上海科技大学
ShanghaiTech University

本科毕业论文（设计）

题　目：多集合交问题的近似迭代重加权算法

学生姓名：高世杰

学　　号：2018533197

入学年份：2018

所在学院：信息科学与技术学院

攻读专业：计算机科学与技术

指导教师：王浩

上海科技大学

2022 年 6 月

上海科技大学
ShanghaiTech University

THESIS

Subject: Approximated iterative re-weighting algorithm for multiple sets intersection problem

Student Name: Shijie Gao

Student ID : 2018533197

Year of Entrance: 2018

School: School of Informtaion Science and Technology

Major: Computer Science and Technology

Advisor: Hao Wang

ShanghaiTech University

Date: 06/2022

上海科技大学
毕业论文(设计)学术诚信声明

本人郑重声明：所呈交的毕业论文（设计），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名：高世杰

日期：2022 年 5 月 26 日

上海科技大学
毕业论文（设计）版权使用授权书

本毕业论文（设计）作者同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海科技大学可以将本毕业论文（设计）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业论文（设计）。

保　密□，在___年解密后适用本授权书。

本论文属于

不保密☑。

（请在以上方框内打“√”）

作者签名：高世杰　　指导教师签名：王浩

日期：2022 年 5 月 26 日　　日期：2022 年 5 月 31 日

摘 要

多个闭凸集的交作为约束的凸复合问题经常出现在各种机器学习，大尺度优化问题中。在过去数十年间人们提出了许多一阶算法来求解约束集拥有高效计算投影的算法的问题，但是对于没有此类高效算法的约束集为多个集合交集的问题依然具有挑战。迭代重加权法 (IRWA) 通过对目标函数进行加权近似来求解单集合交集的投影问题，但是该算法的子问题需要精确求解，而这对于大尺度问题来说计算量比较大。本文提出了一种近似迭代重加权法 (AIRWA)，通过设计原对偶算法近似求解 IRWA 子问题，从而加速 IRWA 算法。本文证明了近似求子问题解的原对偶算法具有 $O(\frac{1}{N})$ 的收敛速率。数值实验部分对二次规划问题进行了测试，实验结果显示 AIRWA 算法比 IRWA 算法所需计算时间更短。

关键词：凸组合优化，惩罚函数法，近似重加权迭代法，原对偶算法，强对偶，线性收敛，F-范数，Fenchel-共轭

Abstract

The problem of minimizing a convex function on the intersection of multiple sets appears in many machine learning and large-scale optimization problems. The main difficulty of this problem is to find the projection of points outside the intersection on the intersection In the past decades, many first-order algorithms have been proposed to solve the problem that constraint sets have efficient projection counting algorithms, but it is still challenging to solve the problem that constraint sets without such efficient algorithms such as the constraint sets are the intersection of multiple sets. The iterative re-weighting method (IRWA) solves the intersection projection onto multipl sets by the weighted approximation of the objective function, but the subproblems of this problem need to be solved exactly, which requires large computational cost in large scale setting. In this paper, we propose an approximated iterative re-weighting algorithm (AIRWA), which inexactly solves the subproblems by a primal-dual algorithm. We prove the sub-problem algorithm has a convergence rate of $O(\frac{1}{N})$. In the numerical experiment, we show that the proposed algorithm outperforms the IRWA algorithm by test results on large scale quadratic programming problems.

Keywords: convex composite optimization, penalty method, approximated iterative re-weighting methods, strong duality, primal-dual algorithms, Frobenius-norm, Fenchel conjugate

目 录

第 1 章　引言

1.1　研究背景

凸复合优化问题出现在许多机器学习任务中，如矩阵补全问题 [27]：通过将矩阵所拥有的每个属性及其已知的项编码为约束集，矩阵补全可以简化为找到包含在一个有限集合族的交集中的一个点的问题；图转导问题通过提出的监督核学习与特定的半监督公式的等价性也将其转化为凸复合问题的形式 [28]；稀疏主成分分析可以转化为具有单位迹的正半定 (PSD) 矩阵集与 l_1 范数球的交集的投影问题 [29]；计算机科学中的相关聚类 [30]、图匹配 [31] 等问题也需要解决该问题。典型的凸复合问题的形式为

$$\begin{aligned} &\min_{x\in\mathcal{X}} f(x) \\ &\text{s.t.}\ \ x \in C, \end{aligned} \tag{1.1}$$

其中 f 是凸函数, $\mathcal{X} \in \mathcal{R}^n$ 是一个闭凸集，集合 $C = \bigcap_{i=1}^{N} C_i$，每个集合 $C_i \subseteq \mathcal{R}^n$ 代表一个闭凸集。

在过去数时年间提出了非常多的一阶算法用来求解集合 C 存在高效计算投影的算法时的问题 (1.1)，例如针对非光滑目标函数的镜像下降算法 [32]，镜像近段算子算法 [33]；Frank-Wolfe 算法和梯度投影法是最常见的两种算法，Frank-Wolfe 算法每步迭代需要进行两个操作：第一步寻找迭代方向，通过找寻可行集 C 中与负梯度内积最大的方向作为迭代方向，第二步确定新的迭代点，通过在迭代方向上进行线搜索确定新的迭代点；梯度投影法每步迭代也需要进行两个操作：第一步与传统梯度下降法相同，通过求该点的负梯度得到中间点，第二步将中间点向约束集进行投影，从而得到新的迭代点。在出现以上众多成熟的算法后，也出现了大量尝试改进以上算法的研究，如通过修正 Frank-Wolfe 算法中所寻找的迭代方向使其能够更好拟合负梯度方向的加速 Frank-Wolfe 算法 [34] 等。尽管当集合 C 存在高效计算投影的算法时如何求解问题 (1.1) 已经存在许多研究，但是在实际问题中投影到可行集 $C = \bigcap_{i=1}^{N} C_i$ 是难以计算的，注意到 Frank-Wolfe 算法在某些特殊形式的问题中仍然适用 [35]，但是并非一直有效。因此对约束集 C 不存在高效计算投影的算法时，如何求解问题 (1.1) 的研究是非常有意义的。

1.2 相关工作

解决这类问题的典型方法是将问题 (1.1) 中的约束惩罚至目标函数上:

$$\min_{x\in\mathcal{X}} J(x) = f(x) + \lambda \sum_{i=1}^{m} dist_2(x \mid C_i), \tag{1.2}$$

其中 $dist_2(x \mid C_i)$ 代表向量 x 到闭凸集 C_i 的距离。这么做的目的是将带约束优化问题 (1.1) 转变为无约束优化问题 (1.2)，然后就可对其使用牛顿法，并在求每步迭代步长时使用线搜索法或者信赖域法来保证目标函数 J 的下降。除此之外也有其他一些算法尝试解决问题 (1.1)：

1.2.1 分割投影次梯度法 (SPS)

SPS 算法引入了绝对范数以及定义了函数 h_P 的概念：称 P 为在 $\mathcal{R}^m$ 上的绝对范数如果其满足 $P(u) = P(|u|), \forall u \in \mathcal{R}^m$，其中 $|u|$ 表示向量 u 每个元素取模后得到的向量；称 P 为 $\mathcal{R}^m$ 上的绝对范数，并定义函数 $h_P : \mathcal{R}^n \to \mathcal{R}^+$ 满足 $h_P(x) = P(dist_2(x|C_1), \cdots, dist_2(x|C_m))$。SPS 算法基于这个定义将问题 (1.1) 改写为

$$\min_{x} J(x) = f(x) + \lambda h_P(x) \tag{1.3}$$

注意到对绝对范数 P 的选择有很多，当选择 P 为二范数时，则问题 (1.3) 等价于问题 (1.2)。SPS 第 t 步的迭代公式为

$$x_{t+1} := Proj_{\mathcal{X}}(x_t - \gamma_t \xi_t)$$

其中 $Proj_{\mathcal{X}}(x)$ 代表向量 x 在集合 $\mathcal{X}$ 上的投影，γ_t 为第 t 步的步长，ξ_t 为函数 J 在 x_t 处的次梯度。

SPS 的算法的优点在于可以具体问题自行选择绝对范数 P，最常见的选法有 1 范数，2 范数与无穷范数。当问题 (1.1) 的约束均为线性约束时，ADMM 算法也能有效解决该问题。

1.2.2 交替乘子法 (ADMM)

ADMM 算法适用的问题为

$$\begin{aligned} &\min_{x,z} f_1(x) + f_2(z) \\ &\text{s.t.} \quad Ax + Bz = b \end{aligned} \tag{1.4}$$

其中函数 f_1, f_2 均为凸函数，A，B 为矩阵。其第 t 步的迭代为

$$x_{t+1} = \arg\min_x \{f_1(x) + \frac{\rho}{2}||Ax + Bz_t - b + \frac{1}{\rho}\lambda_t||_2^2\}$$
$$z_{t+1} = \arg\min_x \{f_2(z) + \frac{\rho}{2}||Ax_{t+1} + Bz - b + \frac{1}{\rho}\lambda_t||_2^2\}$$
$$\lambda_{t+1} = \lambda_t + \rho(Ax_{t+1} + Bz_{t+1} - b)$$

其中 ρ 为步长。当问题 (1.1) 的约束均为线性约束时可以写作

$$\min_x f(x) \tag{1.5}$$
$$\text{s.t.} \quad Cx = b$$

因此令问题 (1.4) 中的函数 $f_1 = f_2 = \frac{1}{2}f$，向量 $x = z$, 矩阵 $A = B = \frac{1}{2}C$，那么就可以使用 ADMM 算法来求解。

1.3 主要贡献和内容安排

1.3.1 主要贡献

本文的主要工作是针对 IRWA 算法存在的问题进行改进。IRWA 算法需要精确求解子问题，因此在求解较大规模的问题时，求解子问题的计算量较大。本文对其子问题进行转化，将其写成原对偶问题的形式，通过设计原对偶算法对该子问题进行近似求解，并证明了求解子问题的原对偶算法具有 $O(\frac{1}{N})$ 收敛性。结合 IRWA 算法并将 IRWA 原本精确求解的子问题改为使用原对偶算法近似求解得到了最终的 AIRWA 算法。在数值实验部分对二次规划问题进行了测试，实验结果显示 AIRWA 算法比 IRWA 算法所需计算时间更短。

1.3.2 内容安排

本文的内容安排具体如下：

第一章引入了本文的所研究的问题，并对该问题的研究背景进行介绍，阐述了其拥有的研究价值，然后对相关工作进行概述。

第二章给出了本文的假设，文章所使用的符号与定义以及证明所使用到的相关定理。

第三章引入了 IRWA 算法，将原问题 (1.1) 进行重构，通过推导给出了 IRWA 子问题的形式。

第四章对子问题进行了重构，设计了原对偶算法求解子问题，并给出了该算法的收敛性证明；结合 IRWA 算法并将 IRWA 原本精确求解的子问题改为使用

原对偶算法近似求解得到了最终的 AIRWA 算法，并对 IRWA 算法与 AIRWA 算法两者进行了对比分析。

第五章选择二次规划问题进行数值实验，实验包含大规模二次规划问题，小规模二次规划问题以及算法参数探究，并证明了 AIRWA 算法在性能上表现由于 IRWA 算法。

第 2 章　基础知识

2.1　符号与定义

本文关于符号的大多数定义是基于 [1][2] 的定义。为方便起见，我们在符号部分回顾部分定义。集合 $\mathcal{R}^n$ 代表 n 维实欧几里得空间，关于 n 维向量的 2 范数记为 $||\cdot||_2$，其内积定义为 $\langle u,v\rangle = u^T v, \forall u,v \in \mathcal{R}^n$。$\mathcal{R}^n_+$ 表示 $\mathcal{R}^n$ 的非负卦。实 $m \times n$ 矩阵构成的集合将被表示为 $\mathcal{R}^{m\times n}$。关于在 $\mathcal{R}^{m\times n}$ 空间上矩阵的 Frobenius 范数记为 $||\cdot||_F$，其内积定义为 $\langle A,B\rangle = tr(A^T B)$，$\forall A,B \in \mathcal{R}^{m\times n}$。

定义 2.1.1. 给定集合 $\mathcal{X}$, 关于集合 $\mathcal{X}$ 的凸示性函数定义为

$$\mathbf{1}_{\mathcal{X}}(x) = \begin{cases} 0, & \text{if } x \in X, \\ +\infty, & \text{if } x \notin X. \end{cases} \tag{2.1}$$

定义 2.1.2. 函数 $f : \mathcal{X} \to \mathcal{R}$ 称为是 μ-strongly convex 当其满足

$$f(y) \geq f(x) + \nabla f(x)^T (x-y) + \frac{\mu}{2}||x-y||_2^2 \quad \forall x,y \in \mathcal{X}. \tag{2.2}$$

特别的当 $\mu = 0$ 时我们称函数 f 是凸的，满足

$$f(y) \geq f(x) + \nabla f(x)^T (x-y), \quad \forall x,y \in \mathcal{X}. \tag{2.3}$$

定义 2.1.3. 函数 $f : \mathcal{X} \to \mathcal{R}$ 称为是 L-smooth 当其满足

$$f(y) \leq f(x) + \nabla f(x)^T (x-y) + \frac{L_f}{2}||x-y||_2^2 \quad \forall x,y \in \mathcal{X}. \tag{2.4}$$

定义 2.1.4. 如果函数 f 是凸的，我们称向量 g 是关于函数 f 在 x 的次梯度，满足

$$f(z) \geq f(x) + g^T (z-x), \forall z \in \mathcal{R}^n. \tag{2.5}$$

并且关于函数 f 在 x 的所有次梯度构成的集合称为函数 f 在 x 的次微分, 记为

$$\partial f(x) = \{g | f(z) \geq f(x) + g^T (z-x), \forall z \in \mathcal{R}^n\}. \tag{2.6}$$

定义 2.1.5. 给定闭凸集 $X \subset \mathcal{R}^n$, X 在点 $\bar{x} \in X$ 处的法锥定义为

$$N(\bar{x} \mid X) = \{z \mid \langle z, x-\bar{x}\rangle \leq 0 \; \forall x \in X\}. \tag{2.7}$$

定义 2.1.6. 点 x 在集合 C 上的投影记为

$$\mathcal{P}_C(x) = \arg\min_{z\in C} ||x - z||_2. \tag{2.8}$$

定义 2.1.7. 点 x 到集合 C 的距离函数定义为

$$dist_2(x|C) = \min_{z\in C} ||x - z||_2 = ||x - \mathcal{P}_C(x)||_2. \tag{2.9}$$

很容易可以证明 $\frac{1}{2}dist_2^2(x|C) = \frac{1}{2}||x - \mathcal{P}_C(x)||_2^2$ 的梯度为 $x - \mathcal{P}_C(x)$。

定义 2.1.8. $h(x)$ 是一个函数, 则关于函数 h 的 $Fenchel$ 共轭定义为

$$h^*(y) = \sup_x\{\langle y, x\rangle - h(x)\} \tag{2.10}$$

定义 2.1.9. 令 $f : C \mapsto \mathcal{R}$ 是在 C 上的严格可微凸函数, 则函数 f 的布雷格曼分歧定义为

$$D_f(x, z) = f(x) - f(z) - \langle \nabla f(z), x - z\rangle \tag{2.11}$$

定义 2.1.10. 给定 $\varepsilon > 0$, 我们将 $x_\varepsilon \in \mathcal{X}$ 称作是关于最小化问题 $\min_{x\in\mathcal{X}} f(x)$ 的 ε-最优解，如果其满足

$$f(x_\varepsilon) - f(x^*) \le \varepsilon \tag{2.12}$$

其中 x^* 是 $\min_{x\in\mathcal{X}} f(x)$ 的最优解

定义 2.1.11. 一个最小化问题被称为凸问题如果其满足

$$\begin{aligned} &\min_{x\in\mathcal{X}} f(x) \\ &\text{s.t. } x \in C, \end{aligned} \tag{2.13}$$

其中 $f(x)$ 为凸函数，C 为凸集。

2.1.1 算子 K

2.1.1.1 算子定义

令 $K = WA$ 是一个算子, 其中 $W = \begin{pmatrix} w_1 & \cdots & w_1 \\ \vdots & \cdots & \vdots \\ w_m & \cdots & w_m \end{pmatrix}$，其中 $w_i \ge 0$; 向量 $x \in \mathcal{R}^n$，矩阵 $Y \in \mathcal{R}^{m\times n}$，算子 $A : x \to \begin{pmatrix} x_{(1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x_{(n)} \end{pmatrix}$。

2.1.1.2 算子范数

令 L_K 等于算子的范数，定义为

$$L_K := \|K\| = \sup_{\|x\|\le 1,\|Y\|\le 1} \langle Kx, Y\rangle \overset{*}{=} \sup_{||x||\le 1} ||Kx|| \tag{2.14}$$

其中 $(*)$ 等式是因为

$$\sup_{\|x\|\le 1,\|Y\|\le 1} \langle Kx, Y\rangle = \sup_{||x||\le 1,||Y||\le 1} ||Kx||_F||Y||_F \tag{2.15}$$

$$= \sup_{||x||\le 1} ||Kx||_F \tag{2.16}$$

等式 (2.15) 是因为三角不等式 $|\langle Kx, Y\rangle| \le ||Kx||_F||Y||_F$ 且 Y 是任意的，当 Y 取与 Kx 平行时等号可取得。为了证明上述范数 K 的定义是完备的，需要验证其满足范数的三个基本性质。

2.1.1.3 算子范数定义验证

$||K||$ 是一个定义完备的范数满足

1. 非负性：$||K|| \ge 0$，

$$||K|| = \sup_{||x||\le 1,||Y||\le 1} \langle Kx, Y\rangle \tag{2.17}$$

$$\ge \langle W \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix}, \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} \rangle \tag{2.18}$$

$$\ge 0\ , ||K|| = 0 \Leftrightarrow W = 0 \Leftrightarrow K = 0. \tag{2.19}$$

2. 齐次性：$||cK|| = |c|||K||, c \in \mathcal{R}$，

$$||cK|| = \sup_{\|x\|\le 1,\|Y\|\le 1} \langle cWAx, Y\rangle \tag{2.20}$$

$$= \sup_{\|x\|\le 1,\|Y\|\le 1} \langle WA(cx), Y\rangle \tag{2.21}$$

$$= \sup_{\|x\|\le 1,\|Y\|\le 1} \langle WA(|c|x), Y\rangle \tag{2.22}$$

$$= |c| \sup_{\|x\|\le 1,\|Y\|\le 1} \langle WAx, Y\rangle \tag{2.23}$$

$$= |c|||K|| \tag{2.24}$$

其中第一个等号是因为定义，第三个等号是因为向量 x 可以任取，因此对 c 施加绝对值整体的符号不改，第四个等号是因为内积 $\langle Kx, Y\rangle$ 的齐次性。

3. 三角不等式：$||K_1 + K_2|| \leq ||K_1|| + ||K_2||$，

$$||K_1 + K_2|| = \sup_{\|x\|\leq 1, \|Y\|\leq 1} \langle (K_1 + K_2)x, Y\rangle \tag{2.25}$$

$$= \sup_{\|x\|\leq 1, \|Y\|\leq 1} \langle (W_1 + W_2)Ax, Y\rangle \tag{2.26}$$

$$\leq \sup_{\|x\|\leq 1, \|Y\|\leq 1} \langle W_1 Ax, Y\rangle + \sup_{\|x\|\leq 1, \|Y\|\leq 1} \langle W_2 Ax, Y\rangle \tag{2.27}$$

$$= ||K_1|| + ||K_2||, \tag{2.28}$$

其中第一个，第二个等号是由于范数的定义; 第一个不等号是因为 *sup* 函数的性质。

2.1.1.4 算子性质

根据定义有下列不等式成立

$$\langle Kx, Y\rangle \leq ||Kx||_F ||Y||_F \overset{*}{\leq} L_K ||x||_2 ||Y||_F \tag{2.29}$$

其中 $(*)$ 不等式是因为

$$||Kx||_F = ||K\frac{x}{||x||_2}||_F ||x||_2 \tag{2.30}$$

$$\leq \sup_{||x||_2=1} ||Kx||_F ||x||_2 \tag{2.31}$$

$$\leq \sup_{||x||_2\leq 1} ||Kx||_F ||x||_2 \tag{2.32}$$

$$= L_K ||x||_2 \tag{2.33}$$

第一个不等号是因为 sup 函数的定义，第二个不等号是因为扩大可行集的范围会扩大 *sup* 函数的值，第二个等号是因为范数 $||K||$ 的定义。

2.2 定理

定理 2.2.1. *矩阵 $x, y \in \mathcal{R}^{m\times n}$ 满足*

$$||x + y||_F \leq ||x||_F + ||y||_F. \tag{2.34}$$

特别的当 $n = 1$ 时不等式退化为

$$||x + y||_2 \leq ||x||_2 + ||y||_2. \tag{2.35}$$

定理 2.2.2. *对凸集 $C \subseteq \mathcal{R}^n$，$x, y \in \mathcal{R}^n$ 有*

$$||\mathcal{P}_C(x) - \mathcal{P}_C(y)||_2 \leq ||x - y||_2. \tag{2.36}$$

定理 2.2.3. 如果函数 ϕ 是连续凸函数，$\hat{u} = \arg\min \phi(u) + D(u, \bar{u})$，则有

$$\forall u, \phi(u) + D(u, \bar{u}) \geq \phi(\hat{u}) + D(\hat{u}, \bar{u}) + D(u, \hat{u}) \tag{2.37}$$

这可以通过 $\hat{u}$ 的最优性条件推出 *[9]*。

定理 2.2.4. 若一个优化问题是凸问题，那么它满足

$$\max_{\lambda} \min_{x \in X} \mathcal{L}(x, \lambda) = \min_{x \in X} \max_{\lambda} \mathcal{L}(x, \lambda) \tag{2.38}$$

其中 $\mathcal{L}(x, \lambda) = f(x) + \langle \lambda, dist_2(x|C) \rangle$。

2.3 问题假设

本文假设均是沿用 [1][2] 所给出的假设，在相关问题研究中是具有普适性的。

假设 2.3.1. $\mathcal{X}, C_1, ..., C_N$ 是 $\mathcal{R}^n$ 上的闭凸集且满足 $\mathcal{X} \supset C = \bigcap_{i=1}^{N} C_i$

假设 2.3.2. 集合 $\mathcal{X}, C_1, ..., C_N$ 都存在高效计算投影的算法。

假设 2.3.3. $f : \mathcal{X} \to \mathcal{R}$ 是凸函数且是 *L-smooth* 的，关于 *L-smooth* 的常数记为 $L_f > 0$。

第 3 章　问题重构

我们将问题 (1.1) 的约束惩罚到目标函数上，则问题可以被转化为

$$\min_{x\in\mathcal{X}} J(x) = f(x) + \lambda \sum_{i=1}^{m} dist_2(x \mid C_i), \tag{3.1}$$

其中 $dist_2(x|C_i)$ 代表变量 x 到凸集 C_i 的距离。当惩罚参数 λ 足够大时，变量 x 会被严格限制在凸集 C_i 中，即 $dist(x|C_i) = 0$，由此满足问题 (1.1) 和问题 (3.1) 的等价性。

直接求解问题 (3.1) 并不是一个好的选择，因为当 x 趋近于凸集 C_i 时，$dist(x|C_i)$ 会变得非常小，同时导致计算量的增加以及产生数值误差问题。这个问题可以通过对函数 $J(x)$ 做光滑近似来解决。给定 $\epsilon \in \mathbb{R}^n_+$, 函数 $J(x)$ 的 ϵ-smoothing 定义为

$$\hat{J}(x,\epsilon) = f(x) + \lambda \sum_{i=1}^{m} \sqrt{dist_2^2(x \mid C_i) + \epsilon_i^2}, \tag{3.2}$$

因为存在变量 ϵ，因此不会产生极端数值。

基于此定义考虑子问题，在给定 x_t 点处通过给出松弛向量 $\epsilon^t \in \mathbb{R}^n_{++}$，将变量 x 到凸集 C_i 的距离作为权重惩罚在 $dist(x|C_i)$ 前，我们得到了子问题处关于问题 (3.2) 的加权近似

$$G(x \mid x_t, \epsilon_t) = f(x) + \frac{\lambda}{2} \sum_{i=1}^{m} w_i^t dist_2^2(x \mid C_i), \tag{3.3}$$

其中 $w_i^t := \frac{1}{\sqrt{dist_2^2(x_t|C_i)+(\epsilon_i^t)^2}}$。更进一步，对变量 x 的投影做局部近似，定义 $\hat{G}(x \mid x_t, \epsilon_t)$ 为

$$\hat{G}(x \mid x_t, \epsilon_t) = f(x) + \frac{\lambda}{2} \sum_{i=1}^{m} w_i^t (||x - \mathcal{P}_{C_i}(x_t)||_2^2). \tag{3.4}$$

由此我们得到了迭代重加权算法 (IRWA) 所求解的子问题形式。

迭代重加权算法 (IRWA) 的目标是求解问题 (3.2) 从而获得问题 (3.1) 的解。在每一轮迭代中通过精确求解子问题 (3.4) 来获得下一步的迭代点。然而精确求解 (3.4) 在大规模问题中会产生较大计算量，从而导致算法收敛速率下降。因此本文设计原对偶算法近似求解该子问题，实现单次迭代上计算量的减少，从而提升整体算法的运行速率。本文将在下一章具体分析子问题 (3.4) 以及求解它的算法。

Algorithm 1 迭代重加权算法 (IRWA)

输入: 选择初始点 x_0, 权重 $\eta \in (0,1), \lambda > 0, \gamma > 0$, $M > 0$

输出: x_t

for $t = 0, 1, 2, \ldots$ **do**

    求投影 $\mathcal{P}_{C_i}(x_t)$

    求解子问题

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \hat{G}(x \mid x_t, \epsilon_t) \tag{3.5}$$

    if $||x_{t+1} - x_t|| < M(\epsilon_i^t)^{1+\gamma}$ **then**

        更新 α_{t+1} 和 ϵ_{t+1}

    end if

    if $||x_{t+1} - x_t|| < \sigma_1$ and $||\epsilon^t||_2 < \sigma_2$ **then**

        break;

    end if

    t = t+1

end for

第 4 章　近似迭代重加权算法

4.1　子问题重构

近似迭代重加权算法 (AIRWA) 旨在通过近似求解以下子问题来提升迭代重加权算法 (IRWA) 的速度

$$\min_{x\in\mathcal{X}} \hat{G}(x \mid x_t, \epsilon_t) = f(x) + \frac{\lambda}{2}\sum_{i=1}^{m} w_i^t(||x - \mathcal{P}_{C_i}(x_t)||_2^2), \tag{4.1}$$

其中 $w_i^t := \frac{1}{\sqrt{dist_2^2(x_t|C_i)+(\epsilon_i^t)^2}}$，$x, x_t, P_{C_i}(x_t) \in \mathcal{R}^n$。

为了设计原对偶算法来近似求解问题 (4.1)，首先需要将其重构成原对偶算法所对应的问题形式，即最小最大化问题。考虑到原对偶算法需要两个变量交替迭代，因此除了已有的变量 x 外需要再创造出另外一个变量 Y，考虑到目标函数中存在求和函数，因此增加的变量的维度应该是求和函数的个数乘以向量 x 的维度，即 $m \times n$ 的矩阵，所以第一步需要将问题 (4.1) 的求和函数写成矩阵的形式。

首先将权重 w_i^t 乘入范数中得到

$$\min_{x\in\mathcal{X}} f(x) + \sum_{i=1}^{m}\frac{\lambda}{2}||w_i^t x - w_i^t\mathcal{P}_{C_i}(x_t)||_2^2, \tag{4.2}$$

将 m 个二范数按列合并进行转置得到关于 $m \times n$ 的矩阵的 F 范数

$$\min_{x\in\mathcal{X}} f(x) + \frac{\lambda}{2}||\begin{pmatrix} w_1^t x_{(1)} & \cdots & w_1^t x_{(n)} \\ \vdots & \cdots & \vdots \\ w_m^t x_{(1)} & \cdots & w_m^t x_{(n)} \end{pmatrix} - \begin{pmatrix} w_1^t\mathcal{P}_{C_1}(x_t)_1 & \cdots & w_1^t\mathcal{P}_{C_1}(x_t)_n \\ \vdots & \cdots & \vdots \\ w_m^t\mathcal{P}_{C_1}(x_t)_1 & \cdots & w_m^t\mathcal{P}_{C_1}(x_t)_n \end{pmatrix}||_F^2, \tag{4.3}$$

令 $b_i = \begin{pmatrix} b_{i1} \\ \vdots \\ b_{in} \end{pmatrix} = \begin{pmatrix} w_i^t\mathcal{P}_{C_i}(x_t)_1 \\ \vdots \\ w_i^t\mathcal{P}_{C_i}(x_t)_n \end{pmatrix}$, 我们有

$$\min_{x\in\mathcal{X}} f(x) + \frac{\lambda}{2}||\begin{pmatrix} w_1^t x_{(1)} & \cdots & w_1^t x_{(n)} \\ \vdots & \cdots & \vdots \\ w_m^t x_{(1)} & \cdots & w_m^t x_{(n)} \end{pmatrix} - \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \cdots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix}||_F^2, \tag{4.4}$$

观察到变量 x_i 在第 i 列重复出现，因此我们可以将权重 w_i 与变量 x_i 剥离开，变

成两个矩阵相乘，经过简单调整得到

$$\min_{x\in\mathcal{X}} f(x)+\frac{\lambda}{2}||\begin{pmatrix} w_1^t & \cdots & w_1^t \\ \vdots & \cdots & \vdots \\ w_m^t & \cdots & w_m^t \end{pmatrix}\begin{pmatrix} x_{(1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x_{(n)} \end{pmatrix}-\begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \cdots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix}||_F^2, \tag{4.5}$$

令 $W^t=\begin{pmatrix} w_1^t & \cdots & w_1^t \\ \vdots & \cdots & \vdots \\ w_m^t & \cdots & w_m^t \end{pmatrix}\in\mathcal{R}^{m\times n}$, $B=\begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \cdots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix}\in\mathcal{R}^{m\times n}$, 并定义算子 A : $x\to\begin{pmatrix} x_{(1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x_{(n)} \end{pmatrix}$, 问题 (4.5) 可以写成

$$\min_{x\in\mathcal{X}} f(x)+\frac{\lambda}{2}||W^tAx-B||_F^2, \tag{4.6}$$

在成功将问题写成矩阵形式后，我们令 $W^tX=Y\in\mathcal{R}^{m\times n}$ 并把它写作约束得到

$$\begin{aligned} &\min_{x\in\mathcal{X},Y} f(x)+\frac{\lambda}{2}||Y-B||_F^2 \\ &\text{s.t. } W^tAx=Y, \end{aligned} \tag{4.7}$$

这样做的好处是可以通过写问题 (4.6) 的拉格朗日方程将问题转化成原对偶问题的形式, 从而满足设计原对偶算法的要求。

首先将问题 (4.6) 的矩阵等式约束看做由 $m\times n$ 个单个变量构成的等式约束，因此问题 (4.6) 的拉格朗日方程可以写作

$$\max_{\alpha}\min_{x\in\mathcal{X},Y} f(x)+\frac{\lambda}{2}||Y-B||_F^2+\sum_{i=1}^{m}\sum_{j=1}^{n}\alpha_{ij}(w_ix_j-y_{ij}), \tag{4.8}$$

将最后一项合并成矩阵形式可以写作

$$\max_{\alpha}\min_{x\in\mathcal{X},Y} f(x)+\frac{\lambda}{2}||Y-B||_F^2+\langle\alpha,W^tAx-Y\rangle, \tag{4.9}$$

其中 $\alpha\in\mathcal{R}^{m\times n}$, $\langle\alpha,W^tAx-Y\rangle=\text{Trace}(\alpha^T(W^tAx-Y))$

将关于变量 x 的项和变量 Y 的项分别合并，可以得到

$$\max_{\alpha}\min_{x\in\mathcal{X}}\{f(x)+\langle W^tAx,\alpha\rangle\}+\min_{Y}\{\frac{\lambda}{2}||Y-B||_F^2-\langle\alpha,Y\rangle\}, \tag{4.10}$$

根据定理 2.2.4 可以得到

$$\min_{x\in\mathcal{X}}\max_{\alpha}\{f(x)+\langle W^tX,\alpha\rangle\}-\max_{Y}\{\langle\alpha,Y\rangle-\frac{\lambda}{2}||Y-B||_F^2\}, \tag{4.11}$$

令 $h(Y)=\frac{\lambda}{2}||Y-B||_F^2$, h^* 是函数 h 的 Fenchel-共轭，$g(x)=\mathbf{1}_{\mathcal{X}}(x)$, 根据凸问题的强对偶性，并把 α 记作 Y，我们得到子问题的最终形式为

$$\min_x \max_Y \mathcal{L}(x,Y)=\langle W^tAx,Y\rangle+f(x)+g(x)-h^*(Y),\ g(x)=\mathbf{1}_{\mathcal{X}}(x) \tag{4.12}$$

4.2 原对偶算法

Algorithm 2 原对偶算法

输入: 选择初始点 x_0, 权重 $W,\tau,\sigma,K>0$

输出 t: x_k

1: **for** $k=0,1,2,\ldots$ **do**

2: 选择中间点 $\tilde{Y}$

3: 求解子问题

$$x_{k+1}=\arg\min_x f(x_k)+\langle\nabla f(x_k),x-x_k\rangle+g(x)+\langle WAx,\tilde{Y}\rangle+\frac{1}{\tau}D_x(x,x_k) \tag{4.13}$$

4: 选择中间点 $\tilde{x}$

5: 求解子问题

$$Y_{k+1}=\arg\min_Y h^*(Y)-\langle WA\tilde{x},Y\rangle+\frac{1}{\sigma}D_Y(Y,Y_k) \tag{4.14}$$

6: **if** k = K **then**

7: break;

8: **end if**

9: k = k+1

10: **end for**

根据 Chambolle A, Pock T.[2]，我们设计得到了求解子问题 (4.11) 的原对偶算法，其中 $D_x(x,x_k)=\frac{1}{2}\|x-x_k\|_2^2 D_Y(Y,Y_k)=\frac{1}{2}\|Y-Y_k\|_F^2$ 是关于变量 x 和变量 Y 的布雷格曼分歧。每次迭代需要起点 (x_k,Y_k) 和中间点 $(\tilde{x},\tilde{Y})$ 作为输入，并输出新的迭代点 (x_{k+1},Y_{k+1})。由于在求解子问题时 (4.11) 权重矩阵 W^t 是不变的，为方便起见在下述子问题收敛性分析中，记 $W=W^t$，记算子 $K=WA$，可以验证满足第二章中关于算子 K 的定义。

4.3 收敛性分析

引理 4.3.1. 如果满足上述原对偶算法，则对任意 $x \in \mathcal{R}^n, Y \in \mathcal{R}^{m\times n}, K$ 是上述定义的算子, 则有

$$\begin{aligned}\mathcal{L}(x_{k+1},Y)-\mathcal{L}(x,Y_{k+1}) \le & \frac{1}{\tau}D_x(x,x_k)-\frac{1}{\tau}D_x(x,x_{k+1})-\frac{1}{\tau}D_x(x_{k+1},x_k)\\ &+\frac{L_f}{2}\|x_{k+1}-x_k\|^2+\frac{1}{\sigma}D_Y(Y,Y_k)-\frac{1}{\sigma}D_Y(Y,Y_{k+1})\\ &-\frac{1}{\sigma}D_Y(Y_{k+1},Y_k)+\langle K(x-x_{k+1}),\tilde{Y}-Y_{k+1}\rangle\\ &-\langle K(\tilde{x}-x_{k+1}),Y-Y_{k+1})\rangle\end{aligned} \tag{4.15}$$

证明. 首先研究函数 f, 根据定理 4, 令函数 $\phi(x)=\tau(\langle\nabla f(x_k),x\rangle+g(x)+\langle WAx,\tilde{Y}\rangle)$ 则有

$$\begin{aligned}\langle\nabla f(x_k),x\rangle+g(x)+\langle WAx,\tilde{Y}\rangle+\frac{1}{\tau}D_x(x,x_k)\ge & \\ \langle\nabla f(x_k),x_{k+1}\rangle+g(x_{k+1})+\langle Kx_{k+1},\tilde{Y}\rangle+\frac{1}{\tau}D_x(x_{k+1},x_k)&\\ +\frac{1}{\tau}D_x(x,x_{k+1})&\end{aligned} \tag{4.16}$$

根据函数 f 的凸性和 *L-smooth*，有不等式

$$f(x)\ge f(x_k)+\langle\nabla f(x_k),x-x_k\rangle\ge f(x_{k+1})+\langle\nabla f(x_k),x-x_k\rangle-\frac{L_f}{2}\|x_{k+1}-x_k\|^2 \tag{4.17}$$

将 *(4.35)* 和 *(4.36)* 两式合并，可以得到

$$\begin{aligned}f(x)+g(x)+\frac{1}{\tau}D_x(x,x_k)+\frac{L_f}{2}\|x_{k+1}-x_k\|^2\ge & \\ f(x_{k+1})+g(x_{k+1})+\langle K(x_{k+1}-x),\tilde{Y}\rangle+\frac{1}{\tau}D_x(x_{k+1},x_k)&\\ +\frac{1}{\tau}D_x(x,x_{k+1})&\end{aligned} \tag{4.18}$$

然后研究函数 h，根据定理 4，令函数 $\phi(Y)=\sigma(h^*(Y)-\langle WA\tilde{x},Y\rangle)$ 则有

$$\begin{aligned}h^*(Y)-\langle WA\tilde{x},Y\rangle+\frac{1}{\sigma}D_Y(Y,Y_k)\ge & \\ h^*(Y_{k+1})-\langle WA\tilde{x},Y_{k+1}\rangle+\frac{1}{\sigma}D_Y(Y_{k+1},Y_t)+\frac{1}{\sigma}D_Y(Y,Y_{k+1})&\end{aligned} \tag{4.19}$$

将 *(4.37)* 和 *(4.38)* 两式合并，可以得到

$$\begin{aligned} f(x_{k+1})+g(x_{k+1})+\langle K(x_{k+1}-x),\tilde{Y}\rangle+\frac{1}{\tau}D_x(x_{k+1},x_k)+\frac{1}{\tau}D_x(x,x_{k+1}) \\ +h^*(Y_{k+1})-\langle WA\tilde{x},Y_{k+1}\rangle+\frac{1}{\sigma}D_Y(Y_{k+1},Y_t) \\ +\frac{1}{\sigma}D_Y(Y,Y_{k+1}) \\ \leq f(x)+g(x)+\frac{1}{\tau}D_x(x,x_k)+\frac{L_f}{2}\|x_{k+1}-x_k\|^2 \\ +h^*(Y)-\langle WA\tilde{x},Y\rangle+\frac{1}{\sigma}D_Y(Y,Y_k) \end{aligned} \tag{4.20}$$

根据问题 *(4.11)* 对函数进行合并可以得到

$$\begin{aligned} &[f(x_{k+1})+g(x_{k+1})+\langle Kx_{k+1},Y\rangle-h^*(Y)]-[f(x)+g(x)+\langle Kx,Y_{k+1}\rangle-h^*(Y_{k+1})] \\ &\leq\frac{1}{\tau}D_x(x,x_k)-\frac{1}{\tau}D_x(x,x_{k+1})-\frac{1}{\tau}D_x(x_{k+1},x_k) \\ &+\frac{L_f}{2}\|x_{k+1}-x_k\|^2+\frac{1}{\sigma}D_Y(Y_{k+1},Y_k)-\frac{1}{\sigma}D_Y(Y,Y_{k+1}) \\ &-\frac{1}{\sigma}D_Y(Y_{k+1},Y_k)-\langle K(x_{k+1}-x),\tilde{Y}\rangle+\langle K\tilde{x},Y_{k+1}-Y\rangle \\ &+\langle Kx_{k+1},Y\rangle-\langle Kx,Y_{k+1}\rangle \end{aligned} \tag{4.21}$$

对 *(4.40)* 最后四项进行整理

$$\langle K\tilde{x},Y_{k+1}-Y\rangle-\langle K(x_{k+1}-x),\tilde{Y}\rangle+\langle Kx_{k+1},Y\rangle-\langle Kx,Y_{k+1}\rangle \tag{4.22}$$

$$=\langle K\tilde{x},Y_{k+1}\rangle-\langle K\tilde{x},Y\rangle-\langle Kx_{k+1},\tilde{Y}\rangle+\langle Kx,Y_{k+1}\rangle+\langle Kx_{k+1},Y\rangle \tag{4.23}$$

$$-\langle Kx,Y_{k+1}\rangle$$

$$=\langle Kx,\tilde{Y}-Y_{k+1}\rangle+\langle K\tilde{x},Y_{k+1}-Y\rangle-\langle Kx_{k+1},\tilde{Y}-Y\rangle \tag{4.24}$$

$$=\langle K(x-x_{k+1}),\tilde{Y}-Y_{k+1}\rangle+\langle Kx_{k+1},\tilde{Y}-Y_{k+1}\rangle-\langle Kx_{k+1},\tilde{Y}-Y\rangle \tag{4.25}$$

$$+\langle K\tilde{x},Y_{k+1}-Y\rangle$$

$$=\langle K(x-x_{k+1}),\tilde{Y}-Y_{k+1}\rangle-\langle K(\tilde{x}-x_{k+1}),Y-Y_{k+1}\rangle \tag{4.26}$$

将 *(4.45)* 代入 *(4.40)* 可以获得

$$\begin{aligned} \mathcal{L}(x_{k+1},Y)-\mathcal{L}(x,Y_{k+1})\leq\frac{1}{\tau}D_x(x,x_k)-\frac{1}{\tau}D_x(x,x_{k+1})-\frac{1}{\tau}D_x(x_{k+1},x_k) \\ +\frac{L_f}{2}\|x_{k+1}-x_k\|^2+\frac{1}{\sigma}D_Y(Y,Y_k)-\frac{1}{\sigma}D_Y(Y,Y_{k+1}) \\ -\frac{1}{\sigma}D_Y(Y_{k+1},Y_k)+\langle K(x-x_{k+1}),\tilde{Y}-Y_{k+1}\rangle \\ -\langle K(\tilde{x}-x_{k+1}),Y-Y_{k+1}\rangle\rangle \quad \text{证毕.} \end{aligned} \tag{4.27}$$

为了分析收敛速率，对原对偶算法取特定中间点 $(\tilde{x},\tilde{Y})=(2x_{k+1}-x_k,Y_k)$。

Algorithm 3 $O(\frac{1}{N})$ 非线性原对偶算法 (NLPD)

输入: 选择初始点 x_0, 权重 $W,\tau,\sigma,K>0$

输出: x_k

1: **for** $k=0,1,2,\ldots$ **do**

2: 求解子问题

$$x_{k+1}=\arg\min_x f(x_k)+\langle\nabla f(x_k),x-x_k\rangle+g(x)+\langle WAx,Y^k\rangle+\frac{1}{\tau}D_x(x,x_k) \tag{4.28}$$

$$Y_{k+1}=\arg\min_Y h^*(Y)-\langle WA(2x_{k+1}-x_k),Y\rangle+\frac{1}{\sigma}D_Y(Y,Y_k) \tag{4.29}$$

3: **if** k = K **then**

4: break;

5: **end if**

6: k = k+1

7: **end for**

引理 4.3.2. 如果 $(\frac{1}{\tau}-L_f)\frac{1}{\sigma}\geq L_K^2$, 则 $\forall x,x'\in\mathcal{R}^n,Y,Y'\in R^{m\times n}$ 有

$$(\frac{1}{\tau}-L_f)D_x(x,x')+\frac{1}{\sigma}D_y(Y,Y')-\langle K(x-x'),Y-Y'\rangle\geq 0. \tag{4.30}$$

证明:

$$\left(\frac{1}{\tau}-L_f\right)D_x(x,x')+\frac{1}{\sigma}D_Y(Y,Y')-\langle K(x-x'),Y-Y'\rangle \tag{4.31}$$

$$\geq\sigma L_K^2D_x(x,x')+\frac{1}{\sigma}D_y(Y,Y')-\langle K(x-x'),Y-Y'\rangle \tag{4.32}$$

$$=\frac{\sigma L^2}{2}||x-x'||_2^2+\frac{1}{2\sigma}||Y-Y'||_F^2-\langle K(x-x'),Y-Y'\rangle \tag{4.33}$$

$$\geq 2\sqrt{\frac{L^2}{4}||x-x'||_2^2\frac{1}{2\sigma}||Y-Y'||_F^2}-\langle K(x-x'),Y-Y'\rangle \tag{4.34}$$

$$=L_K||x-x'||_2||Y-Y'||_F-\langle K(x-x'),Y-Y'\rangle \tag{4.35}$$

$$\geq 0. \tag{4.36}$$

第一个不等式是因为引理 *4.4.2* 的条件 *(4.50)*，第二个不等式是通过基本不等式获得，最后一个不等式是因为算子性质中的 $\langle Kx,Y\rangle\leq L_K||x||_2||Y||_F$。

定理 4.3.3. 令 $(x^k, y^k), k = 0, \ldots, N-1$ 是非线性原对偶算法生成的序列。令选择的参数 $\tau, \sigma > 0$ 满足 $x, x' \in \mathcal{R}^n$ 且 $Y, Y' \in \mathcal{R}^{m\times n}$

$$(\frac{1}{\tau} - L_f)D_x(x, x') + \frac{1}{\sigma}D_y(Y, Y') - \langle K(x - x'), Y - Y'\rangle \geq 0 \tag{4.37}$$

则对任意 $(x, Y) \in \mathcal{X} \times \mathcal{R}^{n\times m}$ 满足

$\mathcal{L}(\bar{X}, Y) - \mathcal{L}(x, \bar{Y}) \leq \frac{1}{N}(\frac{1}{\tau}D_x(x, x_0) + \frac{1}{\sigma}D_Y(Y, Y_0) - \langle K(x - x_0), Y - Y_0\rangle)$

其中 $\bar{X} = \frac{1}{N}\sum_{k=1}^{N} x_k$, $\bar{Y} = \frac{1}{N}\sum_{k=1}^{N} Y_k$.

证明. 根据引理 *4.4.1* 和 *NLPD* 算法，可以得到

$$\begin{aligned}\mathcal{L}(x_{k+1}, Y) - \mathcal{L}(x, Y_{k+1}) \leq{}& \frac{1}{\tau}D_x(x, x_k) - \frac{1}{\tau}D_x(x, x_{k+1}) - \frac{1}{\tau}D_x(x_{k+1}, x_k) \\ &+ \frac{L_f}{2}\|x_{k+1} - x_k\|^2 + \frac{1}{\sigma}D_Y(Y, Y_k) - \frac{1}{\sigma}D_Y(Y, Y_{k+1}) \\ &- \frac{1}{\sigma}D_Y(Y_{k+1}, Y_k) + \langle K(x - x_{k+1}), \tilde{Y} - Y_{k+1}\rangle \\ &- \langle K(\tilde{x} - x_{k+1}), Y - Y_{k+1}\rangle\end{aligned} \tag{4.38}$$

整理式子 *(4.57)* 并进行分组，则有

$$\begin{aligned}\mathcal{L}(x_{k+1}, Y) - \mathcal{L}(x, Y_{k+1}) \leq{}& [\frac{1}{\tau}D_x(x, x_k) + \frac{1}{\sigma}D_Y(Y, Y_k) - \langle K(x - x_k), Y - Y_k\rangle] \\ &- [\frac{1}{\tau}D_x(x, x_{k+1}) + \frac{1}{\sigma}D_Y(Y, Y_{k+1}) - \langle K(x - x_{k+1}), Y - Y_{k+1}\rangle] \\ &- [\frac{1}{\tau}D_x(x_{k+1}, x_k) + \frac{1}{\sigma}D_Y(Y_{k+1}, Y_k) \\ &- \langle K(x_{k+1} - x_k), Y_{k+1} - Y_k\rangle - \frac{L_f}{2}\|x_{k+1} - x_k\|^2]\end{aligned} \tag{4.39}$$

根据条件 *(4.56)* 和 D_x 的 $1 - convex$ 性, 所有括号中的东西是非负的。

现在对两侧同时进行求和，令 $k = 0, \cdots, N-1$，则有

$$\begin{aligned}\sum_{k=1}^{N}[\mathcal{L}(x_k, Y) - \mathcal{L}(x, Y_k)] \leq{}& \sum_{k=0}^{N-1}[\frac{1}{\tau}D_x(x, x_k) + \frac{1}{\sigma}D_Y(Y, Y_k) - \langle K(x - x_k), Y - Y_k\rangle] \\ &- \sum_{k=1}^{N}[\frac{1}{\tau}D_x(x, x_k) + \frac{1}{\sigma}D_Y(Y, Y_k) - \langle K(x - x_k), Y - Y_k\rangle]\end{aligned} \tag{4.40}$$

对不等号右边进行同项相消可以得到

$$\sum_{k=1}^{N}[\mathcal{L}(x_k,Y)-\mathcal{L}(x,Y_k)]=\frac{1}{\tau}D_x(x,x_0)+\frac{1}{\sigma}D_Y(Y,Y_0)-\langle K(x-x_0),Y-Y_0\rangle \tag{4.41}$$

$$-[\frac{1}{\tau}D_x(x,x_N)+\frac{1}{\sigma}D_Y(Y,Y_N)-\langle K(x-x_N),Y-Y_N\rangle]$$

根据条件 *(4.56)* 可知方括号内大于等于零，因此有

$$\sum_{k=1}^{N}[\mathcal{L}(x_k,Y)-\mathcal{L}(x,Y_k)]\leq\frac{1}{\tau}D_x(x,x_0)+\frac{1}{\sigma}D_Y(Y,Y_0)-\langle K(x-x_0),Y-Y_0\rangle \tag{4.42}$$

根据函数 $\mathcal{L}(\cdot,x)-\mathcal{L}(Y,\cdot)$ 的凸性，有

$$\frac{1}{N}\sum_{k=1}^{N}[\mathcal{L}(x_k,Y)-\mathcal{L}(x,Y_k)]\geq\mathcal{L}\left(\frac{\sum_{t=1}^{N}}{N}x_t,Y\right)-\mathcal{L}\left(x,\frac{\sum_{t=1}^{N}Y_t}{N}\right) \tag{4.43}$$

$$=\mathcal{L}(\bar{X},Y)-\mathcal{L}(x,\bar{Y}) \tag{4.44}$$

将 *(4.63)* 与 *(4.61)* 合并，可以得到

$$\mathcal{L}(\bar{X},Y)-\mathcal{L}(x,\bar{Y})\leq\frac{1}{\tau}D_x(x,x_0)+\frac{1}{\sigma}D_Y(Y,Y_0)-\langle K(x-x_0),Y-Y_0\rangle \tag{4.45}$$

由此我们证明了定理 *4.4.3*。

更进一步我们有

$$\mathcal{L}(\bar{X},\bar{Y})\leq\mathcal{L}(x^*,\bar{Y})+\frac{1}{N}(\frac{1}{\tau}D_x(x^*,x_0)+\frac{1}{\sigma}D_Y(\bar{Y},Y_0)-\langle K(x^*-x_0),\bar{Y}-Y_0\rangle)) \tag{4.46}$$

$$\leq\mathcal{L}(x^*,Y^*)+\frac{1}{N}(\frac{1}{\tau}D_x(x^*,x_0)+\frac{1}{\sigma}D_Y(Y^*,Y_0)-\langle K(x^*-x_0),Y^*-Y_0\rangle)) \tag{4.47}$$

$$\leq\mathcal{L}(x^*,Y^*)+\frac{c}{N},c>0 \tag{4.48}$$

$$\mathcal{L}(\bar{X},\bar{Y})-\mathcal{L}(x^*,Y^*)\leq\frac{c}{N} \tag{4.49}$$

其中第二个不等式是因为固定 x, $\mathcal{L}(\cdot,Y)$ 是一个最大化问题，第三个不等式是因为条件 *(4.56)*。因此我们证明了 *NLPD* 算法具有 $O(\frac{1}{N})$ 收敛速率。

4.4 近似迭代重加权算法

近似迭代重加权算法 (AIRWA) 的问题和假设与迭代重加权算法 (IRWA) 相同，区别在于两者针对每步迭代的子问题的求解方式有所不同。首先回顾 IRWA 算法，其目标是求解问题

$$\min_{x\in\mathcal{X}} \hat{J}(x,\epsilon) = f(x) + \lambda \sum_{i=1}^{m} \sqrt{dist_2^2(x \mid C_i) + \epsilon_i^2}, \tag{4.50}$$

而这可以通过多次迭代，每次迭代精确求解子问题

$$\min_{x\in\mathcal{X}} \hat{G}(x \mid x_t, \epsilon_t) = f(x) + \frac{\lambda}{2} \sum_{i=1}^{m} w_i^t(||x - \mathcal{P}_{C_i}(x_t)||_2^2), \tag{4.51}$$

实现。

近似迭代重加权算法 (AIRWA) 是在迭代重加权算法 (IRWA) 上进行改良获得。由于迭代重加权算法 (IRWA) 需要精确求解子问题，所以会产生较大计算量。近似迭代重加权算法使用非线性原对偶算法 (NLPD) 近似求解 IRWA 中需要精确求解的子问题 (4.73)，其他保持不变，从而降低整体算法的计算量，提升算法速率。尽管近似求解子问题因为存在误差所以会导致迭代的次数增加，但是每次求解问题的计算量是下降的，这是由于不需要进行求逆等需要大计算量的操作。在整体算法层面，近似迭代重加权算法的求解速率相较迭代重加权算法也是更快的。根据 4.4 节分析，我们证明了 NLPD 算法的复杂度是 $O(\frac{1}{N})$。在第五章数值实验中，我们研究了大规模二次规划问题，证实近似迭代重加权算法比迭代重加权算法所需求解时间更短。

Algorithm 4 近似迭代重加权算法 (AIRWA)

输入: 选择初始点 $x_0, \eta \in (0,1)$, 权重 $\lambda > 0, \gamma > 0, \tau > 0, \sigma > 0, M > 0, K > 0$

输出: x_t

for $t = 0, 1, 2, \ldots$ **do**

求投影 $\mathcal{P}_{C_i}(x_t)$

求解子问题

$$x_{t+1} = NLPD(x_t, W_t, \tau, \sigma, K) \tag{4.52}$$

非线性原对偶算法:

选择初始点 $x_0 = x_t$, 权重 $W = W_t$

for $k = 0, 1, 2, \ldots$ **do**

求解子问题

$$x_{k+1} = \arg\min_x f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + g(x) + \langle WAx, Y^k \rangle + \frac{1}{\tau} D_x(x, x_k) \tag{4.53}$$

$$Y_{k+1} = \arg\min_Y h^*(Y) - \langle WA(2x_{k+1} - x_k), Y \rangle + \frac{1}{\sigma} D_Y(Y, Y_k) \tag{4.54}$$

if k = K **then**

break;

end if

k = k+1

end for

if $||x_{t+1} - x_t|| < M(\epsilon_i^t)^{1+\gamma}$ **then**

更新 α_{t+1} 和 ϵ_{t+1}

end if

非线性原对偶算法结束。

if $||x_{t+1} - x_t|| < \sigma_1$ and $||\epsilon^t||_2 < \sigma_2$ **then**

break;

end if

t = t+1

end for

第 5 章　数值实验

在本节中，我们通过二次规划问题的数值实验来验证近似迭代重加权算法 (IRWA) 的可行性，并与迭代重加权算法 (IRWA) 进行比较。比较内容主要以迭代次数，运行总时长以及 cpu 运行时间等为主。所有代码均通过 Python 实现，并在 AMD Ryzen 9 5900HX 3.3GHz*16 和 28GB 内存的机器上测试。

5.1　问题建模

典型二次规划问题可以被表示成以下形式：

$$\begin{aligned} \min_x \ & f(x) = \frac{1}{2}\|Hx - g\|_2^2 \\ \text{s.t.} \quad & A_i x = b_i, \quad i = 1, 2, ..., N \end{aligned} \tag{5.1}$$

其中矩阵 $H \in \mathcal{R}^{m\times n}$, 矩阵 $A_i \in \mathcal{R}^{m\times n}$，向量 $g \in \mathcal{R}^{m\times 1}$, 向量 $b_i \in \mathcal{R}^{m\times 1}$, 在大多数情况下可以假设 $m \le n$。

5.2　算法迭代分析

5.2.1　IRWA 算法

对 IRWA 算法, 需求解下列子问题：

$$x_{t+1} = \arg\min_x f(x) + \frac{\lambda}{2}\sum_{i=1}^{N} w_i^t(||x - P_{C_i}(x_t)||_2^2) \tag{5.2}$$

由于目标函数是凸函数，因此目标函数最小值在梯度等于 0 时取得，因此问题等价为:

$$\nabla f(x_t) + \frac{\lambda}{2}\sum_{i=1}^{N} w_i^t(x - P_{C_i}(x_t)) = 0 \tag{5.3}$$

其中 $\nabla f(x_t) = H^T(Hx_t - g)$, $P_{C_i}(x_t) = \arg\min_{x\in C_i} ||x - x_t||_2$ 将其写作优化问题的形式可以得到：

$$\begin{aligned} \min_x \ & \frac{1}{2}||x - x_t||_2^2 \\ \text{s.t.} \quad & A_i x = b_i \end{aligned} \tag{5.4}$$

其拉格朗日方程为：

$$\mathcal{L}(x,\lambda)=\frac{1}{2}||x-x_t||_2^2+\lambda^T(A_ix-b_i) \tag{5.5}$$

对 (x,λ) 求偏导并解方程可得 $P_{C_i}(x_t)=x_t-A_i^T(A_iA_i^T)^\dagger(A_ix_t-b_i)$

因此对 IRWA 来说每步的迭代为解方程：

$$(H^TH+(\lambda\sum_{i=1}^{N}w_i^t)I)x_{t+1}=H^Tg+\lambda\sum_{i=1}^{N}w_i^t(x_t-A_i^T(A_iA_i^T)^\dagger(A_ix_t-b_i)) \tag{5.6}$$

这一般通过对 x_{t+1} 左侧矩阵取逆得到。

5.2.2 AIRWA 算法

对 AIRWA 算法，同样需要解决子问题：

$$x_{t+1}=\arg\min_x f(x)+\frac{\lambda}{2}\sum_{i=1}^{N}w_i^t(||x-P_{C_i}(x_t)||_2^2) \tag{5.7}$$

在 AIRWA 算法中通过使用非线性原对偶算法近似求解子问题，每步更新需求解：

$$x_{k+1}=\arg\min_x F(x)=f(x_k)+\langle\nabla f(x_k),x-x_k\rangle+g(x)+\langle WAx,\tilde{Y}\rangle+\frac{1}{\tau}D_x(x,x_k) \tag{5.8}$$

$$Y_{k+1}=\arg\min_Y G(Y)=h^*(Y)-\langle WA\tilde{x},Y\rangle+\frac{1}{\delta}D_Y(Y,Y_k)$$

即求解:

$$\begin{aligned}\nabla F(x)&=\nabla f(x_k)+\tilde{Y}^Tw+\frac{1}{\tau}(x-x_k)=0\\ \nabla G(Y)&=\frac{1}{\lambda}Y+B-WAx+\frac{1}{\delta}(Y-Y_k)=0\end{aligned} \tag{5.9}$$

因此对非线性对偶算法每步的迭代为:

$$\begin{aligned}x_{k+1}&=x_k-\tau(\tilde{Y}^Tw+H^T(Hx_k-g))\\ Y_{k+1}&=\frac{\lambda\delta}{\lambda+\delta}(WA\tilde{x}-B+\frac{1}{\delta}Y_k)\end{aligned} \tag{5.10}$$

针对外层算法每步的更新则为非线性对偶算法 (NLPD) 求解子问题获得的近似解:

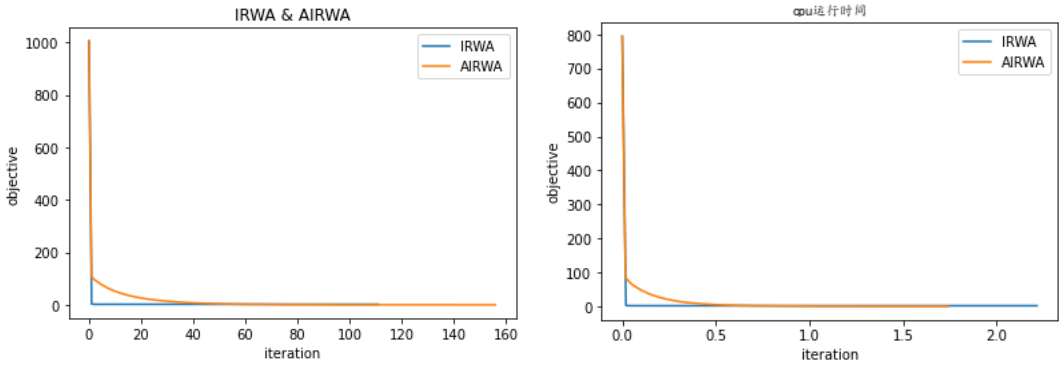
$$x_{t+1}=NLPD(x_t) \tag{5.11}$$

5.3 实验数据

对二次规划问题，我们选择以下生成数据方式: 矩阵大小 m 与 n 以及约束个数 N 由不同实验决定；矩阵 $H \in \mathcal{R}^{m\times n}$ 中的任意元素服从 $\mathcal{U}(-10,10)$，随机生成；向量 $g \in \mathcal{R}^{m\times 1}$ 中的任意元素服从 $\mathcal{U}(-10,10)$，随机生成；针对约束矩阵 $A_i \in \mathcal{R}^{m\times n}, i = 1,...,N$，每个矩阵中的任意元素服从 $\mathcal{U}(-10,10)$，随机生成；针对每个约束矩阵 A_i 所对应的向量 b_i, 随机生成向量 u_i 服从 $\mathcal{U}(-1,1)$, 令 $b_i = A_i u_i$。

5.4 实验一：大规模二次规划

在本实验中，选择矩阵满足 n=256，m=64，约束个数满足 N=10，初始迭代点 $x_0 = e$，其余参数满足 $\lambda = 100, \epsilon_0 = e^T \in \mathcal{R}^{1\times 10}$，$\eta = 0.02e^T \in \mathcal{R}^{1\times 10}$，$\gamma = 0.15$, $M = 100$，程序在 $||x_{t+1} - x_t||_2 \le 10^{-4}$ 且 $||\epsilon_t||_2 \le 10^{-5}$ 或者迭代次数 > 5000 时停止。针对 AIRWA 算法中的子问题求解部分，选择参数满足 $\tau = \theta = 0.00001$, $x_0 = x_t$。程序在 $||x_{k+1} - x_k||_2 < 10^{-2}$ 且 $||Y_{K+1} - Y_K||_F < 10^{-2}$ 或者迭代次数 > 40 时停止。



	迭代次数	时长 (s)
IRWA	112	2.222955274581909
AIRWA	137	1.7419405469894409

通过上述图片以及图表可以得到结论：

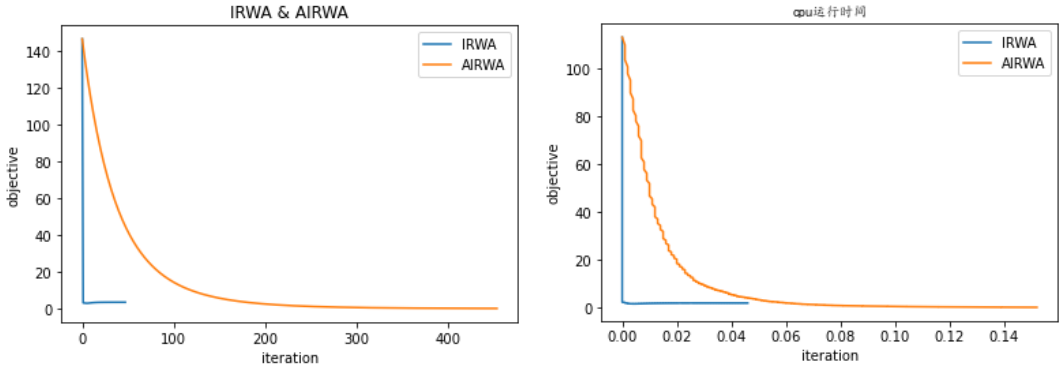
1) IRWA 与 AIRWA 本质通过相同方法求解目标函数最小值，因此两个算法最终目标函数值趋于相同

2) AIRWA 对子问题进行近似求解，因此迭代次数多于 IRWA。

3) AIRWA 通过对子问题近似求解，减少了原先 IRWA 在求解子问题上的计算量，因此总时长 AIRWA 小于 IRWA。

5.5 实验二：小规模二次规划

在本实验中，选择矩阵满足 n=32，m=8，约束个数满足 N=2，初始迭代点 $x_0 = e$，其余参数满足 $\lambda = 100, \epsilon_0 = e^T \in \mathcal{R}^{1\times 2}$，$\eta = 0.02e^T \in \mathcal{R}^{1\times 10}$，$\gamma = 0.15$, $M = 100$，程序在 $||x_{t+1} - x_t||_2 \leq 10^{-4}$ 且 $||\epsilon_t||_2 \leq 10^{-5}$ 或者迭代次数 > 5000 时停止。针对 AIRWA 算法中的子问题求解部分，选择参数满足 $\tau = \theta = 0.00001$, $x_0 = x_t$。程序在 $||x_{k+1} - x_k||_2 < 10^{-2}$ 且 $||Y_{K+1} - Y_K||_F < 10^{-2}$ 或者迭代次数 > 40 时停止。



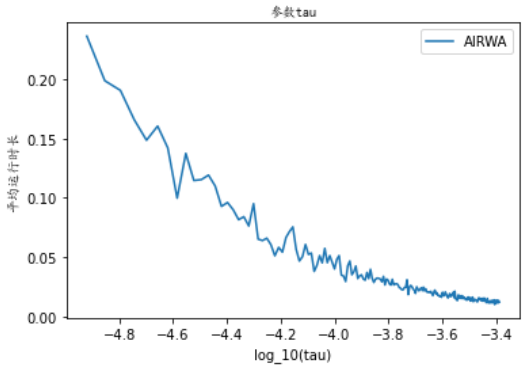
	迭代次数	时长 (s)
IRWA	47	0.046005651473999023
AIRWA	465	0.14104764938354492

通过上述图片以及图表可以得到结论：

1) 在小规模问题上 IRWA 求解精确子问题的计算量小，而通过原对偶算法近似求解子问题反而会增加总的迭代次数和运行时间，因此在小规模问题上 IRWA 算法的表现好于 AIRWA 算法。

5.6 实验三：参数 τ 探究

针对实验二中参数 τ 进行调整，以 10 个实验为一组求平均运行时长。



可以看到当 τ 趋近于 $10^{-3.4}$ 时算法表现最优，但是如果再增大 τ 的值将会导致程序出现平凡的崩溃，这是由于引理 4.4.2 的限制。

第 6 章　总结与展望

问题 (1.1) 经常出现在各种机器学习，大尺度优化问题中。在过去数十年间人们提出了许多一阶算法来求解约束集拥有高效计算投影的算法的问题，但是对于没有此类高效算法的约束集为多个集合交集的问题依然具有挑战。迭代重加权算法 (IRWA) 对该问题进行了研究，具有重要价值，然而该算法需要精确求解子问题，这会导致巨大的计算量。

本文的贡献在于：

(1) 设计非线性原对偶算法近似求解 IRWA 算法中的子问题。

(2) 证明了非线性对偶算法的收敛速率。

(3) 通过使用非线性对偶算法近似求解子问题，设计近似迭代重加权算法 (AIRWA)，提升了 IRWA 算法的计算速率。

(4) 使用二次规划问题进行数值实验，证明了 AIRWA 算法在性能上表现由于 IRWA 算法。

本文的工作还有许多不足，如未进行 AIRWA 算法的整体收敛性证明；算法只适用于满足 L-smooth 的函数等。这些问题还有待进一步的研究和改进。

具体改进如下：

(1) 对 AIRWA 算法进行整体收敛性证明。

(2) 对目标函数为非光滑函数的情况进行算法设计。

(3) 对 AIRWA 算法中参数的选择进行进一步的探讨。

参考文献

[1] Burke J V, Curtis F E, Wang H, et al. Matrix-Free Solvers for Exact Penalty Subproblems[J]. arXiv preprint arXiv:1402.1917, 2014.

[2] Chambolle A, Pock T. On the ergodic convergence rates of a first-order primal-dual algorithm[J]. Mathematical Programming, 2016, 159(1): 253-287.

[3] Kundu A, Bach F, Bhattacharya C. Convex optimization over intersection of simple sets improved convergence rate guarantees via an exact penalty approach[C]//International Conference on Artificial Intelligence and Statistics. PMLR, 2018: 958-967.

[4] Beck A, Teboulle M. Mirror descent and nonlinear projected subgradient methods for convex optimization[J]. Operations Research Letters, 2003, 31(3): 167-175.

[5] Chambolle A, Pock T. A first-order primal-dual algorithm for convex problems with applications to imaging[J]. Journal of mathematical imaging and vision, 2011, 40(1): 120-145.

[6] Esser E, Zhang X, Chan T F. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science[J]. SIAM Journal on Imaging Sciences, 2010, 3(4): 1015-1046.

[7] Beaton A E, Tukey J W. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data[J]. Technometrics, 1974, 16(2): 147-185.

[8] Schlossmacher E J. An iterative technique for absolute deviations curve fitting[J]. Journal of the American Statistical Association, 1973, 68(344): 857-859.

[9] Chen G, Teboulle M. Convergence analysis of a proximal-like minimization algorithm using Bregman functions[J]. SIAM Journal on Optimization, 1993, 3(3): 538-543.

[10] Burke J V. Descent methods for composite nondifferentiable optimization problems[J]. Mathematical Programming, 1985, 33(3): 260-279.

[11] Burke J V. Second order necessary and sufficient conditions for convex composite NDO[J]. Mathematical programming, 1987, 38(3): 287-302.

[12] Burke J V. A sequential quadratic programming method for potentially infeasible mathematical programs[J]. Journal of Mathematical Analysis and Applications, 1989, 139(2): 319-351.

[13] Fletcher R. A model algorithm for composite nondifferentiable optimization problems[M]// Nondifferential and Variational Techniques in Optimization. Springer, Berlin, Heidelberg, 1982: 67-76.

[14] Goldfarb D, Ma S, Scheinberg K. Fast alternating linearization methods for minimizing the sum of two convex functions[J]. Mathematical Programming, 2013, 141(1): 349-382.

[15] Goldstein T, O'Donoghue B, Setzer S, et al. Fast alternating direction optimization methods [J]. SIAM Journal on Imaging Sciences, 2014, 7(3): 1588-1623.

[16] Gould N I M, Lucidi S, Roma M, et al. Solving the trust-region subproblem using the Lanczos method[J]. SIAM Journal on Optimization, 1999, 9(2): 504-525.

[17] Nesterov Y E. A method for solving the convex programming problem with convergence rate O (1/k^ 2)[C]//Dokl. akad. nauk Sssr. 1983, 269: 543-547.

[18] O'Leary D P. Robust regression computation using iteratively reweighted least squares[J]. SIAM Journal on Matrix Analysis and Applications, 1990, 11(3): 466-480.

[19] Alvarez F, Attouch H. An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping[J]. Set-Valued Analysis, 2001, 9(1): 3-11.

[20] Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems[J]. SIAM journal on imaging sciences, 2009, 2(1): 183-202.

[21] Chen G, Teboulle M. Convergence analysis of a proximal-like minimization algorithm using Bregman functions[J]. SIAM Journal on Optimization, 1993, 3(3): 538-543.

[22] Chen Y, Lan G, Ouyang Y. Optimal primal-dual methods for a class of saddle point problems[J]. SIAM Journal on Optimization, 2014, 24(4): 1779-1814.

[23] Drori Y, Sabach S, Teboulle M. A simple algorithm for a class of nonsmooth convex-concave saddle-point problems[J]. Operations Research Letters, 2015, 43(2): 209-214.

[24] Eckstein J, Bertsekas D P. On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators[J]. Mathematical Programming, 1992, 55(1): 293-318.

[25] Boyle J P, Dykstra R L. A method for finding projections onto the intersection of convex sets in Hilbert spaces[M]//Advances in order restricted statistical inference. Springer, New York, NY, 1986: 28-47.

[26] Bauschke H H, Borwein J M. On projection algorithms for solving convex feasibility problems[J]. SIAM review, 1996, 38(3): 367-426.

[27] Artacho F J A, Borwein J M, Tam M K. Douglas − Rachford feasibility methods for matrix completion problems[J]. The ANZIAM Journal, 2014, 55(4): 299-326.

[28] Zhang T, Ando R K. Analysis of spectral kernel design based semi-supervised learning[J]. Advances in neural information processing systems, 2005, 18.

[29] d'Aspremont A, Ghaoui L, Jordan M, et al. A direct formulation for sparse PCA using semidefinite programming[J]. Advances in neural information processing systems, 2004, 17.

[30] Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms [M]. Society for Industrial and Applied Mathematics, 2010.

[31] Zaslavskiy M, Bach F, Vert J P. A path following algorithm for the graph matching problem [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 31(12): 2227-2242.

[32] Juditsky A, Nemirovski A. First order methods for nonsmooth convex large-scale optimization, i: general purpose methods[J]. Optimization for Machine Learning, 2011, 30(9): 121-148.

[33] Juditsky A, Nemirovski A. First order methods for nonsmooth convex large-scale optimization, ii: utilizing problems structure[J]. Optimization for Machine Learning, 2011, 30(9): 149-183.

[34] Combettes C, Pokutta S. Boosting Frank-Wolfe by chasing gradients[C]//International Conference on Machine Learning. PMLR, 2020: 2111-2121.

[35] Jaggi M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization[C]// International Conference on Machine Learning. PMLR, 2013: 427-435.

致　谢

感谢王浩教授给予的指导。王浩教授在大三上的数值最优化课程中将我领进了数值优化这一个研究方向，使我了解了最基本的非线性优化的各种算法。在经过大三下的机器学习，凸优化课程后，我了解了更多的优化算法包括与本问密切相关的原对偶算法。我于大三暑假进入王浩老师的课题组并接触到了与重加权算法相关的文章，自此开始了对相关问题的研究。在研究过程中王浩老师给出了许多知识上的指导，然后同一课题组的石乾坤学长也参与了相关问题的研究，他和我一同探讨问题，也教会了我许多优化研究相关的知识，对本文的完成也起到了十分重要的帮助作用，在此也十分感谢石乾坤学长给予我的帮助。最后也感谢上海科技大学给予了我能够探寻感兴趣的研究方向并能够学习相关知识，研究相关问题的机会。