实验 1：线性分类器中的感知器学习

1 简介

分类器（Classifier）是一种能够将属于不同类别的数据区分开来的特殊函数，在一个 n 维的特征空间中，这个函数可以表示为一个超曲面（Hyperplane），在超曲面两侧分别是两个不同的类别。当特征空间是 1，2，3 维时，超平面分别对应为一个点、一条曲线和一张曲面。更特殊的情况下，如果分类器是线性的，则曲线和曲面实际上为直线和平面，这也是最简单和最容易实现的分类器。当我们给定两类线性可分的数据时，有很多种方式能够找到这一超平面，最简单的一种是感知器算法（Perceptron），在本文中将详细介绍它的原理和实验过程。

2 感知器原理

在特征空间$\vec{\mathrm{X}}\epsilon\mathrm{R}^{\mathrm{n}}$中定义两个类别 ω_1,ω_2，希望找到的一个线性分类器：

$$\mathrm{g}(\vec{\mathrm{X}}) = \vec{\mathrm{w}}^{\mathrm{T}}\vec{\mathrm{X}} + \mathrm{w}_0 = \hat{\mathrm{w}}^{\mathrm{T}}\hat{\mathrm{X}} = 0 \tag{1}$$

其中 $\hat{\mathrm{w}} = [w_0, w_1, w_2, \dots, w_n] = [w_0, \vec{\mathrm{w}}]$ 定义为该超平面的权向量，$\hat{\mathrm{X}} = [1, x_1, x_2, \dots, x_n] = [1, \vec{\mathrm{X}}]$ 对应的是特征空间中的某一点，可以看出$\vec{\mathrm{w}}^{\mathrm{T}}$对应的是与超平面正交的向量。

所有在超平面上的特征点$\hat{\mathrm{X}}$都满足公式(1)的形式，在这一超平面的两侧分别是两个不同的类别，人为地定义当$\vec{\mathrm{X}} \in \omega_1$时，$\mathrm{g}(\vec{\mathrm{X}}) > 0$；当$\vec{\mathrm{X}} \in \omega_2$时，$\mathrm{g}(\vec{\mathrm{X}}) < 0$，这里也可以采取相反的定义方法，得到的是与同一个超平面正交的两个方向相反的向量，对应的超平面相同，所以对判别结果没有影响。

当任意给定一个初始权向量$\hat{\mathrm{w}}_0$，即初始平面时，不一定能将所有的特征向量按照正确的类别划分，有可能出现当$\vec{\mathrm{X}} \in \omega_1$时，$\mathrm{g}(\vec{\mathrm{X}}) < 0$; 当$\vec{\mathrm{X}} \in \omega_2$时，$\mathrm{g}(\vec{\mathrm{X}}) > 0$ 的情况，因此需要通过一个恰当的代价函数和优化算法来寻找正确的权向量。我

们把所有被错误分类的向量构成的集合定义为Y，同时定义函数$J(\hat{w})$为感知器代价函数，如公式(2)所示：

$$J(\hat{w}) = \sum_{\vec{X}\in Y}(\delta_x \hat{w}^T \hat{X}); \quad \delta_x = \begin{cases} 1, & \hat{w}^T\hat{X} > 0 \\ -1, & \hat{w}^T\hat{X} < 0 \end{cases} \tag{2}$$

按照公式(2)定义的代价函数有两个特点：$J(\hat{w}) \geq 0$，当且仅当$Y = \emptyset$时等式成立，即所有特征向量都被正确分类时，代价函数最小；当错误分类的点偏离超平面越远时，代价函数越大。根据这两个特点，我们可以利用梯度下降法对权向量进行迭代，从而得到满足分类要求的权向量，即：

$$w\widehat{(t+1)} = \widehat{w(t)} - h\frac{\partial J}{\partial \hat{w}} = \widehat{w(t)} - h_t \sum_{\vec{X}\in Y}(\delta_x \hat{X}) \tag{3}$$

其中$\widehat{w(t)}$是当前迭代次数t时的权向量，h_t是学习率(Learning rate)，应该满足$h_t > 0$的条件，学习率可以是某一个预先选定的固定值，也可以是在迭代过程中变化的参数，学习率的选择对迭代过程有一定的影响，可以在下文中对实验结果的分析中看出。

我们把上述得到超平面的算法称为感知器算法，可以看出他的原理并不复杂，而且对于线性可分的数据能够很好的找到满足条件的解，下面我们将对具体的实验过程进行介绍。

3 生成数据集

线性可分的数据集可以是n维特征空间的样本点的集合，但无论样本点的维度是多少，线性分类器的构造过程都是十分类似的，高维的分类器可以采取与二维分类器类似的方法构造，但很难直观地表现出来，因此本文中仅对比较直观的二维数据进行分析。

实际应用当中的样本特征大多数都是服从正态分布的随机向量，不同类别的样本可能具有不同的均值$\vec{\mu}$和方差$\sum$，因此我们在实验中也假设样本点服从正态分布，并定义样本间距（Amount of separation）为两类样本均值之间的距离：

$$d = \|\overrightarrow{\mu_1} - \overrightarrow{\mu_2}\|$$

样本点示例如图 1 所示，类别 1 用*号标明，类别 2 用+号标明，每个类别随机生成 100 个样本点，其中 80 个被用来训练分类器，用红色标出，剩下的 20 个用于测试，用蓝色标出。从图 1 上可以明显看出该数据集是线性可分的。

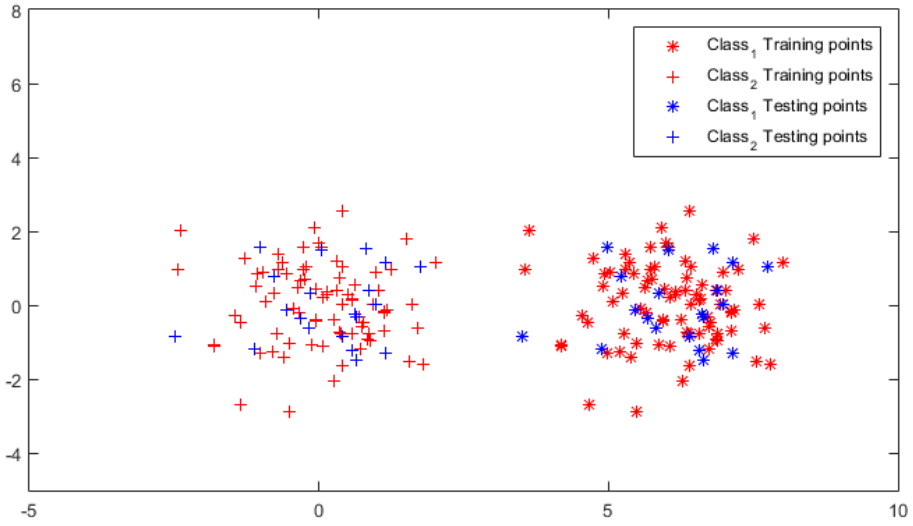

图 1　服从二元正态分布的随机数据集

4　感知器算法

根据感知器的设计原理，我们可以按照如下方式循环迭代得到能将两类样本分开的超平面：

1. 设定初始的权向量$\hat{w}_0$；
2. 设置学习率h_t;
3. 循环:

   $J(\hat{w}) = 0$;

   对每一个样本点，计算$\hat{w}^T\hat{X}$;

   如果分类错误，则$J(\hat{w}) = J(\hat{w}) + \delta_x\hat{w}^T\hat{X}$；$\hat{w} = \hat{w} - h_t\delta_x\hat{X}$;

   迭代步数$t = t + 1$;
4. 直到 $J(\hat{w}) == 0$

5　实验过程

5.1 仿真示例

数据集：类别 1 的样本点服从分布$\{\vec{X}|\vec{X}\in\omega_1\}\sim N\left(\begin{pmatrix}6\\0\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right)$,类别 2 的样本点服从分布 $\{\vec{X}|\vec{X}\in\omega_2\}\sim N\left(\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right)$，根据定义，类别间距为$d=6$。为了保证实验的可重复性，我们所选择的具有相同均值和方差的数据集完全相同。

初始化参数：初始权向量$\hat{w}_0=[-1\ 1\ 1]^T$；学习率$h_t=0.1$

经过迭代收敛得到超平面在图 2 中用黑色实线画出，迭代过程中每一步的权向量表示的超平面用绿色虚线画出，迭代收敛的步数$T=28$，测试结果能够很好的区别开来。

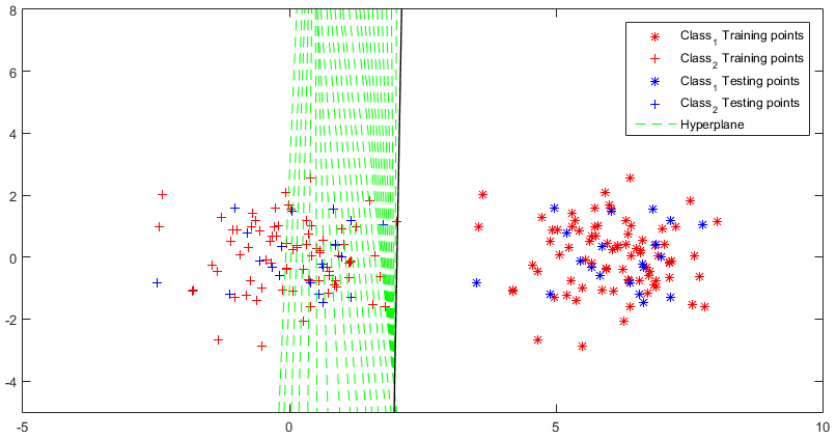

图 2　学习率为$h_t=0.1$时收敛得到的超平面

5.2 学习率对收敛过程的影响

在不改变其他参数的情况下，改变学习率h_t，重复多次实验，得到的迭代收敛步数如表 1 所示，值得注意的是，收敛步数并没有随着学习率的降低而单调递增，而是以先减小再增大的方式变化，当$h_t=0.0035$时，收敛步数最小，$T_{min}=17$。

表 1　收敛步数随学习率的变化规律

h_t	1	0.1	0.01	0.004	0.0035	0.002	0.001	0.0001	0.00001
T	28	28	25	20	17	21	40	363	>1000

对比$h_t = 0.1$和$h_t = 0.0035$的超平面，可以发现两者的差别比较明显，且收敛过程也有很大差别，可以看出用感知器算法得到的分类器对学习率比较敏感。

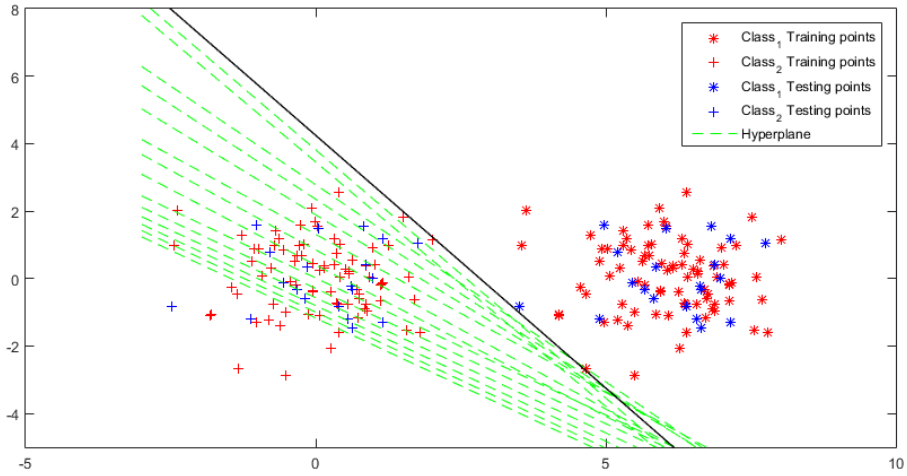

图 3　学习率为$h_t = 0.0035$时收敛得到的超平面

从图 4 可以看出当$h_t < 0.01$时，权向量可以很好的收敛到距离最近的极值点，当$h_t > 0.01$时，权向量跳过了最近的极值点，并被吸引到另一个极值点上去，由于满足分类要求的超平面有无穷多个，且比较分散，因此收敛步数对学习率十分敏感，且不难发现，初始权向量的位置同样也决定了收敛步数的大小，当初始权向量与收敛点距离较远时，收敛步数也就越大。

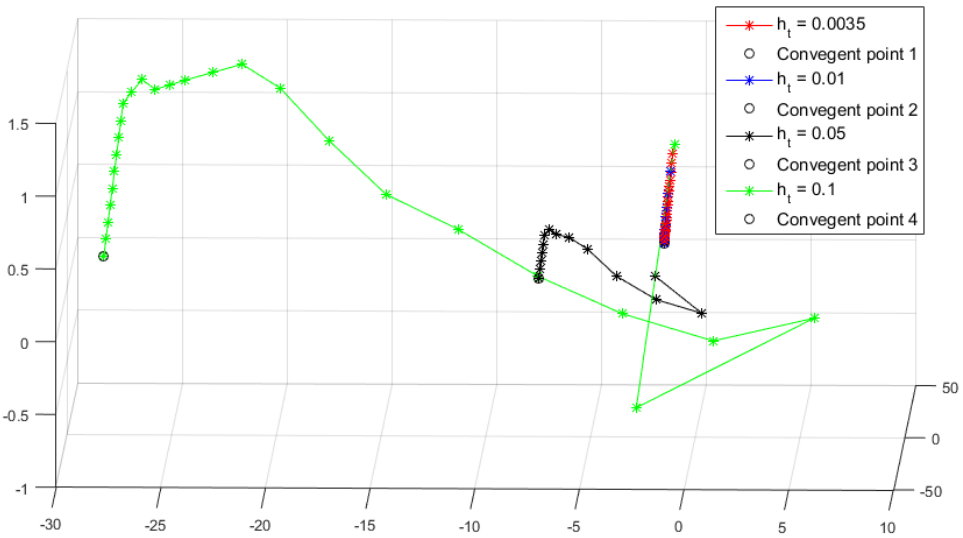

图 4　不同学习率下的权向量收敛轨迹

5.3 类别间距对收敛过程的影响

数据集：为了改变样本的类别间距，我们让类别 1 的样本点分别服从分布$\{\vec{X}|\vec{X}\in\omega_1\}\sim N\left(\begin{pmatrix}4.5\\0\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right)$和$\{\vec{X}|\vec{X}\in\omega_1'\}\sim N\left(\begin{pmatrix}8\\0\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right)$，类别 2 的样本点服从分布$\{\vec{X}|\vec{X}\in\omega_2\}\sim N\left(\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}1&0\\0&1\end{pmatrix}\right)$，类别间距分别为$d=4.5$和$d'=8$。

初始化参数：初始权向量$\hat{w}_0=[-1\ 1\ 1]^T$；学习率$h_t=0.0035$

分别选择两个均值不同的类别 1 样本ω_1和ω_1'与类别 2 样本ω_2作为分类器的数据集，得到的收敛过程迭代次数与类别间距的关系如表 2 所示。可以看出类别间距d较小的数据集的收敛速度明显比d较大的要慢。

表 2　超平面收敛过程的迭代次数随类别间距的变化规律

类别间距 d	4.5	6	8
迭代次数 T	211	17	26

以ω_1和ω_2作为数据集的类别，分析在类别间距较小的情况下超平面的收敛过程，从图 5 中我们发现当两个不同类别的样本点相距较近时，每一步收敛过程中权向量的改变量都特别小，因此在这些点附近会让收敛速度大大降低。

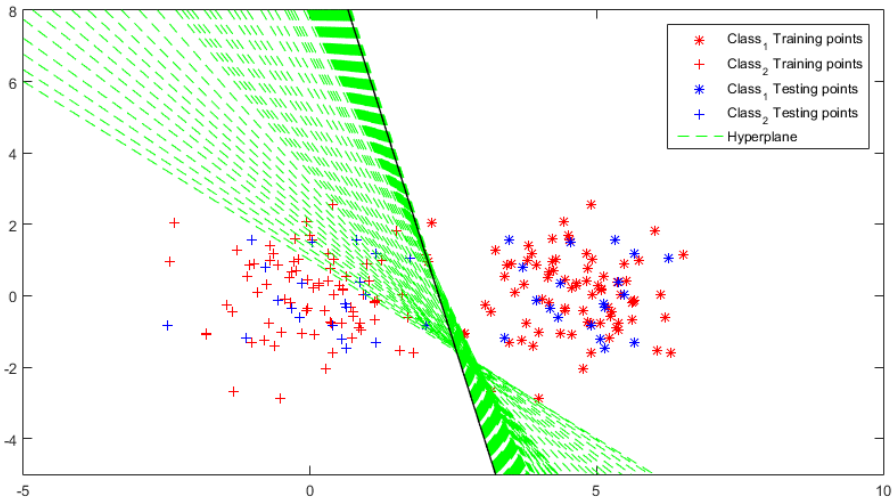

图 5　类别间距$d=4.5$时超平面的收敛过程

以ω_1'和ω_2作为数据集的类别，分析在类别间距较大的情况下超平面的收敛过程，从图 6 中可以看出，由于两类样本点之间相距较远，不存在图 5 中两个相距较近的点的约束，因此收敛过程较快，但超平面几乎与其中一类样本点的边界重合，如果该类别中的样本点再稍微往外一点就有可能出现判别出错的情况。

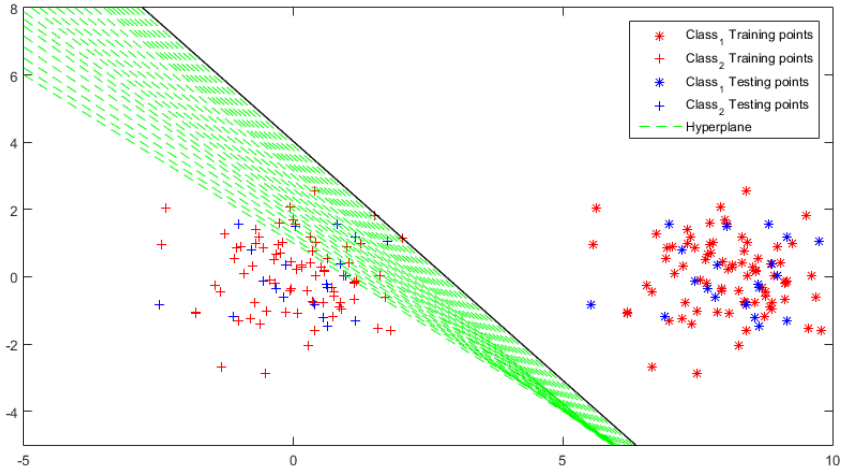

图 6　类别间距 d=8 时超平面的收敛过程

5.4 初始权向量对收敛过程的影响

以ω_1和ω_2作为数据集的类别，改变初始权向量的位置，我们得到了多个如图 7 所示的能将两类样本分开的超平面，可以看出超平面的位置主要受到相距最近的两个点的约束，当样本中存在这样两个点时，超平面所在范围较小，对初始权向量的位置不太敏感。

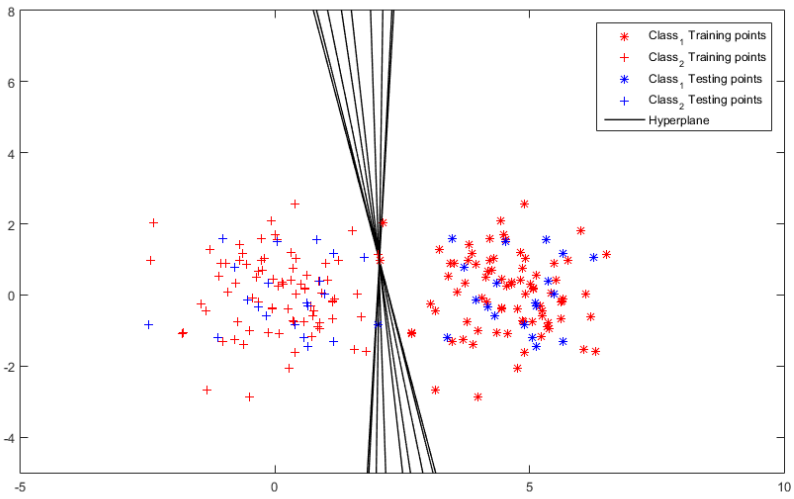

图 7　类别间距 d=4.5 时不同初始条件得到超平面

从图 8 中给出的权向量收敛过程可以看出，初始权向量在不同位置下得到收敛点差别不大，因此可以认为在利用感知器学习处理类别间距较小的数据集时，初始参数对结果的影响不大。

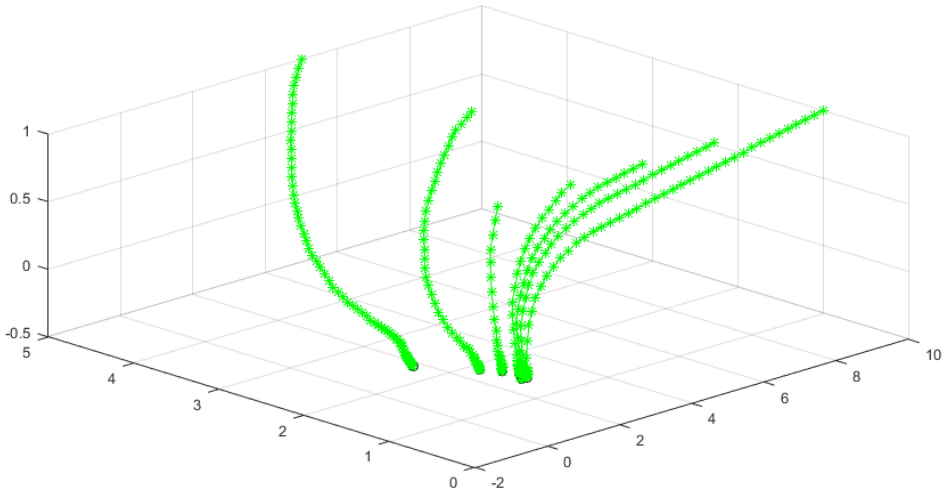

图 8　类别间距 d=4.5 时不同初始条件下权向量的收敛过程

研究ω_1'和ω_2类别的数据集，从图 9 中可以看出，在类别间距较大情况下，如果改变初始权向量的位置，超平面的收敛结果会有很明显的差别，其中的某些超平面与样本点相距较近，虽然能够将当前的样本点分开，但对于其他测试用的样本点很有可能会出现分类错误，因此感知器算法对于类别间距较大的数据集的适用性较差。

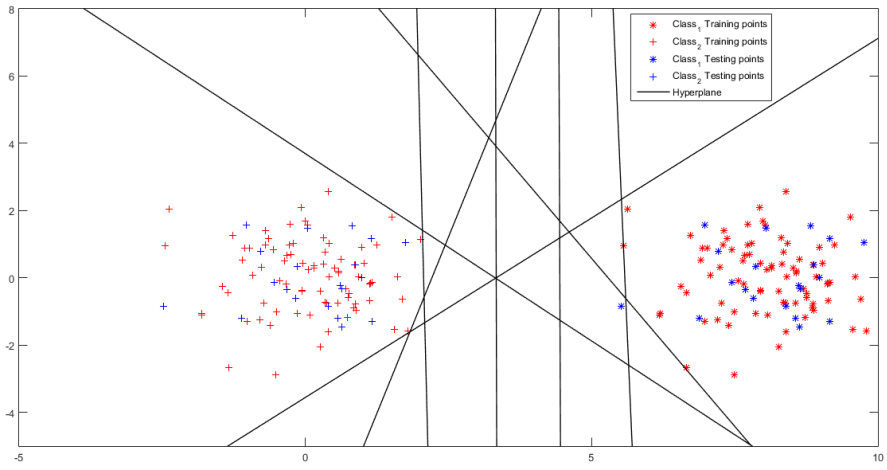

图 9　类别间距 d=8 时不同初始条件得到超平面

6 结论

通过对感知器算法中学习率μ_t，数据集类别间距d，以及初始权向量$\hat{w}_0$对收敛过程的分析，我们可以得到以下结论：

1. 收敛步数并不是严格地随着学习率降低而增大。对于给定的数据集和初始权向量，存在一个最佳的学习率使得收敛步数最小。当学习率小于某一值后，收敛步数会随着学习率下降而急剧增大；
2. 当类别间距较小时，如果存在不同类别相距较近的点，则在这几个点附近收敛速度会下降很多，使得收敛步数增加；因为存在超平面的区域较小，初始参数对收敛结果不会产生很大影响。
3. 当类别间距较大时，由于不同类别之间的样本点相距较远，限制条件不严格，因此超平面的范围较大，收敛步数较小，但收敛结果对初始条件比较敏感，很难取到最佳的超平面。如果能在方程中增加约束，使得超平面到两类样本的距离最远，则可以得到适用性更强的超平面。