机器学习讲义(L11-A)：支持向量机

授课教师：王贝伦 / 助教：张嘉琦，黄旭，谈笑，徐浩卿

支持向量机（support vector machines, SVM）是一种二类分类模型。它的基本模型是定义在特征空间上的获取间隔最大的线性分类器，间隔最大使它有别于感知机；支持向量机还包括核技巧，这使它成为实质上的非线性分类器。支持向量机的学习策略就是间隔最大化，可形式化为一个求解凸二次规划（convex quadratic programming）的问题。支持向量机的学习算法是求解凸二次规划的最优化算法。
在 1994 年，SVM 被用于对手写数字进行分类，取得了良好的效果，自此广为人知，现如今已被广泛运用于计算机视觉，文本分类，时间序列分析等领域。

1 线性分类器

在机器学习的分类问题中，线性分类器通过对特征的线性组合来做出分类决定。线性分类器种类繁多，本节内容介绍的线性可分支持向量机也是其中之一。
我们现在给定一个数据集，其中的数据点分属于两个不同的类，我们需要找到一个线性分类器将其分类。该线性分类器的学习目标便是在样本所属的维度空间内找到一个超平面（hyperplane）将数据点划分为两类，见图 1 所示。

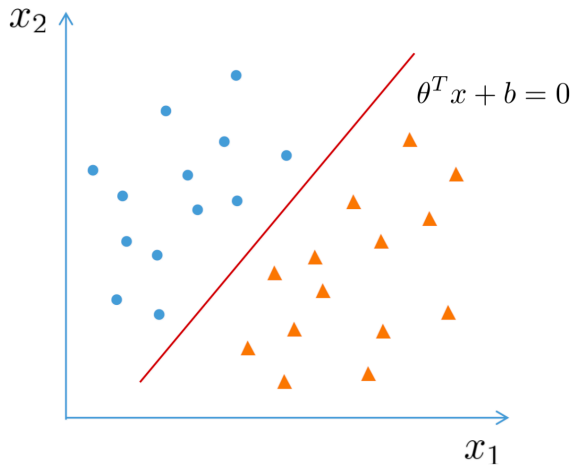

Figure 1: 线性分类器

在 L8 我们已经学习过二分类的常用算法——感知机（perceptron）。感知机是二分类的线性分类模型，它通过基于误分类的损失函数进行优化。所以在一定情况上，感知机的解不是唯一的，譬如在损失函数为 0 的情况下，超平面的选择就变得十分多元化，会产生无穷多个解，对应的划分超平面也有无穷多个，如图 2。

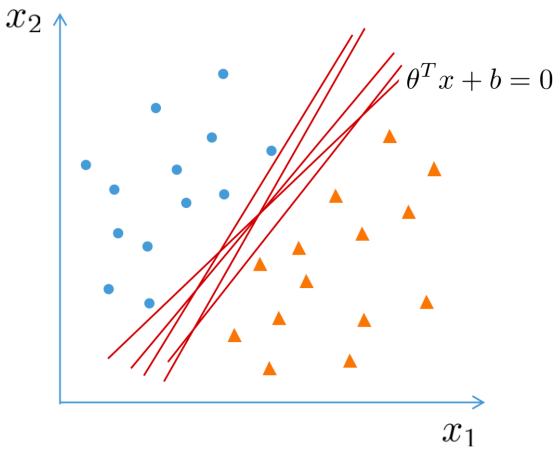

Figure 2: 超平面选择的多样性

如图 3 所示，在这种情况下，我们可以得到众多可能的超平面。其中，绿线离两类数据点都有着一定的距离，这意味着绿线的泛化能力可能更好，也是我们更希望看到的情况。但是由于感知机解的多样，由感知机求得的超平面可能如红线所示，在这种情况下，被黄色圆

© 本课件仅限东南大学机器学习课程使用，版权属于相关作者，严禁传播。

圈圈起来的点如果作为一个新输入的实例点，用红线作为分离的超平面来预测这个点时会判断其属于 △。这显然不是我们想要的，我们更希望得到的结果是绿线这样的分离超平面。

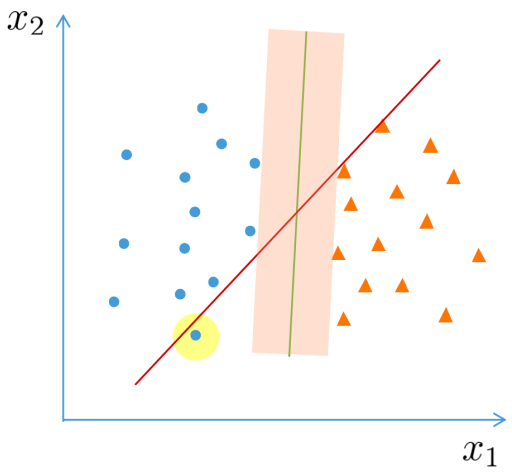

Figure 3: 感知机存在的问题

由图 3 的例子可以直观感受到，即使使用相同的训练集，感知机也会产生不同的模型，从而得到不同的预测结果。而我们希望能得到一个可以进行更好分类的超平面，所以我们提出了泛化性更强的模型，即线性可分支持向量机，**线性可分支持向量机在将两类数据正确划分的的同时使两类数据点间隔最大**。

2 线性可分类支持向量机与硬间隔最大化

2.1 函数间隔与几何间隔

一般来说，一个点距离分离超平面的远近可以表示分类预测的确信程度，离分离超平面越远其确信程度越大。在超平面、$\boldsymbol{\theta}^\top \mathbf{x} + b = 0$ 确定的情况下，$|\boldsymbol{\theta}^\top \mathbf{x} + b|$ 能够相对地表示点 $\mathbf{x}$ 距离超平面的远近。而 $\boldsymbol{\theta}^\top \mathbf{x} + b$ 的符号与类标记 y 的符号是否一致能够表示分类是否正确。所以可用量 $y(\boldsymbol{\theta}^\top \mathbf{x} + b)$ 来表示分类的正确性及确信度，这就是**函数间隔**（functional margin）的概念。

定义：函数间隔

对于给定的训练数据集和超平面 $\boldsymbol{\theta}^\top \mathbf{x} + b$，定义超平面 $\boldsymbol{\theta}^\top \mathbf{x} + b$ 关于样本点 $(\mathbf{x}, y)$ 的函数间隔为

$$\widehat{\gamma} = y(\boldsymbol{\theta}^\top \mathbf{x} + b) \tag{1}$$

函数间隔可以表示分类预测的正确性及确信度。但是选择分离超平面时，只有函数间隔还不够。因为只要成比例地改变 $\boldsymbol{\theta}$ 和 b，例如将它们改为 $2\boldsymbol{\theta}$ 和 $2b$，超平面并没有改变，但是函数间隔却变成原来的 2 倍。这一事实启示我们，可以对分离超平面的法向量 $\boldsymbol{\theta}$ 加某些约束，如规范化，$\|\boldsymbol{\theta}\| = 1$，使得间隔是确定的。这时函数间隔成为**几何间隔**（geometric margin）。

定义：几何间隔

对于给定的训练数据集和超平面 $\boldsymbol{\theta}^\top \mathbf{x} + b$，定义超平面 $(\boldsymbol{\theta}, b)$ 关于样本点 $(\mathbf{x}, y)$ 的几何间隔为

$$\gamma = y\left(\frac{\boldsymbol{\theta}^\top}{\|\boldsymbol{\theta}\|}\mathbf{x} + \frac{b}{\|\boldsymbol{\theta}\|}\right) \tag{2}$$

© 本课件仅限东南大学机器学习课程使用，版权属于相关作者，严禁传播。

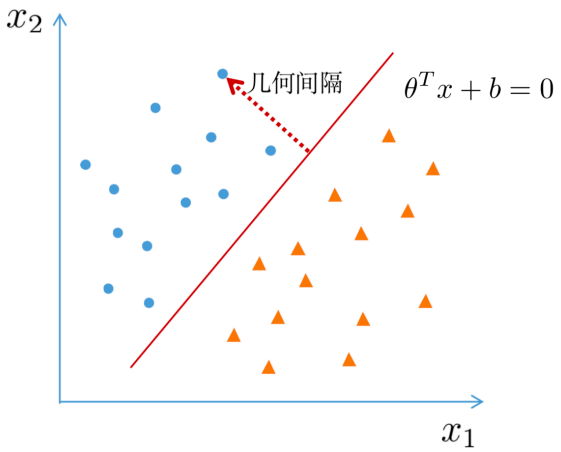

Figure 4: 几何间隔

从定义中的 (1)(2) 式就可以得出以下关系：

$$\gamma_i = \frac{\hat{\gamma}_i}{\|\boldsymbol{\theta}\|} \tag{3}$$

如果 $\|\boldsymbol{\theta}\| = 1$，那么函数间隔和几何间隔相等，改变 $\boldsymbol{\theta}$ 和 b 的值时候，函数间隔会成比例改变，而几何间隔不变。

2.2 间隔最大化

支持向量机学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。对线性可分的训练数据集来说，线性可分分离超平面有无穷多个，但是几何间隔最大的分离超平面却是唯一的。这里的间隔最大化又称为**硬间隔最大化**。硬间隔表示的是，对于训练集中任何一个样本，均可以准确划分；软间隔则允许一定量的样本分类错误。

间隔最大化的直观解释是：**对训练数据集找到几何间隔最大的超平面意味着以充分大的确信度对训练数据进行分类**。我们已知在离分离超平面越远，其确信度越大。间隔最大化意味着对于分离超平面最近的点也有足够大的确信度将其分开。这样的超平面应对未知的新实例有着很好的分类预测能力。

注意，我们这里强调是硬间隔最大化，因为与其相对的还有**软间隔（soft margin），硬间隔（hard margin）只适用于数据是完全线性可分的情况，当数据集本身存在一点噪声时，硬间隔最大化将失效，如图 5**。软间隔最大化由此应运而生，我们以后将学到它。

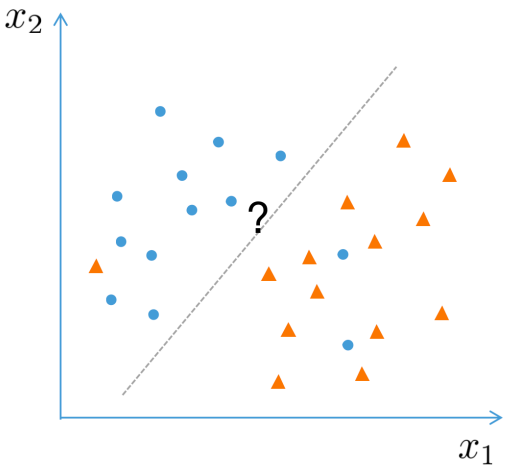

Figure 5: 硬间隔最大化失效的情况

2.2.1 支持向量（support vector）与间隔边界

定义：支持向量

在线性可分情况下，训练数据集的样本点中与分离超平面距离最近的样本点的实例称为支持向量。

© 本课件仅限东南大学机器学习课程使用，版权属于相关作者，严禁传播。

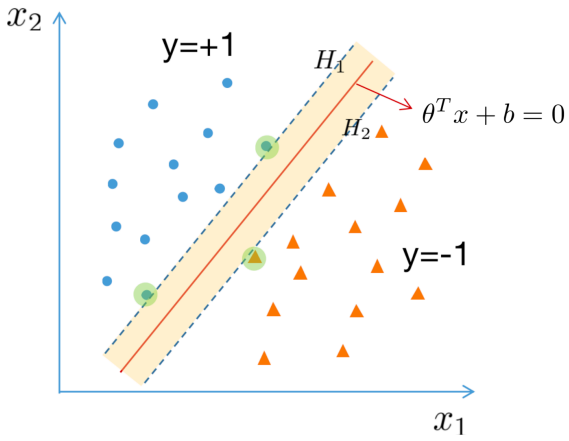

Figure 6: 支持向量，也就是图 6 中的用绿色圆圈圈起来的点

对于上图数据点的分类函数，事实上，我们依旧可以用之前在感知机该部分内容中所使用的：

$$f(\mathbf{x}) = sign(\boldsymbol{\theta}^\top \mathbf{x} + b) = \begin{cases} +1 & \boldsymbol{\theta}^\top \mathbf{x} \geq 0 \\ -1 & \boldsymbol{\theta}^\top \mathbf{x} \leq 0 \end{cases} \tag{4}$$

但在支持向量机中，我们在引入了支持向量的概念后，我们要求，对于任一支持向量 $(\mathbf{x}, y)$，其需要满足约束式：

$$y(\boldsymbol{\theta}^\top \mathbf{x} + b) - 1 = 0 \tag{5}$$

对 $y = +1$ 的正例点，支持向量在超平面 H_1 上，而对于 $y = -1$ 的负例点，支持向量在超平面 H_2 上；

$$H_1 : \boldsymbol{\theta}^\top \mathbf{x} + b = 1 \tag{6}$$

$$H_2 : \boldsymbol{\theta}^\top \mathbf{x} + b = -1 \tag{7}$$

从 (6)(7) 式和图 6 可知，超平面 H_1 和 H_2 平行，它们中间没有实例点，而分离超平面与之平行并位于它们中央。
两个支撑超平面的距离被称作**间隔（margin）**。如图 7 所示，超平面 H_1 和 H_2 之间的距离，也就是紫色线段的长度，即为间隔。而 H_1 和 H_2 两个支撑超平面被称为**间隔边界**。

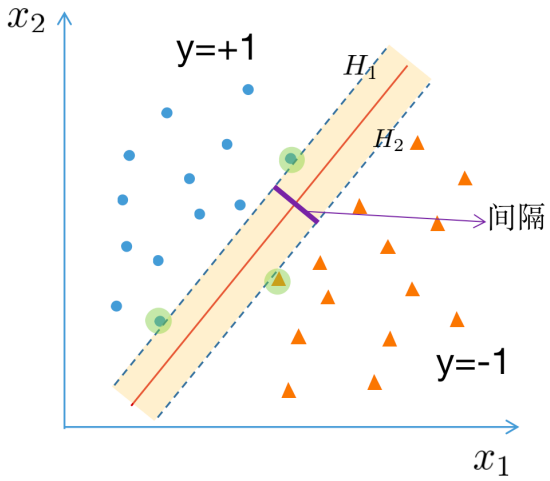

Figure 7: 间隔边界

在决定分离超平面时只有支持向量起作用，其他实例点并不起作用。如果移动这些支持向量将改变所求的解；但是如果在间隔边界以外移动其他实例点，甚至去掉这些点，那么解是不会改变的。且一般来说支持向量的个数很少，所以支持向量机由很少的“重要的”训练样本确定。
在 1.1 的线性分类器中，我们提到感知机的缺陷在于它的解可能并不唯一，可能存在无数个，我们希望能从中选择使整个模型对每个数据点确信度最高的那一个，而在线性可分支持向量机中，这个解的确是唯一的，可见如下定理。

定理 2.1. （**最大间隔分离超平面的存在唯一性**）若训练数据集线性可分，则可将训练数据集中的样本点完全正确分开的最大间隔分离超平面存在且唯一。

有兴趣的同学可以自行上网搜索该定理的证明或翻阅引用 [1] 书目 7.1 节内容，分别从存在性与唯一性两个角度去解释，证明并不复杂困难，在这里不多赘述。

© 本课件仅限东南大学机器学习课程使用，版权属于相关作者，严禁传播。

通过两个间隔边界，我们将数据点分为两类，通过判断数据点特征向量带入 $\boldsymbol{\theta}^\top \mathbf{x}+b$ 的值，我们可以对其进行划分。$-1 \leq \boldsymbol{\theta}^\top \mathbf{x}+b \leq 1$ 表示间隔边界之间的空间，这其中是不含有任何实例点的。

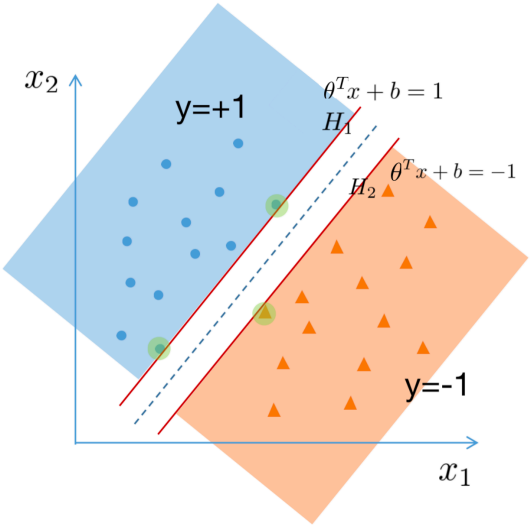

Figure 8: 间隔边界划分的区域

Table 1: 代入获得值与划分的类关系

代入值	划分类
$\boldsymbol{\theta}^\top \mathbf{x}+b \geq 1$	+1
$\boldsymbol{\theta}^\top \mathbf{x}+b \leq 1$	-1
$-1 \leq \boldsymbol{\theta}^\top \mathbf{x}+b \leq 1$	-

3 硬间隔支持向量机

在上一节介绍了最大间隔分类器，可以在样本**线性可分**的情况下，找到分隔样本且间隔最大的分界面。这种情况下，支持向量机可以表述成如下优化问题：

$$\begin{aligned} &\max M \\ &\text{s.t. } \boldsymbol{\theta}^\top \mathbf{x}+b \geq +1, \mathbf{x}\text{属于}+1\text{类} \\ &\quad \boldsymbol{\theta}^\top \mathbf{x}+b \leq -1, \mathbf{x}\text{属于}-1\text{类} \end{aligned} \tag{8}$$

那么如何定义间隔的宽度呢？如何用模型的参数来表示间隔 M？我们先观察一下特殊情况。如图9所示，在一维情况下，样本全部被分成两类，两个类别的边界分别表示为 x^- 和 x^+，那么它们满足以下条件：

$$\begin{cases} \boldsymbol{\theta}^\top \mathbf{x}^- + b = -1 \\ \boldsymbol{\theta}^\top \mathbf{x}^+ + b = 1 \end{cases} \tag{9}$$

两式相减可以得到

$$\boldsymbol{\theta}^\top (\mathbf{x}^+ - \mathbf{x}^-) = 2 \tag{10}$$

而间隔显然为 $M = |\mathbf{x}^+ - \mathbf{x}^-|$，因此 M 可以表示为

$$M = \left|\frac{2}{\boldsymbol{\theta}}\right| = \frac{2}{\sqrt{\boldsymbol{\theta}^\top \boldsymbol{\theta}}} = \frac{2}{\|\boldsymbol{\theta}\|} \tag{11}$$

最大化间隔 M 也就可以等价于最小化 $\|\boldsymbol{\theta}\|$。现在我们可以通过最大化间隔来学习最优的参数。

© 本课件仅限东南大学机器学习课程使用，版权属于相关作者，严禁传播。

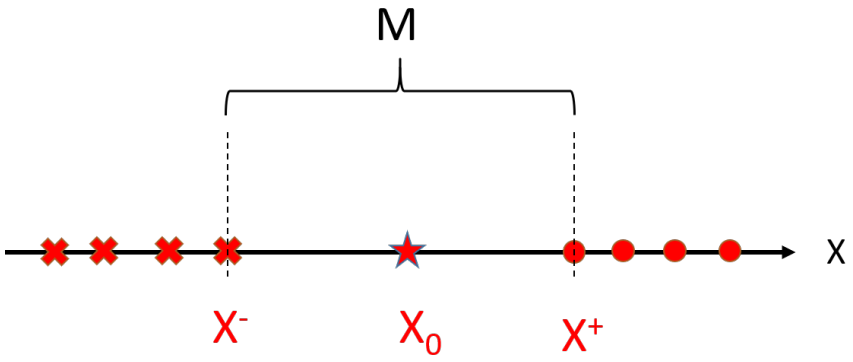

Figure 9: 一维情况下的 SVM

下面我们来推导一般情况下间隔宽度与模型参数的关系。为了便于理解，图10展示了二维情况下的 SVM，实际在高维情况下，分界面是一个超平面。$\mathbf{x}^+$ 和 $\mathbf{x}^-$ 依然是两个分解面上的点，满足式 (9)，且它们的连线垂直于分解面，那么 $\mathbf{x}^+ - \mathbf{x}^-$ 的长度就是间隔宽度。可以观察到，向量 $\boldsymbol{\theta}$ 是垂直于 $+1$ 分界面的，下面就来证明这个结论。假设 $\mathbf{u}$ 和 $\mathbf{v}$ 是 $+1$ 分界面上不同两点，那么对 u 和 v 连起来的向量有：$\boldsymbol{\theta}^\top(\mathbf{u}-\mathbf{v}) = \boldsymbol{\theta}^\top\mathbf{u} - \boldsymbol{\theta}^\top\mathbf{v} = (1-b)-(1-b) = 0$，所以向量 $\boldsymbol{\theta}$ 与平面上任意一条直线垂直，即 $\boldsymbol{\theta}$ 与该平面垂直。同样的 $\boldsymbol{\theta}$ 与 -1 分界面也垂直。由此我们得到向量 $\boldsymbol{\theta}$ 与 $\mathbf{X}^+ - \mathbf{X}^-$ 平行，可以表示成

$$\mathbf{X}^+ = \lambda\boldsymbol{\theta} + \mathbf{X}^- \tag{12}$$

结合式 (9)，可以推理得到：

$$\begin{aligned}
&\boldsymbol{\theta}^\top\mathbf{X}^+ + b = +1 \\
\Rightarrow &\boldsymbol{\theta}^\top(\lambda\boldsymbol{\theta} + \mathbf{X}^-) + b = +1 \\
\Rightarrow &\boldsymbol{\theta}^\top\mathbf{X}^- + b + \lambda\boldsymbol{\theta}^\top\boldsymbol{\theta} = +1 \\
\Rightarrow &-1 + \lambda\boldsymbol{\theta}^\top\boldsymbol{\theta} = +1 \\
\Rightarrow &\lambda = \frac{2}{\boldsymbol{\theta}^\top\boldsymbol{\theta}}
\end{aligned}$$

再结合 $M = |\mathbf{X}^+ - \mathbf{X}^-|$，可以得到：

$$M = |\mathbf{X}^+ - \mathbf{X}^-| = |\lambda\boldsymbol{\theta}| = \lambda\sqrt{\boldsymbol{\theta}^\top\boldsymbol{\theta}} = \frac{2}{\sqrt{\boldsymbol{\theta}^\top\boldsymbol{\theta}}} \tag{13}$$

这与之前在一维情况下得到的结论是一致的。

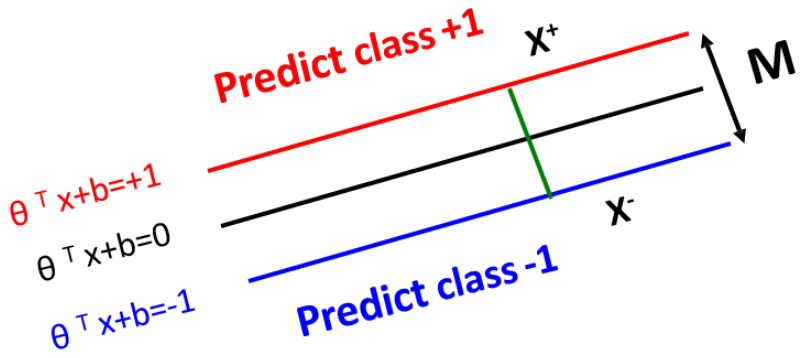

Figure 10: 二维情况下的 SVM

我们得到间隔 M 的值后，可以确定最优化问题，使间隔最大，即最小化 $\boldsymbol{\theta}^\top\boldsymbol{\theta}$，同时满足以下约束条件：对于 $+1$ 类的样本 $\mathbf{x}$，满足 $\boldsymbol{\theta}^\top\mathbf{x}+b \geq 1$；对于 -1 类的样本 $\mathbf{x}$，满足 $\boldsymbol{\theta}^\top\mathbf{x}+b \leq -1$。那么如果我们有 n 个样本，就有 n 个约束条件。为了简化，我们令 y 表示样本的类别，当样本为 $+1$ 类时，$y=1$；当样本为 -1 类时，$y=-1$。那么所有的约束条件都可以表示为 $y(\boldsymbol{\theta}^\top\mathbf{x}+b) \geq 1$。综上，我们的最优化问题可以写成：

$$\begin{aligned}
&\min_{\boldsymbol{\theta},b} \frac{\boldsymbol{\theta}^\top\boldsymbol{\theta}}{2} \\
&\text{subject to } y(\boldsymbol{\theta}^\top\mathbf{x}+b) \geq 1
\end{aligned} \tag{14}$$

对于这个优化问题，目标函数是二次的，约束条件是线性的，它是一个凸二次规划问题，可以使用现成的二次规划优化包来求解。

© 本课件仅限东南大学机器学习课程使用，版权属于相关作者，严禁传播。

二次规划的一般形式包含二次的目标函数和线性的等式约束和不等式约束。

$$\begin{aligned} \min f(\mathbf{x}) &= \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^\top\mathbf{x} \\ \text{s.t. } \mathbf{A}\mathbf{x} &= \mathbf{a} \\ \mathbf{C}\mathbf{x} &\leq \mathbf{c} \end{aligned} \tag{15}$$

其中 $\mathbf{A}, \mathbf{C}, \mathbf{Q} \in \mathbb{R}^{m\times m}$，$\mathbf{a}, \mathbf{c}, \mathbf{q} \in \mathbb{R}^m$，对于 SVM 的优化问题，$\mathbf{Q} = \frac{1}{2}\mathbf{I}$，$\mathbf{q}_i = -1(i \in \{1, 2\cdots, m\})$，$\mathbf{A} = \mathbf{y}$，$\mathbf{a} = \mathbf{0}$，$\mathbf{C} = -\mathbf{I}$，$\mathbf{c} = \mathbf{0}$。

4 编程实现

这里我们利用 Python 中的 scikit-learn 库提供的函数来展示支持向量机的效果。除此之外，国立台湾大学的林轩田（Hsuan-Tien Lin）教授基于 Python 语言编写了专门针对于求解支持向量机的 LibSVM 库。

```python
import numpy as np
from sklearn.svm import LinearSVC
from sklearn.datasets import make_classification
import matplotlib.pyplot as plt

# 自动生成训练数据，有2个类别，2个特征（大概率线性可分）
X, y = make_classification(n_features=2, n_classes=2, n_redundant=0, flip_y=0, class_sep=3)

# 初始化分类器
clf = LinearSVC(loss='hinge')
# 训练
clf.fit(X, y)

# 计算分解面
w = clf.coef_[0]
a = -w[0] / w[1]
b = -clf.intercept_[0] / w[1]
xx = np.linspace(-4, 4)
yy = a*xx + b

# 画图
plt.figure()
plt.scatter(X[:, 0], X[:, 1], c=y, s=30, cmap=plt.cm.Paired)
plt.plot(xx, yy, c='k', linewidth=1.5)
plt.axis()
plt.show()
```

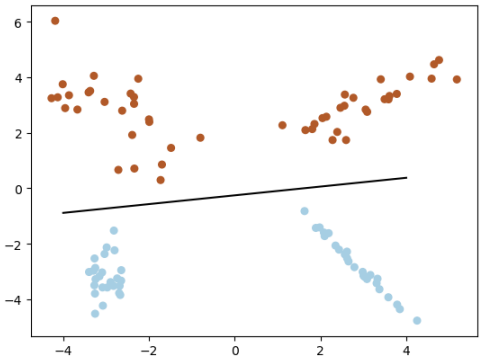

Figure 11: 程序结果图

引用

[1] Hang Li "Statistical Learning"

[2] LibSVM: https://www.csie.ntu.edu.tw/~cjlin/libsvm/。

© 本课件仅限东南大学机器学习课程使用，版权属于相关作者，严禁传播。