ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 30,No. 1,Jan. 2015,pp. 35－46
DOI:10.16337/j.1004-9037.2015.01.003

http://sjcj.nuaa.edu.cn
E-mail:sjcj@nuaa.edu.cn
Tel/Fax:＋86-025-84892742

Lasso 问题的最新算法研究

刘　柳　陶大程

(悉尼科技大学工程与信息技术学院量子计算与智能系统研究中心,悉尼,澳大利亚,2007)

摘　要:随着大规模数据的增加,解决 Lasso 问题成为一个新的热点,以往的方法很难满足大数据背景下的时间和效率问题。为了解决大规模数据及高维数据而带来的计算和储存的困难,本文从三个方面分析最新的算法,即一阶方法、随机方法及并行和分布计算。本文介绍和分析了解决最小收缩和选择算子(Least absolute shrinkage and selection operator, Lasso)问题的最新算法:梯度下降方法、交替方向乘子法(Alternating direction method of multipliers, ADMM)和坐标下降方法。其中梯度下降结合一阶方法和 Nesterov 的加速和光滑技术;交替方向乘子方法将随机方法融入在最新的算法中;坐标下降方法利用其坐标系的特点结合一阶方法、随机方法和并行和分布计算,本文分别从原始目标函数和对偶目标函数的角度对算法进行分析和研究。

关键词:Lasso 问题;一阶方法;随机方法;交替方向乘子法;坐标下降

中图分类号:TP181　　**文献标志码**:A

Review on Recent Method of Solving Lasso Problem

Liu Liu, Dacheng Tao

(Quantum Computation & Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia, 2007)

Abstract: With the increase of big data, solving Lasso problem becomes top research field. Past methods could not satisfy the time and efficient problem under big data situation. In order to deal with difficulty of computation and storage bringing from huge-scale and high-dimension data, this paper analyze the recent Lasso algorithm from three aspects: one-order method, random, and parallel and distributed computation, which play an important roles in dealing with huge-scale data problem. Based on those three aspects, this paper introduces and analyzes the novel algorithms: gradient descent method, Alternating Direction method of multipliers (ADMM), and coordinate descent method. Gradient descent method combine one-order method and Nesterov's accelerate and smoothing method; ADMM put the random algorithm into the recent research; Coordinate descent make use of the character of coordinate system incorporation one-order method, random, and parallel and distributed computation. Moreover, this paper makes a deep analysis and research from primal and dual objective function.

Key words: Lasso problem; one-order method; random method; ADMM; coordinate descent method

收稿日期:2014-10-10;**修订日期**:2014-12-30

引 言

随着科技的进步,收集数据的技术也有了很大的发展。因此如何有效地从数据中挖掘出有用的信息也越来越受到人们的关注。一般采用基于机器学习和统计的方法来解决大规模数据,这种趋势称为“大数据”。它在很多领域中起着非常重要的作用,如人工智能、网络应用、计算生物、医学、金融和市场等。虽然应用在不同的方面,但是大数据问题却又几个共同的特点:(1)数据量非常大,包含几百万个或上亿个训练样本;(2)数据的维度很高,通常会详细记录每一个样本的的信息;(3)大规模的数据通常以分布式的方式储存或收集。在机器学习的算法中,对最新算法的要求既能够解决数据的复杂性又能够采用平行或分布式的方法处理大数据。

Lasso 问题是 1996 年 Tibshirani[1] 使用 L1 惩罚因子解决含有独立高斯分布的线性回归问题,Lasso 问题包含平方误差的求和以及 L1 正则化项。虽然 Lasso 问题出现的比较晚,然而正则化项早在 1943 年 Tikhonov 就提出用来逼近不可解的整数等式问题。通常情况下,在参数中增加约束化项,用来求解评估问题,如最大似然函数。正则化项用来减小变量从而避免过拟合,得到的正则化项的解更稳定,Bickel[2] 给出了正则化项在统计理论中详细的分析。

Lasso 问题在解决稀疏规划中起着非常重要的作用。它可以解释为寻找最小二乘或线性回归问题的稀疏问题,即采用若干个非零的变量。Lasso 问题在信号处理领域中同样起着重要的作用,包括解决稀疏信号修复[3]、稀疏回归[4]、稀疏图回归[5],稀疏逆协方差 [6] 和稀疏字典的学习[7]。生物数据的分析[8],比如选取大数据中的一小部分来预测结果。Lasso 问题同样可以应用在视频中,Geng[9] 采用并行 Lasso 的方法解决大规模视频概念检测,Zhu[5] 使用 Group Lasso 解决视频标签以及 Zhou[10] 用稀疏离群值分割移动目标。在图像处理中,Afonso[11] 使用 Lasso 方法解决图像修复、图像去噪[12] 和去模糊[13],其中正则化项是图像的梯度范数、网页图像排序[14] 及图像质量评估[15],利用稀疏编码处理图像标签[3] 问题。在遥感领域中,Lasso 问题用来解决稀疏解混 [16-18],从而可以得到每个端元都含有哪些物质,在航天、农业领域,对研究物质成分问题起着非常重要的作用。

解决 Lasso 问题的方法有很多,其中 Efron[19] 2004 年提出最小角回归,它能够有效地解决潜在的 Lasso 优化问题,同时在统计和机器学习领域,它为加速 Lasso 算法提供了一个重要的工具。在解决大规模数据问题中,梯度下降方法是最简单的方法。Nesterov[20,21] 提出最优的梯度下降方法和光滑模型,在 Lasso 问题中被广泛的应用,如快速的交替方向优化方法[22],锥模型优化方法[23] 以及快速迭代收缩阈值方法[24]。随着数据量的增加,ADMM 能够解决大数据问题,并且已经成功地应用在很多领域,包括解决 Lasso 问题,Beck[24] 结合 Nesterov 的加速方法,提出了快速的方法,其中 Tom[25] 提出了快速的 ADMM 方法。最近的研究者 Taiji[26,27] 和 Hua[28] 结合大数据的特点采用随机的方法,在原有算法的基础上,提高了算法的收敛率,

在大数据背景下随机坐标下降方法成为一个新的热点,而随机优化方法在解决高维数据和大规模样本有至关重要的作用。Nesterov[29] 和 Ji[30] 分别给出了随机坐标下降方法的理论分析和证明。越来越多的基于随机优化方法出现在 Lasso 问题,包括原始随机坐标下降方法[29,31],对偶随机坐标下降方法[32,33] 和原始随机对偶坐标下降方法[34,35],而且基本的随机下降坐标方法结合了一阶最优方法[36]、随机方法和并行分布系统,这三个方面在解决大规模数据中起着非常重要的作用,弥补了一般算法在大数据背景下的不足,其中 Nesterov 加速的一阶最优方法,只需要一阶梯度信息,从而运算复杂度小、速度快;随机的方法主要考虑大数据的本质,因为最终决定最优解的信息并不需要全部的数据,如果全部的数据都用来计算不仅需要更多储存空间,而且会消耗更多的时间,从而影响算法的性能。最近 Avron[37-39] 介绍了在随机梯度和随机坐标下降方法下的多核运算,Ji[40-42] 在 Kaczmarz 算法和坐标下降算法中提出异步并行计算提高收敛率。

在解决 Lasso 问题的过程中,很多实验室公开了软件包和代码,以供大家学习和研究,其中包含凸优化模型框架 CVX[43],它可以用来解决所有的凸优化问题;压缩感知包 l1-MAGIC[44],它将问题转化为两阶凸规划(Second-order cone program, SOCPs),一阶锥模型(Templates for first-order conic solvers, TFOCS)[23],它主要是将问题转化为标准的锥模型,并利用一阶优化方法;共轭梯度迭代方法(Conjugate gradient iterative shrinkage/thresholding,CGIST)[45],它主要结合向前向后分裂方法和加速步长方法。

1 Lasso 问题

在机器学习的优化算法中,一般的凸优化问题可以表示为

$$\min_{x\in\mathbf{R}^d}\left\{P(\boldsymbol{x})=\sum_{i=1}^{n}\varphi_i(\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x})+\lambda g(\boldsymbol{x})\right\} \tag{1}$$

其对偶函数为

$$\min_{y\in\mathbf{R}}\left\{D(\boldsymbol{y})=\sum_{i=1}^{n}-\varphi_i^{*}(-y_i)-\lambda g^{*}\left(\frac{1}{\lambda}\sum_{i=1}^{n}\boldsymbol{a}_i y_i\right)\right\} \tag{2}$$

式中:$\boldsymbol{a}_1,\cdots,\boldsymbol{a}_n\in\mathbf{R}^d$ 为;n 个数据样本向量;$\varphi_1,\cdots,\varphi_n$ 为损失函数;$\varphi_1^{*},\ldots,\varphi_n^{*}$ 为 φ 的共轭函数;$g(\cdot)$为正则化函数;$g^{*}(\cdot)$为 g 的共轭函数;$\boldsymbol{x}$ 为原始变量;y 为对偶变量;$\lambda\geqslant 0$ 为正则化参数。

每一个样本数据 $\boldsymbol{a}_i$ 对应着一个独立的变量 $\boldsymbol{b}_i$,当 $\varphi_i(\boldsymbol{a}_i^{\mathrm{T}}x)=\frac{1}{2}(\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}-\boldsymbol{b}_i)^2$,$g(\boldsymbol{x})=\|\boldsymbol{x}\|_1$,即得到 l_1 约束线性回归的特殊形式,称为 Lasso 问题[1],即

$$\min\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{x}\|_1 \tag{3}$$

式中:变量 $\boldsymbol{x}\in\mathbf{R}^n$;矩阵 $\boldsymbol{A}\in\mathbf{R}^{m\times n}$;$\boldsymbol{b}\in\mathbf{R}^m$;$\lambda>0$ 是尺度约束参数。Lasso 问题可以解释为寻找最小二乘或线性回归问题的稀疏解。

1.1 一般的 Lasso 问题

$$\min\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{F}\boldsymbol{x}\|_1 \tag{4}$$

其中 $\boldsymbol{F}$ 是任意的线性变换,一个特殊的例子是当 $\boldsymbol{F}$ 为差异矩阵时,有

$$F_{ij}=\begin{cases}1 & j=i+1\\ -1 & j=i\\ 0 & \text{其他}\end{cases} \tag{5}$$

这个问题转变为 TV 降噪问题[46],在信号处理中有广泛的应用。当 $\boldsymbol{A}=\boldsymbol{I},\boldsymbol{F}$ 是二次差分矩阵,问题变为 $L1$ 趋势滤波[47],用来分析在不同学科中的时间序列数据。更新 $\boldsymbol{x}$ 的迭代过程中,矩阵 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+\boldsymbol{F}^{T}\boldsymbol{F}$ 是五对角矩阵,只需运行 $O(n)$的浮点运算。

1.2 组 Lasso 问题

$$\min\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2+\lambda\sum_{i=1}^{N}\|\boldsymbol{x}_i\|_2$$

式中 $\boldsymbol{x}=(x_1,\cdots,x_N)$,$\boldsymbol{x}_i\in\mathbf{R}^{n_i}$,每一个正则化约束项都是分开的,但并不是完全的分开,这种 $L1$ 范数约束的扩展问题称之为组 Lasso 问题[48,49],或者叫做 Sum of norms regularization[50],解决组 Lasso 问题一般采用分裂的方法,并对每一个子问题进行 Lasso 问题的处理。

随着大规模数据的增加,以往解决 Lasso 问题的方法很难满足人们对时间和效率的要求。当数据的维度越来越大,函数值或梯度值计算就会变得越来越困难,尤其是当数据的空间维度大于存储空间时,一般的梯度方法将不再适用。如果储存空间不影响梯度的运算,那么大数据的运算可能会消耗更多

的时间,从而在实际应用中会受到影响,比如图像处理过程。除了数据的大小,并不是所有的数据都要用来描述机器学习的相关问题,其中一部分数据就可以解决优化问题,而不是等到全部的数据都处理完。所以解决这些问题的方法不仅要考虑算法的复杂度,而且还应从实际问题出发,考虑如何能够更快地解决大规模数据带来的困难。

2 Lasso 问题解决方法

大规模数据下的 Lasso 问题主要面临着计算复杂度和时间复杂度问题,而解决这些问题主要从三个方面考虑[51]:(1)一阶方法。一阶方法使用目标函数的梯度信息,虽然获得较低或中等精确的数值结果,但是相比一些复杂的运算过程,一阶方法消耗更短的时间。根据邻近映射原则,一阶方法能够解决非光滑函数问题,并且适合分布和并行计算。(2)随机性。随机问题在其他的逼近问题中尤为突出,因为它能够控制期望值,从而增加一阶方法的可扩展性。关键的问题是随机更新部分变量信息,而不是全部的变量,进而并不需要全部的梯度信息;简单的统计评估就能有很好的计算结果;(3)并行和梯度计算:一阶梯度方法能够用在并行计算和分布优化中,同时还可以从集中通信的并行计算扩展到分散通信的异步算法。这 3 个方面在大数据中起着重要的作用,比如随机一阶方法比确定策略的方法要快,它能够忽略一小部分信息而获得高概率的最优目标函数结果[34]。

2.1 梯度下降方法

梯度下降的方法是基于一阶方法,利用局部梯度信息和迭代公式

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^{k}-a_k\nabla f(\boldsymbol{x}^{k}) \tag{7}$$

当目标函数为光滑函数时,可以使用更快的加速方法,如牛顿方法,但是它需要目标函数更多的信息,比如二阶求导信息,从而消耗更多的时间,而这些信息并不容易在约束函数和非光滑函数中得到。梯度下降法却弥补了这些缺点。一般的梯度下降方法需要 $O(1/\varepsilon)$ 的迭代次数才能达到 ε 的精确结果。Nesterov[20] 提出了一个改进,增加一个额外的动量步长 $\beta=k/(k+1)$,从而到达到 $O(1/\varepsilon^2)$ 的迭代次数,这种方法称为最优一阶方法。表 1 给出了加速方法在凸函数和强凸函数中的算法复杂度。为了能够得到强凸函数,可以在目标函数中增加约束化项。因为如果函数 $f(\boldsymbol{x})-\mu/2\boldsymbol{x}_2^2$ 是凸函数的,那么是 $f(\boldsymbol{x})$ 强凸函数,所以非光滑函数也可以有强凸函数的性质,在 Lasso 函数中增加一个约束项,即 $g(\boldsymbol{x})=\boldsymbol{x}_1+\mu/2\boldsymbol{x}_2^2$。

表 1 梯度下降方法的算法复杂度比较

Table 1 Complexity of the algorithm comparison of gradient ascent method

算法	凸函数	强凸函数
梯度下降	$O(Ld^2/\varepsilon)$	$O(L/\mu\log(d^2\varepsilon))$
加速梯度下降	$O(\sqrt{Ld^2/\varepsilon})$	$O(\sqrt{L/\mu}log(\mathrm{d}^2\varepsilon))$

2.1.1 邻近梯度方法

邻近算法[52]可以看成是解决非光滑、约束、大规模和分布式问题的工具。邻近算法可以看成是求解一部分凸优化问题,从邻近运算定义可以看出

$$\mathrm{prox}_g(\boldsymbol{y})=\arg\min_{x}\left\{g(\boldsymbol{x})+\frac{1}{2}\parallel\boldsymbol{x}-\boldsymbol{y}\parallel_2^2\right\} \tag{8}$$

函数的邻近运算等价于求解函数的梯度步长 $\mathrm{prox}_g(\boldsymbol{y})=\boldsymbol{y}-\nabla g(\boldsymbol{x})$ 。

Lasso 函数是一般混合目标函数的一种特殊形式,邻近梯度方法可以利用混合函数的性质,获得在光滑梯度方法下的相同的收敛速度(如表 1),邻近梯度方法可以看成是最优梯度方法的扩展,于是原始的 Lasso 问题结合最优梯度方法和邻近梯度方法,得到基于 $\boldsymbol{x}^k$ 展开的目标函数

$$\boldsymbol{x}^{k+1}=\arg\min_{\boldsymbol{y}}\left\{f(\boldsymbol{x}^k)+\nabla f(\boldsymbol{x}^k)^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{x}^k)+\frac{1}{2a_k}\|\boldsymbol{y}-\boldsymbol{x}^k\|_2^2+g(\boldsymbol{y})\right\} \tag{9}$$

应用 Nesterov 的加速方法,得到加速的邻近梯度方法

$$\begin{aligned}\boldsymbol{x}^{k+1}&=\mathrm{prox}_{a_k g}(\boldsymbol{v}^k-a_k\nabla f(\boldsymbol{v}^k))\\ \boldsymbol{v}^{k+1}&=\boldsymbol{x}^{k+1}+\beta_k(\boldsymbol{x}^{k+1}-\boldsymbol{x}^k)\end{aligned} \tag{10}$$

在 Lasso 问题中 $g(x)$ 为 $L1$ 范数,邻近运算可以通过软阈值的方法求解,这种加速的方法称为快速收缩阈值算法(Fast iterative shrinkage/thresholding algorithm,FISTA)[24]。除了在迭代步骤能够减少迭代次数、增加收敛率,Blackford[53-55]采用了并行矩阵-向量相乘的方法来加速运算。

2.1.2 对偶锥方法

对偶锥[23]方法结合对偶、光滑和一阶优化方法,用来解决一般的凸锥优化问题。该方法的优点是能够得到稳定、有效的迭代方法。不用于邻近梯度方法,对偶锥方法在对偶函数中加入一阶优化方法。在原始函数中,这两种方法都增加了一个约束项,使得目标函数为强凸函数,从而达到表 1 中的收敛速度。对偶锥方法可以看成是不断的缩减原始和对偶函数差值,当原始函数值和目标函数值相同时,即达到最优解。对偶锥方法包含 4 个步骤。

(1)确定等价的锥形式

Lasso 问题的锥形式可以表示为

$$\begin{aligned}&\min_{\boldsymbol{x}}\|\boldsymbol{x}\|_1\\ &\text{s.t. }\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|\leqslant\varepsilon\end{aligned} \tag{11}$$

(2)确定对偶形式

原始函数不采用一阶优化方法有几种可能:(1)原函数非光滑;(2)投影在 $\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|\leqslant\varepsilon$ 需要消耗大量的时间,所以将问题转化为对偶问题用于减少运算时间,对偶问题表示为

$$\max_{\boldsymbol{z}}\inf_{\boldsymbol{x}}\{\|\boldsymbol{x}\|_1-\langle\boldsymbol{z},\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\rangle\} \tag{12}$$

(3)应用光滑技术

Lasso 问题中含有 $L1$ 范数,从而应用 Nesterov 的光滑技术,在原目标函数中增加约束项,得到光滑的对偶问题

$$D_\mu(z)=\max_{\boldsymbol{z}}\inf_{\boldsymbol{x}}\left\{\|\boldsymbol{x}\|_1+\frac{1}{2}\mu\|\boldsymbol{x}-\boldsymbol{x}_0\|_2^2-\langle\boldsymbol{z},\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\rangle\right\} \tag{13}$$

(4)使用一阶优化方法

在对偶函数中使用一阶优化方法,类似于邻近梯度方法,同样采用 Nesterov 的加速方法,只不过变量增加了对偶变量。对偶锥方法可以看成是鞍点方法,通过交替迭代优化原始函数和对偶函数,达到共同的最优解。具体的算法过程如下

$$\begin{aligned}\boldsymbol{y}_k&=(1-\theta_k)\boldsymbol{v}_k+\theta_k\boldsymbol{z}_k\\ \boldsymbol{x}_k&=\mathrm{SoftThreshold}(x_0-\mu^{-1}\boldsymbol{A}^*\boldsymbol{y}_k,\mu^{-1})\\ \boldsymbol{z}_{k+1}&=\mathrm{Shrink}(\boldsymbol{z}_k-\theta_k^{-1}t_k(\boldsymbol{y}-\boldsymbol{A}x_k),\theta_k^{-1}t_k\varepsilon)\\ \boldsymbol{v}_{k+1}&=(1-\theta_k)\boldsymbol{v}_k+\theta_k\boldsymbol{z}_{k+1}\end{aligned} \tag{14}$$

式中:Shrink(·)是 l_2 收缩运算,SoftThreshold(·)是软阈值运算。图 1 表示的是模拟数据下,基于对偶锥方法和加速对偶锥方法的迭代结果,其中横坐标是外部迭代次数,纵坐标表示迭代点与最优点的迭代误差 $\|\boldsymbol{x}_k-\boldsymbol{x}^*\|_2/\|\boldsymbol{x}^*\|_2$,从图 1 中可以看出,基于 Nesterov 加速的方法收敛更快。

2.2 交替方向乘子方法

交替方向乘子(Alternating direction method of multipliers,ADMM)方法[22]结合分布式凸优化问题,适合解决大规模数据的问题。ADMM 可以看成是融合对偶分解和增广 Lagrangian 的优点,它跟很

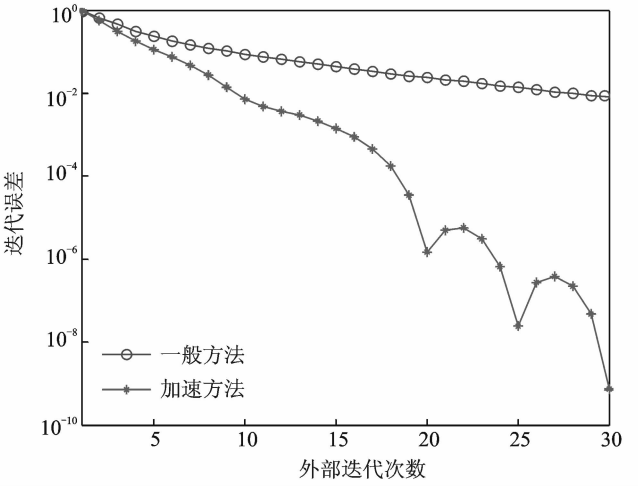

图 1 对偶锥方法和加速的对偶锥方法

Fig. 1 Dual conic and accelerated dual conic method

多算法有相互联系，如 Bregman 迭代方法，Douglas-Rachford 分裂方法及 Dykstra 交替投影方法等。ADMM 方法可以追溯到 1970 年，它在解决大规模分布计算系统和大量的优化问题中起着非常重要的作用。

在 ADMM 形式中，Lasso 问题可以看成

$$\min f(\boldsymbol{x})+g(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{x}-\boldsymbol{z}=0 \tag{15}$$

式中 $f(\boldsymbol{x})=\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2$，$g(\boldsymbol{x})=\lambda\sum_{i=1}^{N}\boldsymbol{x}_{i2}$，得到基于 ADMM 的算法

$$\begin{aligned}\boldsymbol{x}^{k+1}&=(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+\rho\boldsymbol{I})^{-1}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}+\rho(\boldsymbol{z}^k-\boldsymbol{u}^k))\\ \boldsymbol{z}^{k+1}&=S_{\lambda/\rho}(\boldsymbol{x}^{k+1}+\boldsymbol{u}^k)\\ \boldsymbol{u}^{k+1}&=\boldsymbol{u}^k+\boldsymbol{x}^{k+1}-\boldsymbol{z}^k\end{aligned} \tag{16}$$

式中 $\boldsymbol{x}$ 的更新是岭回归过程，解决 Lasso 问题的 ADMM 方法可以解释为交替实施岭回归过程。图 2 表示的是模拟数据下，ADMM 与梯度迭代方法的比较。其中红色曲线表示加速的 ADMM(Fast-ADMM)，

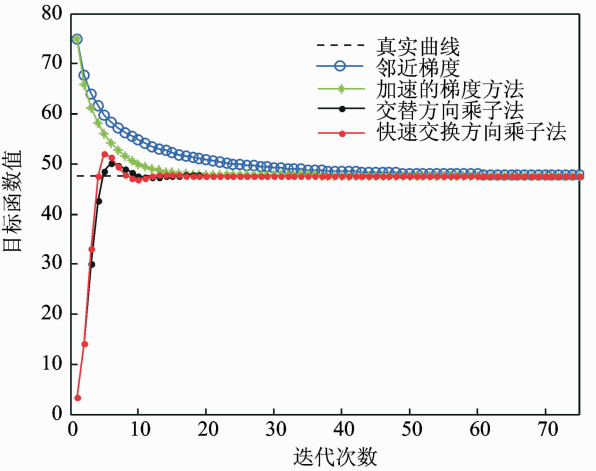

图 2 ADMM 方法和梯度方法

Fig. 2 ADMM and gradient method

黑色曲线表示 ADMM,绿色表示加速的邻近梯度放方法,蓝色表示邻近梯度方法,虚线表示真实值曲线。加速的 ADMM 比没有加速的 ADMM 方法能够更快的接近最优的目标函数值,加速的邻近梯度方法相对于邻近梯度方法能够更快的达到最优的函数值,但是从整体来说,ADMM 方法能更快地收敛最优目标函数值。

一般的 ADMM 算法的收敛速度为 $O(1/k)$,Goldstein[25] 结合 Nesterov[20] 的一阶最小方法能达到 $O(1/k^2)$的全局收敛速度。然而随着数据的快速增加,如自然语言处理、图像识别和生物信息等,随机优化算法已成为在机器学习中重要的方法。Suzuki[28-27] 提出了基于随机优化算法的 ADMM 方法,同样结合 Nesterov 一阶优化方法,应用在一般的凸函数,其收敛速度为 $O(1/\sqrt{k})$,而对于强凸函数能够达到 $O(\log(k)/k)$的收敛速度,Suzuli[27] 结合随机对偶坐标下降方法(Stochastic dual coordinate ascent,SDCA)和 ADMM,由于在每次迭代中需要一个或多个样本,所以 SDCA-ADMM 并不需要非常大的储存空间,这种特点尤其是在面对大数据量时,能够更好地体现出算法的优越性。

2.3 坐标下降方法

在机器学习中,坐标下降方法[56] 在解决大规模优化问题中起着重要的作用,尤其是当样本数 m 非常大时,求全梯度或次梯度需要消耗大量的时间和储存空间,这样求解目标函数的子函数问题就成为研究大规模问题的热点。子函数问题可以看成选取坐标系中的某一个坐标或某一组坐标,保持其他的坐标系的变量不变,在每一次迭代过程,只优化在选取坐标系下的目标函数。

有效的坐标下降方法取决于每一次迭代过程消耗的时间和目标函数的下降量。一个极端的方法是贪婪方法,即选取最大的下降方向。但是贪婪方法面临的问题:(1)要求所有的数据可行;(2)可能会超出计算机的计算能力,比如需要计算全梯度的迭代步长,而不是某个坐标系下的迭代步长。另外两个选择坐标系的方法是周期和随机,一般的选择坐标方法是基于周期循环,循环坐标下降方法的全局和局部收敛性质在文献[57]中给出了详细的分析,它主要分析了在等渗性假设下的光滑凸函数。虽然周期循环是求解连续坐标系的迭代过程,但是并不是在所有坐标系下的求解对目标函数都产生影响。随机选取坐标或坐标块的方法已经成为在大规模数据分析中的研究热点。随机选择坐标系的方法更适合大规模数据,最近的研究表明随机方法确实能够增加收敛率[42,58,59]。在均匀概率下,随机选择坐标系看似等效于周期循环,但是随机坐标系的选择能够避免周期循环下最糟糕的情况,比如某些坐标系下的数据对目标函数并没有作用。除了均匀概率,不同的概率密度函数同样可以增加收敛速度。同时,并行分布式运算也可以用在坐标下降方法中,从而达到更好的运算效率。

Nesterov 加速方法已经成功地应用在全梯度下降方法,将 Nesterov 的加速方法应用在随机坐标下降方法中已经在文献[21,34]中进行了分析。Nesterov[21] 提出了一种加速的随机坐标梯度方法用于解决最小化无约束的光滑函数,Lu 和 Xiao[34] 给出了基于 Nesterov 方法的证明分析。

2.3.1 随机坐标下降方法

随机坐标下降方法[31] 是在某个概率分布下选择坐标系进行迭代的过程,首先将 N 维空间分成 n 个子空间,$\boldsymbol{U}_i \in \mathbf{R}^{N\times N_i}$ 为矩阵的列置换,任何一个 N 维空间的向量 $\boldsymbol{x}$ 可以唯一的表示为 $\boldsymbol{x}=\sum_i \boldsymbol{U}_i x^{\langle i\rangle}$,$x^{\langle i\rangle}=\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{x}\in\mathbf{R}^{N_i}$,当 $N_i=1$ 时,即为单位 $\boldsymbol{e}$ 向量。Lasso 问题可以简单的表示为

$$\min_{x\in\mathbf{R}^n}\{P(\boldsymbol{x})=f(\boldsymbol{x})+\lambda g(\boldsymbol{x})\} \tag{17}$$

每一次迭代过程只需要在随机选择的坐标系下进行,即

$$P(\boldsymbol{x}+\boldsymbol{U}_i t)=f(\boldsymbol{x}+\boldsymbol{U}_i t)+\lambda g(\boldsymbol{x}+\boldsymbol{U}_i t) \tag{18}$$

其中 $f(\boldsymbol{x}+\boldsymbol{U}_i t)$ 的上界可以通过 $f(\boldsymbol{x})$函数的光滑性表示:$f(\boldsymbol{x}+\boldsymbol{U}_i t)\leqslant f(\boldsymbol{x})+\langle\nabla_i f(\boldsymbol{x}),t\rangle+\frac{L_i}{2}\|t$

$\|\ {}^{2}_{(i)}$，L_i 为每个坐标系下的 Lipschitz 常数。每一次的迭代过程需要上一次的迭代结果，得到基本的随机坐标算法

$$\begin{cases} T^{(i)}(\boldsymbol{x}_k) = \arg\min\left\{ \langle \nabla_i f(\boldsymbol{x}_k), t \rangle + \dfrac{L_i}{2} \| t \|^2_{(i)} + g_i(\boldsymbol{x}_k + t), t \in \mathbf{R}^{N_i} \right\} \\ \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{U}_i T^{(i)}(\boldsymbol{x}_k) \end{cases} \tag{19}$$

在 Lasso 问题的目标函数上增加正则化项 $\| \boldsymbol{x} - \boldsymbol{x}_0 \|^2_2$，使得目标函数变为严格凸函数，这种方法[29]应用在基于光滑函数的坐标下降方法中得到更好的迭代结果。表 2 对比了不同算法的复杂度的结果。

表 2 不同算法的复杂度比较(1)

Table 2 Complexity comparison of the different algrithm (Ⅰ)

算法	复杂度
Yun and Tseng[63]	$O(nL(\nabla f)\ \| \boldsymbol{x}^* - \boldsymbol{x}_0 \|^2_2/\varepsilon)$
Sahara and Tewari[57]	$O(nL(\nabla f)\ \| \boldsymbol{x}^* - \boldsymbol{x}_0 \|^2_2/\varepsilon)$
Shalev-Shwartz[59]	$O(n\beta \| \boldsymbol{x}^* - \boldsymbol{x}_0 \|^2_2/\varepsilon)$
Peter Richtarik[31]	$O(n \| \boldsymbol{x}^* - \boldsymbol{x}_0 \|^2_2/\varepsilon)$

2.3.2 邻近随机对偶坐标下降

邻近的随机对偶坐标下降方法(Proximal stochastic dual coordinate ascent, Prox-SDCA)[33]是 Shalev 和 Zhang[32]对随机对偶问题的扩展，允许正则化函数 g 为一般的强凸函数，这样可以用来解决非凸的 $L1$ 正则化函数。在目标函数中增加正则化项 $P(\boldsymbol{x}) + \kappa/2 \| \boldsymbol{x} - \boldsymbol{z} \|^2_2$，其中 $\boldsymbol{z}$ 是上一步的迭代结果，$\kappa: O(1/\gamma n)$，当时 $1/(\lambda\gamma) \gg n$，强正则化项能够使得邻近随机对偶坐标下降的运算时间达到 $O(dn)$。在每次迭代过程中，选择合适的 y 能够达到 $d\sqrt{n/(\lambda\gamma)}$ 的运算时间。表 3 是不同算法的运算时间对比表。

表 3 不同算法的运算时间比较

Table 3 Computing time comparison of the different algorithm

算法	运行时间
SGD[59]	d/ε^2
SCD[31]	dn/ε
FISTA[24]	$dn\sqrt{1/\varepsilon}$
Prox-SDCA[33]	$d(n + \min\{1/\varepsilon, \sqrt{n/\varepsilon}\})$

对偶坐标下降方法的目标是增加对偶目标函数值，随机选取对偶变量的 $\boldsymbol{y}$ 的一个坐标系 e_i，即 y_i 对应着原始问题的第 i 个样本。最大化对偶函数，得到对偶变量的迭代步长

$$\Delta y_i = \arg\min_{y \in \mathbf{R}} \left\{ -\varphi_i^*(-(y_i + \Delta y_i)) - \lambda g^*\left(\lambda^{-1} \sum_{i=1}^{n} a_i y_i + \lambda^{-1} a_i \Delta y_i\right) \right\} \tag{20}$$

为了简化优化问题，对共轭函数 g 做光滑处理，即一阶泰勒展开，从而问题转化为最大化对偶函数的下界问题

$$\Delta y_i = \arg\min_{y \in \mathbf{R}} \left\{ -\varphi_i^*(-(y_i + \Delta y_i)) - \nabla g^*\left(\sum_{i=1}^{n} a_i y_i\right) - (2\lambda)^{-1} a_i \Delta y_i \|^2_D \right\} \tag{21}$$

在 Lasso 问题的正则化项中，$L1$ 正则化项不是强凸正则化项，增加 L2 正则化项，从而定义 L1-L2 正则化项 $g(\boldsymbol{x}) = \lambda \| \boldsymbol{x} \|_1 + \sigma \| \boldsymbol{x} \|_2 \approx \lambda \| \boldsymbol{x} \|_1$($\sigma$ 足够小)。则 $g(\boldsymbol{x})$的共轭函数梯度为

$$\nabla_i g^*(y_i) = \mathrm{sign}(y_i)[|v_i| - \lambda/\sigma]_+ \tag{22}$$

Prox-SDCA 加速方法采用 Nesterov′s 评估序列技术[20,21]用于逼近随机的邻近映射。在每一次 Prox-SDCA 的迭代过程中，改进目标函数 $P(\boldsymbol{x}) + \kappa/2 \| \boldsymbol{x} - \boldsymbol{z}^{(t-1)} \|^2_2$，其中正则化项围绕着向量 $\boldsymbol{z}^{(t-1)} =$

$\boldsymbol{x}^{(t-1)}+\beta(\boldsymbol{x}^{(t-1)}-\boldsymbol{x}^{(t-2)})$,从而得到更快的收敛速度。对于选取坐标系,一般在均匀分布的概率下选取,Zhang[60]提出了在对偶坐标系下根据任意的分布选取对偶变量,并获得了有效的并行和分布式变量方法。

2.3.3 随机原始对偶坐标下降方法

随机原始对偶坐标下降方法(Stochastic primal-dual coordinate,SPDC)[62]是通过交替优化对偶函数和原始函数,其中随机选择对偶变量中的一维空间,由于对偶变量对应着原始目标函数中的样本数。在给出随机原始对偶问题的算法前,先定义原始问题的凸-凹鞍点问题

$$\min_{x\in\mathbf{R}^d}\max_{y\in\mathbf{R}^n}\left\{f(\boldsymbol{x},\boldsymbol{y})=\sum_{i=1}^{n}(y_i(a_i,\boldsymbol{y})-\varphi_i^*(y_i))+\lambda g(\boldsymbol{x})\right\}$$

关于变量 $\boldsymbol{y}$ 通过最大化目标函数 $f(\boldsymbol{x},\boldsymbol{y})$得到,需要消耗 $O(nd)$的计算量,如果 n 很大,需要消耗更多的时间,所以通过随机选取对偶变量 y 的坐标来减少计算量。这样在选取的坐标系下最大化目标函数,然后交替优化原始目标函数。不同于一般的原始对偶问题,根据 Nesterov′s 的加速技巧,在原始和对偶函数中增加了两个正则化项,从而得到更快的收敛速度,Chen[35]介绍了随机原始对偶方法,在原始和对偶目标函数中增加一阶变量并且融入多阶段的 Nesterov 的加速方法,同时不需要采用光滑技术也能达到 Nesterov 光滑后的收敛速度。同时,Zhang[34]提出 Mini-batch 的方法,自动选取多个对偶坐标变量,在并行运算中更新 $\boldsymbol{x}^{(t+1)}$,仅需 $O(d)$的迭代时间,当然处理多个对偶坐标,它的收敛率比一般的 SPDC 方法要快很多,表 4 给出了不同算法的时间复杂度。

表 4 不同算法的复杂度比较(2)

Table 4 Complexity comparison of the different algorithm (Ⅱ)

算法	复杂度
Nesterov[61]	$O((1+\kappa)\log(1/\varepsilon))$
Tseng[62]	$O((n+\sqrt{\kappa})\log(1/\varepsilon))$
Shalev-Shwartz[32]	$O((n+\kappa)\log(1/\varepsilon))$
Zhang and Lin[34]	$O((1+\sqrt{\kappa/n})\log(1/\varepsilon))$

3 结束语

近年来,随着数据量的增加,对解决 Lasso 问题提出了更大的挑战。在解决大规模数据中,必须要考虑计算时间和储存能力,当硬件条件还不能完全满足大数据背景下带来的需求,尤其是面对高维数据,以往的方法很难再满足人们对数据的分析,研究最新的算法就成为最新的热点。一阶方法、随机方法和并行和分布计算在解决大规模数据中起中重要的作用,首先一阶方法计算起来简单,消耗时间短,符合人们对时间效率的要求;再次,随机方法主要从数据的特点来考虑,大规模数据并不代表所有的数据都会对算法产生影响,如果对所有的数据进行处理,那么必将消耗更多的时间和储存空间,随机算法正好弥补了这些缺点;并行和分布计算同样是为了减少时间消耗,加速运算,增加效率。

本文从 3 个典型的算法分析了 Lasso 问题的最新算法:梯度下降方法、交替方向乘子法和坐标下降方法。解决大规模数据往往采用简单的方法更能提高效率。梯度下降是做简单的方法,它结合一阶方法和 Nesterov 的加速和光滑技术,使最终的收敛速度达到 $O(1/k^2)$。ADMM 和坐标下降方法也是在梯度下降方法的基础上进行改进,使得它们能够更适合大规模的数据。同时 ADMM 和坐标下降方法采用随机方法进行迭代,因为在大规模数据中能够减少时间消耗,实验结果也证实了在大规模数据中,基于随机算法能够得到更好的结果。坐标下降方法利用其坐标系的特点结合一阶方法、随机方法和并行和分布计算。坐标下降方法之所以是最近研究的热点,主要是利用了其坐标系的特点,另一个原因是数据样本的增加和数据维度越来越高,为了提高效率,坐标下降方法是可行的方法,在原始目标函数中采

用坐标下降方法是为了避免高维度,而在对偶函数下的坐标下降为了解决大数据样本问题。基于随机的坐标下降方法在未来的研究中仍然起着非常重要的作用,同时在某些概率分布下,对坐标系的随机选择将会是下一个阶段研究的热点问题。

参考文献:

[1] Tibshirani R. Regression shrinkage and selection via the lasso[J]. Journal of the Royal Statistical Society. Series B (Methodological), 1996, 15(1): 267-288.

[2] Bickel P J, Li B, Tsybakov A B, et al. Regularization in statistics[J]. Sociedad de Estadistica e Investigacion Operativa, 2006, 15(2): 271-344.

[3] Fadili M J, Starck J L. Monotone operator splitting for optimization problems in sparse recovery[C]// Image Processing (ICIP), 2009 16th IEEE International Conference on. Cairo, Egypt: [s. n.], 2009: 1461-1464.

[4] Liu J, Yuan L, Ye J. Dictionary Lasso: Guaranteed sparse recovery under linear transformation[C]// Proceedings of Machine Learning, 2012 30th IEEE international Conference. Atlanta, USA: [s. n.], 2012: 1-28.

[5] Zhou T, Tao D. Multi-task copula by sparse graph regression[C]// Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: [s. n.], 2014: 771-780.

[6] Yuan X. Alternating direction methods for sparse covariance selection[J]. Journal of Scientific Computing, 2012, 51(2): 261-273.

[7] Sun Y, Liu Q, Tang J, et al. Learning discriminative dictionary for group sparse representation[J]. IEEE Transaction on Image Processing, 2014, 23(9): 3816-3826

[8] Hastie T, Tibshirani R, Friedman J, et al. The elements of statistical learning: Data mining, inference and prediction[J]. The Mathematical Intelligencer, 2005, 27(2): 83-85.

[9] Geng B, Li Y, Tao D, et al. Parallel lasso for large-scale video concept detection[J]. Multimedia, IEEE Transactions on, 2012, 14(1): 55-65.

[10] Zhou T, Tao D. Shifted subspaces tracking on sparse outlier for motion segmentation[C]// Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. Beijing, China: [s. n.], 2013: 1946-1952.

[11] Afonso M V, Bioucas-Dias J M, Figueiredo M A. Fast image recovery using variable splitting and constrained optimization [J]. Image Processing, IEEE Transactions on, 2010, 19(9): 2345-2356.

[12] Elad M, Matalon B, Zibulevsky M. Image denoising with shrinkage and redundant representations[C]// Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference, New York, USA: [s. n.], 2006, 2: 1924-1931.

[13] Figueiredo M A, Nowak R D. An em algorithm for wavelet-based image restoration[J]. Image Processing, IEEE Transactions on, 2003, 12(8): 906-916.

[14] Yu J, Rui Y, Tao D. Click prediction for web image reranking using multimodal sparse coding[J]. IEEE Transactions on Image Processing, 2014, 23(5): 2019-2032 .

[15] He L, Tao D, Li X, et al. Sparse representation for blind image quality assessment[C]// Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference. Rhode, Island: [s. n.], 2012, 1146-1153.

[16] Bioucas-Dias J M, Figueiredo M A. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing[C]// Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop. Reykjavik, Iceland: [s. n.], 2010: 1-4.

[17] Linear A. Foreword to the special issue on spectral unmixing of remotely sensed data[J]. IEEE Transactions on Geoscience and Remote Sensing, 2011, 49(11): 4103.

[18] Richards J A. Remote sensing digital image analysis[M]. New York: Springer, 1999.

[19] Efron B, Hastie T, Johnstone I, et al. Least angle regression[J]. The Annals of Statistics, 2004, 32(2): 407-499.

[20] Nesterov Y, Nesterov I E. Introductory lectures on convex optimization: A basic course[M]. USA: Springer, 2004.

[21] Nesterov Y. Smooth minimization of non-smooth functions[J]. Mathematical Programming, 2005, 103(1): 127-152.

[22] Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations and Trends R in Machine Learning, 2011, 3(1): 1-122.

[23] Becker S. R, Candès E J, Grant M C. Templates for convex cone problems with applications to sparse signal recovery[J]. Mathematical Programming Computation, 2011, 3(3): 165-218.

[24] 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.

[25] Goldstein T, ODonoghue B, Setzer S. Fast alternating direction optimization methods[J]. SIAM Journal Imaging Sciences,

2014, 7(3):1588-1623.
[26] Suzuki T. Dual averaging and proximal gradient descent for online alternating direction multiplier method[C]// Proceedings of the 30th International Conference on Machine Learning (ICML-13). Atlanta, USA:[s. n.], 2013: 392-400.
[27] Suzuki T. Stochastic dual coordinate ascent with alternating direction multiplier method[EB/OL]. http://arxiv.org/, 2014.
[28] Ouyang H, He N, Tran L,et al. Stochastic alternating direction method of multipliers[C]// Proceedings of the 30th International Conference on Machine Learning, Atlanta, USA:[s. n.], 2013: 80-88.
[29] Nesterov Y. Efficiency of coordinate descent methods on huge-scale optimization problems[J]. SIAM Journal on Optimization, 2012, 22(2):341-362.
[30] Liu J, Wright S J. Asynchronous stochastic coordinate descent: Parallelism and convergence properties [EB/OL]. http://arxiv.org/, 2014.
[31] Richtárik P, Taká˘c M. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function[J]. Mathematical Programming, 2014, 144(1-2):1-38.
[32] Shalev-Shwartz S , Zhang T. Stochastic dual coordinate ascent methods for regularized loss[J]. The Journal of Machine Learning Research, 2013, 14(1):567-599.
[33] Shalev-Shwartz S , Zhang T. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization[EB/OL]. http://link.springer.com/journal/10107, 2015.
[34] Zhang Y, Xiao L. Stochastic primal-dual coordinate method for regularized empirical risk minimization [EB/OL]. http://arxiv.org/, 2014.
[35] 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.
[36] Devolder O, Glineur F, Nesterov Y. First-order methods of smooth convex optimization with inexact oracle[J]. Mathematical Programming, 2014, 146(1-2):37-75.
[37] Avron H, Druinsky A, Gupta A. Revisiting asynchronous linear solvers: Provable convergence rate through randomization [EB/OL]. http://arxiv.org/, 2014.
[38] Recht B, Re C, Wright S, et al. Hogwild: A lock-free approach to parallelizing stochastic gradient descent[C]// Advances in Neural Information Processing Systems. Granada, Spain:[s. n.], 2011: 693-701.
[39] Richtárik P, Taká˘c M. Parallel coordinate descent methods for big data optimization [EB/OL]. http://arxiv.org/, 2014.
[40] Liu J, Wright S J, Ré C, et al. An asynchronous parallel stochastic coordinate descent algorithm [EB/OL]. http://arxiv.org/, 2014.
[41] Liu J, Wright S J, Sridhar S. An asynchronous parallel randomized kaczmarz algorithm [EB/OL]. http://arxiv.org/, 2014.
[42] Strohmer T, Vershynin R. A randomized kaczmarz algorithm with exponential convergence[J]. Journal of Fourier Analysis and Applications, 2009, 15(2):262-278.
[43] Gran M, Boyd S, Ye Y. Cvx: Matlab software for disciplined convex programming[EB/OL]. http://cvxr.com/cvx/, 2008.
[44] Candes E,Romberg J. l1-magic: Recovery of sparse signals via convex programming. [EB/OL]. http:// e- du/l1magic/downloads/l1magic, 2005.
[45] Goldstein T, Osher S. The split bregman method for l1-regularized problems[J]. SIAM Journal on Imaging Sciences, 2009, 2(2):323-343.
[46] Rudin L I Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms[J]. Physica D: Nonlinear Phenomena, 1992, 60(1):259-268.
[47] Kim S J, Koh K,Boyd S. Skystrend filtering[J]. Siam Review, 2009, 51(2):339-360.
[48] Zhu X, Huang Z, Cui J, et al. Video-to-shot tag propagation by graph sparse group lasso[J]. Multimedia, IEEE Transactions on, 2013, 15(3):633-646.
[49] Yuan M, Lin Y. Model selection and estimation in regression with grouped variables[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology) , 2006, 68(1):49-67.
[50] Ohlsson H,Ljung, L, Boyd S. Segmentation of arx-models using sum-of-norms regularization[J]. Automatica, 2010, 46(6):1107-1111.
[51] Cevher V, Becker S, Schmidt M. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics[J]. Signal Processing Magazine, IEEE, 2014,31(5):32-43.
[52] Parikh N, Boyd S. Proximal algorithms[J]. Foundations and Trends in Optimization, 2013, 1(3):123-231.

[53] Blackford L S, Choi J, Cleary A, et al. ScaLAPACK users' guide[EB/OL]. http://www.netlib.org/scalapack/scalapackug.ps. 2006.

[54] DemmelM J W, Heath M T, Van Der Vorst H A. Parallel numerical linear algebra[J]. Acta Numerica, 1993, 2:111-197.

[55] Gallivan K A, Plemmons R J, Sameh A. Parallel algorithms for dense linear algebra computations[J]. SIAM review, 1990, 32(1):54-135.

[57] Saha A , Tewari A. On the finite time convergence of cyclic coordinate descent methods[J]. Siam Journal of Optimization, 2013, 23(1):576-601.

[56] Friedman J, Hastie T, Höfling H, et al. Pathwise coordinate optimization[J]. The Annals of Applied Statistics, 2007, 1(2):302-332.

[58] Leventhal D, Lewis A S. Randomized methods for linear constraints: Convergence rates and conditioning[J]. Mathematics of Operations Research, 2010, 35(3):641-654.

[59] Shalev-Shwartz S, Tewari A. Stochastic methods for l 1-regularized loss minimization[J]. The Journal of Machine Learning Research, 2011, 12:1865-1892.

[60] Qu Z, Richtárik P, Zhang T. Randomized dual coordinate ascent with arbitrary sampling [EB/OL]. http://arxiv.org/, 2014.

[61] Nesterov Y. Gradient methods for minimizing composite functions[J]. Mathematical Programming, 2013, 140(1):125-161.

[62] Tseng P. On accelerated proximal gradient methods for convex-concave optimization [EB/OL]. http ://www.mit.edu/ ~dimitrib/PTseng/papers.html, 2008.

[63] Tseng P, Yun S. Block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization[J]. Journal of Optimization Theory and Applications, 2009, 140(3):513-535.

作者简介:**刘柳**(1987-),女,博士研究生,研究方向:机器学习, E-mail:Liu.Liu-2@student.uts.edu.au;**陶大程**(1978-),男,教授,研究方向:模式识别,机器学习,计算机视觉, E-mail:dacheng.tao@gmail.com。