Adaptive, Limited-Memory BFGS Algorithms for Unconstrained Optimization

Paul T. Boggs (Retired) [1]   Richard H. Byrd [2]

[1]Sandia National Laboratories, Livermore, CA 94551, USA
(paulboggs@comcast.net)

[2]University of Colorado, Boulder, CO, USA
(richard@research.cs.colorado.edu)

U.S. Mexico Workshop on Optimization and Applications

January 4–8, 2016

Sandia National Laboratories
A Department of Energy National Laboratory

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Goal of the Research: An adaptive L-BFGS method

- Limited-Memory BFGS (L-BFGS) has become the workhorse optimization algorithm for large-scale unconstrained problems
- L-BFGS performs erratically as a function of memory size
- No theoretical or practical guidance of how to choose memory size is available

We propose two schemes for developing an adaptive L-BFGS method that varies the memory used at each iteration

Goal of the Research: An adaptive L-BFGS method

- Limited-Memory BFGS (L-BFGS) has become the workhorse optimization algorithm for large-scale unconstrained problems
- L-BFGS performs erratically as a function of memory size
- No theoretical or practical guidance of how to choose memory size is available

We propose two schemes for developing an adaptive L-BFGS method that varies the memory used at each iteration

Outline

Outline

Unconstrained Optimization Problem

$$\min_{x} f(x)$$

where $x \in \mathcal{R}^n$ and f is twice continuously differentiable
Iteration sequence is given by

$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k)$$

where $H \approx \nabla^2 f(x_k)^{-1}$ and α_k is the steplength
L-BFGS provides means to compute $H_k \nabla f(x_k)$ efficiently

L-BFGS

Computation of H_k from H_{k-1}
Relies on the *secant* vectors

$$s_k = x_k - x_{k-1},$$
$$y_k = \nabla f(x_k) - \nabla f(x_{k-1}),$$

that satisfy the secant equation

$$H_k y_k = s_k.$$

L-BFGS

BFGS update is given by

$$H_k = V_k^\mathsf{T} H_{k-1} V_k + \frac{s_k^\mathsf{T} s_k}{s_k^\mathsf{T} y_k}$$

where

$$\rho_k = \frac{1}{s_k^\mathsf{T} y_k}$$

and

$$V_k = I - \rho_k y_k s_k^\mathsf{T}$$

L-BFGS

Suppose we have the last M values of (s, y), say $(s_0, y_0), \ldots (s_{M-1}, y_{M-1})$
Then we use previous formulas to create the 2-loop recursion to compute $H_k \nabla f(x_k)$ as follows

1. Set $q = \nabla f(x_k)$
2. For $i = M-1, M-2, \ldots, 0$
   - $\eta_i = \rho_i s_i^\top q$
   - $q = q - \eta_i y_i$
3. Set $p = H_0 q$
4. for $i = 0, 1, \ldots, M-1$
   - $\beta = \rho_i y_i^\top p$
   - $p = p + (\eta_i - \beta) s_i$
5. Return p

The two-loop recursion algorithm to compute $H_k \nabla f(x_k)$

L-BFGS

Note that the work to apply 2-loop recursion is $2M$ inner products
So total work is $O(Mn)$ operations

Performance of L-BFGS on Some CUTE Problems

Problem	Memory Size									
	5	10	15	20	25	30	35	40	45	50
1	107	80	69	47	50	38	37	37	37	37
2	71	72	71	68	66	67	72	76	77	81
3	211	178	178	176	176	180	169	166	173	174
4	218	176	172	172	173	173	174	172	176	171
5	250	238	253	252	249	251	249	257	257	223
6	177	223	184	215	202	205	199	217	203	203
7	150	110	103	105	101	103	100	104	100	100
8	414	350	316	256	298	282	293	274	276	281
9	235	264	243	245	244	245	240	250	242	235
10	72	85	118	127	112	102	105	105	105	106
11	839	1126	1333	1294	1052	1174	1079	1109	1150	874
12	671	432	440	417	447	551	583	603	612	677
13	268	230	264	277	276	262	284	269	271	271
Total	3683	3564	3744	3651	3446	3633	3584	3639	3679	3433

L-BFGS iterations for given memory size

Performance of L-BFGS on Some CUTE Problems

Problem	Min. Iterations	Max Iterations
1	37	107
2	66	81
3	166	211
4	171	218
5	223	257
6	177	223
7	100	150
8	256	414
9	235	264
10	72	127
11	839	1333
12	417	677
13	230	284

Minimum and maximum number of iterations

Outline

1. Review of L-BFGS
2. Strategy for Memory Size Selection
3. An Efficient Weighted Measure for m
4. Extended Strategies
5. Future Research

Strategy

- Choose maximum memory size M
- Always store the M most recent (s, y) pairs
- At each iteration, choose $m \leq M$ pairs to actually use

Our main contribution is a procedure for choosing m based on some measure of the value of using the most recent m pairs

Strategy

- Choose maximum memory size M
- Always store the M most recent (s, y) pairs
- At each iteration, choose $m \leq M$ pairs to actually use

Our main contribution is a procedure for choosing m based on some measure of the value of using the most recent m pairs

Strategy

Notation: Let $H^{i,j},\ i \geq j$ be the update formed by using pairs

$$(s_{M-i}, y_{M-i}), \ldots, (s_{M-j}, y_{M-j})$$

with initial matrix H_0
What value of m do we use so that $H^{m,1}$ is closest to $\nabla^2 f(x_c)^{-1}$?
One way is to set

$$m^* = \arg\min_m \left\| \left[H^{m,1} - \nabla^2 f(x_c)^{-1} \right] v \right\|$$

for some test vector v

Strategy

- Only info we have on $\nabla^2 f(x_c)^{-1}$ is (s, y) pairs
- Take $v = y_{M-1}$ (the most current information)
- Thus cannot use y_{M-1} to form trial matrix $H^{i,j}$

So, we compute the values

$$e_m = \left\| H^{m,2} y_{M-1} - s_{M-1} \right\|_2^2, \; m = 2, \ldots, M$$

To test the possibility of using only (s_{M-1}, y_{M-1}), we take

$$e_1 = \| H_0 y_{M-1} - s_{M-1} \|_2^2$$

(Usually, $H_0 = \gamma I$ where $\gamma = \frac{s_{M-1}^{\mathsf{T}} y_{M-1}}{y_{M-1}^{\mathsf{T}} y_{M-1}}$)

Strategy

Thus our algorithm is to set

$$m^* = \arg\min_m \{e_m\}$$

Note

- $m^* \geq 1$ so we always use at least the most recent pair
- The step is given by $-H^{m,1}\nabla f(x_c)$

Strategy

Extra work:

- Work for 2-loop recursion using m pairs is $\approx 4nm$ ops
- We do this for each value of $m = 1, \ldots, M$
- Thus total work is $\approx 2nM^2$ ops
- By examining the 2-loop formulas, we can eliminate half the work by computing and saving all of the intermediate q vectors, requiring extra storage of M n-vectors
- The extra work ($\approx nM^2$ ops) will be small for expensive functions

We call our adaptive algorithm
AL-BFGS

Strategy

Extra work:

- Work for 2-loop recursion using m pairs is $\approx 4nm$ ops
- We do this for each value of $m = 1, \ldots, M$
- Thus total work is $\approx 2nM^2$ ops
- By examining the 2-loop formulas, we can eliminate half the work by computing and saving all of the intermediate q vectors, requiring extra storage of M n-vectors
- The extra work ($\approx nM^2$ ops) will be small for expensive functions

We call our adaptive algorithm
AL-BFGS

Numerical Results

Problem	No. Iters	Adaptive/Min	Adaptive/Max
1	51	1.5	0.58621
2	61	1.0702	0.84722
3	177	1.1062	0.86765
4	178	1.092	0.85577
5	252	1.1721	**1.004**
6	204	1.193	0.95327
7	106	1.0928	0.74126
8	256	1.1082	0.67905
9	252	1.115	0.99213
10	40	**0.85106**	0.47059
11	583	**0.76812**	0.49575
12	231	**0.71739**	0.42153
13	150	**0.88235**	0.69124
Total/Avg	**2541**	1.0514	0.72105

Iterations and ratios of iterations of AL-BFGS to minimum and to maximum number of iterations by for L-BFGS

Numerical Results

- We used $M = 50$
- Significant improvement:
  - Only 1 problem worse than max
  - 4 problems better than min
  - Total iterations of 2541 is about 900 fewer than for L-BFGS

Outline

Motivation

- Want to use the *compact form* of the update formulas for H and B, where $B^{-1} = H$
- Recall that the number of ops to use AL-BFGS is $\approx 2nM^2$ with additional storage of M vectors of length n
- We can reduce this to $O(M^3) + O(Mn)$ ops with $O(M^2)$ additional storage
- Still require $O(mn)$ ops to compute the search direction

New Measure

Consider the weighted error measure

$$\begin{aligned} e_m &= \left\| (B^{m,2})^{1/2} \left[H^{m,2} y_{M-1} - s_{M-1} \right] \right\|_2^2, \; m = 2, \ldots, M \\ &= s_{M-1}^{\mathsf{T}} B^{m,2} s_{M-1} - 2 s_{M-1}^{\mathsf{T}} y_{M-1} + y_{M-1}^{\mathsf{T}} H^{m,2} y_{M-1} \end{aligned}$$

As before, we take

$$e_1 = \left\| B_0^{1/2} \left[H_0 y_{M-1} - s_{M-1} \right] \right\|_2^2$$

- Using the compact form, we can compute $s_{M-1}^{\mathsf{T}} B^{m,2} s_{M-1}$ and $y_{M-1}^{\mathsf{T}} H^{m,2} y_{M-1}$ efficiently for any m
- By saving work at each m, we can efficiently compute quantities for $m + 1$

Initial Matrix H_0

- Formulas are only efficient if H_0 is of the form γI
- Usual value of γ minimizes $\|\gamma y_{M-1} - s_{M-1}\|_2^2$
- We are using different norm, so we take γ to minimize $\left\|\gamma^{1/2} y_{M-1} - \gamma^{-1/2} s_{M-1}\right\|_2^2$, i.e., we take

$$\gamma = \sqrt{\frac{s_{M-1}^\mathsf{T} s_{M-1}}{y_{M-1}^\mathsf{T} y_{M-1}}}$$

Using this value of γ makes a significant difference

We call this version A_WL-BFGS

Initial Matrix H_0

- Formulas are only efficient if H_0 is of the form γI
- Usual value of γ minimizes $\|\gamma y_{M-1} - s_{M-1}\|_2^2$
- We are using different norm, so we take γ to minimize $\left\|\gamma^{1/2} y_{M-1} - \gamma^{-1/2} s_{M-1}\right\|_2^2$, i.e., we take

$$\gamma = \sqrt{\frac{s_{M-1}^{\mathsf{T}} s_{M-1}}{y_{M-1}^{\mathsf{T}} y_{M-1}}}$$

Using this value of γ makes a significant difference

We call this version A_WL-BFGS

Numerical Results

Problem	No. Iters	Adaptive/Min	Adaptive/Max
1	37	1.0882	0.42529
2	62	1.0877	0.86111
3	160	1	0.78431
4	160	**0.9816**	0.76923
5	258	1.2	**1.0279**
6	164	**0.95906**	0.76636
7	94	**0.96907**	0.65734
8	234	1.013	0.62069
9	215	**0.95133**	0.84646
10	40	**0.85106**	0.47059
11	249	**0.32806**	0.21173
12	250	**0.7764**	0.4562
13	198	1.1647	0.91244
Total/Avg	**2121**	**0.95156**	0.68408

Iterations and ratios of iterations of A_WL-BFGS to minimum and to maximum number of iterations by for L-BFGS

Numerical Results

- Much better results than for AL-BFGS
- Only 1 problem worse than max for L-BFGS
- 7 problems better than min for L-BFGS and we average better than the min
- Total iterations 2121 for A_WL-BFGS vs 2541 for AL-BFGS
- If we use classical value of γ we only do marginally better than for AL-BFGS

Outline

Extended Strategy

- So far we have used only used 1 (s, y) pair to determine m
- Test more by saving previous e vectors and combining them to create new measure
- Sometimes helped, but was not consistent (but it's cheap)

Alternate Strategy

Create composite test pair:

$$u = \sum_{i=M}^{M-p} s_i \tag{1}$$

$$v = \sum_{i=M}^{M-p} y_i \tag{2}$$

and consider the metric

$$e_m = \left\| H^{m,1} v - u \right\|^2, m = 1, ..., M.$$

For this metric, we take $\gamma = (u^\mathsf{T} v)/(v^\mathsf{T} v)$. We also compute

$$e_0 = \| H_0 v - u \|^2 = \| \gamma v - u \|^2 ,$$

thus allowing us to check for not using any of the stored pairs.

Results Using AL-BFGS and Alternate Strategy

Problem	Number of Combined Pairs Used							
	2	3	4	5	6	7	8	9
1	43	36	38	41	43	43	47	49
2	70	63	71	76	70	70	72	72
3	169	166	167	162	157	172	169	172
4	161	158	164	167	165	169	165	173
5	240	217	198	216	238	222	242	230
6	172	157	177	184	189	201	192	200
7	99	92	97	95	97	93	93	95
8	251	241	248	256	274	262	264	258
9	238	221	214	219	221	210	206	209
10	53	46	47	48	43	31	36	32
11	593	540	482	445	436	465	495	430
12	232	113	302	259	217	169	203	154
13	150	144	92	83	155	124	81	53
Total Iters	2471	2194	2297	2251	2305	2231	2265	2127

Iterations for each problem using AL-BFGS and the indicated number of combined pairs used. Compare to 2541 total iterations for original version of AL-BFGS.

Results Using A_WL-BFGS and Alternate Strategy

Problem	Number of Combined Pairs Used							
	2	3	4	5	6	7	8	9
1	29	29	33	35	36	35	34	36
2	88	88	69	78	75	65	55	60
3	157	153	154	153	155	157	155	157
4	153	149	149	149	148	147	148	149
5	236	228	227	228	228	229	229	228
6	192	183	183	183	183	184	184	187
7	92	91	92	95	92	91	91	91
8	284	252	233	230	232	216	214	242
9	207	220	216	205	217	205	205	218
10	55	62	70	81	64	50	67	66
11	234	209	214	221	250	209	244	270
12	256	222	299	191	220	200	199	124
13	187	197	197	205	203	177	208	195
Total Iters	2170	2083	2136	2054	2103	1965	2033	2023

Iterations for each problem using A_WL-BFGS and the indicated number of combined pairs. Compare to 2121 iterations for original version of A_WL-BFGS.

Remarks

- Reasonable, but not dramatic improvements
- Suggests that there may be other effective means of using additional available information to form measures
- We considered another, but more expensive, measure that appears promising, but is not completely tested

Outline

Future Research Topics

- Are there any other useful error measures?
- Further investigate combining several e vectors
- Exploit possibility of parallelism
- Explore possibility of dynamically changing M
- Compact form developed to extend L-BFGS to bound constrained problems. Can we combine these to create an adaptive method for bound constraints

¡Gracias!