Trust region interior point methods for large sparse l_1 optimization

Ladislav Lukšan, Ctirad Matonoha, Jan Vlček

Institute of Computer Science AS CR, Prague

13th Czech-French-German Conference on Optimization
September 17-21, 2007
Heidelberg, Germany

Outline

1. The l_1 optimization problem
2. How to compute direction vectors
3. Implementation details
4. Numerical experiments
5. Conclusion

6. Trust region methods
7. Computation of the direction vector
8. Numerical comparison
9. Summary

1. The l_1 optimization problem

Introduction

Consider the l_1 optimization problem – minimize the function

$$F(x) = \sum_{i=1}^{m} |f_i(x)|, \tag{1}$$

where

- $f_i : \mathcal{R}^n \to \mathcal{R},\ 0 \leq i \leq m,$ are smooth functions (e.g. twice continuously differentiable on a sufficiently large convex compact set $\mathcal{D}$) depending on n_i variables;
- the function $F(x)$ is partially separable, which means that n and $m = \mathcal{O}(n)$ are large and $n_i = \mathcal{O}(1),\ 0 \leq i \leq m,$ are small.

Equivalent problem

The minimization of F is equivalent to the sparse nonlinear programming problem with $n+m$ variables $x \in \mathcal{R}^n,\ z \in \mathcal{R}^m$:

$$\text{(2)} \qquad \text{minimize} \quad \sum_{i=1}^{m} z_i \quad \text{subject to} \quad -z_i \leq f_i(x) \leq z_i, \quad 1 \leq i \leq m.$$

The necessary first-order (Karush-Kuhn-Tucker) conditions have the form

$$\text{(3)} \qquad \sum_{i=1}^{m} u_i \nabla f_i(x) = 0, \quad z_i = |f_i(x)|, \quad |u_i| \leq 1, \quad \text{and}$$

$$u_i = \frac{f_i(x)}{|f_i(x)|} \quad \text{if} \quad |f_i(x)| > 0$$

where $u_i,\ 1 \leq i \leq m,$ are Lagrange multipliers. This problem can be solved by an arbitrary nonlinear programming method utilizing sparsity:

sequential linear programming, sequential quadratic programming, interior-point, nonsmooth equation

Unconstrained problem

We introduce a trust-region interior-point method that utilizes a special structure of the l_1 optimization problem. Constrained problem (2) is replaced by a sequence of unconstrained problems

$$\text{minimize} \quad B(x,z;\mu) = \sum_{i=1}^{m} z_i - \mu \sum_{i=1}^{m} \log(z_i^2 - f_i^2(x)) \tag{4}$$

with a barrier parameter $0 < \mu \leq \overline{\mu},$ where we assume that

$$z_i > |f_i(x)|,\ 1 \leq i \leq m,$$

and $\mu \to 0$ monotonically. Here

$$B(x,z;\mu) : \mathcal{R}^{n+m} \to \mathcal{R}$$

is a function of $n+m$ variables $x \in \mathcal{R}^n,\ z \in \mathcal{R}^m.$

Iteration process

The interior-point method is a trust-region modification of the Newton method and is iterative, so it generates a sequence of points $x_k \in \mathcal{R}^n,\ k \in \mathcal{N},$ such that

$$x_{k+1} = x_k + \alpha d_k^x, \quad z_{k+1} = z_k + \alpha d_k^z,$$

where $d_k^x,\ d_k^z$ are direction vectors and $\alpha > 0$ is a suitable step size.

In order to compute direction vectors, we proceed from necessary conditions for a minimum of $B(x, z; \mu)$. We obtain a system of $n + m$ nonlinear equations which is solved by the Newton method – this method uses second-order derivatives.

An approximation of the Hessian matrix is computed by gradient differences which can be carried out efficiently if this matrix is sparse.

2. How to compute direction vectors

Necessary conditions for a minimum

Differentiating $B(x,z;\mu) = \sum_{i=1}^m z_i - \mu \sum_{i=1}^m \log(z_i^2 - f_i^2(x))$ we obtain necessary conditions for a minimum:

$$\frac{\partial B(x,z;\mu)}{\partial x} = A(x)u(x,z;\mu) = 0, \tag{5}$$

$$\frac{\partial B(x,z;\mu)}{\partial z} = Z^{-1}f(x) - u(x,z;\mu) = 0, \tag{6}$$

where

$$A(x) = [g_1(x), \ldots, g_m(x)], \quad g_i(x) = \nabla f_i(x), \quad Z = \mathrm{diag}(z_1, \ldots, z_m),$$

$$u(x,z;\mu) = [u_1(x,z_1;\mu), \ldots, u_m(x,z_m;\mu)]^T, \quad u_i(x,z_i;\mu) = \frac{2\mu f_i(x)}{z_i^2 - f_i^2(x)}.$$

System of $n+m$ nonlinear equations (5)-(6) can be solved by the Newton method to obtain increments d_k^x and d_k^z.

Condition $u(x,z;\mu) = Z^{-1}f(x)$

The structure of $B(x,z;\mu)$ allows us to obtain a minimizer $z(x;\mu) \in \mathcal{R}$ of $B(x,z;\mu)$ for a given $x \in \mathcal{R}^n$. The function $B(x,z;\mu)$ (with x fixed) has a unique stationary point which is its global minimizer. This point is characterized by the equations

$$u(x,z;\mu) = Z^{-1}f(x) \quad \Leftrightarrow \quad z_i^2(x;\mu) - f_i^2(x) = 2\mu z_i(x;\mu) \tag{7}$$

which have the solutions

$$z_i(x;\mu) = \mu + \sqrt{\mu^2 + f_i^2(x)}, \quad 1 \le i \le m. \tag{8}$$

Assuming $z = z(x;\mu)$ we denote $B(x;\mu) = B(x,z(x;\mu);\mu)$ and

$$u_i(x;\mu) = \frac{f_i(x)}{z_i(x;\mu)} = \frac{f_i(x)}{\mu + \sqrt{\mu^2 + f_i^2(x)}}, \quad 1 \le i \le m. \tag{9}$$

In this case, the barrier function $B(x;\mu)$ depends only on x. In order to obtain a minimizer $(x,z) \in \mathcal{R}^{n+m}$ of $B(x,z;\mu)$, it suffices to minimize $B(x;\mu)$ over $\mathcal{R}^n$. Note that $B(x;\mu)$ is bounded from below if μ is fixed.

Condition $A(x)u(x;\mu) = 0$

Lemma 1 *It holds*

$$\nabla B(x;\mu) = A(x)u(x;\mu), \tag{10}$$

$$\nabla^2 B(x;\mu) = G(x;\mu) + A(x)V(x;\mu)A^T(x), \tag{11}$$

where

$$G(x;\mu) = \sum_{i=1}^{m} u_i(x;\mu)\nabla^2 f_i(x),$$

$$V(x;\mu) = \mathrm{diag}(v_1(x;\mu),\ldots,v_m(x;\mu)), \quad v_i(x;\mu) = \frac{2\mu}{z_i^2(x;\mu) + f_i^2(x)}.$$

Lemma 2 *Let a vector* $d \in \mathcal{R}^n$ *solve the equation*

$$\nabla^2 B(x;\mu)d = -g(x;\mu), \tag{12}$$

where $g(x;\mu) = \nabla B(x;\mu) \neq 0$. *If the matrix* $G(x;\mu)$ *is positive definite, then* $d^T g(x;\mu) < 0$ *, i.e. the direction vector* d *is descent for* $B(x;\mu)$.

Line-search vs. trust-region

The vector $d \in \mathcal{R}^n$ obtained by solving (12) is descent for $B(x;\mu)$ if the matrix $G(x;\mu)$ is positive definite. Unfortunately, the positive definiteness of this matrix is not assured in a non-convex case, which causes that the standard line-search methods for computing d cannot be used. For this reason, the trust-region methods were developed.

There are two basic possibilities, either a trust-region approach or a line-search strategy with suitable restarts, which eliminate this insufficiency. We have implemented and tested both these possibilities and our tests have shown that the first possibility is more efficient.

Trust-region methods use a direction vector obtained as an approximate minimizer of the quadratic subproblem with a trust region radius Δ.
A computed direction vector $d \equiv d_k^x$ serves for obtaining a new point

$$x_{k+1} = x_k + d \quad (\alpha = 1).$$

3. Implementation details

Quadratic subproblem

The quadratic subproblem has the form

$$\text{(13)} \quad \text{minimize} \quad Q(d) = \frac{1}{2} d^T \nabla^2 B(x;\mu) d + g^T(x;\mu) d \quad \text{s.t.} \quad \|d\| \leq \Delta.$$

Denoting

$$\text{(14)} \quad \rho(d) = \frac{B(x+d;\mu) - B(x;\mu)}{Q(d)} = \frac{\text{actual decrease of } B(x;\mu)}{\text{predicted decrease of } B(x;\mu)},$$

we set

$$x^+ = x \quad \text{if} \quad \rho(d) < \underline{\rho} \quad \text{or} \quad x^+ = x + d \quad \text{if} \quad \rho(d) \geq \underline{\rho}$$

and update the trust region radius in such a way that $\Delta \leq \overline{\Delta}$ and

$$\underline{\beta}\|d\| \leq \Delta^+ \leq \overline{\beta}\|d\| \quad \text{if} \quad \rho(d) < \overline{\rho} \quad \text{or} \quad \Delta \leq \Delta^+ \leq \overline{\gamma}\Delta \quad \text{if} \quad \rho(d) \geq \overline{\rho},$$

where $0 < \underline{\rho} < \overline{\rho} < 1$ and $0 < \underline{\beta} \leq \overline{\beta} < 1 < \overline{\gamma}$.

Direction determination 1 – Moré-Sorensen

We have used two approaches based on direct decompositions of the matrix $\nabla^2 B$, the Moré-Sorensen's optimum step method and the dogleg method of Dennis and Mei.

The optimum step method computes a more accurate solution of (13) by using the Newton method applied to the nonlinear equation

(15) $$\frac{1}{\|d(\lambda)\|} - \frac{1}{\Delta} = 0 \quad \text{where} \quad (\nabla^2 B + \lambda I)d(\lambda) = -g.$$

This system is solved using the Gill-Murray decomposition of the matrix $(\nabla^2 B + \lambda I)$. This way follows from the KKT conditions for (13). Since the Newton method applied to (15) can be unstable, the safeguards (lower and upper bounds to λ) are usually used.

Direction determination 2 – dogleg method

The dogleg method seeks d as a LC of the Cauchy and Newton steps

$$d_C = -(g^T g / g^T \nabla^2 B g) g, \quad d_N = -(\nabla^2 B)^{-1} g.$$

The Newton step is computed by using either

- the sparse Gill-Murray decomposition which has the form

$$\nabla^2 B + E = L D L^T = R^T R,$$

  where E is a positive semidefinite diagonal matrix (which is equal to zero when $\nabla^2 B$ is positive definite), L is a lower triangular matrix, D is a positive definite diagonal matrix and R is an upper triangular matrix; or
- the sparse Bunch-Parlett decomposition which has the form

$$\nabla^2 B = P L M L^T P^T,$$

  where P is a permutation matrix, L is a lower triangular matrix and M is a block-diagonal matrix with 1×1 or 2×2 blocks (which is indefinite when $\nabla^2 B$ is indefinite).

Maximum step length $\overline{\Delta}$

The use of the maximum step length $\overline{\Delta}$ has no theoretical significance but is very useful for practical computations:

- The problem functions can sometimes be evaluated only in a relatively small region (if they contain exponentials) so that the maximum step-length is necessary.
- The problem can be very ill-conditioned far from the solution point, thus large steps are unsuitable.
- If the problem has more local solutions, a suitably chosen maximum step-length can cause a local solution with a lower value of F to be reached.

Therefore, the maximum step-length $\overline{\Delta}$ is a parameter which is most frequently tuned.

Update of μ

A very important part is the update of the barrier parameter μ. There are two requirements which play opposite roles:

1. $\mu \to 0$ should hold since this is the main property of every interior-point method.
2. $\nabla^2 B(x;\mu)$ can be ill-conditioned if μ is too small because

$$\|\nabla^2 B(x;\mu)\| \leq C/\mu \qquad (C \text{ is a constant}).$$

Thus the lower bound $\underline{\mu}$ for μ is used.

We have tested various possibilities for the barrier parameter update including simple geometric sequences which were proved to be unsuitable. Better results were obtained by setting

$$\begin{aligned} \mu^+ &= \max(\underline{\mu}, \|g\|^2) \quad \text{if} \quad \rho(d) \geq \underline{\rho} \quad \text{and} \quad \|g\|^2 \leq \tau\mu, \\ \mu^+ &= \mu \quad \text{otherwise,} \end{aligned}$$

where $0 < \tau < 1$.

4. Numerical experiments

Numerical experiments 1

The primal interior-point method was tested by using two collections of 22 relatively difficult problems with an optional dimension chosen from [Lukšan,Vlček, V767, 1998], which can be downloaded from the web page

www.cs.cas.cz/~luksan/test.html

as TEST 14 and TEST 15. The functions $f_i(x),\ 1 \leq i \leq m,$ serve for defining the objective function

(16)
$$F(x) = \sum_{1 \leq i \leq m} |f_i(x)|.$$

The first set of the tests concerns a comparison of interior-point methods with various *trust-region* and *line-search* [Lukšan,Matonoha,Vlček, V941, 2005] strategies and the bundle variable metric method [Lukšan,Vlček, PJO, 2006]. Medium-size test problems with 200 variables are used. The results of computational experiments are reported in two tables where only summary results (over all 22 test problems) are given.

Columns of tables

Here `M` is the method used: T1 – the dogleg method with the Gill-Murray decomposition, T2 – the dogleg method with the Bunch-Parlett decomposition, T3 – the optimum step method with the Gill-Murray decomposition, L – the line-search method with restarts, B – the bundle variable metric method; `NIT` is the total number of iterations, `NFV` is the total number of function evaluations, `NFG` is the total number of gradient evaluations, `NR` is the total number of restarts, `NL` is the number of problems for which the best known local minimizer was not found (even if the parameter $\overline{\Delta}$ was tuned), `NF` is the number of problems for which no local minimizer was found (either a premature termination occurred or the number of function evaluations exceeded the upper bound), `NT` is the number of problems for which the parameter $\overline{\Delta}$ was tuned (for removing overflows and obtaining the best known local minimum), and `Time` is the total computational time in seconds.

TEST 14, 15 – 22 problems with 200 variables

M	NIT	NFV	NFG	NR	NL	NF	NT	Time
T1 – dogleg GM	2784	3329	23741	1	-	-	4	3.70
T2 – dogleg BP	2392	2755	19912	2	-	1	8	3.19
T3 – optimum GM	3655	4161	32421	4	1	1	7	6.52
L – line-search	5093	12659	30350	1	1	-	6	4.58
B – bundle VM	34079	34111	34111	22	1	1	11	25.72

Table 1: TEST 14 – 22 problems with 200 variables

M	NIT	NFV	NFG	NR	NL	NF	NT	Time
T1 – dogleg GM	3331	4213	18989	17	-	-	6	3.74
T2 – dogleg BP	3170	4027	17452	17	-	1	12	3.68
T3 – optimum GM	5424	6503	31722	11	1	1	10	7.83
L – line-search	8183	20245	52200	36	2	-	9	10.90
B – bundle VM	34499	34745	34745	22	1	-	11	13.14

Table 2: TEST 15 – 22 problems with 200 variables

Numerical experiments 2

The second set of tests concerns a comparison of the interior-point method, realized as the *dogleg* method with the *Gill-Murray* decomposition, with the bundle variable metric method. Large-scale test problems with 1000 variables are used.

The results of computational experiments are given in two tables, where `P` is the problem number, `NIT` is the number of iterations, `NFV` is the number of function evaluations, `NFG` is the number of gradient evaluations, and `F` is the function value reached. The last row of every table contains the summary results including the total computational time in seconds.

The bundle variable metric method was chosen for the comparison since it is based on a quite different principle and can also be used for the large sparse l_1 optimization.

TEST 14 – 22 problems with 1000 variables

	Trust-region interior-point method				Bundle variable metric method			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F
1	1594	1598	6380	0.166502E-09	7819	7842	7842	0.174023E-20
2	415	516	2912	0.106432E-08	127	130	130	0.735523E-17
3	32	33	231	0.604855E-07	89	89	89	0.359364E-14
4	27	39	196	269.499	81	81	81	269.499
5	30	31	186	0.107950E-06	39	39	39	0.122456E-14
6	32	33	462	0.611870E-07	100	100	100	0.110358E-12
7	18	20	171	336.937	211	211	211	336.937
8	18	19	342	761774.	36	39	39	761774.
9	212	259	3834	327.680	6181	6181	6181	327.682
10	970	1176	17460	0.386416E-01	14369	14369	14369	0.740271E-01
11	82	90	498	10.7765	319	319	319	10.7765
12	35	36	144	982.273	115	117	117	982.273
13	27	28	112	0.277182E-06	16	17	17	0.139178E-18
14	1	12	6	0.129382E-08	3	3	3	0.129382E-08
15	202	246	812	1.96106	3948	3957	3957	1.97013
16	161	169	972	0.435729E-15	4505	4556	4556	0.475529E-03
17	484	564	2910	0.165706E-11	441	443	443	0.857271E-06
18	2093	2538	12564	0.105340E-05	1206	1216	1216	0.129694E-03
19	15	16	96	59.5986	182	182	182	59.5986
20	1226	1529	7362	0.154869E-11	7828	7830	7830	0.102202E-04
21	21	22	132	2.13866	29	30	30	2.13866
22	1423	1770	8544	1.00000	337	341	341	1.00000
Σ	9118	10774	66332	Time=42.56	47981	48092	48092	Time=155.67

TEST 15 – 22 problems with 1000 variables

	Trust-region interior-point method				Bundle variable metric method			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F
1	1464	1477	5860	0.123345E-12	359	540	540	0.815757E-08
2	121	181	605	4.00000	453	473	473	0.153343E-07
3	27	31	168	0.775716E-09	114	114	114	0.374913E-08
4	65	76	264	648.232	53	54	54	648.232
5	6	7	42	0.655031E-14	285	285	285	0.422724E-05
6	8	9	126	0.754396E-13	560	560	560	0.649530E-08
7	73	111	296	12029.9	542	650	650	12029.9
8	83	100	252	0.230723E-06	939	942	942	0.380433E-03
9	532	609	3731	2777.75	4428	4429	4429	2780.11
10	103	148	618	658.048	1389	1389	1389	658.048
11	3452	3674	13812	0.821565E-14	411	454	454	0.838373E-09
12	652	773	3918	3117.36	1879	1882	1882	3125.85
13	165	212	996	14808.8	727	728	728	14808.8
14	162	201	1134	566.112	514	514	514	566.112
15	67	93	476	181.926	654	654	654	181.926
16	268	328	1883	66.5333	1376	1376	1376	66.5333
17	122	147	1107	0.146536E-13	9092	9092	9092	0.337978E-08
18	78	89	474	0.619504E-13	3160	3160	3160	0.754900
19	29	31	330	0.382360E-12	15933	15944	15944	0.239244E-08
20	69	86	420	0.131734E-10	1509	1699	1699	0.756975E-08
21	118	195	708	1326.92	425	426	426	1327.95
22	80	112	486	2993.36	9875	9875	9875	2993.37
Σ	7744	8690	37706	Time=30.03	54677	55240	55240	Time=155.90

5. Conclusion

Conclusion

The results introduced in tables indicate the following:

- the trust-region strategies are more efficient than the restarted line-search strategies in connection with the interior-point method for l_1 optimization;
- the trust-region interior-point method T1 (dogleg GM) is less sensitive to the choice of parameters and requires a lower number of iterations and a shorter computational time in comparison with the bundle variable metric method B;
- method T1 also finds the best known local minimum (if l_1 problems have several local solutions) more frequently (see the column `NL` in tables).

We believe that the efficiency of the trust-region interior-point method could be improved by using a better procedure for the barrier parameter update.

6. Trust-region methods

Introduction

Consider a general problem

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

where $F : \mathcal{R}^n \to \mathcal{R}$ is a twice continuously differentiable objective function bounded from below (in the l_1 problem $F \equiv B(x; \mu)$). Basic optimization methods (trust-region as well as line-search methods) generate points $x_i \in \mathcal{R}^n,\ i \in \mathcal{N},$ in such a way that x_1 is arbitrary and

$$x_{i+1} = x_i + \alpha_i d_i, \quad i \in \mathcal{N}, \tag{17}$$

where $d_i \in \mathcal{R}^n$ are direction vectors and $\alpha_i > 0$ are step sizes.

Notation

For a description of trust-region methods we define the quadratic function

$$Q_i(d) = \frac{1}{2} d^T B_i d + g_i^T d$$

which locally approximates the difference $F(x_i + d) - F(x_i)$, the vector

$$\omega_i(d) = (B_i d + g_i)/\|g_i\|$$

for the accuracy of a computed direction, and the number

$$\rho_i(d) = \frac{F(x_i + d) - F(x_i)}{Q_i(d)}$$

for the ratio of actual and predicted decrease of the objective function. Here $g_i = g(x_i) = \nabla F(x_i)$ and $B_i \approx \nabla^2 F(x_i)$ is an approximation of the Hessian matrix at the point $x_i \in \mathcal{R}^n$.

Trust-region methods are based on approximate minimizations of $Q_i(d)$ on the balls $\|d\| \leq \Delta_i$ followed by updates of radii $\Delta_i > 0$.

Description of TR methods

Direction vectors $d_i \in \mathcal{R}^n$ are chosen to satisfy the conditions

$$\|d_i\| \leq \Delta_i, \tag{18}$$

$$\|d_i\| < \Delta_i \quad \Rightarrow \quad \|\omega_i(d_i)\| \leq \overline{\omega}, \tag{19}$$

$$-Q_i(d_i) \geq \underline{\sigma}\|g_i\| \min(\|d_i\|, \|g_i\|/\|B_i\|), \tag{20}$$

where $0 \leq \overline{\omega} < 1$ and $0 < \underline{\sigma} < 1$. Step sizes $\alpha_i \geq 0$ are selected so that

$$\rho_i(d_i) \leq 0 \quad \Rightarrow \quad \alpha_i = 0, \tag{21}$$

$$\rho_i(d_i) > 0 \quad \Rightarrow \quad \alpha_i = 1. \tag{22}$$

Trust-region radii $0 < \Delta_i \leq \overline{\Delta}$ are chosen in such a way that $0 < \Delta_1 \leq \overline{\Delta}$ is arbitrary and

$$\rho_i(d_i) < \underline{\rho} \quad \Rightarrow \quad \underline{\beta}\|d_i\| \leq \Delta_{i+1} \leq \overline{\beta}\|d_i\|, \tag{23}$$

$$\rho_i(d_i) \geq \underline{\rho} \quad \Rightarrow \quad \Delta_i \leq \Delta_{i+1} \leq \overline{\Delta}, \tag{24}$$

where $0 < \underline{\beta} \leq \overline{\beta} < 1$ and $0 < \underline{\rho} < 1$.

Crucial part

A crucial part of each trust-region method is a direction determination. There are various commonly known methods for computing direction vectors satisfying conditions (18)-(20).

How to compute d_i ?

To simplify the notation, the major index i is omitted.

7. Computation of the direction vector

Moré-Sorensen 1983

The most sophisticated method is based on a computation of the optimal locally constrained step. In this case, the vector $d \in \mathcal{R}^n$ is obtained by solving the subproblem

(25) $$\text{minimize} \quad Q(d) = \frac{1}{2} d^T B d + g^T d \quad \text{subject to} \quad \|d\| \leq \Delta.$$

Necessary and sufficient conditions for this solution are

$$\|d\| \leq \Delta, \quad (B + \lambda I)d = -g, \quad B + \lambda I \succeq 0, \quad \lambda \geq 0, \quad \lambda(\Delta - \|d\|) = 0,$$

where λ is a Lagrange multiplier. The MS method is based on solving the nonlinear equation

$$\frac{1}{\|d(\lambda)\|} = \frac{1}{\Delta} \quad \text{with} \quad (B + \lambda I)d(\lambda) + g = 0$$

by the Newton's method using the Choleski decomposition of $B + \lambda I$. This method is very robust but requires 2-3 Choleski decompositions for one direction determination on the average.

Powell 1970, Dennis-Mei 1975

Simpler methods are based on minimization of $Q(d)$ on the two-dimensional subspace containing the Cauchy and Newton steps

$$d_C = -\frac{g^T g}{g^T B g}\, g, \quad d_N = -B^{-1} g.$$

The most popular is the dogleg method where

$$d = d_N \quad \text{if} \quad \|d_N\| \leq \Delta$$

and

$$d = (\Delta / \|d_C\|) d_C \quad \text{if} \quad \|d_C\| \geq \Delta.$$

In the remaining case, d is a combination of d_C and d_N such that $\|d\| = \Delta$. This method requires only one Choleski decomposition for one direction determination.

Steihaug 1983, Toint 1981

If B is not sufficiently small or sparse or explicitly available, then it is either too expensive or not possible to compute its Choleski factorization. In this case, methods based on matrix-vector multiplications are more convenient.

ST is a technique for finding an approximate solution of (25) that does not require the exact solution of a linear system but still produce an improvement on the Cauchy point. This implementation is based on the CG algorithm for solving the linear system $Bd = -g$. We either obtain an unconstrained solution with a sufficient precision or stop on the trust-region boundary (if either a negative curvature is encountered or the constraint is violated). This method is based on the fact that

$$Q(d_{k+1}) < Q(d_k) \quad \text{and} \quad \|d_{k+1}\| > \|d_k\|$$

hold in the subsequent CG iterations if the CG coefficients are positive and no preconditioning is used. For SPD preconditioner C we have

$$\|d_{k+1}\|_C > \|d_k\|_C \quad \text{with} \quad \|d_k\|_C^2 = d_k^T C d_k.$$

Multiple dogleg

The CG steps can be combined with the Newton step $d_N = -B^{-1}g$ in the multiple dogleg method. Let $k \ll n$ (usually $k = 5$) and d_k be a vector obtained after k CG steps of the Steihaug-Toint method. If $\|d_k\| < \Delta$, we use d_k instead of $d_C = d_1$ in the dogleg method.

Preconditioned Steihaug-Toint

There are two possibilities how the Steihaug-Toint method can be preconditioned:

1. To use the norms $\|d_i\|_{C_i}$ (instead of $\|d_i\|$) in (18)–(24), where C_i are preconditioners chosen. This possibility is not always efficient because the norms $\|d_i\|_{C_i},\ i \in \mathcal{N},$ vary considerably in the major iterations and the preconditioners $C_i,\ i \in \mathcal{N},$ can be ill-conditioned.
2. To use the Euclidean norms in (18)–(24) even if arbitrary preconditioners $C_i,\ i \in \mathcal{N},$ are used. In this case, the trust-region can be leaved prematurely and the direction vector obtained can be farther from the optimal locally constrained step than that obtained without preconditioning. This shortcoming is usually compensated by the rapid convergence of the preconditioned CG method.

Our computational experiments indicate that the second way is more efficient in general.

Gould-Lucidi-Roma-Toint 1997

Although the ST method is certainly the most commonly used in trust-region methods, the resulting direction vector may be rather far from the optimal solution even in the unpreconditioned case. This drawback can be overcome by using the Lanczos process. Initially, the CG algorithm is used as in the ST method. At the same time, the Lanczos tridiagonal matrix is constructed from the CG coefficients. If a negative curvature is encountered or the constraint is violated, we switch to the Lanczos process. In this case, $d = Z\tilde{d},$ where $\tilde{d}$ is obtained by solving

(26) $$\text{minimize} \quad \frac{1}{2}\tilde{d}^T T\tilde{d} + \|g\|e_1^T\tilde{d} \quad \text{subject to} \quad \|\tilde{d}\| \leq \Delta.$$

Here $T = Z^T BZ$ (with $Z^T Z = I$) is the Lanczos tridiagonal matrix and e_1 is the first column of the unit matrix. Using a preconditioner C, the preconditioned Lanczos method generates basis such that $Z^T CZ = I.$ Thus we have to use the norms $\|d_i\|_{C_i}$ in (18)–(24), i.e., the first way of preconditioning, which can be inefficient when $C_i,\ i \in \mathcal{N},$ vary considerably in the trust-region iterations or are ill-conditioned.

Shifted Steihaug-Toint

This method applies the ST method to the shifted subproblem

$$\text{(27)} \quad \min \quad \tilde{Q}(d) = Q_{\tilde{\lambda}}(d) = 1/2\, d^T(B + \tilde{\lambda} I)d + g^T d \quad \text{s.t.} \quad \|d\| \leq \Delta.$$

The number $\tilde{\lambda} \geq 0$ approximates λ in MS method. This method combines good properties of the MS and ST methods and can be successfully preconditioned by the second way. The solution is usually closer to the optimal solution than the point obtained by the original ST method.

1. Carry out $k \ll n$ steps of the unpreconditioned Lanczos method to obtain the tridiagonal matrix $T = T_k = Z_k^T B Z_k$.
2. Solve the subproblem

   $$\text{(28)} \qquad \text{minimize} \quad 1/2\, \tilde{d}^T T \tilde{d} + \|g\| e_1^T \tilde{d} \quad \text{subject to} \quad \|\tilde{d}\| \leq \Delta,$$

   using the MS method to obtain the Lagrange multiplier $\tilde{\lambda}$.
3. Apply the (preconditioned) ST method to subproblem (27) to obtain the direction vector $d = d(\tilde{\lambda})$.

Hager 2001 (1)

There are several recently developed techniques for large scale TR subproblems that are not based on conjugate gradients. This method solves (25) with the additional constraint that d is contained in a low-dimensional subspace. They are modified in successive iterations to obtain quadratic convergence to the optimum. We seek vectors $d \in \mathcal{S}$ where $\mathcal{S}$ contains the following vectors:

- The previous iterate. This causes that the value of the objective function can only decrease in consecutive iterations.
- The vector $Bd + g$. It ensures descent if the current iterate does not satisfy the first-order optimality conditions.
- An estimate for an eigenvector of B ass. with the smallest eigenvalue. It will dislodge the iterates from a nonoptimal stationary point.
- The SQP iterate. The convergence is locally quadratic if $\mathcal{S}$ contains the iterate generated by one step of the SQP algorithm applied to (25).

Hager 2001 (2)

- At first, the Lanczos method is used to generate an orthonormal basis for the $k-$dimensional Krylov subspace (usually $k = 10$).
- Problem (25) is reduced to the $k-$dimensional one to obtain an initial iterate.
- An orthonormal basis for the subspace $\mathcal{S}$ is constructed.
- Original problem (25) is reduced to the four-dimensional one.
- A new iterate d is found via this small subproblem.
- The iteration is finished as soon as $\|(B + \lambda I)d + g\|$ with a Lagrange multiplier λ is smaller than some sufficiently small tolerance.

Hager 2001 (3)

The SQP method is equivalent to the Newton's method applied to the nonlinear system

$$(B + \lambda I)d + g = 0, \quad \frac{1}{2}d^T d - \frac{1}{2}\Delta^2 = 0.$$

The Newton iterate can be expressed in the following way:

$$d_{SQP} = d + z, \quad \lambda_{SQP} = \lambda + \nu,$$

where z and ν are solutions of the linear system

$$\begin{aligned} (B + \lambda I)z + d\,\nu &= -\big((B + \lambda I)d + g\big), \\ d^T z &= 0, \end{aligned}$$

which can be solved by preconditioned MINRES or CG methods. The latter case with the incomplete Choleski-type decomposition of the matrix $B + \lambda I$ has shown to be more efficient in practice.

Rojas-Santos-Sorensen 1997, 2000

Another approach for finding the direction vector d is based on the idea of Sorensen. Consider the bordered matrix

$$B_\alpha = \begin{pmatrix} \alpha & g^T \\ g & B \end{pmatrix}$$

where α is a real number and observe that

$$\frac{\alpha}{2} + Q(d) = \frac{1}{2}(1, d^T) B_\alpha \begin{pmatrix} 1 \\ d \end{pmatrix}.$$

Thus there exists a value of α such that we can rewrite problem (25) as

$$\text{(29)} \quad \text{minimize} \quad \frac{1}{2} d_\alpha^T B_\alpha d_\alpha \quad \text{subject to} \quad \|d_\alpha\|^2 \leq 1 + \Delta^2, \quad e_1^T d_\alpha = 1,$$

where $d_\alpha = (1, d^T)^T$ and e_1 is the first canonical unit vector in $\mathcal{R}^{n+1}$. This formulation suggests that we can find the desired solution in terms of an eigenpair of B_α. The resulting algorithm is superlinearly convergent.

8. Numerical comparison

Numerical comparison

The methods (except for RSS) are implemented in the interactive system for universal functional optimization UFO as subroutines for solving trust-region subproblems. They were tested by using two collections of 22 sparse test problems with 1000 and 5000 variables – subroutines TEST14 and TEST15 described in [Lukšan,Vlček, V767, 1998], which can be downloaded from the web page

www.cs.cas.cz/~luksan/test.html.

The results are given in two tables, where NIT is the total number of iterations, NFV is the total number of function evaluations, NFG is the total number of gradient evaluations, NDC is the total number of Choleski-type decompositions (complete for methods MS, DL, MDL and incomplete for methods PH, PST, PSST), NMV is the total number of matrix-vector multiplications, and Time is the total computational time in seconds.

Table 1 – TEST 14

N	Method	NIT	NFV	NFG	NDC	NMV	Time
1000	MS	1911	1952	8724	3331	1952	3.13
	DL	2272	2409	10653	2195	2347	2.94
	MDL	2132	2232	9998	1721	21670	3.17
	ST	3475	4021	17242	0	63016	5.44
	SST	3149	3430	15607	0	75044	5.97
	GLRT	3283	3688	16250	0	64166	5.40
	PH	1958	2002	8975	3930	57887	5.86
	PST	2608	2806	12802	2609	5608	3.30
	PSST	2007	2077	9239	2055	14440	2.97
5000	MS	8177	8273	34781	13861	8272	49.02
	DL	9666	10146	42283	9398	9936	43.37
	MDL	8913	9244	38846	7587	91784	48.05
	ST	16933	19138	84434	0	376576	134.52
	SST	14470	15875	70444	0	444142	146.34
	GLRT	14917	16664	72972	0	377588	132.00
	PH	8657	8869	37372	19652	277547	127.25
	PST	11056	11786	53057	11057	23574	65.82
	PSST	8320	8454	35629	8432	59100	45.57

Table 2 – TEST 15

N	Method	NIT	NFV	NFG	NDC	NMV	Time
1000	MS	1946	9094	9038	3669	2023	5.86
	DL	2420	12291	12106	2274	2573	9.00
	MDL	2204	10586	10420	1844	23139	7.86
	ST	2738	13374	13030	0	53717	11.11
	SST	2676	13024	12755	0	69501	11.39
	GLRT	2645	12831	12547	0	61232	11.30
	PH	1987	9491	9444	6861	84563	11.11
	PST	3277	16484	16118	3278	31234	11.69
	PSST	2269	10791	10613	2446	37528	8.41
5000	MS	7915	33607	33495	14099	8047	89.69
	DL	9607	42498	41958	9299	9963	128.92
	MDL	8660	37668	37308	7689	91054	111.89
	ST	11827	54699	53400	0	307328	232.70
	SST	11228	51497	50333	0	366599	231.94
	GLRT	10897	49463	48508	0	300580	214.74
	PH	8455	36434	36236	20538	281736	182.45
	PST	9360	41524	41130	9361	179166	144.40
	PSST	8634	37163	36881	8915	219801	140.44

Comments

Note that NFG is much greater than NFV in the first table since the Hessian matrices are computed by using gradient differences. At the same time, the problems referred in the second table are the sums of squares having the form

$$F = \frac{1}{2} f^T(x) f(x)$$

and NFV denotes the total number of the vector $f(x)$ evaluations. Since $f(x)$ is used in the expression

$$g(x) = J^T(x) f(x),$$

where $J(x)$ is the Jacobian matrix of $f(x)$, NFG is comparable with NFV in this case.

9. Summary

Summary

The results in the previous tables require several comments. All problems are sparse with a simple sparsity pattern. For this reason, the methods MS, DL, MDL based on complete Choleski-type decompositions (CD) are very efficient, much better than unpreconditioned methods ST, SST, GLRT based on matrix-vector multiplications (MV). Note that the methods PH, RSS are based on a different principle.

- Since TEST 14 contains reasonably conditioned problems, the preconditioned MV methods are competitive with the CD methods.
- On the contrary, TEST 15 contains several very ill-conditioned problems (one of them had to be removed) and thus the CD methods work better than the MV methods.

In general, the CD methods are very efficient for ill-conditioned but reasonably sparse problems but if the problems do not have sufficiently sparse Hessian matrices, then the CD methods can be much worse than the MV methods. The efficiency of the MV methods also strongly depends on a suitable preconditioner.

References

1. Conn A.R., Gould N.I.M., Toint P.L.: ***Trust-region methods***, SIAM, 2000.
2. Fletcher R.: ***Practical Methods of Optimization***, second edition, New York: Wiley, 1987.
3. Lukšan L., Matonoha C., Vlček J.: ***A shifted Steihaug-Toint method for computing a trust-region step***, TR V914, ICS AS CR, 2004.
4. Lukšan L., Matonoha C., Vlček J.: ***Trust-region interior point method for large sparse l_1 optimization***, TR V942, ICS AS CR, 2005.
5. Lukšan L. and Vlček J.: ***Sparse and partially separable test problems for unconstrained and equality constrained optimization***, TR V767, ICS AS CR, 1998.
6. Nocedal J., Wright S.J.: ***Numerical Optimization***, Springer, New York, 1999.

Thank you for your attention!