SLMQN：一个解大规模边界约束非线性规划 子空间 有限储存拟牛顿方法方法的 FORTRAN 优化软件 [1]

倪勤 袁亚湘

(中科院计算数学所科学与工程计算国家重点实验室)

SLMQN: A FORTRAN Code of Subspace Limited Memory Quasi-Newton Method for Large-Scale Bound Constrained Nonlinear Optimization

Q. NI and Y. Yuan
(ICMSEC, Academia Sinica, Beijing 100080)

Abstract

SLMQN is a subspace limited memory quasi-Newton algorithm for solving large-scale bound constrained nonlinear programming problems. The algorithm is suituable to these large problems in which the Hessian matrix is difficult to compute or is dense, or the number of variables is too large to store and compute an $n \times n$ matrix. Due to less storage requirement, this algorithm can be used in PCs for solving medium-sized and large problems. The algorithm is implemented in Fortran 77.

1 导言

SLMQN 优化软件是用于求解下列带有边界约束的非线性规划问题

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && l \le x \le u, \end{aligned} \tag{1}$$

其中 $x \in R^n$。l 和 u 是常数向量，它们的分量可以是无穷大 。这意味着 SLMQN 也可用于解无约束最优化问题 。$f(x)$ 要求是连续可微的 。用户需要提供计算 f 以及它的 梯度$g(x)$ 的子程序 。算法不需要计算 f 的Hesse 阵。

该算法利用 有限储存拟牛顿校正自由变量，利用修正 的梯度法校正积极约束变量 。在每步迭代中，搜索方向由两部分组成，一部分是子空间 有限储存拟牛顿方向，另一部分是 子空间修正 梯度方向 。用户可以通过选择参数 m 来控制 SLMQN 软件 的储存量 。m 是 有限储存BFGS 逆校正 的次数 。软件大约需要 (10+2m)n 储存单元 。由于 m 的适合取值一般为 $2 \le m \le 10$, 因此 SLMQN 软件可以用来解非常大的问题 。

本文结构如下 。在第二节概述 SLMQN 算法 。SLMQN 软件 的使用在第三节中讨论 。在第四节中给出终止原则和出错信息 。数值结果在第五节中给出 。

[1] 由国家自然科学基金、国家教委留学回国人员资助费等资助.

2 SLMQN 算法

SLMQN 算法是一个解问题 (1)的子空间有限储存拟牛顿算法 。首先，我们描述 搜索方向 的选定 。搜索方向由 子空间拟牛顿方向和 子空间修正 梯度方向组成 。为了表示这个组合，我们定义积极约束集 $\mathcal{A}(x)$ 和它 的补集 $\mathcal{B}(x)$ 为

$$\begin{aligned} \mathcal{A}(x) &= \{i: \ \ l_i \le x_i \le l_i + \epsilon_b \quad \text{或} \quad u_i - \epsilon_b \le x_i \le u_i\}, \\ \mathcal{B}(x) &= \{1, \ldots, n\}/\mathcal{A}(x) = \{i: \ \ l_i + \epsilon_b < x_i < u_i - \epsilon_b\}. \end{aligned} \tag{2}$$

下标在 $\mathcal{A}(x)$ 的变量称为积极约束变量，而下标在 $\mathcal{B}(x)$ 的变量称为自由变量 。参数ϵ_b 是足够小 的正数，例如 10^{-8}。

子空间拟牛顿方向是定义在所有自由变量组成 的子空间上，为 $-P_1^{(k)} H_k P_1^{(k)} g(x_k)$. 这里 $g(x) = \bigtriangledown f(x)$, $P_1^{(k)}$ 是一个 $n \times n$ 对角 矩阵, 如果 $i \in \mathcal{B}(x_k)$, 那么第 i 个对角元为 1, 反之为 0。H_k 是一个全空间 Hesse 逆 矩阵的近似 。H_k 通过 有限储存BFGS m 逆校正 (见 [3] 和 [5]) 来 储存和计算 。

为了获得 搜索方向 的另一部分，我们分解 $\mathcal{A}(x)$ 为下列三个部分，

$$\begin{aligned} \mathcal{A}_1(x) &= \{i: \ \ x_i = l_i \ \ \text{或} \ \ x_i = u_i, \ g_i(x)(l_i + u_i - 2x_i) \ge 0\}, \\ \mathcal{A}_2(x) &= \{i: \ \ l_i \le x_i \le l_i + \epsilon_b \ \ \text{或} \ \ u_i - \epsilon_b \le x_i \le u_i, \ g_i(x)(l_i + u_i - 2x_i) < 0\}, \\ \mathcal{A}_3(x) &= \{i: \ \ l_i < x_i \le l_i + \epsilon_b \ \ \text{或} \ \ u_i - \epsilon_b \le x_i < u_i, \ g_i(x)(l_i + u_i - 2x_i) \ge 0\}. \end{aligned} \tag{3}$$

我们固定下标在 $\mathcal{A}_1(x)$ 变量不动，校正下标在 $\mathcal{A}_2(x)$ 和 $\mathcal{A}_3(x)$ 中 的变量 。因此 子空间修正 梯度方向为

$$-(P_2^{(k)} P_2^{(k)T} + P_3^{(k)} P_3^{(k)T} \Lambda_k) g(x_k),$$

其中 $P_j^{(k)}$ 是 $k_j \times n$ 矩阵, k_j 是 $\mathcal{A}_j(x_k)$ 的基数，$P_j^{(k)}$ 的每一列是单位 矩阵中第 i 个列向量且 $i \in \mathcal{A}_j(x_k)$, $j = 2, 3$。而 $\Lambda_k = \mathrm{diag}(\lambda_1^{(k)}, \ldots, \lambda_n^{(k)})$ 且

$$\lambda_i^{(k)} = \begin{cases} 0, & \text{如果} \ \ i \in \mathcal{B}(x_k) \ \ \text{或} \ \mathcal{A}_1(x_k) \ \ \text{或} \ \mathcal{A}_2(x_k) \\ (x_i^{(k)} - l_i)/g_i^{(k)}, & \text{如果} \ \ l_i < x_i \le l_i + \epsilon_b \ \ \text{和} \ \ x_i^{(k)} - g_i^{(k)} \le l_i \\ (x_i^{(k)} - u_i)/g_i^{(k)}, & \text{如果} \ \ u_i - \epsilon_b \le x_i < u_i \ \ \text{和} \ \ x_i^{(k)} - g_i^{(k)} \ge u_i \\ 1, & \text{否则} \end{cases}. \tag{4}$$

最后，我们获得 搜索方向为

$$d_k = -(P_1^{(k)} H_k P_1^{(k)} + P_2^{(k)} P_2^{(k)T} + P_3^{(k)} P_3^{(k)T} \Lambda_k) g(x_k). \tag{5}$$

在 [2] 和 [3] 中投影 搜索用来确定步长 α_k 使得

$$\phi_k(\alpha) \le \phi_k(0) + \mu \phi_k'(0) \alpha \tag{6}$$

在 μ 是 $(0, 1/2)$ 的一个常数。 这里， $\phi_k(\alpha) = f(P_\Omega[x_k + \alpha d_k])$, 其中 $\Omega = \{x \in R^n: \ l \le x \le u\}$, P_Ω 是在 Ω 上 的投影 。

在一些不算强 的条件下，可以证明 SLMQN 算法是全局收敛 的, 即从任意初始点开始，迭代 的每个聚点都是 Kuhn-Tucker 点 [4]。

3 软件的使用

SLMQN 是用 FORTRAN 77 按双精度写 的软件 。用户需要计算函数值 F 和它 的梯度值 DF。为了让用户能完全控制这些计算，软件可采用人机交互方式 。程序 slmqn.f 在变量 TASK 的控制下多次被调用 。SLMQN 的调用命令是

```
call slmqn(N,M,X,UNCON,XL,XU,F,DF,WA,LWA,IACT,FACTR,IPRINT,ISAVE,DSAVE,TASK)
```

子程序中所用 参数解释如下：

N 是一个整数变量，变量 的维数，由用户赋值。

M 是一个整数变量， 有限储存BFGS 逆校正 的次数 (一般 $2 \le m \le 10$), 由用户赋值 。

X(n) 是一个双精度数组 。由用户 赋上变量的初始值。

UNCON 是一个逻辑变量，由用户赋值 。如果无约束它为. TRUE. 否则为. FALSE. 。

XL(n), XU(n) 是两个双精度数组，当 UNCON = .FALSE. 时，由用户 赋上变量 的上下界。

F 是一个双精度变量 。如果程序 slmqn.f 以 TASK(1:2) = 'FG' 或 TASK(1:4) = 'FUNC', 那么 F 必须被用户 赋上 f 在点 x 的函数值 。

DF(n) 是一个双精度数组 。如果程序 slmqn.f 以 TASK(1:2) = 'FG' 或 TASK(1:4) = 'GRAD', 那么 DF 必须被用户 赋上 f 在点 x 的梯度值 。

WA(LWA) 是一个长度为 LWA的双精度数组 。LWA 至少应是 $(5+2M)N+2M$。在调用过程中用户不要改变这个数组 的元素 。

IACT(n) 是一个整数数组 。在调用过程中用户不要改变这个数组 的元素 。

FACTR 是一个由用户赋值 的双精度变量 。这是算法终止准则 的一个 参数。如果

$$(F_k - F_{k+1})/\max(|F_{k+1}|, |F_k|, 1) \le \text{FACTR} * \text{EPSMCH} \tag{7}$$

那么迭代终止 。其中 EPSMCH 是机器精度，它由 SLMQN 自动产生 。在双精度为 15 位 的计算机上，FACTR 的典型取值为：对于低近似度， FACTR $=10^{12}$; 对于中等近似度， FACTR $= 10^7$; 对于高近似度， FACTR = 10。

IPRINT 是一个由用户赋值 的整数变量 。IPRINT ≤ 0 时无任何输出，当 IPRINT > 0, 将产生文件 slmqn.dat 储存所输出 的信息 。当 IPRINT = 1 时只输出最后迭代 的一列结果； IPRINT = 2 时每步迭代输出一列结果； IPRINT = 3 时每步迭代都有较详细结果输出 。

ISAVE 是一个维数为 15 的整数工作数组 。用户不要改变这个数组中 的量 。在以 TASK='NEW_X' 返回时， ISAVE 包含一些用户可以查看 的信息 。ISAVE 含有信息 的详细描述请参看 slmqn.f。

DSAVE 是一个维数为 15 的双精度工作数组 。用户不要改变这个数组中 的量 。在以 TASK='NEW_X' 返回时， DSAVE 包含一些用户可以查看 的信息 。DSAVE 含有信息 的详细描述请参看 slmqn.f。

TASK 是一个长度为 60 的符号串 。在初始输入时, 用户必须 赋TASK 为'START'。在子程序 slmqn.f 。以 TASK(1:2) = 'FG' 返回时, 用户必须计算在 x 的返回值时 的函数 F 和 梯度DF。当子程序 slmqn.f 以 TASK(1:5) = 'NEW_X' 返回时，算法 的一步迭代正好结束， F,DF 分别包含 $f(x)$ 和 $\bigtriangledown f(x)$的值 。用户将决定是继续还是终止迭代 。此外,

TASK(1:4) = 'CONV' 意味着 slmqn.f 中 的终止准则被满足；

TASK(1:4) = 'ABNO' 意味着在没有满足终止准则情况下非正常终止 。这时， X 含有最新求得 的近似点， F 和 DF 分别是 $f(x)$ 和 $\bigtriangledown f(x)$;

TASK(1:5) = 'ERROR' 意味着程序在输入 参数中已发现一个 错误。

当 TASK = 'CONV', 'ABNO' 或'ERROR', 变量 TASK 还包含额外 的信息，用户可以把整个 TASK 打印出来 。除了用户在初始步调用 slmqn 或要终止计算外, 其它情况请用户不要改变 TASK的值 。

此外，启动程序 sdriver.f 显示了如何用最简单 的赋值调用 SLMQN 解一个算例 。程序 sdriver.f 和 slmqn.f 可以通过 ftp LSEC.CC.AC.CN 在目录 /pub/yyx/SLMQN 中获得 。与作者联系也可获得这些程序 。

4 程序的终止和出错信息

用户可以在启动程序 (例 sdriver.f) 中适当地方通过令 TASK(1:4) = 'STOP' 终止算法 的执行 。这使得用户可以自己选择终止准则，例如根据投影 梯度, 计算函数值 的次数或者花费 的时间 。如果停机试验 (7) 被满足 的话，或者一个输入错误被发现，或者算法不能继续进行时，那么算法也将终止 。

另外，算法还有一个关于投影 梯度的停机准则 。如果

$$\|P_\Omega[x - \bigtriangledown f(x)] - x\|_\infty = 0,$$

那么算法终止 。这是因为只有投影梯度是非零时，算法才能进行，否则一个稳定点已经获得了 。在这种情况下， TASK 包含一串字符为'CONVERGENCE: NORM OF PROJECTED GRADIENT =0'。

有时也发生线性 搜索无法进行下去 的情形 。在这种情况下，算法也终止， TASK 包含 的信息为 'ABNORMAL TERMINATION IN LINE SEARCH'。

如果在输入信息中发现错误 的话， TASK(1:5) 将含有字符串'ERROR'。在这种情况下，用户可以打印包含在 TASK(1:60) 中详细错误信息 。

5 数值试验

为了检查 SLMQN 优化软件 的性能, 我们从 CUTE 算例软件包 [1] 中选出一些边界约束问题对 SLMQN 进行数值试验 。我们把 SLMQN 与 [6] 中 L-BFGS-B 进行了比较试验 。

所有计算都是在 Sun20 工作站进行 的。当投影梯度 的无穷大范数小于 10^{-5} 时，即

$$\|P_\Omega(x_k - \bigtriangledown f(x_k)) - x_k\|_\infty < 10^{-5} \tag{8}$$

时，算法终止 。表中所用 **N, IT, NF, CPU** 分别表示变量 的维数，迭代次数，函数值计算次数，和程序运行 的时间 (以秒为单位)。

对于 SLMQN 软件，计算梯度 的次数与迭代次数一样，而对于 L-BFGS-B 软件，计算梯度次数和函数值计算次数相同 。因此，对于每个软件我们只列出 IT,NF 和 CPU。决定 有限储存BFGS 逆迭代次数 的参数m 选为 3。Primal, Dual 和 CG 分别表示 L-BFGS-B 软件中用原始方法，对偶方法或共扼梯度法解子问题 。

表 1 显示了边界约束问题 的数值结果 。除了问题 S368, 其它所有问题 的维数都大于或等于 1000。对于有些问题， SLMQN 和 Primal, Dual, CG 需要相同 的迭代，函数 的计算总次数和几乎相同 的时间 。对于问题 TORSION4 和 TORSION6, SLMQN 比 Primal, Dual 和 CG 要好许多 。其原因可能是 Primal, Dual 和 CG 需要许多解子问题 的内迭代 。对于另一些问题， SMQN 或 Primal 或 Dual 显示出一些优点 。概括地说， SLMQN 明显好于 CG, 和 Primal, Dual 不相上下 。

表 2 显示了 SLMQN 对不同 参数m的影响 。SLMQN 对一些维数从 6 到 600的边界约束问题进行了数值试验 。当 m 增大时, 函数值 的计算次数有所下降 。但是时间随着 m 变大而有所增加 。对于 $n > 1000$ 的问题，计算所需时间随着 m 变大而增加得非常快 。因此我们没有列出这些数值结果 。最后我们给出选择 m 的建议 。对于维数大于或等于 1000 的问题， m 可选 2,3 或 5。m 最好小于 10。对于维数小于 1000 的问题，首先 m 选为 3。如果数值结果不太好 的话，那么增加 m, 以不超过 30 为限，直到好 的结果出现为止 。

Table 1: 边界约束问题试验结果

问题	N	SLMQN	Primal	Dual	CG
BDEXP	1000	14/15/0.99	14/15/1.01	14/15/0.67	14/15/1.03
BDEXP	5000	14/15/5.10	14/15/4.97	14/15/3.07	14/15/4.82
JNLBRNGA	5625	191/199/83.6	181/192/67.0	194/207/71.4	255/266/122.7
JNLBRNGB	1024	557/603/40.1	533/569/31.6	665/704/45.3	667/708/57.1
MCCORMCK	1000	15/27/1.23	12/15/0.90	12/15/0.60	12/15/1.06
NONSCOMP	1000	60/122/4.21	41/46/2.72	41/46/1.57	35/40/2.34
OBSTCLAE	5625	201/256/93.5	258/261/103.7	261/271/98.0	307/315/122.2
OBSTCLAE	10000	256/354/224	418/424/305	455/460/305	476/490/327
OBSTCLAL	1024	41/47/2.90	37/49/2.05	37/39/2.42	37/40/2.71
OBSTCLBM	15625	176/210/265	155/161/188	142/149/150	251/271/279
OBSTCLBU	1024	62/89/4.51	42/46/2.45	42/46/2.74	45/48/2.92
S368	100	11/16/2.48	16/19/2.96	16/19/2.94	20/26/4.03
TORSION1	1024	45/52/3.18	55/60/3.45	55/60/3.66	64/66/4.87
TORSION1	5476	164/199/72.3	131/139/47.3	119/129/44.1	171/182/73.9
TORSION1	10000	202/253/169	133/138/90.5	161/169/119	260/281/182
TORSION2	5476	165/194/73.5	173/180/69.0	174/180/60.2	401/453/166.4
TORSION3	5476	66/77/31.3	57/60/16.9	57/60/23.8	75/80/23.6
TORSION4	5476	67/82/28.7	170/173/59.0	177/181/70.0	242/249/76.8
TORSION4	10000	103/131/83.0	273/276/179	285/288/206	417/431/246
TORSION6	5476	43/60/18.4	123/128/39.3	127/129/53.6	142/145/37.4
TORSION6	10000	48/60/38.0	201/205/119.7	204/208/158	247/254/125
TORSION6	14884	70/94/87.3	306/309/300	300/303/365	358/363/286

IT/NF/CPU sec.

Table 2: 改变 m - 边界约束问题

问题	N	SLMQN m = 3	SLMQN m=5	SLMQN m=10	SLMQN m=17
BDEXP	500	14/15/0.48	14/15/0.58	15/16/0.79	15/16/0.75
BIGGS5	6	337/417/0.33	210/253/0.22	212/257/0.22	212/257/0.22
BQPGASIM	50	26/41/0.08	28/48/0.12	28/48/0.17	29/49/0.22
HATFLDC	25	22/25/0.034	21/25/0.042	20/24/0.055	20/24/0.061
JNLBRNGA	529	52/55/1.83	64/69/2.94	51/56/3.48	49/52/4.56
JNLBRNGB	529	321/346/11.5	282/303/13.2	274/295/20.3	277/293/30.9
LINVERSE	199	631/1089/9.74	334/531/6.65	249/408/7.41	318/488/13.8
MAXLIKA	8	946/1875/19.1	755/1568/15.7	812/1671/16.9	812/1671/16.9
MCCORMCK	500	15/25/0.60	15/25/0.71	15/25/0.86	15/25/0.83
NONSCOMP	529	51/64/1.86	50/61/2.34	54/70/3.83	55/73/5.44
OBSTCLAL	529	34/43/1.19	33/40/1.45	34/44/2.18	34/44/2.96
OBSTCLBL	529	39/51/1.38	35/43/1.55	37/46/2.39	34/43/2.80
OBSTCLBM	529	30/42/1.07	31/40/1.39	29/39/1.85	29/39/2.31
OBSTCLBU	529	37/48/1.30	42/58/1.89	37/51/2.40	36/49/3.03
PROBPENL	500	4/35/0.215	4/35/0.212	4/35/0.212	4/35/0.212
S386	100	11/16/2.44	13/22/3.03	14/23/3.25	14/23/3.24
TORSION1	484	28/33/0.911	29/35/1.19	25/30/1.38	25/31/1.62
TORSION2	484	41/47/1.34	31/37/1.28	35/47/2.14	27/35/1.92
TORSION3	484	19/23/0.579	14/18/0.493	15/19/0.638	16/20/0.672
TORSION4	484	19/25/0.607	18/24/0.705	17/25/0.843	17/25/0.815
TORSION6	484	10/14/0.295	9/14/0.291	10/15/0.327	10/15/0.328

IT/NF/CPU sec.

致谢：

作者非常感谢 J. Nocedal 教授向我们提供了 L-BFGS-B 软件 。这个 SLMQN 软件也参考了 L-BFGS-B 软件一些格式来进行设计 的。

参考文献

[1] I. Bongartz, A.R. Conn, N. Gould, and Ph.L. Toint, “CUTE: constrained and unconstrained testing environment”, Research Report, IBM T.J. Watson Research Center, Yorktown, USA, 1994.

[2] J.J.Morè and G.Toraldo, “ On the solution of large quadratic programming problems with bound constraints”, SIAM J. Optimization Vol.1 (1991), 93-113.

[3] Q. Ni, “General large-scale nonlinear programming using sequential quadratic programming methods”, Bayreuther Mathematische Schriften, Vol.45 (1993), 133-236.

[4] Q. Ni and Y. Yuan, “A subspace limited memory quasi-Newton algorithm for solving large-scale nonlinear bound constrained optimization”, Research Report, Institute of Computational Mathematics and Scientific/Engineering Computing, Academia Sinica, Beijing, China, 1994.

[5] J.Nocedal, “ Updating quasi-Newton matrices with limited storage”, Math.Comp. Vol.35, (1980), 773-782.

[6] C. Zhu, R.H. Byrd, P. Lu, and J. Nocedal, “L-BFGS-B Fortran subroutines for large-scale bound constrained optimization”, Report NAM-11, EECS Department, Northwestern University, 1994.