LN10. 线性支持向量机 (SVM)

李钦宾

先进智能计算与系统团队

邮箱：*qinbin@hust.edu.cn*

2025 月 3 月

目 录

目 录

基本思想:

支持向量机 (Support Vector Machine, SVM) 是一种线性分类器，可以被视为感知机的扩展。如果数据线性可分，感知器可保证找到**某个**超平面，SVM 则找到具有**最大间隔**的分离超平面。

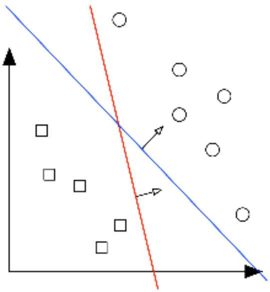


图: 同一数据集两个不同的分离超平面

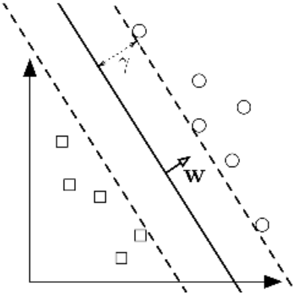


图: 最大间隔超平面

定义:

- 数据集: $\mathrm{D} = \{(\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2),...,(\mathbf{x}_N,\mathbf{y}_N)\}$
- 二元分类标签: $\mathbf{y}_i \in \{-1,+1\}$, $i = 1,...,N$
- 线性分类器: $h(x) = sign(\mathrm{w}^T\mathbf{x} + \mathbf{b})$

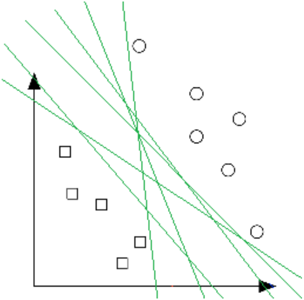


图: 同一个数据集有许多分离超平面

关于感知机的回顾: 如果数据是线性可分的，我们可以通过感知机找到许多不同的超平面。

问题: 什么是最好的分离超平面?

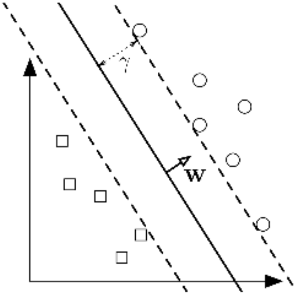


图: 最大间隔 (Margin) 超平面

SVM 的回答: 对两类间到超平面的最小距离最大化，即具有**最大间隔**的超平面。如上图所示，间隔 γ 是从超平面 (实线) 到两个类中最近的点 (平行虚线) 的距离。如果超平面是使 γ 最大的，它必然位于两个类的正中间。

目 录

间隔 (Margin):

- 超平面: $\mathbf{H} = \{ \mathbf{x} | \mathbf{w}^T\mathbf{x} + \mathbf{b} = 0\}$
- 从超平面到两类中最近点的距离: γ

间隔 (Margin):

考虑某个点 $\mathbf{x}$。设 $\mathbf{d}$ 为从超平面 $\mathbf{H}$ 到 $\mathbf{x}$ 的具有最小长度的向量。设 $\mathbf{x}^P$ 为 $\mathbf{x}$ 在 $\mathbf{H}$ 上的投影。则有:

$$\mathbf{x}^P = \mathbf{x} - \mathbf{d}$$

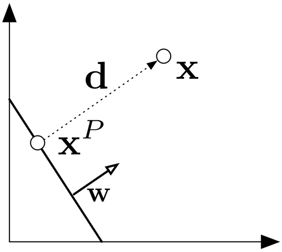
$\mathbf{d}$ 平行于 $\mathbf{w}$ ，因此 $\mathbf{d} = \alpha\mathbf{w}$, $\alpha \in \mathbb{R}$。

由 $\mathbf{x}^P \in \mathcal{H}$ 可知, $\mathbf{w}^T\mathbf{x}^P + b = 0$

所以

$\mathbf{w}^T\mathbf{x}^P + b = \mathbf{w}^T(\mathbf{x} - \mathbf{d}) + b = \mathbf{w}^T(\mathbf{x} - \alpha\mathbf{w}) + b = 0$

推导得到: $\alpha = \frac{\mathbf{w}^T\mathbf{x} + b}{\mathbf{w}^T\mathbf{w}}$



间隔

间隔 (Margin):

由上页，我们得出 $\alpha = \frac{\mathbf{w}^T\mathbf{x}+b}{\mathbf{w}^T\mathbf{w}}$， $\mathbf{d} = \alpha\mathbf{w}$

$\mathbf{d}$ 的模长:

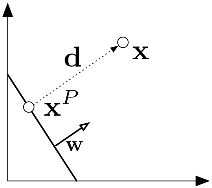
$$\|\mathbf{d}\|_2 = \sqrt{\mathbf{d}^T\mathbf{d}} = \alpha\sqrt{\mathbf{w}^T\mathbf{w}} = \frac{|\mathbf{w}^T\mathbf{x}+b|}{\sqrt{\mathbf{w}^T\mathbf{w}}} = \frac{|\mathbf{w}^T\mathbf{x}+b|}{\|\mathbf{w}\|_2}$$

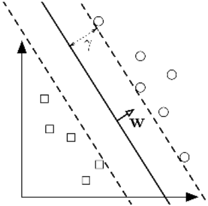
$\mathcal{H}$ 相对于 $\mathbf{D}$ 的间隔距离:

$$\gamma(\mathbf{w}, b) = \min_{\mathbf{x}\in D} \frac{|\mathbf{w}^T\mathbf{x}+b|}{\|\mathbf{w}\|_2}$$

根据定义，间隔相对于超平面具有伸缩不变性，即:

$$\gamma(\beta\mathbf{w}, \beta b) = \gamma(\mathbf{w}, b), \forall\beta \neq 0$$





可以将我们对于最大间隔分离超平面的搜索，表述为一个约束优化问题。目标是在所有数据点必须位于超平面正确一侧的约束下，使间隔最大化:

$$\underbrace{\max_{\mathbf{w},b} \gamma(\mathbf{w}, b)}_{maximize\ \ margin} \quad s.t. \quad \underbrace{\forall i\ y_i(\mathbf{w}^T x_i + b) \geq 0}_{separating\ \ hyperplane}$$

如果添加 γ 的定义，我们得到:

$$\underbrace{\max_{\mathbf{w},b} \underbrace{\frac{1}{\|\mathbf{w}\|_2} \min_{\mathbf{x}_i \in D} \left|\mathbf{w}^T \mathbf{x}_i + b\right|}_{\gamma(\mathbf{w},b)}}_{maximize\ \ margin} \quad s.t. \quad \underbrace{\forall i\ y_i(\mathbf{w}^T x_i + b) \geq 0}_{separating\ \ hyperplane}$$

最大间隔分类器

$$\underbrace{\max_{\mathbf{w},b} \underbrace{\frac{1}{\|\mathbf{w}\|_2} \min_{\mathbf{x}_i \in D} \left|\mathbf{w}^T \mathbf{x}_i + b\right|}_{\gamma(\mathbf{w},b)}}_{maximize \;\; margin} \quad s.t. \quad \underbrace{\forall i \; y_i(\mathbf{w}^T x_i + b) \geq 0}_{separating \;\; hyperplane}$$

由于超平面具有伸缩不变性：$\gamma(\beta\mathbf{w}, \beta b) = \gamma(\mathbf{w}, b), \forall\beta \neq 0$，我们可以固定 $\mathbf{w}, b$ 的伸缩程度，可以这样选择:

$$\min_{\mathbf{x}\in D} \left|\mathbf{w}^T \mathbf{x} + b\right| = 1.$$

我们可以将这种重新缩放作为等式约束。那么我们的目标就是:

$$\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|_2} \cdot 1 = \min_{\mathbf{w},b} \|\mathbf{w}\|_2 = \min_{\mathbf{w},b} \mathbf{w}^\top \mathbf{w}$$

从上面的目标我们知道这是一个凸二次函数。

由上面讨论，新的优化问题为:

$$\min_{\mathbf{w},b} \mathbf{w}^\top \mathbf{w}$$

$$\text{s.t.} \quad \forall i, y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 0 \ ,$$

$$\min_i \left| \mathbf{w}^T \mathbf{x}_i + b \right| = 1$$

然后，可以合并约束条件得到一个更简单的公式:

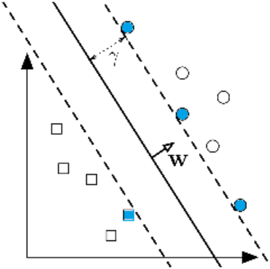
$$\min_{\mathbf{w},b} \mathbf{w}^\top \mathbf{w}$$

$$\text{s.t.} \quad \forall i \ y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$$

这个新公式是一个二次优化问题。目标是二次的，约束条件都是线性的。可以用 QCQP (Quadratic Constrained Quadratic Program) 求解器来有效地求解，得到唯一解。

基于超平面的伸缩不变性，可以找到最简单的超平面 (其中更简单意味着更小的 $\mathbf{w}^\top \mathbf{w}$)，这样所有输入数据在超平面的正确一侧，且距离超平面至少 1 个单位。

支持向量



由于超平面具有缩放不变性，我们可以重新缩放 $\mathbf{w}$，b，使所有点到它的距离至少为 1 个单位。因此，对于最优的 $\mathbf{w}$, b 对，一些训练点将有严格的约束，即:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1.$$

上图中可以清楚地看到这些蓝色的点，恰位于虚线上。我们将这些训练数据点称为**支持向量**。支持向量是特殊的点，因为它们是定义超平面到数据集最大间隔的训练点，决定了超平面的形状。如果移动其中一个并重新训练 SVM，生成的超平面将会改变。

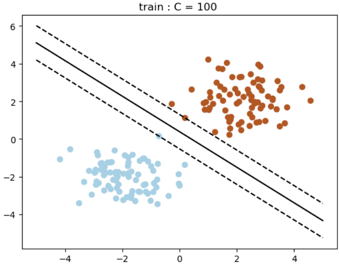


图: Training result

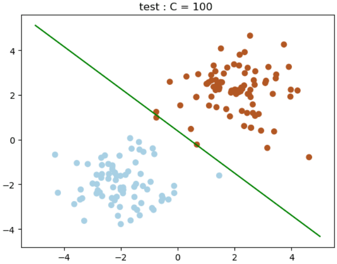


图: Test result

SVM code click here

目 录

如果是低维数据或数据中有噪声，通常情况下，这两类数据间没有可分离的超平面。

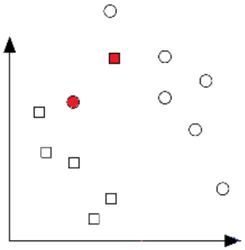
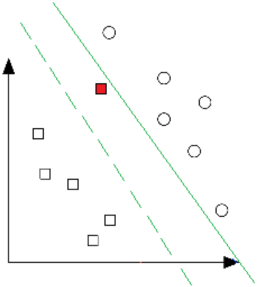


图: 非线性可分

例如，上图中，红色的点可能是噪声点或点的标签是错误的，显然，此优化问题没有解。

软约束 SVM 概述：



图：可能包含噪音

另一种情形如上图。可以找到线性可分离超平面 (实线)。但是这个超平面的间隔太小了。因此，我们可能会认为红点是噪音或其标签是错误的。

如果不考虑红点，可能会找到另一个更好的超平面 (虚线)，这估计也是最好的超平面。

无约束假设

未匹配计数损失 (misMatched Count Loss)

对于上面讨论的情况，可以忽略这些不匹配的噪声点，并将它们视为损失。

计算不匹配点的数量，并将它们添加到目标中，可得:

$$\min_{\mathbf{w},b} \mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}[y_n \neq sign(\mathbf{w}^T\mathbf{x} + \mathbf{b})]$$

$$s.t. \quad \begin{array}{lr} y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, & \textit{if } y_i \textit{ is correct} \\ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq -\infty, & \textit{if } y_i \textit{ is incorrect} \end{array} \tag{1}$$

在约束 (1) 中，如果 y_i 是不正确的，我们不对噪声点添加任何约束。

在上面的目标中，C 是用于**大间隔和噪声容忍**的权衡

在上一页中，我们定义了未匹配计数损失 [·] 来计算错误匹配的数量。然而，它是非线性和不连续的。这对我们的计算很不方便。因此，我们使用线性约束松弛变量 ξ_i 来记录间隔违反的程度而非不匹配的计数:

$$\min_{\mathbf{w},b} \mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n} \xi_i$$

$$\text{s.t.} \qquad \forall i \; y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \;, \forall i \; \xi_i \geq 0$$

松弛变量 ξ_i 允许输入 $\mathbf{x}_i$ 更接近超平面 (甚至在错误的一侧)，但在目标函数中对这种“松弛”有惩罚。

- 如果 C 非常大，SVM 会变得非常严格，并试图让所有点都在超平面正确的一侧。
- 如果 C 非常小，SVM 会变得非常松散，可能会“牺牲”一些点来获得一个更简单的解 (即更低的 $\|\mathbf{w}\|_2^2$)。

无约束假设:

Loss ξ_i

让我们考虑在 $C \neq 0$ 的情况下 ξ_i 的值。可以考虑 ξ_i 作为损失，目标总是尽可能地最小化 ξ_i，则有:

$$\xi_i = \begin{cases} 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b), & \textit{if } y_i(\mathbf{w}^T\mathbf{x}_i + b) < 1 \\ 0, & \textit{if } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1. \end{cases} \tag{2}$$

这等价于下面的形式:

$$\xi_i = \max(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b), 0).$$

无约束假设

无约束假设

如果将这个形式加入到 SVM 优化问题的目标中，可得到如下无约束版本，作为损失函数和正则项:

$$\min_{\mathbf{w},b} \underbrace{\mathbf{w}^T\mathbf{w}}_{l_2-regularizer} + C\sum_{i=1}^{n} \underbrace{\max\left[1-y_i(\mathbf{w}^T\mathbf{x}+b),0\right]}_{hinge-loss}$$

该公式允许我们优化 SVM 的参数 $(\mathbf{w}, b)$，就像逻辑回归 (例如通过梯度下降) 一样。唯一的区别是我们用的是 **hinge-loss** 而不是 **logistic loss**。

软间隔 SVM 的实例

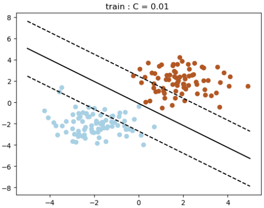


图: train : $C = 0.01$

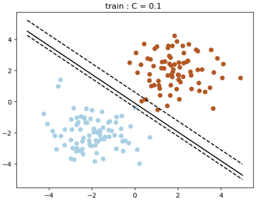


图: train : $C = 0.1$

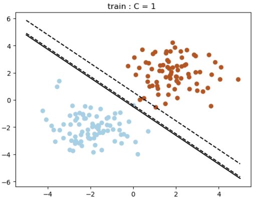


图: train : $C = 1$

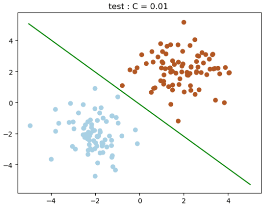


图: test : $C = 0.01$

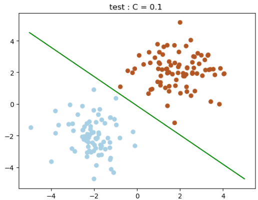


图: test : $C = 0.1$

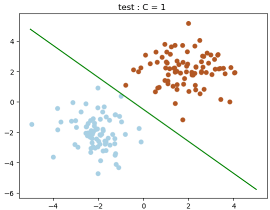


图: test : $C = 1$

软间隔 SVM 实例

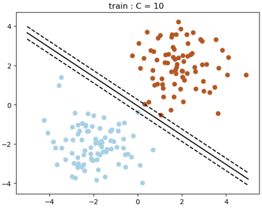


图: train : $C = 10$

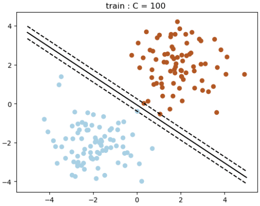


图: train : $C = 100$

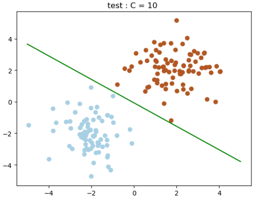


图: test : $C = 10$

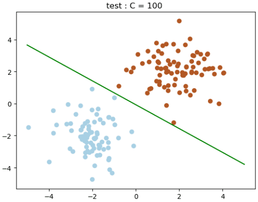


图: test : $C = 100$

软间隔 SVM 实例

从上面的示例图中，可以看到不同的 C 可能得到不同的分离超平面。如果 C 非常大，SVM 会变得非常严格，并试图让所有点都在超平面的右侧。但这可能会导致过拟合。如果 C 非常小，SVM 就会变得非常松弛，可能会“牺牲”一些点来获得一个更简单的解决方案 (即更小的 $\|\mathbf{w}\|_2^2$)。

Soft SVM code click here

目　录

SVM 总结

- 支持向量机 (SVM) 是一种线性分类器，可以被视为感知机的扩展。SVM 找到分离超平面的最大间隔。
- 间隔是超平面到两个类中最近点的距离。
- 最大间隔分类器是在所有数据点必须位于超平面正确一侧的约束下，使间隔最大化。
- 对于最优的 $\mathbf{w}, b$ 对，一些训练点将有严格的约束，即 $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$，称为支持向量。
- 支持向量机是凸二次函数，可以用 QCQP 求解器求解。
- 在目标中加入松弛变量，可以得到无约束支持向量机公式: 带软约束的支持向量机。

The End