梯度法的困难

考虑无约束优化问题

$$\min_{x\in\mathbb{R}^n} \quad f(x),$$

使用梯度下降法, 给出的迭代格式是

$$x^{k+1} = x^k - \alpha_k \nabla f\left(x^k\right).$$

由于梯度下降的基本策略是沿一阶最速下降方向迭代。当$\nabla^2 f(x)$ 的条件数较大时, 它的收敛速度比较缓慢(只用到一阶信息).

如果$f(x)$足够光滑, 我们可以利用$f(x)$的二阶信息改进下降方向, 以加速算法的迭代.

提纲

经典牛顿法

对于可微二次函数$f(x)$, 对于第k步迭代,我们现在考虑目标函数f在点x_k的二阶Taylor近似

$$f\left(x^k+d^k\right)=f\left(x^k\right)+\nabla f\left(x^k\right)^{\mathrm{T}} d^k+\frac{1}{2}\left(d^k\right)^{\mathrm{T}} \nabla^2 f\left(x^k\right) d^k+o\left(\left\|d^k\right\|^2\right),$$

忽略高阶项$o\left(\left\|d^k\right\|^2\right)$, 并将等式右边视作$d^k$的函数并极小化, 得

$$\nabla^2 f\left(x^k\right) d^k=-\nabla f\left(x^k\right). \tag{1}$$

方程(1)被称为**牛顿方程**, d^k被称为**牛顿方向**.
若$\nabla^2 f\left(x^k\right)$非奇异, 则可得到迭代格式

$$x^{k+1}=x^k-\nabla^2 f\left(x^k\right)^{-1} \nabla f\left(x^k\right). \tag{2}$$

注意上式中步长$\alpha_k=1$. (2)这种迭代格式被称为**经典牛顿法**.

提纲

经典牛顿法的收敛性

定理

经典牛顿法的收敛性 假设 f 二阶连续可微,且存在 x^* 的一个邻域 $N_\delta(x^*)$ 及常数 $L>0$ 使得

$$\left\|\nabla^2 f(x)-\nabla^2 f(y)\right\| \leqslant L\|x-y\|, \quad \forall x, y \in N_\delta\left(x^*\right)$$

如果 $f(x)$ 满足 $\nabla f\left(x^*\right)=0, \nabla^2 f\left(x^*\right) \succ 0$, 则对于迭代格式(2)有:

- 如果初始点离 x^* 足够近,则迭代点列 $\left\{x^k\right\}$ 收敛到 x^*;
- $\left\{x^k\right\}$ Q-二次收敛到 x^*;
- $\left\{\left\|\nabla f\left(x^k\right)\right\|\right\}$ Q-二次收敛到 0.

定理证明

根据经典牛顿法定义以及$\nabla f\left(x^*\right)=0$,得

$$\begin{aligned} x^{k+1}-x^* &= x^k-\nabla^2 f\left(x^k\right)^{-1}\nabla f\left(x^k\right)-x^* \\ &= \nabla^2 f\left(x^k\right)^{-1}\left[\nabla^2 f\left(x^k\right)\left(x^k-x^*\right)-\left(\nabla f\left(x^k\right)-\nabla f\left(x^*\right)\right)\right], \end{aligned} \tag{3}$$

注意到

$$\nabla f\left(x^k\right)-\nabla f\left(x^*\right)=\int_0^1 \nabla^2 f\left(x^k+t\left(x^*-x^k\right)\right)\left(x^k-x^*\right)\mathrm{d}t,$$

由此

$$\begin{aligned} &\left\|\nabla^2 f\left(x^k\right)\left(x^k-x^*\right)-\left(\nabla f\left(x^k\right)-\nabla f\left(x^*\right)\right)\right\| \\ =&\left\|\int_0^1\left[\nabla^2 f\left(x^k+t\left(x^*-x^k\right)\right)-\nabla^2 f\left(x^k\right)\right]\left(x^k-x^*\right)\mathrm{d}t\right\| \\ \leqslant&\int_0^1\left\|\nabla^2 f\left(x^k+t\left(x^*-x^k\right)\right)-\nabla^2 f\left(x^k\right)\right\|\left\|x^k-x^*\right\|\mathrm{d}t \\ \leqslant&\left\|x^k-x^*\right\|^2\int_0^1 Lt\,\mathrm{d}t(\text{Lip. 连续性})=\frac{L}{2}\left\|x^k-x^*\right\|^2. \end{aligned} \tag{4}$$

注意到, $\exists r>0$, 当 $\|x-x^*\| \leqslant r$ 时有 $\left\|\nabla^2 f(x)^{-1}\right\| \leqslant 2\left\|\nabla^2 f\left(x^*\right)^{-1}\right\|$ 成立 (请思考为何?), 故结合(3)及(4),得到

$$
\begin{aligned}
&\left\|x^{k+1}-x^*\right\| \\
\leqslant &\left\|\nabla^2 f\left(x^k\right)^{-1}\right\|\left\|\nabla^2 f\left(x^k\right)\left(x^k-x^*\right)-\left(\nabla f\left(x^k\right)-\nabla f\left(x^*\right)\right)\right\| \\
\leqslant &\left\|\nabla^2 f\left(x^k\right)^{-1}\right\| \cdot \frac{L}{2}\left\|x^k-x^*\right\|^2 \\
\leqslant &L\left\|\nabla^2 f\left(x^*\right)^{-1}\right\|\left\|x^k-x^*\right\|^2 .
\end{aligned}
\tag{5}
$$

当初始点 x^0 满足 $\left\|x^0-x^*\right\| \leqslant \min\left\{\delta, r, \frac{1}{2L\left\|\nabla^2 f\left(x^*\right)^{-1}\right\|}\right\}$ 时,迭代点列一直处于邻域 $N_{\hat{\delta}}\left(x^*\right)$ 中,故 $\left\{x^k\right\}$ Q-二次收敛到 x^*.

另一方面,由牛顿方程(1)可知

$$\begin{aligned}
\left\|\nabla f\left(x^{k+1}\right)\right\| &= \left\|\nabla f\left(x^{k+1}\right)-\nabla f\left(x^{k}\right)-\nabla^{2} f\left(x^{k}\right) d^{k}\right\| \\
&= \left\|\int_{0}^{1} \nabla^{2} f\left(x^{k}+t d^{k}\right) d^{k} d t-\nabla^{2} f\left(x^{k}\right) d^{k}\right\| \\
&\leqslant \int_{0}^{1}\left\|\nabla^{2} f\left(x^{k}+t d^{k}\right)-\nabla^{2} f\left(x^{k}\right)\right\|\left\|d^{k}\right\| d t \\
&\leqslant \frac{L}{2}\left\|d^{k}\right\|^{2} \leqslant \frac{1}{2} L\left\|\nabla^{2} f\left(x^{k}\right)^{-1}\right\|^{2}\left\|\nabla f\left(x^{k}\right)\right\|^{2} \\
&\leqslant 2 L\left\|\nabla^{2} f\left(x^{*}\right)^{-1}\right\|^{2}\left\|\nabla f\left(x^{k}\right)\right\|^{2} .
\end{aligned}$$

这证明梯度的范数Q-二次收敛到0.

收敛速度分析

牛顿法收敛速度非常快, 但实际使用中会存在若干限制因素.

- 初始点x^0 需要距离最优解充分近(因为局部收敛性)
  应用时常以梯度类算法先求得较低精度的解, 后用牛顿法加速.
- $\nabla^2 f(x^*)$ 需正定, 半正定条件下可能退化到Q-线性收敛.
- $\nabla^2 f$ 的条件数较高时, 将对初值的选择作出较严苛的要求.

提纲

修正牛顿法

经典牛顿法的基本格式如下:

$$x^{k+1} = x^k - \nabla^2 f\left(x^k\right)^{-1} \nabla f\left(x^k\right).$$

除了计算、存储代价昂贵之外, 经典牛顿法还存在如下问题:

- 海瑟矩阵可能非正定,导致牛顿方向其实并非下降方向;
- 初始点离最优值点较远时候迭代不稳定(步长固定), 因而算法可能不收敛.

为提高算法的稳定性, 从以上两方面考虑, 应该:

- 对$\nabla^2 f(x)$进行修正,使其正定(所有特征值大于0);
- 用线搜索确定步长来增加算法的稳定性(Wolfe, Goldstein, Armijo).

综上考虑, 我们提出下面带线搜索的牛顿方法, 并称其为**修正牛顿方法**.

带线搜索的修正牛顿法

Algorithm 1 带线搜索的修正牛顿法

1: 给定初始点x^0.
2: **for** $k = 0, 1, 2, \cdots$ **do**
3: 确定矩阵E^k使得矩阵$B^k \overset{\text{def}}{=} \nabla^2 f\left(x^k\right) + E^k$正定且条件数较小.
4: 求解修正的牛顿方程$B^k d^k = -\nabla f\left(x^k\right)$得方向$d^k$.
5: 使用任意一种线搜索准则确定步长α_k.
6: 更新$x^{k+1} = x^k + \alpha_k d^k$.
7: **end for**

在上述算法中, B^k (即E^k) 的选取较为关键, 需要注意:

- B^k应具有较低的条件数(原因见收敛性定理);
- 对$\nabla^2 f(x)$的改动较小, 以保存二阶信息;
- B^k本身的计算代价不应太高.

修正牛顿法的全局收敛性

定理

修正牛顿法全局收敛性定理 令 f 在开集 $\mathcal{D}$ 上二阶连续可微,且初始点 x^0 满足 $\{x \in \mathcal{D} : f(x) \leq f(x_0)\}$ 为紧集.若算法**1**中 B^k 的条件数上界存在,即

$$\kappa(B_k) = \|B_k\| \left\|B_k^{-1}\right\| \leq C, \quad \exists C > 0, \forall k = 0, 1, 2, \cdots \tag{6}$$

则

$$\lim_{k \to \infty} \nabla f(x_k) = 0,$$

即算法具有全局收敛性.

上述定理的证明冗长, 读者可具体参考: Newton's method, in Studies in Numerical Analysis, vol. 24 of MAA Studies in Mathematics, The Mathematical Association of America, 1984, pp. 29–82.

修正矩阵E^k的显式选取

进一步地, 在修正步长后, 为还使 B^k 正定, 我们修正海瑟矩阵的特征值.
首先做特征分解

$$\nabla^2 f\left(x_k\right)=Q\Lambda Q^T,$$

其中,Q 为正交矩阵,Λ 为对角矩阵.

取$E^k=\tau_k I$,即单位阵的常数倍, 代入上式, 则有

$$B^k=Q\left(\Lambda+\tau_k I\right)Q^T.$$

当正数τ_k 足够大的时候, 可保证 B^k 正定, 但 d^k 会接近负梯度方向, 退化为一阶方法.
一种简单的技巧是对τ_k做截断, 即

$$\tau_k=\max\left\{0,\delta-\lambda_{\min}(\nabla^2 f\left(x^k\right))\right\},\quad (\text{给定}\delta>0)$$

然而, 此时$\lambda_{\min}(\nabla^2 f\left(x^k\right))$ 通常难以计算. 我们通常使用Cholesky分解试探性取τ_k, 从而避免这个问题.

下面利用Cholesky分解给出另一种算法 (a_{ii} 表示$\nabla^2 f\left(x^k\right)$ 的对角元素)

Algorithm 2 Cholesky分解(增加数量矩阵)

选取β, σ. (如$\beta = 10^{-3}, \sigma = 2$)
if $\min_i \{a_{ii}\} > 0$ **then**
  $\tau_0 = 0$
else
  $\tau_0 = -\min_i \{a_{ii}\} + \beta$
end if
for $t = 0, 1, 2, \cdots$ **do**
  尝试用Cholesky算法计算:$LL^T = \nabla^2 f\left(x^k\right) + \tau_t I$
  if Cholesky算法成功运行 **then**
    终止循环并返回L
  else
    $\tau_{t+1} = \max\{\sigma\tau_t, \beta\}$
  end if
end for

缺陷:可能需要多次的试验,而每一步都对 $\nabla^2 f\left(x^k\right) + \tau_t I$ 做分解的计算代价大(可考虑较大的σ,如$\sigma = 10$)

修正矩阵E^k的隐式选取

直接对$\nabla^2 f\left(x^k\right)$进行Cholesky分解可能会失败, 因此考虑修正分解算法. 回顾Cholesky 分解的定义:

$$A = LDL^{\mathrm{T}},$$

其中, $A = (a_{ij})$对称正定, $L = (l_{ij})$是对角线元素均为1 的下三角矩阵, $D = \mathrm{Diag}\left(d_1, d_2, \cdots, d_n\right)$ 是对角矩阵且对角线元素均为正.

Algorithm 3 标准Cholesky分解

给定对称矩阵$A = (a_{ij})$.
for $j = 1, 2, \cdots, n$ **do**
  $c_{jj} = a_{jj} - \sum_{s=1}^{j-1} d_s l_{js}^2$
  $d_j = c_{jj}$
  for $i = j+1, j+2, \cdots, n$ **do**
    $c_{ij} = a_{ij} - \sum_{s=1}^{j-1} d_s l_{is} l_{js}$
    $l_{ij} = c_{ij}/d_j.$
  end for
end for

修正Cholesky分解

修正的目标:使 d_i 为正数,且 D 和 L 中元素值不太大.
具体地, 选取正的参数δ, β, 并要求在算法(3)中第j个外循环满足:

$$d_j \geq \delta, \quad \left|l_{ij}\sqrt{d_j}\right| \leq \beta, \quad i = j+1, j+2, \ldots, n \tag{7}$$

考虑下面的更新方式:

$$d_j = \max\left\{|c_{jj}|, \left(\frac{\theta_j}{\beta}\right)^2, \delta\right\}, \quad \theta_j = \max_{i>j}|c_{ij}|$$

注意到算法(3)中有$c_{ij} = l_{ij}d_j$, 根据下式可知条件(7)成立:

$$\left|l_{ij}\sqrt{d_j}\right| = \frac{|c_{ij}|}{\sqrt{d_j}} \leq \frac{|c_{ij}|\,\beta}{\theta_j} \leq \beta, \quad \forall i > j$$

- θ_j 可以在之前d_j 计算, 因为c_{ij} 在算法(3)内循环的计算不包含d_j.
- 修正算法其实是在计算$\nabla^2 f\left(x^k\right) + E^k$ 的Cholesky 分解 (E^k 为对角元素非负的对角阵). 当$\nabla^2 f\left(x^k\right)$ 正定且条件数足够小时有$E^k = 0$.
- 可证明, 由修正算法得到的B_k 的条件数存在上界, 即满足(6)

提纲

非精确牛顿法:背景

应用在大规模参数的场合时,牛顿法有以下问题:

- 参数规模大, 可能数以亿计, 直接使用牛顿法则存储和计算都存在困难;
- 无法存储海瑟矩阵, 同时对海瑟矩阵做Cholesky分解的代价过高.

为解决这些问题, 我们引入非精确牛顿法, 即通过解牛顿方程的形式以求出牛顿方向(非精确).

性质

非精确牛顿法的优势　此方法有优良的收敛性质和速度, 且能在海瑟矩阵不定的时候找到有效的下降方向, 甚至可以不显式计算、存储海瑟矩阵*(Hessian-free).*

非精确牛顿法:算法

下面介绍非精确牛顿法的基本想法.

首先回到牛顿方程: $\nabla^2 f\left(x^k\right) d^k = -\nabla f\left(x^k\right)$. 非精确牛顿法不精确求解牛顿方向$d^k$, 而是用迭代法(如共轭梯度法)求解线性方程组. 这两者的区别是, 使用迭代法求解时, 我们可以在一定的精度下提前停机, 以提高求解效率.

为控制精度, 我们引入向量r^k 来表示残差, 将上述方程记为

$$\nabla^2 f\left(x^k\right) d^k = -\nabla f\left(x^k\right) + r^k, \tag{8}$$

因此终止条件可设置为

$$\left\|r^k\right\| \leqslant \eta_k \left\|\nabla f\left(x^k\right)\right\|, \tag{9}$$

其中序列$\{\eta_k\}$的值可自由设置, 不同的设置将导致不同的精度要求, 从而使算法有不同的收敛速度.

收敛性分析

我们叙述非精确牛顿法的局部收敛定理.

定理

非精确牛顿法的收敛定理 设函数$f(x)$二阶连续可微, 且$\nabla^2 f(x^*)$正定, 则在非精确牛顿法中,

(1) 若$\exists t < 1$,使得η_k满足$0 < \eta_k < t, k = 1, 2, \cdots$, 且起始点$x_0$充分靠近$x^*$并迭代最终收敛到$x^*$, 则梯度$\nabla f(x^k)$以*Q*-线性收敛速度收敛;

(2) 若$\lim_{k\to\infty} \eta_k = 0$成立, 则梯度$\nabla f(x^k)$以*Q*-超线性收敛速度收敛;

(3) 若*(1)*或*(2)*成立, 且$\nabla^2 f$在x^*附近*Lip.*连续, $\eta_k = O\left(\left\|\nabla f(x^k)\right\|\right)$, 则梯度$\nabla f(x^k)$以*Q*-二次收敛速度收敛.

定理证明

我们只给出一个不严格的证明, 请读者主要体会证明思路.

注意到$\nabla^2 f(x)$在x^*处正定, 在x^*附近连续, 故存在正常数L, 使得

$$\left\|\left(\nabla^2 f(x^k)\right)^{-1}\right\| \leq L, \quad (\forall x^k \text{同} x^* \text{足够接近})$$

代入(8), 得(第二个不等式用到了$\eta_k < 1$)

$$\left\|d^k\right\| \leq L\left(\left\|\nabla f(x^k)\right\| + \left\|r^k\right\|\right) \leq 2L\left\|\nabla f(x^k)\right\|.$$

利用Taylor展式和$\nabla^2 f$的连续性, 得到

$$\begin{aligned}
\nabla f(x^{k+1}) &= \nabla f(x^k) + \nabla^2 f(x^k) d^k + \int_0^1 \left[\nabla^2 f\left(x^k + t d^k\right) - \nabla^2 f\left(x^k\right)\right] d^k dt \\
&= \nabla f(x^k) + \nabla^2 f(x^k) d^k + o\left(\left\|d^k\right\|\right) \\
&= \nabla f(x^k) - \left(\nabla f(x^k) - r^k\right) + o\left(\left\|\nabla f(x^k)\right\|\right) \\
&= r^k + o\left(\left\|\nabla f(x^k)\right\|\right).
\end{aligned}$$

定理证明

上式中两边取范数, 结合精度控制式(9), 得到

$$\left\|\nabla f(x^{k+1})\right\| \leq \eta_k \left\|\nabla f(x^k)\right\| + o\left(\left\|\nabla f(x^k)\right\|\right) \leq \left(\eta_k + o(1)\right) \left\|\nabla f(x^k)\right\|.$$

当x^k足够接近x^*时,$o(1)$项可被$(1-t)/2$控制,则

$$\left\|\nabla f(x^{k+1})\right\| \leq \left(\eta_k + (1-t)/2\right) \left\|\nabla f(x^k)\right\| \leq \frac{1+t}{2} \left\|\nabla f(x^k)\right\|.$$

由于$t < 1$,故梯度梯度$\nabla f(x^k)$以Q-线性收敛速度收敛.

推论

从以上证明过程可以看出如下的结果:

$$\frac{\left\|\nabla f(x^{k+1})\right\|}{\left\|\nabla f(x^k)\right\|} \leq \eta_k + o(1).$$

若$\lim_{k\to\infty} \eta_k = 0$成立,可有Q-超线性收敛的结论

$$\lim_{k\to\infty} \frac{\left\|\nabla f(x^{k+1})\right\|}{\left\|\nabla f(x^k)\right\|} = 0.$$

若$\nabla^2 f$ 在x^* 附近Lip.连续, 令$\eta_k = O\left(\left\|\nabla f(x^k)\right\|\right)$, 可有二次收敛的结论

$$\left\|\nabla f(x^{k+1})\right\| = O\left(\left\|\nabla f(x^k)\right\|^2\right).$$

因此, 在实际应用时:

- 取$\eta_k = \min\left(0.5, \sqrt{\left\|\nabla f(x^k)\right\|}\right)$, 可成立局部的超线性收敛;
- 取$\eta_k = \min\left(0.5, \left\|\nabla f(x^k)\right\|\right)$, 可成立局部的二次收敛.

线搜索Newton-CG法

最后我们讨论求解牛顿方程的方法.
一种广泛应用的方法是线搜索Newton-CG法, 它利用共轭梯度迭代求解牛顿方程.

- 由于共轭梯度法要求被解方程的系数矩阵正定, 但$\nabla^2 f$ 可能非正定, 因此在迭代时需要辨别$\nabla^2 f$的正定性以保证算法有效.

具体算法在下一面给出. 其中B_k 表示$\nabla^2 f(x^k)$; $\{z_j\}$ 表示CG法从零向量开始的迭代序列, 最终收敛到牛顿方向.
我们采用的迭代终止条件是$\eta_k = \min\left(0.5, \sqrt{\|\nabla f(x^k)\|}\right)$.

CG法解牛顿方程的优势是只用到海瑟矩阵-向量积而无需求出$\nabla^2 f$, 故可以被设计成*Hessian-free*方法, 在大规模问题下解决Hessian矩阵计算或存储的困难.

线搜索Newton-CG法:算法

Algorithm 4 线搜索Newton-CG法

给定初始点x^0.
for $k = 0, 1, 2, \cdots$ **do**
  令$\epsilon_k = \eta_k \left\|\nabla f(x^k)\right\|$, $z_0 = 0, r_0 = \nabla f(x^k), p_0 = -r_0 = -\nabla f(x^k)$.
  for $j = 0, 1, 2, \cdots$ **do**
    if $p_j^T B_k p_j \leq 0$ (判断是否是负曲率方向) **then**
      终止循环.$j = 0$ 则返回$d^k = -\nabla f$,否则返回$d^k = z_j$.
    end if
    $\alpha_j = r_j^T r_j / p_j^T B_k p_j$, $z_{j+1} = z_j + \alpha_j p_j$, $r_{j+1} = r_j + \alpha_j B_k p_j$
    if $r_j < \epsilon_k$ (判断是否达到收敛条件) **then**
      终止循环. 返回$d^k = z_{j+1}$.
    end if
    $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$, $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$
  end for
  线搜索确定步长α_k(符合条件则取$\alpha_k = 1$),更新$x^{k+1} = x^k + \alpha_k d^k$.
end for

提纲

提出正则化牛顿算法的背景

下面我们介绍一种修正牛顿法：正则化牛顿法.

对函数$f(x)$ Taylor展开到二阶, 得到

$$f(x^k+d)=f(x^k)+\nabla f(x^k)^{\mathrm{T}}d+\frac{1}{2}d^{\mathrm{T}}\nabla^2 f(x^k+td)d, \tag{10}$$

其中$t\in(0,1)$与d有关.

利用$f(x)$的二阶近似, 此时在x处有

$$m_k(x)=f(x^k)+\nabla f(x^k)^{\mathrm{T}}(x-x^k)+\frac{1}{2}(x-x^k)^{\mathrm{T}}H^k(x-x^k), \tag{11}$$

其中H^k是$\nabla^2 f(x^k)$或其近似矩阵.

子问题的提出

- 信赖域算法考虑添加约束$\left\|x-x^k\right\| \leqslant \Delta_k$, 其中$\Delta_k$为每步的最大半径. 极小化一系列的子问题:

$$\min_{x\in\mathbb{R}^n} \quad m_k(x), \qquad \text{s.t.} \ \left\|x-x^k\right\| \leqslant \Delta_k. \tag{12}$$

- 利用正则化的思想将约束加在目标函数上, 得到

$$\min_{x\in\mathbb{R}^n} \quad m_k(x) = \nabla f(x^k)^{\mathrm{T}}(x-x^k)+\frac{1}{2}(x-x^k)^{\mathrm{T}}H^k(x-x^k)+\frac{\sigma_k}{2}\left\|x-x^k\right\|^2, \tag{13}$$

其中σ_k是系列正则化参数.

- 通过近似求解每步产生的正则化子问题(13), 我们即可能产生逼近$f(x)$极小点的收敛点列.

子问题(13)的求解

在求解子问题时, 使用**PCG**技术, 即考虑线性方程

$$\nabla m_k(x^k) + \nabla^2 m_k(x^k)[x - x^k] = 0.$$

联合上述线性方程的正则化子问题可由一种改进的共轭梯度方法求解, 其基本思路是:

思路

(1) 我们使用共轭梯度方法求解, 且设置终止条件为残差变小或产生负曲率.

(2) 我们要在共轭方向的基础上构建一个新的梯度相关方向, 并沿此方向执行充分的线搜索, 以充分优化目标函数.

子问题的求解

利用改进的共轭梯度方法求解子问题(13)的具体算法如下.

算法5 改进的共轭梯度算法(解子问题(13))

s0 输入: $T>0, \theta>1, \epsilon \geqslant 0, \eta_0=0, r^0=\nabla m_k(x^k), p^0=r^0, i=0.$
输出: 关于子问题(13)的近似极小点x的更新方向$\xi^k \simeq x-x^k$.
While $i \leqslant n-1$ **do**

s1 计算$\pi_i=\left\langle p^i, \nabla^2 m_k\left(x^k\right)\left[p^i\right]\right\rangle$.

s2 **If** $\pi_i / \left\langle p^i, p^i\right\rangle \leqslant \epsilon$ **then**
If $i=0$ **then** 设置$s^k=-p^0, d^k=0;$
Else 设置$s^k=\eta_i$.
End If
If $\pi_i / \left\langle p^i, p^i\right\rangle \leqslant -\epsilon$ **then**
设置$d^k=p^i, \sigma_{est}=|\pi|_i / \left\langle p^i, p^i\right\rangle;$
Else 设置$d^k=0$.
End If
Break
End If

子问题的求解

s3 设置$\alpha_i = \langle r^i, r^i \rangle / \pi_i$, $\eta_{i+1} = \eta_i + \alpha_i p^i$;
设置$r^{i+1} = r^i + \alpha_i \nabla^2 m_k\left(x^k\right)\left[p^i\right]$.

s4 **If** $\left\|r^{i+1}\right\|_{x^k} \leqslant \min\left\{\left\|r^0\right\|_{x^k}^{\theta}, T\right\}$ **then**
令$s^k = \eta_{i+1}$, $d^k = 0$; **Break**
End If

s5 令$\beta_{i+1} = \langle r^{i+1}, r^{i+1} \rangle / \langle r^i, r^i \rangle$;
令$p^{i+1} = -r^{i+1} + \beta_{i+1} p^i$.
$i \leftarrow i + 1$.
End While

s6 利用以下的截断公式更新ξ^k.

其中, 所谓的截断公式即

$$\xi^k = \begin{cases} s^k + \tau_k d^k, & \text{If} \quad d^k \neq 0, \\ s^k, & \text{Otherwise.} \end{cases} \tag{14}$$

且$\tau_k = \dfrac{\langle d^k, \nabla m_k(x^k) \rangle}{\langle d^k, \nabla^2 m_k(x^k)[d^k] \rangle}$.

子问题的求解

我们从上述算法的具体流程可以分析, 所谓的改进算法与原始共轭梯度方法的不同之处体现在步骤**s2**中, 此时算法生成了s^k和d^k. 它们的含义和作用如下所示:

性质

在上述求解子问题的改进共轭梯度算法中,

- d^k中蕴含负曲率的信息;
- s^k对应共轭梯度算法中的"usual"输出向量.

正如**s2**所述, 只有算法执行时出现了负曲率的情形, d^k才不为0, 因此改进算法的好处在于**如遇负曲率, 则可由负曲率方向和负梯度方向组合形成下降方向**ξ^k.

(ξ^k是下降方向, 相关证明由**参考文献[2]**给出)

子问题的求解

一旦根据子问题生成了下降方向ξ^k, 如同我们在牛顿法中做的那样, 就可以接着使用线搜索沿ξ^k生成子问题的下降点z^k, 即

$$z^k = x^k + \alpha_k \xi^k, \tag{15}$$

其中步长α_k由(单调)Armijo准则产生, 即

$$\alpha_k = \alpha_0 \delta^h,$$

其中α_0是属于$(0,1]$的给定常数, $h \in \mathbb{Z}$是满足条件

$$\begin{cases} m_k\left(z^k\right) \leqslant \rho \alpha_k \left\langle \nabla m_k\left(x^k\right), \xi^k \right\rangle, \\ \rho, \delta \in (0,1), \end{cases}$$

的最小整数.
称上述的z^k是子问题的**试验点**.

子问题近似原问题的方式

解决了子问题的求解后，我们再考虑子问题和原问题之间的关系. 因为子问题的下降解可能不使得原问题也下降，因此需要探究z^k是否使原问题下降.

我们希望接受使得原函数充分下降的试验点，因此需要定义一个衡量预估下降量和实际下降量的相对比例

$$\rho_k \triangleq \frac{f\left(z^k\right)-f\left(x^k\right)}{m_k\left(z^k\right)}, \tag{16}$$

它体现了函数值实际下降量和预估下降量之间的比值.

子问题近似原问题的方式

关于相对比例, 有如下的性质保证了算法的逻辑有效性.

性质

(1) 如果$\rho_k \to 1$, 则说明用$m_k(x)$近似$f(x)$比较成功, 那么既然z^k是可以使$m_k(x)$下降的试验点, 则自然也可以使$f(x)$进一步下降;

(2) 反之, 则说明我们用$m_k(x)$近似的效果不好, 自然也就没办法保证试验点对于$f(x)$的下降性能, 因此应该舍弃试验点.

因此基于相对比例ρ_k的x^{k+1}更新公式应如

$$x^{k+1} = \begin{cases} z^{k+1}, & \text{If} \quad \rho_k \geqslant c_1 > 0, \\ x^k, & \text{Otherwise.} \end{cases} \tag{17}$$

子问题近似原问题的方式

合理的正则化参数需要正确反映子问题对原问题的近似能力与正则化项惩罚力度的关系.
一般而言, 当子问题近似原问题的程度较好时, 我们认为试验点也是$f(x)$的下降点, 因此该扩大Δ_k, 体现在(13)中即是减小σ_k; 反之则应增大.

基于以上思想, 我们可以设计如下分段的系列正则化参数σ_k:

$$\sigma_{k+1} = \begin{cases} (0, \gamma_0\sigma_k], & \text{If} \quad \rho_k \geqslant c_2, \\ [\gamma_0\sigma_k, \gamma_1\sigma_k], & \text{If} \quad c_1 \leqslant \rho_k < c_2, \\ [\gamma_0\sigma_k, \gamma_2\sigma_k], & \text{Otherwise}, \end{cases} \tag{18}$$

其中分段常系数$0 < c_1 \leqslant c_2 < 1$且$0 < \gamma_0 < 1 < \gamma_1 \leqslant \gamma_2$.

自适应的正则化牛顿方法

综上所述, 对于每一步迭代产生的x^k, 先计算子问题(13)的一个近似下降方向, 再求试验点z^k, 然后决定是否接受z^k作为x^{k+1}执行下次迭代以及如何更新正则化参数σ_k, 即是算法的基本框架. 我们可以提出如下自适应的正则化牛顿方法.

算法6 自适应的正则化牛顿方法

---

s0 输入: 可行的初始点$x^0 \in \mathbb{R}^n$, 正则化参数$\sigma_0 > 0$;
设置$0 < c_1 \leqslant c_2 < 1, 0 < \gamma_0 < 1 < \gamma_1 \leqslant \gamma_2, k = 0$.
输出: $f(x)$的极小点x.
While 未达到停机条件 **do**

s1 根据子问题(13), 利用**算法5**求解(13)的下降方向ξ^k;
根据ξ^k, 利用公式(15)求解试验点z^k.

s2 基于(16)计算子问题与原问题在试验点z^k处的相对比例ρ_k.

s3 基于ρ_k和(17)更新x^{k+1}.

s4 基于ρ_k和(18)更新σ_{k+1}.
$k \leftarrow k + 1$.

End While

提纲

逻辑回归模型

考虑二分类的逻辑回归模型

$$\min_x \quad \ell(x) \overset{\text{def}}{=} \frac{1}{m}\sum_{i=1}^{m} \ln\left(1+\exp\left(-b_i a_i^{\mathrm{T}} x\right)\right) + \lambda\|x\|_2^2.$$

为使用牛顿法, 需要计算目标函数的梯度与海瑟矩阵:

$$\begin{aligned}\nabla \ell(x) &= \frac{1}{m}\sum_{i=1}^{m} \frac{1}{1+\exp\left(-b_i a_i^{\mathrm{T}} x\right)} \cdot \exp\left(-b_i a_i^{\mathrm{T}} x\right) \cdot (-b_i a_i) + 2\lambda x \\ &= -\frac{1}{m}\sum_{i=1}^{m} \left(1-p_i(x)\right) b_i a_i + 2\lambda x,\end{aligned}$$

其中$p_i(x) = \frac{1}{1+\exp\left(-b_i a_i^{\mathrm{T}} x\right)}$.

进一步对$\nabla \ell(x)$求导, 成立

$$\begin{aligned}
\nabla^2 \ell(x) &= \frac{1}{m} \sum_{i=1}^{m} b_i \cdot \nabla p_i(x) a_i^{\mathrm{T}} + 2\lambda I \\
&= \frac{1}{m} \sum_{i=1}^{m} b_i \frac{-1}{\left(1 + \exp\left(-b_i a_i^{\mathrm{T}} x\right)\right)^2} \cdot \exp\left(-b_i a_i^{\mathrm{T}} x\right) \cdot \left(-b_i a_i a_i^{\mathrm{T}}\right) + 2\lambda I \\
&= \frac{1}{m} \sum_{i=1}^{m} \left(1 - p_i(x)\right) p_i(x) a_i a_i^{\mathrm{T}} + 2\lambda I \quad \left(b_i^2 = 1\right).
\end{aligned}$$

逻辑回归模型

引入矩阵$A = [a_1, a_2, \cdots, a_m]^{\mathrm{T}} \in \mathbb{R}^{m\times n}$, 向量$b = (b_1, b_2, \cdots, b_m)^{\mathrm{T}}$, 以及

$$p(x) = (p_1(x), p_2(x), \cdots, p_m(x))^{\mathrm{T}},$$

则可重写梯度和海瑟矩阵为

$$\nabla \ell(x) = -\frac{1}{m} A^{\mathrm{T}}(b - b \odot p(x)) + 2\lambda x,$$
$$\nabla^2 \ell(x) = \frac{1}{m} A^{\mathrm{T}} W(x) A + 2\lambda I,$$

其中$W(x)$ 为由$\{p_i(x)\,(1 - p_i(x))\}_{i=1}^m$ 生成的对角矩阵.

则最终牛顿法迭代格式可以写作:

$$x^{k+1} = x^k + \left(\frac{1}{m} A^{\mathrm{T}} W\left(x^k\right) A + 2\lambda I\right)^{-1} \left(\frac{1}{m} A^{\mathrm{T}}\left(b - b \odot p\left(x^k\right)\right) - 2\lambda x^k\right).$$

逻辑回归模型

我们得到了牛顿法的迭代格式, 因此可以调用牛顿法直接求解逻辑回归问题. 正如我们对牛顿方程处理思路的不同, 若变量规模不大, 则可尝试利用正定矩阵的Cholesky分解求解牛顿方程; 若变量规模较大, 则可以使用共轭梯度法对方程进行不精确的求解.

我们使用LIBSVM网站的数据集(具体数据集见下表), 对不同的数据集均调用非精确CG-牛顿法求解, 设置精度条件为

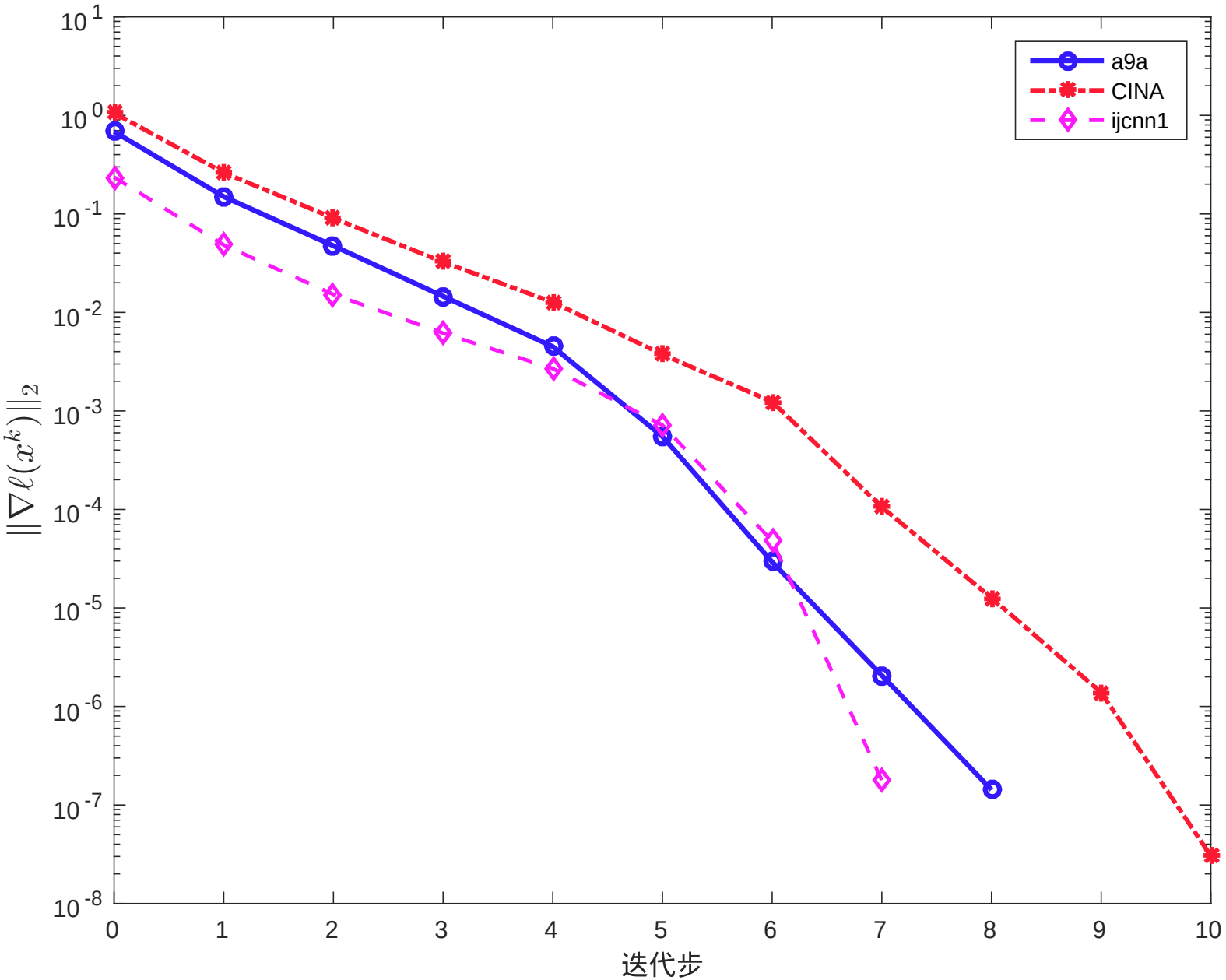
$$\left\|\nabla^2 \ell\left(x^k\right) d^k+\nabla \ell\left(x^k\right)\right\|_2 \leqslant \min\left\{\left\|\nabla \ell\left(x^k\right)\right\|_2^2, 0.1\left\|\nabla \ell\left(x^k\right)\right\|_2\right\},$$

Table: LIBSVM数据集(部分)

名称 \ 维数	m	n
a9a	16281	122
ijcnn1	91701	22
CINA	3206	132

在数据集中进行算法测试, 数值结果如下图所示. 从图中可以看到, 精确解附近梯度范数具有Q-超线性收敛性.

数值结果


a9a
CINA
ijcnn1
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
$\|\nabla \ell(x^k)\|_2$
0
1
2
3
4
5
6
7
8
9
10
迭代步

参考文献

[1] Nocedal J, Wright S. Numerical optimization[M]. Springer Science and Business Media, 2006.

[2] Hu J, Milzarek A, Wen Z, et al. Adaptive quadratically regularized Newton method for Riemannian optimization[J]. SIAM Journal on Matrix Analysis and Applications, 2018, 39(3): 1181-1207.