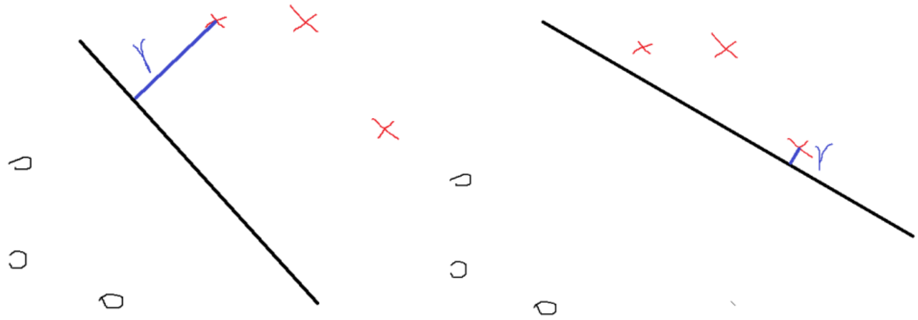
一、 线性可分支持向量机:

顾名思义，线性可分就是存在至少一个超平面，可以将样本中的正负例准确隔开，找到这样一个超平面可以用感知机实现，但是一般这样的超平面是不唯一的，线性可分支持向量机就是在这些超平面集合中选择最优的一个。

那么最优超平面的是怎么定义的呢？在感知机相关知识中，我们知道点到超平面的距离为 $\hat{r}=|\omega \bullet x_i + b|$（https://blog.csdn.net/wzx479/article/details/83143280），因为对所有的实例点来说 ω 都是一样的，所以就省略了，对于所有正确分类的点可以用这种形式取代绝对值号：$\hat{r}=y_i(\omega \bullet x_i + b)$。对于一个确定的超平面π，定义这个超平面关于所有实例点的函数间隔为 $\hat{r}_\pi = min\ \hat{r}$，最优的超平面就是使得 $\hat{r}_\pi$ 最大的那一个，这个比较容易理解：超平面与最近的点之间的距离越大，分类时的不确定性越小，如下两个超平面，都满足感知机的条件，但是直观上来看，第一个要比第二个好一些（为什么呢？我的看法是第一个超平面划分比较靠近中间，留给两侧的“缓冲区”是差不多的，类似于均值不等式会在相等的时候获得最值，有点中庸的意思）。



所以优化的目标就是 $max\ \hat{r}_\pi$，为了排除 ω 和 b 成比例变化带来的影响（例如同时变为两倍，这时超平面实际没变，但 r 变为两倍），定义了几何间隔：$r_\pi = \frac{\hat{r}}{||\omega||}$，因为我们关注的是 ω，所以把 $\hat{r}$ 设为 1，这时优化目标就变成 $max\frac{1}{||\omega||}$，等价于 $min\frac{1}{2}||\omega||^2$，因为定义了 $\hat{r}_\pi = min\ \hat{r} => r_\pi = min\ r$，也就是说 r_π 是所有实例点到超平面的距离最小的，所以约束条件是 $y_i(\omega \bullet x_i + b) \geq r_\pi = 1$，综上，可以得到以下原始优化问题 O1:

$$\begin{aligned} &min\frac{1}{2}||\omega||^2 \\ &s.t.\ y_i(\omega \bullet x_i + b) \geq 1 \end{aligned} \qquad O1$$

利用 Lagrange 对偶性（**https://www.cnblogs.com/breezezz/p/11303722.html**），可以将

原始问题转化为最大化最小值问题：

$$\max_{\alpha} \min_{\omega,b} L(\omega,b,\alpha)$$

其中，$L(\omega,b,\alpha)=\frac{1}{2}||\omega||^2-\sum_{i=1}^{n}\alpha_i y_i(\omega x_i+b)+\sum_{i=1}^{n}\alpha_i(1)$，有点像高数多元函数的有约束极值问题，首先要求 $\min L(\omega,b,\alpha)$，优化变量为 ω 和 b，令 L 偏导数为 0：

$$\frac{\partial L}{\partial \omega}=\omega-\sum_{i=1}^{n}\alpha_i y_i x_i$$
$$\frac{\partial L}{\partial b}=-\sum_{i=1}^{n}\alpha_i y_i$$

分别令其等于零，可以得到：

$$\omega=\sum_{i=1}^{n}\alpha_i y_i x_i$$
$$\sum_{i=1}^{n}\alpha_i y_i=0$$

这个时候解出的 ω 和 b 一定使 L 在α固定的条件下取得极值，至于是极大还是极小，可以通过求二阶偏导数判断，这里偷个懒直接按照书上的，即取的是极小值，将 ω 带入（1）式，并利用 $\sum_{i=1}^{n}\alpha_i y_i=0$ 条件，整理可得：

$$L(\omega,b,\alpha)\min=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(x_i\bullet x_j)+\sum_{i=1}^{n}\alpha_i$$

接下来的任务就是最大化 $L(\omega,b,\alpha)\min$，即：

$$\max(-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(x_i\bullet x_j)+\sum_{i=1}^{n}\alpha_i),$$

优化变量是α，所以与α相关的约束条件要考虑在内：$\sum_{i=1}^{n}\alpha_i y_i=0$，$\alpha_i\ge 0$（$\alpha_i$ 应该是可以小于等于 0 的，不过这样解出来的是-α_i，为了避免正负号转换的麻烦，取 $\alpha_i\ge 0$），整理一下，最原始的优化问题经过一步步变换，转换为以下问题：

$$\min_{\alpha}(\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(x_i\bullet x_j)-\sum_{i=1}^{n}\alpha_i)$$
$$s.t.\sum_{i=1}^{n}\alpha_i y_i=0$$
$$\alpha_i\ge 0$$

这样就把原来的三变量优化问题变为单变量优化问题（但是 α 是个向量，所以相当于还是多变量，呵呵），求解出 α 就可以根据 $\omega=\sum_{i=1}^{n}\alpha_i y_i x_i$ 和 $b=y_i-\omega\bullet x_i$，计算 b 时，只需要带入一对实例点即可，因为对于所有的实例点来说计算出的 b 都是相等的（我猜的）

李航的书里边给了一个例子，可以帮助理解一下上面的过程。正例点 x1(3,3)，x2(4,3)，

负例点 x3(1,1)，求解 ω 和 b，把这些点带到优化目标里：

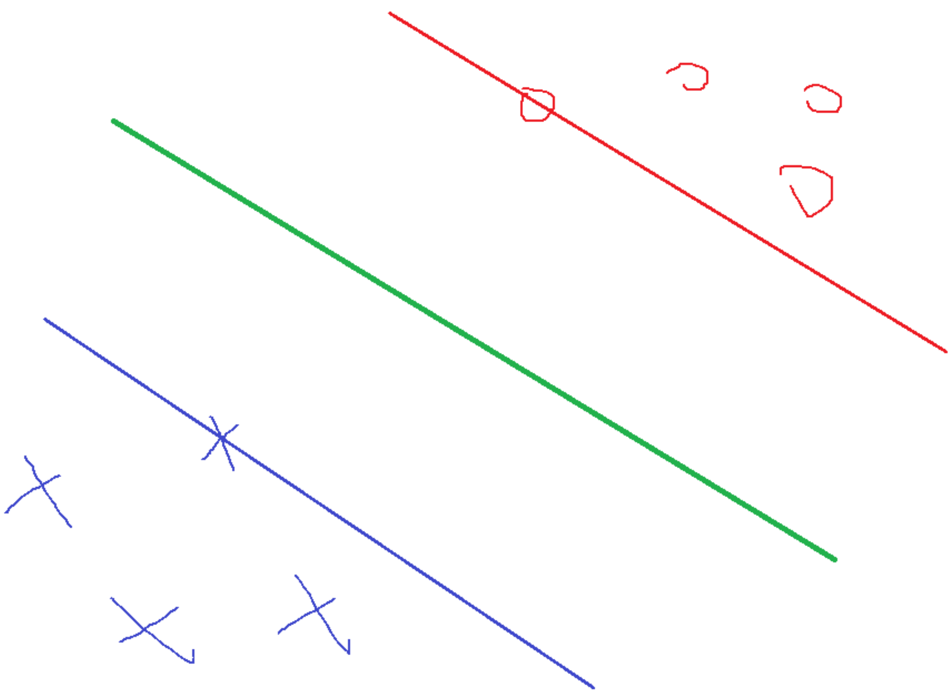
$$\min_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\alpha_i\alpha_j y_i y_j (x_i \bullet x_j) - \sum_{i=1}^{3}\alpha_i\right)$$
$$s.t.\ldots\ldots$$

可以得到：

$$\min_{\alpha}\left[\frac{1}{2}(18\alpha_1^2+25\alpha_2^2-2\alpha_3^2+42\alpha_1\alpha_2-12\alpha_1\alpha_3-14\alpha_2\alpha_3)-\alpha_1-\alpha_2-\alpha_3\right]$$

因为这个时候知道具体的表达式而且比较简单，可以直接求偏导数，令其等于 0，而不用数值方法，求解出来 $\alpha_1=\frac{3}{2};\alpha_2=-1;\alpha_3=\frac{1}{2}$，但是这个解不满足 $\alpha_i \ge 0$，所以正确的解在边界上，分别令 $\alpha_1,\alpha_2,\alpha_3=0$，然后每次求解出其余两个α，把这三组解带到 L 里，看看哪一个使得 L 最小，最终解为 $\alpha_1=\frac{1}{4};\alpha_2=0;\alpha_3=\frac{1}{4}$。在实际的算法中，绝大多数情况肯定是没法用解析的方法求解的。

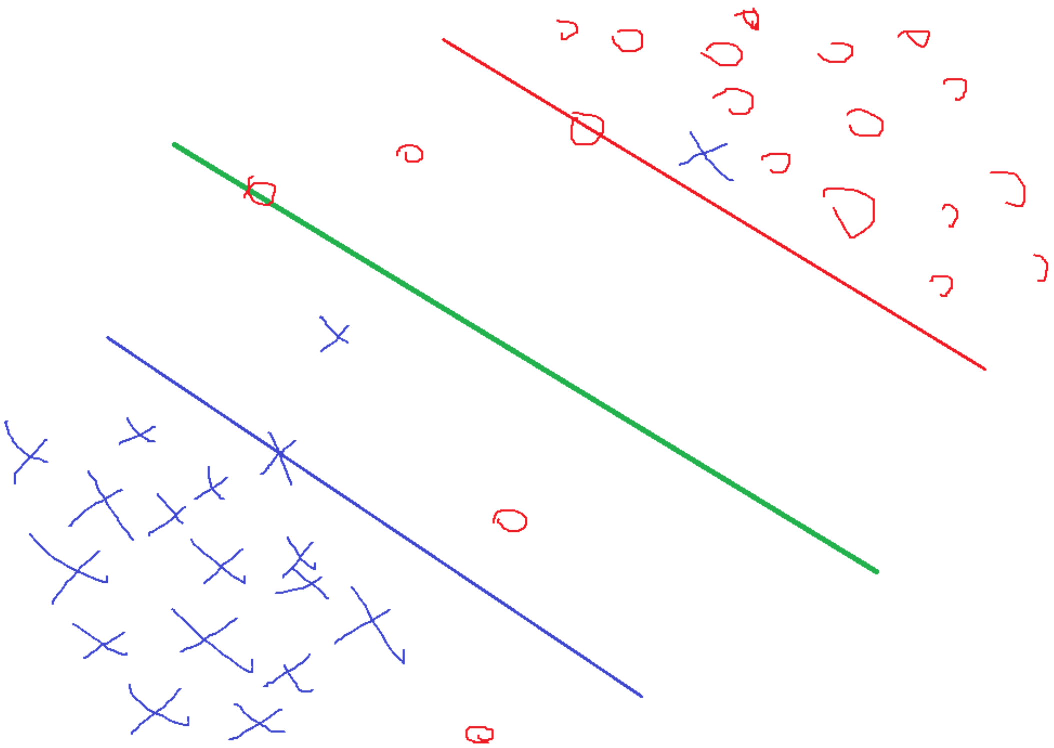
在线性可分的情况下，求解的结果应该如下图所示，其中绿线表示求出的超平面，红线和蓝线表示间隔平面。



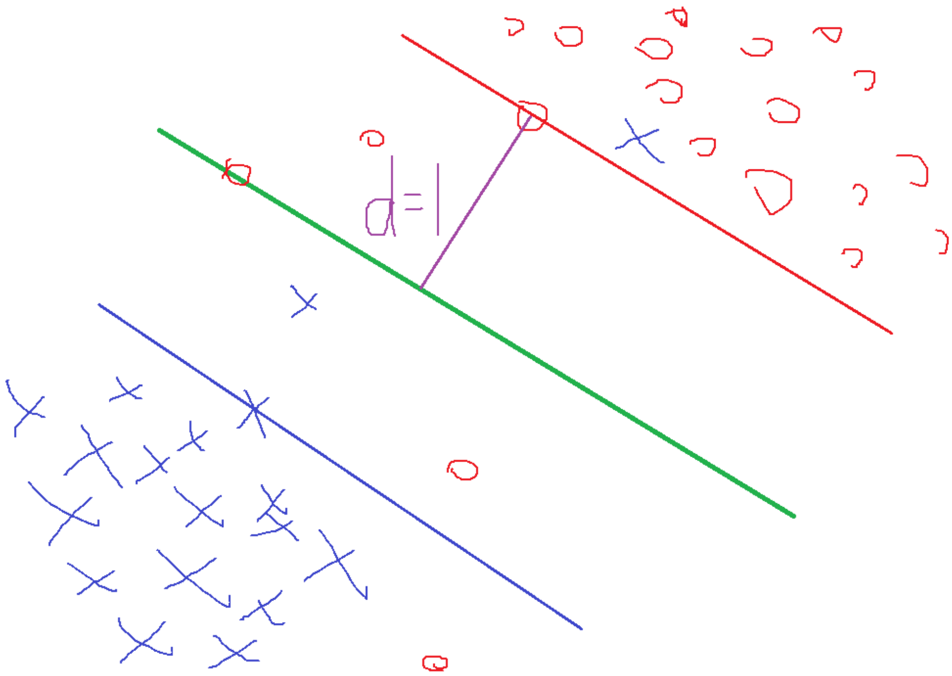
二、 软间隔最大化

在前一节的所有论述中，一个重要前提是超平面存在，即训练空间是线性可分的，实际应用中大部分情况肯定不是线性可分的，如果按照线性可分的要求去找超平面，在下图这种

情况显然是没有解的，但是，可以看到，按照绿线的划分方式，训练出来的模型应该也有不错的分类效果。

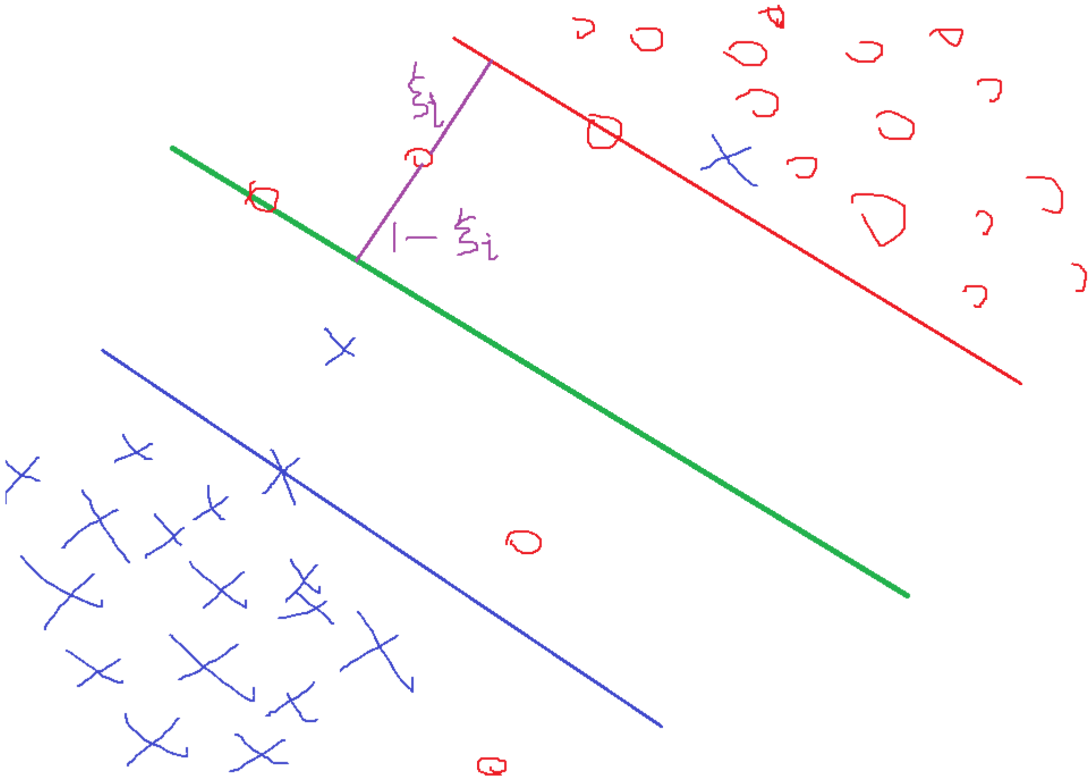


在原先的约束条件中 $y_i(\omega \cdot x_i + b) \geq 1$，这个 1 表示的就是下图中紫色笔画的 d，当然从原始距离变为 1，是经过了一些规范化的步骤的。



如果我们做出一点让步，允许某几个点（异常点）到超平面的距离小于间隔平面到超平面的距离（甚至可以为负值，负值表示点在超平面的另一侧，例如绿线以上的蓝×），这样就

可以使在不是线性可分的训练空间上也可以解出绿线这种模型，并且预期的分类效果还不错。那么需要修改原先的约束条件，改为：$y_i(\omega \bullet x_i + b) \geq 1\text{-}\zeta_i$，对于位于绿线上的红圈，只需要取$\zeta_i$=1（因为$y_i(\omega \bullet x_i + b)=0$就表示实例点在超平面上），对于在绿线上侧的蓝×，只需要取ζ_i>1，这时$y_i(\omega \bullet x_i + b) < 0$就表示蓝×落到了超平面的另一侧。



由此可以看出，加上ζ_i这个变量之后，可以通过调节ζ_i使所有的实例点都满足约束条件。这时优化问题变为 O2：

$$
\begin{aligned}
&\min \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{N}\xi_i \\
&s.t. \, y_i(\omega \bullet x_i + b) \geq 1 - \xi_i \\
&\xi_i \geq 0
\end{aligned}
\qquad \text{O2}
$$

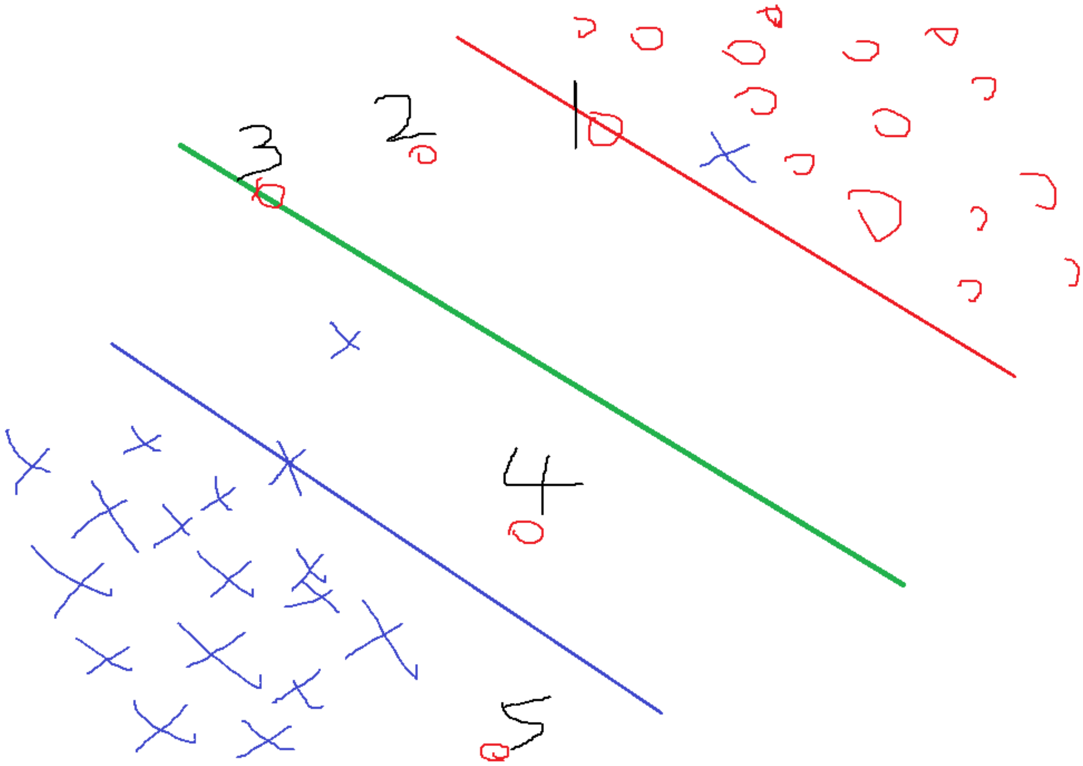
按照与求解 O1 类似的步骤（利用 Lagrange 对偶性，转换为最大化最小值问题之后偏导数，令梯度等于零，解出ω关于α_i的表达式），将尚书右约束最优化问题转换为：

$$
\begin{aligned}
&\min \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \bullet x_j) - \sum_{i=1}^{N}\alpha_i \\
&s.t. \sum_{i=1}^{N}\alpha_i y_i = 0 \\
&0 \leq \alpha_i \leq C
\end{aligned}
$$

利用某种方法（后面会提到，不是梯度下降法）进行优化求解之后可以得到

$$\omega = \sum_{i=1}^{N} \alpha_i y_i x_i$$
$$b = y_i - \omega \bullet x_i$$

这个解的形式与线性可分数据集的硬间隔最大化得到的解是一样的，不同的是，在软间隔最大化中，带入不同的点求解出的 b 可能是不同的，这些 b 都算是解，注意 b 只能由支持向量 x_i 所对应的实例点(x_i , y_i)求得，支持向量 x_i 是指位于间隔边界上的点，即 $0<\alpha_i<C$



关于上图几个点的 ξ 和 α 的取值，李航的书里是这样写的：

1 点： $0<\alpha_1<C \quad \xi_1=0$

2 点： $\alpha_2=C \qquad 0<\xi_2<1$

3 点： $\alpha_3=C \quad \xi_3=1$

4，5 点： $\alpha_{4,5}=C \quad \xi_{4,5}>1$

ξ 的取值还是比较好理解的，因为 1- ξ 就表示实例点到超平面（绿线）的距离，以 1 点为例，它到超平面的距离为 1，所以 $\xi_1=0$， $1-\xi_1=1$ 。α 的取值得从 KKT 条件来理解：见 **https://www.cnblogs.com/houzichiguodong/p/9254898.html**

$$
\begin{array}{ll}
y_i(w\cdot x_i+b)\ge 1-\xi_i, & i=1,2,3\ldots n\\
\xi_i\ge 0, & i=1,2,3\ldots n\\
\alpha_i\ge 0, & i=1,2,3\ldots n\\
\mu_i\ge 0, & i=1,2,3\ldots n\\
C-\alpha_i-\mu_i=0, & i=1,2,3\ldots n\\
\sum_{i=1}^{n}\alpha_i y_i=0, & i=1,2,3\ldots n\\
\alpha_i(y_i(w\cdot x_i+b)-1+\xi_i)=0, & i=1,2,3\ldots n\\
\sum_{i=1}^{n}\mu_i\xi_i=0, & i=1,2,3\ldots n
\end{array}
$$

KKT条件

$0<\alpha_1<C$ 时，由于 $C\text{-}\alpha_i\text{-}\mu_i=0$，所以 $\mu_i\ne 0$，因为 $\xi_i\ge 0,\mu_i\ge 0,\sum_{i=1}^{N}\mu_i\xi_i=0$，所以 $\xi_i=0$，所以由 $\alpha_i(y_i(\omega\bullet x_i+b)-1+\xi_i)=0$ 得 $y_i(\omega\bullet x_i+b)-1=0$，所以在支持向量上；$\alpha=C$ 、$\alpha=0$ 时也可以按照 KTT 条件进行分析。

三、 SMO 算法

前面都是进行最优化问题的化简，最终都转换成关于 α 的优化问题：

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

求解这个最优化问题用的是 SMO 算法，我还不理解原理，只知道步骤。每次选择两个 α 进行优化，而固定其他的 α 值。可以分为两步，一是选择两个 α，二是更新这两个 α

1、选择 α_1 和 α_2

首先说一下我对这一步的感受：太绕了，而且教材上有几个地方可以这样理解，也可以那样理解，得多看几遍次才行。

可以用两层循环实现，第一层选 α_1，第二层选 α_2。按照上面红色文字部分的分析，α_1 要满足以下的关系：

$\alpha_1=0$ 时，$y_1 g(x_1)\ge 1$；$0<\alpha_1<C$ 时，$y_1 g(x_1)=1$；$\alpha_1=C$ 时，$y_1 g(x_1)\le 1$；

α_1 就从违反这些关系的 α 中选择，优先选择 $0<\alpha_1<C$ 但 $y_1 g(x_1)\ne 1$ 的。如果找不到违反这些关系的 α_i，说明已经达到最优，停止搜索。

选出 α_1 之后，选择 α_2 的原则是使 $|E_1-E_2|$ 最大，其中 $E_i=g(x_i)-y_i$，选出 α_2 之

后，按照一定方法（后面会写）更新 α_2 和 α_1。如果更新之后，优化目标函数 $\min L$ 没有产生足够的下降（与预设阈值相比），就放弃当前 α_2，并且不再以 $|E_1-E_2|$ 最大为原则来搜索 α_2，而是选择在（0，C）内的 α_2。如果再次更新 α_2 和 α_1 之后还是没有使优化目标函数 $\min L$ 产生足够的下降，放弃当前 α_2，并再退而求其次，从 α 其他值中选择 α_2（注意，这里并不是说从（0，C）中选一个 α_2 失败之后就可以进入这个 if，而是要把（0，C）内的所有 α_2 都遍历完，如果都失败了，才可以去 $\alpha=0$ 或 $\alpha=C$ 中选择）。如果再次更新之后，依然没有产生足够的下降，那么放弃当前 α_1，重新进入第一层循环以搜索 α_1。

2、更新 α_1 和 α_2

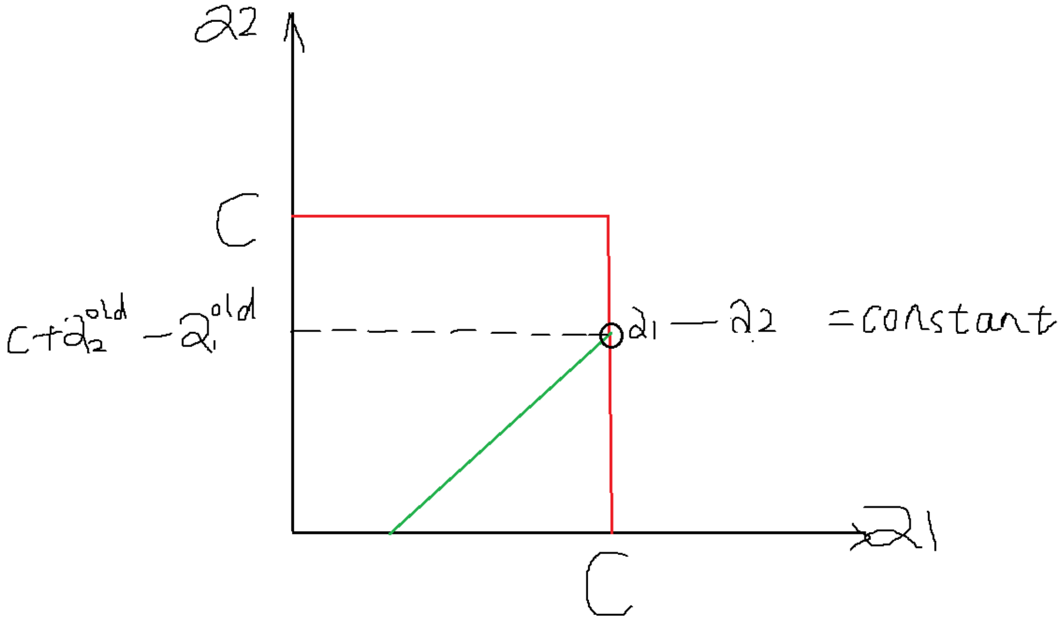
$$
\begin{gathered}
\min \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i \bullet x_j)-\sum_{i=1}^{N}\alpha_i \Leftrightarrow \\
\min \frac{1}{2}\alpha_1^2(x_1 \bullet x_1)+\frac{1}{2}\alpha_2^2(x_2 \bullet x_2)+\alpha_1\alpha_2 y_1 y_2(x_1 \bullet x_2) \\
-(\alpha_1+\alpha_2)+y_1\alpha_1\sum_{i=3}^{N}\alpha_i y_i(x_i \bullet x_1)+y_2\alpha_2\sum_{i=3}^{N}\alpha_i y_i(x_i \bullet x_2)
\end{gathered}
$$

$$
\begin{gathered}
s.t. \alpha_1 y_1+\alpha_2 y_2=-\sum_{i=3}^{N}\alpha_i y_i = constant \\
0 \le \alpha_i \le C
\end{gathered}
$$

因此 α_1、α_2 满足关系：$\alpha_1 y_1+\alpha_2 y_2 = constant$，求出 α_2 后根据这个更新 α_1。那 α_2 是怎么求的呢？公式见下：

$$
\alpha_2^{new,unc}=\alpha_2^{old}+\frac{y_2(E_1-E_2)}{x_1 \bullet x_1+x_2 \bullet x_2-2x_1 \bullet x_2}
$$

为什么上标要加个 unc 呢？因为 α_2 还会受到其他约束，例如 $\alpha_1 y_1+\alpha_2 y_2 = constant$，$0 \le \alpha_2 \le C$，下面对根据 α_2 受到的约束，对 $\alpha_2^{new,unc}$ 进行一个调整。



上图是假设 $y_1=1, y_2=-1$ 且 $0<constant<C$ 时，α_1、α_2 的图像，可以看到这时，α_2 的上界是 $C+\alpha_2^{old}-\alpha_1^{old}$ （这里用到了 $\alpha_1^{old}-\alpha_2^{old}=constant$ ）。同理，根据 y_1 、 y_2 、 $constant$ 的取值不同，可以得到 α_2 的不同上下界，总结一下可以得到：

$$y_1=y_2 \text{ 时，} \alpha_2 \in [max(0,\alpha_1^{old}+\alpha_2^{old}-C), min(C,\alpha_1^{old}+\alpha_2^{old})]$$

$$y_1 \neq y_2 \text{ 时，} \alpha_2 \in [max(0,\alpha_2^{old}-\alpha_1^{old}), min(C,C+\alpha_2^{old}-\alpha_1^{old})]$$

同一记为：$\alpha_2^{new} \in [L,H]$，所以：

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

求出 α_2^{new} 之后，$\alpha_1^{new}=\alpha_1^{old}+y_1y_2(\alpha_2^{old}-\alpha_2^{new})$

其实，到这里就可以写代码了，但是为了完整，再加上核函数的概念

四、 核函数

这个地方书上有一些数学概念，我看不太懂，举个例子说一下我的理解：

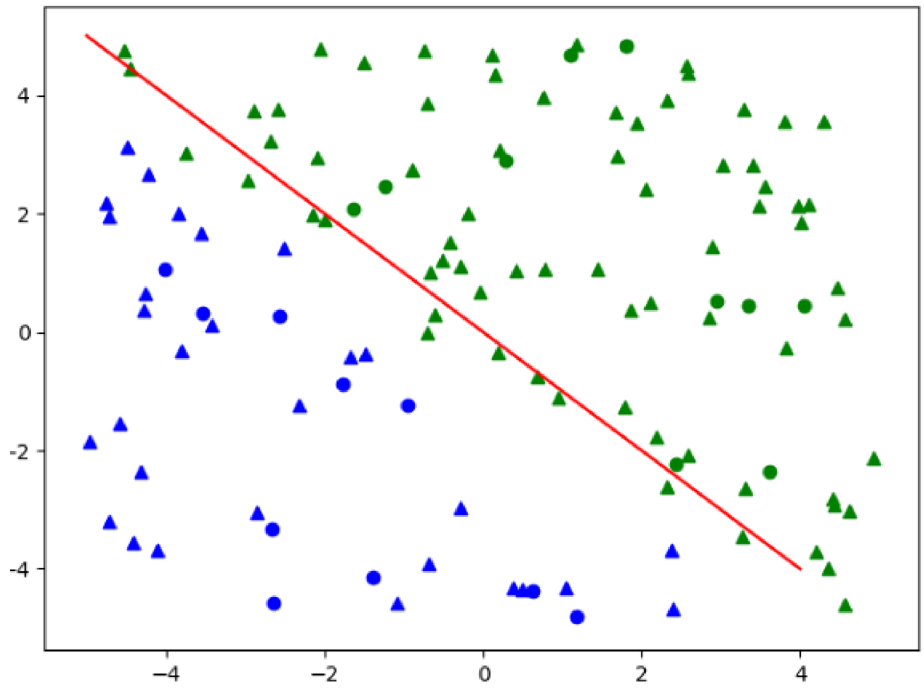
假设有一个椭圆 $\frac{x_1^2}{4}+x_2^2=1$，椭圆内部都是正例点，椭圆外部都是负例点，如果不进行任何变换，虽然利用软间隔的支持向量机可以构建出模型，但是可想而知，一条直线的分类效果无论如何都好不到哪里去。如果在求解之前先进行一个变换，例如：$z=x^2$，即 $z_1=x_1^2, z_2=x_2^2$，分类边界就变为：$\frac{z_1}{4}+z_2=1$，这是个线性模型，求解出来的 SVM 模型准确率肯定很高。

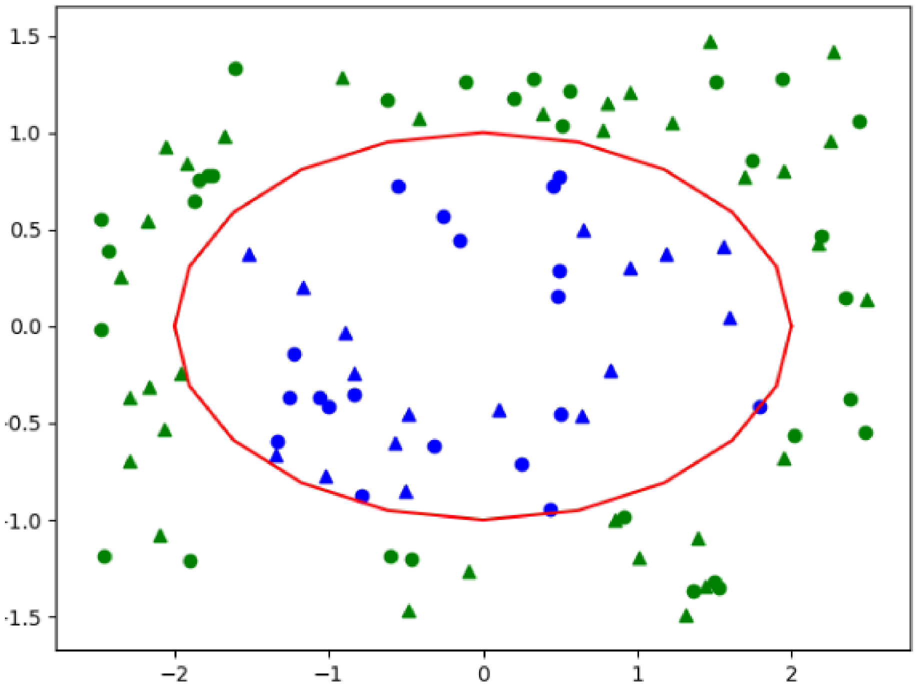
核函数的思想就是这样，把非线性的变为线性的，然后构建 SVM 模型，可以达到比较好的预测效果。创建一个核函数要满足许多数学条件，因此一般用现成的，常用的有多项式核、高斯核等（见统计学习方法）。

在以上化简和求解最优化问题时，可以看到 x_i 都是成对出现的，即都是以 $x_i \bullet x_j$ 的形式出现，所以可以预先定义一个 X 矩阵，用于存储 $x_i \bullet x_j$；对于使用了核函数的 SVM，同样可以定义一个 K 矩阵，用于存储 $z_i \bullet z_j$，并且求解步骤与一二三节完全一致，只需要把 X_{ij} 换成 Z_{ij} 即可。

五、 测试

用了两个案例，上边的是线性可分的，下边是用了核函数的，用颜色表示正负例，圆形代表训练集，三角形代表测试集。没算准确率，不过我感觉效果还是不错的。




1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
−2
−1
0
1
2