设X是连续随机变量，X服从逻辑斯谛分布是指X具有下列累积分布函数和密度函数：

$$F(x) = P(X \leq x) = \frac{1}{1 + e^{-(x-\mu)/s}}$$
$$f(x) = F'(x) = \frac{e^{-(x-\mu)/s}}{s(1 + e^{-(x-\mu)/s})^2}$$

μ为位置参数，s为形状参数.





Mean =μ, Variance = $\frac{s^2\pi^2}{3}$.

$$F(-x+\mu)-\frac{1}{2}=-F(x+\mu)+\frac{1}{2}$$

S形曲线，关于$(\mu,1/2)$中心对称.

6.1.2 二项逻辑斯谛回归模型 （binomial **logistic regression** model）

$$P(Y=1|x)=\frac{\exp(w\cdot x+b)}{1+\exp(w\cdot x+b)}$$
$$P(Y=0|x)=\frac{1}{1+\exp(w\cdot x+b)}$$

令 $w=(w^{(1)},w^{(2)},\cdots,w^{(n)},b)^{\mathrm{T}},\quad x=(x^{(1)},x^{(2)},\cdots,x^{(n)},1)^{T}$

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

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

对逻辑斯谛回归而言，

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

6.1.3 模型参数估计

$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\},y_i\in\{0,1\}$.

设 $P(Y=1|x)=\pi(x),\quad P(Y=0|x)=1-\pi(x)$

似然函数为: $\prod_{i=1}^{N}[\pi(x_i)]^{y_i}[1-\pi(x_i)]^{1-y_i}$

对数似然函数为:

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

Example 1: Fit a Logistic Regression Model

Make a logistic binomial model of the probability of smoking as a function of age, weight, and sex, using a two-way interactions model.

Load the `hospital` dataset array.

```
load hospital
dsa = hospital;
```

Specify the model using a formula that allows up to two-way interactions between the variables age, weight, and sex. Smoker is the response variable.

```
modelspec = 'Smoker ~ Age+Weight+Sex';
```

Fit a logistic binomial model.

```
mdl = fitglm(dsa,modelspec,'Distribution','binomial')
```

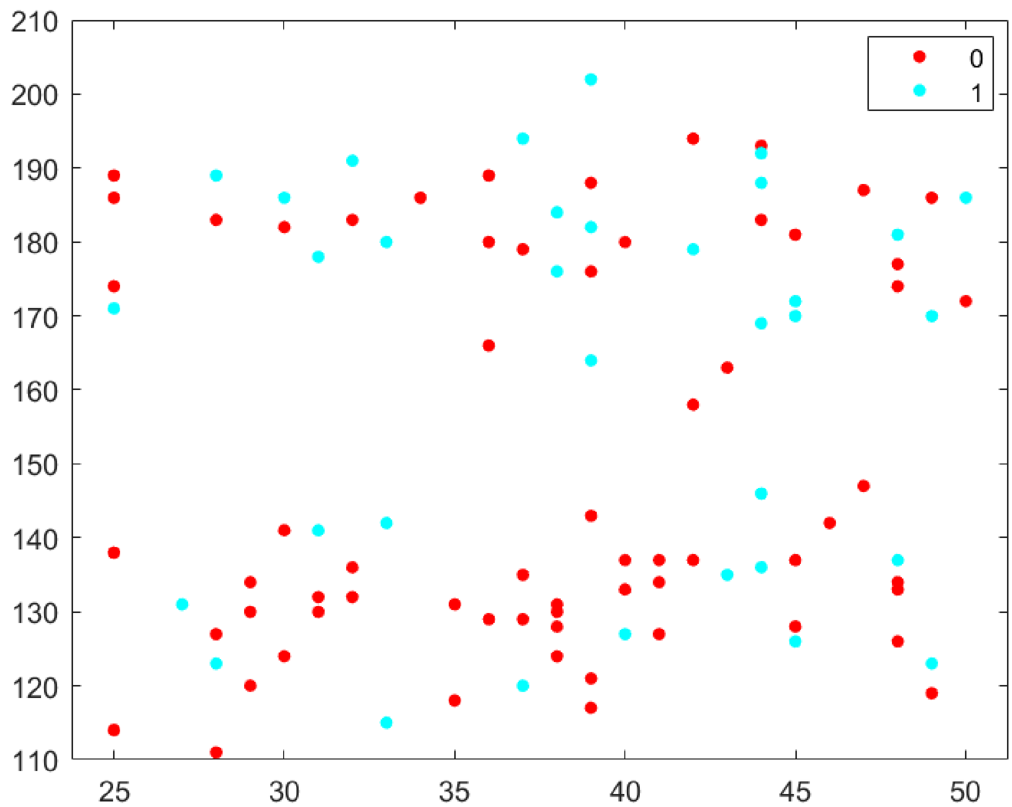
```
mdl =
Generalized linear regression model:
    logit(Smoker) ~ 1 + Sex + Age + Weight
    Distribution = Binomial

Estimated Coefficients:
                   Estimate       SE        tStat      pValue
                   ________    ________    _______    _______

    (Intercept)     -3.1266      3.4864    -0.8968    0.36983
    Sex_Male        0.32242      1.3163    0.24495     0.8065
    Age            0.013086    0.030277    0.43222    0.66558
    Weight         0.011524     0.02502     0.4606    0.64509


100 observations, 96 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 4.94, p-value = 0.177
```

All of the p-values (under pValue) are large. This means none of the coefficients are significant. The large p-value for the test of the model, 0.535, indicates that this model might not differ statistically from a constant model.

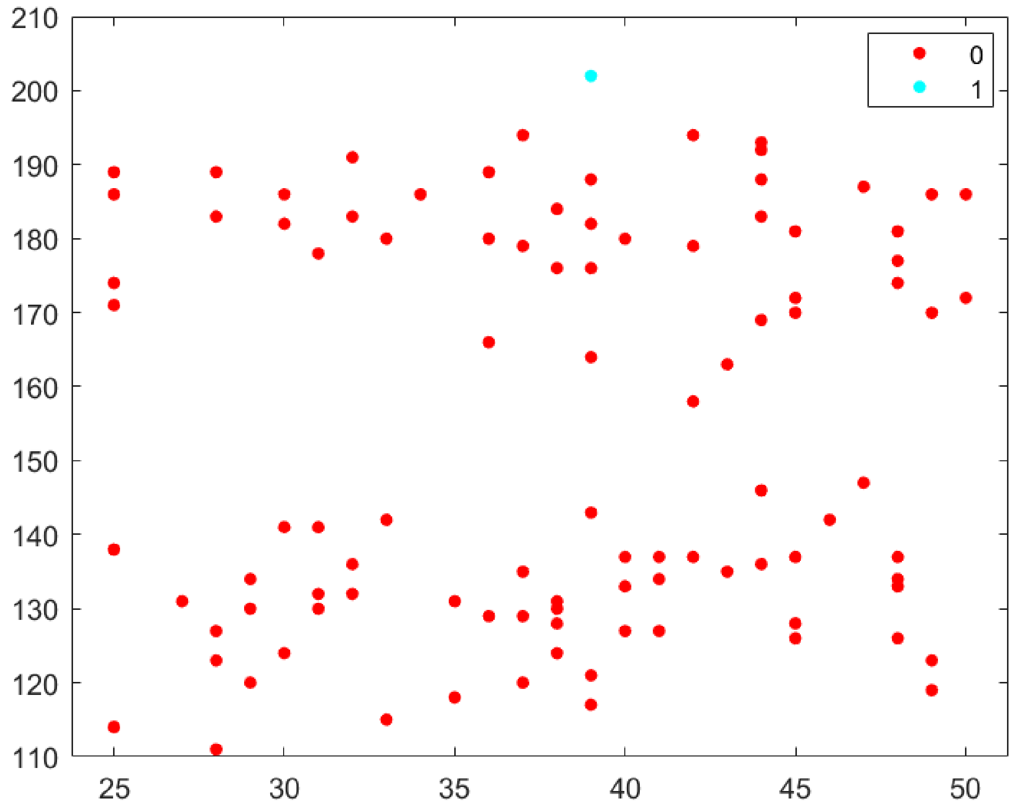
```
predicted_probability = predict(mdl,hospital(:,2:4));
predictedlabels = predicted_probability>0.5;
truelabel = hospital.Smoker;
accuracy = mean(truelabel == predictedlabels)
```

```
accuracy = 0.6700
```

```
gscatter(hospital.Age, hospital.Weight,truelabels)
legend('Location','best')
```




```
gscatter(hospital.Age, hospital.Weight,predictedlabels)
legend('Location','best')
```



Example 2: Logistic Regression for Fisher Iris Data Classification

Load the Fisher iris data.

Attribute Information:

X--1. sepal length in cm (花萼长度)

2. sepal width in cm

3. petal length in cm (花瓣长度)

4. petal width in cm

Y--class:

-- Iris Setosa (山鸢尾)

-- Iris Versicolour (杂色鸢尾)

-- Iris Virginica (维吉尼亚鸢尾)

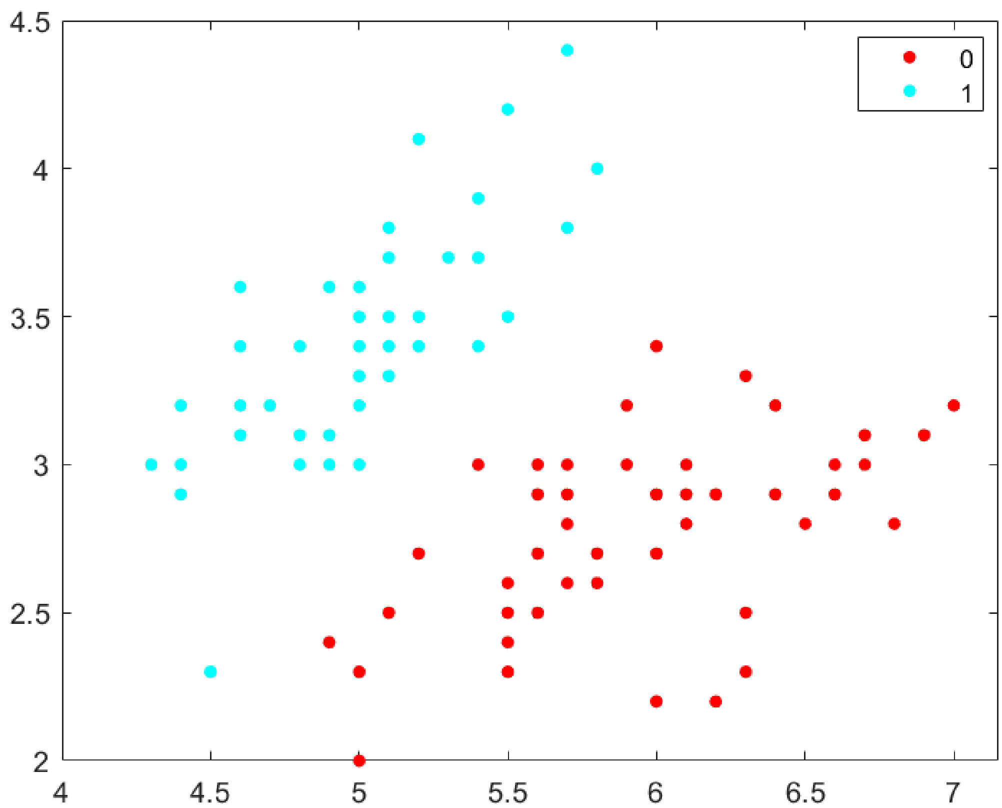
http://archive.ics.uci.edu/ml/datasets/Iris

```
load fisheriris
```

```
X = meas;    % Use all data for fitting
% Y = species; % Response data
X = X(1:100,:);
Y = zeros(100,1);
Y(1:50,:) = 1;
```

2D plot of some attributes

```
x = X(:,1:2);
gscatter(x(:,1),x(:,2),Y)
legend('Location','best')
```



Construct the logistic regression model

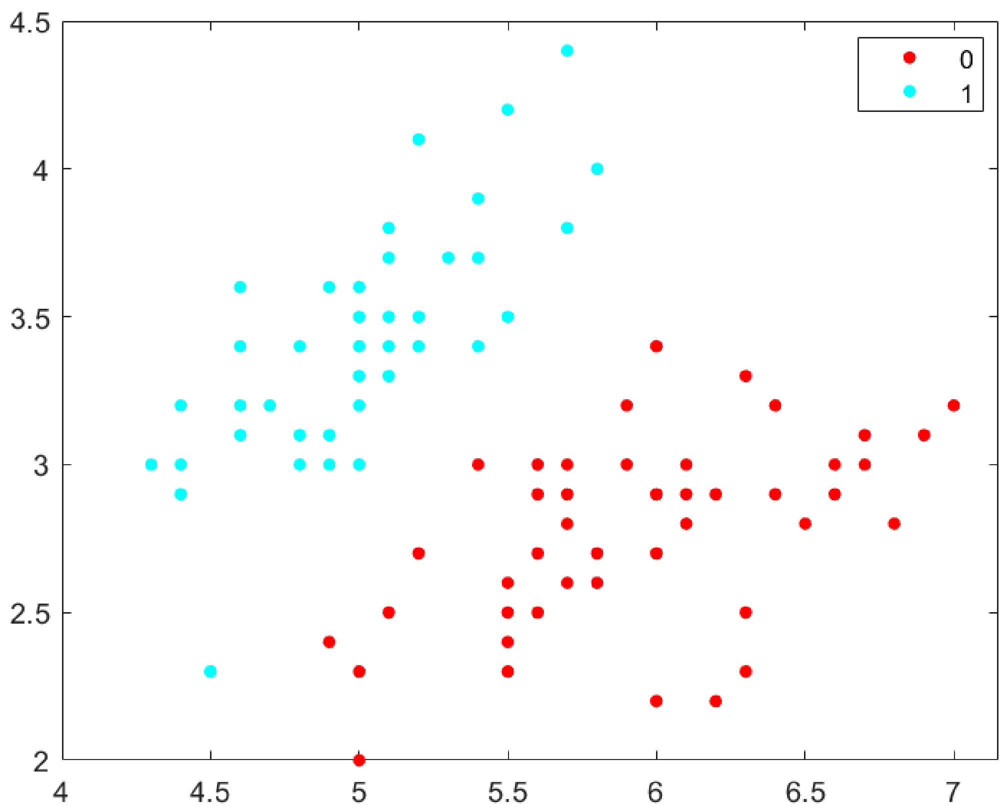
```
Mdl = fitglm(X,double(Y),'linear')
```

```
Mdl =
Generalized linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4
    Distribution = Normal

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    __________
```

```
    (Intercept)       0.6303      0.12551      5.0221    2.3897e-06
    x1               0.02849      0.03368     0.84589       0.39974
    x2                0.1682     0.033178      5.0696    1.9646e-06
    x3              -0.20313     0.040957     -4.9596     3.087e-06
    x4              -0.28785     0.087976     -3.2719     0.0014897


100 observations, 95 error degrees of freedom
Estimated Dispersion: 0.00963
F-statistic vs. constant model: 625, p-value = 2.66e-67
```

Predict the classification of given flowers.

```
predictClass = predict(Mdl,X)>0.5;
error = Y~=predictClass;
gscatter(x(:,1),x(:,2),predictClass)
legend('Location','best')
```




```
accuracy = mean(Y==predictClass)
```

```
accuracy = 1
```

6.1.4 多项逻辑斯谛回归

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

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

$x\in\mathbf{R}^{n+1},w_k\in\mathbf{R}^{n+1}$

6.2 最大熵模型

6.2.1 最大熵原理

假设离散型随机变量X的概率分布是$P(X)$，则

$$H(P)=-\sum_x P(x)\log P(x)\Rightarrow 0\le H(P)\le\log|X|$$

$$\begin{aligned}
&\max_P H(P)=-\sum_x P(x)\log P(x),\ s.t.,\sum_x P(x)=1\\
&\Rightarrow \mathscr{L}=-\sum_x P(x)\log P(x)+\lambda(\sum_x P(x)-1)\\
&\Rightarrow \frac{\partial\mathscr{L}}{\partial P(x)}=-1-\log(P(x))+\lambda=0\\
&\Rightarrow P(x)=\exp(\lambda-1)\\
&\sum_x P(x)-1\Rightarrow |X|\exp(\lambda-1)=1\\
&\Rightarrow P(x)=\frac{1}{|X|}\Rightarrow H(P)\le\log|X|
\end{aligned}$$

最大熵原理:

最大熵原理指出，当我们需要对一个随机事件的概率分布进行预测时，我们的预测应当满足全部已知的条件，而对未知的情况不要做任何主观假设。在这种情况下，概率分布最均匀，预测的风险最小。因为这时概率分布的信息熵最大，所以人们称这种模型叫“最大熵模型”。我们常说，不要把所有的鸡蛋放在一个篮子里，其实就是最大熵原理的一个朴素的说法，因为当我们遇到不确定性时，就要保留各种可能性。说白了，就是要保留全部的不确定性，将风险降到最小。

学习概率模型时，在所有可能的概率模型(分布)中，熵最大的模型是最好的模型，表述为在满足约束条件的模型集合中选取熵最大的模型.

例**6.1** 假设随机变量X有5个取值{A,B,C,D,E}, 估计各个值的概率。

解：满足$P(A)+P(B)+P(C)+P(D)+P(E)=1$

等概率估计：$P(A) = P(B) = P(C) = P(D) = P(E) = \frac{1}{5}$

加入一些先验：

$$P(A) + P(B) = \frac{3}{10}$$
$$P(A) + P(B) + P(C) + P(D) + P(E) = 1$$

于是 $P(A) = P(B) = \frac{3}{20}$，$P(C) = P(D) = P(E) = \frac{7}{30}$

$$P(A) + P(C) = \frac{1}{2}$$
$$P(A) + P(B) = \frac{3}{10}$$

再加入约束：$P(A) + P(B) + P(C) + P(D) + P(E) = 1$

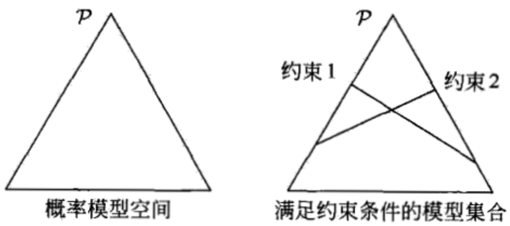


图 6.2 概率模型集合

再举一个例子. 假设我们要学习一个条件概率分布 P(y|x). 举例，x是病人身体指标，体温、血压、血糖，y是各种可能的疾病，可简化为小病、中病、大病三种。

现在，我们有一个样本x1={体温：30，血压：160，血糖：60}，那么P(y|x1)就是一个概率分布，该分布的值就是上面简化的三种，小病、中病、大病。可能的概率分布如下所示：

小病	中病	大病
1/2	1/4	1/4
1/4	1/3	5/12
1/3	1/3	1/3

当然，这样的分布有无数种，上面只是举例说明而已。那么，问题来了，在这无数种概率分布中，哪一个才是好的呢？

为了选出一个好的分布，可以做如下两步：

- 1、看看以往的病例中，指标x1={体温：30，血压：160，血糖：60}和三种病之间的关系，如果没有这样的病例，也就是说我们没有过往的经验可以参考，那么，就直接选一个熵最大的分布就是，也就是上面表格中的第三个分布，因为均匀分布总是同类分布中熵最大的分布。
- 2、如果查看以往病例后，我们得到一个经验，指标x1={体温：30，血压：160，血糖：60}有1/2的概率是小病，于是我们有了一定的经验知识，此时，最好的分布就是符合这个经验知识的前提下，熵最大的分布，显然，第一个分布就是最好的分布。

总结来说，最大熵的思想是，当你要猜一个概率分布时，如果你对这个分布一无所知，那就猜熵最大的均匀分布，如果你对这个分布知道一些情况，那么，就猜满足这些情况的熵最大的分布。

假设我们通过观察这N个样本，发现了一个事实：

当体温小于38，血压小于100，血糖小于30时，总是得小病。这就是一个综合后的先验知识。我们可以据此定义一个特征函数：

f(x,y) = 1 当且仅当 x ={体温小于38，血压小于100，血糖小于30}，y=小病.

将f(x,y)运用到任一个样本（xi，yi）上，我们就可以知道该样本是不是满足上述事实。你可以认为，f(x,y)是对样本是否符合某个事实的判定函数。

6.2.2 最大熵模型的定义

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

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

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:

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

如果模型能够获取训练数据中的信息，那么就可以假设这两个期望值相等，即$E_P(f)=E_{\widetilde{P}}(f)$

最大熵模型定义：假设满足所有约束条件的模型集合为：

$$\mathscr{C}\equiv\left\{P\in\mathscr{P}\,|\,E_P(f_i)=E_{\widetilde{P}}(f_i),\quad i=1,2,\cdots,n\right\}$$

定义在条件概率分布P(Y|X)上的条件熵：

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

则模型集合$\mathscr{C}$中条件熵$H(P)$最大的模型称为最大熵模型.

6.2.3 最大熵模型的学习

最大熵模型的学习等价于约束最优化问题:

$$\begin{aligned}&\min_{P\in C}-H(P)=\sum_{x,y}\widetilde{P}(x)P(y|x)\log P(y|x)\\ &\text{s.t.}\quad E_P(f_i)-E_{\widetilde{P}}(f_i)=0,\quad i=1,2,\cdots,n\\ &\sum_y P(y|x)=1\end{aligned}$$

定义拉格朗日函数

$$
\begin{aligned}
L(P,w) &\equiv -H(P) + w_0\Big(1 - \sum_y P(y|x)\Big) + \sum_{i=1}^{n} w_i\big(E_{\widetilde{P}}(f_i) - E_P(f_i)\big) \\
&= \sum_{x,y} \widetilde{P}(x)P(y|x)\log P(y|x) + w_0\Big(1 - \sum_y P(y|x)\Big) \\
&+ \sum_{i=1}^{n} w_i\Big(\sum_{x,y} \widetilde{P}(x,y)f_i(x,y) - \sum_{x,y} \widetilde{P}(x)P(y|x)f_i(x,y)\Big)
\end{aligned}
$$

$\min\limits_{P\in C}\max\limits_{w} L(P,w) \Leftrightarrow \max\limits_{w}\min\limits_{P\in C} L(P,w)$

$$
\begin{aligned}
\frac{\partial L(P,w)}{\partial P(y|x)} &= \widetilde{P}(x)(\log P(y|x)+1) - \sum_y w_0 - \sum_{x,y}\left(\widetilde{P}(x)\sum_{i=1}^{t} w_i f_i(x,y)\right) \\
&= \widetilde{P}(x)\left(\log P(y|x) + 1 - w_0 - \sum_{i=1}^{n} w_i f_i(x,y)\right)
\end{aligned}
$$

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

由于$\sum_y P(y|x) = 1$，有

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

记 $\Psi(w) = \min\limits_{P\in \mathbf{C}} L(P,w) = L(P_w,w)$

其解为$P_w = \arg\min\limits_{P\in C} L(P,w) = P_w(y|x)$

最大熵模型的学习归结为对偶函数$\Psi(w)$的极大化。

例**6.2** 学习例6.1中的最大熵模型.

解：分别以y_1, y_2, y_3, y_4, y_5表示A,B, C, D和E,于是最大熵模型学习的最优化问题是

$$
\begin{aligned}
&\min -H(P) = \sum_{i=1}^{5} P(y_i)\log P(y_i) \\
&\text{s.t. } P(y_1) + P(y_2) = \widetilde{P}(y_1) + \widetilde{P}(y_2) = \frac{3}{10} \\
&\sum_{i=1}^{3} P(y_i) = \sum_{i=1}^{3} \widetilde{P}(y_i) = 1
\end{aligned}
$$

引进拉格朗日乘子w_0, w_1，定义拉格朗日函数

$$
L(P,w) = \sum_{i=1}^{5} P(y_i)\log P(y_i) + w_1\Big(P(y_1) + P(y_2) - \frac{3}{10}\Big) + w_0\Big(\sum_{i=1}^{3} P(y_i) - 1\Big)
$$

对偶问题: $$\max_w \min_P L(P,w)$$

$$\frac{\partial L(P,w)}{\partial P(y_1)} = 1+\log P(y_1)+w_1+w_0$$
$$\frac{\partial L(P,w)}{\partial P(y_2)} = 1+\log P(y_2)+w_1+w_0$$
$$\frac{\partial L(P,w)}{\partial P(y_3)} = 1+\log P(y_3)+w_0$$
$$\frac{\partial L(P,w)}{\partial P(y_4)} = 1+\log P(y_4)+w_0$$
$$\frac{\partial L(P,w)}{\partial P(y_3)} = 1+\log P(y_3)+w_0$$

于是
$$P(y_1)=P(y_2)=e^{-w_1-w_0-1}$$
$$P(y_3)=P(y_4)=P(y_3)=e^{-w_0-1}$$

$$\min_P L(P,w)=L(P_w,w)=-2e^{-w_1-w_0-1}-3e^{-w_0-1}-\frac{3}{10}w_1-w_0$$

$$\max_w L(P_w,w) \Longrightarrow \begin{cases} P(y_1)=P(y_2)=\frac{3}{20} \\ P(y_3)=P(y_4)=P(y_5)=\frac{7}{30} \end{cases}$$

6.2.4 极大似然估计

最大似然函数的一般形式是X中各个样本的联合概率:

$$L(x_1,x_2,\ldots,x_n;\theta)=\prod_{i=1}^{n} p(x_i;\theta)$$

对数似然函数的一般形式为

$$L_{\widetilde{p}}=\log\prod_x p(x)^{\widetilde{p}(x)}$$

假设样本集的大小为n, x的取值范围为$\{v_1,v_2,\ldots,v_k\}$, 用$C(X=v_i)$表示在观测值中样本v_i出现的频数，于是

$$L(x_1,x_2,\ldots,x_n;\theta)=\prod_{i=1}^{k} p(v_i;\theta)^{C(X=v_i)}$$

$$L(x_1,x_2,\ldots,x_n;\theta)^{\frac{1}{n}}=\prod_{i=1}^{k} p(v_i;\theta)^{\frac{C(X=v_i)}{n}}$$

因为经验概率$\widetilde{p}(X=v_i)=\dfrac{C(X=v_i)}{n}$, 所以简写得到

$$L(x_1, x_2, \ldots, x_n; \theta)^{\frac{1}{n}} = \prod_{x} p(x;\theta)^{\widetilde{p}(x)}$$

联合概率密度的似然函数

$$\begin{aligned}
L_{\widetilde{p}} &= \log \prod_{x,y} p(x,y)^{\widetilde{p}(x,y)} \\
&= \sum_{x,y} \widetilde{p}(x,y) \log p(x,y) \\
&= \sum_{x,y} \widetilde{p}(x,y) \log[\widetilde{p}(x) p(y|x)] \\
&= \sum_{x,y} \widetilde{p}(x,y) \log p(y|x) + \sum_{x,y} \widetilde{p}(x,y) \log \widetilde{p}(x)
\end{aligned}$$

上述公式第二项是一个常数项（都是样本的经验概率），一旦样本集确定，就是个常数，可以忽略.

证明对偶函数的极大化等价于最大熵模型的极大似然估计.

条件概率分布的对数似然函数为

$$L_{\widetilde{P}}(P_w) = \log \prod_{x,y} P(y|x)^{\widetilde{P}(x,y)} = \sum_{x,y} \widetilde{P}(x,y) \log P(y|x)$$

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

$$\begin{aligned}
\Psi(w) &= \sum_{\mathbf{x},\mathbf{y}} \widetilde{P}(x) P_{\mathbf{w}}(y|x) \log P_w(y|x) \\
&\quad + \sum_{i=1}^{m} w_i \left(\sum_{z,y} \widetilde{P}(x,y) f_i(x,y) - \sum_{x,y} \widetilde{P}(x) P_w(y|x) f_i(x,y) \right) \\
&= \sum_{x,y} \widetilde{P}(x,y) \sum_{i=1} w_i f_i(x,y) + \sum_{x,y} \widetilde{P}(x) P_w(y|x) \left(\log P_w(y|x) - \sum_{i=1}^{n} w_i f_i(x,y) \right) \\
&= \sum_{x,y} \widetilde{P}(x,y) \sum_{i=1}^{n} w_i f_i(x,y) - \sum_{x,y} \widetilde{P}(x) P_w(y|x) \log Z_x(x) \\
&= \sum_{x,y} \widetilde{P}(x,y) \sum_{i=1}^{n} w_i f_i(x,y) - \sum_{x} \widetilde{P}(x) \log Z_x(x)
\end{aligned}$$

$$\Rightarrow \Psi(w) = L_{\widetilde{P}}(P_w)$$

6.3 模型的最优化算法

目标函数为光滑凸函数，可用梯度下降法，改进的迭代尺度法，牛顿法或拟牛顿法

6.3.1 改进的迭代尺度法（improved **iterative** scaling， IIS）

已知最大熵模型为

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

对数似然函数是 $L(w)=\sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^n w_i f_i(x,y)-\sum_x \widetilde{P}(x)\log Z_w(x)$

IIS的想法是，假设最大熵模型当前的参数向量是w，希望找到一个新的参数向量$w+\delta$，使得模型的对数似然值增大.

$$\begin{aligned}L(w+\delta)-L(w) &=\sum_{x,y}\widetilde{P}(x,y)\log P_{w+\delta}(y|x)-\sum_{x,y}\widetilde{P}(x,y)\log P_w(y|x)\\ &=\sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^{\infty}\delta_i f_i(x,y)-\sum_x\widetilde{P}(x)\log\frac{Z_{w+\delta}(x)}{Z_w(x)}\end{aligned}$$

利用不等式$-\log\alpha\geq 1-\alpha,\quad \alpha>0$

$$\begin{aligned}L(w+\delta)-L(w) &\geq\sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_x\widetilde{P}(x)\frac{Z_{w+\delta}(x)}{Z_w(x)}\\ &=\sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_x\widetilde{P}(x)\sum_y P_w(y|x)\exp\sum_{i=1}^n\delta_i f_i(x,y)\end{aligned}$$

右端记为$A(\delta|w)$，于是有 $L(w+\delta)-L(w)\geq A(\delta|w)$

如果能找到适当的δ使下界$A(\delta|w)$提高， 那么对数似然函数也会提高． 然而， 函数$A(\delta|w)$中的δ是一个向量，含有多个变量， 不易同时优化. IIS 试图一次只优化其中一个变量，其余不动.

令 $$f^{\#}(x,y)=\sum_i f_i(x,y)$$

$$A(\delta|w)=\sum_{k,y}\widetilde{P}(x,y)\sum_{n=1}^{n}\delta_i f_i(x,y)+1-\sum_k\widetilde{P}(x)\sum_y P_w(y|x)\exp\left(f^{\#}(x,y)\sum_{k=1}^n\frac{\delta_i f_i(x,y)}{f^{\#}(x,y)}\right)$$

有Jensen不等式

$$\exp\left(\sum_{i=1}^n\frac{f_i(x,y)}{f^{\#}(x,y)}\delta_i f^{\#}(x,y)\right)\leq\sum_{i=1}^n\frac{f_i(x,y)}{f^{\#}(x,y)}\exp\left(\delta_i f^{\#}(x,y)\right)$$

$$A(\delta|w)\geq\sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_x\widetilde{P}(x)\sum_y P_w(y|x)\sum_{i=1}^n\left(\frac{f_i(x,y)}{f^{\#}(x,y)}\right)\exp\left(\delta_i f^{\#}(x,y)\right):=B(\delta|w)$$

$$L(w+\delta)-L(w) \geq B(\delta|w)$$

$$\frac{\partial B(\delta|w)}{\partial \delta_i}=\sum_{x,y}\widetilde{P}(x,y)f_i(x,y)-\sum_i \widetilde{P}(x)\sum_y P_w(y|x)f_i(x,y)\exp\left(\delta_i f^{\#}(x,y)\right)=0$$

$$\sum_{x,y}\widetilde{P}(x)P_w(y|x)f_i(x,y)\exp\left(\delta_i f^{\#}(x,y)\right)=E_{\widetilde{P}}(f_i)$$

上述方程依次求解δ_i可得δ.

算法**6.1** (改进的迭代尺度算法**IIS)**

输入：特征函数$f_1,f_2,\cdots,f_n$, 经验分布$\widetilde{P}(X,Y)$, 模型$P_w(y/x)$.
输出：最优参数值w, 最优模型P.

- (1) 对所有$i \in \{1,2,\cdots,n\}$，取初值$w_i=0$
- (2) 对每一$i \in \{1,2,\cdots,n\}$
- (a) 求方程的解 δ_i：$\sum_{x,y}\widetilde{P}(x)P(y|x)f_i(x,y)\exp\left(\delta_i f^{\#}(x,y)\right)=E_{\widetilde{p}}(f_i)$
- (b) $w_i \leftarrow w_i+\delta_i$
- (3) 如果不是所有w_i都收敛，重复步骤(2).

若$f^{\#}(x,y)=M$, $\delta_i=\frac{1}{M}\log\frac{E_{\widetilde{p}}(f_i)}{E_p(f_i)}$

若$f^{\#}(x,y)$不是常数，只能数值计算，比如牛顿法

$$g(\delta_t)=0 \Rightarrow \delta_i^{(k+1)}=\delta_i^{(k)}-\frac{g\left(\delta_i^{(k)}\right)}{g'\left(\delta_i^{(k)}\right)}$$

由于方程有单根，牛顿法收敛，且收敛速度很快.

6.3.2 拟牛顿法

目标函数：$\min_{v\in\mathbb{R}^n} f(w)=\sum_x \widetilde{P}(x)\log\sum_y \exp\left(\sum_{i=1}^n w_i f_i(x,y)\right)-\sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^n w_i f_i(x,y)$

梯度：$g(w)=\left(\frac{\partial f(w)}{\partial w_1},\frac{\partial f(w)}{\partial w_2},\cdots,\frac{\partial f(w)}{\partial w_n}\right)^{\mathrm{T}}$

$$\frac{\partial f(w)}{\partial w_i}=\sum_{x,y}\widetilde{P}(x)P_w(y|x)f_i(x,y)-E_{\widetilde{P}}(f_i), \quad i=1,2,\cdots,n$$

算法 6.2（最大熵模型学习的 BFGS 算法）

输入：特征函数 $f_1, f_2, \cdots, f_n$；经验分布 $\tilde{P}(x,y)$，目标函数 $f(w)$，梯度 $g(w) = \nabla f(w)$，精度要求 ε；

输出：最优参数值 w^*；最优模型 $P_{w^*}(y|x)$.

（1）选定初始点 $w^{(0)}$，取 B_0 为正定对称矩阵，置 $k=0$

（2）计算 $g_k = g(w^{(k)})$. 若 $\|g_k\| < \varepsilon$，则停止计算，得 $w^* = w^{(k)}$；否则转 (3)

（3）由 $B_k p_k = -g_k$ 求出 p_k

（4）一维搜索：求 λ_k 使得

$$f(w^{(k)} + \lambda_k p_k) = \min_{\lambda \geqslant 0} f(w^{(k)} + \lambda p_k)$$

（5）置 $w^{(k+1)} = w^{(k)} + \lambda_k p_k$

（6）计算 $g_{k+1} = g(w^{(k+1)})$，若 $\|g_{k+1}\| < \varepsilon$，则停止计算，得 $w^* = w^{(k+1)}$；否则，按下式求出 B_{k+1}：

$$B_{k+1} = B_k + \frac{y_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} \delta_k} - \frac{B_k \delta_k \delta_k^{\mathrm{T}} B_k}{\delta_k^{\mathrm{T}} B_k \delta_k}$$

其中，

$$y_k = g_{k+1} - g_k, \quad \delta_k = w^{(k+1)} - w^{(k)}$$

（7）置 $k = k+1$，转 (3). ■

<u>作业</u>

数据集：Fisher iris data

分类算法：逻辑斯谛回归模型的梯度下降法

要求写出多项逻辑斯谛回归的对数似然函数和梯度下降法的表达式，用python编程实现分类，给出分类精度.