

# 作业 1

林宇浩 21311274

## (1) SVM 模型的一般理论

### 1、最大边缘分类器

支持向量机其最初的形式是最大边缘分类器，该模型的核心思想是寻找一个最优的超平面来将不同类别的样本点分开，并且使得两侧距离最近的样本点到超平面的距离最大化。例如，考虑一个二维空间中的两类数据点，可以通过一条直线将它们分开。最大边缘分类器的目标是找到一条直线，使得离这条直线最近的两个类别的数据点到直线的距离最大。

### 2、最优化问题

为了找到最佳的超平面，需要解决一个最优化问题，涉及到找到能够最大化边缘的超平面，同时确保所有数据点都在超平面两侧的约束条件。这个优化问题可以被转化成一个凸二次优化问题，可以通过引入约束条件来实现最优化求解。

### 3、对偶问题

为了解决优化问题，使用拉格朗日乘子法将原始问题转化为对偶问题。通过引入拉格朗日乘子，将约束条件纳入目标函数中，形成拉格朗日函数。对拉格朗日函数求极值，得到对偶问题，从而简化优化过程。通过对这个拉格朗日函数进行求导，并令导数等于零，可以得到一组方程，从而可以得到一组拉格朗日乘子的值，对应着每个数据点在超平面位置上的重要性。

### 4、支持向量

在 SVM 中，只有少数样本点会对决策边界产生影响，这些样本点被称为支持向量。由于拉格朗日乘子的稀疏性，只有与支持向量对应的拉格朗日乘子不为零，其他样本点的乘子都为零。支持向量是在最优超平面确定过程中位于边缘上的数据点，支持向量决定了最终超平面的位置。

### 5、软间隔

当数据不是线性可分时，最大边缘分类器通过引入松弛变量来容忍一定程度上的误分类，可以得到软边缘分类器。通过引入松弛变量来允许一些数据点位于错误的一侧或在边缘附近，松弛变量表示了数据点与正确分类超平面之间的函数间隔或几何间隔之间的差距。优化问题变为了一个权衡边缘最大化和错误容忍度之间的折中问题。

### 6、支持向量机

支持向量机在软最大边缘分类器基础上发展起来，可以处理线性可分和线性不可分的数据，通过核函数处理非线性可分的数据，使得原本线性不可分的问题可以在新的特征空间中找到最优超平面，从而提高了模型的适用范围和灵活性。

### 7、核函数

通过引入核函数，SVM 将原始空间中的数据映射到一个高维的特征空间，从而使得非线性问题也能够在高维空间中线性可分。常用的核函数包括线性核、多项式核、高斯核等。数学上，核函数必须满足所谓的 Mercer 条件，即任何满足 Mercer 条件的函数都可以被用作核函数。满足 Mercer 条件的核函数可以保证 SVM 的有效性和性能。

### 8、合页损失

合页损失是 SVM 中的一种损失函数，用于最小化分类错误和最大化分类边界的间隔。合页损失函数可以被视为对逻辑回归中的对数损失函数的一种泛化形式，在 SVM 中用于最大化间隔并确保正确分类的同时，对误分类样本有适当的惩罚。

## (2) 采用不同核函数的模型和性能比较及分析

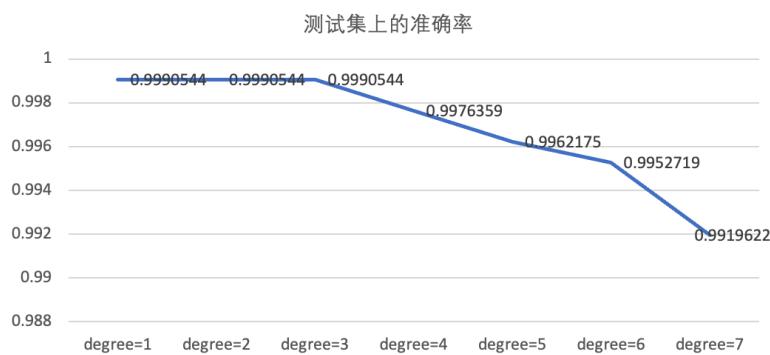
核函数	训练集上的准确率	测试集上的准确率
线性核函数	1.0000000	0.9990544
高斯核函数	0.9999210	0.9995272

可以看到，与线性核函数相比，高斯核函数在训练集上的准确率更低，但是在测试集上的准确率更高。说明在本例中，高斯核函数相比线性核函数有着更好的泛化能力。

高斯核函数的映射空间是无穷维的，其可以将数据映射到一个无限维的特征空间。而线性核函数的映射空间是有穷维的，其在一个更低维的空间中进行线性分类。实验结果说明，在本例中，更高维度的映射空间似乎更有利于数据的划分。

除了以上两个核函数外，还使用了多项式核函数对映射空间维度的影响进行评估。多项式核函数将数据映射到一个更高维的特征空间，其维度取决于指定的多项式阶数。比如，假设原始数据是二维的，即  $x=(x_1, x_2)$ ，映射到三次多项式空间后为  $x' = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_2^3, x_1^2x_2, x_1x_2^2)$ 。实验结果如下：

多项式核函数的维度	训练集上的准确率	测试集上的准确率
degree=1	1.0000000	0.9990544
degree=2	1.0000000	0.9990544
degree=3	1.0000000	0.9990544
degree=4	1.0000000	0.9976359
degree=5	1.0000000	0.9962175
degree=6	1.0000000	0.9952719
degree=7	1.0000000	0.9919622



可以看到，随着维度增加，模型在测试集上的准确率下降，说明过高的多项式阶数导致了过拟合的问题，并不是特征空间维度越高越好。这也说明了高斯核函数的无穷维特征空间并不是简单地通过提高特征空间的维度来实现更好的泛化性。高斯核函数的平滑性质指出了在其输入空间中相邻的数据点对应的核函数值变化平稳，能够较好地处理噪声、异常值，使得在处理非线性关系时更不容易出现过拟合问题。

## (3) 采用 hinge loss 的线性分类模型和 SVM 模型之间的关系

对于线性分类模型，比如感知机，其输出可以表示为  $y = f(wx + b)$ ，其中  $f(x)$  为阶跃函数（如果  $x < 0$ ,  $f(x)=0$ ；如果  $x \geq 0$ ,  $f(x)=1$ ）。某个数据的损失函数为  $g(y)$ ，对于标签  $y'$ ，如果  $y=y'$ ，则  $g(y)=0$ ；如果  $y \neq y'$ ，则  $g(y)=1$ 。总的损失  $L(w, b) = \sum_i^N g(y_i) = \sum_i^N g(f(wx_i + b))$ 。可以得到这个损失与下面的损失是等价的，其结果均为计算分错类别的

数据的数量：

$$L(\mathbf{w}, b) = \sum_{n=1}^N E_{Ideal}(y^{(n)} h^{(n)}) \quad \text{where } E_{Ideal}(z) = 0 \text{ if } z \geq 0; 1 \text{ otherwise}$$

对于逻辑回归，根据课程 PPT 中公式的改版，其损失函数有下式成立：

$$\begin{aligned} L(\mathbf{w}, b) &= - \sum_{n=1}^N [\tilde{y}^{(n)} \log \sigma(h^{(n)}) + (1 - \tilde{y}^{(n)}) \log(1 - \sigma(h^{(n)}))] \\ &= \sum_{n=1}^N \log(1 + \exp(-\mathbf{y}^{(n)} \mathbf{h}^{(n)})) \\ &= \sum_{n=1}^N E_{LR}(y^{(n)} h^{(n)}) \end{aligned}$$

Note:  
 $\tilde{y} \in \{0, 1\}$ ,  $y \in \{-1, 1\}$

where  $E_{LR}(z) = \log(1 + \exp(-z))$

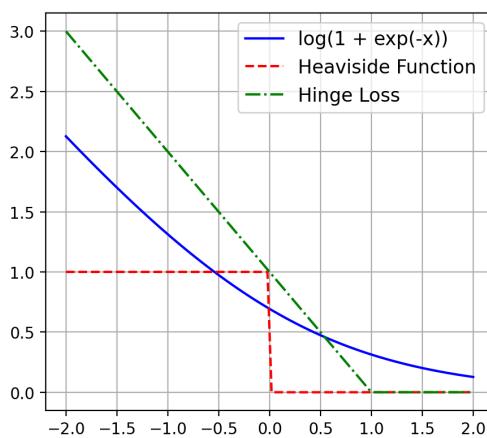
对于 SVM 模型，根据课程 PPT 中的公式，其损失函数如下所示：

$$\begin{aligned} L(\mathbf{w}, b) &= C \sum_{n=1}^N E_{SV}(y^{(n)} h^{(n)}) + \frac{1}{2} \|\mathbf{w}\|^2 \\ &= \sum_{n=1}^N E_{SV}(y^{(n)} h^{(n)}) + \lambda \|\mathbf{w}\|^2 \quad \text{where } E_{SV}(z) = \max(0, 1 - z) \end{aligned}$$

根据线性分类模型损失的形式，对于采用 hinge loss 的线性分类模型，其损失函数则为如下形式：

$$L(\mathbf{w}, b) = \sum_{n=1}^N E_{SV}(y^{(n)} h^{(n)})$$

可以发现，相比 SVM 模型，采用 hinge loss 的线性分类模型，其损失函数不包含  $\lambda \|\mathbf{w}\|^2$  这项。如果加上这一项，对其而言相当于添加了一个正则化项，可以使模型参数趋向于较小的值，防止模型过拟合。



从模型的含义上来讲，对于  $y \cdot h$ ，当数据被分类正确时， $y \cdot h > 0$ ；当数据被分类错误时， $y \cdot h < 0$ ；当数据位于判决边界上时， $y \cdot h = 0$ 。而  $y \cdot h$  的值越大，其被分类正确，且离判决边界越远。 $y \cdot h$  的值越小，其被分类错误，且跨过判决边界越远。

上图为三种损失函数对应的曲线。线性模型使用 hinge loss 后，只有数据被分类正确且离判决边界较远，即  $y \cdot h > 1$  时，损失才会为 0。相比感知机，hinge loss 条件更为苛刻，即使

数据被分类正确，但如果  $y \cdot h < 1$ ，仍然会产生损失。这则对应了我们期望数据与判决边界之间有较大的边缘的目标。并且由于 hinge loss 左侧是斜率为 -1 的直线，数据被分错后离判决边界越远，损失值也随之线性增加。这则对应了我们希望对于分错类的数据点，它们到判决边界的距离之和尽量小。

另一方面，相比逻辑回归，当对于分类正确且  $y \cdot h > 1$  的数据，它们在损失上的体现是一样的，均为 0。而对于逻辑回归，被分类正确的数据，其离判决边界越远，损失也会越小。

这则是对应了支持向量的概念，对于边缘外的点，它们不直接决定判决边界的位置。

综上，可以发现，对于采用 hinge loss 的线性分类模型，其产生的效果与 SVM 模型是一致的。对于软边缘分类器，合页损失和正则化项  $\lambda \|w\|^2$  都是其数学推导过程中自然产生的。而对线性分类模型使用 hinge loss，则是人为对线性分类模型的改进，原本的损失并不带有正则化项  $\lambda \|w\|^2$ ，我们可以根据需要更进一步的改进其损失，详见第（6）部分的分析。

同时，SVM 引入了基函数  $\Phi(\cdot)$ ，通过基函数将原始数据  $x$  转换到特征空间，如课程 PPT 中下式所示：

$$\begin{aligned} \min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{n=1}^N \xi_n \\ s.t.: y^{(n)} \cdot (w^T \phi(x^{(n)}) + b) \geq 1 - \xi_n, \\ \xi_n \geq 0, \quad \text{for } n = 1, 2, \dots, N \end{aligned}$$

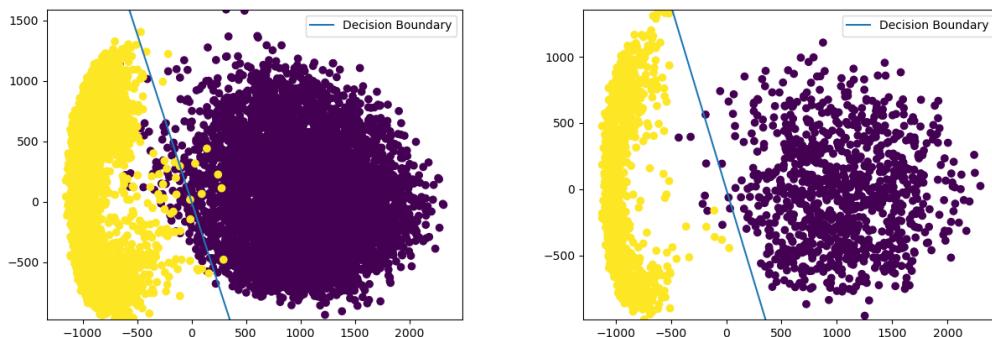
当基函数对应的核函数不是线性核时，SVM 的在原始数据空间的判决边界可以不是线性的，从而分离非线性数据。而采用 hinge loss 的线性分类模型，其判决边界只能是线性的。

#### (4) 采用 hinge loss 线性分类模型和 cross-entropy loss 线性分类模型比较

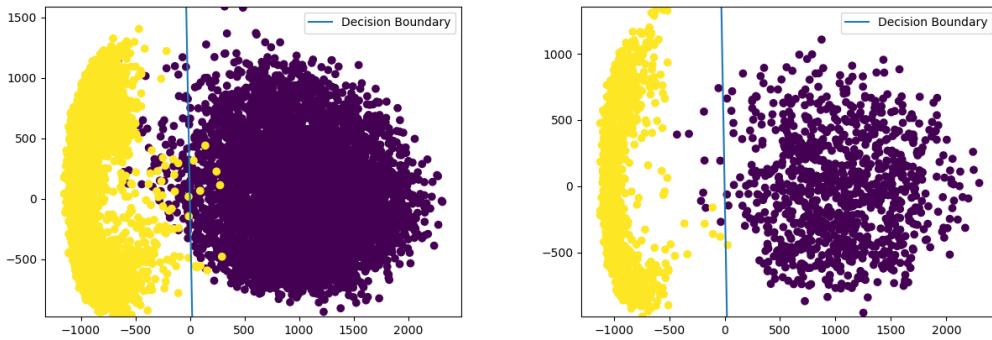
使用的是损失函数	训练集上的准确率	测试集上的准确率
hinge loss	0.9992894	0.9990544
hinge loss 带 L2 正则化	0.9994473	0.9990544
cross-entropy loss	0.9760758	0.9815603

我们将数据映射到二维空间，以查看判决边界的划分情况，如下所示，左侧是训练集，右侧是测试集：

a. 采用 hinge loss 线性分类模型：



b. 采用 cross-entropy loss 线性分类模型：



可以看到，数据的分布情况是黄色类别下方会有一些数据偏右，紫色类别上方会有一些类别偏左。采用 hinge loss 线性分类模型能够注意到这个边界信息，所以判决边界是一条斜线，以让各个分错类的数据点到判决边界的距离之和尽量小。在采用 cross-entropy loss 线性分类模型的图中，可以看到存在一些分错类的数据点跨过边界很远的情况，而在 hinge loss 的图中，分错类的数据点离边界的距离相比之下都较近。采用 cross-entropy loss 线性分类模型中，由于每个数据都会贡献损失，所以其判决边界体现更多的是数据整体的情况。

#### (5) 训练过程（包括初始化方法、超参数参数选择、用到的训练技巧等）

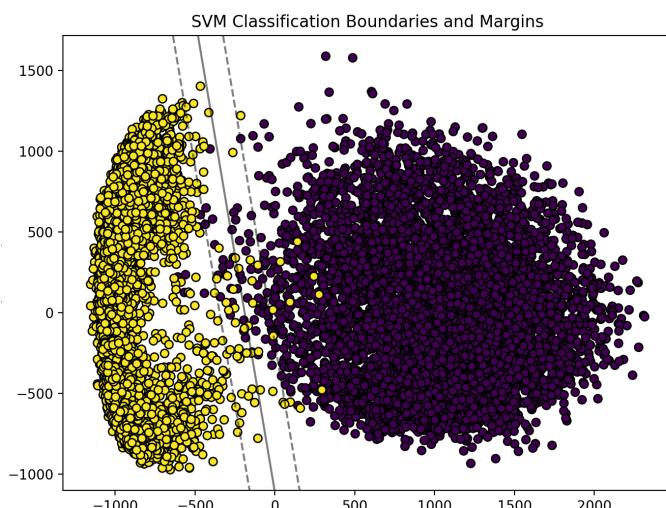
初始化方法：随机初始化，设置了随机种子以确保结果的可重复性和可比性。

超参数选择：采用随机梯度下降法，学习率设定为 0.01，正则化项系数为 0.01，训练轮数均为 10 轮。

训练技巧：使用 hinge loss 时，原始数据的标签为 0 和 1，这导致  $y \cdot h$  的含义和数学推导中不一致，所以一开始则将标签从 {0, 1} 映射到 {-1, 1}，能够给后面代码的构建带来便利。

#### (6) 实验结果、分析及讨论

下图为 (2) 部分中使用线性核函数的 SVM 模型的判决边界以及边缘。数据采用同样的方式映射到二维空间中以作展示。



可以看到其判决边界的位置，和采用 hinge loss 的线性分类模型的判决边界类似，正则化

项在本例中的影响并不大。

我们进一步分析边缘约束对判决边界划分情况的影响。

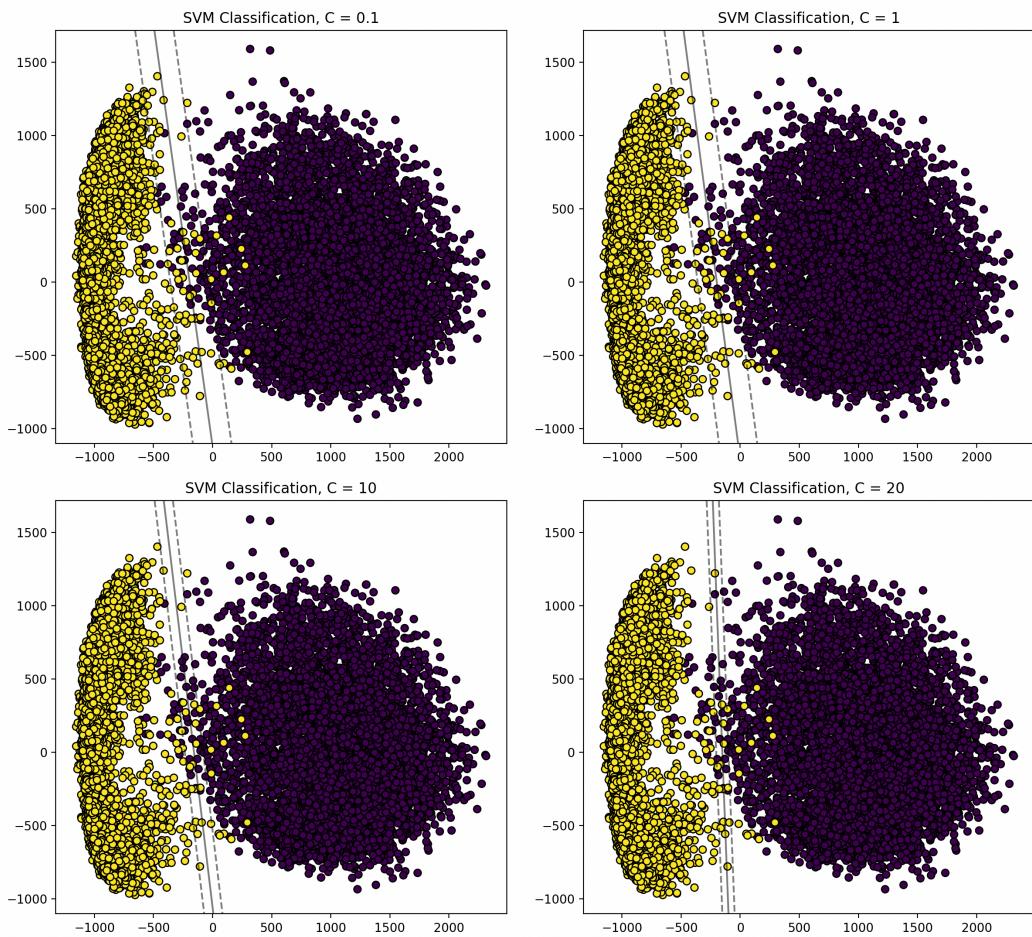
$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

$$s.t.: y^{(n)} \cdot (\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b) \geq 1 - \xi_n, \\ \xi_n \geq 0, \quad \text{for } n = 1, 2, \dots, N$$

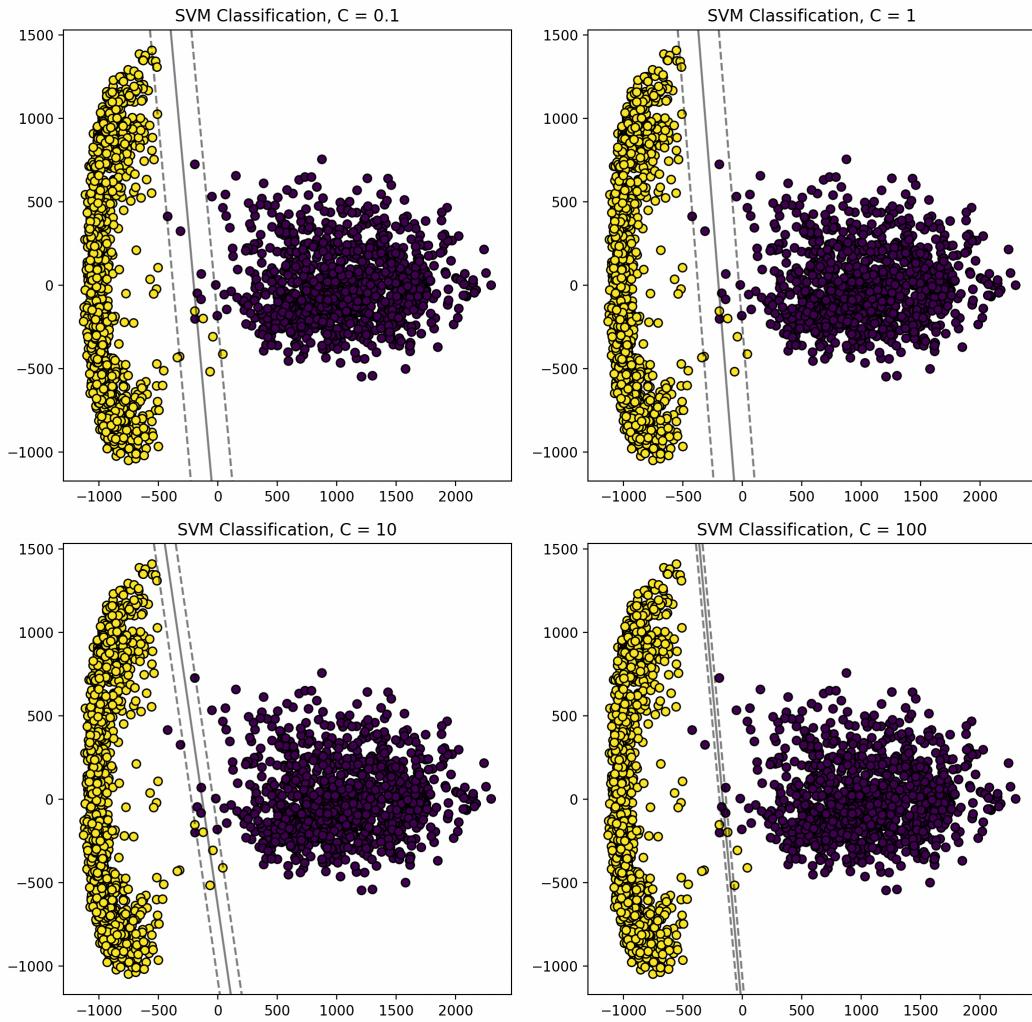
$$L(\mathbf{w}, b) = C \sum_{n=1}^N E_{SV}(y^{(n)} h^{(n)}) + \frac{1}{2} \|\mathbf{w}\|^2$$

SVM 的优化问题如上式所示，根据限制条件即得到右边的损失函数。当 C 越小时，松弛变量之和这项的权重越小，约束条件更为宽松。当 C 越大时，松弛变量之和这项的权重越大，约束条件更为严格。

下图为不同 C 值时，采用线性核函数的 SVM 在训练集上的判决边界和边缘的情况：



下图为不同 C 值时，采用线性核函数的 SVM 在测试集上判决边界和边缘的情况：

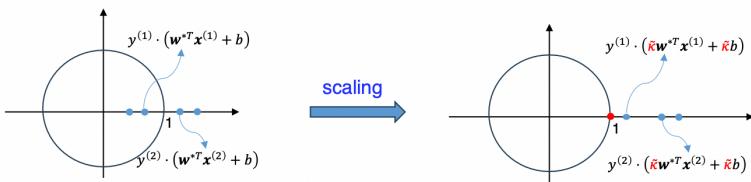


可以看到，C 值越大导致约束越严格，边缘范围变得越小，模型会尽可能地正确分类所有的样本点，但可能导致对训练数据过拟合，更容易受到噪声或异常点的影响。

下面进一步分析 SVM 的损失函数。我们知道 SVM 中  $y \cdot h \geq 1$  的限制是在引入约束处理最优化问题时假设的，如课程 PPT 中下式所示。那么我们是否可以引入其他  $y \cdot h \geq k > 0$  的限制，随着 k 的不同，将导致合页函数沿着 x 轴平移。

$$y^{(\ell)} \cdot (\tilde{\kappa} \mathbf{w}^{*T} \mathbf{x}^{(\ell)} + \tilde{\kappa} b^*) \geq 1 \text{ for all } \ell = 1, 2, \dots, n$$

and at least one '=' must hold



对不同 k 值的实验测试结果如下：

k	训练集上的准确率	测试集上的准确率
0.1	0.9992894	0.9990544
1	0.9992894	0.9990544
5	0.9994473	0.9995272
10	0.9992894	0.9995272
100	0.9994473	0.9990544
1000	0.9994473	0.9990544

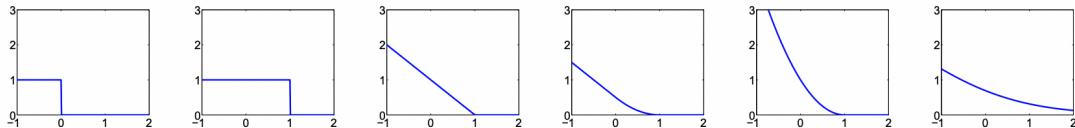
可以看到，取不同的 k 值对实验结果几乎没有影响，这与数学推导时的情况相符，只要 w 和 b 的值相应变化，就可以抵消 k 值的变化，所以 k 的取值并不影响模型的结果。

接下来从线性模型的视角继续分析损失函数。合页损失来自 SVM 模型，对于线性模型，合页损失并不是从数学推导中得来的，我们通过损失函数效果的视角，可以对合页损失进行改进。比如分段平滑的合页损失[1]和二次平滑的合页损失[2]，公式分别如下所示：

$$\ell(y) = \begin{cases} \frac{1}{2} - ty & \text{if } ty \leq 0, \\ \frac{1}{2}(1 - ty)^2 & \text{if } 0 < ty \leq 1, \\ 0 & \text{if } 1 \leq ty \end{cases}$$

$$\ell(y) = \frac{1}{2\gamma} \max(0, 1 - ty)^2$$

平滑的合页损失处理了合页转折处不可导的问题，在优化过程中具有更好的稳定性。下面论文中还对比了大量其他合页损失的变种：



[1]Rennie, Jason D. M.; Srebro, Nathan (2005). Loss Functions for Preference Levels: Regression with Discrete Ordered Labels (PDF). Proc. IJCAI Multidisciplinary Workshop on Advances in Preference Handling.

[2]Zhang, Tong (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms (PDF). ICML.