支持向量机（SVM）

标签（空格分隔）： 监督学习

@author : duanxxnj@163.com
@time : 2016-07-31

在看这篇文章之间，建议先看一下感知机，这样可以更好的看懂SVM。

一、最大间隔分类器

在之前的文章感知机中提到过，感知机模型对应于特征空间中将实例划分为正负两类的分离超平面，但是，感知机的解却不止一个。对同一个数据集而言，可以计算得到很多的感知机模型，不同的感知机的 **训练误差** 都是一样的，都为0。

那么，这些训练误差都为0的感知机模型中，如何针对当前数据集，选择一个最好的感知机模型呢？现在就需要考虑模型的 **泛化误差** 了。即，在所有训练误差为0的感知机中，选择泛化误差最小的那个感知机，这就

是SVM算法最初的目的。

基本的SVM（最大间隔分类器）是一种二分类模型，它是定义在特征空间上的间隔最大的线性分类器，间隔最大是SVM和感知机不同的地方，间隔最大化对应于泛化误差最小。

1.1 决策面

面对一个线性可分的二分类问题，将正负样本分开的超平面，称为决策面。

和 线性回归 模型一样，这里一般会使用一些特征函数$\phi(x)$ ϕ(x)，将输入空间映射到新的特征空间中，再进行计算。

$$y(x) = f(w^T \phi(x) + b)$$

y(x)=f(wTϕ(x)+b)

这里$f(\cdot)$ f(⋅) 叫做激活函数，ww是线性模型的系数，bb一般被叫做偏置：

$$f(a) = \begin{cases} +1 & a \geq 0 \\ -1 & a < 0 \end{cases}$$

f(a)={+1a≥0−1a<0

这里输出的取值为 $t \in \{+1, -1\}$t∈{+1,−1} ，即正负样本。这里的 $\{+1, -1\}$ {+1,−1} 仅仅是一个标号，代表正负样本，并不是具体的数值。如果感觉不喜欢$\{+1, -1\}$ {+1,−1}，可以和Logistics Regression一样，使用$\{0, 1\}$ {0,1}也行。而且，可以使用$\{+1, -1\}$ {+1,−1}主要也是因为这里是二分类问题，遇到多分类问题的时候，还得考虑其他的标号方式。

> 如果感觉$\phi(x)$ ϕ(x)这种表述方式不太习惯，可以考虑所有的$\phi(x) = x$ ϕ(x)=x，这样就和一般的书上的公式一致了。这种表达仅仅是我个人的喜好，式主要是为了强调，在实际问题中，输入空间xx一般不会作为模型的输入，而是要将输入空间通过一定的特征转换算法$\phi(x)$ ϕ(x)，转换到特征空间，最后在特征空间中做算法学习。而且，在实际问题中，各种算法基本上都是死的，但是，特征变换的这个过程$\phi(x)$ ϕ(x)却是活的，很多时候，决定一个实际问题能不能很好的解决，$\phi(x)$ ϕ(x)起着决定性的作用。举个简单的例子，比如Logistics Regression，数据最好要做归一化，如果数据不归一化，那么那些方差特别大的特征就会成为主特征，影响模型的计算，$\phi(x)$ ϕ(x)就可以做这个事情。再或者后面要降到的核函数问题，核函数SVM能解决非线性可分问题，主要也是基于使用 $\phi(x)$ ϕ(x) 来做非线性特征变换。

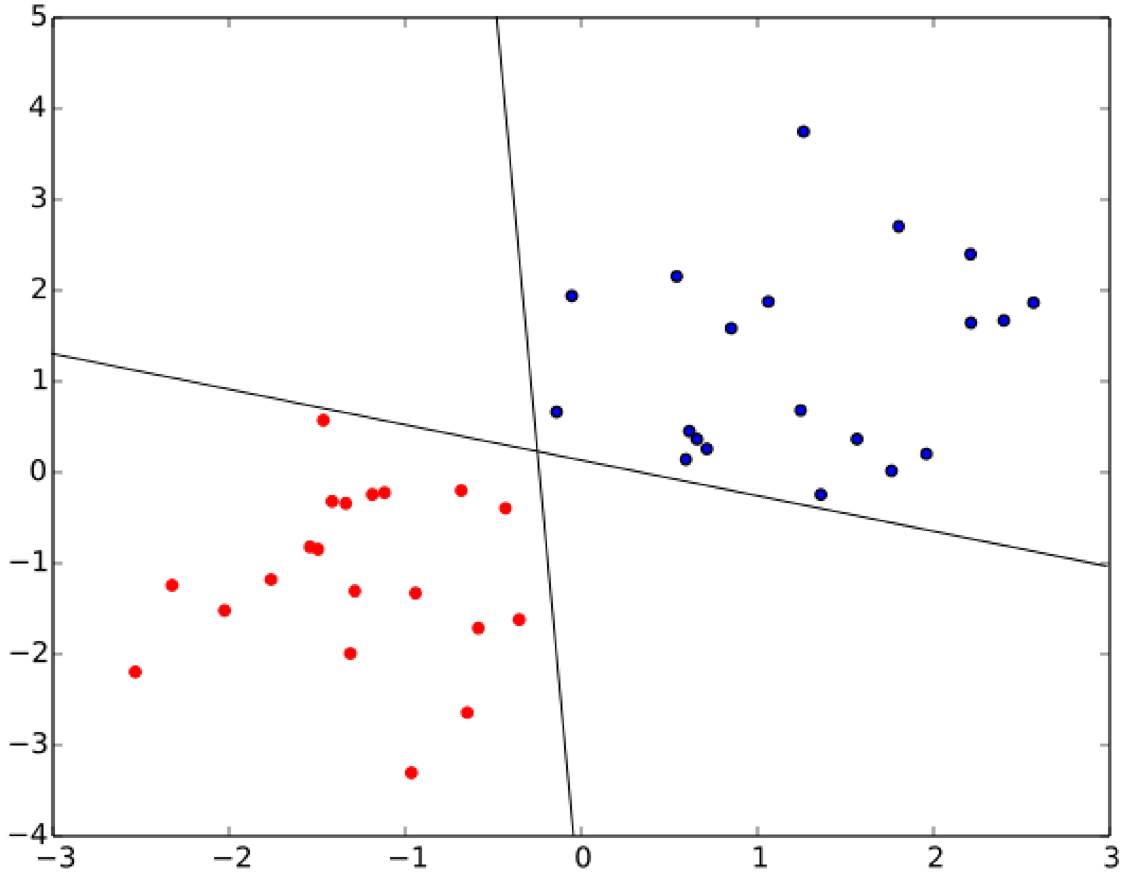
作为一个决策面：
当样本的标号 $t_n = +1$ tn=+1 的时候，该样本为正样本。如果样本被正确分类，那么$w^T \phi(x_n) + b > 0$wTϕ(xn)+b>0，$f(w^T \phi(x) + b) = +1$ f(wTϕ(x)+b)=+1
当样本的标号 $t_n = -1$ tn=−1 的时候，该样本为负样本。如果样本被正确分类，那么$w^T \phi(x_n) + b < 0$wTϕ(xn)+b<0，$f(w^T \phi(x) + b) = -1$ f(wTϕ(x)+b)=−1

> **究竟什么是决策面？
> 答：决策面就是能够将正负样本分开的点的集合，比如上面的模型中，决策面的数学表达式为：$w^T \phi(x) + b = 0$wTϕ(x)+b=0，决策面就是这个式子的解集，所以，一般我们就直接用这个式子代表决策面了。可以看到，对于这个决策面的数学表达式而言，ww 和 bb 的数值并不重要。比如，$w^T \phi(x) + b = 0$wTϕ(x)+b=0 和

> $2w^T\phi(x) + 2b = 0$ 2wTϕ(x)+2b=0，后一个决策面的参数数值是前一个决策面数值的两倍，但这两个决策面的解是一样的，也就是说其得到的点集是一样的，那么这两个表达式所表示的就是同一个决策面。所以，这里我要强调一点，对于一个线性决策面而言，重要的不是ww 和 bb 的取值，重要的是 ww 和 bb 的**比值**， ww 和 bb 的**比值**决定了一个决策面的点集，也就决定了一个决策面。**

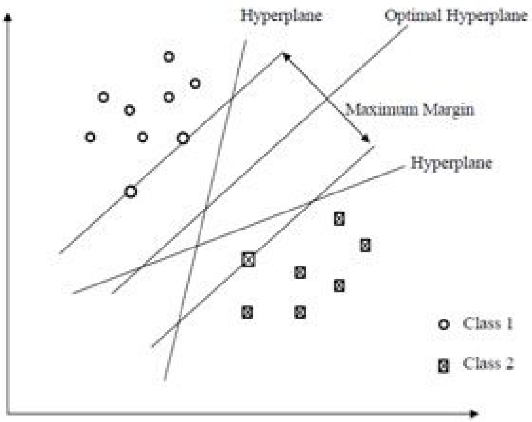
1.2 最优决策面

对于线性可分的二分类问题而言，使用 感知机 算法，可以得到很多很多满足上述要求的决策面，比如下图中，就是可以将正负两类数据分开的两个决策面。



那么，在这些决策面中，哪个决策面，才是最优的决策面呢？首先，需要明确，上述的决策面，都是可以将训练样本正确分类的决策面，也就是说，上述决策面的 **训练误差** 都为0。要从这些训练误差都为0的模型中，选出一个最好的模型，很自然的，就需要考虑模型的 **泛化误差** 。

最大间隔分类器认为，决策面的泛化误差可以用训练样本集合中，离决策面最近的样本点到决策面的间隔（margin）来表示，离决策面最近的样本点，到决策面的间隔（margin）越大，那么，这个决策面的泛化误差就越小。直观的来讲，最优决策面差不多就是下面这幅图中，中间的那个决策面。



1.3 最小间隔

什么是间隔?
答：首先，要搞清楚是谁和谁的间隔，在这里指的是一个 **训练样本点** 和 **决策面** 之间的间隔。

那么，间隔又如何定义的呢?
答：间隔，就是样本点到决策面之间的距离，由中学的知识就可以知道，一个样本点 x_n 到一个面 $w^T\phi(x)+b=0$，的距离为（这里可以直接认为是点到直线的距离）：

$$r_n=\frac{w^T\phi(x_n)+b}{||w||}$$

但这个距离在数值上会存在正负的问题，由于样本点类别取值 $t_n\in\{+1,-1\}$，所以样本点到决策面的间隔可以改为：

$$r_n=\frac{t_n\{w^T\phi(x_n)+b\}}{||w||}$$

当 $t_n=+1$的时候，如果样本被正确分类， $w^T\phi(x_n)+b>0$，上述样本点到决策面的间隔 r_n 取正值
当 $t_n=-1$的时候，如果样本被正确分类， $w^T\phi(x_n)+b<0$，上述样本点到决策面的间隔 r_n 仍然取正值
这样，分类正确的样本点的间隔就永远是正的了。

显然，一旦决策面有了，那么训练集合中的每个样本点 x_n 到决策面都会有一间隔 r_n 。自此，就可以定义样本集合到决策面最小的间隔 r 为：

$$r=\min_n\frac{t_n\{w^T\phi(x_n)+b\}}{||w||};n\in\{1,2,...,N\}$$

既然 r 是最小间隔，毫无疑问，对于任意一个样本点 x_n 而言：

$$\frac{t_n\{w^T\phi(x_n)+b\}}{||w||}\ge r$$

tn{wTϕ(xn)+b}||w||≥r

1.4 最小间隔最大化

前面已经说了，要使用决策面在训练样本中的最小间隔 r 来表示决策面的训练误差：最小间隔 r 越大，那么其泛化误差就越小，模型就越好。而我们这里，就是在所有可选的决策中，找出其对应的最小间隔 r 最大的那个决策面，而决策面是用参数 w 和 b 定义的，所以，最小间隔最大化可以形式化为：

$$\arg\max_{w,b}\{\min_n \frac{t_n\{w^T\phi(x_n)+b\}}{||w||}\}$$

argmaxw,b{minntn{wTϕ(xn)+b}||w||}

用优化理论的形式重写一下上面的式子，可以得到：

$$\max_{w,b} r$$
$$st：\frac{t_i\{w^T\phi(x_i)+b\}}{||w||}\ge r \quad ;i=1,2,\ldots N$$

maxw,brst：ti{wTϕ(xi)+b}||w||≥r ;i=1,2,...N

将上面的约束条件变一下型可得：

$$\max_{w,b} r$$
$$st：t_i\{w^T\phi(x_i)+b\}\ge r||w|| \quad ;i=1,2,\ldots N$$

maxw,brst：ti{wTϕ(xi)+b}≥r||w|| ;i=1,2,...N

在前面已经提到过，对于一个决策面 $w^T\phi(x_i)+b=0$ 而言， 重要的不是 w 和 b 的取值，$w^T\phi(x)+b=0$ 和 $2w^T\phi(x)+2b=0$ 的所得到的 x 的点集是一样的，即其决策面是一样的，这里真正重要的是 w 和 b 的**比值。**

w 和 b 的**比值**，决定了一个决策面的点集，也就决定了一个决策面。

所以，只要有一个决策面，那么，唯一确定的是 w 和 b 的**比值**，但是，w 和 b 具体的取值是可以改变的，只要 w 和 b 按比例改变，决策面就是确定不变的。

也就是说，w 和 b 的**比值**比值不变的情况下，w 是可以任意取值的。这样的话，为了便于计算，我们就取：

$$||w||=\frac{1}{r}$$

||w||=1r

那么，上面的式子又可以改写为：

$$\max_{w,b}\frac{1}{||w||}$$

$$st：t_i\{w^T\phi(x_i)+b\}\geq 1;i=1,2,\ldots N$$

maxw,b1||w||st：ti{wTϕ(xi)+b}≥1;i=1,2,...N

可以看到，上面的式子最大化的目标函数变成了 $1/||w||$1/||w||，很容易知道，其等价于 最小化 $||w||^2$||w||2，那么最小间隔最大化，最终就可以变为下面这个最小化的约束优化问题：

$$\min_{w,b}||w||^2$$

$$st：t_i\{w^T\phi(x_i)+b\}\geq 1;i=1,2,\ldots N$$

minw,b||w||2st：ti{wTϕ(xi)+b}≥1;i=1,2,...N

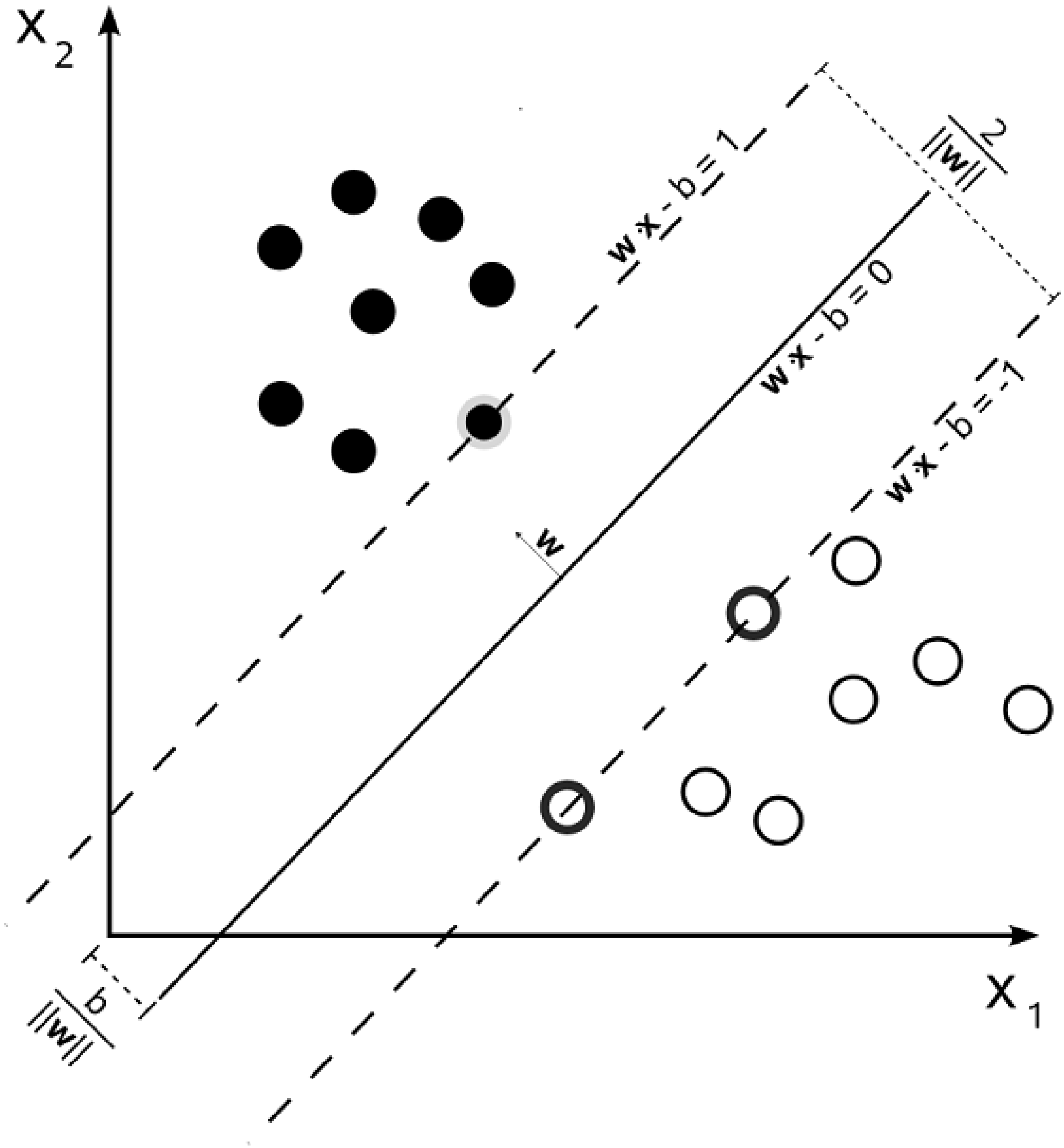
这里，最小间隔为：

$$r=\frac{1}{||w||}$$

r=1||w||

> 在网上博客还有现有的书籍中，对于最小间隔最大化的解释，都使用了函数间隔和几何间隔的概念，我个人在第一次接触这两个概念的时候，就有些被弄糊涂了，理解了这两个概念之后，又感觉这种解释过于冗余、牵强，完全是为了解释最小间隔最大化而解释最小间隔最大化。所以，这里我直接抛弃函数间隔和几何间隔的概念，给出我个人的一种解释。

最小间隔最大化后得到的决策面，就是我们要找的泛化误差最小的决策面。对于下图中两特征的二分类问题而言，可以看出， 最小间隔为$1/||w||$1/||w||，即，正样本到决策面的最小间隔为$1/||w||$1/||w||；同时，负样本到决策面的最小间隔也为$1/||w||$1/||w||。所以正负样本之间的最小间隔为$2/||w||$2/||w||。



由于间隔最小的样本点 x_n xn 满足:

$$\frac{t_n\{w^T\phi(x_n)+b\}}{||w||}=r=\frac{1}{||w||}$$

tn{wTϕ(xn)+b}||w||=r=1||w||

所以，间隔最小的正样本点 x_n xn 满足:

$$w^T\phi(x_n)+b=1$$

wTϕ(xn)+b=1

间隔最小的负样本点 x_n xn 满足:

$$w^T\phi(x_n)+b=-1$$

wTϕ(xn)+b=−1

这里定义，拥有最小间隔的正负样本点为支持向量（support vector），也就是上图中 $w^T\phi(x_n)+b=1$ wTϕ(xn)+b=1 和 $w^T\phi(x_n)+b=-1$ wTϕ(xn)+b=−1 所对应的那三个样本点。

> 关于支持向量机的名字，这里可以稍微说一下，因为这个名字并不是特别的好理解。首先是支持（support），根据柯林斯词典，support的主要意思：the activity of providing for or maintaining by supplying with money or necessities，表示的是提供一些必须品的意思。vector指的就是样本点，这个问题不大。那么问题来了，为什么将间隔最小的那些样本点叫做support vector（或者可以直接说是support point）呢？答：从图中可以看出，对于决策面而言，要想唯一的确定这个决策面，就是要根据最小的间隔 r 来得到，也就是说，决策面仅仅和拥有最小间隔的那些样本点相关，和其他那些间隔大于最小间隔的样本点，是没有关系的。即：拥有最小间隔的那些样本点是决策面所必须的，而间隔大于最小间隔的样本点的有无，对决策面并不构成影响。所以将拥有最小间隔的那些样本点叫做support point， 也就是support vector。

1.5 拉格朗日对偶性

在求解约束最优化问题的过程中，我们常常会使用拉格朗日对偶性（Lagrange duality），把原始问题（primal problem）转换为对偶问题（dual problem）来求解，基于对偶问题的求解来得到原始问题的解。这个方法在统计学中经常使用，不仅仅本文的SVM算法用到了拉格朗日对偶性，后面要讲的最大熵模型也用到了拉格朗日对偶性。这里仅仅对拉格朗日对偶性做一个简述，我个人认为，主要知道其概念和结果即可，无需深究。拉格朗日对偶性的详细说明，可以参见Boyd的《Convex Optimization》，该书用一整个章节的篇幅来详细的论述了拉格朗日对偶性的问题。

> 对于一个线性规划问题，我们称之为原始问题，都有一个与之对应的线性规划问题我们称之为对偶问题。原始问题与对偶问题的解是对应的，得出一个问题的解，另一个问题的解也就得到了。并且原始问题与对偶问题在形式上存在很简单的对应关系：
> - 目标函数对原始问题是极大化，对偶问题则是极小化
> - 原始问题目标函数中的系数，是对偶问题约束不等式中的右端常数，而原始问题约束不等式中的右端常数，则是对偶问题中目标函数的系数
> - 原始问题和对偶问题的约束不等式的符号方向相反
> - 原始问题约束不等式系数矩阵转置后，即为对偶问题的约束不等式的系数矩阵
> - 原始问题的约束方程数对应于对偶问题的变量数，而原始问题的变量数对应于对偶问题的约束方程数
> - 对偶问题的对偶问题是原始问题

1、原始问题

假设，有$f(x), c_i(x), h_j(x)$，他们是定义在空间R^n上的连续可微的函数，也就是可导函数的意思。其约束最优化问题为：

$$
\begin{aligned}
&\min_{x\in R^n} f(x)\\
s.t.\quad & c_i(x) \le 0; i = 1,2,\dots,k\\
& h_j(x) = 0; j = 1,2,\dots,l
\end{aligned}
$$

minx∈Rnf(x)s.t. ci(x)≤0;i=1,2,...,khj(x)=0;j=1,2,...,l

这里$c_i(x)$是不等式优化，而$h_j(x)$是等式优化。

上面这种约束优化问题的形式成为原始问题。

现在，引入拉格朗日函数：

$$L(x,\alpha,\beta) = f(x) + \underbrace{\sum_{i=1}^{k} \alpha_i c_i(x)}_{\text{不等式约束}} + \underbrace{\sum_{j=1}^{l} \beta_j h_j(x)}_{\text{等式约束}}$$

L(x,α,β)=f(x)+∑i=1kαici(x)不等式约束+∑j=1lβjhj(x)等式约束

这里 α_i 和 β_j 是拉格朗日乘子，且，$\alpha_i \geq 0$。也就是说，不等式约束 $c_i(x)$ 的拉格朗日乘子要大于等于0，而等式约束 $h_j(x)$ 的拉格朗日乘子并没有限制。

现在考虑最大化这个拉格朗日函数，定义：

$$\theta_p(x) = \max_{\alpha,\beta;\alpha_i \geq 0} L(x,\alpha,\beta) = \max_{\alpha,\beta;\alpha_i \geq 0} \{f(x) + \sum_{i=1}^{k} \alpha_i c_i(x) + \sum_{j=1}^{l} \beta_j h_j(x)\}$$

θp(x)=maxα,β;αi≥0L(x,α,β)=maxα,β;αi≥0{f(x)+∑i=1kαici(x)+∑j=1lβjhj(x)}

对于上面的最大化式子有：

- 如果不等式约束 $c_i(x) < 0$ ，等式约束 $h_j(x) = 0$ ，那么上式最大化就会使得 $\alpha_i = 0$，β_j 可以取任意值。 此时所有满足约束条件的，并且最大化结果为 $\theta_p(x) = f(x)$。
- 如果不等式约束 $c_i(x) = 0$ ，等式约束 $h_j(x) = 0$，那么上式最大化就会使得 $\alpha_i > 0$,β_j 可以取任意值。此时是满足约束条件的，并且最大化结果为 $\theta_p(x) = f(x)$。
- 如果存在不等式约束 $c_i(x) > 0$, 等式约束 $h_j(x) = 0$，即此时存在不等式约束违反约束条件，那么要使得上式最大化，必然会使得 $\alpha_i = \infty$ ，进而使得 $\theta_p(x) = \infty$。
- 如果不等式约束 $c_i(x) \leq 0$, 存在 $h_j(x) \neq 0$ ，那么要使得上式最大化，必然会使得 $\beta_j = \infty$ ，进而使得 $\theta_p(x) = \infty$。

综上所述，可以知道：

$$\theta_p(x) = \max_{\alpha,\beta;\alpha_i \geq 0} L(x,\alpha,\beta) = \begin{cases} f(x) & c_i(x) \leq 0 \text{ 和 } h_j(x) = 0 \text{ 都满足时} \\ \infty & \text{当存在违反} c_i(x) \leq 0 \text{ 或者} h_j(x) = 0 \text{ 条件时} \end{cases}$$

θp(x)=maxα,β;αi≥0L(x,α,β)={f(x)ci(x)≤0 和 hj(x)=0 都满足时∞当存在违反ci(x)≤0 或者hj(x)=0 条件时

由于原始最优化问题就是要在满足约束条件下，求解最小化的 $f(x)$。而由上可知，在满足约束条件的情况下，$f(x) = \theta_p(x) = \max_{\alpha,\beta;\alpha_i \geq 0} L(x,\alpha,\beta)$。

也就是说，原始约束最优化问题：

$$\begin{aligned} &\min_{x \in R^n} f(x) \\ s.t. \quad & c_i(x) \leq 0; i = 1,2,\ldots,k \\ & h_j(x) = 0; j = 1,2,\ldots,l \end{aligned}$$

minx∈Rnf(x)s.t. ci(x)≤0;i=1,2,...,khj(x)=0;j=1,2,...,l

中的 $f(x)，c_i(x) \le 0，h_j(x) = 0$ f(x) ， ci(x)≤0 ， hj(x)=0 就可以用 $\theta_p(x) = \max_{\alpha,\beta;\alpha_i \ge 0} L(x,\alpha,\beta)$ θp(x)=maxα,β;αi≥0L(x,α,β) 来代替。

于是乎，原始约束最优化问题就变成了一个极小极大化的问题：

$$\min_x \theta_p(x) = \min_x \max_{\alpha,\beta;\alpha_i \ge 0} L(x,\alpha,\beta)$$

minxθp(x)=minxmaxα,β;αi≥0L(x,α,β)

很显然，这个极小极大化问题，和原始问题是等价的。

为了方便，这里定义原始问题（primal problem）的最优解为：

$$p^* = \min_x \theta_p(x)$$

p∗=minxθp(x)

2、对偶问题

> 对于一块磁铁而言，磁铁有N极和S极，N极和S极只是同一块磁铁的不同表现而已，这两个极性虽然不同，但是却拥有相同的本质：磁。他们是相辅相成的，是同一个事物的两种不同表现。就像有阴必有阳；有光明必有黑暗；而阴阳本为一体，明暗实为一物，他们都是同一个东西的不同表现而已，这个是世间万物的规律。

对偶问题和原始问题也是一样，他们是优化问题的两个不同表现形式而已，他们本质上是一个东西，只是表现的方式相反罢了。

考虑原始问题：

$$\theta_p(x) = \max_{\alpha,\beta;\alpha_i \ge 0} L(x,\alpha,\beta)$$

θp(x)=maxα,β;αi≥0L(x,α,β)

原始问题是以 α,β α,β 为参数，以 x x 为变量的极大化问题。既然对偶问题是原始问题关于优化问题的相反的表达方式，那么对偶问题就可以写成，以 α,β α,β 为变量，以 x x 为参数的极小化问题(记住一点：本质相同，表现相反)：

$$\theta_D(\alpha,\beta) = \min_x L(x,\alpha,\beta)$$

θD(α,β)=minxL(x,α,β)

原始问题最终是被极小化，成为了一个极小极大化的问题：

$$\min_x \theta_p(x) = \min_x \max_{\alpha,\beta;\alpha_i \ge 0} L(x,\alpha,\beta)$$

minxθp(x)=minxmaxα,β;αi≥0L(x,α,β)

对偶问题，要和其相反，就需要被极大化，而称为一个极大极小化的问题：

$$\max_{\alpha,\beta;\alpha_i \ge 0} \theta_D(\alpha,\beta) = \max_{\alpha,\beta;\alpha_i \ge 0} \min_x L(x,\alpha,\beta)$$

maxα,β;αi≥0θD(α,β)=maxα,β;αi≥0minxL(x,α,β)

上面的式子，就称为拉个朗日函数的极大极小问题，并定义对偶问题（dual problem）的最优解为：

$$d^* = \max_{\alpha,\beta;\alpha_i \geq 0} \theta_D(\alpha,\beta)$$

d∗=maxα,β;αi≥0θD(α,β)

由于：

$$\theta_D(\alpha,\beta) = \min_x L(x,\alpha,\beta) \leq L(x,\alpha,\beta) \leq \max_{\alpha,\beta;\alpha_i \geq 0} L(x,\alpha,\beta) \leq \theta_p(x)$$

θD(α,β)=minxL(x,α,β)≤L(x,α,β)≤maxα,β;αi≥0L(x,α,β)≤θp(x)

所以：

$$\theta_D(\alpha,\beta) \leq \theta_p(x)$$

θD(α,β)≤θp(x)

上面这是式子说明，$\theta_D(\alpha,\beta)$θD(α,β)的所有的解，都不大于$\theta_p(x$）θp(x）的解。那么，毫无疑问，对偶问题的最优解和原始问题的最优解也满足这个式子：

$$d^* = \max_{\alpha,\beta;\alpha_i \geq 0} \theta_D(\alpha,\beta) \leq \min_x \theta_p(x) = p^*$$

d∗=maxα,β;αi≥0θD(α,β)≤minxθp(x)=p∗

在我们常见的问题中，只要满足一定的条件，就可以使得$d^* = p^* = L(x^*,\alpha^*,\beta^*)$ d∗=p∗=L(x∗,α∗,β∗)，这里 x^*,α^*,β^* x∗,α∗,β∗ 就是最优解。这里所说的一定的条件，指的就是KKT条件。

3、KKT条件

对于原始问题 和 对偶问题而言，x^*,α^*,β^* x∗,α∗,β∗ 分别是原始问题和对偶问题的解的充分必要条件是 x^*,α^*,β^* x∗,α∗,β∗ 满足KKT条件：

$$\begin{aligned}
\nabla_x L(x^*,\alpha^*,\beta^*) &= 0 \\
\nabla_\alpha L(x^*,\alpha^*,\beta^*) &= 0 \\
\nabla_\beta L(x^*,\alpha^*,\beta^*) &= 0 \\
\alpha_i^* c_i(x^*) &= 0; i = 1,2...k \\
c_i(x^*) &\leq 0; i = 1,2...k \\
\alpha_i^* &\geq 0; i = 1,2...k \\
h_j(x^*) &= 0; j = 1,2,\ldots l
\end{aligned}$$

∇xL(x∗,α∗,β∗)=0∇αL(x∗,α∗,β∗)=0∇βL(x∗,α∗,β∗)=0αi∗ci(x∗)=0;i=1,2...kci(x∗)≤0;i=1,2...kαi∗≥0;i=1,2...khj(x∗)=0;

上述的KKT条件，看起来很吓人，其实很容易理解：函数 $L(x,\alpha,\beta)$ L(x,α,β) 是以 x,α,β x,α,β 为参数的，那么其最优解 x^*,α^*,β^* x∗,α∗,β∗ 定然满足 函数 $L(x,\alpha,\beta)$ L(x,α,β) 的梯度为0，这就是KKT条件的前三个等式：

$$\begin{aligned}
\nabla_x L(x^*,\alpha^*,\beta^*) &= 0 \\
\nabla_\alpha L(x^*,\alpha^*,\beta^*) &= 0 \\
\nabla_\beta L(x^*,\alpha^*,\beta^*) &= 0
\end{aligned}$$

∇xL(x∗,α∗,β∗)=0∇αL(x∗,α∗,β∗)=0∇βL(x∗,α∗,β∗)=0

前面已经说明过，函数 $L(x,\alpha,\beta)$ L(x,α,β) 要能够使用，最初的优化问题必须满足不等式约束 $c_i(x) \le 0$ ci(x)≤0 以及其相关的拉格朗日乘子 α_i αi 的约束：

$$\begin{aligned} \alpha_i^* c_i(x^*) &= 0; i = 1,2 \ldots k \\ c_i(x^*) &\le 0; i = 1,2 \ldots k \\ \alpha_i^* &\ge 0; i = 1,2 \ldots k \end{aligned}$$

αi∗ci(x∗)=0;i=1,2...kci(x∗)≤0;i=1,2...kαi∗≥0;i=1,2...k

而最后一个KKT条件，对应的就是 $L(x,\alpha,\beta)$ L(x,α,β) 的等式约束：

$$h_j(x^*) = 0; j = 1,2,\ldots l$$

hj(x∗)=0;j=1,2,...l

1.6 最小间隔最大化求解

求解最小间隔最大化，就是要求解式子：

$$\begin{aligned} &\min_{w,b} \frac{1}{2}||w||^2 \\ s.t.\quad &t_i\{w^T\phi(x_i)+b\} \ge 1; i=1,2,\ldots N \end{aligned}$$

minw,b12||w||2s.t. ti{wTϕ(xi)+b}≥1;i=1,2,...N

而求解上面的式子，用的到方法，就是拉格朗日对偶性。这里，在原来的最小化的目标函数前面加了 $1/2$ 1/2 ,并不会影响最后的最优解，但是对后面的公式推导相对有利，故而加上了个 $1/2$ 1/2。

将它作为原始的优化问题，应用拉格朗日对偶性，通过求解对偶问题（dual problem）来得到原始问题（primal problem）的最优解，这个最优解，就对应于最优的决策面。

这样做的优点主要有两个：
一、对偶问题相对来说比较容易求解
二、可以很自然的引入核函数，进而推广到非线性分类器中

首先，构建拉格朗日函数，由于上面的约束优化问题中只有不等式约束，所以为所有的不等式约束添加拉格朗日乘子： $\alpha_i \ge 0; i=1,2,\ldots N$ αi≥0;i=1,2,...N，则拉格朗日函数为：

$$L(w,b,\alpha) = \underbrace{\frac{1}{2}||w||^2}_{\text{优化目标}} - \underbrace{\sum_{i=1}^{N}\alpha_i(t_i\{w^T\phi(x_i)+b\}-1)}_{\text{不等式约束}}$$

L(w,b,α)=12||w||2⏟优化目标−∑i=1Nαi(ti{wTϕ(xi)+b}−1)⏟不等式约束

根据前面关于拉格朗日对偶性的说明可以很容易知道，这个原始问题为极小极大问题：

$$\min_{w,b}\max_{\alpha;\alpha_i\ge 0} L(w,b,\alpha)$$

minw,bmaxα;αi≥0L(w,b,α)

其对应的对偶问题为极大极小问题：

$$\max_{\alpha;\alpha_i\geq 0}\min_{w,b} L(w,b,\alpha)$$

maxα;αi≥0minw,bL(w,b,α)

求解内部极小化

这里首先求解内部的极小化问题：

$$\min_{w,b} L(w,b,\alpha)$$

minw,bL(w,b,α)

显然，这个极小化问题是以 w,b 为参数的，那么，先使 $L(w,b,\alpha)$ 对 w,b 求导，并令其为0：

$$\begin{aligned}\nabla_w L(x,\alpha,\beta) &= w-\sum_{i=1}^{N}\alpha_i t_i \phi(x_i)=0\\ \nabla_b L(x,\alpha,\beta) &= \sum_{i=1}^{N}\alpha_i t_i=0\end{aligned}$$

∇wL(x,α,β)=w−∑i=1Nαitiϕ(xi)=0∇bL(x,α,β)=∑i=1Nαiti=0

那么，就有：

$$\begin{aligned}w &= \sum_{i=1}^{N}\alpha_i t_i \phi(x_i)\\ \sum_{i=1}^{N}\alpha_i t_i &= 0\end{aligned}$$

w=∑i=1Nαitiϕ(xi)∑i=1Nαiti=0

将 $w=\sum_{i=1}^{N}\alpha_i t_i \phi(x_i)$ 带入到 $L(w,b,\alpha)$ 中，就可以得到：

$$\begin{aligned}L &= \frac{1}{2}||w||^2-\sum_{i=1}^{N}\alpha_i(t_i\{w^T\phi(x_i)+b\}-1)\\ &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\{\alpha_i\alpha_j t_i t_j <\phi(x_i),\phi(x_j)>\}-\sum_{i=1}^{N}\alpha_i(t_i\{\{\sum_{j=1}^{N}\alpha_j t_j\phi(x_j)\}\phi(x_i)+b\}-1)\\ &= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\{\alpha_i\alpha_j t_i t_j <\phi(x_i),\phi(x_j)>\}+b\sum_{i=1}^{N}\alpha_i t_i+\sum_{i=i}^{N}\alpha_i\\ &= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\{\alpha_i\alpha_j t_i t_j <\phi(x_i),\phi(x_j)>\}+\sum_{i=i}^{N}\alpha_i\end{aligned}$$

L=12||w||2−∑i=1Nαi(ti{wTϕ(xi)+b}−1)=12∑i=1N∑j=1N{αiαjtitj<ϕ(xi),ϕ(xj)>}−∑i=1Nαi(ti{{∑j=1Nαjtjϕ(xj)}ϕ(xi)+b}−1)=−12∑i=1N∑j=1N{αiαjtitj<ϕ(xi),ϕ(xj)>}+b∑i=1Nαiti+∑i=iNαi=−12∑i=1N∑j=1N{αiαjtitj<ϕ(xi),ϕ(xj)>}+∑i=iNαi

上面推导的导数第二步使用了：$\sum_{i=1}^{N} \alpha_i t_i = 0$∑i=1Nαiti=0，最终可以得到：

$$\min_{w,b} L(w,b,\alpha) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\{\alpha_i\alpha_j t_i t_j < \phi(x_i),\phi(x_j) >\} + \sum_{i=i}^{N}\alpha_i$$

minw,bL(w,b,α)=−12∑i=1N∑j=1N{αiαjtitj<ϕ(xi),ϕ(xj)>}+∑i=iNαi

这里的$< \phi(x_i),\phi(x_j) >$<ϕ(xi),ϕ(xj)> 是 $\phi(x_i)$ϕ(xi) 和 $\phi(x_j)$ϕ(xj) 的内积。

求解外部极大化

前面已经将内部的极小化求解得到了$\min_{w,b} L(w,b,\alpha)$minw,bL(w,b,α)，这里再在其求解的结果上加上外层的极大化，那么就有下面这个约束优化问题：

$$\max_{\alpha}\{-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\{\alpha_i\alpha_j t_i t_j < \phi(x_i),\phi(x_j) >\} + \sum_{i=i}^{N}\alpha_i\}$$

maxα{−12∑i=1N∑j=1N{αiαjtitj<ϕ(xi),ϕ(xj)>}+∑i=iNαi}

$$\begin{aligned} s.t. \quad \sum_{i=1}^{N}\alpha_i t_i &= 0 \\ \alpha_i &\geq 0; i = 1,2,\ldots N \end{aligned}$$

s.t. ∑i=1Nαiti=0αi≥0;i=1,2,...N

这个就是原问题的对偶问题，当然了，可以将这个对偶问题的目标函数的符号换一下，让它成为一个最小化的问题:

$$\min_{\alpha}\{\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\{\alpha_i\alpha_j t_i t_j < \phi(x_i),\phi(x_j) >\} - \sum_{i=i}^{N}\alpha_i\}$$

minα{12∑i=1N∑j=1N{αiαjtitj<ϕ(xi),ϕ(xj)>}−∑i=iNαi}

$$\begin{aligned} s.t. \quad \sum_{i=1}^{N}\alpha_i t_i &= 0 \\ \alpha_i &\geq 0; i = 1,2,\ldots N \end{aligned}$$

s.t. ∑i=1Nαiti=0αi≥0;i=1,2,...N

公式推导到这个地方，就可以知道，上面这个最小化问题，就是我们最终要求解的问题，其最小化的目标函数是以 αα 为参数的。

也就是说，假设最优解是 $\alpha^* = （\alpha_1^*,\alpha_2^*,\ldots\alpha_N^*）$ α∗=（α1∗,α2∗,...αN∗）。这里暂时不讨论如何求得这个最优解，其具体的求解算法会在后面详细的论述，现在假设，我们可以通过某种算法，将上面的这个最小化的优化问题的最优解求出来。

那么，在已知这个最优解的情况下，我们来看一下，基于这个最优解，SVM的决策面是什么样的?

根据原始问题$L(w,b,\alpha)$L(w,b,α)的KKT条件可以知道：

$$\begin{aligned}
\nabla_w L(x^*, \alpha^*, \beta^*) &= w^* - \sum_{i=1}^{N} \alpha_i^* t_i \phi(x_i) = 0 \\
\nabla_b L(x^*, \alpha^*, \beta^*) &= \sum_{i=1}^{N} \alpha_i^* t_i = 0 \\
\alpha_i^* (t_i \{ w^{*T} \phi(x_i) + b^* \} - 1) &= 0; i = 1, 2, \ldots N \\
t_i \{ w^{*T} \phi(x_i) + b^* \} - 1 &\geq 0; i = 1, 2, \ldots N \\
\alpha_i^* &\geq 0; i = 1, 2, \ldots N
\end{aligned}$$

∇wL(x∗,α∗,β∗)=w∗−∑i=1Nαi∗tiϕ(xi)=0∇bL(x∗,α∗,β∗)=∑i=1Nαi∗ti=0αi∗(ti{w∗Tϕ(xi)+b∗}−1)=0;i=1,2,...Nti{w∗Tϕ(xi)+b∗}−1≥0;i=1,2,...Nαi∗≥0;i=1,2,...N

那么，就有：

$$w^* = \sum_{i=1}^{N} \alpha_i^* t_i \phi(x_i)$$

w∗=∑i=1Nαi∗tiϕ(xi)

这里，根据已经求得的 α_i^* ,就可以将 w^* 求出来了。决策面的参数有两个: w, b，w 求出来了，剩下的就是 b 了。

在前面讨论过，SVM是间隔最大化的决策面，支持向量对应的就是间隔最小的那些点，由前面关于间隔最大化的讨论可以知道，支持向量满足下面这个公式：

$$t_i \{ w^T \phi(x_i) + b \} - 1 = 0$$

ti{wTϕ(xi)+b}−1=0

而根据前面对原始问题的讨论，可以知道，满足这个公式的点 x_i （支持向量） ,在拉格朗日函数中，所对应的 $\alpha_i^* > 0$ 。所以，我们只需要找到一个$\alpha_i^* > 0$, 就可以得到 b^*:

$$b^* = \frac{1}{t_i} - w^{*T} \phi(x_i)$$

b∗=1ti−w∗Tϕ(xi)

同时，需要注意：$t_i^2 = 1$，并带入 w^{*T}，就可以将上面这个式子重写为：

$$b^* = t_i - \sum_{j=1}^{N} \alpha_j^* t_j < \phi(x_j), \phi(x_i) >$$

b∗=ti−∑j=1Nαj∗tj<ϕ(xj),ϕ(xi)>

当然，为了稳妥起见，很多时候，我们会将所有的支持向量 $x_i \in S$ 对应的 b_i^* 都求出来，然后用其均值，作为最终的 b^*。这里 S 是支持向量的集合，也就是 $\alpha_i^* > 0$ 所对应的点集:

$$b^* = \frac{1}{N_s} \sum_{i=1}^{N_s} b_i^*$$

b∗=1Ns∑i=1Nsbi∗

这样，就可以求得最终的决策超平面为：

$$\sum_{i=1}^{N}\{\alpha_i^* t_i < \phi(x_i), \phi(x) >\} + b^* = 0$$

∑i=1N{αi∗ti<ϕ(xi),ϕ(x)>}+b∗=0

分类决策函数可以写为：

$$f(x) = \text{sign}\{\sum_{i=1}^{N}\{\alpha_i^* t_i < \phi(x_i), \phi(x) >\} + b^*\}$$

f(x)=sign{∑i=1N{αi∗ti<ϕ(xi),ϕ(x)>}+b∗}

这里就可以发现:

- 在预测的时候，w^* 和 b^* 仅仅依赖于训练集合中 $\alpha_i^* > 0$ 的那些样本点，而其他样本点对 w^* 和 b^* 没有影响。但是，为了求得 w^* 和 b^* ，在训练阶段，还是需要整个样本集合。也就是说，在做预测的时候，支持向量机需要的内存空间是非常小的，只需要存储支持向量即可，预测过程和非支持向量无关。
- 分类决策函数仅仅依赖于输入x 和 训练样本之间的内积。这个内积是后面核函数的雏形，也是SVM得以广泛应用的关键。

1.7 SVM、LDA、Logistics Regression 算法比较

在之前的文章线性判别分析（Linear Discriminant Analysis）中就说过：凡是分类算法，必定有决策面，而这些分类算法所不同的是:决策面是线性的还是非线性的；以及如果得到这个决策面。

对于Logistics Regression

对于一个二分类问题，在Logistics Regression中，假设后验概率为Logistics 分布：

$$P(C_1|x) = \frac{1}{1+\exp(w^T x)}$$
$$P(C_2|x) == \frac{\exp(w^T x)}{1+\exp(w^T x)}$$

P(C1|x)=11+exp(wTx)P(C2|x)==exp(wTx)1+exp(wTx)

这里，使用一个单调的变换函数，logit 函数：$\log[p/(1-p)]$，那么就可以得到：

$$\log\frac{P(C_1|x)}{P(C_2|x)} = w^T x$$

logP(C1|x)P(C2|x)=wTx

所以Logistics Regression的决策面就是：

$$w^T x = 0$$

wTx=0

对于Linear Discriminant Analysis

这里假设$f_k(x)$是类别C_k的类条件概率密度函数，π_k是类别C_k的先验概率，毫无疑问有$\sum_k \pi_k = 1$。根据贝叶斯理论有：

$$P(C_k|x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}$$

P(Ck|x)=fk(x)πk∑l=1Kfl(x)πl

LDA假设$f_k(x)$是均值不同，方差相同的高斯分布，所以其类条件概率密度函数可以写为：

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)\right)$$

fk(x)=1(2π)p/2|Σ|1/2exp(−12(x−μk)TΣ−1(x−μk))

这里，特征x的维度为p维，类别C_k的均值为μ_k，所有类别的方差为Σ。

LDA 和 前 面 提 到 的 Logistics Regression 采 用 的 单 调 变 换 函 数 一 样 ， 都 是 logit 函 数 ：$\log[p/(1-p)]$，对于二分类问题有：

$$\begin{aligned}\log\frac{P(C_1|x)}{P(C_2|x)} &= \log\frac{f_1(x)}{f_2(x)} + \log\frac{\pi_1}{\pi_2} \\ &= x^T\Sigma^{-1}(\mu_1-\mu_2) - \frac{1}{2}(\mu_1+\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2) + \log\frac{\pi_1}{\pi_2}\end{aligned}$$

logP(C1|x)P(C2|x)=logf1(x)f2(x)+logπ1π2=xTΣ−1(μ1−μ2)−12(μ1+μ2)TΣ−1(μ1−μ2)+logπ1π2

所以LDA的决策面就是：

$$x^T\Sigma^{-1}(\mu_1-\mu_2) - \frac{1}{2}(\mu_1+\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2) + \log\frac{\pi_1}{\pi_2} = 0$$

xTΣ−1(μ1−μ2)−12(μ1+μ2)TΣ−1(μ1−μ2)+logπ1π2=0

显然，在LDA中，和Logistics Regression不同的是，LDA的决策面是由各个数据分布的方差和类别中心决定的。

对于SVM

对于SVM而言，我么假设，最大间隔会带来最小的泛化误差。但是，反过来思考，这个假设真的是真确的吗？在讨论LDA和Logistics Regression的时候，我们其实也是说他们的决策面是最优的。那么，到底那个算法得到的决策面才是最好的呢？

说到这个问题，让我突然想起来前段时间的一件事情，一同学面试，面试官问他：Logistics Regression对数据分布有什么要求？他并不知道怎么回答，后来回来和他们实验室的人讨论了许久，也没有结果。其实这个回答非常的简单：

我们所面对的所有的机器学算法，都是有适用范围的，或者说，我们所有的机器学习算法都是有约束的优化问题。而这些约束，就是我们在推导算法之前所做的假设。

比如：Logistics Regression，上面已经明确说明了，在Logistics Regression中，假设后验概率为Logistics分布；再比如：LDA假设$f_k(x)$ fk(x)是均值不同，方差相同的高斯分布；这些都是我们在推导算法之前所做的假设，也就是算法对数据分布的要求。

而对于SVM而言，它并没有对原始数据的分布做任何的假设，这就是SVM和LDA、Logistics Regression区别最大的地方。这表明SVM模型对数据分布的要求低，那么其适用性自然就会更广一些。如果我们事先对数据的分布没有任何的先验信息，即，不知道是什么分布，那么SVM无疑是比较好的选择。

但是，如果我们已经知道数据满足或者近似满足高斯分布，那么选择LDA得到的结果就会更准确。如果我们已经知道数据满足或者近似满足Logistics 分布，那么选择Logistics Regression就会有更好的效果。

> 通过这三个方法的比较，我只想说明一件事情：机器学习算法是死的，但，人是活的。使用什么机器学习算法，是根据实际问题要求，和数据的具体分布而定的。