
最优化理论课程报告

基于 Zoutendijk 方法的 SVM 训练器设计



指导老师：_____李瑞_____

学生姓名：_____曹文旭_____

学号：_____202121060129_____

学院：_____自动化工程学院_____

专业：_____控制科学与工程_____

一 实验背景

SVM(Support Vector Machine, 支持向量机)是一类经典的机器学习算法, 一个最基础的支持向量机可以实现线性二分类, 随着机器学习理论的发展, 在此基础上又出现了可以实现非线性分类的核函数 SVM, 以及适应不易分模型的软间隔 SVM, 还有用于回归问题的 SVR(Support Vector Regression, 支持向量回归)模型等等.

SVM 的实现思路是, 对于一个二分类问题, 在给定的正负样本集上, 找到一个最佳的分界面, 在保证把两类样本划分开的基础上, 使两类样本分别与该分界面的距离最大, 而解决这个问题(下称 SVM 问题)的优化器称作 SVM 训练器. SVM 问题的本质是最优化问题, 并且一个基础的 SVM 问题是凸优化问题, 对于该问题的求解有两种基本方案: 直接对原问题求解或对其对偶问题进行求解.

然而, 自从 SVM 问世, 一些更加高效的 SVM 训练算法层出不穷, SMO (Sequential Minimum Optimization)就是一个经典的代表. 现在, 台湾大学林智仁教授团队已经开发出了一个专门用于求解 SVM 与 SVR 的软件包 libsvm, 该软件包支持数十种编程语言, 为科研人员的研究工作提供了优秀且方便的支持.

二 实验内容

尽管 SVM 问题的求解方法已经获得了充分且成熟的研究, 现在我们已经可以方便地通过软件库快速地搭建属于自己的 SVM 平台, 但是这些软件库为我们节省大量脑力和体力的同时也向我们屏蔽了太多求解算法的细节过程, 而有时钻研和学习这些算法细节能够帮助我们加深对机器学习理论的理解和对最优化算法的理解, 给我们带来实践经验和科研灵感.

本次实验的切入点是一个基础的软间隔 SVM 问题, 在对该问题的求解过程中, 本文首先阐述了该问题的数学表达, 然后给出了该问题的凸二次优化标准型, 然后基于 python, 使用 Zoutendijk 方法作为主迭代算法, 配合黄金分割法实现的一维搜索函数以及线性规划函数实现了一个简易的软间隔 SVM 训练器.

在本实验的最后, 通过产生随机正负样本来测试所设计 SVM 训练器的实际运行效果, 并作出了相应的图示表达和数据分析.

三 SVM 原理

3.1 基础 SVM 原理

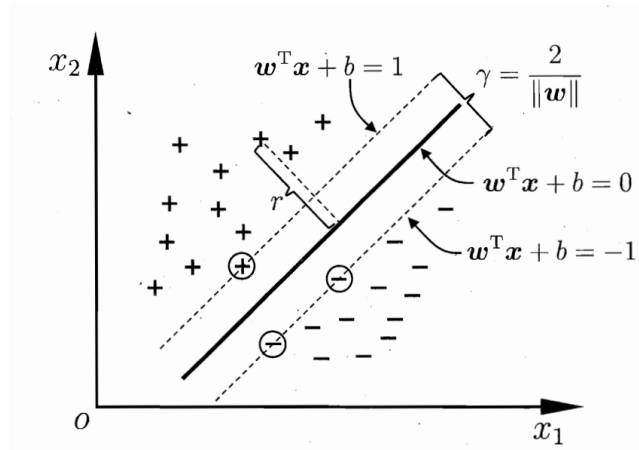


图 3-1 SVM 问题图式化描述

如图 3-1 所示, 设正负样本共有 m 个, 且 $\mathbf{x}_i \in \mathbb{R}^n, i=1,2,\dots,m$, 正样本的标签为 $y_i=1$, 负样本的标签为 -1 . 一个基本的线性分类的 SVM 问题可以被描述为: 在正负样本之间, 寻找一个分界面 $\mathbf{w}^T \mathbf{x} + b = 0$, 使得不同类的样本分别落在与该分界面的距离为 $1/\|\mathbf{w}\|$ 的两侧范围之外, 而我们将这个距离最大化, 所以这个基本的 SVM 问题可以被描述为问题 A:

$$\begin{aligned} \min_{\mathbf{w}, b} f(\mathbf{w}, b) &= \frac{1}{2} \|\mathbf{w}\|^2 = \frac{1}{2} \mathbf{w}^T \mathbf{w} \\ \text{s.t. } y_i (\mathbf{w}^T \mathbf{x}_i + b) &\geq 1, i=1, 2, \dots, m \end{aligned} \quad (1)$$

对一个优化问题 $\min f(\mathbf{x})$ 的求解可以转化为对其对偶问题 $\max \varphi(\boldsymbol{\alpha})$ 的求解, 一个优化问题与其对偶问题之间总是存在弱对偶关系, 即 $\min f(\mathbf{x}) \geq \max \varphi(\boldsymbol{\alpha})$, 一般而言, 凸优化问题与其对偶问题之间存在强对偶关系(等价关系), 即 $\min f(\mathbf{x}) = \max \varphi(\boldsymbol{\alpha})$. 问题 A 的目标函数的 Hesse 矩阵是单位阵, 所以问题 A 是一个凸优化问题, 具有强对偶性, 所以我们可以将对问题 A 的求解转化为对其对偶问题的求解. 首先引入拉格朗日乘数 $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_m]^T$, 得到

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^m \alpha_i [y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1] \quad (2)$$

问题 A 的对偶问题可以表示为问题 B

$$\begin{aligned} \max_{\alpha} \varphi(\alpha) &= \max_{\alpha} \min_{\omega, b} L(\omega, b, \alpha) \\ s. t. \quad &\alpha_i > 0, i = 1, 2, \dots, m \end{aligned} \quad (3)$$

为了得到问题 B 的具体表达形式，需要令 $L(\omega, b, \alpha)$ 分别对 ω, b 的偏导数为 0

$$\begin{aligned} \frac{\partial L}{\partial \omega} &= \omega - \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i = 0 \\ \frac{\partial L}{\partial b} &= \sum_{i=1}^m \alpha_i y_i = 0 \end{aligned} \quad (4)$$

将(4)代入(2)可得:

$$\varphi(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

考虑(4)中表示的约束关系，我们就得到了问题 B 的具体表达形式

$$\begin{aligned} \max_{\alpha} \varphi(\alpha) &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ s. t. \quad &\alpha_i \geq 0, \sum_{i=1}^m \alpha_i y_i = 0, \\ &i = 1, 2, \dots, m \end{aligned} \quad (5)$$

3.2 软间隔 SVM 原理

然而当正负样本之间存在重叠时，(1)中的约束条件必然不可能成立，所以这里需要引入软间隔的概念，也就是允许正负样本进入距离分界面 $1/\|\omega\|$ 的间隔内部甚至超出间隔进入另一侧，对于这些临界和越界的样本，在目标函数中增加相应的惩罚项，使得在优化的过程中，这样的情况被尽可能减少。

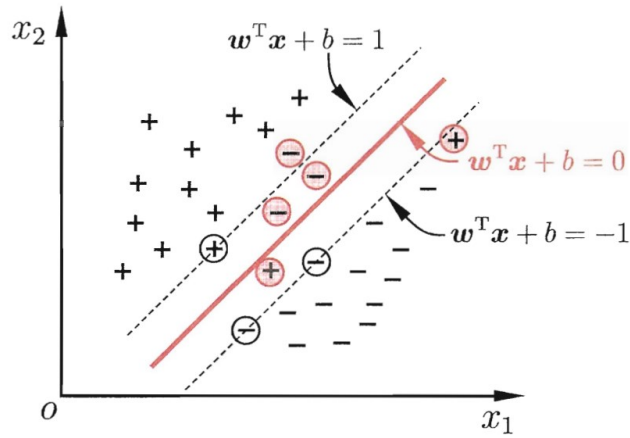


图 3-2 软间隔 SVM

一个基础的软间隔 SVM 问题可以被描述为问题 C:

$$\begin{aligned} \min_{\boldsymbol{\omega}, b, \boldsymbol{\xi}} f(\boldsymbol{\omega}, b) &= \frac{1}{2} \boldsymbol{\omega}^T \boldsymbol{\omega} + C \sum_{i=1}^m \xi_i \\ s.t. \quad y_i (\boldsymbol{\omega}^T \mathbf{x}_i + b) &\geq 1 - \xi_i, \quad (6) \\ \xi_i &\geq 0, \\ i &= 1, 2, \dots, m \end{aligned}$$

采用相同的方法求出问题 C 的对偶问题 D:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \varphi(\boldsymbol{\alpha}) &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ s.t. \quad \alpha_i &\geq 0, \alpha_i \leq C, \\ \sum_{i=1}^m \alpha_i y_i &= 0, \\ i &= 1, 2, \dots, m \end{aligned}$$

可以看出, 问题 D 与问题 B 的形式类似, 只是对 α_i 增加了上界约束. 此外, 由于问题 D 与问题 C 之间也具有强对偶关系, 其最优解满足问题 C 的 KKT 条件, 而问题 C 的 KKT 条件隐含了一个重要的事实:

若 $0 < \alpha_i < C$, 则该系数对应的样本位于分界间隔的边界上, 这时称该样本为支持向量; 若 $\alpha_i = C$, 则该系数对应的样本位于分界间隔内部或越过了分界间隔进入了另一个类别区域.

3.3 软间隔 SVM 的凸优化标准型

凸优化问题是相对容易求解的, 为了分析问题 D 是否为凸优化问题, 首先将其目标函数写为矩阵运算的形式

$$\varphi(\boldsymbol{\alpha}) = -\frac{1}{2} \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} + \mathbf{e}^T \boldsymbol{\alpha} \quad (7)$$

其中

$$\begin{aligned} \mathbf{e} &= (1, 1, \dots, 1)^T \\ \mathbf{Q} &= \begin{pmatrix} y_1 y_1 \mathbf{x}_1^T \mathbf{x}_1 & \cdots & y_1 y_m \mathbf{x}_1^T \mathbf{x}_m \\ \vdots & \ddots & \vdots \\ y_m y_1 \mathbf{x}_m^T \mathbf{x}_1 & \cdots & y_m y_m \mathbf{x}_m^T \mathbf{x}_m \end{pmatrix} = \mathbf{Y}^T \mathbf{X}^T \mathbf{X} \mathbf{Y} \quad (8) \end{aligned}$$

其中

$$\mathbf{Y} = \begin{pmatrix} y_1 & & \\ & \ddots & \\ & & y_m \end{pmatrix} \in \mathbb{R}^{m \times m} \quad (9)$$

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2 \cdots \mathbf{x}_m) \in \mathbb{R}^{n \times m} \quad (10)$$

为了判断 \mathbf{Q} 的正定性, 构造 \mathbf{Q} 的二次型

$$\mathbf{x}^T \mathbf{Q} \mathbf{x} = \mathbf{x}^T \mathbf{Y}^T \mathbf{X}^T \mathbf{X} \mathbf{Y} \mathbf{x} = (\mathbf{X} \mathbf{Y} \mathbf{x})^T \mathbf{X} \mathbf{Y} \mathbf{x} = \|\mathbf{X} \mathbf{Y} \mathbf{x}\|^2 \geq 0 \quad (11)$$

所以 \mathbf{Q} 是半正定的, 问题 D 是一个凸优化问题. 将其进一步转化为等价的凸优化标准型

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \varphi(\boldsymbol{\alpha}) = \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} \\ \text{s.t.} \quad & \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \boldsymbol{\alpha} \geq -C \begin{pmatrix} \mathbf{0} \\ \mathbf{e} \end{pmatrix} \\ & \mathbf{y}^T \boldsymbol{\alpha} = 0 \end{aligned} \quad (12)$$

这是一个经典的线性约束的凸优化问题, 通过求解得到的 $\boldsymbol{\alpha}$ 可以反解出问题 C 的解中的 $\boldsymbol{\omega}$. 最后得到分界面的方程:

$$\sum_{i=1}^m \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b = 0 \quad (13)$$

四 实验过程

4.1 实验结构

从(12)对 SVM 问题的表达中可以看出,该形式与约束优化方法 Zoutendijk 的标准型完全一致,因此本实验采用 Zoutendijk 方法用于 SVM 问题的求解.

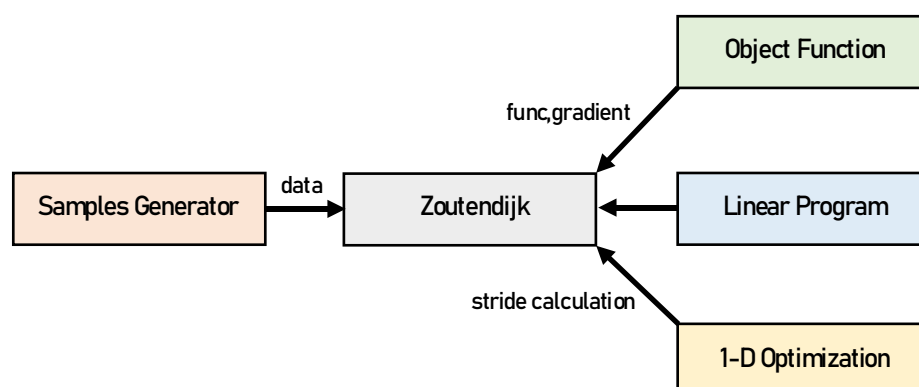


图 4-1 实验总体结构

本次实验通过 Python 实现,整个实验工程总体上包括样本发生器, Zoutendijk, 目标函数, 线性规划和一维搜索五个组成部分. 其中,样本发生器用于产生两个类别的样本数据并输入 Zoutendijk. 目标函数中定义了 SVM 目标函数的函数表达式及其梯度表达式. 由于 Zoutendijk 算法中需要分别通过线性规划和一维搜索来确定搜索方向和搜索步长,因此还需要线性规划函数和一维搜索函数的支持. 在上述组成部分中,除了线性规划函数使用的是 python 的优化模块 scipy 提供的 API 支持,其他部分都是自主编写的.

4.1.1 样本发生器对象

样本发生器对象实现了对正负类样本的随机产生. 由于 SVM 问题的求解维度是由样本数量决定的而不是由样本特征数决定的,所以在不影响求解算法性能展现的前提下,为了方便直观展示,使用样本发生器产生二维的样本数据,标记为正类和负类,每个类别的样本在给定的区域内呈随机分布.

4.1.2 Zoutendijk 对象

Zoutendijk 对象实现了整个 Zoutendijk 最优化算法流程,该算法的求解标准型如下

$$\begin{aligned} \min_x \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{Ax} \geq \mathbf{b}, \\ & \mathbf{Mx} = \mathbf{n} \end{aligned} \quad (14)$$

在本实验中

$$\mathbf{x} = \alpha, f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{Qx} - \mathbf{e}^T \mathbf{x} \quad (15)$$

$$\mathbf{A} = \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix}, \mathbf{b} = -\mathbf{C} \begin{pmatrix} \mathbf{0} \\ \mathbf{e} \end{pmatrix}, \mathbf{M} = \mathbf{y}^T, \mathbf{n} = \mathbf{0} \quad (16)$$

关于 Zoutendijk 的具体寻优过程本实验不再详述. 该算法的一个迭代过程包括两步: 通过 LP 问题计算搜索方向和通过一维搜索计算搜索步长, 于是在 Zoutendijk 对象中定义了两个方法 `opt_d()` 和 `opt_lambda()` 用于实现这两个过程, 在其运行方法 `search()` 中循环调用上述两个过程方法, 便实现了迭代搜索. 当检测到 LP 问题的最优值接近于 0 时, 迭代结束, 得到 Zoutendijk 的最优解.

4.1.3 目标函数对象

目标函数对象主要实现了目标函数及其梯度函数的表达.

目标函数的实现对任何一种最优化算法来说都是必需的, 另外, 在使用 Zoutendijk 迭代求解的过程中, 需要不断地获得目标函数在迭代点的梯度值来构造 LP 问题, 因此我们还必须给出目标函数的梯度函数的表达式. 在本实验中, 该梯度函数是一个向量值函数

$$\nabla f(\mathbf{x}) = \mathbf{Qx} - \mathbf{e} \quad (17)$$

4.1.4 一维搜索对象

一维搜索对象实现了 Zoutendijk 算法中的搜索步长的求解.

标准 Zoutendijk 算法的搜索步长求解采用的是精确一维搜索方法, 这种方法一般要求获得搜索函数的导函数表达, 然后求出导函数的零点, 这显然不利于编程实现; 另外, Zoutendijk 算法中的一维搜索问题是一个在受限范围内的寻优问题, 受限范围的下界是 0, 上界是依据当前迭代点的积极约束和松约束矩阵的具体数值而定的, 若采用求导的方法解决该问题, 还必须要考虑受限范围与导函数零点的位置关系, 增大了求解的复杂程度.... 庆幸的是, 黄金分割法恰好适用于解决这种条件下的一维搜索问题, 这是因为:

1. 黄金分割法具有很好的指数收敛性.
2. 黄金分割法天生就用于实现受限范围内的一维寻优.
3. 黄金分割法具有数据复用的特点, 运算量小, 且适于使用递归程序实现.

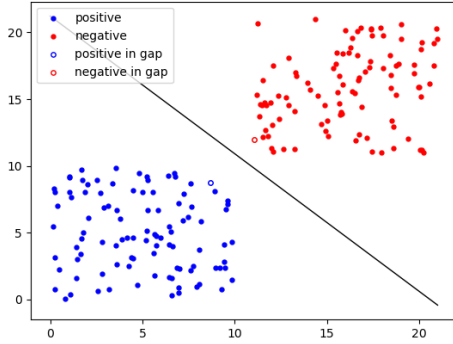
因此本实验中采用递归程序实现黄金分割法, 用于 Zoutendijk 算法中的步长搜索.

4.2 实验结果展示

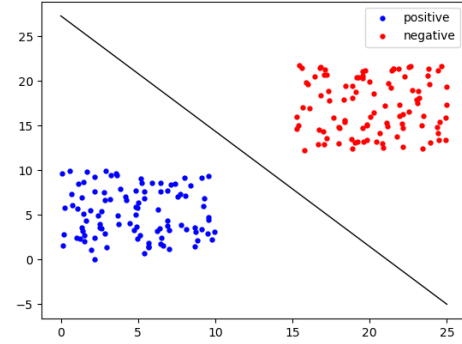
为了验证本实验设计的 SVM 训练器的实际工作效果, 在给定 C 的前提下, 采用上述样本发生器生成一定范围内的随机样本数据, 输入至 Zoutendijk 求解对象, 得出最优解 α^* , 并反算出其对应的分界面参数 $\omega^* = \sum_{i=1}^m \alpha_i^* y_i \mathbf{x}_i$, 需要注意的是, 在求取分界面参数 b 时, 理论上仅需将 $|\omega^* \mathbf{x}_i + b| = 1$ (18) 代入任一支持向量 (满足 $0 < \alpha_i < C$ 的样本), 然而在实际的程序运行结果中, 所有的 α_i 都是通过浮点数表示的, 基本不存在与 0 或 C 严格相等的关系.

所以这里采用了一种鲁棒性更好的方法: 设定一个阈值, 并获取在该阈值的允许下属于支持向量的 k 个样本, 将 (18) 代入这些样本, 分别求取 b_j^* , 将它们的平均值 $b^* = \frac{1}{k} \sum_{j=1}^k b_j^*$ 作为 b 的最终解. 在得到分界面参数后, 计算出分界面方程

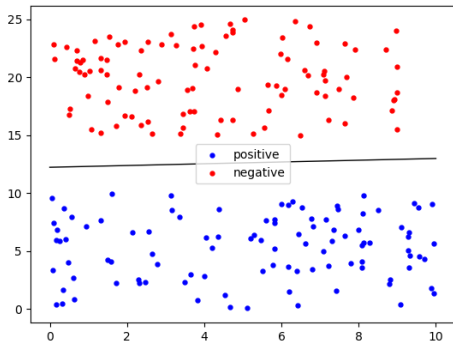
$$\omega^* \mathbf{x} + b^* = 0.$$



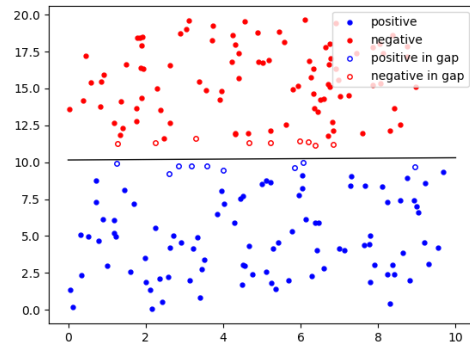
(a)



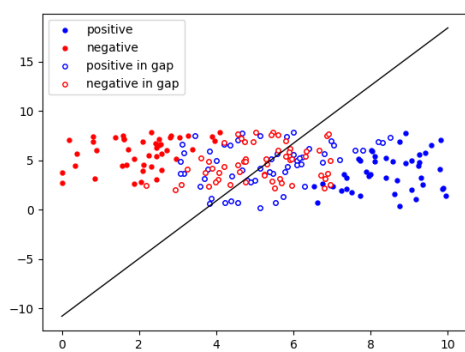
(b)



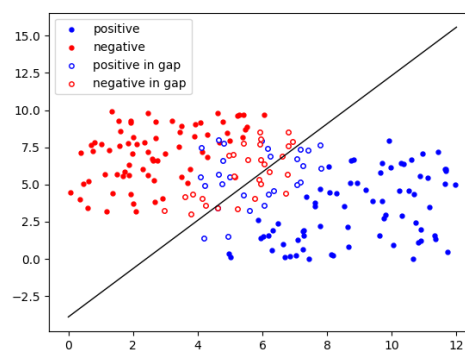
(c)



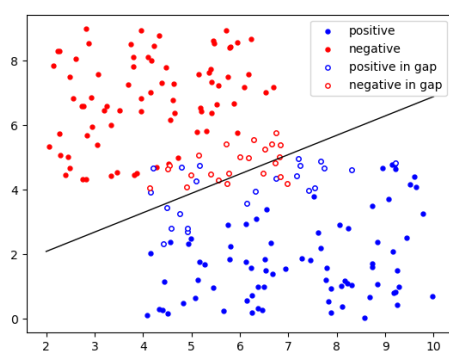
(d)



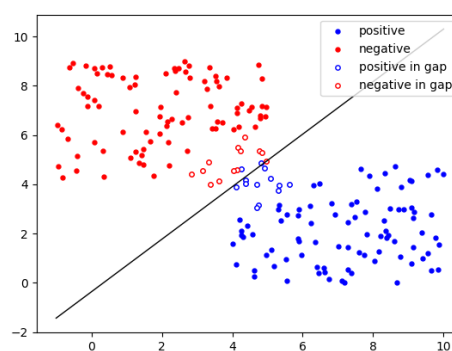
(e)



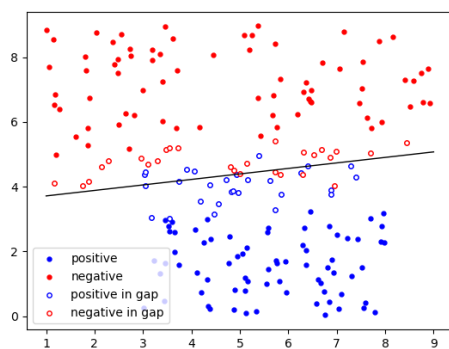
(f)



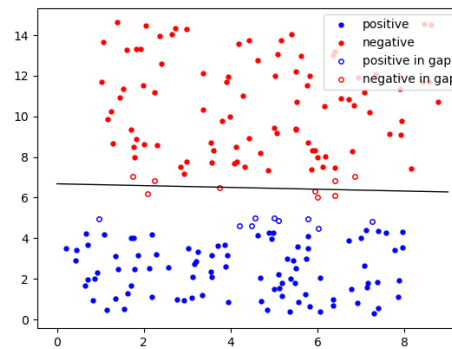
(g)



(h)



(i)



(j)

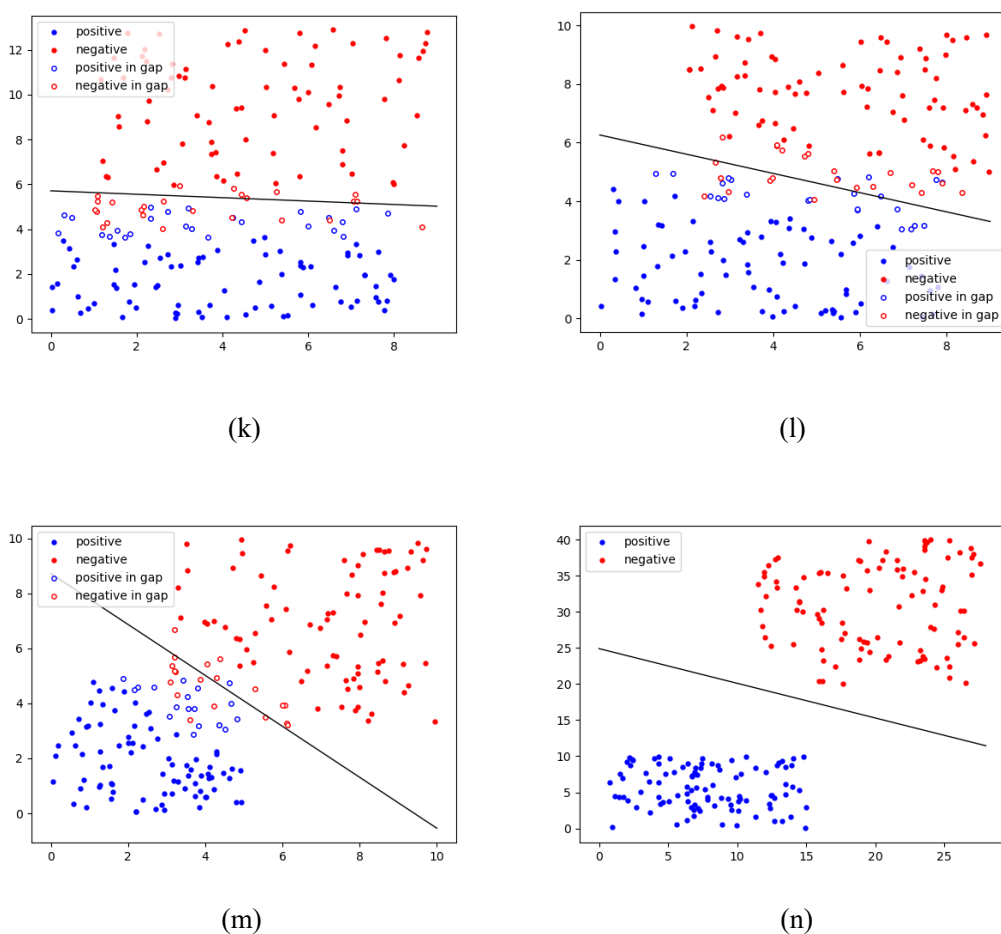


图 4-2 实验效果展示

图 4-2 展示了在给定多种样本分布情况下, 所设计 SVM 训练器的迭代结果. 所有样本都是 2 维的向量, 图 4-2 中分别展示了样本发生器产生的正样本(positive)和负样本(negative), 训练得到的分界面由每个图中的直线表示. 另外, 根据 SVM 的计算结果以及解的 KKT 条件, 可以通过 $\alpha_i = C$ 条件筛选出那些处在分类间隙内部以及越过分类间隙的样本, 在图 4-2 中表示为间隙正样本(positive in gap)和间隙负样本(negative in gap).

图 4-2 中的结果显示:

1. 当两个类别之间的间隔较大时(b)(c)(n), 训练器能够找到一个分界面使得正负样本被完全分开, 且该分界面的位置和斜率符合物理直觉. 由于模型非常易分, 并不会会有样本被迫进入分类间隙或越过分类间隙, 其表现就是解中的所有的 α_i 都小于 C .
2. 在保持线性完全可分的前提下, 当两个类别之间的间隔缩小时(a)(d)(j),

训练器仍然能够找到一个使正负样本被完全分开的分界面，只是这时有一些样本被迫进入了分类间隙或越过了分类间隙。

3. 当两个类别进一步靠近，以至于出现重叠时(g)(h)(i)(k)(l)(m)，训练器仍然能够在两个类别的交界处找到一个符合物理直觉的分界面，只是这时被迫进入分类间隙或超越分类间隙的样本数进一步增多。
4. 当两个类别存在较大重叠时(e)(f)，实验结果依然令人满意，只是这时被迫超越间隙的样本数量明显增多(如果在与分界面的距离为某个值的地方，存在他类样本被标记成了间隙样本，而相同距离处的本类样本并没有被标记为间隙样本，这说明该他类样本是一个超越了间隙的样本，比如图(e)中最左侧的几个正类样本就被迫成为了超越间隙样本)。

通过以上实验结果可以看出，无论两个类别样本之间存在多大的重叠，本实验设计的 SVM 训练器都能够在正负样本之间成功地找到一个符合物理直觉的分界面。随着类别重叠的程度加深，被迫进入间隙或超越间隙的样本也在增多，这符合 SVM 的原理，也符合直觉理解。

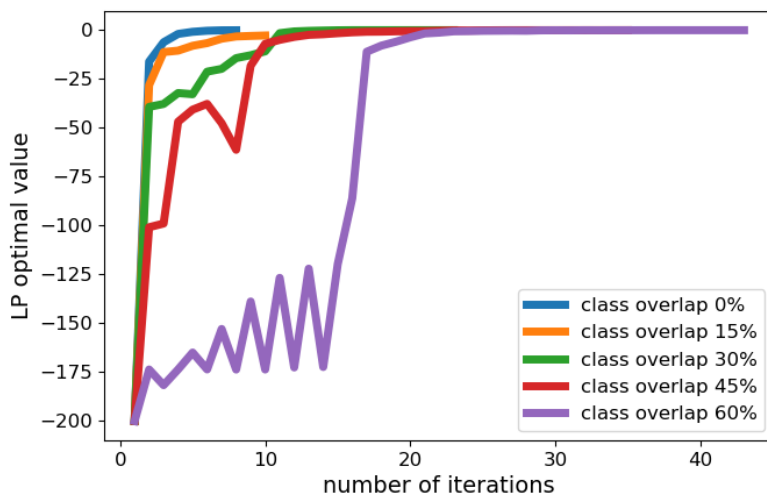


图 4-3 迭代次数随类别重叠程度的变化

在实验的过程中，我们观察到一个有趣的现象：随着类别重叠程度的加深，Zoutendijk 算法收敛的速度变慢，如图 4-3 所示。

在给定样本分布形状的前提下，当类别重叠率为 0 时，只需 9 次迭代便可得出最优解；当类别重叠率为 15%，需要 11 次迭代；当类别重叠率为 30%时，需要 25 次迭代；当类别重叠率为 45%时，需要 35 次迭代；当类别重叠率为 60%时，需要 43 次迭代。Zoutendijk 算法的收敛速度是一个无法预测的值，尽管对于上述现象，我们目前还不能给出一个理论的解释，但我们能够在直觉上感受到它的合理性：随着类别重叠程度增大，模型越来越不易分，分类工作自然要花费更多的

时间.

另外, C 作为一个超参数, 其值的选择也是至关重要的, 经过不断的实验探索, 总结出该超参数的选择依据主要有以下几个方面:

1. 当类别重叠加深, 模型不易分时, 意味着存在更多的样本需要被迫称为间隙样本, 所以应该降低 C , 使 α_i 更容易达到 C .
2. 当样本数量增加时, 意味着要得出 ω^* 需要累加更多的 α_i , 由于样本的随机范围没有变化, 所以 ω^* 的值不会产生大的变动, 所以每个 α_i 的值势必减小; 而样本数量的增多还会使间隙样本的数量增加, 所以需要有更多的 α_i 达到 C . 为了适应这些变化, 应该降低 C .
3. 根据观察, 当样本数据值总体增大时, 得出的 ω^* 值也会增大, 在样本数量不变的前提下, 每个 α_i 的值有增大的趋势, 这时应该增大 C 来适应 α_i 的变化. 为了避免样本数据值对 C 的选择带来的影响, 通常可以将样本数据进行归一化后再交给训练器训练.

五 总结

本次实验使用 Zoutendijk 方法作为主迭代算法, 配合黄金分割法实现的一维搜索函数以及线性规划函数实现了一个简易的软间隔 SVM 训练器.

通过使用随机样本对该训练器的测试, 我们发现该训练器能够展现出正确且符合物理直觉的分类效果, 且无论类别重叠程度的大小, 该训练器都能够取得令人满意的结果. 在分析间隙样本的数量与分布的过程中, 我们给出了一个同时符合原理与直觉的解释. 之后, 我们展示并分析了不同类别重叠率对 Zoutendijk 算法收敛速度的影响趋势, 并通过理论分析给出了关于超参数 C 的选择依据.

代码工程已上传 GitHub: <https://github.com/MiskaMoska/SVM-Trainer>