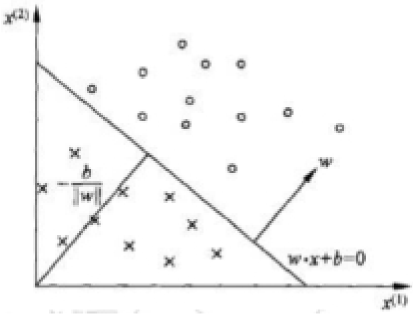
1. 感知机学习

用超平面将数据二分类（分为正类和负类），注意平面不是只有一个（图中的直接就是其中一个超平面）



Review:

关于平面和点到平面的距离知识

http://www.cnblogs.com/graphics/archive/2010/07/10/1774809.html

||w||指一个2-范数，是各元素的平方和再开方。w可能是一个向量，所以要用||w||表示

圈圈表示正类，而叉叉表示负类。圈圈与叉叉之间的直线即上文所说的分离超平面（注意分离超平面并不是唯一的！）它将所有的样本划分为两部分。位于分离超平面上方的为正类，记为+1，位于分离超平面下方的为负类，记为-1。也就是说，假设给一个样本的特征向量x，如果$w\cdot x+b>0$, 那么样本为正类(+1)，反之若$w\cdot x+b<0$, 样本则属于负类(-1)。我们 引入符号函数sign(x)，即

$$sign(x)=\{^{+1,\ x\geq 0}_{-1,\ x<0}$$

由此我们可以得到由输入空间到输出空间的函数

$$f(x)=sign(w\cdot x+b)$$

这就叫做感知机。其中，w和b为感知机参数，$w\in R^n$叫做权值或权值向量，$b\in R$叫做偏置，$w\cdot x$表示w和x的内积。感知机学习的目的就在于确定参数w和b的值。

学习策略：

给定一个线性可分的数据集

$$T=\{(x_1,y_1),(x_2,y_2),\dots(x_N,y_N)\}$$

其中$x_i\in X=R^n$，$y_i\in Y=\{+1,-1\}$，$i=1,2,3,\dots N$。

为了确定感知机模型的参数w和b，需要确定一个学习策略，即定义一个损失函数并将损失函数极小化。感知机采用的损失函数为误分类点到超平面的总距离。首先写出输入空间R^n中任一点x_0到分离超平面的距离

$$\frac{1}{\|w\|}|w\cdot x_0+b|$$

这里$\|w\|$ 是w的L_2范数。

1. 任一点x0到超平面的距离公式中,||w||因为是x0的参数。由点到平面的距离公式，是各变量的系数平方和开方，因为感知机超平面方程中只有w是变量的参数，所以等于||w||

有了计算距离的方式，下面我们来看看损失函数究竟怎么定义。由于对于模型来说，在分类错误的情况下，若$w\cdot x_i+b>0$，则实际的y_i应该是等于-1,而当$w\cdot x_i+b<0$时，y_i等于1,因此由这个特性我们可以去掉上面的绝对值符号，将公式转化为:

$$len(\boldsymbol{x_i}) = -\frac{1}{\|\boldsymbol{w}\|} y_i(\boldsymbol{w}\cdot \boldsymbol{x_i} + b)$$

如此得到最终的损失函数为:

$$L(\boldsymbol{w},b) = \sum_{\boldsymbol{x_i}\in M} len(\boldsymbol{x_i}) \Rightarrow$$
$$L(\boldsymbol{w},b) = \sum_{\boldsymbol{x_i}\in M} -y_i(\boldsymbol{w}\cdot \boldsymbol{x_i} + b)$$

正如上面所示，$\frac{1}{\|w\|}$这个因子在这儿可以不用考虑，因为它对结果的影响与w,b是等效的，因此只用单独考虑w,b就可以，这样可以减小运算复杂度。到这一步问题就变得简单了，那就是求L(w,b)的极小值。对于极大值极小值的求解方法有许多，这儿首先讲述一种梯度下降的方法求极小值，根据梯度的定义，我们可以得到损失函数的梯度有:

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

（损失函数只对误分类的点计算距离）

显然，损失函数L(w,b)是非负的。如果没有误分类点，损失函数值为0，而且，误分类点越少，误分类点离超平面越近，损失函数的值越小。

如何理解损失函数为何要乘上-y：

注意前提是在*分类错误的情况下，意思是通过f(x)=sign()计算出来的**y**是相反的*。通过比较计算得出的y和实际的y不等，知道这是一个错误分类的点。**再去计算损失函数。**

w*x+b>0, 计算的y=-1，因为距离为正数，所以乘上y=-1就是正确的。反之同理。

如何理解**最终的损失函数**求和：

通过比较计算得出的y和实际的y不等，知道这是一个错误分类的点。再计算损失函数，对w和b调整后再求和。

感知机学习算法是误分类驱动的，具体采用随机梯度下降法。首先，任意选取一个超平面w_0, b_0，然后用梯度下降法不断地极小化损失函数。极小化过程中不是一次使M中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。损失函数$L(w,b)$的梯度为

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

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

随机选取一个误分类点(x_i, y_i)，对w, b进行更新：

$$w \leftarrow w + \eta y_i x_i$$

$$b \leftarrow b + \eta y_i$$

式中$\eta (0 < \eta \leq 1)$是步长，在统计学习中又称为学习率。

综上所述，得到如下算法(**感知机学习算法的原始形式**)

输入： 训练集 $T = \{(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)\}$，其中 $x_i \in X = R^n$，$y_i \in Y = \{+1, -1\}$，$i = 1, 2, 3, \ldots N$；学习率$\eta (0 < \eta \leq 1)$;

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

(1)选取初值w_0，b_0

(2)在训练集中选取数据(x_i, y_i)

(3)如果$y_i(w \cdot x_i + b) \leq 0$

对参数w, b进行更新:

根据定义，梯度也就是代价函数对每个参数的偏导。在上面的损失函数

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

参数是w和b，展开后分别求偏导就得到

$$w \leftarrow w + \eta y_i x_i$$

$$b \leftarrow b + \eta y_i$$

(η是学习速率)

https://blog.csdn.net/u013358387/article/details/53303932#commentBox