从线性到非线性模型

1、线性回归，岭回归，Lasso回归，局部加权线性回归

2、logistic回归，softmax回归，最大熵模型

3、广义线性模型

4、Fisher线性判别和线性感知机

5、三层神经网络

6、支持向量机

一、线性回归

一、线性回归

假设有数据有$T=\left\{\left(x^{(1)}, y^{(1)}\right), \ldots,\left(x^{(i)}, y^{(i)}\right), \ldots,\left(x^{(m)}, y^{(m)}\right)\right\}$其中$x^{(i)}=\left\{x_1^{(i)}, .., x_j^{(i)}, \ldots, x_n^{(i)}\right\}, y^i \in \mathbf{R}$。其中m为训练集样本数，n为样本维度，y是样本的真实值。线性回归采用一个多维的线性函数来尽可能的拟合所有的数据点，最简单的想法就是最小化函数值与真实值误差的平方（概率解释-高斯分布加最大似然估计）。即有如下目标函数：

$$
\begin{aligned}
J(\theta) &= \frac{1}{2} \sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)^2 \\
&\min_\theta J(\theta)
\end{aligned}
$$

其中线性函数如下：

$$
\begin{aligned}
h_\theta\left(x^{(i)}\right) &= \theta_0+\theta_1 x_1^{(i)}+\theta_2 x_2^{(i)}+..+\theta_n x_n^{(i)} \\
&= \sum_{j=1}^{n} \theta_j x_j^{(i)} \\
&= \theta^T \mathbf{x}^{(i)}
\end{aligned}
$$

构建好线性回归模型的目标函数之后，接下来就是求解目标函数的最优解，即一个优化问题。常用的梯度优化方法都可以拿来用，这里以梯度下降法来求解目标函数。

$$
\begin{aligned}
\theta_j: &= \theta_j-\alpha \frac{\partial}{\partial \theta_j} J(\theta) \\
&= \theta_j-\alpha \frac{\partial}{\partial \theta_j} \frac{1}{2} \sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2 \\
&= \theta_j-\alpha \sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right) \frac{\partial}{\partial \theta_j} \\
&= \theta_j-\alpha \sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right) x_j^{(i)}
\end{aligned}
$$

另外，线性回归也可以从最小二乘法的角度来看，下面先将样本表示向量化，$X \in R^{n \times m}$，$Y \in R^{m}$，构成如下数据矩阵。

$$\begin{bmatrix} - & (x^1\ ,\ y^1)^T & - \\ - & (x^2\ ,\ y^2)^T & - \\ - & \cdot & - \\ - & (x^m\ ,\ y^m)^T & - \end{bmatrix}_{(n+1) \times m}$$

那么目标函数向量化形式如下：

$$J(\theta) = \frac{1}{2}\left(\theta^T X - y^T\right)\left(\theta^T X - y^T\right)^T$$

可以看出目标函数是一个凸二次规划问题，其最优解在导数为0处取到，矩阵导数详细参考。

$$\begin{aligned} \bigtriangledown_\theta J(\theta) &= XX^T - XY = 0 \\ \Rightarrow \theta &= \left(XX^T\right)^{-1} XY \end{aligned}$$

值得注意的上式中存在计算矩阵的逆，一般来讲当样本数大于数据维度时，矩阵可逆，可以采用最小二乘法求得目标函数的闭式解。当数据维度大于样本数时，矩阵线性相关，不可逆。此时最小化目标函数解不唯一，且非常多，出于这样一种情况，我们可以考虑奥卡姆剃刀准则来简化模型复杂度，使其不必要的特征对应的w为0，可以考虑0范数使得模型中w非0个数最少（实际上采用的是0范数的一个凸近似）。当然，岭回归，lasso回归的最根本的目的不是解决不可逆问题，而是防止过拟合。

概率解释

损失函数与最小二乘法采用最小化平方和的概率解释。假设模型预测值与真实值的误差为 $\epsilon^{(i)}$，那么预测值 $h_\theta(x^{(i)})$与真实值$y^{(i)}$之间有如下关系：

$$y^{(i)} = h_\theta(x^{(i)}) + \epsilon^{(i)}$$

根据中心极限定理，当一个事件与很多独立随机变量有关，该事件服从正态分布 。一般来说，连续值我们都倾向于假设服从正态分布。假设每个样本的误差$\epsilon^{(i)}$独立同分布均值为0，方差为σ的高斯分布$\epsilon^{(i)} - N(0, \sigma^2)$,所以有：

$$p\left(\epsilon^{(i)}\right) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{\left(\epsilon^{(i)}\right)^2}{2\sigma^2}\right)$$

即表示 $y^{(i)}$满足以均值为 $h_\theta(x^{(i)})$,方差为$\epsilon^{(i)}$的高斯分布。

$$p\left(y^{(i)}|x^{(i)};\theta\right) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)$$

由最大似然估计有：

$$
\begin{aligned}
\max L(\theta) &= L(\theta; x^{(i)}, y) = p(y^{(i)}|x^{(i)};\theta) \\
L(\theta; X, y) &= \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) \\
&= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}} exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \\
\max logL(\theta) &= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}} exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \\
&= mlog\frac{1}{\sqrt{2\pi}} - \frac{1}{2\sigma^2} \cdot \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2 \\
&\Leftrightarrow \min \frac{1}{2\sigma^2} \cdot \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2 = J(\theta)
\end{aligned}
$$

二、岭回归和Lasso回归

岭回归的目标函数在一般的线性回归的基础上加入了正则项，在保证最佳拟合误差的同时，使得参数尽可能的“简单”，使得模型的泛化能力强。正则项一般采用一，二范数，使得模型更具有泛化性，同时可以解决线性回归中不可逆情况，比如二范数对应的岭回归：

$$
\min_{\theta} \frac{1}{2} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \|\theta\|^2
$$

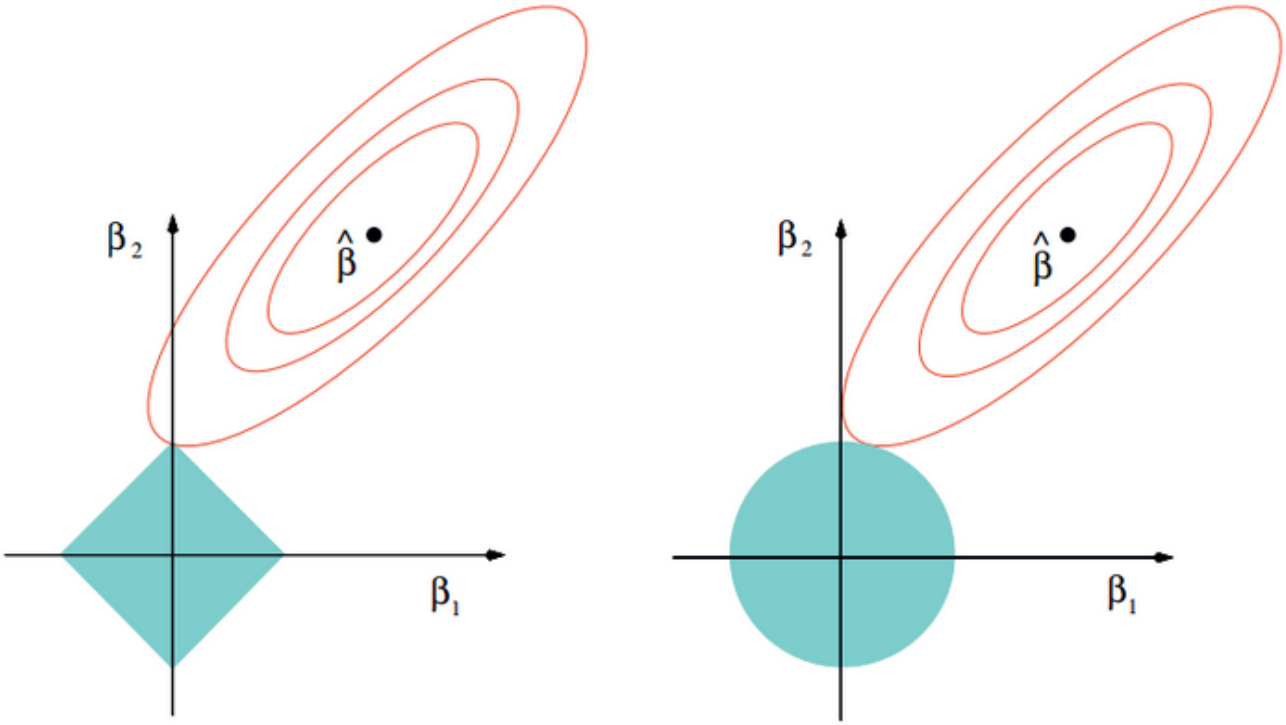
其迭代优化函数如下：

$$
\theta_j := \theta_j - \alpha \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)} - 2\lambda\theta_j
$$

另外从最小二乘的角度来看，通过引入二范正则项，使其主对角线元素来强制矩阵可逆。

$$
\begin{aligned}
\bigtriangledown_\theta J(\theta) &= XX^T\theta - XY + \lambda\theta = 0 \\
\Rightarrow \theta &= \left(XX^T + \lambda I\right)^{-1} XY
\end{aligned}
$$

Lasso回归采用一范数来约束，使参数非零个数最少。而Lasso和岭回归的区别很好理解，在优化过程中，最优解为函数等值线与约束空间的交集，正则项可以看作是约束空间。可以看出二范的约束空间是一个球形，一范的约束空间是一个方形，这也就是二范会得到很多参数接近0的值，而一范会尽可能非零参数最少。



值得注意的是线性模型的表示能力有限，但是并不一定表示线性模型只能处理线性分布的数据。这里有两种常用的线性模型非线性化。对于上面的线性函数的构造，我们可以看出模型在以$x_0, x_1, .., x_n$的坐标上是线性的，但是并不表示线性的模型就一定只能用于线性分布问题上。假如我们只有一个特征x_0，而实际上回归值是$y = x_0^2$等问题，我们同样可以采用线性模型，因为我们完全可以把输入空间映射到高维空间(x_1^3, x_1^2, x_1^1)，其实这也是核方法以及PCA空间变换的一种思想，凡是对输入空间进行线性，非线性的变换，都是把输入空间映射到特征空间的思想，所以只需要把非线性问题转化为线性问题即可。另外一种实现线性回归非线性表示能力的是局部线性思想，即对每一个样本构建一个加权的线性模型。

三、局部加权线性回归

考虑到线性回归的表示能力有限，可能出现欠拟合现象。局部加权线性回归为每一个待预测的点构建一个加权的线性模型。其加权的方式是根据预测点与数据集中点的距离来为数据集中的点赋权重，当某点距离预测点较远时，其权重较小，反之较大。由于这种权重的机制引入使得局部加权线性回归产生了一种局部分段拟合的效果。由于该方法对于每一个预测点构建一个加权线性模型，都要重新计算与数据集中所有点的距离来确定权重值，进而确定针对该预测点的线性模型，计算成本高，同时为了实现无参估计来计算权重，需要存储整个数据集。 局部加权线性回归，在线性回归基础上引入权重，其目标函数（下面的目标函数是针对一个预测样本的）如下：

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

$$\min_\theta J(\theta)$$

一般选择下面的权重函数，权重函数选择一般考虑数据的分布特性。

$$w^{(i)} = exp\left(-\frac{x^{(i)} - x}{2\sigma^2}\right)$$

其中x是待预测的一个数据点。

对于上面的目标函数，我们的目标同样是使得损失函数最小化，同样局部加权线性回归可以采用梯度的方法，也可以从最小二乘法的角度给出闭式解。

$$\nabla_\theta J(\theta) = XWX^T\theta - XWY = 0$$
$$\Rightarrow \theta = \left(XWX^T I\right)^{-1} XWY$$

其中W是对角矩阵，$W_{ii} = w^{(i)}$。

线性回归核心思想最小化平方误差，可以从最小化损失函数和最小二乘角度来看，也有概率解释。优化过程可以采用梯度方法和闭式解。在闭式解问题中需要注意矩阵可逆问题。考虑到过拟合和欠拟合问题，有岭回归和lasso回归来防止过拟合，局部加权线性回归通过加权实现非线性表示。

二、Logistic回归和SoftMax回归，最大熵模型

一、Logistic回归

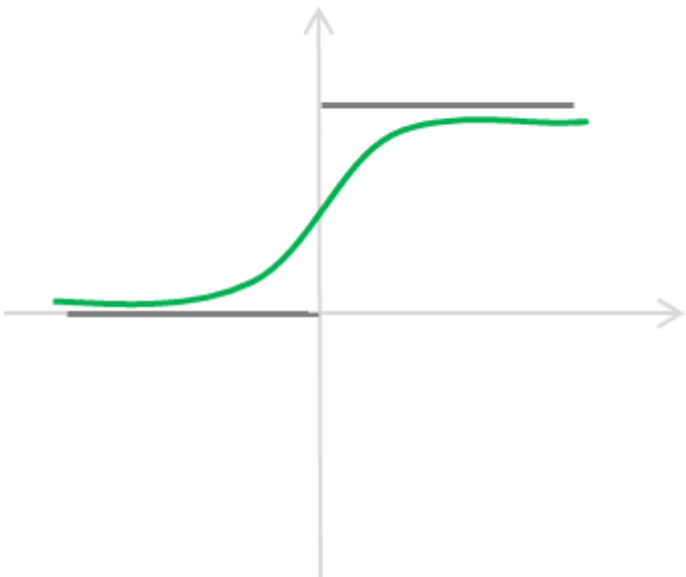
分类问题可以看作是在回归函数上的一个分类。一般情况下定义二值函数，然而二值函数构成的损失函数非凸，一般采用sigmoid函数平滑拟合（当然也可以看作是一种软划分，概率划分）：从函数图像我们能看出，该函数有很好的特性，适合二分类问题。至于为何选择Sigmoid函数，后面可以从广义线性模型导出为什么是Sigmoid函数。

逻辑回归可以看作是在线性回归的基础上构建的分类模型，理解的角度有多种（最好的当然是概率解释和最小对数损失），而最直接的理解是考虑逻辑回归是将线性回归值离散化。即一个二分类问题如下：（二值函数）

$$h_\theta(x^{(i)}) = g(\theta^T x) = \begin{cases} 1\ ,\ if\theta^T x \geq t \\ 0\ ,\ if\theta^T x < t \end{cases}$$

sigmoid函数

$$g(z) = \frac{1}{1+e^{-z}}, g^{`}(z) = g(z)(1-g(z))$$



$0-1$损失的二分类问题属于一种硬划分，即是与否的划分，而sigmoid函数则将这种硬划分软化，以一定的概率属于某一类（且属于两类的加和为1）。Sigmoid函数将线性回归值映射到$[0,1]$的概率区间，从函数图像我们能看出，该函数有很好的特性，适合二分类问题。 因此逻辑回归模型如下：

$$h_\theta(x^{(i)}) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

这里对于目标函数的构建不再是最小化函数值与真实值的平方误差了，按分类原则来讲最直接的损失因该是0-1损失，即分类正确没有损失，分类错误损失计数加1。但是0-1损失难以优化，存在弊端。结合sigmoid函数将硬划分转化为概率划分的特点，采用概率$h_\theta(x^{(i)})$的对数损失（概率解释-N次伯努利分布加最大似然估计），其目标函数如下：

$$J(\theta) = \sum_{i=1}^{m} - \left(y^{(i)} log(h_\theta(x^{(i)})) + (1-y^{(i)}) log(1-h_\theta(x^{(i)})) \right)$$
$$\min J(\theta)$$

同样采用梯度下降的方法有：

$$\begin{aligned}
\theta_j &:= \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j} \\
&= \theta_j - \alpha \frac{\partial \sum_{i=1}^{m} - \left(y^{(i)} log(h_\theta(x^{(i)})) + (1-y^{(i)}) log(1-h_\theta(x^{(i)})) \right)}{\partial \theta_j} \\
&= \theta_j - \alpha \left(-\frac{y^{(i)}}{h_\theta(x^{(i)})} + \frac{(1-y^{(i)})}{(1-h_\theta(x^{(i)}))} \right) \frac{\partial h_\theta(x^{(i)})}{\partial \theta_j} \\
&= \theta_j - \alpha \left(\frac{y^{(i)} - h_\theta(x^{(i)})}{h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))} \right) \frac{\partial h_\theta(x^{(i)})}{\partial \theta_j}
\end{aligned}$$

又：

$$\begin{aligned}
\frac{\partial h_\theta(x)}{\partial \theta} &= \left(\frac{1}{1+e^{-\theta^T x}} \right)' \\
&= \frac{\left(e^{-\theta^T x} \right)}{\left(\frac{1}{1+e^{-\theta^T x}} \right)^2} x \\
&= \left(\frac{1}{1+e^{-\theta^T x}} \right) \left(1 - \frac{1}{1+e^{-\theta^T x}} \right) x \\
&= h_\theta(x) \left(1 - h_\theta(x) \right) x
\end{aligned}$$

所以有：

$$\theta_j = \theta_j - \alpha \left(y^{(i)} - h_\theta(x^{(i)}) \right) x$$

概率解释

逻辑回归的概率解释同线性回归模型一致，只是假设不再是服从高斯分布，而是$p\left(y|x;\theta\right)$服从0-1分布，由于 ，假设随机变量y服从伯努利分布是合理的 。即：

$$\begin{aligned}
&p\left(y=1|x;\theta\right) = h_\theta(x) \\
&p\left(y=0|x;\theta\right) = 1 - h_\theta(x) \\
&p\left(y|x;\theta\right) = (h_\theta(x))^y . (1-h_\theta(x))^{(1-y)}
\end{aligned}$$

所以最大化似然估计有：

$$
\begin{aligned}
\max L(\theta) &= p\left(y|x;\theta\right) \\
&= \prod_{i=1}^{m} p\left(y^{(i)}|x^{(i)};\theta\right) \\
&= \prod_{i=1}^{m} \left(h_\theta(x^{(i)})\right)^{y^{(i)}} \cdot \left(1-h_\theta(x^{(i)})\right)^{\left(1-y^{(i)}\right)} \\
&\Leftrightarrow \max logL(\theta) \\
&\Leftrightarrow \min -logL(\theta) = \sum_{i=1}^{m} -\left(y^{(i)} log(h_\theta(x^{(i)})) + (1-y^{(i)}) log(1-h_\theta(x^{(i)}))\right)
\end{aligned}
$$

logistic采用对数损失（对数似然函数）原因

采用对数损失的原因有二：

1)从概率解释来看，多次伯努利分布是指数的形式。由于最大似然估计导出的结果是概率连乘，而概率（sigmoid函数）恒小于1，为了防止计算下溢，取对数将连乘转换成连加的形式，而且目标函数和对数函数具备单调性，取对数不会影响目标函数的优化值。

2）从对数损失目标函数来看，取对数之后在求导过程会大大简化计算量。

二、SoftMax回归

Softmax回归可以看作是Logistic回归在多分类上的一个推广。考虑二分类的另一种表示形式：

$$
[k_1, 1-k_1] \rightarrow \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}
$$

当logistic回归采用二维表示的话，那么其损失函数如下：

$$
\begin{aligned}
J\left(\theta\right) &= -\sum_{i=1}^{m}\sum_{k=1}^{2}\left(y^{(ik)} log\left(\frac{h_{\theta k}(x^{(i)})}{\sum_{k=1}^{K} h_{\theta k}(x^{(i)})}\right)\right) \\
&\min J\left(\theta\right)
\end{aligned}
$$

其中，在逻辑回归中两类分别为k_1，$1-k_1$二在softmax中采用k_1，k_2两个随机变量组成二维向量表示，当然隐含约束$k_1+k_2=1$.为了更好的表示多分类问题，将$y \in \{1,2,..K\}$（不一定理解为y的取值为k，更应该理解为y可以取k类）多分类问题进行如下表示：

$$
T(k) = \begin{bmatrix} 0 \\ 0 \\ . \\ 1 \\ . \\ 0 \end{bmatrix}
$$

其中向量的第k位为1，其他位为0，也就是当$y=k$ 时将其映射成向量时对应第k位为1。采用多维向量表示之后，那么对于每一维就变成了一个单独的二分类问题了，所以softmax函数形式如下：

$$
h_\theta(x^{(i)}) = \frac{1}{\sum_{k=1}^{K} exp\left(\theta_k^T x^{(i)}\right)} \begin{bmatrix} exp\left(\theta_k^T x^{(i)}\right) \\ exp\left(\theta_k^T x^{(i)}\right) \\ . \\ exp\left(\theta_k^T x^{(i)}\right) \end{bmatrix}
$$

其中函数值是一个K维的向量，同样采用对数损失（N元伯努利分布和最大似然估计），目标函数形式是logistic回归的多维形式。

$$J(\theta) = -\sum_{i=1}^{m}\sum_{k=1}^{K}\left(y^{(ik)} log\left(\frac{h_{\theta k}(x^{(i)})}{\sum_{k=1}^{K} h_{\theta k}(x^{(i)})}\right)\right)$$
$$\min J(\theta)$$

其中y^{ik}表示第i个样本的标签向量化后第k维的取值0或者1.可以看出Softmax的损失是对每一类计算其概率的对数损失，而logistic回归是计算两类的回归，其本质是一样。Logistic回归和Softmax回归都是基于线性回归的分类模型，两者无本质区别，都是从伯努利分结合最大对数似然估计。只是Logistic回归常用于二分类，而Softmax回归常用于多分类。而且Logistic回归在考虑多分类时只考虑$n-1$类。

概率解释(求导推导)

二分类与多分类可以看作是二元伯努利分布到多元伯努利分布的一个推广，概率解释同Logistic回归一致。详细解释放到广义线性模型中。

二分类转多分类思想

对于多分类问题，同样可以借鉴二分类学习方法，在二分类学习基础上采用一些策略以实现多分类，基本思路是“拆解法”，假设N个类别$C_1, C_2, .C_i., C_n$，经典的拆分算法有“一对一”，“一对多”，“多对多”，

一对一的基本思想是从所有类别中选出两类来实现一个两分类学习器，即学习出$C_N^2 = N(N-1)/2$个二分类器，然后对新样本进行预测时，对这 C_N^2个分类器进行投票最终决定属于那一类。

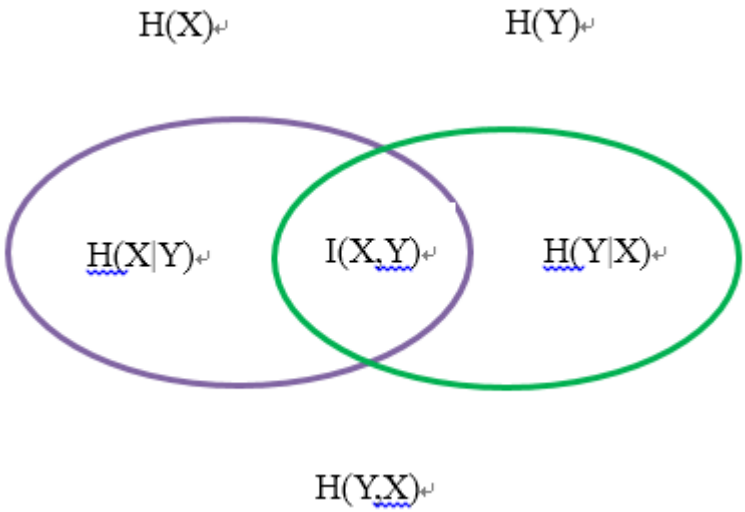
一对多的基本思想是把所有类别进行二分类，即属于C_i类和非C_i两类，这样我们就需要N个分类器，然后对新样本进行预测时，与每一个分类器比较，最终决定属于哪一类。这其实就是Softmax的思想，也是SVM多分类的思想。

//多对多的基本思想是

三、最大熵模型

很奇怪，为什么会把最大熵模型放到这，原因很简单，它和Logistic回归和SoftMax回归实在是惊人的相似，同属于对数线性模型。

熵的概念



信息熵：熵是一种对随机变量不确定性的度量，不确定性越大，熵越大。若随机变量退化成定值，熵为0。均匀分布是“最不确定”的分布 。

假设离散随机变量X的概率分布为$P(X)$，则其熵为：

$$H(X) = -\sum_{x} P(x)logP(x)$$

其中熵满足不等式$0 \leq H(P) \leq log|X|$，$|X|$为X取值数。

联合熵：对于多个随机变量的不确定性可以用联合熵度量

假设离散随机变量X, Y的联合概率分布为$P(X, Y)$，则其熵为：

$$H(X, Y) = -\sum_{x}\sum_{y} P(x, y)logP(x, y)$$

条件熵：在给定条件下描述随机变量的不确定性

假设离散随机变量X, Y，在给定Y的条件下X的不确定性为条件熵H(X|Y)，也就等于$H(X, Y) - H(Y)$

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

互信息：衡量两个随机变量相关性的大小$I(X, Y) = H(X) + H(Y) - H(X, Y)$

$$I(X, Y) = -\sum_{x,y} P(x, y)log\frac{P(x, y)}{P(x)P(y)}$$

相对熵（KL散度）：衡量对于同一个随机变量两个概率分布$p(x), q(x)$的差异性

$$D(p||q) = \sum_{x} p(x)log\frac{p(x)}{q(x)} = E_{p(x)}log\frac{p(x)}{q(x)}$$

有互信息和相对熵的定义有下式：

$$I(X, Y) = D(P(X, Y)||P(X)P(Y))$$

关于熵的介绍就到此，不细究，虽然上面的这些定义在机器学习中都会遇到，不过后面涉及到的主要还是熵和条件熵，互信息。

最大熵模型

最大熵原理是概率模型学习中的一个准则。最大熵原理认为，学习概率模型时，在所有可能的概率模型分布中（满足所有条件下），熵最大的模型是最好的模型。熵最大即为最均匀的分布，从某种角度讲均匀分布总是符合我们理解的损失风险最小，也就是“不要不所有的鸡蛋放到一个篮子里，均匀的放置”。

给定训练集$T = \{(x_1, y_1), (x_2, y_2)..(x_m, y_m)\}$，假设$X \in \chi \subseteq R^n$表示输入，$y \in \phi$表示输出，分类模型是一个以条件概率分布$P(Y|X)$输出$Y$，也就是说在满足条件的所有可能集中，条件熵$P(Y|X)$最大的模型即为最好的模型。其中条件为隐藏在数据的期望。

一般来讲，最大熵模型常用于处理离散化数据集，定义随机变量X, Y的特征模板，从数据中统计他们的期望作为最大熵模型的条件

特征函数：

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

约束条件：对于任意的特征函数f，我们可以统计其在数据中的经验分布$\widetilde{P}(x,y)$的期望：

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

特征函数f关于模型$P(Y|X)$和先验$\widetilde{P}(X)$的条件期望：

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

所以，满足约束条件的模型集合为：

$$\Omega\equiv\{P\in\boldsymbol{P}|E_p(f_i)=E_{\tilde{p}}(f_i),i=1..n\}$$

因此最大熵模型的形式化表示如下：

$$\begin{aligned}&\max_{P\in C}H(P)=-\sum_{x,y}\widetilde{P}(x)P(y|x)logp(y|x)\\\Leftrightarrow&\min_{P\in C}-H(P)=\sum_{x,y}\widetilde{P}(x)P(y|x)logp(y|x)\\&s.t.E_p(f_i)=E_{\tilde{p}}(f_i),i=1..n\\&\sum_y P(y|x)=1\end{aligned}$$

由拉格让日乘子法，引入拉格让日乘子，定义拉格让日函数：

$$\begin{aligned}&L(P,w)=-H(P)+w_0(1-\sum_y P(y|x))+\sum_i w_i(E_p(f_i)-E_{\tilde{p}}(f_i))\\&=\sum_{x,y}\widetilde{P}(x)P(y|x)logp(y|x)+w_0(1-\sum_y P(y|x))+\sum_i w_i(\sum_{x,y}(\widetilde{P}(x)P(y|x)f_i(x,y)-\sum_{x,y}\widetilde{P}(x,y)f_i(x,y))\\&s.t.\bigtriangledown L(P,w)=0\\&(1-\sum_y P(y|x))=0\\&\sum_{x,y}(\widetilde{P}(x)P(y|x)f_i(x,y)-\sum_{x,y}\widetilde{P}(x,y)f_i(x,y)=0,i=1..n\\&w_i\geq 0,i=1..n\end{aligned}$$

根据拉格朗日乘子法，$L(P)\geq L(P,w)$，当且仅当满足拉格朗日乘子法的所有必要条件等式成立，原问题也就是一个最小化最大问题

$$\min_{P\in C}\max_{w}L(P,w)$$

里层是max最大化$L(P,w)$，外层的min最小化$L(P)$。

对偶问题是：

$$\max_{w}\min_{P\in C}L(P,w)$$

求解对偶问题，第一步最小化内部$\min_{P\in C}L(P,w)$，$\min_{P\in C}L(P,w)$是关于w的函数，最优解记为P_w：

$$P_w=arg\min_{P\in C}L(P,w)=P_w(y|x)$$

那么外层最大化目标函数为：

$$\max_w \Phi(w)$$
$$\Phi(w) = \min_{p \in C} L(P, w) = L(P_w, w)$$

为了求解$P_w(y|x)$，根据KKT条件对$P(y|x)$求偏导：

$$\begin{aligned}\frac{\partial L(P,w)}{\partial P(y|x)} &= \sum_{x,y} \widetilde{P}(x)(logP(y|x)+1) - \sum_y w_0 - \sum_{x,y}\left(\widetilde{P}(x)\sum_i w_i f_i(x,y)\right) \\ &= \sum_{x,y}\widetilde{P}(x)\left(logP(y|x)+1-w_0-\sum_i w_i f_i(x,y)\right) \\ &= 0\end{aligned}$$

求解得：

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

这里，虽然我们不知道w_0，但是由于$\sum_y P(y|x) = 1$,所以分母一定是对y的所有可能的归一化因子

$$P_w(y|x) = \frac{1}{z_w(x)}\left(exp\sum_i w_i f_i(x,y)\right)$$
$$z_w(x) = \sum_y exp(\sum_i w_i f_i(x,y))$$

因此，$\max_w \Phi(w)$的最优解为：

$$w^* = arg\max_w \Phi(w)$$

代回$P_w(y|x)$，我们可以得到最终的分类模型，同样我们发现最大熵模型也是一个对数线性模型。

回顾对偶函数，内部最小化求解得到了$P_w(y|x)$，回到外部目标$\max_w \Phi(w)$，将$P_w(y|x)$代回拉格朗日函数有：

$$\begin{aligned}\Phi(w) &= \sum_{x,y}\widetilde{P}(x)P_w(y|x)logP_w(y|x) + \sum_{i=1}^n w_i\left(\sum_{x,y}\widetilde{P}(x,y)f_i(x,y) - \sum_{x,y}\widetilde{P}(x)P_w(y|x)f_i(x,y)\right) \\ &= \sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^n w_i f_i(x,y) + \sum_{x,y}\widetilde{P}(x)P_w(y|x)\left(logP_w(y|x) - \sum_{i=1}^n w_i f_i(x,y)\right) \\ &= \sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^n w_i f_i(x,y) - \sum_{x,y}\widetilde{P}(x)P_w(y|x)logz_w(x) \\ &= \sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^n w_i f_i(x,y) - \sum_x \widetilde{P}(x)logz_w(x)\sum_y P_w(y|x) \\ &= \sum_{x,y}\widetilde{P}(x,y)\sum_{i=1}^n w_i f_i(x,y) - \sum_x \widetilde{P}(x)logz_w(x)\end{aligned}$$

概率解释：

已知训练集的经验概率分布$\widetilde{P}(x,y)$，条件概率分布$P(y|x)$的对数似然函数为：

$$L_{\tilde{P}}(P_w) = log\prod_{x,y} P(y|x)^{\tilde{P}(x,y)} = \sum_{x,y}\tilde{P}(x,y)logP(y|x)\text{，特征统计}$$

$$Logistic: \max logL(\theta) = p\left(y|x;\theta\right) = log\prod_{i=1}^{m}\prod_{k=1}^{K}\left(h_\theta(x^{(i)})\right)^{y^{(i)}}\text{，样本统计}$$

其中，我们发现对数似然函数与条件熵的形式一致，最大熵模型目标函数前面有负号（这与最大化对数似然函数完全相反），同时最大熵模型中有约束条件。也正是因为约束条件，我们将原问题转化为对偶问题后发现，在满足约束条件的对偶函数的极大化等价于最大化对数似然函数。

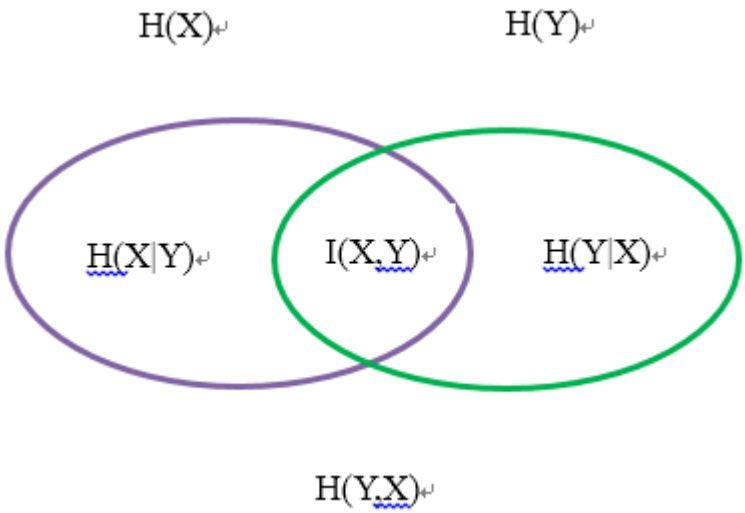
当条件概率$P(y|x)$满足约束条件，在对偶问题求解过程中我们有：

$$P_w(y|x) = \frac{1}{z_w(x)}\left(exp\sum_i w_i f_i(x,y)\right)$$
$$z_w(x) = \sum_y exp(\sum_i w_i f_i(x,y))$$

代入到对数似然函数，同样有：

$$\begin{aligned}
L_{\tilde{P}}(P_w) &= \sum_{x,y}\tilde{P}(x,y)logP(y|x) \\
&= \sum_{x,y}\tilde{P}(x,y)\left(\sum_{i=1}^{n} w_i f_i(x,y) - logz_w(x)\right) \\
&= \sum_{x,y}\tilde{P}(x,y)\sum_{i=1}^{n} w_i f_i(x,y) - \sum_{x,y}\tilde{P}(x,y)logz_w(x) \\
&= \sum_{x,y}\tilde{P}(x,y)\sum_{i=1}^{n} w_i f_i(x,y) - \sum_{x}\tilde{P}(x)logz_w(x) = \Phi(w)
\end{aligned}$$

最后，我们再来看对偶函数表达式，我们发现，第一项其实是X,Y的联合熵$H(X,Y)$，第二项是X的信息熵$H(X)$，回看熵的示意图，我们发现，我们的目标还是最大化条件熵$H(Y|X)$。



下面再来对比下Logistic回归，SoftMax回归，最大熵模型

1）同属于对数线性模型

2）Logistic回归和SoftMax回归都基于条件概率$P(y|x)$，满足一个伯努利分布，N重伯努利分布；而最大熵模型以期望为准，没有该假设

3）由于都采用线性模型，三者都假设特征之间是独立的

最大熵模型的优化问题

最大熵模型从拉格朗日乘子法最大化对偶函数，还是从最大化对数似然函数，其目标函数如下：

$$L_{\widetilde{P}}(P_w) = \sum_{x,y} \widetilde{P}(x,y) \sum_{i=1}^{n} w_i f_i(x,y) - \sum_{x} \widetilde{P}(x) log Z_w(x)$$

常用的梯度优化算法都可以，另外对于最大熵模型也有专门的算法有GIS IIS 算法 。

三、广义线性模型

从线性回归，logistic回归，softmax回归，最大熵的概率解释来看，我们会发现线性回归是基于高斯分布+最大似然估计的结果，logistic回归是伯努利分布+对数最大似然估计的结果，softmax回归是多项分布+对数最大似然估计的结果，最大熵是基于期望+对数似然估计的结果。前三者可以从广义线性模型角度来看。

指数分布家族

指数分布家族是指可以表示为指数形式的概率分布，指数分布的形式如下：

$$p(y;\eta) = b(y) exp(\eta^T T(y) - \alpha(\eta))$$

其中η是分布的自然参数，$T(y)$是充分统计了，通常$T(y) = y$.当参数a, b, T都固定的时候，就定义了一个以η为参数的函数族。

实际上大多数的概率分布都属于指数分布家族，比如

1）伯努利分布 0-1问题

2）二项分布，多项分布 多取值 多次试验

3）泊松分布 计数过程

4）伽马分布与指数分布

5）β分布

6）Dirichlet分布

7）高斯分布

现在我们将高斯分布和伯努利分布用指数分布家族的形式表示：

高斯分布

$$\begin{aligned} p\left(y^{(i)}|x^{(i)};\theta\right) &= \frac{1}{\sqrt{2\pi}} exp\left(-\frac{\left(y^{(i)}-\mu\right)^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}y^2 - \frac{1}{2}\mu^2 + \mu y) \\ &= \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}y^2) exp(\mu y - \frac{1}{2}\mu^2) \end{aligned}$$

对应到指数分布家族有：

$$
\begin{aligned}
&b(y)=\frac{1}{\sqrt{2\pi}}exp(-\frac{1}{2}y^2)\\
&T(y)=y\\
&\eta=\mu\\
&a(\eta)=\frac{1}{2}\mu^2
\end{aligned}
$$

伯努利分布

$$
\begin{aligned}
p\left(y|x;\theta\right)&=(\phi)^y.(1-\phi)^{(1-y)}\\
&=exp(ylog\phi+(1-y)log(1-\phi))\\
&=1\cdot exp(log(\frac{\phi}{1-\phi})y+log(1-\phi))
\end{aligned}
$$

对应到指数分布家族有：

$$
\begin{aligned}
&b(y)=1\\
&T(y)=y\\
&\eta=log\frac{\phi}{1-\phi}\Rightarrow\phi=\frac{1}{1+e^{-\eta}}\\
&a(\eta)=-log(1-\phi)=log(1+e^{\eta})
\end{aligned}
$$

广义线性模型

在了解指数分布家族之后，我们再来看广义线性模型的形式定义与假设：

1）$y|x;\theta\sim ExpFamily(\eta)$; 给定样本x与参数$\theta$，样本分类y服从指数分布家族的某个分布

2）给定一个x，我们目标函数为$h_\theta(x)=E[T(y)|x]$

3) $\eta=\theta^Tx$

三条假设，第一条是为了能在指数分布范围内讨论y的概率，第二条假设是为了使得预测值服从均值为实际值得一个分布，第三条假设是为了设计的决策函数（模型）是线性的。

由高斯分布的指数家族分布形式与广义线性模型的定义有线性回归的模型为：

$$h_\theta(x)=E[T(y)|x]=E[y|x]=\mu=\eta=\theta^Tx$$

同样由伯努利分布的指数家族分布形式与广义线性模型的定义有logistic回归的模型为（解释了为什么是sigmoid函数）：

$$
\begin{aligned}
h_\theta(x)=E[T(y)|x]=E[y|x]=p(y=1|x;\theta)=\phi\\
\phi=\frac{1}{1+e^{-\eta}}=\frac{1}{1+e^{-\theta^Tx}}
\end{aligned}
$$

所以，在广义线性模型中，决策函数为线性函数是基于广义线性模型的第三条假设，而最终的模型是依赖于模型服从什么样的分布，比如 高斯分布，伯努利分布。

同样，我们应用logistic回归到softmax回归的一套定义，下面再来看多项分布对应的softmax回归

$$
\begin{aligned}
p\left(y|x;\theta\right) &= \left(\phi_1\right)^{l(y=1)}.\left(\phi_2\right)^{l(y=2)}\ldots\left(\phi_{k-1}\right)^{l(y=k-1)}.\left(\phi_k\right)^{l(y=k)} \\
&= \left(\phi_1\right)^{l(y=1)}.\left(\phi_2\right)^{l(y=2)}\ldots\left(\phi_{k-1}\right)^{l(y=k-1)}.\left(\phi_k\right)^{1-\sum_{i=1}^{k-1}l(y=i)} \\
&= 1\cdot exp(log((\phi_1)^{l(y=1)}.(\phi_2)^{l(y=2)}\ldots(\phi_{k-1})^{l(y=k-1)}.(\phi_k)^{1-\sum_{i=1}^{k-1}l(y=i)})) \\
&= exp(\sum_{i=1}^{k-1}l(y=i)log(\phi_i)+(1-\sum_{i=1}^{k-1}l(y=i))log(\phi_k)) \\
&= exp(\sum_{i=1}^{k-1}l(y=i)log\frac{\phi_i}{\phi_k}+log\phi_k) \\
&= exp(\sum_{i=1}^{k-1}T(y)_i log\frac{\phi_i}{\phi_k}+log\phi_k) \\
&= exp(\eta^T T(y)-\alpha(\eta))
\end{aligned}
$$

其中ϕ_i是表示$y=i$的概率，$l(y=i)$是一个指示函数，为真是取值为1，否则为0，$T(y)_i$采用softmax中向量化的定义。

对应到指数分布家族有：

$$
\begin{aligned}
b(y) &= 1 \\
T(y) &= \sum_{i=1}^{k-1}T(y)_i \\
\eta &= \begin{bmatrix} log\left(\phi_1/\phi_k\right) \\ log\left(\phi_2/\phi_k\right) \\ . \\ log\left(\phi_{k-1}/\phi_k\right) \end{bmatrix} \\
a(\eta) &= -log(\phi_k)
\end{aligned}
$$

由η推出：

$$
\eta_i = log\phi_i/\phi_k \Rightarrow \phi_i = \phi_k e^{\eta_i}\, i\in 1..k-1
$$

为了方便定义$\eta_k = log\phi_k/\phi_k = 0$，由于多项分布所有值取值概率加和为1有：

$$
\sum_{i=1}^{k}\phi_i = \sum_{i=1}^{k}\phi_k e^{\eta_i} = 1 \Rightarrow \phi_k = \frac{1}{\sum_{i=1}^{k}e^{\eta_i}}
$$

所以有$p(y=i|x;\phi)=\phi_i=\frac{e^{\eta_i}}{\sum_{i=1}^{k}e^{\eta_i}}$.

再由广义线性模型的第二条假设，同时将第三条线性假设$\eta=\theta^T x$带入有：

$$
\begin{aligned}
h_\theta(x) &= E[T(y)|x;\theta] \\
&= \begin{bmatrix} \phi_1 \\ \phi_2 \\ \cdot \\ \phi_k \end{bmatrix} \\
&= \begin{bmatrix} \frac{exp(\theta_k^T x^{(1)})}{\sum_{k=1}^K exp(\theta_k^T x^{(i)})} \\ \frac{exp(\theta_k^T x^{(i)})}{\sum_{k=1}^K exp(\theta_k^T x^{(i)})} \\ \cdot \\ \frac{exp(\theta_k^T x^{(k)})}{\sum_{k=1}^K exp(\theta_k^T x^{(i)})} \end{bmatrix}
\end{aligned}
$$

最后由最大似然估计有softmax的目标函数如下：

$$
\begin{aligned}
L(\theta) &= \sum_{i=1}^{m} logp(y^{(i)}|x^{(i)};\theta) \\
&= \sum_{i=1}^{m}\sum_{k=1}^{K}\left(y^{(ik)} log\left(\frac{h_{\theta k}(x^{(i)})}{\sum_{k=1}^K h_{\theta k}(x^{(i)})}\right)\right) \\
&\max L(\theta)
\end{aligned}
$$

到此，广义线性模型解释线性回归，logistic回归，softmax回归基本算完，可以看出线性函数是基于广义线性模型的第三条假设，采用sigmoid函数是因为伯努利分布，而softmax回归是logistic回归高维推广。

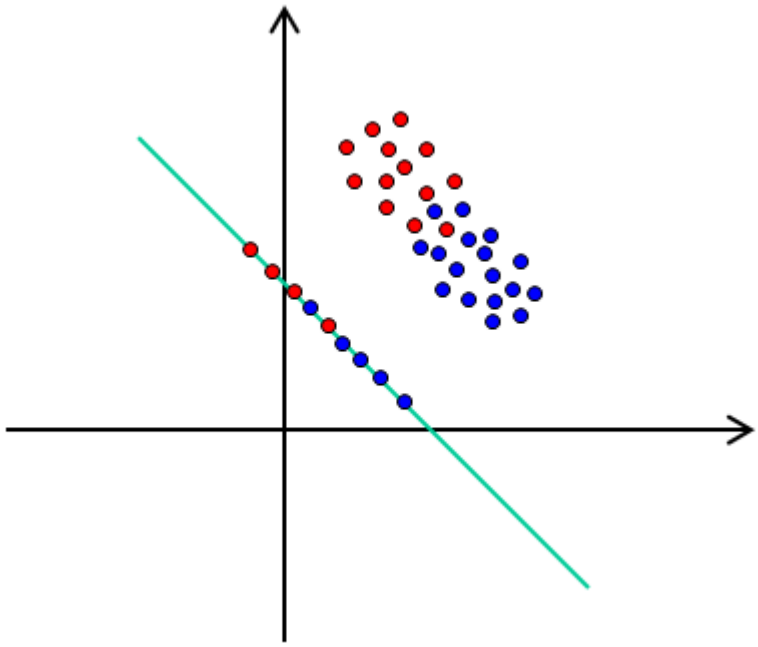
四、Fisher线性判别与线性感知机

Fisher线性判别和线性感知机都是针对分类任务，尤其是二分类，二者的共同之处在于都是线性分类器，不同之处在于构建分类器的思想，但是二者有异曲同工之妙。同时二者又可以与logistic回归进行对比，当然logistic回归的理论基础是概率。

一、Fisher线性判别

Fisher线性判别是一种线性分类思想，其核心是找一个投影方向将d维数据投影（降维）到一维，使得类内紧致，类间分离。在确定投影方向之后，决策分类器还并未完成，我们还需要分界点来划分不同的类。一般而言很少用Fisher线性判别作为分类模型，更多是借鉴Fisher线性判别的思想来指导降维。

以两类问题来分析：



由类内紧致，类间分离准则确定投影方向，我们可以定义如下类内距离和类间距离

$$\tilde{m}_i = \frac{1}{n_i}\sum_{y_j \in X_i}^{i=1,2} y_j = \frac{1}{n_i}\sum_{y_j \in X_i}^{i=1,2} w^T x_j = w^T m_i$$

其中$\tilde{m}_i$表示降维后的类中心，m_i表示原始空间的类中心。

类内距离：

$$\begin{aligned}
\tilde{S}_w &= \sum_{y_j \in X_i}^{i=1,2} (y_j - \tilde{m}_i)^2 \\
&= \sum_{x_j \in X_i}^{i=1,2} \left(w^T x_j - \frac{1}{n_i}\sum_{x_j \in X_i}^{i} w^T x_j \right)^2 \\
&= w^T \sum_{x_j \in X_i}^{i=1,2} \left(x_j - \frac{1}{n_i}\sum_{x_j \in X_i}^{i} x_j \right)^2 w \\
&= w_T S_w w
\end{aligned}$$

其中$S_w = \sum_{x_j \in X_i}^{i=1,2} (x_j - m_i)^2$表示每类样本在原始空间的一个类内距离。

类间距离：

$$\begin{aligned}
\tilde{S}_b &= (\tilde{m}_1 - \tilde{m}_2)^2 \\
&= \left(\frac{1}{n_1}\sum_{x_j \in X_1} w^T x_j - \frac{1}{n_2}\sum_{x_j \in X_2} w^T x_j \right)^2 \\
&= w^T \left(\frac{1}{n_1}\sum_{x_j \in X_1} x_j - \frac{1}{n_2}\sum_{x_j \in X_2} x_j \right)^2 w \\
&= w^T S_b w
\end{aligned}$$

其中$S_b = \sum_{x_j \in X_i}^{i=1,2} (m_1 - m_2)^2$表示每类样本在原始空间的一个类间距离。

所以为了使类内紧致，类间分离，可以最大化如下目标函数：

$$\max J(w) = \frac{\tilde{S}_b}{\tilde{S}_w} = \frac{w^T S_b w}{w^T S_w w}$$

等价于$\max(w^T S_b w) s.t. w^T S_w w = c$,由拉格朗日乘子法有

$$L(w, \lambda) = w^T S_b w - \lambda(w^T S_w w - c)$$

可以看出目标函数是关于w的二次凸规划，极值在导数为0处取到，对上式求导有$S_b w^* - \lambda S_w w^* = 0$,如果$S_w$可逆的话，即

$$(S_b - \lambda S_w)w = 0$$
$$S_w^{-1} S_b w^* = \lambda w^*$$

也就是说w是$(S_b - \lambda S_w)$的特征向量。将$S_b = (m_1 - m_2)^2$带入有

$$S_w^{-1}(m_1 - m_2)(m_1 - m_2^T)w^* = \lambda w^*$$

又$(m_1 - m_2^T)w^*$是一个标量，所以$w^* = S_w^{-1}(m_1 - m_2)$.

虽然我们确定了投影方向，但是真正的决策函数还是未能确定，一般最简单的做法是直接在一维上找一个阈值直接将两类分开，但是如何确定阈值还需要定义分类的损失函数（类一致性准则）。

比如我们直接采用0-1损失，那么决策界则尽可能多的正确分类样本，另外如果我们采用Logistic回归的对数损失，那么我们的决策边界就不一样。从这个角度来看，线性判别分析和Logistic回归都是将数据映射到一维来进行分类，有没有不用降维直接进行分类的方法，下面就是感知机的分类思想。

二、线性感知机

线性感知机同样是基于线性分类思想，其核心是直接在高维空间找到一个超平面将两类样本尽可能的分开。即，定义点到超平面的距离，在保证线性可分的基础上最小化点到超平面的距离（等价于使得最难分的样本离超平面距离尽可能的大），但由于没有线性可分前提，所以感知机的目标函数是最小化错分样本到超平面的距离之和（线性损失），而分类正确的样本无损失。

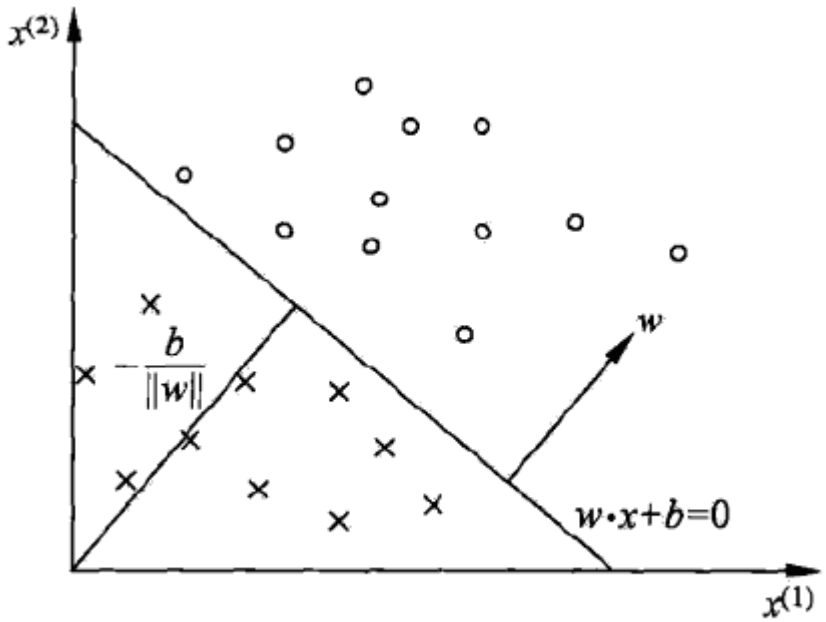


图 2.1　感知机模型

首先定义线性超平面：

$$w \cdot x + b = 0$$

点到超平面的距离为$\frac{1}{||w||^2}(w \cdot x_i + b)$，即错分的样本到超平面的距离一定为负：

$$-\frac{1}{||w||^2} y_i (w \cdot x_i + b) > 0$$

感知机只考虑分类错误的样本，目标函数为最小化错分样本到超平面的距离之和：

$$\min -\sum_{x_i \in M} \frac{1}{||w||^2} y_i (w \cdot x_i + b)$$

其中M为错分样本集合。因为感知机优化的目标是针对错误样本集来不断的调整参数，所以在使用梯度下降算法的时候梯度的计算只依赖于错分的样本。

找到超平面之后我们就可以基于超平面定义决策函数为

$$f(x) = sign(w \cdot x + b)$$
$$sign(x) = \begin{cases} +1\ , \ x \geq 0 \\ -1, x < 0 \end{cases}$$

再回过头看损失函数，我们发现感知机的损失与Logistic回归的对数损失不一样，感知机采用的损失直接是错误样本的$f(x)$值，而Logistic回归的损失是所有样本的对数损失（当然Logistic回归的$f(x) = sigmoid(x)$）。

感知机的对偶形式

感知机的对偶形式是logistic回归一致，同SVM对偶形式推导一致

可以说神经网络和SVM都是线性感知机的一种延伸，神经网络是引入非线性激活函数，而SVM则是使用核函数，SVM同时提出了软间隔分类。虽然说神经网络是从感知机过来的，但是神经网络引入非线性激活函数后，不仅失去了解释性，也使其与感知机渐行渐远，笔者倒是觉得SVM更像感知机，不仅提升了精度，同时保留了很好的解释性。

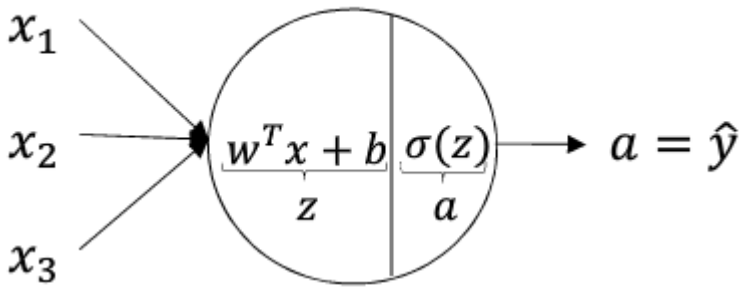
五、三层神经网络

一、神经单元

深度学习的发展一般分为三个阶段，感知机-->三层神经网络-->深度学习（表示学习）。早先的感知机由于采用线性模型，无法解决异或问题，表示能力受到限制。为此三层神经网络放弃了感知机良好的解释性，而引入非线性激活函数来增加模型的表示能力。三层神经网络与感知机的两点不同

1）非线性激活函数的引入，使得模型能解决非线性问题

2）引入激活函数之后，不再会有0损失的情况，损失函数采用对数损失，这也使得三层神经网络更像是三层多元（神经单元）逻辑回归的复合

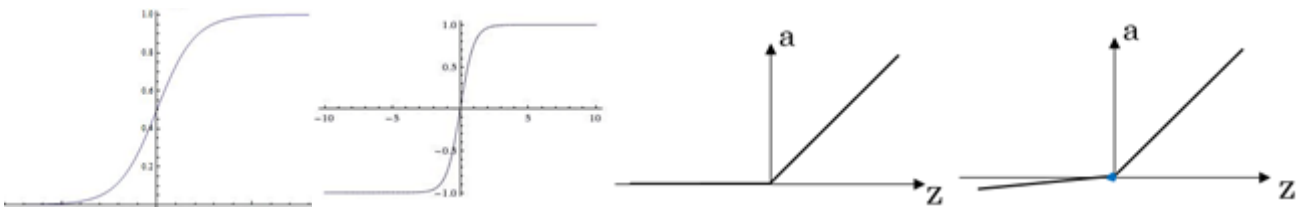


$$z = w^T x + b$$

$$a = \sigma(z)$$

神经网络中每一个神经元都可以看作是一个逻辑回归模型，三层神经网络就是三层逻辑回归模型的复合，只是不像逻辑回归中只有一个神经元，一般输入层和隐藏层都是具有多个神经元，而输出层对应一个logistic回归单元或者softmax单元，或者一个线性回归模型。

这里对一些常用的非线性激活函数做一些简单的介绍（图像，性质，导数）



$$sigmoid(z) = \frac{1}{1+e^{-z}}, tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, relu(z) = max\{0, z\}, leakyrelu(z) = \begin{cases} \alpha z, z < 0 \\ z, z \geqslant 0 \end{cases}$$

性质：对于以上几个非线性激活函数都可以看作是$\begin{cases}0, z<0\\1, z\geqslant 0\end{cases}$，的一个近似。采用近似的一个重要原因是为了求导，早起常采用平滑的sigmoid和tanh函数，然而我们可以发现这两个函数在两端都存在导数极小的情况，这使得多层神经网络在训练时梯度消失，难以训练。而Relu函数则很好的解决两端导数极小的问题，也是解决神经网络梯度消失问题的一种方法。

导数：

$$sig(z)=\frac{1}{1+e^{-z}}, d(z)=-\frac{-e^{-z}}{(1+e^{-z})^2}=\frac{e^{-z}+1}{(1+e^{-z})^2}-\frac{1}{(1+e^{-z})^2}=\frac{1}{(1+e^{-z})}-\left(\frac{1}{1+e^{-z}}\right)^2=z(1-z)$$

$$tanh(z)=\frac{e^z-e^{-z}}{e^z+e^{-z}}dz=\frac{\left(e^z+e^{-z}\right)^2-\left(e^z-e^{-z}\right)^2}{(e^z+e^{-z})^2}=\frac{(e^z+e^{-z})-(e^z-e^{-z})}{(e^z+e^{-z})^2}=1-z^2$$

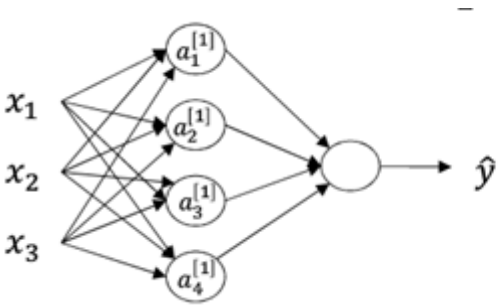
$$relu(z)=max\left\{0,z\right\}dz=\left\{0,1\right\}$$

$$leakyrelu(z)=max\left\{0,z\right\}dz=\left\{\alpha,1\right\}$$

二、前向传播

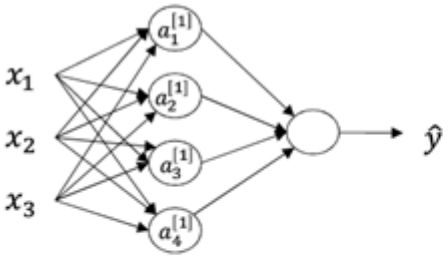
前向传播是一个复合函数的过程，每一个神经元都是一个线性函数加一个非线性函数的复合，整个网络的结构如下：其中上标表示网络层，所以$z^{[2]}$表示输出层。

向量形式：



$$z_1^{[1]}=w_1^{[1]T}x+b_1^{[1]},\ a_1^{[1]}=\sigma(z_1^{[1]})$$
$$z_2^{[1]}=w_2^{[1]T}x+b_2^{[1]},\ a_2^{[1]}=\sigma(z_2^{[1]})$$
$$z_3^{[1]}=w_3^{[1]T}x+b_3^{[1]},\ a_3^{[1]}=\sigma(z_3^{[1]})$$
$$z_4^{[1]}=w_4^{[1]T}x+b_4^{[1]},\ a_4^{[1]}=\sigma(z_4^{[1]})$$

矩阵形式：



Given input x:

$$z^{[1]}=W^{[1]}x+b^{[1]}$$
$$a^{[1]}=\sigma(z^{[1]})$$
$$z^{[2]}=W^{[2]}a^{[1]}+b^{[2]}$$
$$a^{[2]}=\sigma(z^{[2]})$$

其中线性函数还是$z=w^Tx+b$，不过要注意的是这里由于每一层不仅一个神经元，所以逻辑回归中的向量w则扩展为矩阵，表示有多个神经元（也正是因为多个神经元，导致神经网络具有提取特征的能力）。非线性函数则可以有以下选择，目前来看Relu函数具有一定的优势。

其中值得注意的是矩阵的行列，深度学习常采用一列表示一个样本，所以网络中数据矩阵的大小如下：

$$X=(n,m),Y=(1,m),W=(n^{(l)},n^{(l-1)}),b=(n^{(l)},1),Z=(n^{(l)},m),A=(n^{(l)},m)$$

损失函数同样采用对数损失(二分类)：

$$J(\theta)=\sum_{i=1}^{m}-\left(y^{(i)}log(h_\theta(x^{(i)}))+(1-y^{(i)})log(1-h_\theta(x^{(i)}))\right)$$
$$\min J(\theta)$$

三、反向传播

由于神经网络是一个多层的复合函数，前向传播就是在计算复合函数，所以反向传播就是一个链式求导过程，确定所有参数的负梯度方向，采用梯度下降的方法来更行每一层网络的参数。

1) 损失函数：

$$\frac{\partial J(\theta)}{\partial AL}=-\left(Y\frac{1}{AL}-(1-Y)\frac{1}{(1-AL)}\right)=\frac{Y-AL}{AL(1-AL)}$$

2) 激活函数：

$$\frac{\partial AL}{\partial Z}=AL(1-AL),\frac{\partial AL}{\partial Z}=1-AL^2,\frac{\partial AL}{\partial Z}=\begin{cases}1,Z\geq 0\\0,Z<0\end{cases}$$

3) 线性函数：

$$\frac{\partial Z}{\partial W}=\frac{1}{m}A^{(l-1)},\frac{\partial Z}{\partial b}=\frac{1}{m},\frac{\partial Z}{\partial A^{(l-1)}}=\frac{1}{m}W^{(l)}$$

对于损失函数直接对各个变量求导如下：

$$\begin{aligned}\frac{\partial J(\theta)}{\partial AL}&=\frac{Y-AL}{AL(1-AL)}\\\frac{\partial J(\theta)}{\partial Z}&=\frac{Y-AL}{AL(1-AL)}*(AL(1-AL))=Y-AL,(sigmoid)\\\frac{\partial J(\theta)}{\partial W^{(l-1)}}&=\frac{1}{m}(Y-AL)A^{(l-1)^T}\\\frac{\partial J(\theta)}{\partial b^{(l-1)}}&=\frac{1}{m}(Y-AL)\\\frac{\partial J(\theta)}{\partial A^{(l-1)}}&=W^T(Y-AL)\end{aligned}$$

值得注意的是激活函数是一个数值操作，不涉及矩阵求导，线性函数中$\frac{1}{m}$是因为w是作用于m个样本，所以在确定负梯度方向时需要m个样本取均值，而对A求导则不需要求均值。

六、支持向量机

在线性模型中，Fisher线性判别和线性感知机可以说是以上所有模型的分类依据，前者是映射到一维执其两端进行分类，后者是在高维空间找一个线性超平面将两类分开（两类可扩展到多类）。支持向量机属于后者，但主要有以下几点改进：

1）提出硬间隔线性可分，在感知机的基础上提出了线性可分假设（无损失），最大化最小间隔

2）提出软间隔线性可分，得到了hinge损失代替感知机的线性损失(后面补充一个线性模型损失对比图)

3）结合核函数将数据映射到高维空间，使得模型具有非线性能力

4）具有感知机的一切解释性，同时目标函数的对偶形式是凸二次规划问题

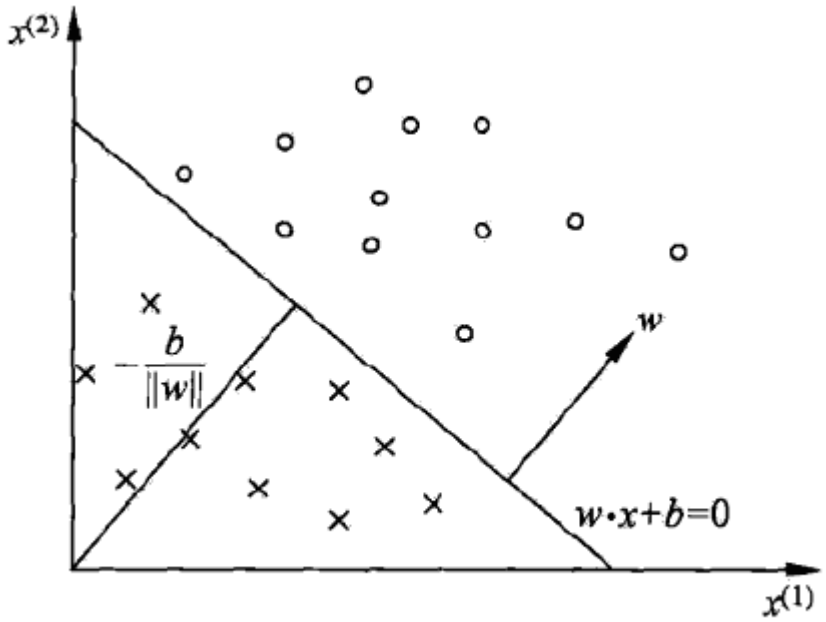


图 2.1　感知机模型

硬间隔（最大化最小间隔分类器）：

线性感知机中由于没有线性可分假设，所以其目标函数定义为最小化错分样本的损失，而硬间隔SVM则提出了一个线性可分假设，即样本在高维空间中线性可分，那么使得两类分开的超平面一定有无限个。硬间隔SVM则在这些超平面中找出最优的（即所有样本到超平面距离加和最小化），所以有如下目标函数：

$$\min \sum_{i=1}^{m} \frac{1}{||w||^2} y_i(w \cdot x_i + b)$$

其中$\frac{1}{||w||^2} y_i(w \cdot x_i + b)$为点到平面的几何间隔，去掉系数为函数间隔。最大化最小间隔分类器则采用等价形式—使得最难分的样本离超平面距离尽可能的大—最大化最小间隔分类器

$$\max_{w,b} \gamma$$

$$s.t. \frac{1}{||w||^2} y_i(w \cdot x_i + b) > \gamma, i \in 1, 2...m$$

$$\max_{w,b} \frac{\gamma}{||w||^2}$$

$$s.t. y_i(w \cdot x_i + b) > \gamma, i \in 1, 2...m$$

令$\gamma = 1$有：

$$\min_{w,b} \frac{1}{2}||w||^2$$
$$s.t.\, y_i(w \cdot x_i + b) - 1 > 0, i \in 1, 2...m$$

到此，上式为硬间隔分类器的原问题最终形式。上述问题可使用拉格朗日乘子法和对偶问题进行求解。

拉格朗日函数

$$\min_{w,b} \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i (y_i(w \cdot x_i + b) - 1)$$
$$s.t. \bigtriangledown L(w, b, \alpha_i) = 0$$
$$\alpha_i (y_i(w \cdot x_i + b) - 1) = 0$$
$$\alpha_i \geq 0$$
$$y_i(w \cdot x_i + b) - 1 > 0, i \in 1, 2...m$$

其中$\bigtriangledown L(w, b, \alpha_i) = 0$由Fritz John条件得出，$\alpha_i (y_i(w \cdot x_i + b) - 1) = 0$为互补松弛条件，互补松弛条件与支持向量有密切关系。由上述约束条件有：

$$\frac{\bigtriangledown L(w, b, \alpha_i)}{w} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0$$
$$\frac{\bigtriangledown L(w, b, \alpha_i)}{b} = \sum_{i=1}^{m} \alpha_i y_i = 0$$
$$b = y_j - \sum_{i=1}^{m} \alpha_i y_i x_i \cdot x_j$$

将上式带入到拉格朗日函数，得到关于α表示的函数：

$$L(w, b, \alpha) = -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_j y_j x_i x_j + \sum_{i=1}^{m} \alpha_i$$

最大化关于α的函数即为原问题的对偶问题,如下：

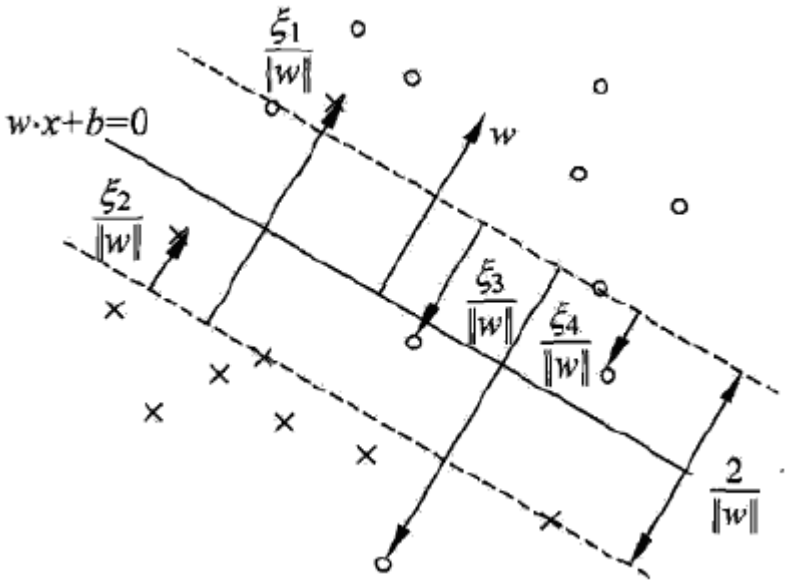
$$\max L(w, b, \alpha) = -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_j y_j x_i x_j + \sum_{i=1}^{m} \alpha_i$$
$$\Leftrightarrow \min \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_j y_j x_i x_j - \sum_{i=1}^{m} \alpha_i$$
$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0$$
$$\alpha_i \geq 0$$

解出上式目标函数α后，有w, b

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i$$
$$b = y_j - \sum_{i=1}^{m} \alpha_i y_i x_i \cdot x_j$$

其中可以看出，w和b有样本点与α内积确定。

但是回过头来想，线性可分假设是不现实，所以SVM在硬间隔线性可分的基础上提出软间隔线性可分。即允许线性不可分，但是需要进行一定的惩罚。如下图为软间隔线性可分，其中在支持向量里面的点和错分的样本为线性不可分的点，虚线上的点为支持向量。



软间隔SVM：

线性不可分意味着某些样本不满足函数间隔大于1的约束条件，为了解决这个问题，可以对**每个样本**引入一个松弛变量$\xi_i \geq 0$，使得函数间隔加上松弛变量大于等于1，这样约束条件变为：

$$y_i(w \cdot x_i + b) > 1 - \xi_i, i \in 1, 2..m$$

同时对于线性不可分的样本进行惩罚，因此目标函数变为：

$$min_{w,b} \frac{1}{2}||w|| + C\sum_{i=1}^{m}\xi_i$$

因此最终的线性不可分SVM的目标函数如下：

$$min_{w,b} \frac{1}{2}||w|| + C\sum_{i=1}^{m}\xi_i$$
$$s.t. yi(w \cdot xi + b) > 1 - \xi_i, i \in 1, 2..m$$
$$\xi_i \geq 0, i \in 1, 2..m$$

拉格朗日函数

$$min_{w,b}\frac{1}{2}||w||^2+C\sum_{i=1}^{m}\xi_i-\sum_{i=1}^{m}\alpha_i(y_i(w\cdot x_i+b)-1+\xi_i)-\sum_{i=1}^{m}\beta_i\xi_i$$
$$s.t.\ \triangledown L(w,b,\xi_i,\alpha_i,\beta_i)=0$$
$$\alpha_i(y_i(w\cdot x_i+b)-1+\xi_i)=0$$
$$\beta_i\xi_i=0$$
$$\alpha_i\geq 0$$
$$\beta_i\geq 0$$
$$y_i(w\cdot x_i+b)-1\geq 0, i\in 1,2...m$$
$$\xi_i\geq 0, i\in 1,2..m$$

由上述约束条件有：

$$\frac{\triangledown L(w,b,\alpha_i)}{w}=w-\sum_{i=1}^{m}\alpha_i y_i x_i=0$$
$$\frac{\triangledown L(w,b,\alpha_i)}{b}=\sum_{i=1}^{m}\alpha_i y_i=0$$
$$\frac{\triangledown L(w,b,\alpha_i)}{\xi_i}=C-\alpha_i-\beta_i=0$$
$$b=y_j-\sum_{i=1}^{m}\alpha_i y_i x_i\cdot x_j$$

将上式带入到拉格朗日函数，得到目标函数关于α，β表示的函数,同硬间隔的对偶函数一致：

$$L(w,b,\alpha)=-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_j y_j x_i x_j+\sum_{i=1}^{m}\alpha_i$$

最大化关于α的函数即为原问题的对偶问题，而对偶问题为原问题提供一个下界，即原问题的对偶问题如下：

$$\max L(w,b,\alpha)=-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_j y_j x_i x_j+\sum_{i=1}^{m}\alpha_i$$
$$\Leftrightarrow \min\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_j y_j x_i x_j-\sum_{i=1}^{m}\alpha_i$$
$$s.t.\sum_{i=1}^{m}\alpha_i y_i=0$$
$$C-\alpha_i-\beta_i=0$$
$$\alpha_i\geq 0$$
$$\beta_i\geq 0$$
$$\Leftrightarrow \min\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_j y_j x_i x_j-\sum_{i=1}^{m}\alpha_i$$
$$s.t.\sum_{i=1}^{m}\alpha_i y_i=0$$
$$0\leq\alpha_i\leq C$$

解出上式目标函数α,β后，有w,b

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i$$

$$b = y_j - \sum_{i=1}^{m} \alpha_i y_i x_i \cdot x_j$$

可以看出，w和b由样本点与α内积确定，当$\alpha_i = 0$表示 第i个样本点满足$y_i(w \cdot x_i + b) - 1 \geq 0$条件，该点不在支持向量内部，w与该点无关，支持向量机的参数w只与支持向量以内的点有关。

对比硬间隔和软间隔SVM发现两者的对偶问题非常相似，唯一不同的在于$0 \leq \alpha$，$0 \leq \alpha \leq C$，也就是说在约束条件上不能让α值太大。而α不为0的意义就是该点线性不可分—在支持向量以内，不能让α太大的意义就是尽可能的不要让样本在支持向量太里面。这也就是惩罚项引入后的结果。

下面根据α, β的取值来分析样本点的一个位置，以及样本点对SVM参数的影响：

当$\alpha_i = 0$,则$\beta_i = C$，$\xi_i = 0$，表示样本点在支持向量上或者以外的，以外的点对参数w无价值

当$0 < \alpha_i < C$,则$0 < \beta_i < C$，$\xi_i = 0$，表示样本点在支持向量上

当$\alpha_i = C$,则$0 = \beta_i$，如果$0 < \xi_i < 1$，表示样本在支持向量内部，但分类正确

当$\alpha_i = C$,则$0 = \beta_i$，如果$\xi_i = 1$，表示样本在超平面上

当$\alpha_i = C$,则$0 = \beta_i$，如果$\xi_i > 1$，表示样本分类错误

核函数：

核函数的应用主要是解决线性不可分问题，通过选择合适的核函数将样本从低维线性不可分映射到高维之后容易线性可分，本质上是一次空间上的非线性变换（特征映射），核函数可以嫁接到很多线性模型上，使其具有非线性能力，只是核函数的选择是一件难定的事。

而SVM与核函数有着天然的契合度，因为在SVM的对偶问题中，需要计算样本之间的内积，而核函数的引入则可以使得内积操作直接在核函数中隐式完成。

$$L(w, b, \alpha) = -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_j y_j x_i \cdot x_j + \sum_{i=1}^{m} \alpha_i$$

在上式中有$x_i \cdot x_j$内积操作，当我们使用核技巧时，往往需要定义一个核函数$\phi(x)$进行特征空间变换，然后在新的特征空间中进行$\phi(x_i) \cdot \phi(x_j)$内积操作，这使得计算过程分两步完成。如果我们隐式的定义核函数如下：

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

$$L(w, b, \alpha) = -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_j y_j K(x_i, x_j) + \sum_{i=1}^{m} \alpha_i$$

直接定义$K(x_i, x_j)$作为核函数，而不管实际的核函数$\phi(x)$是如何将x映射到$\phi(x)$空间，然后在新的特征空间计算内积。这样，我们就隐式完成了内积操作，将核函数与内积操作一步完成为$K(x_i, x_j)$。当然，核函数必须满足核函数的性质。

一般常采用的核函数有：

线性核 $K(x_i, x_j) = x_i^T x_j$

多项式核 $K(x_i, x_j) = (x_i^T x_j)^d$

高斯核$K(x_i,x_j)=exp(-\frac{(x_i-x_j)^2}{2\sigma^2})$

拉普拉斯核$K(x_i,x_j)=exp(-\frac{||x_i-x_j||}{2\sigma^2})$

sigmoid核$K(x_i,x_j)=tanh(\beta x_i^T x_j+\theta)$

然而核技巧中，最盲目的是如何选择合适核函数，或者多核。

这里需要解释的是，SVM对核函数有一个自身的要求，核的大小一定是m^2。因为SVM在做内积时是所有点彼此做内积，所以复杂度是m^2。这也是SVM难以适应大规模数据的场景，SVM的复杂度m^2d体现在内积上，带核的SVM的复杂度体现在核函数的计算上。而这不是核函数的特点，核函数中核的大小是自定义的。

SMO优化算法

$$\min \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_j y_j x_i x_j-\sum_{i=1}^{m}\alpha_i$$
$$s.t.\sum_{i=1}^{m}\alpha_i y_i=0$$
$$0\le\alpha_i\le C$$

SVM优化问题是一个典型的带约束凸二次规划，传统的梯度方法不能直接应用于带约束优化问题，下面先介绍一种坐标上升优化算法，算法的思想是对于多个参数的优化求解问题，可以每次只考虑一个变量，而固定其他所有变量，对一个变量进行目标优化，内循环每一个变量进行优化，外循环直到迭代到收敛。其收敛性类似于EM算法。

Loop until convergence: {

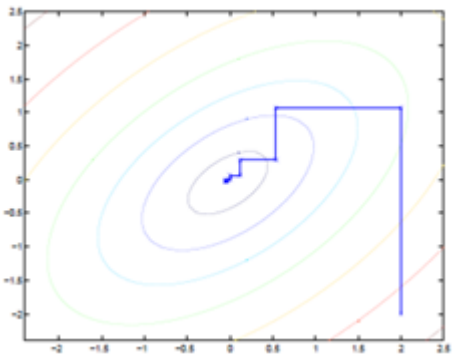
For $i=1,\dots,m$, {

$\alpha_i:=\arg\max_{\hat{\alpha}_i} W(\alpha_1,\dots,\alpha_{i-1},\hat{\alpha}_i,\alpha_{i+1},\dots,\alpha_m)$.

}

}

因为内层循环每次只改变一个变量，所以坐标上升算法的搜索路径与坐标轴平行



然而，如果每次只改变一个变量来优化SVM，那么必然不满足$\sum_{i=1}^{m}\alpha_i y_i=0$约束。所以SMO算法在坐标上升算法基础上又以下两点改进：

1）为了满足$\sum_{i=1}^{m}\alpha_i y_i=0$约束，每次迭代优化选择两个变量，其中一个主动变量，另一个被动变量

2）在选择两个变量进行优化时，采用启发式搜索策略，主动变量选择违反KKT条件最严重的一个变量α_1，在选定α_1后，被动变量α_2选择变化范围最大的，在优化α_1和α_2时使用上下剪辑来使得α_1和α_2满足$0 \leq \alpha_i \leq C$约束

现在来看SMO算法，固定m-2个变量不变，将目标函数转化为关于α_1和α_2的函数：

$$\min \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_j y_j K(x_i,x_j) - \sum_{i=1}^{m}\alpha_i$$

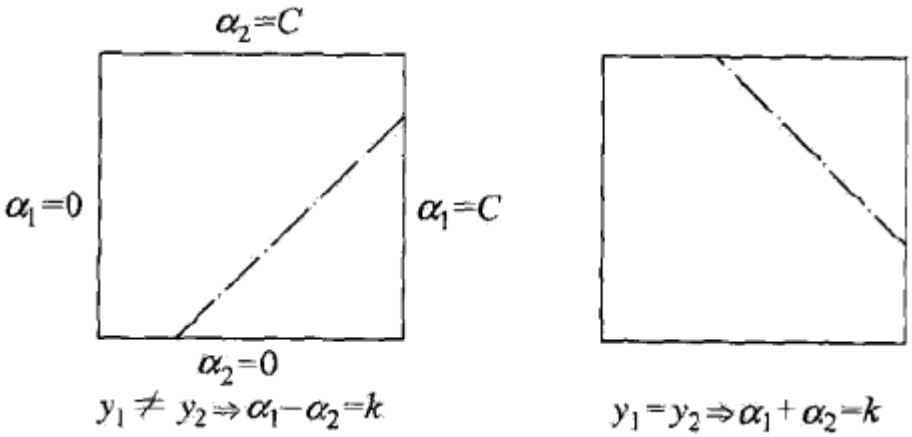
$$\min W(\alpha_1,\alpha_2) = \frac{1}{2}\alpha_1^2 K_{11} + \frac{1}{2}\alpha_2^2 K_{22} + y_1 y_2 \alpha_1 \alpha_2 K_{12} + y_1\alpha_1\sum_{i=3}^{m} y_i\alpha_i K_{i1} + y_2\alpha_2\sum_{i=3}^{m} y_i\alpha_i K_{i2} - (\alpha_1+\alpha_2)$$

$$s.t.\ \alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{m}\alpha_i y_i = \varsigma$$

$$0 \leq \alpha_i \leq C$$

其中$K_{ij} = K(x_i, x_j)$。

为了求解两个变量的二次规划问题，首先我们分析约束条件，可以看出α_1和α_2的可行域是盒子内的一条对角线上，其中盒子由不等式确定，对角线由等式确定，而且由于y_1和y_2的不确定性导致存在两种情况：



至于对角线的位置取决于当前α_1和α_2的值。由于优化过程中，我们首先优化的是α_2，而后由等式约束确定α_1，所以我们分析α_2的变化范围：

当$y_1 \neq y_2$时：$L = max(0, \alpha_2 - \alpha_1)$，$H = min(C, C + \alpha_2 - \alpha_1)$

当$y_1 = y_2$时：$L = max(0, \alpha_2 + \alpha_1 - C)$，$H = min(C, \alpha_2 + \alpha_1)$

其中L是为了保证α_2的变化不会让$\alpha_1 < 0$，H是为了保证α_2的变化不会让$\alpha_1 > C$。

同样，由于我们首先优化的是α_2，所以我们采用α_2来表示α_1：

$\alpha_1 = \frac{(\varsigma - \alpha_2 y_2)}{y_1}$，代入$\min W(\alpha_1,\alpha_2)$有（省略了推导步骤）：

$$W(\alpha_2) = a\alpha_2^2 + b\alpha_2 + c$$

求导后得到：

$$\frac{\nabla W(\alpha_2)}{\alpha_2} = \frac{y_2(((g(x_2) - y_2) - (g(x_1) - y_1)))}{(K_{11} + K_{22} - 2K_{12})}$$

记$E_i = g(x_i) - y_i$，$\eta = (K_{11} + K_{22} - 2K_{12})$有：

$$\frac{\triangledown W(\alpha_2)}{\alpha_2} = \frac{y_2(E_2 - E_1)}{\eta}$$

所以：

$$\alpha_2^{unew} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\eta}$$

回到上下剪辑，最终α_2的更新值为：

$$\alpha_2^{new} = \begin{cases} H, \alpha_2^{unew} > H \\ \alpha_2^{unew}, L \leq \alpha_2^{unew} \leq H \\ L, \alpha_2^{unew} < L \end{cases}$$

再由$\sum_{i=1}^{m} \alpha_i y_i = 0$得：

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new})$$

最后更新b，由KKT条件当$0 \leq \alpha_j \leq C$时，有$b = y_j - \sum_{i=1}^{m} \alpha_i y_i K_{ij}$

当$0 \leq \alpha_1 \leq C$时：

$$\begin{aligned} b^{new} &= y_1 - \sum_{i=1}^{m} \alpha_i y_i K_{i1} + \alpha_1^{old} y_1 K_{11} + \alpha_2^{old} y_2 K_{21} - \alpha_1^{new} y_1 K_{11} - \alpha_2^{new} y_2 K_{21} \\ &= -E_1 - y_1 K_{11}(\alpha_1^{new} - \alpha_1^{old}) - y_2 K_{21}(\alpha_2^{new} - \alpha_2^{old}) + b^{old} \end{aligned}$$

同样，当$0 \leq \alpha_2 \leq C$时：b由α_2来确定。

如果两者同时满足条件时,那么两者确定的b是一致的，如果等式取到的话，说明点在支持向量上或者以内，此时b取两者之间。

下面来看SMO的启发式搜索策略：

1）主动变量选择违反KKT条件最严重的点，即优先判断支持向量上的点是否满足KKT条件，其次检验整个训练样本是否满足KKT条件

由上面对α与样本点位置的分析可得到如下关系：

$$\begin{gathered} \alpha_i = 0 \Leftrightarrow y_i g_i \geq 1 \\ 0 \leq \alpha_i \leq C \Leftrightarrow y_i g_i = 1 \\ \alpha_i = C \Leftrightarrow y_i g_i \leq 1 \end{gathered}$$

由上面关系，可以知道哪些点在支持向量上，哪些点在支持向量外，哪些点在支持向量内，优先选择支持向量上的点来判断是否违反KKT条件，因为这些点是违反KKT条件最严重的点，也是对超平面最有价值的点。

2）被动变量选择在给定主动变量后，被动变量随之变化范围最大的点，由于前面导出$\alpha_2^{unew} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\eta}$所以被动变量选择依赖于$|E_1 - E_2|$的大小，选择最大的，加速计算速度。

3）值得注意的是，每次迭代更新α_1^{new}和α_2^{new}之后，需要更新E_1^{new}和E_2^{new}。

支持向量机回归

支持向量机回归利用的就是Hinge损失来定义目标函数，同样是线性模型$h_\theta(x)=\theta^Tx$，由Hinge损失定义如下目标函数：

$$\min_\theta C\sum_{i=1}^m L_\epsilon(h_\theta(x^i)-y^i)+\lambda||w||^2$$

其中$L_\epsilon(z)=\begin{cases}0,|z|\leq\epsilon\\|z|-\epsilon,other\end{cases}$，可以看出支持向量机回归其实就是借用Hinge损失，而其理论解释值得思考。

损失函数加正则项的一般理解

机器学习模型中，绝大多数的模型可以理解损失函数加正则项的形式，本文从线性到非线模型中提到的所有模型都可以理解为损失函数加正则项：

$$\arg\min_w L(w)+\lambda\Omega(w)$$
$$s.t.$$

其中正则项主要包括 0范数 1范数 2范数，损失函数主要包括以下

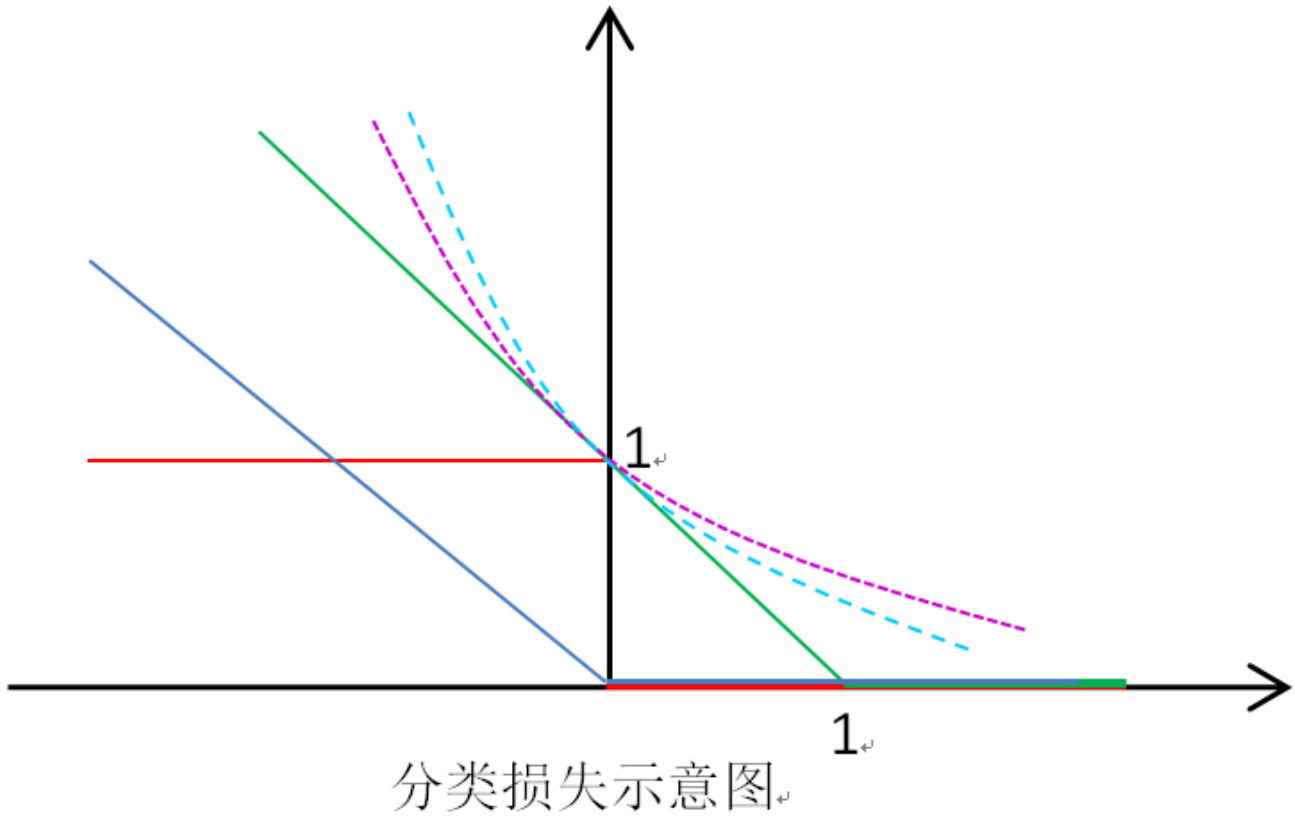
平方损失 $L(z)=(y-\theta^Tx)^2$ 线性回归

线性损失 $L(z)=y-\theta^Tx$ 线性感知机

对数损失 $L(z)=log(1+exp(-z))$ Logistic回归，softmax回归，

Hinge损失$L(z)=max(0,1-z)$，支持向量机

指数损失$L(z)=exp(-z)$，Adaboost



分类损失示意图

红：0-1损失，蓝：线性损失，绿：Hinge损失，紫虚：对数损失，青虚：指数损失

如何选择合适的损失函数加正则项是模型选择的一个依据，损失函数的选择依赖于数据的分布，而且不同的模型都有各自的特点，在选择模型时很难说那个模型优于其他模型，需要综合各方面因素选择。

总结：

矩阵运算补充

正则项-范数

拉格让日乘子法与对偶问题补充

拉格朗日乘子法通过引入松弛变量得到目标函数局部最优解的必要条件

拉格朗日乘子法的一般形式：

$$\min_x f(x)$$
$$s.t.\ g_i(x) \geq 0$$
$$h_j(x) = 0$$

引入松弛变量w, v也称拉格朗日乘子，朗格朗日函数如下：

$$\min L(x, w, v) = f(x) - w_i g_i(x) - v_j h_j(x)$$

如果$\bar{x}$是目标函数的局部最优解，那么$\bar{x}$的一阶必要条件如下：

$$\triangledown L(x, w, v) = 0$$
$$w_i g_i(x) = 0$$
$$w_i \geq 0$$
$$g_i(x) \geq 0$$
$$h_j(x) = 0$$

其中$w_i g_i(x) = 0$为互补松弛条件,梯度为0条件由Frizt John条件得到。

一般来讲，到拉格朗日乘子法之后我们还不能解出目标函数的局部最优解，因为目标函数还是一个引入松弛变量的带约束优化问题。不过我们可以通过分析拉格朗日函数的局部最优解来得到其对偶问题。

在给定x时，对$L(x, w, v)$求极大值时，当x不满足所有必要条件时，那么必然导致$L(x, w, v)$无最大值，当且仅当x满足所有必要条件时$L(x, w, v)$有极大值，且极大值为$f(x)$

$$L = \begin{cases} f(x), x\text{满足必要条件} \\ -\infty，\text{否则} \end{cases}$$

所以，所有约束条件的等价条件是$L(x, w, v)$存在极大值，所以原问题就变成了一个极小极大问题

$$\min_x L(x, w, v)(s.t.\ ) = \min_x \max_{w,v} L(x, w, v)$$

定义一个对偶问题，即定义一个用w, v变量来表示的目标函数：

$$\theta_D(w, v) = \min_x L(x, w, v)$$

最大化$\theta_D(w, v)$即为原问题的对偶问题，下面证明对偶问题为原问题提供下界：

$$\max_{x,v,w_i>0} \theta_D(w, v) = \max_{w,v,w_i>0} \min_x L(x, w, v)$$

又因为：

$$\max_{x,v,w_i>0} \min_x L(x, w, v) \leq \min_x \max_{w,v} L(x, w, v)$$

所以对偶问题为原问题提供下界。