Topic 4: 分类
Lecture 1: 支持向量机

林荣荣
linrr@gdut.edu.cn

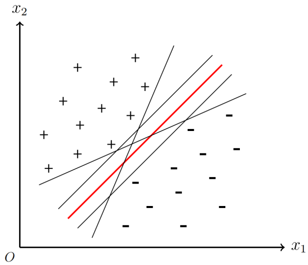
数学与统计学院
广东工业大学

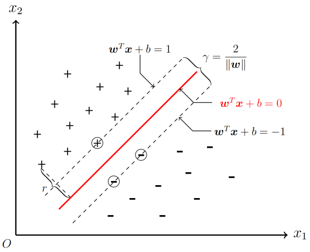
2023年8月29日

支持向量机[3]:最大间隔超平面

支持向量机(Support Vector Machines, SVM)

- **硬间隔**线性支持向量机(Hard Margin SVM)–V. N. Vapnik, 1982
- **软间隔**线性支持向量机(Soft Margin SVM)
- **核(非线性)**支持向量机(Kernel SVM)–Boser, Guyon, V. N. Vapnik, 1992[1,2]





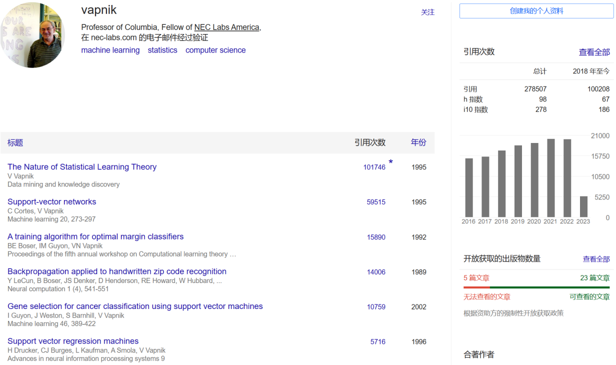
[1] B. Boser, I. Guyon, V. Vapnik, A training algorithm for optimal margin classifiers, In Fifth Annual Workshop on Computational Learning Theory, 1992:144–152.

[2] Cortes C, Vapnik V. Support-vector networks, Machine learning, 1995[60200+]

[3] LIBSVM工具包: https://www.csie.ntu.edu.tw/~cjlin/libsvm/

支持向量机之父–Vladimir Vapnik[4]

Vladimir Vapnik是俄罗斯统计学家、数学家。他是统计学习理论(Statistical Learning Theory)的主要创建人之一，该理论也被称作VC 理论(Vapnik Chervonenkis Theory)



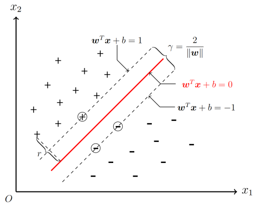
[4]支持向量机的发展历史: https://www.svms.org/history.html

1. 硬间隔线性支持向量机

- 超平面：$h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b = 0$[5], $\mathbf{x} \in \mathbb{R}^d$
- 线性分类器：

$$y = \mathrm{sgn}\,(h(x)) = \begin{cases} +1 & \text{若 } h(\mathbf{x}) > 0, \\ -1 & \text{若 } h(\mathbf{x}) < 0. \end{cases}$$

- 点$\mathbf{x}$(标签为$y \in \{-1, 1\}$)到超平面$h(\mathbf{x}) = 0$的距离：$\delta = \frac{yh(\mathbf{x})}{\|\mathbf{w}\|}$
- 间隔：$\rho = \min_{\mathbf{x}_i} \left\{ \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|} \right\}$；选取$\mathbf{w}$ 使得δ最大
- 支持向量：取到最小距离的点或者在$\mathbf{w}^\top \mathbf{x} + b = \pm 1$上的点[6]。



[5] $a * (\mathbf{w}^\top \mathbf{x} + b) = 0$对于不同的倍数$a \neq 0$对应同一个超平面

[6] 微软的手写数字识别案例

支持向量机：线性可分情形

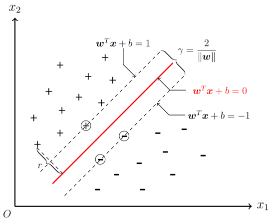
- 找到最大间隔超平面:

$$h^* = \arg\max_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \left\{ \frac{2}{\|\mathbf{w}\|} \right\}.$$

- 原问题：

目标函数: $\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \left\{ \frac{\|\mathbf{w}\|^2}{2} \right\}$

线性约束: [7] $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \forall i = 1, 2, \ldots, n$



[7] 矩阵形式: $\operatorname{diag}(\mathbf{y})(\mathbf{D}\mathbf{w} + b\mathbf{1}) \geq \mathbf{1}$, 其中$\mathbf{D} = [\mathbf{x}_i : i = 1, 2, \ldots, n]$为数据矩阵

拉格朗日乘子法

- 转化为无约束优化问题，拉格朗日函数(凸函数)

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \Big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \Big)$$

其中$\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n)^\top$ 是非负乘子向量。

- 拉格朗日函数$L(\mathbf{w}, b, \alpha)$对$\mathbf{w}$求梯度并令为零得

$$\frac{\partial}{\partial \mathbf{w}} L = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = \mathbf{0} \text{ 或 } \boxed{\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \ (\text{表示定理})}$$

$$\frac{\partial}{\partial b} L = \sum_{i=1}^{n} \alpha_i y_i = 0$$

拉格朗日乘子法

- 转化为无约束优化问题，拉格朗日函数(凸函数)

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \Big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \Big)$$

其中$\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n)^\top$ 是非负乘子向量。

- 拉格朗日函数$L(\mathbf{w}, b, \alpha)$对$\mathbf{w}$求梯度并令为零得

$$\frac{\partial}{\partial \mathbf{w}} L = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = \mathbf{0} \text{ 或 } \boxed{\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \ (\text{表示定理})}$$

$$\frac{\partial}{\partial b} L = \sum_{i=1}^{n} \alpha_i y_i = 0$$

- 将$\mathbf{w}$带回$L(\mathbf{w}, b, \alpha)$可得

$$L = \frac{1}{2}\mathbf{w}^\top \mathbf{w} - \mathbf{w}^\top \Big(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i\Big) - b\sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\mathbf{w}^\top \mathbf{w}.$$

对偶问题: 凸二次规划问题

对偶问题(注：约束条件相比原问题变的很简单)

$$\textbf{目标函数：} \max_{\alpha\in\mathbb{R}^n} L_{dual}(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$$\textbf{线性约束：} \alpha_i \geq 0, \forall i = 1, 2, \ldots, n, \text{ 且 } \sum_{i=1}^{n}\alpha_i y_i = 0$$

简洁的矩阵形式：

$$\boxed{\min_{\alpha\in\mathbb{R}^n} \frac{1}{2}\alpha^\top H\alpha - \mathbf{1}^\top\alpha \text{ s.t. } \alpha \geq 0, \mathbf{y}^\top\alpha = 0}$$

其中乘子向量$\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n)^\top \in [0, +\infty)^n$

$$H := [H_{ij} : i, j = 1, 2, \ldots, n] := [y_i y_j \mathbf{x}_i^\top \mathbf{x}_j : i, j = 1, 2, \ldots, n] = \mathrm{diag}(\mathbf{y})\mathbf{D}\mathbf{D}^\top\mathrm{diag}(\mathbf{y})$$

Karush-Kuhn-Tucker (KKT) 最优性条件

回顾拉格朗日函数 $L(\mathbf{w},\alpha)=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{n}\alpha_i\big(y_i(\mathbf{w}^\top\mathbf{x}_i+b)-1\big)$.

KKT最优性条件

$$\alpha_i\big(y_i(\mathbf{w}^\top\mathbf{x}_i+b)-1\big)=0.$$

上述可分为以下两种情形

(1) $\alpha_i=0$, 或

(2) $y_i(\mathbf{w}^\top\mathbf{x}_i+b)-1=0$, 即$y_i(\mathbf{w}^\top\mathbf{x}_i+b)=1$

Karush-Kuhn-Tucker (KKT) 最优性条件

回顾拉格朗日函数 $L(\mathbf{w}, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1\big)$.

KKT最优性条件

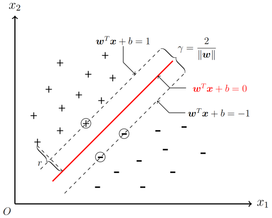
$$\alpha_i \big(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1\big) = 0.$$

上述可分为以下两种情形

(1) $\alpha_i = 0$, 或

(2) $y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 = 0$, 即$y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$

若$\alpha_i > 0$, 则 $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$, 此时 $\mathbf{x}_i$是支持向量. 若 $y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 1$, 则 $\alpha_i = 0$, 即非支持向量对应得乘子$\alpha_i = 0$.



权重和偏置

(i) 权重：由表示定理可得(稀疏选择)

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

(ii) 偏置：$\alpha_i > 0$对应的支持向量落在± 1超平面$y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$可得

$$b = \text{avg}_{\alpha_i > 0}\{b_i\} = \text{avg}_{\alpha_i > 0}\{y_i - \mathbf{w}^\top \mathbf{x}_i\}.$$

(iii) SVM分类器:

$$\hat{y} = \text{sign}(h(\mathbf{z})) = \text{sign}(\mathbf{w}^\top \mathbf{z} + b).$$

(iv) 间距：由$b = y_i - \sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_i^\top \mathbf{x}_j$可得

$$\sum_{i=1}^{n} \alpha_i y_i b = \sum_{i=1}^{n} \alpha_i y_i^2 - \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \Leftrightarrow 0 = \sum_{i=1}^{n} \alpha_i - \|\mathbf{w}\|^2$$

$$\Rightarrow \rho = \frac{1}{\|\mathbf{w}\|} = \frac{1}{\sqrt{\|\alpha\|_1}}$$

例1： 硬间隔线性支持向量机的计算

1. $\mathbb{R}^2$中的数据集

$\mathbf{x}^\top$	y
(0,0)	-1
(1,1)	-1
(0,2)	1

2. 硬间隔线性支持向量机的原问题

$$\min_{\mathbf{w}\in\mathbb{R}^2, b\in\mathbb{R}} \frac{\|\mathbf{w}\|^2}{2} \text{ s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \forall i = 1, 2, 3.$$

3. 硬间隔线性支持向量机的对偶问题

$$\max_{\alpha\in\mathbb{R}^3} L(\alpha) = \sum_{i=1}^{3} \alpha_i - \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \text{ s.t. } \alpha_i \geq 0, \forall i \text{ 且 } \sum_{i=1}^{3} \alpha_i y_i = 0$$

注意到$\mathbf{x}_1 = (0,0)^\top$. 计算($H = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & -2 \\ 0 & -2 & 4 \end{pmatrix}$)

$$
\begin{aligned}
L(\alpha) &= \alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2}(\alpha_2^2 y_2^2 \mathbf{x}_2^\top \mathbf{x}_2 + 2\alpha_2\alpha_3 y_2 y_3 \mathbf{x}_2^\top \mathbf{x}_3 + \alpha_3^2 y_3^2 \mathbf{x}_3^\top \mathbf{x}_3) \\
&= \alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2}(2\alpha_2^2 - 4\alpha_2\alpha_3 + 4\alpha_3^2) \\
&= \alpha_1 + \alpha_2 + \alpha_3 - \alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_3^2.
\end{aligned}
$$

由$\sum_{i=1}^3 \alpha_i y_i = -\alpha_1 - \alpha_2 + \alpha_3 = 0$可得$\alpha_3 = \alpha_1 + \alpha_2$. 代入上式可得

$$L(\alpha) := 2\alpha_1 + 2\alpha_2 - 2\alpha_1^2 - \alpha_2^2 - 2\alpha_1\alpha_2.$$

4. 对$L(\alpha_1,\alpha_2)$关于α_1,α_2分别求偏导并令为0:

$$2-4\alpha_1-2\alpha_2=0,\ 2-2\alpha_2-2\alpha_1=0$$

可得$\alpha_1=0$, $\alpha_2=1$. 进而可得$\alpha_3=1$.

5. 由于$\alpha_2>0$, $\alpha_3>0$, 可得支持向量为$\mathbf{x}_2,\mathbf{x}_3$
6. 超平面$h(\mathbf{x})=\mathbf{w}^\top\mathbf{x}+b$的参数的精确解

$$\mathbf{w}=\sum_{\alpha_i>0}\alpha_i y_i\mathbf{x}_i=(-1,1)^\top$$

$$b_2=y_2-\mathbf{w}^\top\mathbf{x}_2=-1, b_3=y_3-\mathbf{w}^\top\mathbf{x}_3=-1,\ b=\frac{1}{2}(b_2+b_3)=-1$$

硬间隔线性支持向量机分类器: 给定新的一个点$\mathbf{z}\in\mathbb{R}^2$, 预测标签为[8]

$$y=\mathrm{sign}(h(\mathbf{z}))=\mathrm{sign}(-z_1+z_2-1),\ \mathbf{z}=(z_1,z_2)^\top.$$

[8]注意: $\mathbf{x}_1=(0,0)$也在$\mathbf{w}^\top\mathbf{x}+b=-1$超平面上，但$\alpha_1=0$, 因而不是支持向量

例2: 硬间隔线性支持向量机的计算[9]

1. 数据集

标签	数据点
1	$\mathbf{x}_1 = (3,3)$, $\mathbf{x}_2 = (4,3)$
-1	$\mathbf{x}_3 = (1,1)$

2. 硬间隔线性支持向量机的对偶问题

$$\min_{\alpha \in \mathbb{R}^3} L(\alpha) = 9\alpha_1^2 + \frac{25}{2}\alpha_2^2 + \alpha_3^2 + 21\alpha_1\alpha_2 - 6\alpha_1\alpha_3 - 7\alpha_2\alpha_3 - (\alpha_1 + \alpha_2 + \alpha_3)$$

$$\text{s.t. } \alpha_1 + \alpha_2 - \alpha_3 = 0, \ \alpha_1, \alpha_2, \alpha_3 \geq 0$$

3. 将$\alpha_3 = \alpha_1 + \alpha_2$带入上式可得

$$L(\alpha_1, \alpha_2) = 4\alpha_1^2 + \frac{13}{2}\alpha_2^2 + 10\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2$$

[9] 李航，机器学习方法、第三版，清华大学出版社，2022，第7章，例7.2

4. 对$L(\alpha_1,\alpha_2)$关于α_1,α_2求偏导数可得$(\frac{3}{2},-1)$处取极值，不满足$\alpha_2 \geq 0$. 所以$L(\alpha_1,\alpha_2)$最小值在边界取得。 由于$L(0,\frac{2}{13}) = -\frac{2}{13} > L(\frac{1}{4},0) = -\frac{1}{4}$, 可得最优解 $\alpha_1 = 1/4$, $\alpha_2 = 0$, 从而$\alpha_3 = 1/4$.
5. $\alpha_1,\alpha_3 > 0$对应的点$\mathbf{x}_1,\mathbf{x}_3$为支持向量
6. 超平面$h(\mathbf{x}) = \mathbf{w}^\top\mathbf{x} + b$的参数的精确解

$$\mathbf{w} = \sum_{\alpha_i>0} \alpha_i y_i \mathbf{x}_i = \alpha_1 y_1 \mathbf{x}_1 + \alpha_3 y_3 \mathbf{x}_3 = (\frac{1}{2},\frac{1}{2})^\top$$

$$b_1 = y_1 - \mathbf{w}^\top\mathbf{x}_1 = -2, b_3 = y_3 - \mathbf{w}^\top\mathbf{x}_3 = -2, \ b = \frac{1}{2}(b_1 + b_3) = -2$$

7. 分类器：对于新的一个点$\mathbf{z} = (z_1,z_2)^\top \in \mathbb{R}^2$

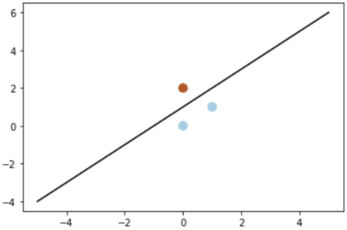
$$h(\mathbf{z}) = \mathrm{sign}\left(\frac{1}{2}z_1 + \frac{1}{2}z_2 - 2\right)$$

Python实现(调用)

```python
import numpy as np
import matplotlib.pylab as plt
from sklearn import svm
X=np.array([[0,0],[1,1],[0,2]])
Y=np.array([-1,-1,1])
plt.scatter(X[:,0],X[:,1],c=Y,s=80,cmap=plt.cm.Paired)
clf=svm.SVC(kernel='linear')
clf.fit(X,Y)
w=clf.coef_[0]
print('w =',w)
a=-w[0]/w[1]
print('b =', a)
xx=np.linspace(-5,5)
yy=a*xx-(clf.intercept_[0])/w[1]
plt.plot(xx,yy,'k-')
```

```
w = [-1.  1.]
b = 1.0

[<matplotlib.lines.Line2D at 0x1fb781cbe20>]
```



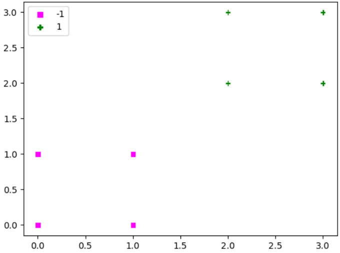
例3： 硬间隔线性支持向量机的计算 [10] [11]

x=np.array([[0,0],[0,1],[1,0],[1,1],[2,2],[2,3],[3,2],[3,3]])

y = np.array([-1, -1, -1, -1, 1, 1, 1, 1])

alpha=[0,0,0,1,1,0,0,0]

w=[1,1], b=-3



[10] M. Gopal, Applied Machine Learning, McGraw-Hill Education, 2018, Example 4.1

[11] SciPy求解： https://machinelearningmastery.com/method-of-lagrange-multipliers-the-theory-behind-support-vector-machines-part-3-im

拓展：二次规划(Quadratic Programming)

- 一般约束二次规划标准型[12]:

$$\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^\top \mathbb{P}\mathbf{x} + \mathbf{q}^\top \mathbf{x} \text{ s.t. } G\mathbf{x} \le h, A\mathbf{x} = b,\ lb \le \mathbf{x} \le ub$$

  其中 $\mathbb{P}$是半正定的。

- 等式约束的二次规划的标准型：

$$\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\mathbf{x}^\top \mathbf{P}\mathbf{x} + \mathbf{q}^\top \mathbf{x} \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \ge 0$$

- 交替乘子法(ADMM[13])

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) + g(\mathbf{z}) \text{ s.t. } \mathbf{x} - \mathbf{z} = 0$$

  其中$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{P}\mathbf{x} + \mathbf{q}^\top \mathbf{x}$, $\mathrm{dom}(f) = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}\}$, g 是$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \ge 0\}$上指示函数

---

[12] pip install qpsolvers: https://github.com/qpsolvers/qpsolvers

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Foundations and Trend in Machine Learning, Vol. 3, No. 1 (2010), p.36. [Citation: 20100+]

拓展：支持向量机的交替乘子法(不要求)[14][15]

迭代形式：

$$
\begin{aligned}
\mathbf{x}^{t+1} &= \arg\min_{\mathbf{x}}(f(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{x} - \mathbf{z}^t + \mathbf{u}^t\|^2) \\
\mathbf{z}^{t+1} &= \max\{0, \mathbf{x}^{t+1} + \mathbf{u}^t\} \\
\mathbf{u}^{t+1} &= \mathbf{u}^t + \mathbf{x}^{t+1} - \mathbf{z}^{t+1}
\end{aligned}
$$

其中$\mathbf{x}$的更新是一个等式约束的最小二乘问题：

$$
\begin{pmatrix} \mathbf{P} + \rho I & \mathbf{A}^\top \\ \mathbf{A} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}^{t+1} \\ \nu \end{pmatrix} + \begin{pmatrix} \mathbf{q} - \rho(\mathbf{z}^t - \mathbf{u}^t) \\ -\mathbf{b} \end{pmatrix} = 0.
$$

[14] J. Gallier, J. Quaintance, Algebra, Topology, Differential Calculus, and Optimization Theory For Computer Science and Machine Learning, 2022, Chapter 53.

[15] 林宙辰， 李欢，方聪， 机器学习中的交替乘子法，科学出版社，2023

拓展：支持向量机的交替乘子法(不要求)[14][15]

迭代形式：

$$\begin{aligned}
\mathbf{x}^{t+1} &= \arg\min_{\mathbf{x}}(f(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{x} - \mathbf{z}^t + \mathbf{u}^t\|^2) \\
\mathbf{z}^{t+1} &= \max\{0, \mathbf{x}^{t+1} + \mathbf{u}^t\} \\
\mathbf{u}^{t+1} &= \mathbf{u}^t + \mathbf{x}^{t+1} - \mathbf{z}^{t+1}
\end{aligned}$$

其中$\mathbf{x}$的更新是一个等式约束的最小二乘问题：

$$\begin{pmatrix} \mathbf{P} + \rho I & \mathbf{A}^\top \\ \mathbf{A} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}^{t+1} \\ \nu \end{pmatrix} + \begin{pmatrix} \mathbf{q} - \rho(\mathbf{z}^t - \mathbf{u}^t) \\ -\mathbf{b} \end{pmatrix} = 0.$$

特别的对于支持向量机，我们有

$$\mathbf{P} = (y_i y_j \mathbf{x}_i^\top \mathbf{x}_j : i, j = 1, 2, \ldots, n), \; \mathbf{q} = -(1, 1, \ldots, 1)^\top$$

$$\mathbf{A} = \mathrm{diag}(y_1, y_2, \ldots, y_n), \; \mathbf{x} = \alpha, \mathbf{b} = \mathbf{0}.$$

[14] J. Gallier, J. Quaintance, Algebra, Topology, Differential Calculus, and Optimization Theory For Computer Science and Machine Learning, 2022, Chapter 53.

[15] 林宙辰， 李欢，方聪， 机器学习中的交替乘子法，科学出版社，2023

例4: 线性可分

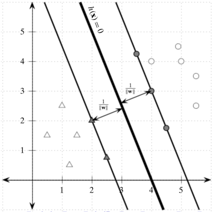
$\mathbf{x}_i$	x_{i1}	x_{i2}	y_i
$\mathbf{x}_1$	3.5	4.25	$+1$
$\mathbf{x}_2$	4	3	$+1$
$\mathbf{x}_3$	4	4	$+1$
$\mathbf{x}_4$	4.5	1.75	$+1$
$\mathbf{x}_5$	4.9	4.5	$+1$
$\mathbf{x}_6$	5	4	$+1$
$\mathbf{x}_7$	5.5	2.5	$+1$
$\mathbf{x}_8$	5.5	3.5	$+1$
$\mathbf{x}_9$	0.5	1.5	-1
$\mathbf{x}_{10}$	1	2.5	-1
$\mathbf{x}_{11}$	1.25	0.5	-1
$\mathbf{x}_{12}$	1.5	1.5	-1
$\mathbf{x}_{13}$	2	2	-1
$\mathbf{x}_{14}$	2.5	0.75	-1

$\mathbf{x}_i$	x_{i1}	x_{i2}	y_i	α_i
$\mathbf{x}_1$	3.5	4.25	$+1$	0.0437
$\mathbf{x}_2$	4	3	$+1$	0.2162
$\mathbf{x}_4$	4.5	1.75	$+1$	0.1427
$\mathbf{x}_{13}$	2	2	-1	0.3589
$\mathbf{x}_{14}$	2.5	0.75	-1	0.0437

权重向量: $\mathbf{w} = \sum_{\alpha_i>0} \alpha_i y_i \mathbf{x}_i = (0.833, 0.334)^\top$

偏置: $b = \text{avg}_{\alpha_i>0}\{y_i - \mathbf{w}^\top \mathbf{x}_i\} = -3.332$

最优超平面: $h(\mathbf{x}) = (0.833, 0.334)\mathbf{x} - 3.332 = 0$



2 软间隔支持向量机: 线性不可分情形

引入松弛变量$\xi_i \geq 0$(衡量不符合线性可分的程度)的线性约束[16]：

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \text{ 或 } \xi_i = \max\{0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\}$$

[16]矩阵形式: $\mathrm{diag}\,(\mathbf{y})(\mathbf{D}\mathbf{w} + b\mathbf{1}) + \xi \geq \mathbf{1}$, 其中$\mathbf{D} = [\mathbf{x}_i : i = 1, 2, \ldots, n]$为数据矩阵

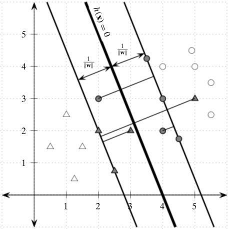
2 软间隔支持向量机: 线性不可分情形

引入松弛变量$\xi_i \geq 0$(衡量不符合线性可分的程度)的线性约束[16]：

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \text{ 或 } \xi_i = \max\{0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\}$$

- 若 $\xi_i = 0$, 则 $\mathbf{x}_i$ 离超平面的距离大于等于$\frac{1}{\|\mathbf{w}\|}$;
- 若$0 < \xi_i < 1$, 则该点位于间隔内并且依然分类正确;
- 若 $\xi_i \geq 1$, 则该点分类错误，并且出现在了超平面的另一侧。

下图阴影点是支持向量



[16]矩阵形式: $\mathrm{diag}(\mathbf{y})(\mathbf{D}\mathbf{w} + b\mathbf{1}) + \xi \geq \mathbf{1}$, 其中$\mathbf{D} = [\mathbf{x}_i : i = 1, 2, \ldots, n]$为数据矩阵

原问题：随机梯度下降算法

- 不可分线性支持向量机(凸)优化问题($b=0$, $\xi_i = \max\{0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i\}$)：

$$\min_{\mathbf{w}} J = \frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{n} \xi_i.$$

- 给定点$(\mathbf{x}_i, y_i)$, 其中$y_i \in \{-1, 1\}$的梯度

$$\nabla J(\mathbf{w}, \mathbf{x}_i) = \begin{cases} \mathbf{w}, & \text{if } \max\{0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i\} = 0, \\ \mathbf{w} - C y_i \mathbf{x}_i, & \text{其他}. \end{cases}$$

原问题：随机梯度下降算法

- 不可分线性支持向量机(凸)优化问题($b = 0$, $\xi_i = \max\{0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i\}$)：

$$\min_{\mathbf{w}} J = \frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{n} \xi_i.$$

- 给定点$(\mathbf{x}_i, y_i)$, 其中$y_i \in \{-1, 1\}$的梯度

$$\nabla J(\mathbf{w}, \mathbf{x}_i) = \begin{cases} \mathbf{w}, & \text{if } \max\{0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i\} = 0, \\ \mathbf{w} - C y_i \mathbf{x}_i, & \text{其他}. \end{cases}$$

- 随机选定点$(\mathbf{x}_i, y_i)$的迭代算法($\gamma_t = \frac{\gamma_0}{1+t}$):

$$\mathbf{w} \leftarrow \mathbf{w} - \gamma_t \nabla J(\mathbf{w}, \mathbf{x}_i) = \begin{cases} (1-\gamma_t)\mathbf{w} + \gamma_t C y_i \mathbf{x}_i, & \text{if } y_i \mathbf{w}^\top \mathbf{x}_i \le 1, \\ (1-\gamma_t)\mathbf{w}, & \text{其他}. \end{cases}$$

- 对于强凸函数，梯度算法复杂度$O(nd \ln \frac{1}{\epsilon})$, 随机梯度算法$O(\frac{d}{\epsilon})$.

铰链损失(Hinge Loss)对偶问题：误差项+ 正则项

$$\text{目标函数:}\ \min_{\mathbf{w},b,\xi_i}\left\{\frac{\|\mathbf{w}\|^2}{2}+C\sum_{i=1}^n \xi_i\right\}$$

$$\text{线性约束:}\ y_i(\mathbf{w}^\top \mathbf{x}_i+b)\geq 1-\xi_i, \xi_i\geq 0, \forall i=1,2,\ldots,n.$$

对偶问题

$$\max_{\alpha\in\mathbb{R}^n} L_{dual}(\alpha)=\sum_{i=1}^n \alpha_i-\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

满足$0\leq \alpha_i\leq C$, 且 $\sum_{i=1}^n \alpha_i y_i=0$. 矩阵形式：

$$\min_{\alpha\in\mathbb{R}^n} \frac{1}{2}\alpha^\top H\alpha-\mathbf{1}^\top\alpha,\ \text{s.t. } \mathbf{0}\leq\alpha\leq C\mathbf{1}, \mathbf{y}^\top\alpha=0$$

最优参数

$$\mathbf{w}=\sum_{\alpha_i>0}\alpha_i y_i \mathbf{x}_i, b=\text{avg}_{0<\alpha_i<C}\{y_i-\mathbf{w}^\top\mathbf{x}_i\}$$

二次损失(Quadratic Loss)对偶问题：误差项+正则项

$$\text{目标函数：} \min_{\mathbf{w},b,\xi_i} \left\{ \frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{n}(\xi_i)^2 \right\}$$

$$\text{线性约束：} y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i = 1, 2, \ldots, n.$$

$$\max_{\alpha\in\mathbb{R}^n} L_{dual}(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \left(\mathbf{x}_i^\top \mathbf{x}_j + \frac{1}{2C}\delta_{ij}\right)$$

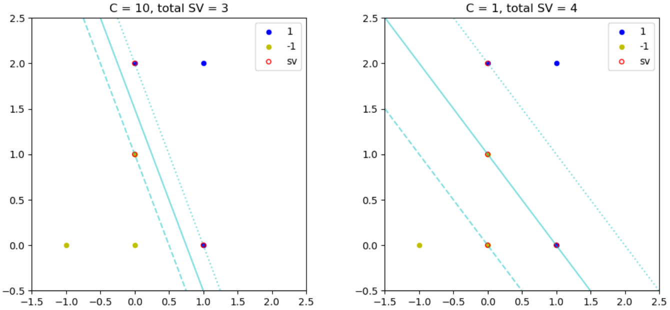
满足$\alpha_i \geq 0$，且 $\sum_{i=1}^{n}\alpha_i y_i = 0$．其中$\delta_{ij}$为Kronecker delta函数．矩阵形式：

$$\boxed{\min_{\alpha\in\mathbb{R}^n} \frac{1}{2}\alpha^\top\left(H + \frac{1}{2C}\mathbb{I}_n\right)\alpha - \mathbf{1}^\top\alpha, \text{ s.t. } \mathbf{0} \leq \alpha, \mathbf{y}^\top\alpha = 0}$$

X=np.array([[0,0],[-1,0],[0,1],[1,0],[0,2],[1,2]])

y=np.array([-1,-1,-1,1,1,1])

H=[[0. 0. 0. 0. 0. 0.] [0. 1. 0. 1. 0. 1.] [0. 0. 1. 0. -2. -2.] [0. 1. 0. 1. 0. 1.] [0. 0. -2. 0. 4. 4.] [0. 1. -2. 1. 4. 5.]]



$\alpha = [0, 0, 10, 4, 6, 0], [4, 2]\cdot\mathbf{x} - 3 = 0, \alpha \approx [1, 0, 1, 1, 1, 0], [1.0003, 0.9978]\cdot\mathbf{x} - 0.9984 = 0$

例6：线性不可分(教材案例)

$\mathbf{x}_i$	x_{i1}	x_{i2}	y_i
$\mathbf{x}_1$	3.5	4.25	$+1$
$\mathbf{x}_2$	4	3	$+1$
$\mathbf{x}_3$	4	4	$+1$
$\mathbf{x}_4$	4.5	1.75	$+1$
$\mathbf{x}_5$	4.9	4.5	$+1$
$\mathbf{x}_6$	5	4	$+1$
$\mathbf{x}_7$	5.5	2.5	$+1$
$\mathbf{x}_8$	5.5	3.5	$+1$
$\mathbf{x}_9$	0.5	1.5	-1
$\mathbf{x}_{10}$	1	2.5	-1
$\mathbf{x}_{11}$	1.25	0.5	-1
$\mathbf{x}_{12}$	1.5	1.5	-1
$\mathbf{x}_{13}$	2	2	-1
$\mathbf{x}_{14}$	2.5	0.75	-1
$\mathbf{x}_{15}$	4	2	$+1$
$\mathbf{x}_{16}$	2	3	$+1$
$\mathbf{x}_{17}$	3	2	-1
$\mathbf{x}_{18}$	5	3	-1

最优超平面

令 $k=1$ and $C=1$. $0<\alpha_i<C$对应的支持向量落在±1平面上, $\alpha_i=C$对应的$\xi_i>0$.

最优超平面:

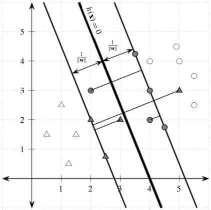
$\mathbf{x}_i$	x_{i1}	x_{i2}	y_i	α_i
$\mathbf{x}_1$	3.5	4.25	$+1$	0.0271
$\mathbf{x}_2$	4	3	$+1$	0.2162
$\mathbf{x}_4$	4.5	1.75	$+1$	0.9928
$\mathbf{x}_{13}$	2	2	-1	0.9928
$\mathbf{x}_{14}$	2.5	0.75	-1	0.2434
$\mathbf{x}_{15}$	4	2	$+1$	1
$\mathbf{x}_{16}$	2	3	$+1$	1
$\mathbf{x}_{17}$	3	2	-1	1
$\mathbf{x}_{18}$	5	3	-1	1

$$h(\mathbf{x})=(0.834,0.333)\mathbf{x}-3.334=0,\ \ \mathbf{x}\in\mathbb{R}^2.$$

其中权重和偏置分别为

$$\mathbf{w}=\sum_{\alpha_i>0}\alpha_i y_i \mathbf{x}_i$$

$$b=\mathrm{avg}_{0<\alpha_i<C}\{b_i\}=\mathrm{avg}_{0<\alpha_i<C}\{y_i-\mathbf{w}^\top\mathbf{x}_i\}.$$



松弛变量

对所有不在间隔上的支持向量有

$\mathbf{x}_i$	$\mathbf{w}^\top \mathbf{x}_i$	$\mathbf{w}^\top \mathbf{x}_i + b$	$\xi_i = 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)$
$\mathbf{x}_{15}$	4.001	0.667	0.333
$\mathbf{x}_{16}$	2.667	-0.667	1.667
$\mathbf{x}_{17}$	3.167	-0.167	0.833
$\mathbf{x}_{18}$	5.168	1.834	2.834

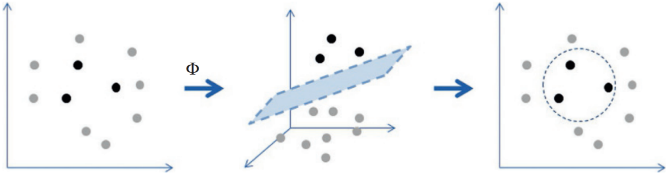
松弛变量的总和：

$$\sum_i \xi_i = \xi_{15} + \xi_{16} + \xi_{17} + \xi_{18} = 0.333 + 1.667 + 0.833 + 2.834 = 5.667.$$

3. 核支持向量机[18]

特征空间中的数据$\mathbf{D}_\phi = \{\phi(\mathbf{x}_i), y_i\}_{i=1}^n \subseteq \mathcal{F} \times \{-1, 1\}$. 核技巧：

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j).$$



常用的核函数[17]：多项式核$(c^2 + \mathbf{x}^\top \mathbf{y})^q$，高斯核/径向基函数(radial basis function, rbf)$e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$，Laplacian核$e^{-\frac{\|\mathbf{x}-\mathbf{y}\|_1}{\sigma}}$, Inverse Multiquadric 核：$\frac{1}{\sqrt{\|\mathbf{x}-\mathbf{y}\|^2+c^2}}$

[17] Google Scholar: polynomial kernel (53900+), Gaussian kernel (649000+)

[18] 核技巧：M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control, 25:821-837, 1964

核支持向量机优化问题：误差项+正则项

$$\text{目标函数:}\quad \min_{\mathbf{w}\in\mathcal{F}, b\in\mathbb{R}, \xi_i} \left\{ \frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{n}(\xi_i)^k \right\}$$

线性约束：[19] $y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i$

$$\xi_i \geq 0, \ \forall i = 1, 2, \ldots, n$$

其中 $\mathbf{w}$, b, ξ_i分别是特征空间中的权重向量，偏置和松弛变量, $k \in \{1, 2\}$.

[19] 损失函数$\xi_i = \max\{0, 1 - y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b)$

核支持向量机：对偶问题

- **铰链损失(Hinge Loss)**:

$$
\begin{aligned}
\max_{\alpha\in\mathbb{R}^n} L_{dual}(\alpha) &= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j) \\
&= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j).
\end{aligned}
$$

其中$0 \le \alpha_i \le C$, 且 $\sum_{i=1}^{n}\alpha_i y_i = 0$.

核支持向量机：对偶问题

- 铰链损失(**Hinge Loss**):

$$\max_{\alpha\in\mathbb{R}^n} L_{dual}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$$
$$= \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j).$$

其中$0 \le \alpha_i \le C$, 且 $\sum_{i=1}^n \alpha_i y_i = 0$. 矩阵形式(注意加负号，max变成min)：

$$\min_{\alpha\in\mathbb{R}^n} \frac{1}{2}\alpha^\top H\alpha - \mathbf{1}^\top\alpha \text{ s.t. } \mathbf{y}^\top\alpha = 0, \mathbf{0} \le \alpha \le C\mathbf{1}$$

其中$H = \mathrm{diag}(\mathbf{y})\mathbf{K}\mathrm{diag}(\mathbf{y})$.

核支持向量机：对偶问题

- 铰链损失(**Hinge Loss**):

$$\max_{\alpha\in\mathbb{R}^n} L_{dual}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$
$$= \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j).$$

其中$0 \le \alpha_i \le C$, 且 $\sum_{i=1}^n \alpha_i y_i = 0$. 矩阵形式(注意加负号，max变成min)：

$$\min_{\alpha\in\mathbb{R}^n} \frac{1}{2}\alpha^\top H\alpha - \mathbf{1}^\top\alpha \text{ s.t. } \mathbf{y}^\top\alpha = 0, \mathbf{0} \le \alpha \le C\mathbf{1}$$

其中$H = \mathrm{diag}(\mathbf{y})\mathbf{K}\mathrm{diag}(\mathbf{y})$.

- 二次损失(**Quadratic loss**): 对应于核的一个变换(即$\mathbf{K}_q := \mathbf{K} + \frac{1}{2C}I_n$)

$$\boxed{K_q(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j) + \frac{1}{2C}\delta_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{2C}\delta_{ij}}$$

核支持向量机：权重向量和偏置

- 权重向量和偏置

$$\mathbf{w} = \sum_{\alpha_i > 0} \alpha_i y_i \phi(\mathbf{x}_i)$$

$$b_i = y_i - \mathbf{w}^\top \phi(\mathbf{x}_i) = y_i - \sum_{0 < \alpha_j < C} \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i).$$

- 表示定理：决策函数(稀疏性，少数点)：

$$h(\mathbf{x}) = \sum_{\alpha_i > 0} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

曲面边界：$h(\mathbf{x}) = 0$ 即$h(\mathbf{x})$等高线为0

- 分类器：对于新的点$\mathbf{x}$的预测类别:

$$\hat{y} = \mathrm{sign}(\mathbf{w}^\top \phi(\mathbf{x}) + b) = \mathrm{sign}\Big(\sum_{\alpha_i > 0} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\Big).$$

- 若选线性核即$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$, 则核支持向量机退化为线性支持向量机。

不失一般性，α_1, α_2是变量，$\alpha_i (i = 3, 4, \dots, n)$固定. 由$K = [K_{ij} : 1 \le i, j \le n]$ $h(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$ 可得

$$\min_{\alpha_1,\alpha_2} L(\alpha_1, \alpha_2) = \frac{1}{2} K_{11}\alpha_1^2 + \frac{1}{2} K_{22}\alpha_2^2 + y_1 y_2 K_{12}\alpha_1\alpha_2 - (\alpha_1 + \alpha_2)$$

$$+ y_1\alpha_1(h(\mathbf{x}_1) - \sum_{i=1}^{2} y_i\alpha_i K_{i1} - b) + y_2\alpha_2(h(\mathbf{x}_2) - \sum_{i=1}^{2} y_i\alpha_i K_{i2} - b)$$

$$\text{s.t. } \alpha_1 = (-\sum_{i=3}^{n} y_i\alpha_i - \alpha_2 y_2) y_1$$

$$0 \le \alpha_1, \alpha_2 \le C$$

[20] John C. Platt, Sequential Minimal Optimization (SMO): A Fast Algorithm for Training Support Vector Machines, 1998, Technical Report MSR-TR-98-14 [4100+]

[21] SMO (每次优化α两个分量): https://github.com/Kaslanarian/PySVM

[22] SMO: https://emilemathieu.fr/posts/2018/08/svm/

[23] 李航，机器学习方法、第三版，清华大学出版社，2022，7.4节

科研问题：任何连续曲面可否被核函数逼近？

任意一个紧集$\mathcal{I} \subseteq \mathbb{R}^d$上连续函数$f \in C(\mathcal{I})$，刻画核函数$K : \mathcal{I} \times \mathcal{I} \to \mathbb{R}$使得

$$\sum_{i=1}^{m} c_i K(\mathbf{x}_i, \mathbf{x}) \to f \text{ 当 } m \to +\infty.$$

- I. Steinwart, Consistency of support vector machines and other regularized kernel classifiers, IEEE TIT, 2005, 51(1): 128-142
- C.A. Micchelli, Y. Xu, H. Zhang, Universal Kernels, Journal of Machine Learning Research, 2006, 7: 2651-2667
- W. Chen, B. Wang, H. Zhang, Universalities of reproducing kernels revisited, Applicable Analysis, 2016, 95(8): 1776-1791

增广权重向量(偏置b不单独写出)

- 增广数据点: $\tilde{\phi}(\mathbf{x}_i) = (\phi(\mathbf{x}_i)^\top, 1)^\top$
- 增广权重: $\tilde{\mathbf{w}} = (\mathbf{w}^\top, b)^\top$, $\mathbf{w} \in \mathcal{F}$, $b \in \mathbb{R}$
- 特征空间中的超平面$h(\mathbf{x}) = \tilde{\mathbf{w}}^\top \tilde{\phi}(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$
- 增广核函数:

$$\tilde{K}(\mathbf{x}_i, \mathbf{x}_j) = \tilde{\phi}(\mathbf{x}_i)^\top \tilde{\phi}(\mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)^\top + 1 = K(\mathbf{x}_i, \mathbf{x}_j) + 1$$

- 表示定理: $\tilde{\mathbf{w}} = \sum_{i=1}^n \alpha_i y_i \tilde{\phi}(\mathbf{x}_i)$
- 对偶问题

$$\max_{\alpha \in \mathbb{R}^n} L_{dual}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \tilde{K}(\mathbf{x}_i, \mathbf{x}_j) \text{ s.t } \mathbf{0} \le \alpha \le C\mathbf{1}$$

- 一个点$\mathbf{x}_k$的目标函数:

$$\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^n \alpha_i - \frac{1}{2} \alpha_k y_k \sum_{i=1}^n \alpha_i y_i \tilde{K}(\mathbf{x}_i, \mathbf{x}_k)$$

对偶核SVM随机梯度算法(教材给的算法有误)

Kernel-SVM-DUAL(D, K, loss, C, ϵ):

if loss=hinge then
    $\mathbf{K} \leftarrow (K(\mathbf{x}_i, \mathbf{x}_j) : i, j = 1, 2, \ldots, n)$% kernel matrix, hinge loss
else if loss=quadratic then
    $\mathbf{K} \leftarrow (K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{2C}\delta_{i,j} : i, j = 1, 2, \ldots, n)$ % kernel matrix, quadratic loss
$\tilde{\mathbf{K}} = \mathbf{K} + 1$ % augmented kernel matrix (核矩阵每个元素加1)
for $k = 1, 2, \ldots, n$ do
    $\eta_k \leftarrow 1/\tilde{K}(\mathbf{x}_k, \mathbf{x}_k)$, $t \leftarrow 0$, $\alpha^0 = 0 \in \mathbb{R}^n$
repeat
    $\alpha \leftarrow \alpha^t$
    for each $k = 1, 2, \ldots, n$ do
        $\alpha_k \leftarrow \alpha_k + \eta_k(1 - y_k \sum_{i=1}^n \alpha_i y_i \tilde{K}(\mathbf{x}_i, \mathbf{x}_k))$ %update kth component of α
        if $\alpha_k < 0$ then $\alpha_k \leftarrow 0$
        if loss=hinge and $\alpha_k > C$ then $\alpha_k \leftarrow C$
    $\alpha^{t+1} \leftarrow \alpha$
    $t \leftarrow t + 1$
until $\|\alpha^t - \alpha^{t-1}\| \leq \epsilon$

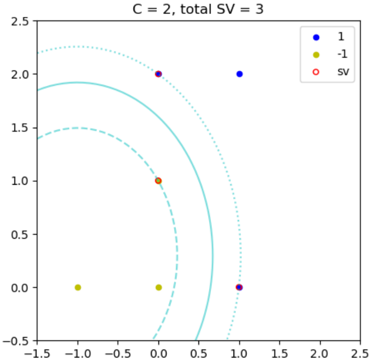
例7: 二次非齐次多项式核

X=np.array([[0,0],[-1,0],[0,1],[1,0],[0,2],[1,2]])

y=np.array([-1,-1,-1,1,1,1])

$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^2$

$\alpha = [0, 0, 1.317, 0.780, 0.537, 0]$, $b = -1.341$

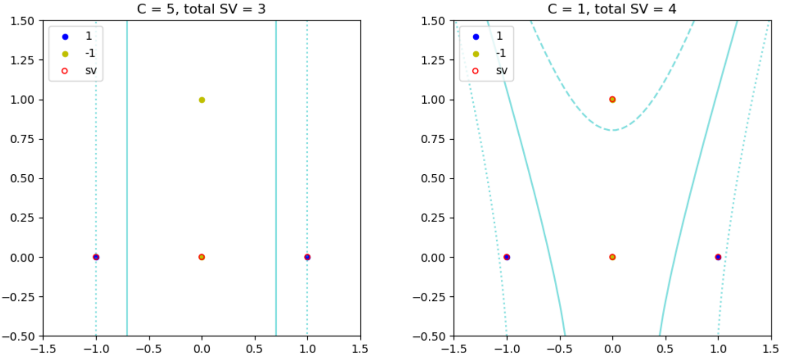


例8：二次非齐次多项式核

X=np.array([[0,0],[0,1],[-1,0],[1,0]]) y=np.array([-1,-1,1,1])

$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^2$

$C = 5$: $\alpha = [2,0,1,1]$; $C = 1$: $\alpha = [1, \frac{1}{4}, \frac{5}{8}, \frac{5}{8}]$



例9：异或问题–高斯核

x=np.array([[0,0],[1,1],[0,1],[1,0]]), y=np.array([-1,-1,1,1])

核支持向量机对偶问题模型

$$\min_{\alpha\in\mathbb{R}^n} \frac{1}{2}\alpha^\top H\alpha - \mathbf{1}^\top\alpha \text{ s.t. } \mathbf{y}^\top\alpha = 0, \mathbf{0} \le \alpha \le C\mathbf{1}$$

取高斯核函数$K(\mathbf{x},\mathbf{t}) := e^{-\|\mathbf{x}-\mathbf{t}\|^2}$可得

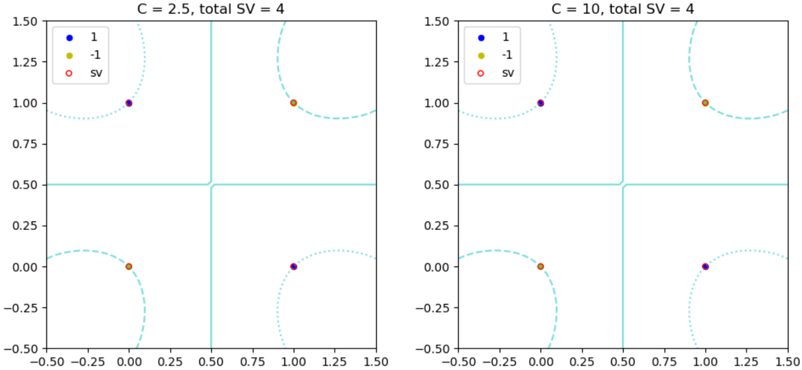
$$H = \text{diag}(\mathbf{y})\mathbf{K}\text{diag}(\mathbf{y}) = \begin{pmatrix} 1 & e^{-2} & -e^{-1} & -e^{-1} \\ e^{-2} & 1 & -e^{-1} & -e^{-1} \\ -e^{-1} & -e^{-1} & 1 & e^{-2} \\ -e^{-1} & -e^{-1} & e^{-2} & 1 \end{pmatrix}$$

由$\alpha_4 = \alpha_1 + \alpha_2 - \alpha_3$可得

$$L(\alpha_1,\alpha_2,\alpha_3) = (1-e^{-1})\alpha_1^2 + (1-2e^{-1}+e^{-2})\alpha_1\alpha_2 - (1-e^{-2})\alpha_1\alpha_3 + (1-e^{-1})\alpha_2^2$$

$$-(1-e^{-2})\alpha_2\alpha_3 + (1-e^{-2})\alpha_3^2 - 2(\alpha_1+\alpha_2) \text{ s.t. } 0 < \alpha_1,\alpha_2,\alpha_3 < C$$

例10：异或问题–高斯核[24]



若$C = 2.5$，则$\alpha = 2.5 * [1, 1, 1, 1]$，$b = 0$

若$C = 10$，则$\alpha = \frac{1}{(1-e^{-1})^2}[1, 1, 1, 1] = [2.50265, 2.50265, 2.50265, 2.50265]$，$b = 0$

[24] rbf_kernel:https://scikit-learn.org.cn/view/567.html

Python代码(通过qpsolvers)[25]

```
import numpy as np
from qpsolvers import Problem, solve_problem
from sklearn.metrics.pairwise import rbf_kernel
C=100    Xnp.array([[0,0],[1,1],[0,1],[1,0]])
y=np.array([-1,-1,1,1])    Y=np.diag(y)
K = rbf_kernel(X_train, Y=None, gamma=1)
P=np.dot(np.dot(Y,K),Y)    q = -1*np.ones(4)
A = np.array([[-1,-1,1,1]])    b = np.array([0])
G = np.vstack([-np.eye(4), np.eye(4)])    h = np.hstack([np.zeros(4),C*np.ones(4)])
problem = Problem(P, q, G, h, A, b)
solution = solve_problem(problem, solver='osqp')
alpha=solution.x    sv=np.where(np.logical_and(alpha > 1e-4,alpha<C))
# linear kernel: w=np.dot(X_train[sv].T,np.multiply(alpha[sv],y[sv]))
b=np.mean(y[sv] -np.dot(K[sv,:],np.multiply(alpha,y)))
```

[25] pip install qpsolvers: https://github.com/qpsolvers/qpsolvers

```
plt.rcParams['font.size'] = '10'    plt.rcParams['savefig.dpi'] = 300
plt.rcParams['figure.figsize'] = (5, 5)
id1 = np.where(y==1)    id2 = np.where(y==-1)
plt.scatter(X[id1,0], X[id1,1], marker = 'o', color = 'b', label='1', s = 20)
plt.scatter(X[id2,0], X[id2,1], marker = 'o', color = 'y', label='-1', s = 20)
ax = plt.gca()
xx, yy = np.meshgrid(np.linspace(-0.5,1.5, 50), np.linspace(-0.5, 1.5, 50))
K_classifier = rbf_kernel(np.c_[xx.ravel(), yy.ravel()],X_train, gamma=1)
Y_classifiervalue=np.dot(K_classifier,np.multiply(alpha,y) ) + b
Z=Y_classifiervalue.reshape(xx.shape)
plt.contour(xx,yy,Z,colors='c',levels=[-1, 0, 1], alpha=0.5,linestyles=['--', '-', ':'])
sv_points = X[sv]
plt.scatter(sv_points[:, 0], sv_points[:, 1], s=20, linewidth=1, facecolors='none',
edgecolors='r',label='sv')
plt.legend(loc = 'upper left')
title = 'C = ' + str(C) + ', total SV = ' + str(len(alpha[alpha > 10e-4]))
plt.title(title)
```

例11：异或问题–多项式核[26] $(\mathbf{x}^\top \mathbf{y}+1)^2$

H=[[1. 1. -1. -1.][1. 9. -4. -4.][-1. -4. 4. 1.][-1. -4. 1. 4.]], C=5 可得

$$L(\alpha_1,\alpha_2,\alpha_3)=\frac{1}{2}\left[3\alpha_1^2-6\alpha_1\alpha_3+5\alpha_2^2-6\alpha_2\alpha_3+6\alpha_3^2\right]-2(\alpha_1+\alpha_2)$$

$$\alpha=[\frac{10}{3},2,\frac{8}{3},\frac{8}{3}],\ \ b=-1$$

$$h(\mathbf{z})=\sum_{i=1}^{4}\alpha_i y_i(\mathbf{x}_i^\top\mathbf{z}+1)^2+b=\frac{2}{3}(z_1^2+z_2^2+2z_1+2z_2)-4z_1z_2-1,\ \ \mathbf{z}=(z_1,z_2)^\top\in\mathbb{R}^2$$

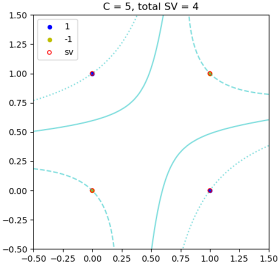
$$\phi(x_1,x_2)=(x_1^2,x_2^2,\sqrt{2}x_1x_2,\sqrt{2}x_1,\sqrt{2}x_2,1)$$

$$(0,0)\to(0,0,0,0,0,1)$$

$$(1,1)\to(1,1,\sqrt{2},\sqrt{2},\sqrt{2},1)$$

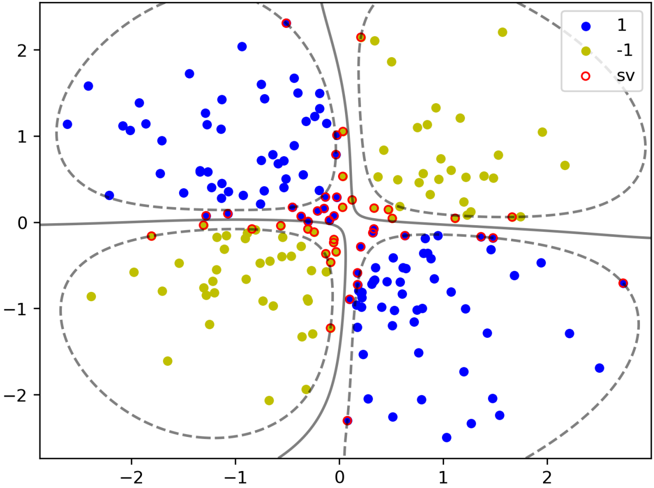
$$(0,1)\to(0,1,0,0,\sqrt{2},1)$$

$$(1,0)\to(1,0,0,\sqrt{2},0,1)$$



[26] polynomial_kernel: https://scikit-learn.org.cn/view/566.html

例12：异或数据集–高斯核



Python代码(通过sklearn.svm)

```
import random    import numpy as np    import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
X = np.random.randn(400,2)
y = np.logical_xor(X[:,0]>0,X[:,1]>0)    y = np.where(y,1,-1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.5,random_state=2)
clf=SVC(C=10,kernel='rbf').fit(X_train,y_train) % default='squared_hinge'
print('Train accuracy of KSVC:', clf.score(X_train,y_train))
print('Test accuracy of KSVC:', clf.score(X_test,y_test))
print('Number of SV for each class :', clf.n_support_)
print('Number of support vectors for KSVM:',np.sum(clf.n_support_))
print('Indices of support vectors for KSVM:', clf.support_)
print('Dual coefficients of the support vector:', clf.dual_coef_)
% sv_labels= clf.dual_coef_.ravel()>0
```

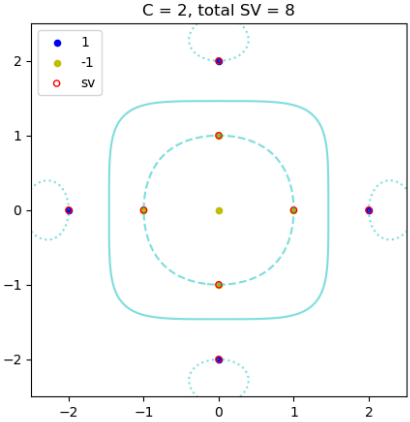
```
support_vectors =clf.support_vectors_
decision_function = clf.decision_function(X_train)
sv_indices_canonical = np.where(np.abs(decision_function) <= 1 + 1e-15)[0]
print('Total number of SV on canonical hyperplane:',sv_indices_canonical.shape[0])
id1 = np.where(y_train==1)
id2 = np.where(y_train==-1)
plt.scatter(X_train[id1,0], X_train[id1,1], marker = 'o', color = 'b', label='1', s = 20)
plt.scatter(X_train[id2,0], X_train[id2,1], marker = 'o', color = 'y', label='-1', s = 20)
ax = plt.gca()    xlim = ax.get_xlim()    ylim = ax.get_ylim()
xx, yy = np.meshgrid(np.linspace(xlim[0], xlim[1], 200), np.linspace(ylim[0], ylim[1],
200))
Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
plt.contour(xx,yy,Z,colors='k',levels=[-1, 0, 1], alpha=0.5,linestyles=['--', '-', '--'])
plt.scatter(support_vectors[:, 0], support_vectors[:, 1], s=20,linewidth=1,
facecolors='none', edgecolors='r', label='sv')
plt.legend(loc = 'upper right')
```

例13：圆环–高斯核

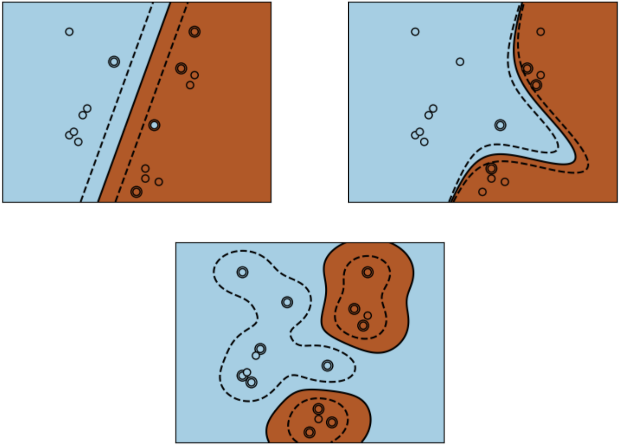
X=np.array([[0,0],[0,1],[1,0],[0,-1],[-1,0],[0,2],[0,-2],[2,0],[-2,0]])

y=np.array([-1,-1,-1,-1,-1,1,1,1,1])

alpha=[0,1.31,1.31,1.31,1.31,1.31,1.31,1.31,1.31], b=0.18887



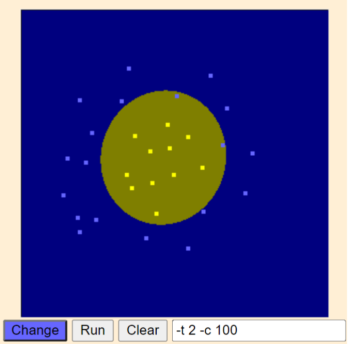
例14：sklearn数值案例 [27] [28]



[27] 代码：https://scikit-learn.org/stable/auto_examples/svm/plot_svm_kernels.html
[28] https://scikit-learn.org/stable/modules/svm.html：clf.coef_，clf.dual_coef_

动画演示

1. LIBSVM图形界面: https://www.csie.ntu.edu.tw/~cjlin/libsvm/




```
options:
-s svm_type : set type of SVM (default 0)
        0 -- C-SVC
        1 -- nu-SVC
        2 -- one-class SVM
        3 -- epsilon-SVR
        4 -- nu-SVR
-t kernel_type : set type of kernel function (default 2)
        0 -- linear: u'*v
        1 -- polynomial: (gamma*u'*v + coef0)^degree
        2 -- radial basis function: exp(-gamma*|u-v|^2)
        3 -- sigmoid: tanh(gamma*u'*v + coef0)
-d degree : set degree in kernel function (default 3)
-g gamma : set gamma in kernel function (default 1/num_features)
-r coef0 : set coef0 in kernel function (default 0)
-c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
-n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)
-p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)
-m cachesize : set cache memory size in MB (default 100)
-e epsilon : set tolerance of termination criterion (default 0.001)
-h shrinking: whether to use the shrinking heuristics, 0 or 1 (default 1)
-b probability_estimates: whether to train a SVC or SVR model for probability estimates
-wi weight: set the parameter C of class i to weight*C, for C-SVC (default 1)
```

2. Andrej Karpathy: RBF-SVM动画演示

https://cs.stanford.edu/~karpathy/svmjs/demo/

https://github.com/karpathy/svmjs

4. 拓展：最小二乘核支持向量机[29][30]

目标函数: $$\min_{\mathbf{w}\in\mathcal{F},b,\xi_i}\left\{\frac{\|\mathbf{w}\|^2}{2}+C\sum_{i=1}^{n}\xi_i^2\right\}$$

线性约束: $y_i(\mathbf{w}^\top\phi(\mathbf{x}_i)+b)=1-\xi_i, \forall i=1,2,\ldots,n$

等价于无约束优化问题

$$\min_{\mathbf{w}\in\mathcal{F},b}\frac{1}{2}\|\mathbf{w}\|^2+C\frac{1}{2}(1-y_i(\mathbf{w}^\top\phi(\mathbf{x}_i)+b))^2.$$

求解线性方程组:

$$\begin{bmatrix}0 & \mathbf{y}\\ \mathbf{y} & H+\frac{1}{C}I\end{bmatrix}\begin{bmatrix}b\\ \alpha\end{bmatrix}=\begin{bmatrix}0\\ \mathbf{1}\end{bmatrix}$$

其中$H:=[H_{i,j}=y_iK(\mathbf{x}_i,\mathbf{x}_j)y_j:i,j=1,2,\ldots,n]$.

[29] J.A.K. Suykens, J. Vandewalle, Least Squares Support Vector Machine Classifiers, Neural Processing Letters, 9(1999), 293–300

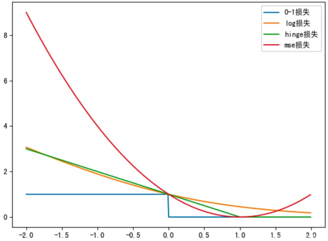
[30] Mingzhen He, Fan He, Lei Shi, Xiaolin Huang, Johan A.K. Suykens, Learning with Asymmetric Kernels: Least Squares and Feature Interpretation, Feb 2022

拓展：$\ell_{0/1}$损失的核支持向量机[31][32]

$$\min_{\mathbf{w}\in\mathcal{F},b\in\mathbb{R}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\ell_{0/1}(1-y_i(\mathbf{w}^\top\phi(\mathbf{x}_i)+b))$$

其中分类损失$\ell_{0/1}(t)=1$若$t\neq 0$且$\ell_{0/1}(0)=0$.

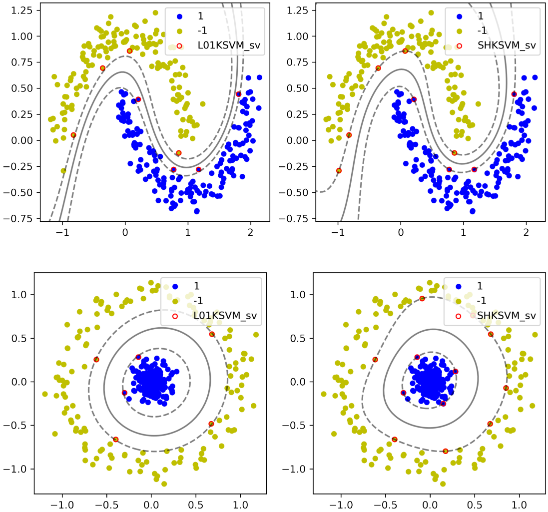
难点：$\ell_{0/1}$非凸，不连续，离散



[31] J. Tang, N. Zhang, Q. Li, Robust binary classification via ℓ_0-svm, in 2018 IEEE International Conference on Data Mining Workshops, 2018, 1263-1270

[32] Huajun Wang et al., Support Vector Machine Classifier via $L_{0/1}$ Soft-Margin Loss, IEEE Trans Pattern Anal Mach Intell. 2022

Python实验: Double Moons, Double Circles[33]



[33] 论文Python代码: https://github.com/Rongrong-Lin/L0KSVM

机器学习或优化模型的表示定理(Representer Theorem)

损失函数：$L(y,y')$ 满足$L(y,y') \in [0,+\infty)$且$L(y,y)=0$对任意的y,y'. 如

$$L(y,y') = |y-y'|^2, y,y' \in \mathbb{R}, L(y,y') = \max\{0, 1-yy'\}, y,y' \in \{-1,1\}.$$

机器学习或优化模型的表示定理(Representer Theorem)

损失函数：$L(y,y')$ 满足$L(y,y')\in[0,+\infty)$且$L(y,y)=0$对任意的y,y'. 如

$$L(y,y')=|y-y'|^2, y,y'\in\mathbb{R}, L(y,y')=\max\{0,1-yy'\}, y,y'\in\{-1,1\}.$$

Theorem (表示定理)

若L是一个损失函数，$\mathcal{F}$是一个希尔伯特空间，则

$$\min_{\mathbf{w}\in\mathcal{F}} J(\mathbf{w})=\sum_{i=1}^{m} L(y_i,\langle \mathbf{w},\phi(\mathbf{x}_i)\rangle_{\mathcal{F}}+b)+\frac{1}{2}\|\mathbf{w}\|_{\mathcal{F}}^2$$

的最优解具有如下形式

$$\mathbf{w}=\sum_{i=1}^{m} c_i\phi(\mathbf{x}_i).$$

注：上述表示定理适用于岭回归、核岭回归和三种支持向量机算法等

证明

任意的$\mathbf{w} \in \mathcal{F}(\mathcal{F} = V \oplus V^{\perp})$ 可以正交分解为

$$\mathbf{w} = \mathbf{w}_1 + \mathbf{w}_2,\ \mathbf{w}_1 \in V := \mathrm{span}\{\phi(\mathbf{x}_i) : i = 1, 2, \ldots, n\}, \mathbf{w}_2 \in V^{\perp}.$$

可得

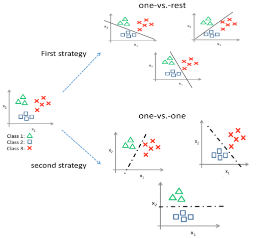
$$\langle \mathbf{w}_2, \phi(\mathbf{x}_i) \rangle_{\mathcal{F}} = 0, \forall i = 1, 2, \ldots, n.$$

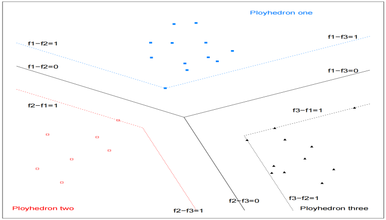
计算

$$\begin{aligned}
J(\mathbf{w}) &= \sum_{i=1}^{m} L(y_i, \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle_{\mathcal{F}} + b) + \frac{1}{2}\|\mathbf{w}\|_{\mathcal{F}}^2 \\
&= \sum_{i=1}^{m} L(y_i, \langle \mathbf{w}_1 + \mathbf{w}_2, \phi(\mathbf{x}_i) \rangle_{\mathcal{F}} + b) + \frac{1}{2}\|\mathbf{w}_1 + \mathbf{w}_2\|_{\mathcal{F}}^2 \\
&= \sum_{i=1}^{m} L(y_i, \langle \mathbf{w}_1, \phi(\mathbf{x}_i) \rangle_{\mathcal{F}} + b) + \frac{1}{2}\|\mathbf{w}_1\|_{\mathcal{F}}^2 + \frac{1}{2}\|\mathbf{w}_2\|_{\mathcal{F}}^2 \\
&\geq \sum_{i=1}^{m} L(y_i, \langle \mathbf{w}_1, \phi(\mathbf{x}_i) \rangle_{\mathcal{F}} + b) + \frac{1}{2}\|\mathbf{w}_1\|_{\mathcal{F}}^2 = J(\mathbf{w}_1)
\end{aligned}$$

拓展：K分类支持向量机[34]

- **One-vs-rest方法**。解决K个二分类问题，将第k类与其他类分类，$k=1,2,\ldots,K$. [Ryan Rifkin, Aldebaro Klautau, In Defense of One-Vs-All Classification, JMLR 5 (2004) 101-141]
- **one-vs-one方法**。将第k类与第j类分开，$1\le j\neq k\le K$. [S. Park， J. Furnkranz, Efficient Pairwise Classification, ECML 2007, 17-21]
- 多任务学习。学习一个向量值函数。





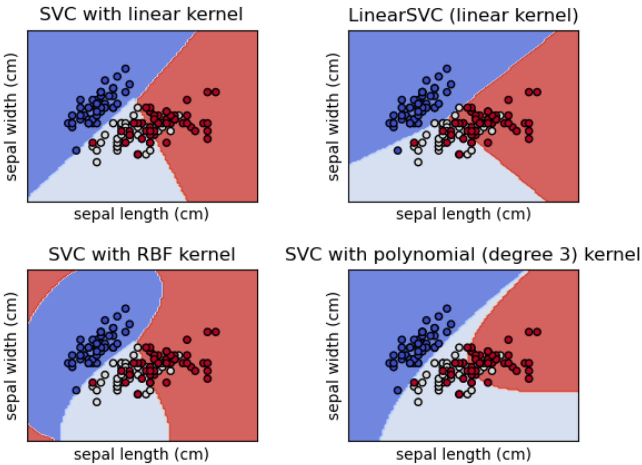
[34] Mohri Mehryar, Afshin Rostamizadeh, Ameet Talwalkar, Foundations of Machine Learning, Second Edition, MIT, 2018, Chapter 9

注释[35]

- One-vs-rest方法学习K个分类器，并且每一个分类器f_y 学习都需要用到整个数据，可得
$$\hat{y} = \arg \max_{y \in \{1,2,\ldots,K\}} f_y(\mathbf{x})$$
- one-vs-one方法学习$K(K-1)/2$个分类器， 每个分类器$f_{y,y'}$的学习需要用到y和y'两类对应的数据。最终的预测需要依次根据分类器的结果投票。
- 多任务学习的相关矩阵维数为Kn, 当数据个数n比较大时候通常会造成计算困难。
- 其他变体: Directed Acyclic Graph SVM [Platt et al., 2000], Crammer Singer SVM [Crammer and Singer, 2002]

[35]SVM类型综述：V.K. Chauhan, K. Dahiya, A. Sharma, Problem formulations and solvers in linear SVM: a review[J]. Artificial Intelligence Review, 2019, 52(2): 803-855.

多分类核支持向量机：鸢尾花[36]



[36] https://scikit-learn.org/stable/auto_examples/svm/plot_iris_svc.html

了解: 大规模数据的支持向量算法

基于核学习需要用到核矩阵，其数据规模影响存储和计算复杂度。

- Nyström方法 [37]: Williams C, Seeger M. Using the Nyström method to speed up kernel machines[J]. Advances in Neural Information Processing Systems, 2000.
- 随机傅里叶特征: T. Yang, Y. Li, M. Mahdavi, R. Jin, Z. Zhou, Nystroem Method vs Random Fourier Features: A Theoretical and Empirical Comparison, Advances in Neural Information Processing Systems, 2012.
- 边界识别: Tang L, Tian Y, Wang X, et al. A simple and reliable instance selection for fast training support vector machine: Valid Border Recognition [J]. Neural Networks, 2023.
- 在线学习: Tarres P, Yao Y. Online learning as stochastic approximation of regularization paths: Optimality and almost-sure convergence[J]. IEEE Transactions on Information Theory, 2014, 60(9): 5716-5735.

[37] https://scikit-learn.org/stable/modules/generated/sklearn.kernel_approximation.Nystroem.html: Nystroem, RbFSampler

支持向量机的优点[39]

$$\min_{\mathbf{w},b}\left\{\frac{\|\mathbf{w}\|^2}{2}+C\sum_{i=1}^{n}(1-y_i(\mathbf{w}^\top\mathbf{x}_i+b))_+\right\}$$

- 寻找具有最大间隔的超平面，具有很好的泛化误差;
- 核技巧使得支持向量机可以高效地解决非线性问题;
- 是一种非参数化方法[38]

课题: 稀疏支持向量机($\|\mathbf{w}\|_1$)、支持矩阵机(输入为矩阵)、p范数支持向量机($\|\mathbf{w}\|_p$, $1 \le p < +\infty$), 最小二乘支持向量机(最小二乘损失)、 孪生支持向量机回归(非平行平面); 0 范数支持向量机($\|\mathbf{w}\|_0$), 带非凸损失函数的支持向量机。

[38] 参数化方法的参数数量是固定的(强的假设先验), 不随着训练样本数量的变化而变化。 例如线性回归、线性判别分析、朴素贝叶斯、MLP、CNN等算法都是参数化方法。而k近邻、 SVM、 分片线性函数、decision tree等, 都是非参数化方法.

[39] Stuart J.Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, 2020, Subsection 19.7.5

总结：支持向量机

1. 核技巧和特征映射：$K(\mathbf{x},\mathbf{y})=\phi(\mathbf{x})^\top\phi(\mathbf{y})$(半正定的)
2. 线性支持向量机(SVM): 最大间隔超平面
3. 核支持向量机(Kernel SVM): 特征空间中的最大间隔超平面(在输入空间是曲面)
4. 对偶问题矩阵形式: $H=\mathrm{diag}(\mathbf{y})\mathbf{K}\mathrm{diag}(\mathbf{y})$且核矩阵$\mathbf{K}=[K(\mathbf{x}_i,\mathbf{x}_j):1\le i,j\le n]$

$$\boxed{\min_{\alpha\in\mathbb{R}^n}\frac{1}{2}\alpha^\top H\alpha-\mathbf{1}^\top\alpha,\ \text{s.t.}\ \mathbf{0}\le\alpha\le C\mathbf{1},\mathbf{y}^\top\alpha=0}$$

   (i) 硬间隔SVM: K为线性核且$C=+\infty$; (ii) 软间隔SVM: K为线性核且$C<\infty$; (iii) 核/非线性SVM: K为核函数且$C<\infty$

5. 数值[40]: scikit-learn[41]、SMO算法[42] 、二次规划[43]、ADMM、对偶随机梯度下降

不等式约束优化问题 $\xLeftrightarrow{\text{拉格朗日乘子法}}$ 无约束优化问题 $\xLeftrightarrow{\text{对偶性}}$ 对偶问题

[40] https://github.com/Rongrong-Lin/DataMining: Topic4Lecture3_SVM.ipynb
[41] sklearn.svm.SVC: https://scikit-learn.org.cn/view/781.html
[42] SMO algorithm: https://github.com/Kaslanarian/PySVM
[43] pip install qpsolvers: https://github.com/qpsolvers/qpsolvers

作业1[44]

(1)请建立原始的线性支持向量机模型并手动计算最优分割超平面

(2)Python编程实现

(3) 画图: 数据点、分离超平面、间隔边界及支持向量

标签	数据点
1	$(1,2)$, $(2,3)$, $(3,3)$
-1	$(2,1)$, $(3,2)$

感谢大家的聆听!

欢迎交流和讨论!

[44]李航，机器学习方法、第三版，清华大学出版社，2022，第7章, 练习7.2

支持向量机的教材

- Mohammed J. Zaki, Wagner Meira Jr., Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd Edition, CUP, 2020, Chapter 21
- Mohri Mehryar, Afshin Rostamizadeh, Ameet Talwalkar, Foundations of Machine Learning, Second Edition, MIT, 2018, Chapters 5-6, Exercise 5.4 (Linear SMO), Exercise 15.4 (Nyström method)
- B. Schölkopf, A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond, MIT, 2002, Chapters 7-10
- M. Gopal, Applied Machine Learning, McGraw Hill Edition, 2018, Chapter 4.
- Ethem Alpaydin, Introduction to Machine Learning, Third Edition, MIT 2014
- Andrea Giussani, Applied Machine Learning with Python, 2020, Section 2.5
- G. James, D. Witten, T. Hastie, R. Tibshirani, J. Taylor, An Introduction to Statistical Learning with Applications in Python, 2023
- 李航，机器学习方法、第三版，清华大学出版社，2022，第7章