

支持向量网络

郭富士

南京航空航天大学

日期：2020 年 2 月 19 日

摘 要

支持向量网络是一种针对二分类问题的新学习机。从概念上实现了以下想法：输入向量被非线性映射到一个非常高维的特征空间，并在该特征空间中，构造一个线性决策面，此决策面的特殊性质保证了学习机有较强的泛化能力。支持向量网络背后的想法是在很有限的情况下实现的，即训练数据可以无误差地被分割。我们将在这里将结果扩展到不可分割的训练数据。

我们证明了利用多项式输入变换后支持向量网络的强泛化能力。并且将支持向量网络的性能与各种经典学习算法进行了比较，所有这些算法都参与了光学字符识别的基础研究。

关键词：模式识别，高效学习算法，神经网络，径向基函数分类器，多项式分类器。

1 简介

六十多年前 R.A.Fisher (Fisher, 1936) 提出了第一个模式识别算法。他考虑了 n 维向量 \mathbf{x} 的一个包含两个正态分布， $N(\mathbf{m}_1, \Sigma_1)$ 和 $N(\mathbf{m}_2, \Sigma_2)$ 的模型，其中均值向量为 \mathbf{m}_1 和 \mathbf{m}_2 ，以及协方差矩阵为 Σ_1 和 Σ_2 ，并证明了最优（贝叶斯）解是一个二次决策函数：

$$F_{sq}(\mathbf{x}) = \text{sign} \left[\frac{1}{2} (\mathbf{x} - \mathbf{m}_1)^T \Sigma_1^{-1} (\mathbf{x} - \mathbf{m}_1) - \frac{1}{2} (\mathbf{x} - \mathbf{m}_2)^T \Sigma_2^{-1} (\mathbf{x} - \mathbf{m}_2) + \ln \frac{|\Sigma_2|}{|\Sigma_1|} \right] \quad (1)$$

在 $\Sigma_1 = \Sigma_2 = \Sigma$ 的情况下，二次决策函数 (1) 退化为线性函数：

$$F_{lin}(\mathbf{x}) = \text{sign} \left[(\mathbf{m}_1 - \mathbf{m}_2)^T \Sigma^{-1} \mathbf{x} - \frac{1}{2} (\mathbf{m}_1^T \Sigma^{-1} \mathbf{m}_1 - \mathbf{m}_2^T \Sigma^{-1} \mathbf{m}_2) \right] \quad (2)$$

为了估计二次决策函数，必须确定 $\frac{n(n+3)}{2}$ 个自由参数；估计线性函数，仅需确定 n 个自由参数。在观测次数少（例如小于 $10n^2$ ）的情况下，估计 $o(n^2)$ 数量的参数是不现实的。因此，Fisher 建议，即使在 $\Sigma_1 \neq \Sigma_2$ 的情况下，也应使用线性决策函数 (2)，其中 Σ 为以下形式：

$$\Sigma = \tau \Sigma_1 + (1 - \tau) \Sigma_2 \quad (3)$$

其中 τ 是某个常数¹。对于两个分布都不为正态分布的情况，Fisher 还是建议使用线性决策函数。因为用于模式识别的算法从一开始就与线性决策面的构造相关。

1962 年，Rosenblatt (Rosenblatt, 1962) 探索了另一种学习机：感知机或神经网络。感知机由连接的神经元组成，其中每个神经元都实现了一个分离的超平面，因此感知机在整体上实现了一个分段的线性分离平面，见图 1。在 Rosenblatt 的时代，并没有找到一种通过调整网络的所有权重来最小化向量集误差的算法，而 Rosenblatt 提出了一种只有输出单元的权重为自适应的

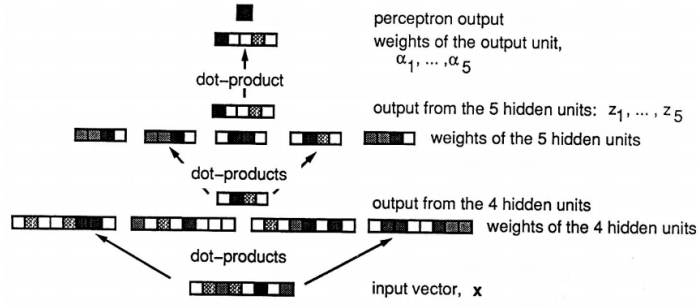


图 1: 具有 8 个输入单元, 2 层隐藏单元和 1 个输出单元的简单前反馈感知机。向量中的灰色阴影反映了它们的数值。

方法。根据其他权重的设置, 将输入向量非线性变换为最后一层单元的特征空间 Z 。在这个空间中, 构造了线性决策函数:

$$I(\mathbf{x}) = \text{sign} \left(\sum_i \alpha_i z_i(\mathbf{x}) \right) \quad (4)$$

通过对第 i 个隐藏单元到输出单元的权重 α_i 进行调整, 以最小化训练数据上的一些误差。作为 Rosenblatt 方法的结果, 决策函数的构造再次与一些空间中线性超平面的构造产生了联系。

1986 年发现了一种算法, 可以对神经网络的所有权重进行局部调整, 以最大程度地减小属于模式识别问题的一组向量上的误差 (Rumelhart, Hinton & Williams, 1986, 1987; Parker, 1985; Phil, 1985; LeCun, 1985), 且在当时反向传播算法已被发现。该解决方法涉及了对神经元数学模型的轻微修改, 从而神经网络实现了“分段线性型”决策函数。

在本文中, 我们构造了一种新型的学习机, 即所谓的支持向量网络。它实现了以下想法: 使用先验选择的一些非线性映射, 将输入向量映射到一些高维特征空间 Z 中, 并构造具有特殊性质的线性决策面, 确保网络的强泛化能力。

例子: 为了获得与二阶多项式相对应的决策面, 可以构造特征空间 Z , 有 $N = \frac{n(n+3)}{2}$ 个坐标:

$$\begin{aligned} z_1 &= x_1, \dots, z_n = x_n, & n & \text{ coordinates,} \\ z_{n+1} &= x_1^2, \dots, z_{2n} = x_n^2, & n & \text{ coordinates,} \\ z_{2n+1} &= x_1 x_2, \dots, z_N = x_n x_{n-1}, & \frac{n(n-1)}{2} & \text{ coordinates,} \end{aligned}$$

其中 $x = (x_1, \dots, x_n)$ 。超平面就在这个空间中所构造。

上述方法有两个问题: 一个是概念上的, 另一个是技术上的。概念上的问题是如何找到一个可以很好地泛化的分离超平面: 特征空间的维数将是巨大的, 而且并非所有分离训练数据的超平面都能够很好地泛化²。技术上的问题是如何在计算上处理此类高维空间: 要在 200 维空间中构造 4 或 5 阶多项式, 可能需要在十亿维特征空间中构造超平面。对于此类问题, 概念部分于 1965 年得到解决 (Vapnik, 1982)。此处将最优超平面定义为线性决策函数, 在两类向量之间具有最大间距, 见图 2。并观察到, 构造此类最优超平面时, 只需考虑少量训练数据即可, 即所谓的确该间隔的支持向量。结果表明, 如果训练向量被最优超平面无误差地分离, 则在测试

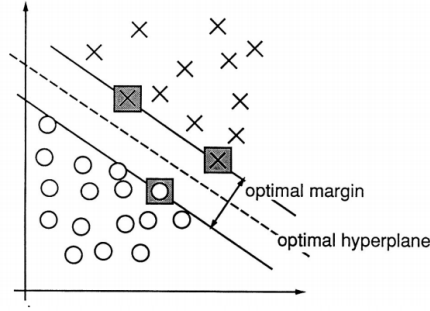


图 2: 二维空间中可分离问题的一个示例。用灰色正方形标记的支持向量定义了两个类别之间最大的间隔。

示例上犯错的概率的期望值将受到支持向量数量的期望值与训练向量数量之间比例的限制：

$$E[\text{Pr}(\text{error})] \leq \frac{E[\text{number of support vectors}]}{\text{number of training vectors}} \quad (5)$$

请注意，此边界未明确包含间隔空间的维数。然后得出结论：如果可以从相对于训练集大小的少数支持向量构造最优超平面，那么即使在无限维空间中，泛化能力也会很强。在第 5 节中，我们将证明现实问题的比例 (5) 可以低至 0.03，并且最优超平面可以在十亿维空间中很好地泛化。使：

$$\mathbf{w}_0 \mathbf{z} + b_0 = 0$$

为特征空间中的最优超平面。我们将表明特征空间中的最优超平面的权重 \mathbf{w}_0 可以写作一些支持向量的线性组合

$$\mathbf{w}_0 = \sum_{\text{support vectors}} \alpha_i \mathbf{z}_i \quad (6)$$

特征空间中的线性决策函数 $I(\mathbf{z})$ 可以写作以下形式：

$$I(\mathbf{z}) = \left(\sum_{\text{support vectors}} \alpha_i \mathbf{z}_i \mathbf{z} + b_0 \right) \quad (7)$$

其中 $\mathbf{z}_i \mathbf{z}$ 是支持向量 \mathbf{z}_i 和向量 \mathbf{z} 在特征空间中的点积。因此，决策函数可以描述为两层网络（图 3）。然而，即使最优超平面能够很好地泛化，但如何处理高维特征空间的技术问题仍然存在。

在 1992 年 (Boser, Guyon, & Vapnik, 1992) 证明了可以交换构造决策函数的操作顺序，以代替对输入向量进行非线性变换，然后在特征空间中对带有支持向量的点积进行变换，可以先比较输入空间中的两个向量（例如，通过对它们的点积或某种距离度量进行比较），然后对结果的值进行非线性变换（见图 4）。这使得我们能够构造多种类型的决策面，例如任意阶的多项式决策面。我们将这种学习机称为支持向量网络³。支持向量网络方法最初是针对在没有误差的情况下分离训练数据的有限情况而开发的。在本文中，我们扩展了支持向量网络的用途，以涵盖不可能没有误差地分离训练数据的情况。通过这种扩展，我们将支持向量网络视为一种新的学习机，与神经网络一样强大且通用。在第 5 节中，我们将展示它的高维空间（256 维）中对高阶多项式决策面（至多 7 阶）的泛化效果有多好。并将算法的性能与经典学习机的性能进行比较，例如线性分类器、 k -近邻分类器和神经网络。第 2、3 和 4 节专门介绍算法的推导要点，并讨论其一些特性。派生的详细信息参见附录。

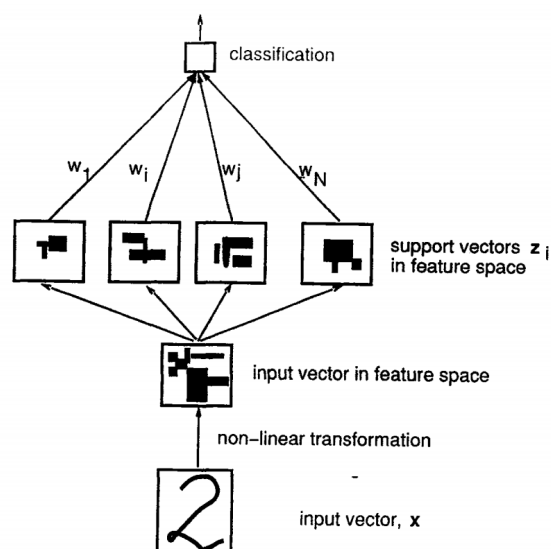


图 3: 从概念上说, 通过支持向量网络对未知模式进行分类是通过先将模式变换为一些高维特征空间来完成的。在此特征空间中构造的最优超平面决定了输出。与图 1 相比, 可以看到两层感知机之间的相似性。

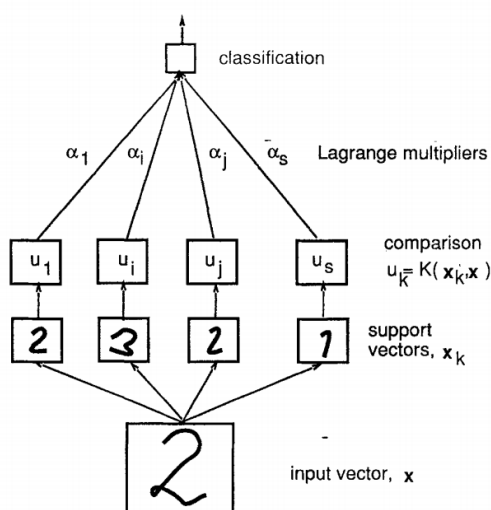


图 4: 支持向量网络对未知模式的分类。与支持向量相比, 该模式在输入空间中, 结果的值被非线性变换。这些变换值的线性函数确定分类器的输出。

2 最优超平面

在本节中，我们将回顾在无误差地分离训练数据的任务中构造最优超平面的方法（Vapnik, 1982）。在下一节中，我们将介绍软间隔的概念，这会允许对学习存在误差的训练集进行分析处理。

2.1 最优超平面算法

标记过的训练模式集

$$(y_1, \mathbf{x}_1), \dots, (y_\ell, \mathbf{x}_\ell), \quad y_i \in \{-1, 1\} \quad (8)$$

被称为线性可分的，如果存在一个向量 \mathbf{w} 和一个标量 b 使得以下不等式

$$\begin{aligned} \mathbf{w}\mathbf{x}_i + b &\geq 1 & \text{if } y_i = 1, \\ \mathbf{w}\mathbf{x}_i + b &\leq -1 & \text{if } y_i = -1, \end{aligned} \quad (9)$$

对训练集 (8) 的所有元素成立。下面我们将 (9) 写成另一种形式⁴：

$$y_i(\mathbf{w}\mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, \ell. \quad (10)$$

超平面

$$\mathbf{w}_0\mathbf{x} + b_0 = 0 \quad (11)$$

是唯一一个以最大间隔分割训练数据的：回想一下图 2，它确定了方向 $\mathbf{w}/|\mathbf{w}|$ ，使得两个不同类别训练向量的投影之间的距离最大。距离 $\rho(\mathbf{w}, b)$ 由以下公式给出：

$$\rho(\mathbf{w}, b) = \min_{\{\mathbf{x}; y=1\}} \frac{\mathbf{x}\mathbf{w}}{|\mathbf{w}|} - \max_{\{\mathbf{x}; y=-1\}} \frac{\mathbf{x}\mathbf{w}}{|\mathbf{w}|}. \quad (12)$$

最优超平面 (\mathbf{w}_0, b_0) 是使距离 (12) 最大化的参数。由 (10) 和 (12) 得到

$$\rho(\mathbf{w}_0, b_0) = \frac{2}{|\mathbf{w}_0|} = \frac{2}{\sqrt{\mathbf{w}_0\mathbf{w}_0}}. \quad (13)$$

这意味着最优超平面是在约束 (10) 下最小化 $\mathbf{w}\mathbf{w}$ 的唯一平面。因此，构造最优超平面成为了一个二次规划问题。

$y_i(\mathbf{w}\mathbf{x}_i + b) = 1$ 中的向量 \mathbf{x}_i 称为支持向量。在附录 A.1 中我们表明了确定最优超平面的向量 \mathbf{w}_0 可以表示为训练向量的线性组合：

$$\mathbf{w}_0 = \sum_{i=1}^{\ell} y_i \alpha_i^0 \mathbf{x}_i \quad (14)$$

其中 $\alpha_i^0 \geq 0$ 。由于 $\alpha > 0$ 仅用于支持向量（见附录），所以表达式 (14) 呈现了 \mathbf{w}_0 的紧凑形式。

我们还展示了参数 α_i 的向量：

$$\mathbf{\Lambda}_0^T = (\alpha_1^0, \dots, \alpha_\ell^0),$$

必须解决以下的二次规划问题：

$$W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T \mathbf{1} - \frac{1}{2} \mathbf{\Lambda}^T \mathbf{D} \mathbf{\Lambda} \quad (15)$$

其中 $\mathbf{\Lambda}^T = (\alpha_1, \dots, \alpha_\ell)$ ，并受约束：

$$\mathbf{\Lambda} \geq \mathbf{0}, \quad (16)$$

$$\mathbf{\Lambda}^T \mathbf{Y} = \mathbf{0}, \quad (17)$$

其中 $\mathbf{1}^T = (1, \dots, 1)$ 是 ℓ -维单位向量, $\mathbf{Y}^T = (y_1, \dots, y_\ell)$ 是 ℓ -维标签向量, \mathbf{D} 是对称的 $\ell \times \ell$ 矩阵, 元素为

$$D_{ij} = y_i y_j \mathbf{x}_i \mathbf{x}_j, \quad i, j = 1, \dots, \ell. \quad (18)$$

不等式 (16) 表明了非负区域。因此, 在 (17) 的约束下, 我们必须在非负区域中最大化二次型 (15)。

当训练数据 (8) 可以无误差地分割时, 我们在附录 A 中展示函数 (15) 的最大值, $(\mathbf{\Lambda}_0, b_0)$ 和 (13) 中的最大间隔 ρ_0 有如下关系:

$$W(\mathbf{\Lambda}_0) = \frac{2}{\rho_0^2}. \quad (19)$$

如果对于一些 $\mathbf{\Lambda}_*$ 和大常数 W_0 不等式

$$W(\mathbf{\Lambda}_*) > W_0 \quad (20)$$

成立, 那么可以断言, 所有将训练数据 (8) 分割的超平面都具有间隔

$$\rho < \sqrt{\frac{2}{W_0}}.$$

如果训练集 (8) 无法通过超平面分割, 那么两类模式之间的间隔会变得任意小, 从而导致函数 $W(\mathbf{\Lambda})$ 的值变得任意大。因此, 在约束 (16) 和 (17) 下使函数 (15) 最大化, 一个要么达到最大值 (在这种情况下, 可以用最大间隔 ρ_0 构造超平面), 要么发现最大值超过了某个给定的 (大) 常数 W_0 (在这种情况下, 不可能以大于 $\sqrt{2/W_0}$ 的间隔分割训练数据)。

利用以下方法可以非常有效地解决在约束 (16) 和 (17) 下最大化函数 (15) 的问题。将训练数据划分为多个部分, 每个部分中的训练向量数应尽可能少。首先解决由训练数据的第一部分确定的二次规划问题。对于此问题, 有两种可能的结果: 要么无法通过超平面分割数据的这一部分 (在这种情况下, 也无法分割整个数据集), 或者用于分割训练数据的第一部分的最优超平面被找到。

令在能够分割第一部分的情况下使函数 (15) 最大化的向量为 $\mathbf{\Lambda}_1$ 。 $\mathbf{\Lambda}_1$ 的一些坐标等于 0。他们对应于该部分的非支持训练向量。生成一组新的训练数据, 其中包含来自训练数据第一部分的支持向量和不满足约束 (10) 的第二部分向量, 其中 \mathbf{w} 由 $\mathbf{\Lambda}_1$ 确定。对于此集合, 构造一个新函数 $W_2(\mathbf{\Lambda})$ 并使之在 $\mathbf{\Lambda}_2$ 处最大化。继续逐步构造覆盖训练数据所有部分的解向量 $\mathbf{\Lambda}_*$ 的过程, 要么发现不可能无误差地分割训练集, 要么构造出针对整个数据集地最优分割超平面 $\mathbf{\Lambda}_* = \mathbf{\Lambda}_0$ 。注意, 在此过程中, 函数 $W(\mathbf{\Lambda})$ 的值单调增加, 因为在优化过程中考虑了越来越多的训练向量, 导致两类之间的间隔越来越小。

3 软间隔超平面

考虑不可能无误差地分割训练数据的情况。在这种情况下, 可能希望以最少的误差数来分割训练集。为了正式展示这一点, 让我们介绍一些非负变量 $\xi \geq 0, i = 1, \dots, \ell$ 。

现在我们可以最小化函数

$$\Phi(\xi) = \sum_{i=1}^{\ell} \xi_i^\sigma \quad (21)$$

对于小 $\sigma > 0$ ，受约束

$$y_i(\mathbf{w}\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, \ell, \quad (22)$$

$$\xi_i \geq 0, \quad i = 1, \dots, \ell. \quad (23)$$

如果 σ 足够小，函数 (21) 则描述了训练误差的数量⁵。

最小化 (21) 可以找到一些训练误差的最小子集：

$$(y_{i_1}, \mathbf{x}_{i_1}), \dots, (y_{i_k}, \mathbf{x}_{i_k}).$$

如果将这些数据从训练集中排除，则可以无误差地将训练集中的其余部分分割。为了分割训练数据的其余部分，可以构造最优分割超平面。

这个想法可以正式表示为：最小化函数

$$\frac{1}{2}\mathbf{w}^2 + CF\left(\sum_{i=1}^{\ell}\xi_i^{\sigma}\right) \quad (24)$$

受约束 (22) 和 (23)，其中 $F(u)$ 是单调凸函数， C 是常数。

对于足够大的 C 和足够小的 σ ，在约束 (22) 和 (23) 下将函数 (24) 最小化的向量 \mathbf{w}_0 和常数 b_0 ，确定了使训练集上的误差数最小化的超平面，并用最大间隔分割了其余部分元素。

但是请注意，构造一个使训练集上的误差数量最小化的问题通常是 NP -完全的。为了避免问题的 NP -完全性，我们将考虑 $\sigma = 1$ 的情况（优化问题 (15) 具有唯一解的 σ 的最小值）。在这种情况下，函数 (24) 描述了（对于足够大的 C 而言）构造分割超平面的问题，该分割超平面最小化了训练误差的偏差之和，并最大化了正确分类的向量的间隔。如果可以无误差地分割训练数据，则构造的超平面与最优分割超平面重合。

与 $\sigma < 1$ 的情况相反，存在一种在 $\sigma = 1$ 的情况下找到 (24) 的解的有效方法。我们称之为软间隔超平面。

在附录 A 中我们考虑了最小化函数的问题

$$\frac{1}{2}\mathbf{w}^2 + CF\left(\sum_{i=1}^{\ell}\xi_i\right) \quad (25)$$

受约束 (22) 和 (23)，其中 $F(u)$ 是单调凸函数且 $F(0) = 0$ 。为了简化公式，我们在本节中仅描述 $F(u) = u^2$ 的情况。对于此函数，优化问题仍然是二次规划问题。

在附录 A 中，我们表明，对于最优超平面算法，向量 \mathbf{w} 可以写成支持向量 \mathbf{x} 的线性组合：

$$\mathbf{w}_0 = \sum_{i=1}^{\ell} \alpha_i^0 y_i \mathbf{x}_i.$$

为了找到向量 $\mathbf{\Lambda}^T = (\alpha_1, \dots, \alpha_{\ell})$ ，必须解决最大化对偶二次规划问题

$$W(\mathbf{\Lambda}, \delta) = \mathbf{\Lambda}^T \mathbf{1} - \frac{1}{2} \left[\mathbf{\Lambda}^T \mathbf{D} \mathbf{\Lambda} + \frac{\delta^2}{C} \right] \quad (26)$$

受约束

$$\mathbf{\Lambda}^T \mathbf{Y} = 0, \quad (27)$$

$$\delta \geq 0, \quad (28)$$

$$\mathbf{0} \leq \mathbf{\Lambda} \leq \delta \mathbf{1}, \quad (29)$$

其中 $\mathbf{1}$, $\mathbf{\Lambda}$, \mathbf{Y} , 和 \mathbf{D} 与用于构造最优超平面的优化问题中使用的元素相同， δ 是标量，并且 (29)

描述了向量不等式。

注意，(29) 表明函数 (26) 中允许的最小 δ 值为

$$\delta = \alpha_{\max} = \max(\alpha_1, \dots, \alpha_\ell).$$

因此，为了找到一个软间隔分类器，必须找到一个向量 $\mathbf{\Lambda}$ 最大化

$$W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T \mathbf{1} - \frac{1}{2} \left[\mathbf{\Lambda}^T \mathbf{D} \mathbf{\Lambda} + \frac{\alpha_{\max}^2}{C} \right] \quad (30)$$

受约束 $\mathbf{\Lambda} > \mathbf{0}$ 和 (27) 的约束。该问题与仅通过函数 (30) 中具有最大值的余项来构造最优间隔分类器的问题不同。因此，构造软间隔分类器的问题的解决方法是唯一的，并且对任意数据集都适用。

由于带有 α_{\max} 的余项，所以函数 (30) 不是二次的。在约束 $\mathbf{\Lambda} > \mathbf{0}$ 和 (27) 的情况下最大化 (30)，属于所谓的凸规划问题。因此，构造一个软间隔分类器既可以解决参数 $\mathbf{\Lambda}$ 的 ℓ -维空间中的凸规划问题，也可以解决 $\mathbf{\Lambda}$ 和 S 的对偶 $\ell + 1$ 空间的二次规划问题。我们的实验通过解决对偶二次规划问题构造了软间隔超平面。

4 点积在特征空间中的卷积方法

前面各节中描述的算法在输入空间中构造了超平面。要在特征空间中构造超平面，首先必须通过选择 N -维向量函数 ϕ 将 n -维输入向量 \mathbf{x} 变换为 N -维特征向量：

$$\phi : \mathbb{R}^n \rightarrow \mathbb{R}^N.$$

然后为这组变换向量构造一个 N -维线性分割 \mathbf{w} 和一个偏置 b

$$\phi(\mathbf{x}_i) = \phi_1(\mathbf{x}_i), \phi_2(\mathbf{x}_i), \dots, \phi_N(\mathbf{x}_i), \quad i = 1, \dots, \ell$$

未知向量 \mathbf{x} 的分类是通过首先将向量变换到分割空间 ($\mathbf{x} \mapsto \phi(\mathbf{x})$)，然后取函数的符号来完成的

$$f(\mathbf{x}) = \mathbf{w}\phi(\mathbf{x}) + b. \quad (31)$$

根据软间隔分类器方法的性质，向量 \mathbf{w} 可以表示为支持向量的线性组合（在特征空间中）。这意味着

$$\mathbf{w} = \sum_{i=1}^{\ell} y_i \alpha_i \phi(\mathbf{x}_i) \quad (32)$$

点积的线性性质表明，(31) 中未知向量 \mathbf{x} 的分类函数 f 只与点积有关：

$$f(\mathbf{x}) = \phi(\mathbf{x})\mathbf{w} + b = \sum_{i=1}^{\ell} y_i \alpha_i \phi(\mathbf{x})\phi(\mathbf{x}_i) + b. \quad (33)$$

构造支持向量网络的想法来自于，考虑 Hilbert 空间中点积的一般形式 (Anderson & Bahadur, 1966)：

$$\phi(\mathbf{u})\phi(\mathbf{v}) \equiv K(\mathbf{u}, \mathbf{v}) \quad (34)$$

根据 Hilbert-Schmidt 理论 (Courant & Hilbert, 1953)，任何对称函数 $K(\mathbf{u}, \mathbf{v}) \in L_2$ ，都可以展开为

$$K(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{u})\phi_i(\mathbf{v}), \quad (35)$$

其中 $\lambda_i \in \Re$ 和 ϕ_i 是特征值和特征函数

$$\int K(\mathbf{u}, \mathbf{v}) \phi_i(\mathbf{u}) d\mathbf{u} = \lambda_i \phi_i(\mathbf{v}).$$

在积分下被内核 $K(\mathbf{u}, \mathbf{v})$ 所定义。能确保 (34) 定义了特征空间中的一种点积的充分条件是，在展开式 (35) 中所有特征值都是正的。为了确保这些系数都是正的，以下是充分必要条件 (Mercer 's 理论)，即

$$\int \int K(\mathbf{u}, \mathbf{v}) g(\mathbf{u}) g(\mathbf{v}) d\mathbf{u} d\mathbf{v} > 0$$

对所有这样的 g 均满足

$$\int g^2(\mathbf{u}) d\mathbf{u} < \infty.$$

因此，满足 Mercer 's 理论的函数可以被用作点积。Aizerman, Braverman 和 Rozonoer (1964) 考虑了特征空间中点积的一种卷积，卷积函数为

$$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{|\mathbf{u} - \mathbf{v}|}{\sigma}\right), \quad (36)$$

被他们称为势函数。

然而，特征空间中点积的卷积可以被任意满足 Mercer' s 调节的函数给出；特别地，为了在 n 维输入空间中构造 d 阶多项式分类器，可以使用以下函数

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{v} + 1)^d. \quad (37)$$

使用不同的点积 $K(\mathbf{u}, \mathbf{v})$ 能够构造有任意类型决策面的不同的学习机 (Boser, Guyon & Vapnik, 1992)。这些学习机的决策面有这样的形式

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i),$$

其中 \mathbf{x}_i 是支持向量在输入空间中的像，且 α_i 是支持向量在特征空间中的权重。

为了找到向量 \mathbf{x}_i 和权重 α_i ，可以遵循相同的关于原最优间隔分类器或软间隔分类器的解决方法。唯一的差别是矩阵 D (被 (18) 确定)，另一个则使用矩阵

$$D_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \quad i, j = 1, \dots, l.$$

5 支持向量网络的一般特性

5.1 通过支持向量网络构造决策规则是高效的

为了构造支持向量网络决策规则，必须解决以下二次最优化问题：

$$W(\Lambda, \delta) = \Lambda^T \mathbf{1} - \frac{1}{2} \left[\Lambda^T D \Lambda + \frac{\delta^2}{C} \right]$$

受简单的约束：

$$\mathbf{0} \leq \Lambda \leq \delta \mathbf{1},$$

$$\Lambda^T \mathbf{Y} = 0,$$

其中矩阵

$$D_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \quad i, j = 1, \dots, l.$$

被训练集的元素所定义，且 $K(\mathbf{u}, \mathbf{v})$ 是决定点积的卷积函数。

通过解决由训练数据确定的当前构成支持向量的中间优化问题，可以有效地找到优化问题的解决方法。第 3 节中介绍了此方法。获得的最优决策函数是唯一的⁶。

可以使用任意标准技巧来解决任意优化问题。

5.2 支持向量网络是一种通用机

通过更改决定点积的卷积函数 $K(\mathbf{u}, \mathbf{v})$ 可以获得不同的网络。

在下一节中我们将考虑使用多项式决策面的支持向量网络。为了指明不同阶的多项式，可以使用下列点积的卷积函数

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}\mathbf{v} + 1)^d.$$

径向基函数机有决策函数

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^n \exp \left\{ -\frac{|\mathbf{x} - \mathbf{x}_i|^2}{\sigma^2} \right\} \right)$$

可以使用以下的卷积函数实现

$$K(\mathbf{u}, \mathbf{v}) = \exp \left\{ -\frac{|\mathbf{u} - \mathbf{v}|^2}{\sigma^2} \right\}.$$

在这种情况下，支持向量网络机将构造出近似函数的中心 \mathbf{x}_i 和权重 α_i 。

通过构造特殊的卷积函数，还可以结合当前问题的先验知识。因此，支持向量网络是一种相当通用的学习机，可以很简单地通过改变点积形式来改变决策函数集。

5.3 支持向量网络与泛化能力控制

要控制学习机的泛化能力，必须控制两个不同的因子：训练数据的误差率和通过 VC-维所测量的学习机的能力 (Vapnik, 1982)。在测试集上存在的误差率有以下形式的界限：有 $1 - \eta$ 的概率，不等式

$$\text{Pr}(\text{test error}) \leq \text{Frequency}(\text{training error}) + \text{Confidence Interval} \quad (38)$$

成立。在界限 (38) 中，置信区间取决于学习机的 VC-维，训练集中的元素数量以及 η 的值。

(38) 中的两个因子形成了一种相互制约：学习机函数集的 VC-维越小，置信区间越小，但错误频率的值越大。

解决这种相互制约的一般方法被提出为结构风险最小化原理：对于给定的数据集，必须找到一种使其总和最小化的解决方法。结构风险最小化原理的一种特殊情况是奥卡姆剃刀原理：保持第一项为零，并最小化第二项。

已知线性指标函数集的 VC-维

$$I(\mathbf{x}) = \text{sign}(\mathbf{w}\mathbf{x} + b), \quad |\mathbf{x}| \leq C_{\mathbf{x}}$$

其中固定阈值 b 等于输入空间维数。然而，子集的 VC-维

$$I(\mathbf{x}) = \text{sign}(\mathbf{w}\mathbf{x} + b), \quad |\mathbf{x}| \leq C \quad |\mathbf{w}| \leq C_{\mathbf{w}}$$

(具有权重的有界函数的函数集) 可以小于输入空间的维数，并取决于 $C_{\mathbf{w}}$ 。

从这个角度看，最优间隔分类器遵循了奥卡姆剃刀原理。他使 (38) 的第一项为零（通过满足不等式 (9)），并且使第二项最小化（通过使函数 $\mathbf{w}\mathbf{w}$ 最小化）。这种最小化可以防止过拟合的

问题。

然而，即使在训练数据是可分割的情况下，也可以通过使 (38) 中的置信度最小化来获得更好的泛化能力，甚至以训练集的误差为代价。在软间隔分类器方法中，则可以通过选择适当的参数 C 来完成。在支持向量网络算法中，即使在更普遍的情况，训练集上不存在零误差的解决方法，也可以通过更改参数 C 来权衡决策规则的复杂性和误差率。因此，支持向量网络可以控制学习机泛化能力的两个因子。

6 实验分析

为了演示支持向量网络方法，我们进行了两种类型的实验。我们在平面上构造认为生成的模式集，并使用二阶多项式决策面进行实验，再针对数字识别的实际问题进行实验。

6.1 平面实验

利用以下形式的点积

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}\mathbf{v} + 1)^d \quad (39)$$

$d = 2$ ，我们为平面中的不同模式集构造决策规则。这些实验的结果可以可视化，并且可以很好地说明算法的功能。示例如图 5 所示。这两个类别分别由黑白符号表示。再图中，我们用圆圈表示支持模式，而用交叉表示误差。在不存在误差叫做小的二阶多项式的情况下，该解决方法是最优的。请注意，相对于训练模式的数量，支持模式的数量很小。

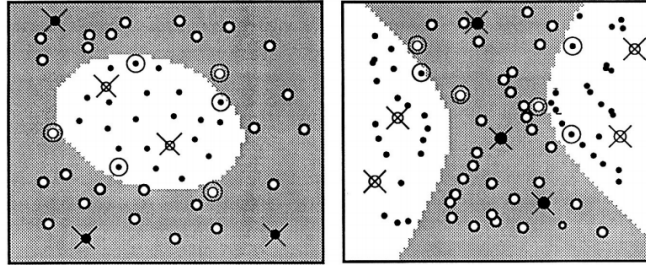


图 5: $d = 2$ 时点积 (39) 的示例。圆圈表明支持模式，交叉表明误差。

6.2 数字识别实验

我们构造支持向量网络的实验利用了两个不同的数据库进行位图数字识别，这是一个小型数据库和大型数据库。较小的是美国邮政服务数据库，其中包含 7300 种训练模式和 2000 种测试模型。该数据库的分辨率为 16×16 像素，一些典型的示例如图 6 所示。在该数据库上，我们进行了各阶多项式的实验研究。大型数据库包含 60000 个训练模式和 10000 个测试模式，并且是 NIST⁷ 训练集和测试集的 50-50 混合。这些模式的分辨率为 28×28 ，产生的输入维度为 784。在此数据库上，我们仅构造了 4 次多项式分类器。将该分类器的性能与参加基准研究的其他类型的学习机进行比较 (Bottou, 1994)。

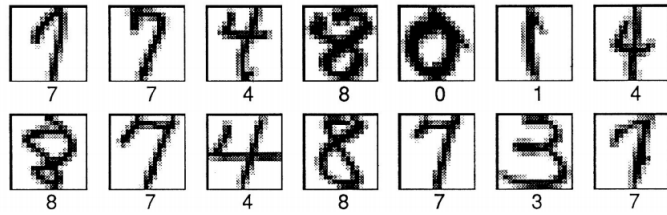


图 6: 美国邮政服务数字数据库中带有标签的模式示例。

在我们所有的实验中，构造了十个分割，每个类别一个。每个超平面都使用相同的点积和数据预处理。根据这十个分类器的最大输出完成未知模式的分类。

6.2.1 美国邮政服务数据库实验

美国邮政服务数据库使用了真实的邮件，该数据库已被另外一些研究人员报告过。在表 1 中，我们列出了从出版物的和自己实验中收集的各种分类器的性能数据。J. Bromley & E. Sackinger (Bromley & Sackinger, 1991) 报告了人类的表现结果。CART 的结果由新泽西州 Murray Hill, 贝尔实验室的 Daryl Pregibon 和 Michale D. Riley 获得。本文中，由 Corinna Cortes 和 Bernard Schoelkopf 分别获得了 C4.5 和最优 2 层神经网络（最优隐藏单元数）的结果。Y. LeCun 等人获得了具有 5 层专用神经网络架构 LeNet1 的结果（1990）。在使用美国邮政服务数据库进行的实

表 1: 来自出版物和自己实验的各种分类器的性能数据。参考资料。

Classifier	Raw error,%
Human performance	2.5
Decision tree, CART	17
Decision tree, C4.5	17
Best 2 layer neural network	6.6
Special architecture 5 layer network	5.1

验中，我们使用了预处理（居中，倾斜和平滑），以结合目前有关问题不变性的了解。在 (Boser, Guyon & Vapnik, 1992) 中研究了平滑数据库作为支持向量网络的预处理的影响。对于实验，我们选择高斯平滑内核，标准差 $\sigma = 0.75$ ，与 (Boser, Guyon & Vapnik, 1992) 一致。

基于形式 (39) 的点积。输入维数为 256，多项式的阶数范围为 1 到 7。表 2 呈现了实验结果，训练数据不是线性可分的。请注意，支持向量的数目增加得非常缓慢。7 阶多项式得支持向量仅比 3 阶多项式多 30%，甚至少于 1 阶多项式。然而，用于 7 阶多项式分类器的特征空间维数比用于 3 阶多项式分类器的大 1010 倍。再次注意，性能几乎不会随着空间维数的增加而改变，这说明没有过拟合的问题。

线性分割的支持向量数量相对较高是由于其不可分割性：数字 200 包含了支持向量和具有非零 ξ -值的训练向量。如果 $\xi > 1$ 的训练向量被错误分类；线性情况下，训练集上的错误分类次数为平均 34 次每分类器。对于 2 阶分类器，训练集上的错误分类次数降低到 4。这 4 种模式如图 7 所示。值得注意的是，在我们所有的实验中，当我们考虑获得的支持向量数量而不是该数量的期望值时，泛化能力的上限 (5) 成立。在所有情况下，单个分类器的错误概率上限不超过

表 2: 不同阶数的多项式的点积获得的结果。“支持向量”的数量是每个分类器的平均值。

Degree of polynomial	Raw error, %	Support vectors	Dimensionality of feature space
1	12.0	200	256
2	4.7	127	~ 33000
3	4.3	148	$\sim 1 \times 10^6$
4	4.3	175	$\sim 1 \times 10^9$
5	4.3	175	$\sim 1 \times 10^{12}$
6	4.2	185	$\sim 1 \times 10^{14}$
7	4.3	190	$\sim 1 \times 10^{16}$

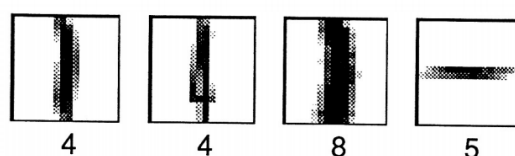


图 7: 标注的 2 阶多项式支持向量分类器在训练集上出错的示例。

3% (在测试数据上, 单个分类器的实际误差不超过 1.5%)。

构造多项式分类器的训练时间不取决于多项式的阶数, 而仅取决于支持向量的数量。即使在最坏的情况下, 它也比性能最好的神经网络 LeNet1 更快 (LeCun 等, 1990)。该神经网络的误差为 5.1%。阶数 2 或更高的多项式优于 LeNet1。

6.2.2 NIST 数据库实验

NIST 数据库仅用于进行了 2 周的基准研究。在有限的时间范围内, 只能构造出一种类型的分类器, 因此我们选择了没有预处理的 4 阶多项式。这种选择基于我们在美国邮政数据库中的经验。

表 3 列出了 10 个分类器种每个分类器的支持向量的数量, 并给出了分类器在训练集和测试集上的性能。请注意, 即使是 4 阶多项式 (具有超过 108 个自由参数) 在此训练集上也会出错。每个类别的平均训练误差率为 0.02% ~ 12。图 8 显示了分类器 1 的 14 个分类错误的测试模式。再次注意, 对于获得的支持向量数量, 上限 (5) 是如何成立的。

表 3: 在 NIST 数据库上针对 4 阶多项式分类器获得的结果。训练集的大小为 60000, 测试集的大小为 10000 模式。

	C1.0	C1.1	C1.2	C1.3	C1.4	C1.5	C1.6	C1.7	C1.8	C1.9
Supp.patt.	1379	989	1958	1900	1224	2024	1527	2064	2332	2765
Error train	7	16	8	11	2	4	8	16	4	1
Error test	19	14	35	35	36	49	32	43	48	63

测试集上 10 个分类器的综合性能为 1.1% 误差率。该结果应与基准研究种其他参与分类器的结果进行比较。这些其他分类器包括一个线性分类器, 一个具有 60000 原型的 $k = 3$ 最近邻分

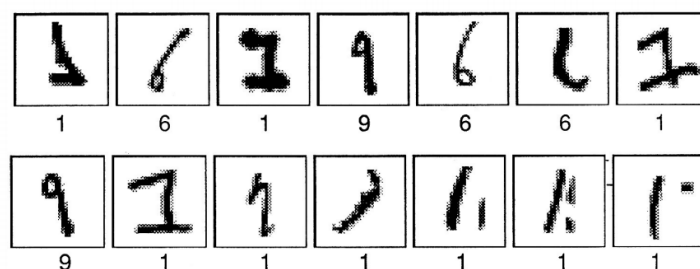


图 8: 分类器 1 分类错误的 14 个带有标签的测试模式。带有标签“1”的模式为假阴性，其他的为假阳性。

类器，以及两个专门为数字识别而构建的神经网络 (LeNet1 和 LeNet4)。作者仅为支持向量网络的结果做出了贡献。基准测试结果如图 9 所示。我们通过引用描述基准测试结果的论文 (Bottou

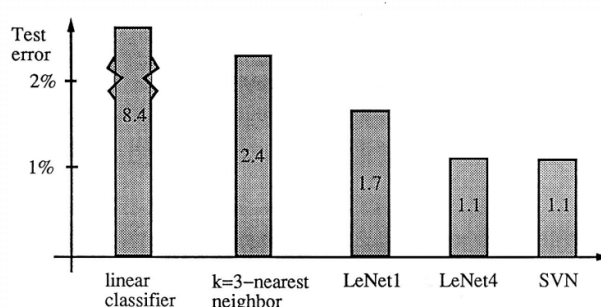


图 9: 来自基准研究的结果。

等, 1994) 来结束本节:

长期以来, LeNet1 一致被认为是最新技术..., 通过一系列体系结构实验, 结合对识别错误特征的分析, 精心设计了 LeNet4...

支持向量网络具有出色的准确性, 最值得注意的是, 因为与其他分类器不同, 他不包括有关问题几何结构的认知。实际上, 如果图像像素被加密 (例如通过固定的随机的排列), 分类器也将做到同样的事情。

最后一句话表明, 可以通过构建点积 $K(\mathbf{u}, \mathbf{v})$ 的函数来反映支持向量的性能, 这些函数反映了有关当前问题的先验信息。

7 总结

这篇论文介绍了作为二分类问题的新学习机—支持向量网络。

支持向量网络结合了 3 个想法: 来自最优超平面的求解方法 (允许在支持向量上扩展解向量), 点积的卷积想法 (将求解表面从线性扩展到非线性) 和软间隔的概念 (以考虑训练集上的误差)。

该算法已经经过测试, 并与其他经典算法的性能进行了比较。尽管决策面的设计十分简单, 但新算法在对比研究中仍表现出非常好的性能。

其他功能 (如容量控制和易于更改已构造的决策面) 使支持向量网络成为功能强大且通用的学习机。

A 构造分割超平面

在本附录中，我们推导构造最优超平面和软间隔超平面的方法。

A.1 最优超平面算法

在第 2 节中表明，构造最优超平面

$$\mathbf{w}_0 \mathbf{x} + b = 0 \quad (40)$$

分割训练数据集

$$(y_1, \mathbf{x}_1), \dots, (y_\ell, \mathbf{x}_\ell),$$

必须最小化函数

$$\Phi = \mathbf{w} \mathbf{w},$$

受约束

$$y_i(\mathbf{x}_i \mathbf{w} + b) \geq 1, \quad i = 1, \dots, \ell. \quad (41)$$

为此，我们使用标准的优化方法。我们构造一个拉格朗日函数

$$L(\mathbf{w}, b, \Lambda) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{\ell} \alpha_i [y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1], \quad (42)$$

其中 $\Lambda = (\alpha_1, \dots, \alpha_\ell)$ 对应约束 (41) 的非负拉格朗日乘子。

众所周知，最优化问题的结果由 \mathbf{w} 、 Λ 和 b 的 $2\ell + 1$ -维空间中的拉格朗日方程的鞍点确定，其中应相对于参数 \mathbf{w} 和 b 取最小值，并应相对对拉格朗日乘子 Λ 取最大值。

在最小值（关于 \mathbf{w} 和 b ）的点上，获得：

$$\left. \frac{\partial L(\mathbf{w}, b, \Lambda)}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_0} = \left(\mathbf{w}_0 - \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i \right) = 0, \quad (43)$$

$$\left. \frac{\partial L(\mathbf{w}, b, \Lambda)}{\partial b} \right|_{b=b_0} = \sum_{\alpha_i} y_i \alpha_i = 0. \quad (44)$$

从等式 (43) 我们得到

$$\mathbf{w}_0 = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i, \quad (45)$$

这表明最优超平面可以写成训练向量的线性组合。请注意，只有 $\alpha_i > 0$ 的训练向量 \mathbf{x} 对总和 (45) 有贡献。

将 (45) 和 (42) 代入 (42) 我们得到

$$W(\Lambda) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \mathbf{w}_0 \cdot \mathbf{w}_0 \quad (46)$$

$$= \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j. \quad (47)$$

可以用向量重写为

$$W(\Lambda) = \Lambda^T \mathbf{1} - \frac{1}{2} \Lambda^T \mathbf{D} \Lambda \quad (48)$$

其中 $\mathbf{1}$ 为 ℓ -维单位向量, \mathbf{D} 是对称 $\ell \times \ell$ -矩阵, 元素为

$$D_{ij} = y_i y_j \mathbf{x}_i \mathbf{x}_j.$$

为了找到所需的鞍点, 仍然需要在约束 (43) 下找到 (48) 的最大值

$$\mathbf{\Lambda}^T \mathbf{Y} = 0,$$

其中 $\mathbf{Y}^T = (y_1, \dots, y_\ell)$, 并且

$$\mathbf{\Lambda} \geq 0.$$

Kuhn-Tucker 定理在优化理论中起着重要的作用。根据该定理, 在 \mathbf{w}_0 , $\mathbf{\Lambda}_0$ 和 b_0 的鞍点处, 任何拉格朗日乘子 α_i^0 及其对应的约束均通过等式连接起来

$$\alpha_i [y_i (\mathbf{x}_i \cdot \mathbf{w}_0 + b_0) - 1] = 0, \quad i = 1, \dots, \ell.$$

通过等式得出非零值 α_i 仅在以下情况实现

$$y_i (\mathbf{x}_i \cdot \mathbf{w}_0 + b_0) - 1 = 0.$$

换句话说: $\alpha_i \neq 0$ 仅在等号成立的情况下成立。我们称向量 \mathbf{x}_i 满足

$$y_i (\mathbf{x}_i \cdot \mathbf{w}_0 + b_0) = 1$$

时为支持向量。注意, 在此术语中, 等式 (45) 指出, 解向量 \mathbf{w}_0 可以用支持向量所表达。

另一个基于 Kuhn-Tucker 的观察。等式 (44) 和 (45) 的最优解, 是最大值 $W(\mathbf{\Lambda}_0)$ 和分割距离 ρ_0 之间的联系:

$$\mathbf{w}_0 \cdot \mathbf{w}_0 = \sum_{i=1}^{\ell} \alpha_i^0 y_i \mathbf{x}_i \cdot \mathbf{w}_0 = \sum_{i=1}^{\ell} \alpha_i^0 (1 - y_i b_0) = \sum_{i=1}^{\ell} \alpha_i^0.$$

将等式代入到 (46) 我们得到 $W(\mathbf{\Lambda}_0)$

$$W(\mathbf{\Lambda}_0) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \mathbf{w}_0 \cdot \mathbf{w}_0 = \frac{\mathbf{w} \cdot \mathbf{w}}{2}.$$

再代入到第 2 节中的表达式 (13) 我们得到

$$W(\mathbf{\Lambda}_0) = \frac{2}{\rho_0^2},$$

其中 ρ_0 是最优超平面的间隔。

A.2 软间隔超平面算法

接下来我们先考虑 $F(u) = u^k$ 的情况。然后我们描述单调凸函数 $F(u)$ 的一般结果。

为了构造软间隔分割超平面, 我们最大化函数

$$\Phi = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \left(\sum_{i=1}^{\ell} \xi_i \right)^k, \quad k > 1,$$

受约束

$$y_i (\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \xi_i, \quad i = 1, \dots, \ell, \quad (49)$$

$$\xi_i \geq 0, \quad i = 1, \dots, \ell. \quad (50)$$

此问题的拉格朗日函数为

$$L(\mathbf{w}, \xi, b, \mathbf{\Lambda}, \mathbf{R}) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \left(\sum_{i=1}^{\ell} \xi_i \right)^k - \sum_{i=1}^{\ell} \alpha_i [y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i] - \sum_{i=1}^{\ell} r_i \xi_i, \quad (51)$$

其中，非负乘子 $\mathbf{\Lambda}^T = (\alpha_1, \alpha_2, \dots, \alpha_\ell)$ 由约束 (49) 产生，乘子 $\mathbf{R}^T = (r_1, r_2, \dots, r_\ell)$ 由约束 (50) 产生。

我们需要找到函数的鞍点（对应于变量 \mathbf{w}_i , b 和 ξ_i 取最小值，对应于变量 α_i 和 r_i 取最大值）。

让我们在极值点使用使其函数最小的条件：

$$\left. \frac{\partial L}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_0} = \mathbf{w}_0 - \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i = 0, \quad (52)$$

$$\left. \frac{\partial L}{\partial b} \right|_{b=b_0} = \sum_{i=1}^{\ell} \alpha_i y_i = 0, \quad (53)$$

$$\left. \frac{\partial L}{\partial \xi_i} \right| = kC \left(\sum_{i=1}^{\ell} \xi_i^0 \right)^{k-1} - \alpha_i - r_i. \quad (54)$$

如果我们令

$$\sum_{i=1}^{\ell} \xi_i^0 = \left(\frac{\delta}{Ck} \right)^{\frac{1}{k-1}}, \quad (55)$$

我们可以把等式 (54) 重写为

$$\delta - \alpha_i - r_i = 0. \quad (56)$$

从等式 (52)-(55) 我们得到

$$\begin{aligned} \mathbf{w}_0 &= \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i, \\ \sum_{i=1}^{\ell} \alpha_i y_i &= 0, \end{aligned} \quad (57)$$

$$\delta = \alpha_i + r_i. \quad (58)$$

将表达式 \mathbf{w}_0 , b_0 和 δ 代入到拉格朗日函数 (51) 我们得到

$$W(\mathbf{\Lambda}, \delta) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j - \frac{\delta^{k/k-1}}{(kC)^{1/k-1}} \left(1 - \frac{1}{k} \right). \quad (59)$$

为了找到软间隔超平面解，必须在约束 (57)-(58) 的条件下，相对于非负变量 α_i , r_i , $i = 1, \dots, \ell$ 最大化函数 (59)。用向量方法将 (59) 重写为

$$W(\mathbf{\Lambda}, \delta) = \mathbf{\Lambda}^T \mathbf{1} - \left[\frac{1}{2} \mathbf{\Lambda}^T \mathbf{D} \mathbf{\Lambda} + \frac{\delta^{k/k-1}}{(kC)^{1/k-1}} \left(1 - \frac{1}{k} \right) \right]. \quad (60)$$

其中 $\mathbf{\Lambda}$ 和 \mathbf{D} 在上面已被定义。因此，为了找到所需的鞍点，必须最大化 (60)，受约束

$$\mathbf{\Lambda}^T \mathbf{Y} = 0, \quad (61)$$

$$\mathbf{\Lambda} + \mathbf{R} = \delta \mathbf{1}, \quad (62)$$

$$\mathbf{\Lambda} \geq \mathbf{0}, \quad (63)$$

和

$$\mathbf{R} \geq \mathbf{0}. \quad (64)$$

从 (62) 和 (64) 我们得到向量 $\mathbf{\Lambda}$ 满足条件

$$\mathbf{0} \leq \mathbf{\Lambda} \leq \delta \mathbf{1}. \quad (65)$$

从条件 (62) 和 (64)，我们还可以得出结论来最大化 (60)

$$\delta = \alpha_{\max} = \max(\alpha_1, \dots, \alpha_\ell).$$

将 δ 的值代入 (60)，我们得到

$$W(\mathbf{\Lambda}, \delta) = \mathbf{\Lambda}^T \mathbf{1} - \left[\frac{1}{2} \mathbf{\Lambda}^T \mathbf{D} \mathbf{\Lambda} + \frac{\alpha_{\max}^{k/k-1}}{(kC)^{1/k-1}} \left(1 - \frac{1}{k} \right) \right]. \quad (66)$$

因此，要找到软间隔超平面，要么可以在约束 (61) 和 (65) 下找到二次型 (51) 的最大值，要么必须在约束 (61) 和 (56) 下找到凸函数 (60) 的最大值。对于本文的实验，我们使用 $k = 2$ ，并解决了二次优化问题 (51)。

在 $F(u) = u$ 的情况下，同样的方法使我们必须解决以下的二次优化问题：最小化函数

$$W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T \mathbf{1} - \frac{1}{2} \mathbf{\Lambda}^T \mathbf{D} \mathbf{\Lambda},$$

受约束

$$\mathbf{0} \leq \mathbf{\Lambda} \leq C \mathbf{1}$$

和

$$\mathbf{\Lambda}^T \mathbf{Y} = 0.$$

单调凸函数 $F(u)$ 的一般解也能从该方法中取得。软间隔超平面有一种形式

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

其中 $\mathbf{\Lambda}_0^T = (\alpha_1^0, \dots, \alpha_\ell^0)$ 是以下对偶凸优化问题的解：最大化函数

$$W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T \mathbf{1} - \left[\frac{1}{2} \mathbf{\Lambda}^T \mathbf{D} \mathbf{\Lambda} + \left(\alpha_{\max} f^{-1} \left(\frac{\alpha_{\max}}{C} \right) \right) - CF \left(f^{-1} \left(\frac{\alpha_{\max}}{C} \right) \right) \right],$$

受约束

$$\mathbf{\Lambda}^T \mathbf{Y} = 0,$$

$$\mathbf{\Lambda} \geq \mathbf{0},$$

我们记

$$f(u) = F'(u).$$

对于 $F(0) = 0$ 的单调凸函数 $F(u)$ ，以下不等式成立：

$$uF'(u) > F(u).$$

因此，方括号中的第二项是整数，且在 α_{\max} 趋于无穷大时趋于无穷大。

最后，我们可以考虑最小化表达式的超平面

$$\frac{1}{2} \left(\mathbf{w} \cdot \mathbf{w} + \sum_{i=1}^{\ell} \xi_i^2 \right)$$

受约束 (49)-(50)。其中，第二项最小化误差的最小平方差。这又引出以下二次规划问题：最大化函数

$$W(\Lambda) = \Lambda^T \mathbf{1} - \frac{1}{2} \left[\Lambda^T \mathbf{D} \Lambda + \frac{1}{C} \Lambda^T \Lambda \right]$$

条件为非负区域 $\Lambda \geq \mathbf{0}$ ，且受约束 $\Lambda^T \mathbf{Y} = 0$ 。

Notes

1. τ 的最优值在 70 年代被发现 (Anderson & Bahadur, 1966)。
2. 回想一下 Fisher 对少量数据和二次判别函数的关注。
3. 用这个名字，我们强调在支持向量上扩展解的思想对于这些学习机有多么关键。在支持向量学习算法中，构造的复杂性不取决于特征空间的维数，而取决于支持向量的数量。
4. 注意在不等式 (9) 和 (10) 中的被归一化的右侧，而不是 \mathbf{w} 。
5. 这里将训练误差定义为不等式 (22) 保持 $\xi > 0$ 。
6. 决策函数是唯一的，但不是在支持向量上的扩展。
7. 美国国家标准技术研究院特殊数据 3。