Efficient Quasi-Newton Proximal Method for Large Scale Sparse Optimization

Xiaocheng Tang
Department of Industrial and Systems Engineering
Lehigh University
Bethlehem, PA 18015
xct@lehigh.edu

Katya Scheinberg
Department of Industrial and Systems Engineering
Lehigh University
Bethlehem, PA 18015
katyas@lehigh.edu

1 Introduction

In this paper we propose an efficient, general purpose algorithmic approach and efficient implementation for the following, well-studied, convex optimization problem:

$$\min_{x\in\mathbb{R}^n} F(x) \equiv f(x) + g(x) \tag{1.1}$$

where $f, g : \mathbb{R}^n \to \mathbb{R}$ are both convex functions such that f is twice differentiable, and $g(x)$ is such that the problem of minimizing $F(x)$: $\min_{x\in\mathbb{R}^n} \left\{g(x) + \frac{1}{2}\|x - z\|_H^2\right\}$ is easy to solve, possibly approximately, for any $z \in \mathbb{R}^n$ and some class of positive definite matrices H, and where $\|y\|_H^2 = y^\top Hy$. While our algorithmic approach is general, our implementation presented here is developed for the sparse optimization, where $g(x) = \lambda\|x\|_1$ for some $\lambda > 0$.

Problems of form (1.1) with $g(x) = \lambda\|x\|_1$ have been the focus of much research lately in the fields of signal processing and machine learning. This form encompasses a variety of machine learning models, in which feature selection is desirable, such as sparse logistic regression [19, 20, 16], sparse inverse covariance selection [8, 12, 15] and unconstrained Lasso [17], etc. These settings often present common difficulties to optimization algorithms due to their large scale. During the past decade most optimization effort aimed at these problems focused on development of efficient first-order methods (see, e.g., [10, 1, 18, 20, 7, 6, 15, 14]). These methods enjoy low per-iteration complexity, but typically have low local convergence rates. Their performance is often hampered by small step sizes.

Recently several new methods were proposed for sparse optimization which make careful use of second-order information [8, 20, 12, 3]. These methods explore the following special properties of the sparse problems: at optimality many of the elements of x are expected to equal 0, hence methods which explore active set-like approaches can benefit from small sizes of subproblems. Whenever the subproblems are not small, these new methods exploit the idea that the subproblems do not need to be solved accurately. In particular several successful methods employ coordinate descent to approximately solve the subproblems. Other approaches to solve Lasso subproblem were considered in [3], but none generally outperform coordinate descent. [20] proposes a specialized GLMNET [7] implementation for sparse logistic regression, where coordinate descent method is applied to the unconstrained Lasso subproblem constructed using the Hessian of $f(x)$ – the smooth component of the objective $F(x)$. Two main improvements increase efficiency of GLMNET for larger problems – exploitation of the special structure of the Hessian to reduce the complexity of each coordinate step so that it is linear in the number of training instances, and a two-level shrinking scheme proposed to focus the minimization on smaller subproblems. Similar ideas are used in [8] in a specialized algorithm called QUIC for sparse inverse covariance selection.

We propose an efficient *general purpose* algorithm that does not rely on the Hessian structure, is backed by theoretical analysis and convergence rates [**?**], and yet outperforms the state-of-the-art specialized methods such as QUIC and GLMNET. As these two and other methods, mentioned above, we consider the following general framework:

- At each iteration $f(x)$ is approximated, near the current iterate x^k, by a convex quadratic function $q(x)$.
- Then an algorithm is applied to optimize (approximately) the function $q(x) + g(x)$, to compute a trial point.
- The trial point is accepted as the new iterate if it satisfies some sufficient decrease condition.

- Otherwise, a different model $q(x)$ may be computed, or a line search applied to compute a new trial point.

We make use of similar ideas as in [20] and [8], but further improve upon them to obtain efficient general schemes, which apply beyond the special cases of sparse logistic regression and covariance selection and for which convergence rates can be established. In particular, we use limited memory BFGS Hessian approximations [11] because the low rank form of the Hessian estimates can be exploited to reduce the coordinate descent step complexity to a constant. Instead of line search procedure, we update a prox term of our quadratic model, which allows us to extend global convergence bounds of proximal gradient methods to the case of proximal quasi-Newton methods. The criteria for accepting a new iteration is based on sufficient decrease condition (much like in trust region methods, and unlike that in proximal gradient methods). We show that proximal method based on sufficient decrease condition leads to an improvement in performance and robustness of the algorithm compared to the line search approach.

To keep the current article short and focused, we leave the theoretical analysis of a general inexact quasi-Newton proximal scheme out of this paper. Our contributions covered by this paper are as follows

- We replace the exact Hessian computation by LBFGS Hessian approximations and exploit the low-rank model Hessian structure within coordinate descent approach to solve the subproblems.
- We use a different working set selection strategy than those used in [8]. We choose to select a working set by observing the largest violations in the dual constraints. Similar technique has been successfully used in optimization for many decades, for instance in Linear Programming [2] and in SVM [13]. We do not include active set strategies into our convergence analysis, but treat is a heuristic at this point.
- We replace line search approach with a updates of proximal parameter, and use sufficient decrease condition for step acceptance.
- We present an efficient C++ implementations based on our algorithmic ideas and show that it can be superior to QUIC and GLMNET, while not exploiting specific structure of the smooth component of the objective function, hence being more general than these methods.

2 Optimization Algorithm

In this section we briefly describe the specifics of the algorithm that we implemented and that takes advantage of some second order information while maintaining low complexity of subproblem optimization steps. The algorithm is designed to solve problems of the form 1.1 with $g(x) = \lambda\|x\|_1$. Here we note, again, that the algorithm does not use any special structure of the smooth part of the objective, $f(x)$.

At iteration k a step d_k is obtained, approximately, as follows

$$d_k = \arg\min_d\{\nabla f(x^k)^T d + d^T H_k d + \lambda\|x^k + d\|_1;\ \text{s.t. } d_i = 0, \forall i \in \mathcal{A}_k\} \tag{2.1}$$

with $H_k = B_k + \frac{1}{2\mu}I$ - a positive definite matrix and $\mathcal{A}_k$ - a set of coordinates fixed at the current iteration.

The positive definite matrix B_k is computed by a limited memory BFGS approach. In particular, we use a specific form of the low-rank Hessian estimate, (see e.g. [4, 11]),

$$B_k = \gamma_k I - QRQ^T = \gamma_k I - Q\hat{Q} \quad \text{with } \hat{Q} = RQ^T, \tag{2.2}$$

where Q, γ_k and R are defined below,

$$Q = [\gamma_k S_k \quad T_k]\,,\ R = \begin{bmatrix}\gamma_k S_k^T S_k & M_k \\ M_k^T & -D_k\end{bmatrix}^{-1},\ \gamma_k = \frac{t_{k-1}^T t_{k-1}}{t_{k-1}^T s_{k-1}} \tag{2.3}$$

S_k and T_k are the $n\times m$ matrices with column coming from vector pairs $\{s_i, t_i\}_{i=k-m}^{k-1}$ satisfy $s_i^T t_i > 0$, $s_i = x^{i+1} - x_i$ and $t_i = \nabla f(x^{i+1}) - \nabla f(x^i)$, with m being a small integer which defines the number of latest BFGS updates that are "remembered" at any given iteration (we used $10 - 20$).

M_k and D_k are the $m \times m$ matrices which are defined by inner products of select vector pairs $\{s_i, t_j\}_{i,j=k-m}^{k-1}$

2.1 Greedy Active-set Selection $\mathcal{A}_k(\mathcal{I}_k)$

An active-set selection strategy maintains a sequence of sets of indices $\mathcal{A}_k$ that iteratively estimates the optimal active set $\mathcal{A}^*$ which contains indices of zero entries in the optimal solution x^* of (1.1).The complement set of $\mathcal{A}_k$ is $\mathcal{I}_k =$

$\{i \in \mathcal{P} \mid i \notin \mathcal{A}_k\}$. Let $(\partial F(x^k))_i$ be the i-th component of the subgradient of $F(x)$ at x^k. We define two sets,

$$\mathcal{I}_k^{(1)} = \{i \in \mathcal{P} \mid (\partial F(x^k))_i \neq 0\},\ \mathcal{I}_k^{(2)} = \{i \in \mathcal{P} \mid (x^k)_i \neq 0\} \tag{2.4}$$

We select $\mathcal{I}_k$ to include the entire set $\mathcal{I}_k^{(2)}$ and a small subset of indices from $\mathcal{I}_k^{(1)}$ for which $\partial F(x^k))_i$ is the largest. In contrast, the strategy used by [20] and [8] select the entire set $\mathcal{I}_k^{(1)}$, which results in a larger size of subproblems (2.1) at the early stages of the algorithm.

2.2 Solving the inner problem via coordinate descent

We apply coordinate descent method to the piecewise quadratic subproblem (2.1) to obtain the direction d_k and exploit the special structure of H_k. Suppose j-th coordinate in d is updated, hence $d' = d + ze_j$ (e_j is the j-th vector of the identity). Then z is obtained by solving the following one-dimensional problem

$$\min_z (H_k)_{jj} z^2 + ((\nabla f(x^k))_j + 2(H_k d)_j)z + \lambda |(x^k)_j + (d)_j + z|$$

which has a simple closed-form solution [5, 8].

The most costly step of one iteration is computing $(H_k d)_j$. The special form of B_k in $H_k = B_k + \frac{1}{\mu} I$ allows us to reduce the complexity from up to $O(n)$ to $O(m)$ with m the rank of B_k, which is chosen to be constant. We compute $(B_k d)_i$, whenever it is needed, by maintaining a $2m$ dimensional vector $v := \hat{Q} d$, and using $(B_k d)_i = \gamma_k d_i - q_i^T v$. After each coordinate step v is updated by $v \leftarrow v + z_i \hat{q}_i$ (q_i^T and $\hat{q}_i$ are, respectively, the i-th row and column vector of Q and $\hat{Q}$). The total memory requirement is $O(4mn + 2n + 2m)$ space, which is essentially $O(4mn)$ when $n \gg m$.

2.3 Sufficient decrease and inexact condition

We solve the subproblem inexactly for maximal efficiency. Our termination criteria is derived based on randomized coordinate descent analysis and requires only that we increase the number of subproblem coordinate descent steps as a linear function of the outer iteration counter.

After a descent direction d_k is computed, we evaluate the *sufficient decrease condition* to ensure sufficient reduction on the objective function. In particular, given a constant $0 < \rho \leq 1$ and the model function defined as $Q_k := f(x_k) + \langle d_k, \nabla f(x_k) \rangle + \frac{1}{2}\langle d_k, H_k d_k \rangle + g(x_k + d_k)$, the sufficient decrease condition requires

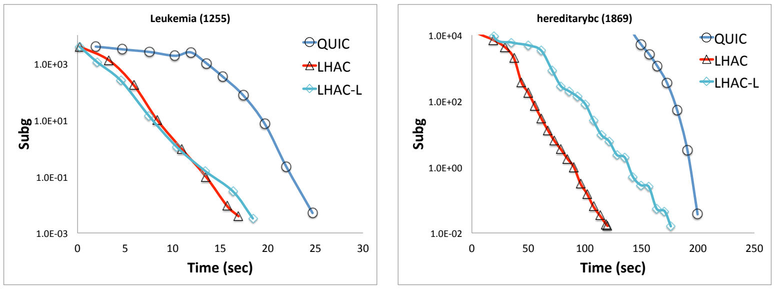
$$F(x_k + d_k) - F(x_k) \leq \rho(Q_k - F(x_k)) \tag{2.5}$$

If the condition (2.5) fails, we then increase the prox parameter μ, as used in $H_k = B_k + \frac{1}{2\mu} I$, and solve the subproblem (2.1) again. This basic idea is the cornerstone of step size selection in most nonlinear optimization algorithms. Instead of insisting on achieving ”full” predicted reduction of the objective function, a fraction of this reduction is usually sufficient. In our experiments small values of ρ provided much better performance than values close to 1. Moreover, updating μ_k and re-optimizing (2.1), not only gives an algorithm for which global convergence rates can be established, but also improves performance compared to line search.

3 Computational experiments

The Low rank Hessian approximation in Active-set Coordinate descent (LHAC) is a C/C++ package that implements the algorithm described here for solving general ℓ_1 regularization problems. We conduct experiments on two of the most well-known ℓ_1 regularized models – Sparse Inverse Covariance Selection (SICS) and Sparse Logistic Regression (SLR). The two specialized C/C++ solvers, QUIC [8] and GLMNET [7, 20] are included in our comparisons. Both of these two packages have been shown to be the state-of-art solvers in their respective categories (see e.g. [20, 19, 8, 12]). We downloaded the latest C/C++ version of the publicly available source code from their official websites, compiled and built the software on the local machine, where all experiments were executed, with 2.4GHz quad-core Intel Core i7 processor, 16G RAM and Mac OS. Both QUIC and GLMNET adopt line search to drive global convergence. We have implemented line search in LHAC as well to see how it compares against backtracking on prox parameter. In all the experiments presented below LHAC denotes the version with prox parameter update and LHAC-L - the version with line search. For all the experiments we choose the initial point $x_0 = \mathbf{0}$, and we terminate the algorithm when $\partial F(x_k) \leq tol \cdot \partial F(x_0)$, with $tol = 10^{-6}$ is satisfied .

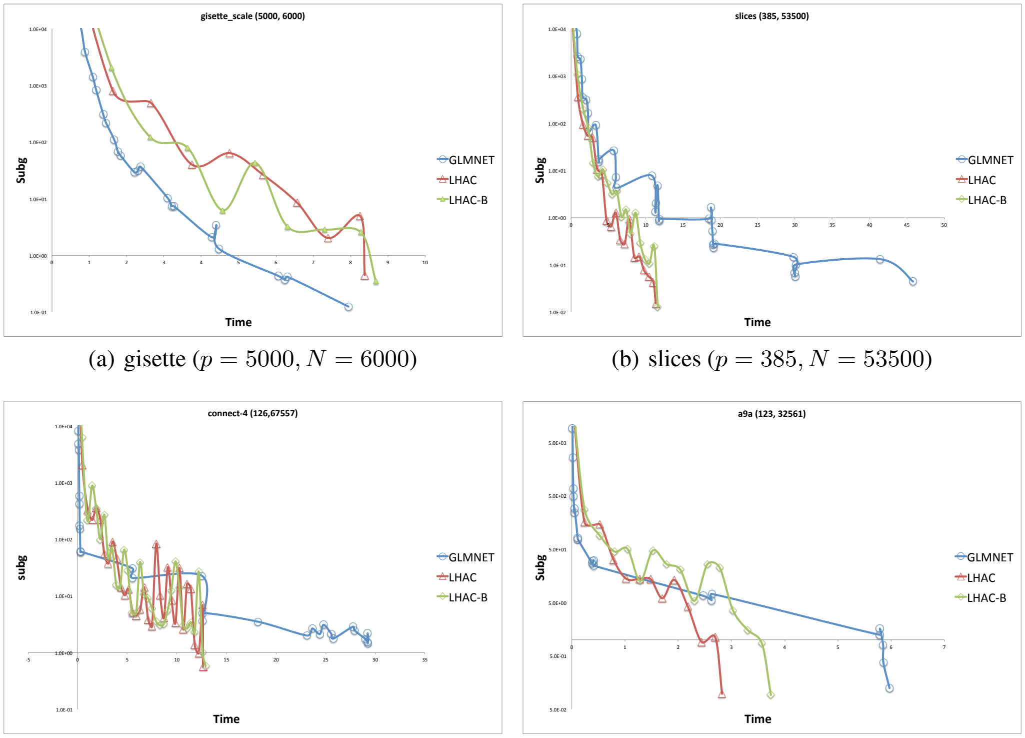
In Figure 1 we report the results of solving sparse inverse covariance selection on two largest real world data sets from gene expression networks preprocessed by [9]. Here p denotes the dimension of the variable matrix. We set



(a) Leukemia. (p = 1255).
(b) hereditarybc. (p = 1869).

Figure 1: Convergence plots on SICS (the y-axes on log scale).

the regularization parameter $\lambda = 0.5$ for both experiments as suggested in [9]. It can be seen that LHAC drives the sub-gradient of the objective to zero nearly 30% faster than QUIC on the smaller data set, Leukemia, and more than 40% faster on the larger set. We also note that backtracking on prox parameter performs better than using line search, thanks to more flexible step sizes and better search directions.



(a) gisette ($p = 5000, N = 6000$)
(b) slices ($p = 385, N = 53500$)
(c) connect-4 ($p = 126, N = 67557$)
(d) a9a ($p = 123, N = 32561$)

Figure 2: Convergence plots on SLR (the y-axes on log scale).

In Figures 2 we compare LHAC to GLMNET on sparse logistic regression problems. The size of the training set and the number of features are denoted by N and p respectively. Note that the evaluation of F requires $O(pN)$ flops and the Hessian requires $O(Np^2)$ flops. We report results on four well-known classification problems from the UCI Adult benchmark set. We see from that LHAC outperforms GLMNET in all but one experiment with data set *gisette* whose size is the smallest of all. On *gisette*, however, the difference in time is within 0.5 secs, while on all others LHAC is generally 2-4 times faster than GLMNET. Again, LHAC scales well and performs robustly in all cases.

References

[1] A. Beck and M. Teboulle, *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183–202.

[2] R. E. Bixby, J. W. Gregory, I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Very large-scale linear programming: A case study in combining interior point and simplex methods*, Operations Research, 40 (1992), pp. pp. 885–897.

[3] R. Byrd, G. Chin, J. Nocedal, and F. Oztoprak, *A family of second-order methods for convex l1-regularized optimization*, tech. rep., (2012).

[4] R. H. Byrd, J. Nocedal, and R. B. Schnabel, *Representations of quasi-newton matrices and their use in limited memory methods*, Mathematical Programming, 63 (1994), pp. 129–156.

[5] D. Donoho, *De-noising by soft-thresholding*, Information Theory, IEEE Transactions on, 41 (1995), pp. 613 –627.

[6] J. Friedman, T. Hastie, and R. Tibshirani, *Sparse inverse covariance estimation with the graphical lasso.*, Biostatistics Oxford England, 9 (2008), pp. 432–41.

[7] ——, *Regularization paths for generalized linear models via coordinate descent*, Journal of Statistical Software, 33 (2010), pp. 1–22.

[8] C.-J. Hsieh, M. Sustik, I. Dhilon, and P. Ravikumar, *Sparse inverse covariance matrix estimation using quadratic approximation*, NIPS, (2011).

[9] L. Li and K.-C. Toh, *An inexact interior point method for L1-regularized sparse covariance selection*, Mathematical Programming, 2 (2010), pp. 291–315.

[10] Y. Nesterov, *Gradient methods for minimizing composite objective function*, CORE report, (2007).

[11] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, NY, USA, 2nd ed., 2006.

[12] P. A. Olsen, F. Oztoprak, J. Nocedal, and S. J. Rennie, *Newton-Like Methods for Sparse Inverse Covariance Estimation*, 2012.

[13] K. Scheinberg, *An efficient implementation of an active set method for SVMs*, JMLR, 7 (2006), pp. 2237–2257.

[14] K. Scheinberg, S. Ma, and D. Goldfarb, *Sparse inverse covariance selection via alternating linearization methods*, NIPS, (2010).

[15] K. Scheinberg and I. Rish, *SINCO - a greedy coordinate ascent method for sparse inverse covariance selection problem*, tech. rep., (2009).

[16] S. Shalev-Shwartz and A. Tewari, *Stochastic methods for l1 regularized loss minimization*, ICML, (2009), pp. 929–936.

[17] R. Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society Series B Methodological, 58 (1996), pp. 267–288.

[18] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, *Sparse reconstruction by separable approximation*, Trans. Sig. Proc., 57 (2009), pp. 2479–2493.

[19] G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, *A comparison of optimization methods and software for large-scale l1-regularized linear classification*, JMLR, 11 (2010), pp. 3183–3234.

[20] G.-X. Yuan, C.-H. Ho, and C.-J. Lin, *An improved GLMNET for l1-regularized logistic regression and support vector machines*, National Taiwan University, (2011).