

基于稀疏逻辑回归的加速分裂增广拉格朗日法

周璇, 党亚峥, 汪默然

上海理工大学管理学院, 上海

收稿日期: 2021年11月15日; 录用日期: 2021年12月17日; 发布日期: 2021年12月27日

摘要

近几年来,稀疏优化在最小二乘、压缩感知尤其在统计学的逻辑回归模型中均得到了极大的研究及关注。逻辑回归是经典的分类方法,广泛应用于数据挖掘、机器学习、生物信息学。现研究带有 l_0 范数约束的稀疏逻辑回归问题,为了求解此类问题,提出了加速分裂增广拉格朗日法(ASALM)。首先我们通过变量分裂,将稀疏逻辑回归问题转化为带有一个等式约束和一个基数约束的优化问题。接着在分裂增广拉格朗日法(SALM)中运用Nesterov加速技术,来提高算法的全局收敛性。然后在算法迭代中交替地求解一个无约束凸优化问题和一个带基数约束的二次优化问题,同时证明了ASALM的收敛性。最后,通过数值实验说明了ASALM的有效性和可行性,并且证明了ASALM的收敛速度比SALM更快。

关键词

稀疏逻辑回归, 变量分裂, Nesterov加速, 增广拉格朗日法

Accelerated Splitting Augmented Lagrange Method Based on Sparse Logistic Regression

Xuan Zhou, Yazheng Dang, Moran Wang

Business School, University of Shanghai for Science and Technology, Shanghai

Received: Nov. 15th, 2021; accepted: Dec. 17th, 2021; published: Dec. 27th, 2021

Abstract

In recent years, sparse optimization has received great research and attention in the least squares, compressed sensing, especially in statistical logistic regression models. Logistic regression is a classic classification method, which is widely used in data mining, machine learning, and bioin-

formatics. Now we study the sparse logistic regression problem with l_0 norm constraints. In order to solve such problems, the accelerated split augmentation Lagrangian method (ASALM) is proposed. First, we use variable splitting to transform the sparse logistic regression problem into an optimization problem with an equality constraint and a cardinality constraint. Next, Nesterov's acceleration technology is used in the split augmented Lagrangian method (SALM) to improve the global convergence of the algorithm. Then, an unconstrained convex optimization problem and a quadratic optimization problem with cardinality constraints are alternately solved in the algorithm iteration; at the same time, the convergence of ASALM is proved. Finally, numerical experiments demonstrate the effectiveness and feasibility of ASALM, and prove that the convergence speed of ASALM is faster than SALM.

Keywords

Sparse Logistic Regression, Splitting Variable, Nesterov's Acceleration, Augmented Lagrangian Method

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在如今的大数据时代下，稀疏优化的应用十分广泛，机器学习、压缩感知、分子生物学等领域都有很大的研究与发展。近几年来，稀疏优化在统计学的逻辑回归模型中得到了极大的研究及关注。逻辑回归分析是一种有效的数据处理分类方法，被广泛应用于文本分类[1] [2]、机器学习、生物信息学[3] [4] [5] [6]等领域。经典的逻辑回归是一种线性的分类方法，具有训练速度快、模型的可解释性好等特点。本文考虑的是二项逻辑回归模型，通过给定的训练数据集，运用极大似然估计法估计模型参数得到逻辑回归模型。

在实际应用中，需要从大量特征中选择出对模型起关键作用的特征，排除与模型不相关或对模型作用很小的特征。同时也需要控制逻辑回归模型中参数的稀疏性，起到稀疏优化的效果。为了不造成特征泛滥，即需要控制模型中参数 w 非零元素的个数 $\|w\|_0$ 。而带有 l_0 范数的稀疏优化模型是 NP-难问题，并且模型是非凸的，这给我们的计算带来了挑战，为了克服这个困难，一般的方法是加入正则项来进行松弛，将 l_0 范数松弛为 l_1 范数是常用的方法。带有 l_1 范数的正则化模型有着很好的凸性，能够防止过拟合，同时能够满足得到模型稀疏解，而被广泛应用在各个领域。例如带有 l_1 范数的线性回归模型通常又记为 Lasso 模型[7]，在疾病分类中有着广泛的应用。添加 l_1 范数后的稀疏逻辑回归模型是一个凸优化问题，可以运用经典的凸优化方法去求解，例如一阶算法[8]、迭代算法[9]、内点算法[10]。

在文献[11]中，通过对 Nesterov 加速[12]的研究，提出了加速增广拉格朗日法(AALM)，建立了该方法的迭代复杂度为 $O(1/k^2)$ ，并且又推导出了增广拉格朗日法(ALM)的迭代复杂度为 $O(1/k)$ ，证明了 AALM 比 ALM 具有更好的加速效果。本文将 Nesterov 加速方法应用到 SALM 框架中，提高了算法的全局收敛速度。针对 ASALM 框架中无约束凸优化子问题，运用拟牛顿法(BFGS 算法)求解。针对 ASALM 框架中带基数约束的子问题，通过等价变换得到带 l_0 范数约束二次优化问题的显式解。

本文首先介绍了逻辑回归的基础知识，建立了逻辑回归模型和稀疏逻辑回归模型，并给出了最优性条件；然后针对稀疏逻辑回归模型，提出了加速分裂增广拉格朗日算法(ASALM)；又给出了该算法的一

阶收敛性证明；然后，通过数值实验说明了 ASALM 的有效性和可行性。最后，对全文进行了总结。

2. 稀疏逻辑回归

稀疏逻辑回归基于逻辑回归模型，并在逻辑回归模型基础上添加 l_0 范数约束来控制模型中非零参数的个数。在实际应用中，需要从大量特征中选择出对模型起关键作用的主要特征。例如：在文字识别中，为了选择出对文字有鉴别意义的特征，需要在逻辑回归模型中添加 l_0 范数约束来控制模型中非零参数的个数。下面将对稀疏逻辑回归模型给出简单的回顾。

2.1. 二项逻辑回归模型

给定 $x = (x_1, \dots, x_n)^T$ 表示输入变量，其中 x_i 表示 x 的第 i 个属性(特征)， $y \in \{-1, +1\}$ 表示对应输出变量， $y_i = -1$ 表示第 i 个样本 x^i 输出为负类， $y_i = +1$ 表示第 i 个样本 x^i 输出为正类。对于给定的输入样本 $x \in \mathbb{R}^n$ ，输出标签为 $y \in \{-1, +1\}$ 的条件概率为：

$$P(y = -1 | x) = \frac{1}{1 + e^{w^T x + b}},$$

$$P(y = +1 | x) = \frac{e^{w^T x + b}}{1 + e^{w^T x + b}},$$

其中 $w \in \mathbb{R}^n, b \in \mathbb{R}$ 分别为权值向量与偏差，也为模型的待估计参数。给定训练数据集

$D = \{(x^1, y_1), \dots, (x^N, y_N)\} \subset \mathbb{R}^n \times \{-1, +1\}$ ，其中 $y_i = -1$ 表示第 i 个样本 x^i 输出为负类， $y_i = +1$ 表示第 i 个样本 x^i 输出为正类。采用极大似然估计法估计参数 w, b ，等价变形后，通过求解极小化平均逻辑似然函数估计模型参数，得到如下逻辑回归模型。

$$P(1) \min_{w, b} l_{avg}(w, b) = \min \frac{1}{N} \sum_{i=1}^N \log(1 + \exp(-y_i (w^T x^i + b))),$$

其中 $w \in \mathbb{R}^n, b \in \mathbb{R}$ 为待估计参数。

2.2. 稀疏逻辑回归模型

稀疏逻辑回归是基于经典逻辑回归模型构建，并在经典逻辑回归模型基础上添加 l_0 范数约束来控制模型中非零参数的个数。因此稀疏逻辑回归模型构建为：

$$P(2) \min_{w, b} l_{avg}(w, b), \text{ s.t. } \|w\|_0 \leq K,$$

其中 $\|\cdot\|_0$ 为欧式范数， $K > 0$ 为控制 w 中非零元素个数的参数。带 l_0 范数约束的逻辑回归问题高度非线性，且是 NP-难问题。通过添加 l_1 范数，稀疏逻辑回归模型转化为带 l_1 正则项的凸优化问题：

$$P(3) \min_{w, b} l_{avg}(w, b) + \tau \|w\|_1,$$

其中 $\tau > 0$ 为正则化参数。凸优化问题可以用迭代算法[9]、内点算法[10]、一阶算法[8]等多种凸优化方法求解。

2.3. 最优性条件

下面给出稀疏逻辑回归的最优性条件[13] [14]。令 $\alpha = (b, w) \in \mathbb{R}^{n+1}$ ，即 $l_{avg}(b, w) = l_{avg}(\alpha)$ 。若 α^* 是稀疏逻辑回归模型的局部最优解，则存在 $\delta^* \in \mathbb{R}^{n+1}$ 与 α^* 满足如下的 KKT 条件：

$$\begin{aligned} \nabla l_{avg}(\alpha^*) + \delta^* &= 0, \\ \delta^* &= 0, \quad j \in J. \end{aligned} \quad (1.1)$$

其中 $J^* = \{1\} \cup \{2 \leq j \leq n+1: \alpha_j^* \neq 0\}$ 。

3. 加速分裂增广拉格朗日法(ASALM)

3.1. 变量分裂

稀疏逻辑回归模型是一个带有 l_0 范数约束的最优化问题，采用变量分裂的技术，将问题转化为带有等式约束和一个基数约束的最优化问题。由于 w 、 b 分别代表逻辑回归模型中权值向量和偏差。令 $\alpha = (b, w)$ ，则 $b = \alpha_1, w = \alpha_j$ ，其中 $J = \{2, \dots, n+1\}$ 。引入辅助变量 $\beta \in \mathbb{R}^n$ ，则稀疏逻辑回归模型等价转化为：

$$P(4) \min_{\alpha, \beta} l_{avg}(\alpha_1, \alpha_j), \quad \text{s.t. } \alpha_j = \beta, \|\beta\|_0 \leq K.$$

3.2. Nesterov 加速

Nesterov 提出一种用于梯度方法的加速策略，在 Nesterov 加速[12]算法中，引入一个或者多个辅助变量，并利用当前变量的梯度来更新参数。具体而言，在 SALM 框架中，拉格朗日乘子更新后得到 $\hat{\gamma}^k$ ，增加了对拉格朗日乘子预测并修正的步骤，如下：

$$\begin{aligned} t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \\ \gamma^{k+1} &= \hat{\gamma}^k + \frac{t_k - 1}{t_{k+1}}(\hat{\gamma}^k - \hat{\gamma}^{k-1}) + \frac{t_k}{t_{k+1}}(\hat{\gamma}^k - \gamma^k). \end{aligned}$$

3.3. 加速分裂增广拉格朗日法

关于等式约束 $\alpha_j = \beta$ 的增广拉格朗日函数形式为：

$$L_\rho(\alpha, \beta, \gamma) = l_{avg}(\alpha_1, \alpha_j) + \gamma^\top (\beta - \alpha_j) + \frac{\rho}{2} \|\beta - \alpha_j\|^2,$$

其中 $\gamma \in \mathbb{R}^n$ 为等式约束 $\alpha_j = \beta$ 对应的拉格朗日乘子， $\rho > 0$ 为罚参数。亦即：

$$L_\rho(\alpha, \beta, \gamma) = \frac{1}{N} \sum_{i=1}^N \log(1 + \exp(-B_i \alpha)) + \frac{\rho}{2} \left\| \alpha_j - \beta - \frac{1}{\rho} \cdot \gamma \right\|^2 - \frac{1}{2\rho} \|\gamma\|^2,$$

其中 B_i 为矩阵 B 的第 i 行，

$$B = \begin{pmatrix} y_1 & y_1 (x^1)^\top \\ y_2 & y_2 (x^2)^\top \\ \vdots & \vdots \\ y_N & y_N (x^N)^\top \end{pmatrix} \in \mathbb{R}^{N \times (n+1)}. \quad (2.1)$$

下面给出加速分裂增广拉格朗日法(ASALM)算法步骤：

初始步： 给定允许误差 $\varepsilon \geq 0$ ，罚参数 $\rho > 0$ ，初始化 $\hat{\gamma} \in \mathbb{R}^n$ ，初始化 β ，令 $\gamma^1 = \hat{\gamma}^0$ ， $t_1 = 1$ ， $k = 1$ 。

迭代步: 判断精度, 若 $\|\alpha_j^k - \beta^k\| \leq \varepsilon$ 输出 α^k , 停止计算;

令 $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$, 计算:

$$\alpha^{k+1} = \arg \min_{\alpha \in \mathbb{R}^{n+1}} L_\rho(\alpha, \beta^k, \gamma^k), \quad (2.2)$$

$$\beta^{k+1} = \arg \min_{\|\beta\|_0 \leq K} L_\rho(\alpha^{k+1}, \beta, \gamma^k), \quad (2.3)$$

$$\hat{\gamma}^k = \gamma^k + \rho(\beta^{k+1} - \alpha_j^{k+1}), \quad (2.4)$$

$$\gamma^{k+1} = \hat{\gamma}^k + \frac{t_k - 1}{t_{k+1}}(\hat{\gamma}^k - \hat{\gamma}^{k-1}) + \frac{t_k}{t_{k+1}}(\hat{\gamma}^k - \gamma^k), k := k + 1. \quad (2.5)$$

由 ASALM 算法迭代步可知, 子问题(2.2)是一个无约束光滑凸优化问题, 可以用拟牛顿法求解, 其核心思想是用不含二阶导数的矩阵 U_k 替代牛顿法中的 H_k^{-1} (Hesse 矩阵的逆矩阵), 然后沿搜索方向 $-U_k \cdot g_k$ 做一维搜索, 根据不同的 U_k 构造方法有不同的拟牛顿法。

令 $f(\alpha) = L_\rho(\alpha, \beta^k, \gamma^k)$, 具体拟牛顿法(BFGS 法)步骤如下:

初始步: 给定初始点 $\alpha^{(1)}$, 允许误差 ε , 设置 B_1^{-1} (通常取单位阵), $k = 1$ 。

迭代步: 1) 记 $g_k = \nabla f(\alpha^{(k)})$, 计算搜索方向 $d^{(k)} = -B_k^{-1} \cdot g_k$,

2) 从点 $\alpha^{(k)}$ 出发, 沿着 $d^{(k)}$ 做一维搜索, 获得最优步长并更新参数:

$$\lambda_k = \arg \min_{\lambda} f(\alpha^{(k)} + \lambda \cdot d^{(k)}),$$

$$\alpha^{(k+1)} = \alpha^{(k)} + \lambda_k \cdot d^{(k)},$$

3) 判断精度, 若 $\|g_{k+1}\| < \varepsilon$, 则停止迭代, 否则转(4);

4) 计算 $\Delta g = g_{k+1} - g_k, \Delta \alpha = \alpha^{(k+1)} - \alpha^{(k)}$ 。更新 B^{-1} :

$$B_{k+1}^{-1} = \left(I_{n+1} - \frac{\Delta \alpha \Delta g^T}{\Delta \alpha^T \Delta g} \right) B_k^{-1} \left(I_{n+1} - \frac{\Delta g \Delta \alpha^T}{\Delta \alpha^T \Delta g} \right) + \frac{\Delta \alpha \Delta \alpha^T}{\Delta \alpha^T \Delta g},$$

5) $k = k + 1$, 转 1)。

子问题(2.3)是一个带有基数约束的最优化问题, 通过等价变换可以转化为带有基数约束的二次优化问题如下:

$$\min_{\beta} \sum_{i=1}^n (\beta_i - \xi_i)^2,$$

$$\text{s.t. } \|\beta\|_0 \leq K.$$

其中 $\xi = \alpha_j - \gamma/\rho$, 且问题的最优解[14]为如下形式: $y_i^* = \xi_i, i \in M$ 。 $y_i^* = 0, i \notin M$ 。其中 $M \subset \{1, 2, 3, \dots, n\}$ 表示向量 ξ^2 的前 K 个最大元素所对应的指标集。

3.4. ASALM 算法的收敛性

ASALM 算法求解稀疏逻辑回归问题的一阶收敛性[13]如下:

令 $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$ 为由 ASALM 算法产生点列 $\{(\alpha^k, \beta^k, \gamma^k)\}_{k=1}^{\infty}$ 的聚点。假设 $\{\gamma^k\}_{k=1}^{\infty}$ 有界, 且

$$\lim_{k \rightarrow \infty} [(\alpha^{k+1}, \beta^{k+1}, \gamma^{k+1}) - (\alpha^k, \beta^k, \gamma^k)] = 0.$$

则 $\hat{\alpha}^k$ 满足问题 P(4)的一阶最优性条件(1.1)。

4. 数值实验

本章节主要通过数值实验来验证加速分裂增广拉格朗日算法(ASALM)的可行性和有效性。我们将所要研究的稀疏逻辑回归问题应用于随机合成的模拟数据,通过对实验结果的数据分析和绘图分析客观地描述了加速分裂增广拉格朗日算法的性能与效率。本次实验的代码由 MATLAB R2019a 编写,并且在 Intel(R) Core(TM) i5-9300H 2.4GHz CPU, 8GB RAM 的个人笔记本电脑上进行了数值实验。

我们考虑下列稀疏逻辑回归问题:

$$\min_{\alpha, \beta} \left\{ \frac{1}{N} \sum_{i=1}^N \log(1 + \exp(-B_i \alpha)) \right\}, \quad \text{s.t. } \alpha_j = \beta, \|\beta\|_0 \leq K. \quad (3.1)$$

其中 $\alpha = (\alpha_1, \alpha_j)$, B 是一个 $N \times (n+1)$ 矩阵如上(2.1)所示。设置罚参数 $\rho = 0.5$, $k = 1$, $t = 1$, 初始化 β^1, γ^0 为零向量, 即 γ^1 亦为零向量。重新定义矩阵 B 的维度尺寸为 $n \times m$, 定义 $r^{k+1} = \alpha_j^{k+1} - \beta^{k+1}$ 和 $s^{k+1} = -\rho(\beta^{k+1} - \beta^k)$ 分别为原始残差和对偶残差, 设置绝对误差 $\varepsilon^{abs} = 10^{-6}$ 和相对误差 $\varepsilon^{rel} = 10^{-3}$, 实验中的迭代停止准则为 $\|r^k\| \leq \varepsilon^{pri}$ 和 $\|s^k\| \leq \varepsilon^{dual}$, 其中 $\varepsilon^{pri} = \sqrt{n}\varepsilon^{abs} + \varepsilon^{rel} \max\{\|\alpha^k\|, \|\beta^k\|\}$, $\varepsilon^{dual} = \sqrt{n}\varepsilon^{abs} + \varepsilon^{rel} \|\gamma^k\|$ 。

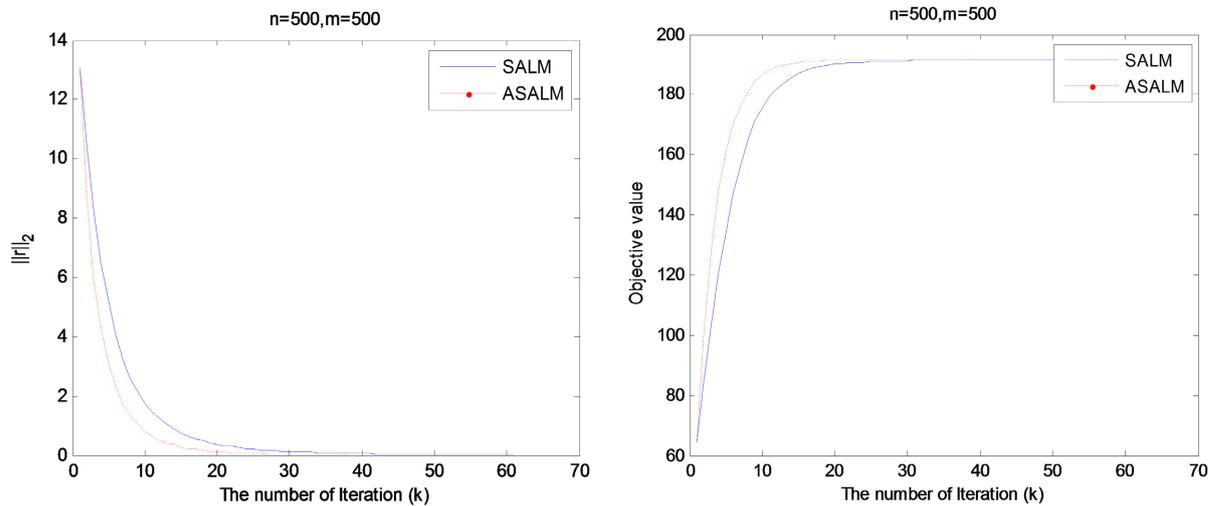
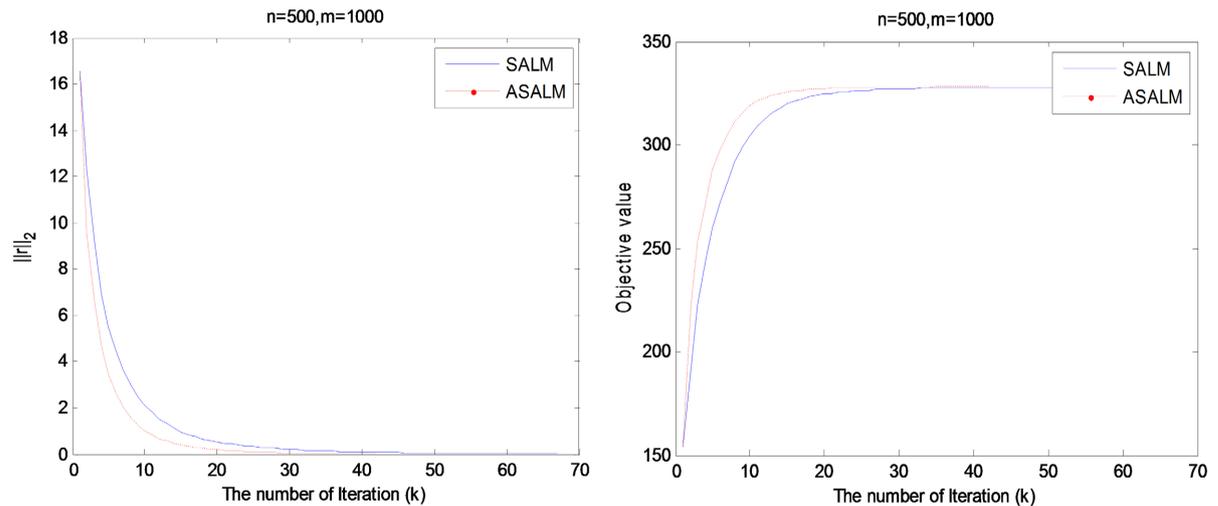
为了直观地描述 ASALM 的性能与效率,我们将 ASALM 与分裂增广拉格朗日算法(SALM)进行了实验对比,测试了四种不同维度的应用实例,即 $(n, m) = (500, 500), (500, 1000), (1000, 1000), (1000, 1500)$, 数值实验结果见表 1。在四种不同的维度条件下,表 1 中分别比较了两种算法的迭代次数、CPU 运算时间和目标函数值。从表 1 中可以分析得出 ASALM 比 SALM 有着更少的迭代次数和 CPU 运算时间。因此,针对问题(3.1) ASALM 是可行的且比 SALM 更有效。

为了进一步观察 ASALM 和 SALM 的收敛性,我们可视化了两种算法收敛性的变化,见如下图 1 和图 2。我们观察了 $(n, m) = (500, 500), (500, 1000)$ 两种条件下原始残差和目标函数值随着迭代次数(k)的变化情况,图 1 和图 2 都展示了 ASALM 的原始残差比 SALM 下降的更快,且 ASALM 比 SALM 更快达到最优目标函数值。因此,ASALM 的收敛速度比 SALM 更快,且性能与效率更好。

Table 1. Comparison of two algorithms based on simulated data

表 1. 基于模拟数据的两种算法比较

尺寸	算法	迭代次数/次	CPU 时间/秒	目标函数值
$n = 500, m = 500$	SALM	61	5.3689	191.77
	ASALM	38	4.2988	191.77
$n = 500, m = 1000$	SALM	67	11.4928	328.06
	ASALM	42	9.5626	328.07
$n = 1000, m = 1000$	SALM	51	28.1707	225.47
	ASALM	41	18.2236	225.49
$n = 1000, m = 1500$	SALM	72	42.0699	381.58
	ASALM	44	37.2547	381.59

Figure 1. $n = 500, m = 500$ 图 1. $n = 500, m = 500$ Figure 2. $n = 500, m = 1000$ 图 2. $n = 500, m = 1000$

5. 结论与展望

本文针对带有 l_0 范数约束的逻辑回归问题，提出了加速分裂增广拉格朗日法求解此类问题。该算法在迭代中交替求解两个子问题，借助 BFGS 算法求解其中一个子问题，并通过等价变形求解出另一个子问题的显式解。在 SALM 框架中，运用 Nesterov 加速技术，提高算法的全局收敛性，给出了 ASALM 算法的一阶收敛性证明。最后，通过数值实验说明了 ASALM 的收敛速度比 SALM 更快，且性能与效率更好。此类方法亦可以应用到带有范数约束的多项逻辑回归问题上，在提高全局收敛性的性能上优于其它凸优化方法。

参考文献

- [1] Maghbouleh, A. (1996) A Logistic Regression Model for Detecting Prominences. *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, Philadelphia, PA, 3-6 October 1996, 2443-2445.

-
- [2] Brzezinski, J.R. and Knafel, G.J. (1999) Logistic Regression Modeling for Context-Based Classification. *Tenth International Workshop on Database and Expert Systems Applications, DEXA 99*, Florence, 3 September 1999, 755-759. <https://doi.org/10.1109/DEXA.1999.795279>
- [3] Asgary, M.P., Jahandideh, S., Abdolmaleki, P., *et al.* (2007) Analysis and Identification of jS-Turn Types Using Multinomial Logistic Regression and Artificial Neural Network. *Bioinformatics*, **23**, 3125-3130. <https://doi.org/10.1093/bioinformatics/btm324>
- [4] Liao, J.G. and Chin, K.V. (2007) Logistic Regression for Disease Classification Using Microarray Data: Model Selection in a Large P and Small N Case. *Bioinformatics*, **23**, 1945-1951. <https://doi.org/10.1093/bioinformatics/btm287>
- [5] Sartor, M.A., Leikauf, G.D., Medvedovic, M., *et al.* (2009) LRpath: A Logistic Regression Approach for Identifying Enriched Biological Groups in Gene Expression Data. *Bioinformatics*, **25**, 211-217. <https://doi.org/10.1093/bioinformatics/btn592>
- [6] Zhu, J. and Hastie, T. (2004) Classification of Gene Microarrays by Penalized Logistic Regression. *Bio-Statistic*, **5**, 427-443. <https://doi.org/10.1093/biostatistics/kxg046>
- [7] Tibshirani, R. (1996) Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society: Series B*, **58**, 267-288. <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>
- [8] Liu, J., Ji, S. and Ye, J. (2016) SLEP: Sparse Learning with Efficient Projections. <http://yelabs.net/software/SLEP/>
- [9] Lee, S.I., Lee, H., Abbeel, P., *et al.* (2006) Efficient ℓ_1 Regularized Logistic Regression. Proceedings of the National Conference on Artificial Intelligence. AAAI Press, Massachusetts, 401-408.
- [10] Kim, S.J., Koh, K., Lustig, M., *et al.* (2007) An Interior-Point Method for Large-Scale ℓ_1 -Regularized Least Squares. *IEEE Journal of Selected Topics in Signal Processing*, **1**, 606-617. <https://doi.org/10.1109/JSTSP.2007.910971>
- [11] Wang, X., Wang, Y. and Wang, G. (2020) An Accelerated Augmented Lagrangian Method for Multi-Criteria Optimization Problem. *Journal of Industrial & Management Optimization*, **16**, 1-9. <https://doi.org/10.3934/jimo.2018136>
- [12] Nesterov, Y.Y.E. (1983) A Method for Solving the Convex Programming Problem with Convergence Rate $O(1/k^2)$. *Doklady Akademii Nauk SSSR*, **269**, 543-547.
- [13] Bai, Y.Q., Liang, R.L. and Yang, Z.W. (2016) Splitting Augmented Lagrangian Method for Optimization Problems with a Cardinality Constraint and Semicontinuous Variables. *Optimization Methods and Software*, **31**, 1089-1109. <https://doi.org/10.1080/10556788.2016.1196206>
- [14] Lu, Z. and Zhang, Y. (2013) Sparse Approximation via Penalty Decomposition Methods. *SIAM Journal on Optimization*, **23**, 2448-2478. <https://doi.org/10.1137/100808071>