COMP9417 - 机器学习教程。核方法

问题1（双感知器）

回顾讲座中Perceptron训练算法的发展。现在将其与 "核方法 "讲座中介绍的*双倍形式的*Perceptron训练算法进行比较。这两种算法非常相似，但是在一些关键的地方有所不同。解释一下双重版本的算法与原始算法的关系。

解决方案。

在本答案中，我们忽略了学习率η，或者等同于假设$\eta=1$。在最初的Perceptron训练算法中，我们尝试对当前的例子进行分类，并检查是否有错误，如果$y_i \ \mathbf{w}, \ \mathbf{x}_i \ 0$，就会出现错误。当这种情况发生时，我们更新权重向量$\mathbf{w}$，向其添加$\eta y_i \mathbf{x}_i$，即$y_i \mathbf{x}_i$。

考虑到这一点，显然，对于*每一个*被错误分类的例子，学到的权重向量将被更新0次或更多次。将这个数字表示为α_i，例如$\mathbf{x}_i$，权重向量可以表示为

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \, y_i \, \mathbf{x}_i$$

在这种*双重*观点下，我们学习的是实例权重α_i，而不是特征权重w_j。一个实例$\mathbf{x}$被归类为

$$\hat{y} = \text{sgn}\left[\sum_{i=1}^{n} \alpha_i \, y_i \, (\mathbf{x}_i \, , \mathbf{x})\right]$$

在训练过程中，唯一需要的关于训练数据的信息是所有成对的点积，这些信息可以很容易地计算和存储在一个叫做Gram矩阵的$n \times n$*矩阵*中。

$$\mathbf{G} = \mathbf{XX}^{\mathbf{T}} = \begin{pmatrix} (\mathbf{x}_1 \, , \mathbf{x}_1) & (\mathbf{x}_1 \, , \mathbf{x}_2) & \cdots & (\mathbf{x}_1 \, , \mathbf{x}_n) \\ (\mathbf{x}_2 \, , \mathbf{x}_1) & (\mathbf{x}_2 \, , \mathbf{x}_2) & \cdots & (\mathbf{x}_2 \, , \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{x}_n \, , \mathbf{x}_1) & (\mathbf{x}_n \, , \mathbf{x}_2) & \cdots & (\mathbf{x}_n \, , \mathbf{x}_n) \end{pmatrix}.$$

问题2（特征转换）

回顾一下，XOR函数（如下图所示）不是线性可分的。说明我们如何能够学习

在对原始数据集进行特征转换后，使用线性分类器的XOR函数。作为一个具体的例子，说明如何扩展前一个问题中的双感知器来学习XOR函数。

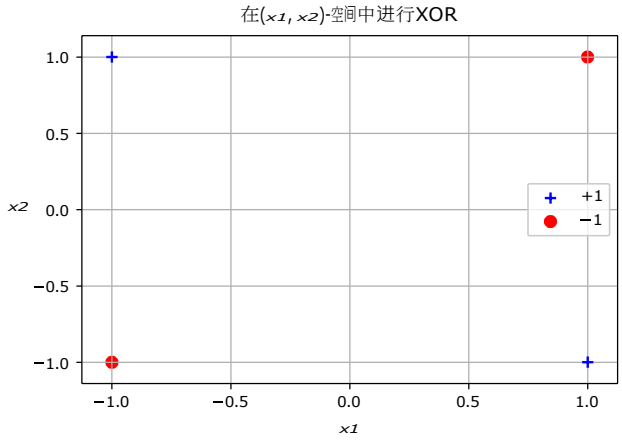


图1：XOR函数

解决方案。

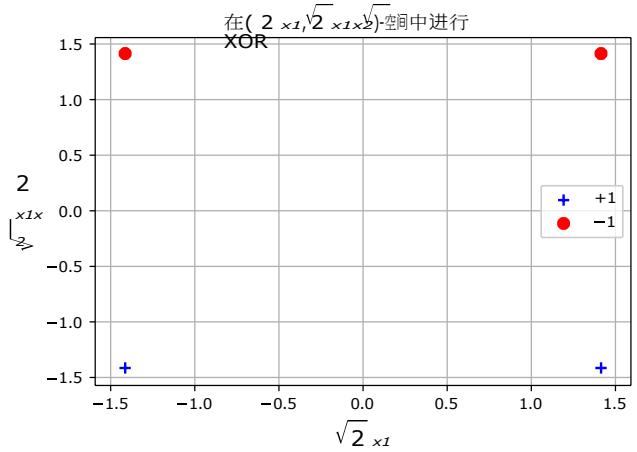
显然，在原始数据上训练的线性分类器不会起作用，因为线性分类器只有在数据是线性可分离的时候才起作用。我们可以尝试学习一个非线性模型，或者像问题中建议的那样，我们可以转换我们的数据，尝试使其成为线性可分离的。有许多可能的转换可以实现这一点。考虑以下情况：变换φ：$R^2 \rightarrow R^6$，定义为

$$\varphi(\mathbf{x}) = \varphi\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ x_1^2 \\ \sqrt{2}x_1 x_2 \\ x_2^2 \end{pmatrix}.$$

换句话说，φ是在二维（原始空间）中获取一个向量$[x_1, x_2]$，并返回一个六维向量（特征空间）的函数。因此，对于我们的XOR问题，我们会有。

$$\varphi\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ \sqrt{2} \\ \sqrt{2} \\ 1 \\ \sqrt{2} \\ 1 \end{pmatrix}, \quad \varphi\begin{pmatrix} -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ -\sqrt{2} \\ -\sqrt{2} \\ 1 \\ \sqrt{2} \\ 1 \end{pmatrix}, \quad \varphi\begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -\sqrt{2} \\ \sqrt{2} \\ 1 \\ -\sqrt{2} \\ 1 \end{pmatrix}, \quad \varphi\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ \sqrt{2} \\ -\sqrt{2} \\ 1 \\ -\sqrt{2} \\ 1 \end{pmatrix}.$$

现在，我们无法将6维空间可视化，但让我们从6个特征中挑选出2个并绘制出来，即，我们将在($2x_1$, $2x_1 x_2$)(分别为第二和第六个特征)的空间中绘图。



很明显，线性分类器现在就可以工作了!因此，我们的想法是首先使用适当的特征变换φ来变换你的数据，然后在变换后的数据上实现我们之前研究过的一些线性分类器。为了使用双感知器学习XOR函数，我们只需在转换后的特征上运行双感知器，这意味着我们只需更新第i个权重，α_i，如果

$$\sum_{j=1}^{N} \alpha_j y_j \varphi(\mathbf{x}_j)\cdot\varphi(\mathbf{x}_i) < 0.$$

所以，所有的变化就是现在我们指的是特征转换的格拉姆矩阵。

$$\mathbf{\Psi} := \begin{pmatrix} (\varphi(\mathbf{x}^1), \varphi(\mathbf{x}^1)) & (\varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2)) & \cdots & (\varphi(\mathbf{x}^1), \varphi(\mathbf{x}^n)) \\ (\varphi(\mathbf{x}_2), \varphi(\mathbf{x}_1)) & (\varphi(\mathbf{x}_2), \varphi(\mathbf{x}_2)) & \cdots & (\varphi(\mathbf{x}_2), \varphi(\mathbf{x}_n)) \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi(\mathbf{x}_n), \mathbf{x}_1) & (\varphi(\mathbf{x}_n), \varphi(\mathbf{x}_2)) & \cdots & (\varphi(\mathbf{x}_n), \varphi(\mathbf{x}_n)) \end{pmatrix}.$$

请注意，我们在这里并不真正需要使用整个特征转换，我们可以简单地使用以下转换$\gamma: \mathbb{R}^2 \to \mathbb{R}^2$：定义如下

$$\gamma(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1 x_2 \end{pmatrix},$$

这显然也允许我们使用一个线性分类器（即使没有$\sqrt{2}$）项。这显然是...

显然，这是一个更好的选择，因为计算一个2维的特征变换比计算一个6维的更便宜。我们将在下一个问题中探讨为什么一般情况下我们会使用高维（甚至无限维）的特征变换。

问题3.（内核诀窍）

利用前一个问题的背景，讨论因必须计算高维特征转换而产生的计算问题。说明如何通过使用内核技巧来缓解这些问题，并利用这一点将双感知器学习扩展到内核感知器学习。

解决方案。

在上一个问题中，我们看到我们可以将线性分类器应用于非线性可分离的数据，方法是先将其转换为特征空间中的线性可分离数据。我们使用了变换φ，并表明可以在变换后的数据上训练双重感知器。这需要我们计算6个维度的变换，然后计算它们之间所有的成对点积--在很多情况下，我们必须计算非常高维度的特征变换，才能让线性分类器在非线性可分离的数据上工作（想想现实世界的数据集）--因此这种方法可能会在计算上变得很困难。这就是核的概念的由来。核函数是一个函数k：$\mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$，它在原始（$p$维）空间中取两个点，并计算出一个数字（两个输入之间的相似性的量度）。有无数的内核函数，下面是几个流行的内核函数。

- 多项式内核：$k(\mathbf{x}_1, \mathbf{x}_2) = (m + \mathbf{x}_1 \cdot \mathbf{x}_2)^d$，其中$m \geq 0$和$d \in \mathbb{N}$是有待选择的超参数。
- 高斯核：$k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}_1 - \mathbf{x}_2\|^2\right)$，超参数$\sigma^2 > 0$。
- 拉普拉斯核：$k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{1}{\sigma}\|\mathbf{x}_1 - \mathbf{x}_2\|_1\right)$，超参数$b > 0$。

请注意，重要的是，上述所有内核都是原始空间中的数据的函数。让我们回到上一个问题中的特征转换。让$\mathbf{x}$，$\mathbf{x}^t$是原始空间中的任何两个点，也就是。

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \mathbf{x}^t = \begin{bmatrix} x_1^t \\ x_2^t \end{bmatrix}。$$

并回顾一下，为了运行一个线性分类器，我们需要所有转换点的点积，其形式为。

$$(\varphi(\mathbf{x}), \varphi(\mathbf{x}^t)) = \begin{pmatrix} 1 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ x_1^2 \\ \sqrt{2}x_1 x_2 \\ x_2^2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \sqrt{2}x_1^t \\ \sqrt{2}x_2^t \\ (x_1^t)^2 \\ \sqrt{2}x_1^t x_2^t \\ (x_2^t)^2 \end{pmatrix} = 1 + 2x_1 x_1^t + 2x_2 x_2^t + x_1^2 (x_1^t)^2 + x_2^2 (x_2^t)^2 + 2x_1 x_1^t x_2 x_2^t$$

这可以简单写成。

$$1 + 2x_1 x_1^t + 2x_2 x_2^t + x_1^2 (x_1^t)^2 + 2x_1 x_1^t x_2 x_2^t = (1 + (\mathbf{x}, \mathbf{x}^t))^2。$$

由此可以看出，点积实际上是对多项式内核的评估，其中包括
m=1，d=2，即

$$k(\mathbf{x}, \mathbf{x}^t) = (\varphi(\mathbf{x}), \varphi(\mathbf{x}^t))。$$

为什么这一点如此重要？那么，与其先计算$\varphi(\mathbf{x})$，然后再计算所有的平行点乘，上面的计算表明，我们只需要计算两个原始向量之间的核函数，而且这种计算相对来说要便宜得多，因为它只对原始空间中的点进行操作换句话说，我们需要做的是计算核的值，而不是进行漫长的点积计算。这一点是可行的，因为回顾一下，我们只需要访问点积进行学习，我们其实不需要明确计算特征转换，这就是所谓的内核技巧。事实上，我们甚至不必再考虑计算特征变换函数φ，我们只需选择一个内核，每个内核都会给我们一个相应的特征变换φ。这就产生了内核感知器，它只是上面定义的关于变换特征的双重感知器的重名，所以我们更新*第*i*个*权重，α_i，如果

$$\sum_{j=1}^{N} \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) < 0。$$

因此，所有的变化是现在我们指的是内核矩阵，它是所选内核k在原始数据上的配对评估。

$$\mathbf{K} := \begin{matrix} K(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ K(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{x}_n, \mathbf{x}_1) & K(\mathbf{x}_n, \mathbf{x}_2) & \cdots & K(\mathbf{x}_n, \mathbf{x}_n) \end{matrix},$$

等同于前一个问题中的$\mathbf{\Phi}$，但是相对于计算feature transforations然后取点积的天真方法来说，计算起来更便宜。回到我们为什么选择6维表示的问题上，现在应该很清楚，这样做是为了让我们能够回到内核的选择上。在实践中，如果我们对8度以下的多项式特征感兴趣，那么我们可以简单地选择一个d=8的多项式核。

问题4.（核子及其特征表示）。

~~在这个问题中，我们将展示内核的选择如何给我们带来不同的特征转换。请注意，在实践中，我们将简~~单地选择一个核，而不太关心确切的特征转换，但重要的是要知道不同的核对应于不同的表示。

(a) 让$\mathbf{x}$，$\mathbf{y}\in\mathbb{R}^2$（即$\mathbf{x}$和$\mathbf{y}$是二维向量），并考虑内核

$$k(\mathbf{x}, \mathbf{y}) = (2(\mathbf{x}, \mathbf{y})+3)^3。$$

计算与此核相关的特征向量$\varphi(\mathbf{x})$。(换句话说，$\mathbf{x}$和$\mathbf{y}$的特征代表，使$(\varphi(\mathbf{x}), \varphi(\mathbf{y})) = k(\mathbf{x}, \mathbf{y})$)

解决方案。

写下$\mathbf{x} = (x_1, x_2)^T$和$\mathbf{y} = (y_1, y_2)^T$那么

$$k(\mathbf{x}, \mathbf{y}) = (2(\mathbf{x}, \mathbf{y}) + 3)^3$$

$$= 8x_1^3 y_1^3 + 8x_2^3 y_2^3 + 24x_1^2 y_1^2 x_2 y_2 + 24x_1 y_1 x_2^2 y_2^2 + 36x_1^2 y_1^2 + 36x_2^2 y_2^2 + 72x_1 y_1 x_2 y_2 + 54x_1 y_1 + 54x_2 y_2 + 27$$

从这里，我们不难看出

$$\varphi^T(x) = [\sqrt{8}x_1^3, \sqrt{8}x_2^3, \sqrt{24}x_1^2 x_2, \sqrt{24}x_1 x_2^2, \sqrt{36}x_1^2, \sqrt{36}x_2^2, \sqrt{72}x_1 x_2, \sqrt{54}x_1, \sqrt{54}x_2, 27]$$

(b) **挑战。**让$x, y \in \mathbb{R}$，并考虑高斯核。

$$k(x, y) = \exp\left(-\frac{1}{2\sigma^2}(x - y)^2\right)。 \quad \sigma^2 > 0。$$

提示：使用泰勒展开法将指数改写为求和。

解决方案。

这个练习显示了内核技巧的全部威力，因为高斯内核允许我们计算出**无限**维度的特征表征！在这个练习中，我们可以看到，高斯内核是一个很好的例子。

$$k(x, y) = \exp\left(-\frac{1}{2\sigma^2}(x - y)^2\right) \quad \text{(高斯核的定义)}$$

$$= \exp\left(-\frac{x^2}{2\sigma^2}\right) \text{曝光} \left(-\frac{y^2}{2\sigma^2}\right) \text{泰勒扩展的exp} \left(\frac{xy}{\sigma^2}\right)$$

$$= \exp\left(-\frac{x^2}{2\sigma^2}\right) \text{曝光} \left(-\frac{y^2}{2\sigma^2}\right) \sum_{k=0}^{\infty} \frac{(xy)^k}{\sigma^{2k} k!}$$

$$= \exp\left(-\frac{x^2}{2\sigma^2}\right) \text{曝光} \left(-\frac{y^2}{2\sigma^2}\right) \sum_{k=0}^{\infty} \frac{x^k}{\sigma^k \sqrt{k!}} \frac{y^k}{\sigma^k \sqrt{k!}}$$

可以写成$\langle \varphi(x), \varphi(y) \rangle$，其中。

$$\varphi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \left[1, \frac{x}{\sigma\sqrt{1!}}, \frac{x^2}{\sigma^2\sqrt{2!}}, \frac{x^3}{\sigma^3\sqrt{3!}}, \ldots\right]^T,$$

所以我们看到，使用高斯核时的隐含特征表示是无限维的！这意味着通过在原始空间中计算简单的函数（通过核函数），我们能够得到强大的数据表示，而不必在无限维空间中进行任何计算。这意味着，通过在原始空间中计算简单的函数（通过核函数），我们能够得到强大的数据表示，而不需要在无限维空间中进行任何计算

请再次注意，在实践中，我们实际上从来不需要计算明确的特征向量，因为我们总是可以通过评估k来计算特征空间中的点乘，而且对于许多内核来说，这样的计算无论如何是不可能的。

问题5（更多关于内核的技巧）

你被告知，"内核技巧 "意味着可以从原始的非线性映射中实现

数据表示到一个新的、隐含的特征空间，只需在原始数据中的实例对的点积上定义一个核函数。要知道为什么会这样，你拿两个实例$\mathbf{x} = [1, 2]^T$和$\mathbf{y} = [3, 2]^T$，然后取它们的点积$\mathbf{x}, \mathbf{y}$，得到答案7。很明显，将这个点积提高到2的幂，将得到$(\mathbf{x}, \mathbf{y})^2 = 49$。现在把这个表达式展开，说明如果你只是对原始数据做了一组特征变换，这也是你会得到的答案。

解决方案。

在阅读这篇文章之前，如果你对特征表示和内核技巧感到困惑，请看一下后面的补充评论。在这个问题中，我们被告知，原始空间是$\mathbf{R}^2$，我们有两个点。

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix},$$

并进一步说明我们正在使用内核。

$$k(\mathbf{x}, \mathbf{y}) = (\varphi(\mathbf{x}), \varphi(\mathbf{y}))。$$

我们的目标是要弄清楚当我们利用这个内核时，我们选择的是什么特征表示（φ）。为了弄清这一点，请注意。

$$\begin{aligned} k(\mathbf{x}, \mathbf{y}) &= (\mathbf{x}, \mathbf{y})^2 \\ &= \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right)^2 \\ &= (x_1 y_1 + x_2 y_2)^2 \\ &= x_1^2 y_1^2 + x_2^2 y_2^2 + 2 x_1 y_1 x_2 y_2 \\ &= \left(\begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \end{bmatrix}, \begin{bmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2} y_1 y_2 \end{bmatrix} \right) \\ &= (\varphi(\mathbf{x}), \varphi(\mathbf{y}))。 \end{aligned}$$

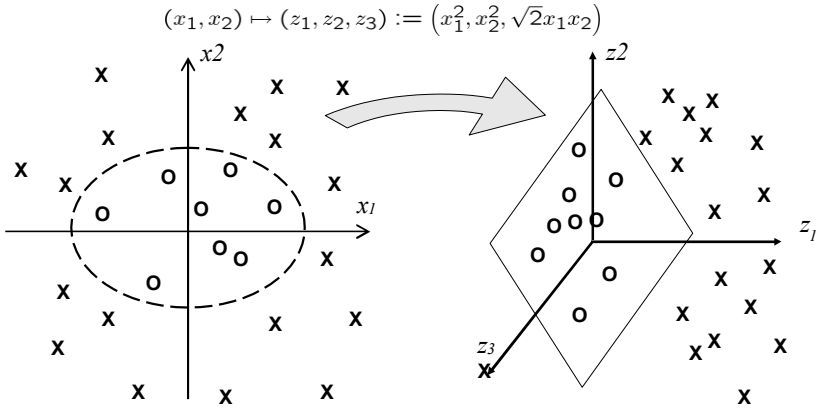
因此，使用内核$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}, \mathbf{y})^2$，相当于使用特征映射。

$$\varphi \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \end{bmatrix}.$$

问题6（更多特征转换）

考虑以下从二维空间（$\mathbf{R}^2$）到三维空间（$\mathbf{R}^3$）的特征转换的描述。为什么我们要使用这样的转换？

从原空间的类中产生一个例子，从原空间的类中产生另一个例子，并显示它们在新空间的转换值。



解决方案。

这种转换使我们能够使用线性分类器对数据进行正确分类。在原始空间中，这些点不是线性可分离的，在这里做分类的唯一方法是使用非线性模型，这可能很困难。对变换的巧妙选择使我们能够使用我们现有的工具进行线性分类，如perceptron。我们可以证明这一点

在两点上的变换，在○类中，考虑$x^{\circ} = (2, 2)$，在×类中考虑$x^{\times} = (2\sqrt{2}, 3\sqrt{2})$，然后用$\varphi$表示变换，其中

$$\varphi : \mathbb{R}^2 \to \mathbb{R}^3 。 \quad \varphi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)。$$

那么我们有

$$\varphi(x^{\circ}) = (2, 2, 2\sqrt{2})$$
$$\varphi(x^{\times}) = (8, 8, 18).$$

问题7（支持向量机）

支持向量机本质上是一种学习线性分类器的方法，但使用的目标函数与我们之前看到的方法不同，即*最大化边际*。这个问题的学习算法通常使用二次优化求解器，但是对于少量的支持向量来说，也有可能手动推导出解决方案。

这里有一个由三个例子组成的玩具数据集，显示为矩阵$\mathbf{X}$，其中前两个被分类为正数，第三个被分类为负数，显示为向量$\mathbf{y}$。首先构建这个数据的*格拉姆矩阵*，纳入类标签，即形成矩阵$\mathbf{X}^t\ (\mathbf{X}^t)^T$。然后求解，找到支持向量，它们的拉格朗日乘数$\alpha$，然后确定权重向量$\mathbf{w}$，阈值$t$和边际$m$。

$$\mathbf{X} = \begin{matrix} 3 & 1 \\ 1 & 1 \\ 0 & 1 \end{matrix} \qquad \mathbf{y} = \begin{matrix} +1 \\ +1 \\ -1 \end{matrix} \qquad \mathbf{X}^t = \begin{matrix} 1 & 3 \\ 2 & 1 \\ 0 & -1 \end{matrix}$$

要找到最大边际分类器，需要找到**w**、t和边际m的解决方案。为此，我们可以使用以下步骤（参考 "核方法 "讲座的幻灯片30-35）。

1. 为标记的数据设置格拉姆矩阵
2. 设置要被最小化的表达式
3. 取部分导数
4. 设为零，并对每个乘数进行求解
5. 求解**w**
6. 求解t
7. 求解m

解决方案。

回顾一下，计算SVM分类器等同于解决以下受限的优化问题。

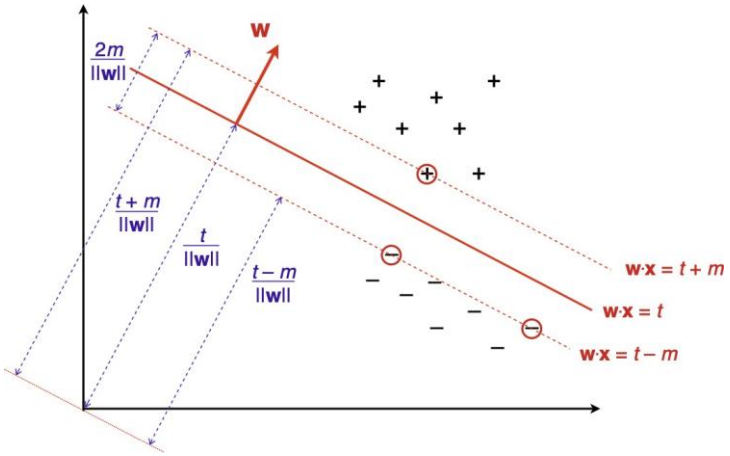
$$\arg\min_{\mathbf{w},t} \frac{1}{\|\mathbf{w}\|^2} \quad \text{受制于} y_i((\mathbf{x}_i, \mathbf{w}) - t) \geq 1 \text{ for } i = 1, \ldots, n.$$

解释这个表达式的方法是，我们正在寻找**w**，t，以便于。

1.我们通过让第一类的观察结果在一边的情况下对所有的点进行正确的分类。
线$(\mathbf{x}_i, \mathbf{w}) = t + 1$，第二类的点在线的另一侧
$(\mathbf{x}_i, \mathbf{w}) = t - 1$。在上述表达式中，通过要求$y_i((\mathbf{x}_i, \mathbf{w}) - t) \geq 1$，就可以照顾到这一点。
对于所有的i，它简洁地抓住了这两个声明。将此与感知器学习所需的较弱条件进行比较：$Y_i((\mathbf{x}_i, \mathbf{w}) - t) \geq 0$。

2.要求两条线$(\mathbf{x}_i, \mathbf{w}) = t \pm 1$之间的余量尽可能的 "肥"。回顾一下
这个保证金的宽度为 $\frac{1}{\|\mathbf{w}\|}$ 的宽度，我们希望将其最大化，或者说，将$\|\mathbf{w}\|$最小化。
或等价于最小化$\frac{1}{2}\|\mathbf{w}\|^2$。

下面的图描述了问题的几何形状（注意，我们取m=1！）。



接下来，讲座中显示，考虑对偶问题更为简单。

$$\arg\max_{\alpha_1,\ldots,\alpha_n} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j (\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{n}\alpha_i \quad \text{，条件是} \quad \sum_{i=1}^{n}\alpha_i y_i = 0。\ \alpha_i \geq 0 \text{ for } i = 1, \ldots, n.$$

在初始问题中，我们学习的是参数$\mathbf{w}$、t，而在对偶问题中，我们转而关注向量$\alpha = (\alpha_1, \ldots, \alpha_n)^T$。请注意，我们仍然可以通过关系从$\alpha$恢复初始参数。

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i,$$

而对于参数t，我们可以解决方程$y_i((\mathbf{w}, \mathbf{x}_i) - t) = 1$，对于任何$\mathbf{x}_i$上的决定边界（任何$\mathbf{x}_i$，是支持向量），即。

$$t = (\mathbf{w}, \mathbf{x}^i) - \frac{1}{y_i}.$$

下面的步骤解释了如何解决问题中提供的简单数据集的对偶问题。

1. 考虑到对偶问题中的第一项，我们看到，我们将需要$(\mathbf{x}_i, \mathbf{x}_j)$的条款，用于所有的$i, j$。设计矩阵是$\mathbf{x}_i$'s的矩阵，定义为

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} \in \mathbb{R}^{n \times p}$$

然后，格拉姆矩阵（所有成对点积的矩阵）是$\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{n\times n}$，在这种情况下，格拉姆矩阵的$(i, j)$个元素给了我们术语$\mathbf{x}_i$，$\mathbf{x}_j$。然而在这里，我们可以注意到，我们进一步需要术语$y_i\mathbf{x}_i$，$y_j\mathbf{x}_j$，我们可以通过定义增强的设计矩阵得到这些术语

$$\mathbf{X}^t = \begin{bmatrix} \mathbf{x}_1^T y_1 \\ \mathbf{x}_2^T y_2 \\ \vdots \\ \mathbf{x}_n^T y_n \end{bmatrix} \in \mathbb{R}^{n\times p}。$$

和增强的格拉姆矩阵$\mathbf{G}^t$，为。

$$\mathbf{g}^t \equiv (\mathbf{x}^t)(\mathbf{x}^t)^T = \begin{bmatrix} 10 & 5 & -3 \\ 5 & 5 & -1 \\ -3 & -1 & 1 \end{bmatrix}$$

这一步只是为了表明，在计算SVM时，我们需要的只是矩阵$\mathbf{G}$，而不再需要随身携带$\mathbf{X}$。为了更清楚地看到这一点，我们可以将对偶问题重写为：。

$$\text{最大值}_{\alpha_1,\ldots,\alpha_n} \frac{-1}{2}\sum_{\substack{i=1\\ j=1}}^{n}\sum^{n} \alpha_i \mathbf{G}^t[i,j] + \sum_{i=1}^{n}\alpha_i \quad \text{受制于} \sum_{i=1}^{n}\alpha_i y_i = 0, \quad \alpha_i \geq \text{对于} i = 1, \ldots, n.$$

2. 因此，双重优化问题是

$$\arg\max_{\alpha_1,\alpha_2,\alpha_3} \frac{1}{2}\left(10\alpha_1^2 + 10\alpha_2^2 + 6\alpha_1\alpha_2 - 6\alpha_1\alpha_3 + 5\alpha_2\alpha_3 - 2\alpha_2\alpha_3\right) + \alpha_1 + \alpha_2 + \alpha_3$$

受 $\alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_3 \geq 0$ 和 $\alpha_1 + \alpha_2 - \alpha_3 = 0$. 这里，我们注意到，由于$\alpha_1 + \alpha_2 - \alpha_3 = 0 \Longrightarrow \alpha_3 = (\alpha_1 + \alpha_2)$，我们可以相应地替换$\alpha_3$的每一次出现，并将我们的问题简化为只对$\alpha_1$和$\alpha_2$进行最大化。

$$\arg\max_{\alpha_1,\alpha_2} \frac{-1}{2}\left(10\alpha_1^2 + 10\alpha_1\alpha_2 - 6\alpha_1(\alpha_1+\alpha_2) + 5\alpha_2^2 - 2\alpha_2(\alpha_1+\alpha_2)\right) + (\alpha_1+\alpha_2)$$

$$= \arg\max_{\alpha_1,\alpha_2} \frac{-1}{2}\left(5\alpha_1^2 + 4\alpha_1\alpha_2 + 2\alpha_2^2\right) + 2\alpha_1 + 2\alpha_2$$

3. 计算关于α的偏导数α_1, α_2。

$$\frac{\partial}{\partial\alpha_1} = -5\alpha_1 - 2\alpha_2 + 2 \qquad \frac{\partial}{\partial\alpha_2} = -2\alpha_1 - 4\alpha_2 + 2$$

4. 将偏导数设为零并求解，得到$\alpha_1 = \frac{1}{4}$，$\alpha_2 = \frac{3}{8}$。另外，由于$\alpha_3 = \alpha_1 + \alpha_2$ 我们有$\alpha_3 = \frac{5}{8}$ 这里要注意，由于$\alpha_i > 0$，这三个点都是支持向量。这是直观的，因为我们的数据集是如此之小。一般来说，只有支持向量（边缘上的点）实际上会有一个非零的α_i项。在这个意义上，α_i捕捉到的是重要的

的第i个点用于学习模型。请注意，深藏在各自类内的点相对来说并不重要，因为如果我们对最接近类之间的边界的点进行了正确的分类，我们也总是对深藏在类内的点进行了正确的分类，因此它们对模型的贡献是零。

5.从幻灯片30：$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$

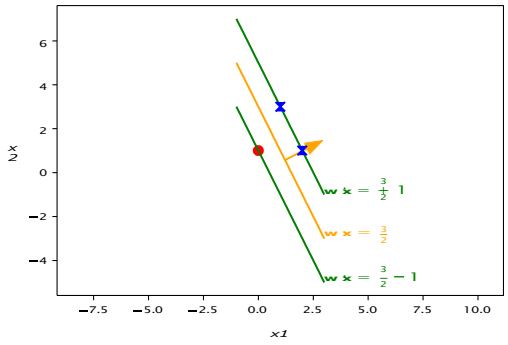
所以

$$\mathbf{w} = \frac{1}{4}\mathbf{x}_1 + \frac{3}{8}\mathbf{x}_2 - \frac{5}{8}\mathbf{x}_3$$

$$1/2$$

6.t可以从任何支持向量，例如$\mathbf{x}$，得到，因为$y(\mathbf{w}, \mathbf{x})-t)=1$；这使得$t=\frac{3}{2}$。

请注意，一般来说，支持向量是那些$\alpha_i \neq 0$的点$\mathbf{x}_i$。我们现在可以将我们所学的模型形象化。



作为一个扩展，考虑添加一个 "深入正类 "的点，例如，$\mathbf{x}_4 = (10, 10)$, $y_4 = +1$。你应该看到，这将增加零的信息，因此我们应该得到一个相同的模型，并且$\alpha_4 = 0$。

7.最后，保证金为：$1/l$ $\mathbf{wl} = 1/(1 + (\ )^2)^{\frac{1}{2}} = \frac{4}{5} = \frac{2}{5}$.