改进的 OWL-QN 方法解稀疏 logistic 回归问题

薛伟^{①*}, 张文生^②

① 南京理工大学计算机科学与工程学院, 南京 210094;

② 中国科学院自动化研究所, 北京 100190

E-mail: 213106000062@njjust.edu.cn, wensheng.zhang@ia.ac.cn

收稿日期: 2014-08-07; 接受日期: 2014-11-17; 网络出版日期: 2015-11-18; * 通信作者

国家自然科学基金 (批准号: U1135005, 61432008 和 61472423) 和南京理工大学研究生科研创新计划 (批准号: 168) 资助项目

摘要 稀疏 logistic 回归, 是机器学习中一类重要的问题, 它在控制论、管理科学和互联网等领域有着广泛的应用. 本文提出了一种改进的基于分象限学习的拟 Newton 算法来求解稀疏 logistic 回归问题. 新算法采用著名的 Barzilai-Borwein 步长策略自适应地近似代替目标函数的 Hesse 阵, 并利用目标函数的整体梯度信息来构造拟 Newton 向量. 在适当的条件下, 证明了新算法的全局收敛性. 数值实验表明新算法是可行的, 并且是有效的.

关键词 无约束优化 logistic 回归 l_1 正则化 拟 Newton 方法 非单调线搜索 Barzilai-Borwein 步长

MSC (2010) 主题分类 65K05, 93B40

1 引言

典型的监督机器学习问题是通过已有类别的样本集学习出样本的一个关系模型, 并通过该模型对新样本的类别进行预测. 例如, 已知 m 个样本 $\mathbf{x}_i$ ($i = 1, 2, \dots, m$), 每个样本包含 n 个特征, 样本 $\mathbf{x}_i$ 对应的类别为 $y_i \in \{-1, +1\}$. 给定 $\mathbf{x}_i$, 我们的目标是建立一个以 $\mathbf{w} \in \mathbb{R}^n$ 为参数的关系模型 (在学习问题中, $\mathbf{w}$ 通常称为权向量), 进而通过该模型对样本 $\mathbf{x}_i$ 的类别进行预测^[1]. 在学习的过程中, 我们使用损失函数 $f(\mathbf{w}; \mathbf{x}_i, y_i)$ 来度量预测的类别和真实类别之间的差异. 常用的损失函数有

最小二乘回归^[1]: $(y - \mathbf{w}^T \mathbf{x})^2$;

Logistic 回归^[1]: $\log(1 + \exp(-y(\mathbf{w}^T \mathbf{x})))$;

l_1 hinge 损失^[2]: $\max(0, 1 - y(\mathbf{w}^T \mathbf{x}))$;

l_2 hinge 损失^[2]: $\max(0, 1 - y(\mathbf{w}^T \mathbf{x}))^2$.

此外, 模型的整体性能通过经验风险函数 $F(\mathbf{w})$ (该函数为所有损失函数的和) 来衡量. 为了避免学习过程中出现过拟合现象, 引入正则化算子 $R(\mathbf{w})$, 于是, 我们得到如下形式的优化问题:

$$\min_{\mathbf{w} \in \mathbb{R}^n} J(\mathbf{w}) \triangleq F(\mathbf{w}) + \lambda R(\mathbf{w}), \quad (1.1)$$

英文引用格式: Xue W, Zhang W S. An improved OWL-QN method for solving sparse logistic regression problems (in Chinese). Sci Sin Math, 2016, 46: 111–121, doi: 10.1360/012015-42

其中 $F(\mathbf{w}) = \sum_{i=1}^m f(\mathbf{w}; \mathbf{x}_i, y_i)$, λ 为权衡因子. 本文选取 $\log(1 + \exp(-y(\mathbf{w}^T \mathbf{x})))$ 作为损失函数, 极小化目标函数 J , 可以得到一个相对最优的权向量 $\mathbf{w}^*$. 进一步, 可以通过 $\mathbf{w}^*$ 对新样本的类别进行预测.

1.1 稀疏 logistic 回归

稀疏 logistic 回归, 也称 l_1 logistic 回归, 可以看成问题 (1.1) 的一种特殊情形, 是取 $R(\mathbf{w}) = \|\mathbf{w}\|_1$ (这里, $\|\mathbf{w}\|_1 = \sum_{j=1}^n |w^j|$), 具体表示如下:

$$\min_{\mathbf{w} \in \mathbb{R}^n} J(\mathbf{w}) \triangleq F(\mathbf{w}) + \sum_{j=1}^n \lambda^j |w^j|, \quad (1.2)$$

其中

$$F(\mathbf{w}) = \sum_{i=1}^m f(\mathbf{w}; \mathbf{x}_i, y_i),$$

$$f(\mathbf{w}; \mathbf{x}_i, y_i) = \log(1 + \exp(-y_i(\mathbf{w}^T \mathbf{x}_i))).$$

如果用 Euclid 范数 (l_2 范数, $\|\mathbf{w}\|_2 = (\sum_{j=1}^n |w^j|^2)^{\frac{1}{2}}$) 代替问题 (1.2) 中的 l_1 范数, 那么我们将得到一个基于 l_2 正则化的 logistic 回归模型. 由于 Euclid 范数是可微的, 所以 l_2 logistic 回归模型的使用比较普遍, 但是 l_1 在理论上可以产生稀疏的解, 能够提高算法的计算效率, 并且非零元更有助于重要的特征选取. 图 1 展示的是 l_1 和 l_2 正则化的几何解释. 从图 1 可以看出, l_1 正则化对应的约束区域类似于一个旋转了 90 度的正方形, l_2 正则化对应的约束区域则为一个圆形. 所谓的稀疏解可以理解为等高线第一次和约束区域相接触的点, 这个点是在含有零元的拐角处得到. 由图可见, l_1 正则化对应的解出现在拐角处的概率要高于 l_2 , 也就是说, l_1 正则化比 l_2 更具稀疏性. 然而, 采用 l_1 正则化时, 问题 (1.2) 的目标函数是不可微的, 这给问题的求解带来了困难. 在求解这类具有 l_1 正则项的 logistic 回归的问题上, 已经有很多的分解方法被提出来, 可参见文献 [2-5].

图 1 二维示意图

1.2 OWL-QN 方法

分象限学习方法 (OWL-QN, 见算法 1) 是由 Andrew 和 Gao 在文献 [6] 中提出的, 该方法是求解问题 (1.2) 的有效方法之一. Andrew 和 Gao 注意到, 给定一象限, l_1 正则化算子则是可微的. 此时, 问题 (1.2) 在给定的象限 (w^j 的符号保持不变) 内就是可微的, OWL-QN 方法便将解搜索空间限制在这样的象限内. 具体而言, 在第 k 步迭代时, OWL-QN 搜索解 $\mathbf{w}_{k+1}$ 的空间为 Ω_k , 其定义如下:

$$\Omega_k \triangleq \{\mathbf{w} \in \mathbb{R}^n \mid \text{sgn}(w^j) = \chi_k^j, j = 1, 2, \dots, n\}, \quad (1.3)$$

其中

$$\chi_k^j = \begin{cases} \text{sgn}(-\tilde{\nabla}_j J(\mathbf{w}_k)), & \text{如果 } w_k^j = 0, \\ \text{sgn}(w_k^j), & \text{其他,} \end{cases} \quad (1.4)$$

$$\tilde{\nabla}_j J(\mathbf{w}) = \begin{cases} \nabla_j F(\mathbf{w}) + \lambda^j, & \text{如果 } w^j > 0 \text{ 或 } w^j = 0, \nabla_j F(\mathbf{w}) < -\lambda^j, \\ \nabla_j F(\mathbf{w}) - \lambda^j, & \text{如果 } w^j < 0 \text{ 或 } w^j = 0, \nabla_j F(\mathbf{w}) > \lambda^j, \\ 0, & \text{其他,} \end{cases} \quad (1.5)$$

这里, $\text{sgn}(\cdot)$ 是符号函数, $\tilde{\nabla}_j J(\mathbf{w})$ 称为函数 J 在 $\mathbf{w}$ 处的“伪梯度”. 接着, OWL-QN 方法通过最优化目标函数 J 的一个二次近似函数 q_k 来获得当前点 $\mathbf{w}_k$ 处的伪搜索方向 $\bar{\mathbf{d}}_k$:

$$\begin{aligned} \min_{\bar{\mathbf{d}} \in \mathbb{R}^n} q_k(\bar{\mathbf{d}}) &\triangleq J(\mathbf{w}_k) + \tilde{\nabla} J(\mathbf{w}_k)^T \bar{\mathbf{d}} + \frac{1}{2} \bar{\mathbf{d}}^T H_k \bar{\mathbf{d}}, \\ \text{s.t. } \mathbf{w}_k + \bar{\mathbf{d}} &\in \Omega_k, \end{aligned} \quad (1.6)$$

其中 H_k 是 $J(\mathbf{w}_k)$ 的 Hesse 阵. 关于 H_k 的求得过程, 这里不作过多的描述, 有兴趣的读者可以参见文献 [7]. 采用二次函数模型的优化方法是普遍的, 这是因为二次函数模型简单, 容易求解. 对 (1.6) 右侧实施求导, 并令结果为 0, 得

$$\bar{\mathbf{d}}_k = -H_k^{-1} \tilde{\nabla} J(\mathbf{w}_k). \quad (1.7)$$

进一步, 为了得到搜索方向 $\mathbf{d}_k$, OWL-QN 方法将伪搜索方向 $\bar{\mathbf{d}}_k$ 投影到 $-\tilde{\nabla} J(\mathbf{w}_k)$ 所在象限, 即如果 $\bar{\mathbf{d}}_k$ 中某一元素的符号与 $-\tilde{\nabla} J(\mathbf{w}_k)$ 中对应位置的符号不一致, 就将该元素的值置为 0. 然后执行一

算法 1 OWL-QN

输入: 启动点 $\mathbf{w}_0$, 初始的拟 Newton 向量 S 和 Y , 权衡因子 λ . $k \leftarrow 0$.

While “不收敛”, Do

 由 (1.5) 计算 $\tilde{\nabla}_j J(\mathbf{w})$.

 通过 S 和 Y 计算得到 H_k , 然后根据 (1.7) 求得 $\bar{\mathbf{d}}_k$.

 根据 (1.9) 计算 $\mathbf{d}_k$.

 通过 (1.8) 和 (1.10) 寻找 $\mathbf{w}_{k+1} \in \Omega_k$.

 通过 $s_k = \mathbf{w}_{k+1} - \mathbf{w}_k$ 更新 S .

 通过 $y_k = \nabla F(\mathbf{w}_{k+1}) - \nabla F(\mathbf{w}_k)$ 更新 Y .

$k \leftarrow k + 1$

End Do

个单调的 Armijo 类线搜索. 给定 $\nu \in (0, 1)$, $\beta \in (0, 1)$, 选择最小的非负整数 t_k 使得步长因子 $\alpha = \beta^{t_k}$ 满足

$$J(\mathbf{w}_{k+1}) \leq J(\mathbf{w}_k) + \nu \tilde{\nabla} J(\mathbf{w}_k)(\mathbf{w}_{k+1} - \mathbf{w}_k). \quad (1.8)$$

以此来寻找步长 α 使得函数满足充分下降性条件 [8]. 用 $\mathbb{P}_S$ 表示符号投影算子, 那么搜索方向 $\mathbf{d}_k$ 的具体构造方式可表示为

$$\mathbf{d}_k^j = \mathbb{P}_S(\mathbf{d}) \triangleq \begin{cases} \bar{d}_k^j, & \text{如果 } \bar{d}_k^j(-\tilde{\nabla}_j J(\mathbf{w}_k)) > 0, \\ 0, & \text{其他.} \end{cases} \quad (1.9)$$

考虑到 $\mathbf{w}_{k+1}$ 必须在给定的象限内, 于是, Andrew 和 Gao 考虑将 $\mathbf{w}_{k+1}$ 中与 $\mathbf{w}_k$ 符号不一致的元素置为 0. 为方便起见, 用 $\mathbb{P}_O$ 表示象限投影算子, 得

$$\mathbf{w}_{k+1}^j = \mathbb{P}_O(\mathbf{w} + \alpha \mathbf{d}) \triangleq \begin{cases} w_k^j + \alpha d_k^j, & \text{如果 } w_k^j(w_k^j + \alpha d_k^j) > 0, \\ 0, & \text{其他.} \end{cases} \quad (1.10)$$

我们注意到 OWL-QN 方法在计算拟 Newton 向量 S 和 Y 的时候, 仅利用了损失函数的梯度信息, 也就是说他们没有充分利用目标函数的整体梯度信息. 另外, Andrew 和 Gao 采用的是单调的线搜索技术来寻找步长, 这种线搜索技术要求函数值是单调下降的, 这势必会影响算法的性能.

为了解决上述两个问题, 我们提出了一个改进的 OWL-QN 方法, 并将新算法命名为 IOWL-QN. 我们的主要思想是利用整体目标函数的梯度信息构造拟 Newton 向量 S 和 Y , 同时, 为了保证新算法具有全局收敛性, 我们引入了非单调线搜索技术. 此外, 新算法还引入了一个自适应近似代替目标函数 Hesse 阵的策略. 下面, 我们将对新算法 IOWL-QN 展开介绍.

2 预备知识

我们考虑如下不带正则项的优化问题:

$$\min F(x), \quad x \in \mathbb{R}^n, \quad (2.1)$$

这里, $F(x)$ 是二次连续可微的. 在第 k 次迭代, 当前迭代点为 x_k , 求解问题 (2.1) 的梯度类迭代方法可以表示为 $x_{k+1} = x_k - \theta_k \nabla F_k$ ($k = 0, 1, 2, \dots$), $\theta_k > 0$ 是步长因子. Barzilai 和 Borwein (BB) 在 1988 年提出了一个非常精美的梯度方法, 具体可参见文献 [9]. 该方法将步长 θ_k 定义为

$$\theta_k^{\text{BB1}} = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}, \quad (2.2)$$

或

$$\theta_k^{\text{BB2}} = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \quad (2.3)$$

其中 $s_{k-1} = x_k - x_{k-1}$, $y_{k-1} = \nabla F_k - \nabla F_{k-1}$. 不难看出 $\theta_k^{\text{BB1}} \geq \theta_k^{\text{BB2}}$, 利用这样的关系, 在每次迭代的时候, 我们便可以通过以下方式自适应地选取 θ_k ,

$$\theta_k = \begin{cases} \theta_k^{\text{BB1}}, & \text{如果 } \frac{\|s_{k-1}\| \|y_{k-1}\|}{s_{k-1}^T y_{k-1}} > \varsigma, \\ \theta_k^{\text{BB2}}, & \text{其他,} \end{cases} \quad (2.4)$$

其中 ς 是一个大于 0 接近于 1 的数. 有实验表明这种自适应选取步长的策略可以提高算法的实际性能^[10]. 关于 BB 方法的更多应用可参见文献 [11–14].

此外, 为保证 BB 类方法具有全局收敛性, 在这篇文章中我们使用了与文献 [6] 中不一样的线搜索技术, 非单调线搜索. 非单调方法不要求函数值呈单调下降趋势, 它允许在迭代过程中函数值出现偶尔上升的情形. 这样使得搜索方向与步长因子的选取更具有弹性. 非单调算法能有效地避免单调算法在一个狭窄的平谷区域内缓慢收敛的现象.

为了进一步阐明 IOWL-QN 方法, 我们先简略地回顾一下非单调线搜索技术的框架. 最早的非单调线搜索是由 Grippo 等人在文献 [15] 中提出来的. 他们采用的非单调线搜索为给定常数 $\delta, \sigma \in (0, 1)$, $0 < \tau_1 < \tau_2$, 及非负整数 M , 进一步假设在第 k 步的试探步长 $\tilde{\gamma}_k \in (\tau_1, \tau_2)$ 已经给定. 目标是寻求最小的非负整数 h_k 使得步长因子 $\gamma_k = \tilde{\gamma}_k \sigma^{h_k}$ 满足

$$F(x_k + \gamma_k d_k) \leq \max_{0 \leq j \leq m_k} F(x_{k-j}) + \delta \gamma_k \nabla F(x_k)^T d_k, \quad (2.5)$$

这里 d_k 为搜索方向, $m_0 = 0$, $0 \leq m_k \leq \min\{m_{k-1} + 1, M\}$. 如果 $m_k \equiv 0$, 上述非单调线搜索则变为标准的 Armijo 线搜索. 大量的数值实验表明非单调线搜索算法要比单调线搜索算法更有效.

3 算法及收敛性分析

基于上面的讨论, 我们现在正式给出 IOWL-QN 方法求解稀疏 logistic 回归问题 (1.2).

3.1 IOWL-QN

结合 (1.7) 和 (2.4), 我们可以得到如下的搜索方向:

$$\hat{d}_k = -\tilde{\theta}_k \tilde{\nabla} J(\mathbf{w}_k), \quad (3.1)$$

其中

$$\tilde{\theta}_k = \begin{cases} \tilde{\theta}_k^1, & \text{如果 } \frac{\|\tilde{s}_{k-1}\| \|\tilde{y}_{k-1}\|}{\tilde{s}_{k-1}^T \tilde{y}_{k-1}} > \tilde{\varsigma}, \\ \tilde{\theta}_k^2, & \text{其他,} \end{cases}$$

$\tilde{\theta}_k^1 = \frac{\tilde{s}_{k-1}^T \tilde{s}_{k-1}}{\tilde{s}_{k-1}^T \tilde{y}_{k-1}}$, $\tilde{\theta}_k^2 = \frac{\tilde{s}_{k-1}^T \tilde{y}_{k-1}}{\tilde{y}_{k-1}^T \tilde{y}_{k-1}}$, $\tilde{s}_{k-1} = \mathbf{w}_k - \mathbf{w}_{k-1}$, $\tilde{y}_{k-1} = \tilde{\nabla} J(\mathbf{w}_k) - \tilde{\nabla} J(\mathbf{w}_{k-1})$, $\tilde{\varsigma}$ 是一个大于 0 接近于 1 的数. 于是, 符号投影算子和象限投影算子为

$$\tilde{d}_k^j = \mathbb{P}_S(\tilde{d}) \triangleq \begin{cases} \hat{d}_k^j, & \text{如果 } \hat{d}_k^j (-\tilde{\nabla}_j J(\mathbf{w}_k)) > 0, \\ 0, & \text{其他,} \end{cases} \quad (3.2)$$

$$\mathbf{w}_{k+1}^j = \mathbb{P}_O(\mathbf{w} + \alpha \tilde{d}) \triangleq \begin{cases} \mathbf{w}_k^j + \alpha \tilde{d}_k^j, & \text{如果 } \mathbf{w}_k^j (\mathbf{w}_k^j + \alpha \tilde{d}_k^j) > 0, \\ 0, & \text{其他.} \end{cases} \quad (3.3)$$

如果 (3.1) 中的 $\tilde{\theta}_k > 0$, 那么得到的搜索方向自然是下降的. 然而, $\tilde{\theta}_k > 0$ 这个条件不可能一直满足, 就是说下降性质不再得到保证. 为了克服这样的缺陷, 我们假设存在充分大的 $\theta_{\max} > 0$ 和充分小的 $\theta_{\min} > 0$, $\tilde{\theta}_k$ 满足

$$\tilde{\theta}_k = \min(\theta_{\max}, \max(\tilde{\theta}_k, \theta_{\min})).$$

这个策略可以保证 $\tilde{\theta}_k$ 有界, 并且每一次迭代所生成的搜索方向 $\tilde{\mathbf{d}}_k$ 都是下降的.

下面, 我们根据以上分析来描述算法 (IOWL-QN) 的流程, 见算法 2. 主要是引入了自适应近似目标函数的 Hesse 阵选取策略和非单调线搜索技术, 以改善算法的性态和保证算法的总体收敛性.

算法 2 IOWL-QN

输入: 启动点 $\mathbf{w}_0$, 初始的拟 Newton 向量 S 和 Y , 权衡因子 $\lambda, M, \zeta, \eta \in (0, 1), \xi \in (0, 1), 0 < \theta_{\min} < \theta_{\max}$, 搜索方向计算函数 $\mathbf{dir}(k, g, S, Y)$. $k \leftarrow 0$.

$$J_k = F(\mathbf{w}_0) + \lambda \|\mathbf{w}_0\|_1$$

$$g_k = \tilde{\nabla} J(\mathbf{w}_0)$$

While “不收敛”, Do

$$\tilde{\mathbf{d}}_k \leftarrow \mathbf{dir}(k, g, S, Y)$$

$$\alpha = \min\{1, 1/\|g_k\|_1\}$$

$$\mathbf{w}_{k+1} \leftarrow \mathbb{P}_O(\mathbf{w}_k + \alpha \tilde{\mathbf{d}}_k)$$

$$J_{k+1} = F(\mathbf{w}_{k+1}) + \sum_{j=1}^n \lambda^j |w_{k+1}^j|$$

$$g_{k+1} = \tilde{\nabla} J(\mathbf{w}_{k+1})$$

$$\text{while } J_{k+1} > \max_{0 \leq j \leq m_k} J(\mathbf{w}_{k-j}) + \eta g_k^T (\mathbf{w}_k - \mathbf{w}_{k-1}), \text{ do}$$

$$\alpha \leftarrow \xi \alpha$$

$$\mathbf{w}_{k+1} \leftarrow \mathbb{P}_O(\mathbf{w}_k + \alpha \tilde{\mathbf{d}}_k)$$

$$J_{k+1} = F(\mathbf{w}_{k+1}) + \sum_{j=1}^n \lambda^j |w_{k+1}^j|$$

$$g_{k+1} = \tilde{\nabla} J(\mathbf{w}_{k+1})$$

end do

$$s_k = \mathbf{w}_{k+1} - \mathbf{w}_k$$

$$y_k = g_{k+1} - g_k$$

$$k \leftarrow k + 1$$

End Do

搜索方向计算函数: $\mathbf{dir}(k, g, S, Y)$

输入: 迭代步 k , 伪梯度 g , 拟 Newton 向量 S 和 Y .

If $k = 0$, then

$$\tilde{\mathbf{d}}_k = -g_k$$

else

$$\tilde{\mathbf{d}}_k = -\tilde{\theta}_k g_k$$

$$\tilde{\mathbf{d}}_k = \mathbb{P}_S(\tilde{\mathbf{d}}_k)$$

End If

3.2 收敛性分析

接下来给出本文方法的收敛性定理.

定理 1 假设水平集 $\mathcal{L} \triangleq \{\mathbf{w} \mid J(\mathbf{w}) < J(\mathbf{w}_0)\}$ 有界. $\{\mathbf{w}_k\}$ 是由算法 2 产生的迭代序列, 那么或者存在 h , 使得 $\tilde{\nabla} J(\mathbf{w}_h) = 0$, 或者有下列的性质成立,

- (1) $\lim_{k \rightarrow \infty} \|\tilde{\nabla} J(\mathbf{w}_k)\| = 0$;
- (2) 序列 $\{\mathbf{w}_k\}$ 的聚点都不是函数 J 的局部最大值点;
- (3) 如果 $\mathcal{L}$ 中的稳定点的个数有限, 那么序列 $\{\mathbf{w}_k\}$ 收敛.

证明 由 $\tilde{\mathbf{d}}_k$ 的定义, 可知存在常数 $c_1 > 0$ 和 $c_2 > 0$, 使得 $\tilde{\mathbf{d}}_k$ 满足 $\tilde{\nabla} J(\mathbf{w}_k)^T \tilde{\mathbf{d}}_k \leq -c_1 \|\tilde{\nabla} J(\mathbf{w}_k)\|^2$ 和 $\|\tilde{\mathbf{d}}_k\| \leq c_2 \|\tilde{\nabla} J(\mathbf{w}_k)\|$. 由算法 2, 我们有

$$\tilde{\nabla} J(\mathbf{w}_k)^T \tilde{\mathbf{d}}_k = \tilde{\nabla} J(\mathbf{w}_k)^T (-\tilde{\theta}_k \tilde{\nabla} J(\mathbf{w}_k)) = -\tilde{\theta}_k \|\tilde{\nabla} J(\mathbf{w}_k)\|^2 \leq -\theta_{\min} \|\tilde{\nabla} J(\mathbf{w}_k)\|^2$$

和

$$\|\tilde{\mathbf{d}}_k\| = \|\tilde{\theta}_k \tilde{\nabla} J(\mathbf{w}_k)\| \leq \|\tilde{\theta}_k\| \|\tilde{\nabla} J(\mathbf{w}_k)\| \leq \max(\|\theta_{\min}\|, \|\theta_{\max}\|) \|\tilde{\nabla} J(\mathbf{w}_k)\|.$$

也就是说, 我们可以令 $c_1 = \theta_{\min}$, $c_2 = \max(\|\theta_{\min}\|, \|\theta_{\max}\|)$. 从而, 对于算法 2 来讲, 文献 [15] 中的所有假设均已成立. 进一步, 联立文献 [15] 中的定理可知定理 1 成立. 证毕. $\square$

4 数值实验

本节为了考察新算法 IOWL-QN 的数值表现, 我们在合成数据和标准数据集上分别进行数值实验, 实验在 Matlab 2010b 上编程实现. 同时, 我们将新算法与文献 [6] 中的算法 OWL-QN 进行比较. 所有的程序均在 Windows 7 系统下, 配置为 Pentium (R) Dual-Core CPU E5200 @ 2.50 GHz, 内存 5 GB 的台式电脑上操作运行的. OWL-QN 的 Matlab 代码可以从网站 (<http://www.di.ens.fr/mschmidt/Software/L1General.html>) 上获得, 有兴趣的读者还可以参见文献 [16]. 对于 OWL-QN 方法, 其参数均设为默认值. 而对于 IOWL-QN, 我们将参数设置为 $S = Y = \mathbf{0}$, $M = 10$, $\eta = 10^{-2}$, $\xi = 0.5$, $\zeta = 0.9$. 启动点 $\mathbf{w}_0 = \mathbf{0}$, 算法迭代的停止条件为 $\tilde{\nabla} J(\mathbf{w}_k)$ 的无穷范数小于某个预先设定的停机精度 tol , 即 $\|\tilde{\nabla} J(\mathbf{w}_k)\|_{\infty} < \text{tol}$. 在数据结果中, 我们用“CT”表示算法停止时所消耗的时间 (以秒为单位), 用“NI”表示算法停止时所需的迭代次数.

4.1 合成数据

我们首先考虑在合成数据上对算法的性能进行测试. 数据的构造方式为

$$\begin{aligned} \mathbf{x} &\leftarrow \text{randn}(m, n), \quad \mathbf{w} \leftarrow \text{randn}(n, 1) < \text{sparcoef}, \quad y \leftarrow \text{sign}(\mathbf{x} * \mathbf{w}), \\ \mathbf{x} &\leftarrow \mathbf{x}^T, \quad y \leftarrow y^T, \quad \lambda_a \leftarrow \text{rand}(1, n)^T, \quad \lambda_b \leftarrow \text{sum}(\lambda_a), \quad \boldsymbol{\lambda} \leftarrow \frac{\lambda_a}{\lambda_b}, \end{aligned}$$

其中 $\text{sparcoef} \geq 0$ 控制着向量 $\mathbf{w}$ 的稀疏程度. 实验中, 我们设定停机精度为 $\text{tol} = 10^{-4}$, 并选取 $m = 200, 400, 800$, $n = m/2$, 分别用两种方法进行测试. 相应的数值结果见表 1 至表 4.

表 1 $\text{sparcoef} = 0$

测试	$m = 200, n = 100$		$m = 400, n = 200$		$m = 800, n = 400$	
方法	CT	NI	CT	NI	CT	NI
OWL-QN	0.423	202	0.784	176	4.311	261
IOWL-QN	0.272	161	0.603	172	3.065	213

表 2 $\text{sparcoef} = 0.1$

测试	$m = 200, n = 100$		$m = 400, n = 200$		$m = 800, n = 400$	
方法	CT	NI	CT	NI	CT	NI
OWL-QN	0.245	120	1.763	385	3.131	194
IOWL-QN	0.205	112	0.683	213	2.659	185

表 3 $\text{sparcoef} = 0.2$

测试	$m = 200, n = 100$		$m = 400, n = 200$		$m = 800, n = 400$	
方法	CT	NI	CT	NI	CT	NI
OWL-QN	0.248	145	0.877	194	3.931	220
IOWL-QN	0.249	153	0.668	201	2.829	208

表 4 $\text{sparcoef} = 0.3$

测试	$m = 200, n = 100$		$m = 400, n = 200$		$m = 800, n = 400$	
方法	CT	NI	CT	NI	CT	NI
OWL-QN	0.654	325	1.366	298	3.273	200
IOWL-QN	0.284	177	0.718	208	2.819	197

4.2 标准数据集

这一节, 我们用 Libsvm 的三个标准数据集 (<http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>) 对算法进行测试. 表 5 介绍了三个数据集各自的样本总数和特征数的情形. 其中数据集 “a9a” 包含两部分: 一部分是训练集, 另一部分是测试集. 首先, 我们在前两个数据集上进行数值实验, 设定停机精度为 $\text{tol} = 10^{-4}$. 表 6 至表 8 给出了权衡因子 λ 在不同取值情形下的实验结果.

由于数据集 “a9a” 包含训练和测试两部分, 故我们先利用 “a9a” (训练集) 求得一个权向量, 然后在 “a9a” (测试集) 上进行分类测试. 停机精度设定为 $\text{tol} = 10^{-3}$. 表 9 报告了两种算法在 “a9a” (训练集) 上所用的计算时间和迭代次数, 还给出了在 “a9a” (测试集) 上的分类误差 Err. 图 2 展示了两个

表 5 标准数据集

标准数据集	样本总数	样本特征数
australian_scale	690	14
sonar_scale	208	60
a9a (训练集)	32561	123
a9a (测试集)	16281	123

表 6 $\lambda = 0.1, \text{tol} = 10^{-4}$

测试	australian_scale		sonar_scale	
方法	CT	NI	CT	NI
OWL-QN	0.179	99	1.961	>1000
IOWL-QN	0.633	155	1.749	673

表 7 $\lambda = 0.5, \text{tol} = 10^{-4}$

测试 方法	australian_scale		sonar_scale	
	CT	NI	CT	NI
OWL-QN	0.204	125	1.632	>1000
IOWL-QN	0.632	138	1.437	471

表 8 $\lambda = 1.5, \text{tol} = 10^{-4}$

测试 方法	australian_scale		sonar_scale	
	CT	NI	CT	NI
OWL-QN	0.332	89	0.799	476
IOWL-QN	0.239	133	0.404	308

表 9 $\text{tol} = 10^{-3}$

测试 方法	$\lambda = 10$			$\lambda = 20$			$\lambda = 30$			$\lambda = 40$		
	CT	NI	Err	CT	NI	Err	CT	NI	Err	CT	NI	Err
OWL-QN	63.836	493	0.149	34.727	279	0.148	27.827	227	0.150	23.293	193	0.151
IOWL-QN	35.871	299	0.149	23.381	199	0.148	22.494	196	0.150	20.199	178	0.151

图 2 (a) 当 $\lambda = 20$ 时, OWL-QN 和 IOWL-QN 在 “a9a” (训练集) 上得到的权向量的差值示意图; (b) 当 $\lambda = 40$ 时, OWL-QN 和 IOWL-QN 在 “a9a” (训练集) 上得到的权向量的差值示意图. 横轴表示特征数, 纵轴表示两个权向量的差值

算法得到的权向量的差值, 可以看出两者之间的差值是很小的. 图 3 展示了 $\|\tilde{\nabla} J(\mathbf{w}_k)\|_\infty$ 的非单调变化趋势, 可以看出非单调线搜索技术比单调线搜索技术更具优越性. 需要说明的是, 由表 9 可以看到, 随着 λ 的增大, 算法所消耗的时间和所需的迭代次数减少, 但分类误差略有偏大.

图 3 当 $\lambda = 20$ 时, 在 “a9a” (训练集) 上, 问题 (1.2) 中目标函数 J 的梯度的无穷范数的变化趋势. 横轴迭代次数 k , 纵轴表示 $\|\tilde{\nabla} J(w_k)\|_\infty$

在这篇文章中, 我们主要关心的是算法在求解稀疏 logistic 回归问题 (1.2) 时的速度. 从数值表现来看, 新算法 IOWL-QN 是可行的、有效的, 在一定程度上是可以和 OWL-QN 方法相媲美的.

5 结论

针对稀疏 logistic 回归问题, 本文提出了一种改进的基于分象限学习的拟 Newton 算法. 在引入非单调线搜索的前提下, 分析了新算法的全局收敛性. 数值比较结果进一步验证了新算法的有效性.

致谢 感谢编委会和审稿人提出的宝贵意见.

参考文献

- 1 Bishop C. Pattern Recognition and Machine Learning. New York: Springer, 2006
- 2 Yuan G X, Chang K W, Hsieh C J, et al. A comparison of optimization methods and software for large-scale L_1 -regularized linear classification. J Mach Learn Res, 2010, 11: 3183–3234
- 3 Koh K, Kim S J, Boyd S. An interior-point method for large-scale l_1 -regularized logistic regression. J Mach Learn Res, 2007, 8: 1519–1555
- 4 Schmidt M. Graphical model structure learning with l_1 -regularization. Dissertation for Ph.D. degree. Vancouver: The University of British Columbia, 2010
- 5 Manninen T, Huttunen H, Ruusuvuori P, et al. Leukemia prediction using sparse logistic regression. PLoS ONE, 2013, 8: e72932
- 6 Andrew G, Gao J. Scalable training of L_1 -regularized log-linear models. In: Proceedings of the 24th International Conference on Machine Learning. Corvallis: ACM, 2007, 33–40
- 7 Liu D C, Nocedal J. On the limited memory BFGS method for large scale optimization. Math Program, 1989, 45: 503–528
- 8 Luenberger D, Ye Y Y. Linear and Nonlinear Programming. New York: Springer, 2007
- 9 Barzilai J, Borwein J M. Two point step size gradient methods. IMA J Numer Anal, 1988, 8: 141–148
- 10 Zhou B, Gao L, Dai Y H. Gradient methods with adaptive step-sizes. Comput Optim Appl, 2006, 35: 69–86
- 11 Yuan Y X. A new stepsize for the steepest descent method. J Comput Math, 2006, 24: 149–156
- 12 Dai Y H. A new analysis on the Barzilai-Borwein gradient method. J Oper Res Soc China, 2013, 1: 187–198
- 13 Han D R, He H J, Yang H, et al. A customized Douglas-Rachford splitting algorithm for separable convex minimization with linear constraints. Numer Math, 2014, 127: 167–200

- 14 Yu G H, Xue W, Zhou Y. A nonmonotone adaptive projected gradient method for primal-dual total variation image restoration. *Signal Process*, 2014, 103: 242–249
- 15 Grippo L, Lampariello F, Lucidi S. A nonmonotone line search technique for Newton's method. *SIAM J Numer Anal*, 1986, 23: 707–716
- 16 Schmidt M, Fung G, Rosales R. Optimization methods for l_1 -regularization. UBC Technical Report TR-2009-19. Vancouver: University of British Columbia, 2009

An improved OWL-QN method for solving sparse logistic regression problems

XUE Wei & ZHANG WenSheng

Abstract Sparse logistic regression, is a class of important problems in machine learning, which has wide applications in cybernetics, management sciences and internet, etc. This paper proposes an Improved Orthant-Wise Limited-memory Quasi-Newton (IOWL-QN) method for sparse logistic regression problems. The proposed method adopts the well-known BB stepsize strategy to approximate the Hessian of the goal objective function, and constructs the quasi-Newton vectors based on the gradient of the goal objective function. Under mild conditions, global convergence of the proposed method is established. Numerical results are reported to illustrate that the proposed method is feasible and efficient.

Keywords unconstrained optimization, logistic regression, l_1 regularization, quasi-Newton method, non-monotone line search, Barzilai-Borwein stepsize

MSC(2010) 65K05, 93B40

doi: 10.1360/012015-42