## Softmax

**定义：**训练集 \{ (x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)}) \}， y^{(i)} \in \{1, 2, \ldots, k\}。

**假设函数:**


\begin{align}
h_\theta(x^{(i)}) =
\begin{bmatrix}
p(y^{(i)} = 1 | x^{(i)}; \theta) \\
p(y^{(i)} = 2 | x^{(i)}; \theta) \\
\vdots \\
p(y^{(i)} = k | x^{(i)}; \theta)
\end{bmatrix}
=
\frac{1}{ \sum_{j=1}^{k}{e^{ \theta_j^T x^{(i)} }} }
\begin{bmatrix}
e^{ \theta_1^T x^{(i)} } \\
e^{ \theta_2^T x^{(i)} } \\
\vdots \\
e^{ \theta_k^T x^{(i)} } \\
\end{bmatrix}
\end{align}


**代价函数：**


\begin{align}
J(\theta) = - \frac{1}{m} \left[ \sum_{i=1}^{m} \sum_{j=1}^{k}  1\left\{y^{(i)} = j\right\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{ \theta_l^T x^{(i)} }}\right]
\end{align}


**“冗余”的参数集：**


\begin{align}
p(y^{(i)} = j | x^{(i)} ; \theta)
&= \frac{e^{(\theta_j-\psi)^T x^{(i)}}}{\sum_{l=1}^k e^{ (\theta_l-\psi)^T x^{(i)}}}  \\
&= \frac{e^{\theta_j^T x^{(i)}} e^{-\psi^Tx^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}} e^{-\psi^Tx^{(i)}}} \\
&= \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{ \theta_l^T x^{(i)}}}.
\end{align}


**权重衰减**


\begin{align}
J(\theta) = - \frac{1}{m} \left[ \sum_{i=1}^{m} \sum_{j=1}^{k} 1\left\{y^{(i)} = j\right\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{ \theta_l^T x^{(i)} }}  \right]
              + \frac{\lambda}{2} \sum_{i=1}^k \sum_{j=0}^n \theta_{ij}^2
\end{align}


**和逻辑回归的关系：**


\begin{align}
h(x) &=

\frac{1}{ e^{\vec{0}^Tx}  + e^{ (\theta_2-\theta_1)^T x^{(i)} } }
\begin{bmatrix}
e^{ \vec{0}^T x } \\
e^{ (\theta_2-\theta_1)^T x }
\end{bmatrix} \\


&=
\begin{bmatrix}
\frac{1}{ 1 + e^{ (\theta_2-\theta_1)^T x^{(i)} } } \\
\frac{e^{ (\theta_2-\theta_1)^T x }}{ 1 + e^{ (\theta_2-\theta_1)^T x^{(i)} } }
\end{bmatrix} \\

&=
\begin{bmatrix}
\frac{1}{ 1  + e^{ (\theta_2-\theta_1)^T x^{(i)} } } \\
1 - \frac{1}{ 1  + e^{ (\theta_2-\theta_1)^T x^{(i)} } } \\
\end{bmatrix}
\end{align}

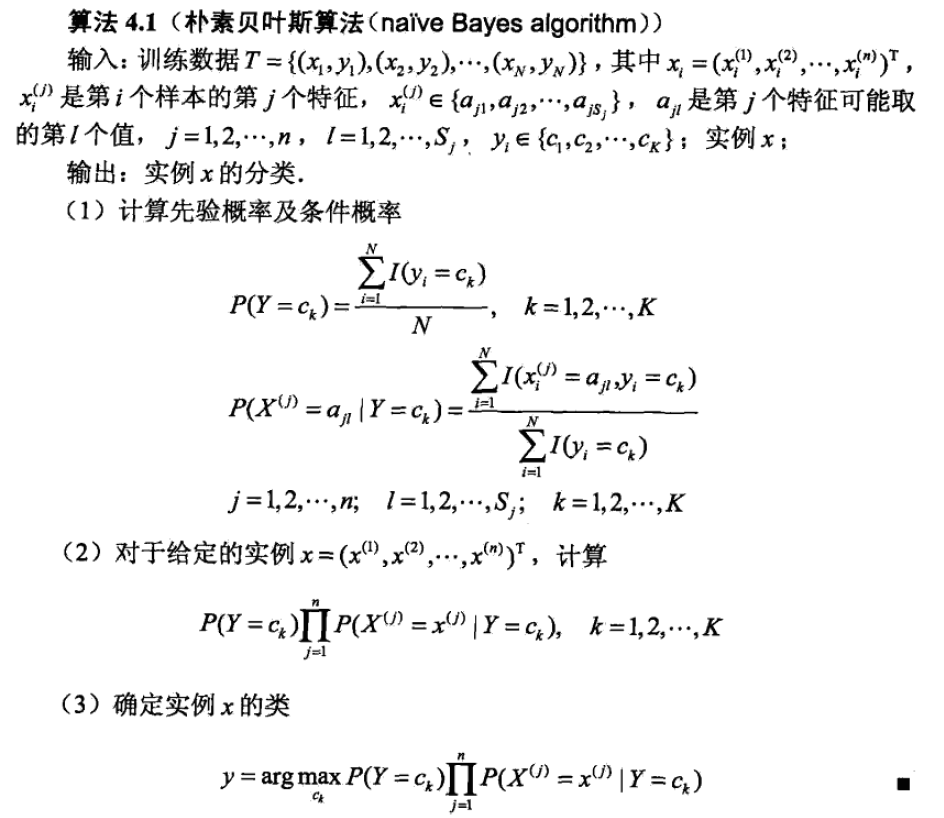

**注意点：**

最小化代价函数，同样可以采用简单而有效的梯度下降，需要提到的是，在程序实现中，我们一般采用批量随机梯度下降，即MSGD，minibatch Stochastic Gradient Descent，简单来说，就是每遍历完一个batch的样本才计算梯度和更新参数，一个batch一般有几十到几百的单个样本。PS：随机梯度下降则是一个样本更新一次。

## 朴素贝叶斯

**基本假设：**条件独立性

**定义（极大似然估计）：**



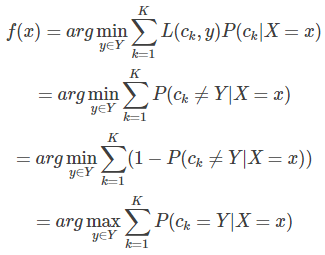
**损失函数（0-1损失函数）**



风险函数：



为了使得期望函数最小化，需要对X=x逐个最小化（条件独立性假设）



由上文看出，计算各个划分的条件概率P(a|y)是朴素贝叶斯分类的关键性步骤，当特征属性为离散值时，只要很方便的统计训练样本中各个划分在每个类别中出现的频率即可用来估计P(a|y)，下面重点讨论特征属性是连续值的情况

**那么当属性为连续值的时候**

当特征属性为连续值时，通常假定其值服从高斯分布（也称正态分布）。即：

http://latex.codecogs.com/gif.latex?g(x,\eta%20,\sigma%20)=\frac%7b1%7d%7b\sqrt%7b2\pi%20%7d\sigma%20%7de%5e-\frac%7b(x-\eta)%5e2%7d%7b2\sigma%5e2%7d

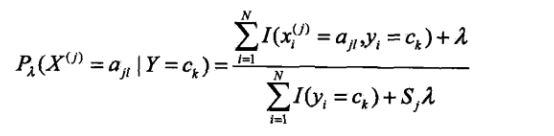
      而

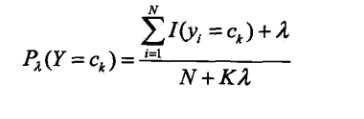
http://latex.codecogs.com/gif.latex?P(a_k|y_i)=g(a_k,\eta_%7by_i%7d,\sigma_%7by_i%7d)

因此只要计算出训练样本中各个类别中此特征项划分的各均值和标准差，代入上述公式即可得到需要的估计值。均值与标准差的计算在此不再赘述。

**下面的是贝叶斯估计**

由于训练量不足会造成有些概率是0，会令分类器质量大大降低。为了解决这个问题，我们引入Laplace校准（这就引出了我们的拉普拉斯平滑），它的思想非常简单，就是对没类别下所有划分的计数加1，这样如果训练样本集数量充分大时，并不会对结果产生影响，并且解决了上述频率为0的尴尬局面。





其中ajl，代表第j个特征的第l个选择，Sj代表第j个特征的个数，K代表种类的个数。λ≥0.当λ=0时是解答似然估计，λ=1时候是拉普拉斯平滑

注意点：使用最大似然估计可能会出现所要估计的概率为0的情况，所以会有贝叶斯估计的出现。

**总结**

1. 朴素贝叶斯是典型的生成学习方法。生成方法由训练数据学习联合概率分布P（X，Y）,然后求得后验概率P(Y|X)。具体来说，利用训练数据学习P(X|Y)和P(Y)的估计。然后得到联合概率分布。

P(X,Y) = P(Y)P(X|Y)

其中概率估计方法可以使用极大似然估计和贝叶斯估计

1. 贝叶斯方法的基本假设是条件独立性：这是一个较强的假设。由于这一假设，模型包含的条件概率的数量大为减少，学习和预测的方法大为简化。
2. 优点：高效，易于实现

缺点：分类性能不一定高

1. 后验概率最大等价于0-1损失函数时的期望风险最小化