感知机和逻辑回归

sysu & spf 机器学习研修班系列讲义　4

主讲人：何福铿

中山大学数学学院

第二章　感知器(Preceptron)

感知机[1]是二分类的线性模型，其输入为实例的特征向量，输出为该实例的类别，用+1和-1标记这两种类别。感知机模型的作用在于把正负两类实例用一个超平面分开,因此属于判别模型。具体的做法是：根据训练数据，导出与误分类有关的损失函数(Loss function)，然后使用梯度下降法(Gradient descend)求出使这个损失函数最小的超平面，并且利用该超平面来预测新实例的类别。感知机模型算法简单并且容易实现，分为原始形式和对偶形式。感知机是支持向量机(Support vector machine) 和神经网络算法(Neural network algorithm)的基础。

本章首先介绍感知器模型的学习策略，并且导出损失函数，再利用梯度下降法优化损失函数，并且介绍对偶形式，最后证明该方法的收敛性。

1　数学准备

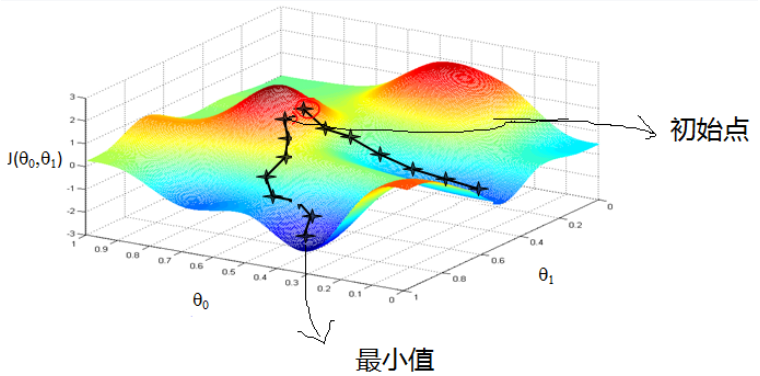
梯度下降法是求无约束最优化问题的一种比较简单的方法，它是一个迭代的算法，每一步都需要求出函数的梯度。

假设$f(x)$ 具有一阶连续的偏导数，无约束最优化问题是

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1}$$

下图是梯度下降算法应用的一个例子：

Figure 1: 梯度下降法的例子



梯度下降法的思想是任意一点在负梯度的方向函数值下降得最快。因此选择一个合适的初值点$x^{(0)}$，求出$x^{(0)}$ 的梯度，并且用此更新x的值，一直迭代下去，直至梯度的范数满足（ϵ 事先给定）

$$\|\nabla f(x^{(k)})\| < \varepsilon \tag{2}$$

更新x 的方法为：

$$x^{(k+1)} \leftarrow x^{(k)} + \lambda_k p_k \tag{3}$$

p_k 表示梯度下降的方向，即$p_k = -\nabla f(x_k)$，λ_k表示的是步长，可以进行一维搜索得到：

$$\lambda_k = \underset{\lambda>0}{\operatorname{argmin}} f(x^{(k)} + \lambda p_k) \tag{4}$$

在实际操作中也可以自由调节步长。λ 属于调节参数，是需要不断训练模型的时候调节的目的当然就是可以通过人为设置，调整模型训练的次数最终达到相对收敛。如果λ 设置过大，则可能越过最小值，导致无法收敛。而如果设置过小则收敛速度过慢。

如果函数可导，且函数的梯度满足李普希兹连续（常数为L），可以采用小于$\frac{1}{L}$ 的步长迭代，这样能保证每次迭代的函数值都不增，保证最终会收敛到梯度为0 的点。

关于梯度下降法收敛性的证明，可以参见[2]。

Algorithm 1 梯度下降法

Input: 目标函数$f(x)$, 目标函数的梯度$g(x) = \nabla f(x)$, 计算精度ϵ

Output: $f(x)$的极小值点x^*

取初始值$x_0 \in \mathbb{R}$，$k = 0$
计算$f(x^{(k)})$ 和$g(x^{(k)})$
while $\|g(x^{(k)})\| > \epsilon$ **do**
    $p_k = -\nabla f(x_k)$
    $\lambda_k = \text{argmin}_{\lambda>0} f(x^{(k)} + \lambda p_k)$
    $x^{(k+1)} \leftarrow x^{(k)} + \lambda_k p_k$
    计算$f(x^{(k+1)})$ 和$g(x^{(k+1)})$
    $k = k + 1$
return $x^{(k)}$

在实际操作中，我们还可以使用随机梯度下降法求优化问题的解。随机梯度下降法是为简化梯度的计算而提出的一种方法。以拟合函数作为例子：

$$h(\theta) = \sum_{j=0}^{n} \theta_j x_j \tag{5}$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (y^i - h_\theta(x^i))^2 \tag{6}$$

其中$h(x)$是要拟合的函数，$J(\theta)$ 为损失函数，θ 是参数，其中m 是训练集的记录条数，j 是参数的个数。

梯度下降的思路为$J(\theta)$ 对θ求偏导，得到每个θ 对应的梯度

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m} \sum_{i=1}^{m} (y^i - h_\theta(x^i)) x_j^i \tag{7}$$

为了最小化风险函数，按每个参数θ的梯度的负方向来更新每个θ （其中λ 为步长）。

$$\theta_j^{'} = \theta_j + \frac{\lambda}{m}\sum_{i=1}^{m}(y^i - h_\theta(x^i))x_j^i \tag{8}$$

如果用这种方法的话，会得到一个全局最优解，但由于每迭代一次都要计算m次梯度，如果m 很大，可想而知这种方法的迭代速度。所以就引入随机梯度下降法。

随机梯度下降法每次随机计算其中一个梯度，用这个梯度近似代替批量梯度。

$$\theta_j^{'} = \theta_j + \lambda(y^i - h_\theta(x^i))x_j^i \tag{9}$$

这里的λ 为步长，x^i 为样本集中随机选取的样本。

虽然不是每次迭代得到的损失函数都向着全局最优方向，但是大的整体的方向是向全局最优解的，最终的结果往往是在全局最优解附近。

随机梯度下降法的优点是迭代速度较快，因为每一次循环都只需要计算一次样本。但是它有缺点，就是有可能会收敛到局部最小值。

2 感知器模型(Preceptron model)

Definition 1 (感知机). 假设输入空间(特征空间)是$\chi \in \mathbb{R}^n$，输出空间是$\Omega = \{+1, -1\}$。输入$x \in \chi$ 表示实例的特征向量，对应于输入空间（特征空间）的点；输出$y \in \Omega$ 表示实例的类别。由输入空间到输出空间的如下函数

$$f(x) = sign(w \cdot x + b) \tag{10}$$

称为感知机。其中w 和b 是感知机模型参数，$w \in \mathbb{R}$ 称为权值（*Weight*）或者权值向量（*Weight vector*）。$b \in \mathbb{R}$ 称为偏置（*Bias*）。$sign(x)$为符号函数，即为

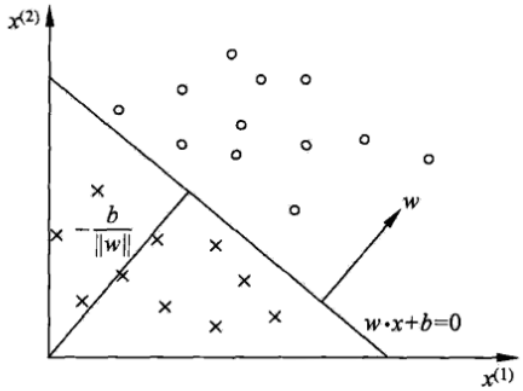
$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{11}$$

感知机是一种线性分类模型，属于判别模型。它的要求的是一个分割$\mathbb{R}^n$的超平面，因此它的假设空间为$\{f|f(x) = w \cdot x + b\}$。

感知机的集合解释为：线性方程

$$w \cdot x + b = 0 \tag{12}$$

Figure 2: 感知机模型



表示的超平面S 将特征空间分为两部分。位于两边的点分别被分为正和负两类。S称为分割超平面。如图二所示。

给定训练数据集（实例的特征向量及其类别）

$$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \tag{13}$$

其中，$x_i \in \mathbb{R}^n$，$y_i \in \{+1, -1\}$，$i = 1$，2，$\ldots, N$，由此求出模型参数w和b。感知器预测，对于新的输入实例给出输出类型。

3 感知器学习策略(Preceptron learning strategy)

Definition 1 (数据集的线性可分性). 给定一个数据集

$$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \tag{14}$$

其中，$x_i \in \mathbb{R}^n$，$y_i \in \{+1, -1\}$，$i = 1$，2，$\ldots, N$，如果存在某个超平面S

$$w \cdot x + b = 0 \tag{15}$$

能够正确地将正实例点和负实例点完全正确地划分到超平面的两侧，即对所有的满足$y_i = +1$的实例i，都有$w \cdot x + b > 0$，对所有的满足$y_i = -1$ 的实例i，都有$w \cdot x + b < 0$，则称数据集T 为线性可分数据集（*Linearly separable data set*），否则称为线性不可分。

假设数据集是线性可分的，感知机学习的目的在于求得一个使数据集完全分开的超平面（即求出参数w 和b）。为此我们需要设计一个损失函数，然后求模型参数使这个损失函数达到最小值。

损失函数的一个自然的想法是误分类点的个数，即

$$L(w,b)=\sum_{i=0}^{N}\mathbf{1}_{(-y_i(w\cdot x_i+b)>0)} \tag{16}$$

，但由于这个函数不是w 和b 的连续可导的函数，不易优化。损失函数的另外一个取法为误分类点到超平面的总距离，这是感知机所使用的。由于$\mathbb{R}^n$ 中的任意一点x_0到超平面S 的距离为：

$$\frac{1}{\|w\|}|w\cdot x_i+b| \tag{17}$$

这里的$\|\cdot\|$ 表示的是二范数。该公式可以由点到平面（直线）的距离公式推广而来。

而误分类等价于下面式子成立：

$$-y_i(w\cdot x_i+b)>0 \tag{18}$$

即$-y_i(w\cdot x_i+b)>0$ 时，$y_i=-1$，而$-y_i(w\cdot x_i+b)<0$ 时，$y_i=1$，所以误分类点x_i 到超平面的距离为

$$-\frac{1}{\|w\|}y_i(w\cdot x_i+b) \tag{19}$$

假设误分类的点的集合为M，则所有误分类点到超平面的距离为

$$-\frac{1}{\|w\|}\sum_{x_i\in M}(y_i(w\cdot x_i+b)) \tag{20}$$

不考虑常数$-\frac{1}{\|w\|}$，我们就得到了感知器的损失函数：

$$\sum_{x_i\in M}(y_i(w\cdot x_i+b)) \tag{21}$$

在这个模型下，如果所有的实例都可以正确分类，那么损失函数的值为0，否则为正数。误分类的点越接近分类平面时，损失函数越小，越接近正确分类，用这个平面来预测新的实例效果越好。

4 感知器学习算法(Preceptron learning algorithm)

根据我们上面的推导，我们把感知机算法归结为如下的最优化问题的算法。给定一个数据集：

$$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \tag{22}$$

其中，$x_i \in \mathbb{R}^n$，$y_i \in \{+1, -1\}$，$i = 1$，2，$\ldots, N$，求参数w和b，使以下损失函数极小化：

$$\min_{w,b} L(w, b) = -\sum_{x_i \in M} (y_i(w \cdot x_i + b)) \tag{23}$$

其中M为误分类数据的集合。

感知机学习算法是根据误分类函数设计的，具体采用随机梯度下降法(Stochastic gradient descent)。首先任意选取一个超平面w_0 和b_0，然后用随机梯度下降法不断优化损失函数。极小化过程中不是一次使M 中的所有误分类点的梯度下降，而是一次随机选取一个误分类点使其梯度下降。

假设误分类点集合M 是固定的，那么损失函数$L(w, b)$ 的梯度由

$$\nabla_w L(w, b) = -\sum_{x_i \in M} y_i x_i \tag{24}$$

$$\nabla_b L(w, b) = -\sum_{x_i \in M} y_i \tag{25}$$

给出。

随机选取一个误分类点(x_i, y_i)，对w 和b 进行更新：

$$w \leftarrow w + \eta y_i x_i \tag{26}$$

$$b \leftarrow b + \eta y_i \tag{27}$$

式中的η是步长，在统计学习中又称为学习率(Learning rate)，这样通过迭代就可以使$L(w, b)$不断缩小直至为0。综上所述，得到上图所示的感知机学习算法的原始形式。

这种学习算法直观上的解释如下：当一个实例点被误分类，即位于分离超平面的错误一侧时，则调整w 和b 的值，使分离超平面向误分类点移动，以减少该误分类点与超平面间的距离，直至超平面越过该误分类点使其被正确分类。

Algorithm 2 感知机学习算法的原始形式

Input: 训练数据集$T=\{(x_1，y_1)，(x_2，y_2)，\ldots，(x_N，y_N)\}$，其中$x_i\in\mathbb{R}^N$，$y_i\in\{-1，+1\}$，$i=1，2，\ldots，N$；学习率$\eta(0<\eta\leq 1)$；

Output: w，b，感知机模型$f(x)=sign(w\cdot x+b)$

1: 取初始值w_0，b_0

2: 在训练集中随机选取数据(x_i,y_i)

3: **if** $y_i(w\cdot x_i+b)\leq 0$ **then**

4: $\quad x\leftarrow w+\eta y_i x_i$

5: $\quad b\leftarrow b+\eta y_i$

6: 转至(2)，直至训练集中没有误分类点。

该算法是感知机学习的基本算法，对应于后面的对偶形式，称为原始形式，感知机学习算法简单且易于实现。

书上的有一个具体计算的实例，由于比较简单，直接套用算法就行了，这里不再赘述。

5 算法的收敛性

现在证明，对于线性可分数据集感知机学习算法原始形式收敛，即经过有限次迭代可以得到一个将训练数据集完全正确划分的分离超平面及感知机模型。

i 为了方便描述，我们记$\hat{w}=(w^T,b)^T$，同样也将输入向量加以扩充，加进常数1，记作$\hat{x}=(x^T,1)^T$。这样，$\hat{x}\in\mathbb{R}^{n+1}$，$\hat{w}\in\mathbb{R}^{n+1}$。显然，
$\hat{w}\cdot\hat{x}=w\cdot x+b$。

Theorem 1 (Novikoff). 设$T=\{(x_1，y_1)，(x_2，y_2)，\ldots，(x_N，y_N)\}$ 是线性可分的，其中$x_i\in\mathbb{R}^N$，$y_i\in\{-1，+1\}$，$i=1，2，\ldots，N$，则
(*1*) 存在满足条件$\|\hat{w}_{opt}\|=1$ 的超平面$\hat{w}_{opt}\cdot\hat{x}=w_{opt}\cdot x+b_{opt}=0$将训练数据集完全正确分开；且存在$\gamma>0$，对所有$i=1，2，\ldots，N$，

$$y_i(\hat{w}_{opt}\cdot\hat{x}_i)=y_i(w_{opt}\cdot x_i+b_{opt})\geq\gamma \tag{28}$$

(*2*) 令$R=\max_{1\leq i\leq N}\|\hat{x}_i\|$，则感知机算法*2.1* 在训练数据集上的误分类次数k满足不等式

$$k\leq(\frac{R}{\gamma})^2 \tag{29}$$

证明 （1）由于训练数据集是线性可分的，按照定义2.2，存在超平面可将训练数据集完全正确分开，取次超平面为$\hat{w}_{opt} \cdot \hat{x} = w_{opt} \cdot x + b_{opt} = 0$，使$\|\hat{w}_{opt}\| = 1$，由于对有限的$i = 1$，$2$，$\ldots$，$N$，均有

$$y_i(\hat{w}_{opt} \cdot \hat{x}_i) = y_i(w_{opt} \cdot x_i + b_{opt}) > 0 \tag{30}$$

所以存在

$$\gamma = \min_i \{y_i(w_{opt} \cdot x_i + b_{opt})\} \tag{31}$$

使

$$y_i(\hat{w}_{opt} \cdot \hat{x}_i) = y_i(w_{opt} \cdot x_i + b_{opt}) \geq \gamma \tag{32}$$

（2）感知机算法从$\hat{w}_0 = 0$ 开始，如果实例被误分类，则重新更新权重。令$\hat{w}_{k-1}$ 是第k 个误分类实例之前的扩充权重向量，即

$$\hat{w}_{k-1} = (w_{k-1}^T, b_{k-1})^T \tag{33}$$

则第k个误分类实例的条件是

$$y_i(\hat{w}_{k-1} \cdot \hat{x}_i) = y_i(w_{k-1} \cdot x_i + b_{k-1}) \leq 0 \tag{34}$$

若(x_i, y_i)是被$w_{k-1} = (w_{k-1}^T, b_{k-1})^T$ 误分类的数据，则w 和b 的更新是

$$w_k \leftarrow w_{k-1} + \eta y_i x_i \tag{35}$$

$$b_k \leftarrow b_{k-1} + \eta y_i \tag{36}$$

即

$$\hat{w}_k = \hat{w}_{k-1} + \eta y_i \hat{x}_i \tag{37}$$

下面推导两个不等式：

（1）

$$\hat{w}_k \cdot \hat{w}_{opt} \geq k\eta\gamma \tag{38}$$

根据（28）和（37），

$$\begin{aligned} \hat{w}_k \cdot \hat{w}_{opt} &= \hat{w}_{k-1} \cdot \hat{w}_{opt} + \eta y_i \hat{w}_{opt} \cdot \hat{x}_i \\ &\geq \hat{w}_{k-1} \cdot \hat{w}_{opt} + \eta\gamma \end{aligned} \tag{39}$$

由此递推即得不等式（2.12）

$$\hat{w}_k \cdot \hat{w}_{opt} \geq \hat{w}_{k-1} \cdot \hat{w}_{opt} + \eta\gamma \geq w_{k-2} \cdot \hat{w}_{opt} + 2\eta\gamma \geq \cdots \geq k\eta\gamma \tag{40}$$

（2）

$$\|\hat{w}_k\|^2 \leq k\eta^2 R^2 \tag{41}$$

由式（34）和（37），

$$\begin{aligned} \|\hat{w}_k\|^2 =& \|\hat{w}_{k-1}\|^2 + 2\eta y_i \hat{w}_{k-1} \cdot \hat{x}_i + \eta^2 \|\hat{x}_i\|^2 \\ &\leq \|\hat{w}_{k-1}\|^2 + \eta^2 \|\hat{x}_i\|^2 \\ &\leq \|\hat{w}_{k-1}\|^2 + \eta^2 R^2 \\ &\leq \|\hat{w}_{k-2}\| + 2\eta^2 R^2 \leq \ldots \\ &\leq k\eta^2 R^2 \end{aligned} \tag{42}$$

结合不等式（40）和（41）即得

$$k\eta\gamma \leq \hat{w}_k \cdot \hat{w}_{opt} \leq \|\hat{w}_k\| \|\hat{w}_{opt}\| \leq \sqrt{k}\eta R \tag{43}$$

$$k^2\gamma^2 \leq kR^2 \tag{44}$$

于是

$$k \leq (\frac{R}{\gamma})^2 \tag{45}$$

定理表明，误分类的次数k是有上界的，经过有限次循环可以将训练数据完全分开的分离超平面。也就是说，当训练数据线性可分时，感知机学习算法原始形式迭代是收敛的。但是感知机学习算法存在很多解，这些解依赖于处置的选择，也依赖于迭代过程中误分类点选择的顺序。为了得到唯一的超平面，需要对分离超平面增加约束条件。这就是第七章将要讲的支持向量机的想法。当训练集线性不可分时，感知机算法不收敛，会出现震荡现象。

6 感知机学习算法的对偶形式

现在考虑感知机学习算法的对偶形式。感知机学习算法有原始形式和对偶形式，这和支持向量机学习算法有原始形式和对偶形式相对应。

对偶形式的基本想法是，将w和b 表示成x_i和y_i的线性组合的形式，通过求解其系数来求解w和b。不失一般性，在原始形式的算法中我们假设$w_0 = 0$，$b_0 = 0$，对误分类点通过

$$w \leftarrow w + \eta y_i x_i \tag{46}$$

$$b \leftarrow b + \eta y_i \tag{47}$$

逐步修改w和b，设修改n次，则w和b关于(x_i, y_i)的增量分别为$\alpha_i y_i x_i$和$\alpha_i y_i$，这里$\alpha_i = n_i \eta$。这样，从学习过程不难看出，最后学习到的w和b可以分别表示为

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{48}$$

$$b = \sum_{i=1}^{N} \alpha_i y_i \tag{49}$$

这里，$\alpha_i \geq 0$，$i = 1$，2，$\ldots$，N，当$\eta = 1$时，n_i表示第i 个实例点由于误分类而进行更新的次数。实例点更新次数越多，意味着它距离分离超平面越近，也就越难正确分类。换句话说，这样的实例对学习结果影响最大。

于是便得到感知机学习算法的对偶形式。

Algorithm 3 感知机学习算法的对偶形式

Input: 训练数据集$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$，其中$x_i \in \mathbb{R}^N$，$y_i \in \{-1, +1\}$，$i = 1, 2, \ldots, N$；学习率$\eta (0 < \eta \leq 1)$；

Output: α，b；感知机模型$f(x) = sign(\sum_{j=1}^{N} \alpha_j y_j x_j \cdot x + b)$，其中$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)^T$

1: $\alpha \leftarrow 0$，$b \leftarrow 0$
2: 在训练集中随机选取数据(x_i, y_i)
3: **if** $y_i(\sum_{j=1}^{N} \alpha_j y_j x_j \cdot x_i) \leq 0$ **then**
4: $\quad \alpha_i \leftarrow \alpha_i + \eta$
5: $\quad b \leftarrow b + \eta y_i$
6: 转至(2)，直至训练集中没有误分类点。

对偶形式中训练实例仅以内积的形式出现。为了方便，可以预先将训练集中实例间的内积计算出来并以矩阵的形式储存，这个矩阵就是Gram矩阵。即

$$G = [x_i \cdot x_j]_{N \times N} \tag{50}$$

关于这个算法，书上有一个具体计算的实例，由于比较简单，直接套用算法就行了，这里不再赘述。

第六章　逻辑回归（Logistic regression）

机器学习中的逻辑回归通过计算新样本属于某一类别的概率从而进行分类，计算概率的方法用到一种叫做logistic的函数。在逻辑回归中，我们确定分界面所使用的方法是极大似然估计法。这里的分界面可以突破平面的限制，只需要添加上一个非线性项就可以实现这一点。

1　逻辑回归模型

Definition 1 (逻辑分布). 设X是连续随机变量，X 服从逻辑分布是指X 具有下列分布函数和密度函数：

$$F(x) = P(X \leq x) = \frac{1}{1 + e^{-(x-\mu)/\gamma}} \tag{51}$$

$$f(x) = F^{'}(x) = \frac{e^{-(x-\mu)/\gamma}}{\gamma(1 + e^{-(x-\mu)/\gamma})^2} \tag{52}$$

式中，μ为位置参数，$\gamma > 0$ 为形状参数。

逻辑分布的密度函数$f(x)$ 和分布函数$F(x)$ 的图形如下图所示：

Figure 3: 逻辑分布的密度函数和分布函数

其图形是一条S 形曲线（Sigmoid curve）。该曲线以点$(\mu, \frac{1}{2})$为对称中心，即满足

$$F(-x + \mu) - \frac{1}{2} = -F(x - \mu) + \frac{1}{2} \tag{53}$$

曲线在中心附近增长速度较快，在两端增长速度较慢。形状参数γ的值越小，曲线在中心附近增长得越快。

2 二项逻辑回归模型

二项逻辑回归模型（Binomial logistic regression model）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分布。这里，随机变量X取值为实数，随机变量Y取值为1或0。我们通过监督学习的方法来估计模型参数。

Definition 1 (逻辑回归模型). 二项逻辑回归模型是如下的条件概率分布：

$$P(Y=1|x)=\frac{exp(w\cdot x+b)}{1+exp(w\cdot x+b)} \tag{54}$$

$$P(Y=0|x)=\frac{1}{1+exp(w\cdot x+b)} \tag{55}$$

这里，$x\in\mathbb{R}^n$是输入，$Y\in\{0,1\}$是输出，$w\in\mathbb{R}^n$和$b\in\mathbb{R}$是参数，w称为权值向量，b称为偏置，$w\cdot x$ 为w和x 的内积。

对于给定的输入实例x，按照上两式可以求出$P(Y=1|x)$和$P(Y=0|x)$，逻辑回归比较两个条件概率值的大小，将实例x分到概率值较大的那一类。

有时为了方便，将权值向量和输入向量加以补充，仍记作w和x，即$w=(w^{(1)},w^{(2)},\ldots,w^{(n)},b)$，$x=(x^{(1)},x^{(2)},\ldots,x^{(n)},1)^T$，这时，逻辑回归模型如下：

$$P(Y=1|x)=\frac{exp(w\cdot x)}{1+exp(w\cdot x)} \tag{56}$$

$$P(Y=0|x)=\frac{1}{1+exp(w\cdot x)} \tag{57}$$

现在考差逻辑回归模型的特点。一个事件的几率(Odds)是指该事件发生的概率和不发生的概率的比值。如果事件发生的概率为p，那么该事件的几率是$\frac{p}{1-p}$，该事件的对数几率(Log odds)或者logit函数是

$$logit(p)=log(\frac{p}{1-p}) \tag{58}$$

对于逻辑回归而言，由（56）（57），

$$log(\frac{P(Y=1|x)}{1-P(Y=1|x)})=w\cdot x \tag{59}$$

这就是说，在逻辑回归中，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，即逻辑回归模型。

换一个角度看，考虑对输入x进行分类的线性函数$w \cdot x$，其值域为实数域。这里$x \in \mathbb{R}^{n+1}$，$w \in \mathbb{R}^{n+1}$。通过逻辑回归模型定义可以将$w \cdot x$转换成概率：

$$P(Y=1|x) = \frac{exp(w \cdot x)}{1 + exp(w \cdot x)} \tag{60}$$

这时，线性函数的值越接近正无穷，概率值就越接近1；线性函数的值越接近负无穷，概率值就越接近0。这样的模型就是逻辑回归模型。

3 模型参数估计

对于给定的数据集$T = \{(x_1, y_1), (x_2, y_2), dots, (x_N, y_N)\}$，$x_i \in \mathbb{R}^n$，$y_i \in \{+1, -1\}$，$i = 1$，$2$，$\ldots, N$，可以应用极大似然估计法估计模型参数。设

$$P(Y=1|x) = \pi(x), P(Y=0|x) = 1 - \pi(x) \tag{61}$$

对数似然函数为

$$\begin{aligned} L(w) &= \sum_{i=1}^{N} [y_i log(\pi(x_i)) + (1 - y_i) log(1 - \pi(x_i))] \\ &= \sum_{i=1}^{N} [y_i log(\frac{\pi(x_i)}{1 - \pi(x_i)}) + log(1 - \pi(x_i))] \\ &= \sum_{i=1}^{N} [y_i (w \cdot x_i) - log(1 + exp(w \cdot x_i))] \end{aligned} \tag{62}$$

对$L(w)$求极大值，得到w 的估计值。

这样，问题就转化为以对数似然函数为目标函数的最优化问题，通常采用梯度下降法和拟牛顿法。

假设w的极大似然估计值是$\hat{w}$，那么学到的逻辑回归模型为

$$P(Y=1|x) = \frac{exp(\hat{w} \cdot x)}{1 + exp(\hat{w} \cdot x)} \tag{63}$$

$$P(Y=0|x) = \frac{1}{1 + exp(\hat{w} \cdot x)} \tag{64}$$

4 多项逻辑回归模型

二项逻辑回归模型用于二分类，而对于多分类，我们使用多项逻辑回归模型假设其随机变量$Y = \{1, 2, \ldots, K\}$，那么多项逻辑回归模型为

$$P(Y = k|x) = \frac{exp(w_k \cdot x)}{1 + \sum_{k=1}^{K-1} exp(w_k \cdot x)}, \quad k = 1, 2, \ldots, K \tag{65}$$

$$P(Y = K|x) = \frac{1}{1 + \sum_{k=1}^{K-1} exp(w_k \cdot x)} \tag{66}$$

这里，$x \in \mathbb{R}^{n+1}$，$w_k \in \mathbb{R}^{n+1}$。

二项逻辑回归的参数估计法同样可以推广到多项逻辑回归。

关于逻辑回归更多用法可以参考[3]。

参考文献

[1] 李航，统计学习方法[M]。北京：清华大学出版社，2012:48-53

[2] http://blog.csdn.net/shenxiaolu1984/article/details/52577996

[3] http://blog.csdn.net/han_xiaoyang/article/details/49123419