CS726, Project, 2008-12-15, Nathanael Fillmore (nathanae)

# 1 Problem formulation

Yamashita [6] aims to solve the following problem:

$$\begin{array}{ll} \min_x & f(x) \\ \text{s.t.} & x \in \mathbb{R}^n \\ \text{where} & f \text{ is twice continuously differentiable} \\ & n \text{ is large, and} \\ & \nabla^2 f(x) \text{ is sparse.} \end{array} \tag{1}$$

Since $n$ is large, basic quasi-Newton methods will not work; existing limited-memory quasi-Newton methods, on the other hand, have slow convergence properties or other disadvantages (see §5 below). Yamashita seeks a quasi-Newton method that achieves fast convergence with limited memory.

# 2 Method

Yamashita's novel method for solving problem (1) is the *matrix completion quasi-Newton* (MCQN) algorithm. In this section, we detail the algorithm; in the next section we derive and motivate it.

- Initially:
  - Choose a starting point $x_0$.
  - Let $E \subseteq V \times V = \{1, \dots, n\} \times \{1, \dots, n\}$ be the sparsity pattern of the true Hessian $\nabla^2 f$.
  - Choose $F$ to be a superset of $E$ such that the graph $G = (V, \bar{F})$ is chordal, where $\bar{F}$ is $F$ without self-edges, i.e. $\bar{F} = F \backslash \{(i,i) \mid i = 1, \dots, n\}$. We also require, by analogy with the true sparsity pattern, that $(i,i) \in F$ for all $i \in V$ and $(i,j) \in F \iff (j,i) \in F$.
  - Choose $\{C_r \mid r = 1, \dots, l\}$ to be a family of maximum cliques of $G$.
  - Choose an initial approximate inverse Hessian $H_0 \in \mathbb{R}^{n \times n}$ so that $H_0$ is positive definite and $(H_0^{-1})_{ij} = 0$ for all $(i,j) \notin F$.
- Repeat for $k = 0, 1, 2, \dots$ until convergence:
  - Let $x_{k+1} = x_k - H_k \nabla f(x_k)$.
  - Obtain $\bar{H}_{ij}$ for $(i,j) \in F$ by a standard quasi-Newton update, e.g. DFP or BFGS.
  - Obtain $H_{k+1}$ by applying the "sparse clique-factorization formula" to $\bar{H}$, as follows.
    First:
    - Let $S_r = C_r \backslash (C_{r+1} \cup C_{r+2} \cup \cdots \cup C_l)$, for $r = 1, \dots, l$.
    - Let $U_r = C_r \cap (C_{r+1} \cup C_{r+2} \cup \cdots \cup C_l)$, for $r = 1, \dots, l$.
    - For $r = 1, \dots, l-1$, let

$$(P_r)_{ij} = \begin{cases} 1 & i = j \\ (\bar{H}^{-1}_{U_r U_r} \bar{H}_{U_r S_r})_{ij} & (i,j) \in U_r \times S_r \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

* For $r = 1, \ldots, l$, let

$$Q_r = \begin{cases} \bar{H}_{S_r S_r} - \bar{H}_{S_r U_r} \bar{H}_{U_r U_r}^{-1} \bar{H}_{U_r S_r} & r \leq l-1 \\ \bar{H}_{S_r S_r} & r = 1 \end{cases} \tag{3}$$

* Let

$$Q_{ij} = \begin{cases} (Q_r)_{ij} & (i,j) \in S_r \times S_r, r = 1, \ldots, l \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Now we can compute $H_{k+1}$. Let[1]

$$H_{k+1} = P_1^T P_2^T \ldots P_{l-1}^T Q P_{l-1} \ldots P_2 P_1 \tag{5}$$

# 3 Derivation

In this section, we derive and motivate Yamashita's MCQN update. For concreteness, we start from the standard DFP update, although another quasi-Newton update such as BFGS could be used instead.

The DFP method's inverse Hessian approximation $H_{k+1}^{DFP} = H_k - \frac{(H_k y_k)(H_k y_k)^T}{y_k^T H_k y_k} + \frac{s_k s_k^T}{s_k^T y_k}$ was shown by [1] to be unique minimizer of the following strictly convex problem:

$$\begin{array}{ll} \min_H & \psi(H_k^{-1/2} H H_k^{-1/2}) \\ \text{s.t.} & H y_k = s_k, H = H^T, H \succ 0 \end{array} \tag{6}$$

where

$$\psi(A) = \operatorname{trace}(A) - \ln\det(A) \tag{7}$$

$$s_k = x_{k+1} - x_k \tag{8}$$

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k) \tag{9}$$

$$H \succ 0 \equiv H \text{ is positive definite} \tag{10}$$

Unfortunately, the solution $H^{DFP}$ is intractable, since it is dense and $n$ is large.

But we know that the true Hessian is sparse, so Yamashita adds a sparsity constraint to the subproblem:

$$\begin{array}{ll} \min_H & \psi(H_k^{-1/2} H H_k^{-1/2}) \\ \text{s.t.} & H y_k = s_k, H = H^T, H \succ 0 \\ & (H^{-1})_{ij} = 0 \text{ for } (i,j) \notin F \end{array} \tag{11}$$

Here, $F$ is defined as in §2 above to be a superset of the true Hessian's sparsity pattern such that the corresponding graph is chordal. Ideally, $F$ will be close to the true sparsity pattern, so that $H_{k+1}^{-1}$ will be approximately as sparse—and therefore tractable—as the true Hessian is.

Unfortunately, problem (11) does not lead to an efficient closed-form update formula for $H_{k+1}$. To find an efficient update, we first replace the secant constraint $H y_k = s_k$ with the constraint that $H_{ij} = H_{ij}^{DFP}$ for $(i,j) \in F$. Since $H^{DFP}$ satisfies the secant constraint, this change can be viewed as a relaxation, not a wholesale abandonment, of the secant constraint. Alternatively, the new constraint can be thought of as

[1] Note that comparison with [2], p. 258, reveals a typo in Yamashita's Eq. (7), p. 7.

finding the best $H$ in the region defined by $Hy_k = s_k$, and then "projecting" into the region defined by $(H^{-1})_{ij} = 0$ for $(i,j) \notin F$; see Yamashita's Figure 4. The new problem is as follows:

$$\begin{array}{ll} \min_H & \psi(H_k^{-1/2} H H_k^{-1/2}) \\ \text{s.t.} & H = H^T, H \succ 0 \\ & H_{ij} = H_{ij}^{DFP} \text{ for } (i,j) \in F \\ & (H^{-1})_{ij} = 0 \text{ for } (i,j) \notin F \end{array} \tag{12}$$

This problem is still difficult to solve directly, but Yamashita shows (pp. 10–11) that it is equivalent to finding a "maximum-determinant positive definite matrix completion" of $H_{ij}^{DFP}$, $(i,j) \in F$, that is, to the following problem:

$$\begin{array}{ll} \max_H & \det(H) \\ \text{s.t.} & H = H^T, H \succ 0 \\ & H_{ij} = H_{ij}^{DFP} \text{ for } (i,j) \in F \end{array} \tag{13}$$

As shown in [2], this type of problem has the unique closed-form solution $H_{k+1}$ given by Eq. (5) above, and this solution is sparse.

# 4 Convergence

Yamashita proves (his §5) that the MCQN algorithm has superlinear convergence, as follows:

**Theorem 4.1.** *Let $x^*$ be a solution of problem (1), and let $\mathcal{N}$ be a neighborhood of $x^*$, i.e. $\mathcal{N} = \{x \in \mathbb{R}^n \mid ||x - x^*|| \leq b\}$ for some $b > 0$. Assume:*

1. *The objective $f$ is twice continuously differentiable in $\mathcal{N}$.*
2. *The inverse Hessian is positive definite and bounded; that is, for some constants $m > 0$, $M > 0$,*

$$m||z||^2 \leq z^T (\nabla^2 f(x))^{-1} z \leq M||z||^2 \quad \forall z \in \mathbb{R}^n \tag{14}$$

   *for all $x \in \mathcal{N}$.*

3. *The starting point $x_0$ is "close enough" to the solution $x^*$; that is, $||x_0 - x^*|| \leq \tau_x$ and $||H_0 - \nabla^2 f(x^*)^{-1}|| \leq \tau_H$ hold for sufficiently small $\tau_x, \tau_H > 0$.*

*Then the sequence $\{x_k\}$ generated by the MCQN update with the DFP method converges to $x^*$ superlinearly.*

# 5 Evaluation

The MCQN algorithm can be evaluated by comparison to its competitors. We discuss limited-memory BFGS (L-BFGS) ([4] and [5] §7.2) and partially separable BFGS (PS-BFGS) ([3] and [5] §7.4). Both are well-known quasi-Newton methods that require only a limited amount of storage.

Compared to L-BFGS, MCQN's main—and significant—theoretical advantage is that MCQN has superlinear convergence under the right conditions, while L-BFGS has only linear convergence. This is because L-BFGS throws away information about the Hessian; MCQN maintains a much better approximation to the Hessian.

PS-BFGS, like MCQN, aims to approximate the Hessian accurately; this allows PS-BFGS to converge quickly. However, to achieve its parsimonious and accurate approximation, PS-BFGS requires the objective function to be partially separable into smaller components. Hence PS-BFGS does not apply as generally as MCQN. Additionally, unless each component is convex, PS-BFGS's approximation to the Hessian sometimes fails to be positive definite away from the solution; avoiding or recovering from this problem can affect performance.

On the other hand, MCQN is not perfect, even theoretically. First, MCQN's approximate Hessian mimics the sparsity pattern not of the true Hessian, but rather of the chordal graph $G = (V, \bar{F})$. In some cases, for example if the true Hessian's sparsity pattern has large cycles, this approximation could be much denser than the true Hessian and require excessive memory. Neither L-BFGS nor PS-BFGS suffer from this particular problem. Indeed, L-BFGS requires only an arbitrary amount of storage, regardless of the particular problem (unless the initial approximation $H_0$ is chosen to mimic the true Hessian's sparsity).

Second, the chordal graph leads to another difficulty with MCQN: how should we compute it? The problem of obtaining a minimal chordal graph from another graph is NP complete in general. In specific cases, for example if the Hessian is tridiagonal, the problem is tractable—though of course taking advantage of specific structure requires specific intervention by the user. In general, we may have to settle for an approximation to the the minimal chordal graph. This leads to a tradeoff: we can spend more time to obtain a good approximation of the minimal chordal graph and a sparser approximate Hessian, or we can settle for a rougher approximation to the minimal chordal graph and hope that the approximate Hessian is not too dense.

We can also evaluate the MCQN algorithm empirically. Yamashita presents results of three problems, evaluated using BFGS, L-BFGS, MCQN with DFP, and MCQN with BFGS. (Unfortunately PS-BFGS is not tested.) The problems are (a) TRIDIA (convex quadratic), (b) the "chained Rosenbrock problem" (nonconvex, nonlinear), and (c) the "boundary value problem" (nonconvex, nonlinear), all with $n = 10, 100, 1000,$ and $10000$; see p. 28 for formulas. On these handpicked examples, MCQN with BFGS requires fewer iterations than L-BFGS for all $n$, and in some cases the difference is quite large; for example, on problem (c) with $n = 1000$, L-BFGS requires 3117 iterations while MCQN with BFGS requires only 54 iterations. MCQN with BFGS even beats standard BFGS in all cases except problems (a) and (c) with $n = 10$, and again, sometimes the difference is substantial; for example, on problem (c) with $n = 10000$, BFGS requires 571 iterations (vs. 54). The reason for both these results seems to be that MCQN has a better approximation to the Hessian even than BFGS does; in particular, MCQN's approximation reflects the true sparsity pattern more closely than BFGS's approximation does, a fact that becomes more significant as $n$ (and the sparsity) increases.

On the other hand, the empirical results show that MCQN with DFP sometimes performs quite poorly. For example, on problem (a), $n = 10000$, MCQN with DFP requires 11626 iterations vs. 1191 for L-BFGS and 528 for MCQN with BFGS. This result is consistent with the consensus about quasi-Newton updates that BFGS generally has better numerical properties than DFP. As mentioned above, Yamashita does not present results using PS-BFGS. He does note that PS-BFGS solves problem (a) quickly even for large $n$. In the future, further tests that pit PS-BFGS against MCQN would be interesting, as would tests on problems that may be less well-suited to the MCQN algorithms's strengths.

# References

[1] R. Fletcher. A new variational result for quasi-newton formulae. *SIAM Journal on Optimization*, 1(1):18–21, 1991.

[2] Mituhiro Fukuda, Masakazu Kojima, Kazuo Murota, and Kazuhide Nakata. Exploiting sparsity in semidefinite programming via matrix completion I: General framework. *SIAM Journal on Optimization*, 11:647–674, 2000.

[3] A. Griewank and Ph.L. Toint. Updating quasi-newton matrices with limited storage. In: Powell, M.J.D. (ed.) Nonlinear Optimization 1981, 301–312. Academic, London, 1982.

[4] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[5] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, second edition, 2006.

[6] Nobuo Yamashita. Sparse quasi-newton updates with positive definite matrix completion. *Math. Program.*, 115(1):1–30, 2008.