Optimizing Costly Functions with Simple Constraints: A Limited-Memory Projected Quasi-Newton Algorithm

Mark Schmidt, Ewout van den Berg, Michael P. Friedlander, and Kevin Murphy
Department of Computer Science
University of British Columbia
{schmidtm,ewout78,mpf,murphyk}@cs.ubc.ca

Abstract

An optimization algorithm for minimizing a smooth function over a convex set is described. Each iteration of the method computes a descent direction by minimizing, over the original constraints, a diagonal plus low-rank quadratic approximation to the function. The quadratic approximation is constructed using a limited-memory quasi-Newton update. The method is suitable for large-scale problems where evaluation of the function is substantially more expensive than projection onto the constraint set. Numerical experiments on one-norm regularized test problems indicate that the proposed method is competitive with state-of-the-art methods such as bound-constrained L-BFGS and orthant-wise descent. We further show that the method generalizes to a wide class of problems, and substantially improves on state-of-the-art methods for problems such as learning the structure of Gaussian graphical models and Markov random fields.

1 Introduction

One-norm regularization is increasingly used in the statistical learning community as a tool to learn sparse or parsimonious models. In the case of i.i.d. regression or classification, there are many efficient algorithms (e.g., Andrew and Gao (2007)) for solving such problems. In the case of structured models, such as Markov random fields (MRFs), the problem becomes much harder because the cost of evaluating the objective function is much higher. In particular, for parameter estimation in chain structured graphs, it takes $O(k^2v)$ time per training case, where v is the number of variables (nodes) in the graph, and k is the number of states; for structure learning in Gaussian MRFs, it takes $O(v^3)$ time per objective evaluation; and for structure learning in discrete MRFs, it takes $O(k^v)$ time per evaluation (see Table 1). This makes learning very expensive.

One-norm regularized maximum likelihood can be cast as a constrained optimization problem—as can several other problems in statistical learning, such as training support vector machines, etc. Although standard algorithms such as interior-point methods offer powerful theoretical guarantees (e.g., polynomial-time complexity, ignoring the cost of evaluating the function), these methods typically require at each iteration the solution of a large, highly ill-conditioned linear system; solving such systems is potentially very difficult and expensive. This has motivated some authors to consider alternatives such as gradient-projection methods (Duchi et al., 2008a; Schmidt et al., 2008), which only use the function gradient; these methods require only $\mathcal{O}(n)$ time per iteration (where n is the number of parameters), plus the cost of projecting onto the constraint set. Because the constraints are often simple, the projection onto the set of feasible values can typically be computed efficiently. Although this leads to efficient iterations, using only first-order information means that these methods typically require a substantial number of iterations to reach an accurate solution.

In the case of unconstrained differentiable optimization with a large number of variables, algorithms based on limited-memory quasi-Newton updates, such as L-BFGS (Liu and Nocedal, 1989), are among the most successful methods that require only first derivatives. In a typical optimization algorithm, a step towards the solution is computed by minimizing a local quadratic approximation to the function; between iterations, the quadratic model is updated with second-order information inferred from observed changes in the gradient.

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5. Copyright 2009 by the authors.

Model	Parameters	Evaluation	Projection
GGM-Struct	$O(v^2)$	$O(v^3)$	$O(n)$
MRF-Struct	$O(k^2v^2)$	$O(k^v)$	$O(n)$
CRF-Params	$O(k^2+kf)$	$O(tvk^2)$	$O(n)$

Table 1: Number of parameters, cost of evaluating objective, and cost of projection for different graphical model learning problems with (group) ℓ_1-regularization. Symbols: v: number of nodes in graphical model; k: number of states per node; f: number of features; t: number of training examples; n: number of optimization variables. Models: GGM-Struct: learning a sparse Gaussian graphical model structure by imposing a (group) ℓ_1 penalty on the precision matrix; (Banerjee et al., 2008; Duchi et al., 2008a; Friedman et al., 2007); MRF-Struct: learning a sparse Markov random field structure with (group) ℓ_1 penalties applied to the edge weights (Lee et al., 2006; Schmidt et al., 2008); CRF-Params: learning the parameters of a chain structured conditional random field by using an ℓ_1 penalty on the local features (Andrew and Gao, 2007).

The information available via the L-BFGS updates often allows these methods to enjoy good convergence rates. Crucially, the overhead cost per iteration is only $\mathcal{O}(mn)$, where m is a small number (typically between five and ten) chosen by the user. Tellingly, one of the most successful large-scale bound-constrained optimization methods is L-BFGS-B (Byrd et al., 1995), which combines L-BFGS updates with a gradient-projection strategy. Also, one of the most effective solvers for (non-differentiable) ℓ_1-regularized optimization problems is also an extension of the L-BFGS method, known as orthant-wise descent (OWD) (Andrew and Gao, 2007). Unfortunately, these extensions crucially rely on the separability of the constraints or of the regularization function; this requirement ensures that the scaled search direction continues to provide descent for the objective even after it is projected. In general, it is not straightforward to efficiently apply such algorithms to problems with more general constraints without a substantial increase in computation.

This paper presents a new algorithm based on a two-level strategy. At the outer level, L-BFGS updates are used to construct a sequence of constrained, quadratic approximations to the problem; at the inner level, a spectral projected-gradient method approximately minimizes this subproblem. The iterations of this algorithm remain linear in the number of variables, but with a higher constant factor than the L-BFGS method, and requiring multiple projections for each iteration. Nevertheless, the method can lead to substantial gains when the cost of the projection is much lower than evaluating the function. We describe the new algorithm in §§2–5; in §6 we show experimentally that it equals or surpasses the performance of state-of-the-art methods on the problems shown in Table 1.

2 Projected Newton

We address the problem of minimizing a differentiable function $f(x)$ over a convex set $\mathcal{C}$:

$$\underset{x}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad x \in \mathcal{C}. \tag{1}$$

We cannot in general compute the solution to this problem analytically and must resort to iterative algorithms. Beginning with a solution estimate x_0, at each iteration k the *projected Newton* method forms a quadratic model of the objective function around the current iterate x_k:

$$q_k(x) \triangleq f_k + (x - x_k)^T g_k + \tfrac{1}{2}(x - x_k)^T B_k (x - x_k).$$

Throughout this paper, we use the shorthand notation $f_k = f(x_k)$ and $g_k = \nabla f(x_k)$; B_k denotes a positive-definite approximation to the Hessian $\nabla^2 f(x_k)$. The projected Newton method computes a *feasible* descent direction by minimizing this quadratic model subject to the original constraints:

$$\underset{x}{\text{minimize}} \quad q_k(x) \quad \text{subject to} \quad x \in \mathcal{C}. \tag{2}$$

Because B_k is positive definite, the direction $d_k \triangleq x - x_k$ is guaranteed to be a feasible descent direction at x_k. (If x_k is stationary, then $d_k = 0$.) To select the next iterate, a backtracking line search along the line segment $x_k + \alpha d_k$, for $\alpha \in (0, 1]$, is used to select a steplength α that ensures that a sufficient decrease condition, such as the Armijo condition

$$f(x_k + \alpha d_k) \le f_k + \nu \alpha g_k^T d_k, \quad \text{with} \quad \nu \in (0, 1),$$

is satisfied. By the definition of d, the new iterate will satisfy the constraints for this range of α. A typical value for the sufficient decrease parameter ν is 10^{-4}. A suitable test of convergence for the method is that the norm of the projected gradient, $\mathcal{P}_{\mathcal{C}}(x_k - g_k) - x_k$, where $\mathcal{P}_{\mathcal{C}}$ is the projection onto $\mathcal{C}$, is sufficiently small. If B_k is chosen as the exact Hessian $\nabla^2 f(x_k)$ whenever it is positive definite, and if the backtracking line search always tests the value $\alpha = 1$ first, this method achieves a quadratic rate of convergence in the neighborhood of any point that satisfies the second-order sufficiency conditions for a minimizer; see Bertsekas (1999, §2.2).

Despite its appealing theoretical properties, the projected Newton method just summarized is inefficient in its unmodified form. The major short-coming of the method is that finding the constrained minimizer of the quadratic model may be almost as difficult as solving the original problem. Further, it becomes completely impractical to use a general n-by-n Hessian approximation B_k as n grows large. In the next section, we summarize the L-BFGS quasi-Newton updates to B_k,

Algorithm 1: Projected quasi-Newton Algorithm

```
Given $x_0$, $c$, $m$, $\epsilon$. Set $k \leftarrow 0$
while not Converged do
    $f_k \leftarrow f(x_k)$, $g_k \leftarrow \nabla f(x_k)$
    if k = 0 then
        $d_k \leftarrow -g_k/||g_k||$
    else
        Solve (2) for $x_k^*$                    [Algorithm 2]
        $d_k \leftarrow x_k^* - x_k$
    if $\|\mathcal{P}_\mathcal{C}(x_k - g_k) - x_k\| \leq \epsilon$ then
        Converged
    $\alpha \leftarrow 1$
    $x_{k+1} \leftarrow x_k + d_k$
    while $f_{k+1} > f_k + \nu\alpha g_k^T d_k$ do
        Choose $\alpha \in (0, \alpha)$          [cubic interpolation]
        $x_{k+1} \leftarrow x_k + \alpha d_k$
    $s_k \leftarrow x_{k+1} - x_k$
    $y_k \leftarrow g_{k+1} - g_k$
    if $k = 0$ then
        $S \leftarrow s_k$, $Y \leftarrow y_k$
    else
        if $k \geq m$ then
            Remove first column of $S$ and $Y$
        $S \leftarrow [S \quad s_k]$
        $Y \leftarrow [Y \quad y_k]$
    $\sigma_k \leftarrow (y_k^T s_k)/(y_k^T y_k)$
    Form $N$ and $M$          [update L-BFGS vectors; see §3]
    $k \leftarrow k + 1$
```

and how they lead to efficient computation of $q_k(x)$ and $\nabla q_k(x)$. In §4 we describe how to efficiently solve (2) with an SPG algorithm. Algorithm 1 summarizes the resulting method.

3 Limited-memory BFGS Updates

Quasi-Newton methods allow us to build an approximation to the Hessian by using the observed gradient vectors at successive iterations. It is convenient to define the quantities

$$s_k \triangleq x_{k+1} - x_k \quad \text{and} \quad y_k \triangleq g_{k+1} - g_k.$$

In the BFGS method, the approximation begins with an initial matrix $B_0 \triangleq \sigma I$ (for some positive σ), and at each iteration an updated approximation B_{k+1} is computed that satisfies the secant equation

$$B_{k+1}s_k = y_k.$$

To uniquely choose among matrices satisfying this interpolation condition, the BFGS method chooses the symmetric matrix whose difference with the previous approximation B_k minimizes a weighted Frobenius norm. This leads to the BFGS formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \tag{3}$$

which is a rank-two change to B_k. The limited-memory variant of BFGS (L-BFGS) maintains only the most recent m vectors s_k and y_k, and discards older vectors. The L-BFGS update is described in Nocedal and Wright (1999, §7.3).

The compact representation

$$B_k = \sigma_k I - NM^{-1}N^T,$$

where N is n-by-m, M is m-by-m, and σ_k is a positive scalar, is given by Byrd et al. (1994). Typically, $\sigma_k = (y_k^T s_k)/(y_k^T y_k)$. The SPG algorithm for solving (2) requires computing $q_k(x)$ and $\nabla q_k(x)$ at each iteration, and hence requires matrix-vector products with B_k. With the compact representation, these products can be computed with $\mathcal{O}(mn)$ cost.

The L-BFGS approximation will be strictly positive definite as long as the curvature condition $y_k^T s_k > 0$ is satisfied. This is guaranteed to be true for strictly convex functions, and in the unconstrained case can be satisfied for general functions by using a suitable line search. Since in our case it may not be possible to satisfy this condition along the feasible line segment, the update is skipped on iterations that do not satisfy this condition. An alternative strategy is to use the "damped" L-BFGS update which employs a suitable modification to the vector y_k; see Nocedal and Wright (1999, §18.3).

4 Spectral Projected Gradient

The SPG method (Birgin et al., 2000) is a modification of the classic projected-gradient method. While SPG continues to use projections of iterates along the steepest descent direction, which guarantees that it can generate feasible descent directions, the line search in SPG differs in two crucial ways. First, SPG uses a nonmonotone line search in which sufficient descent is determined relative to a fixed number of previous iterations, rather than just the last. This allows the objective to temporarily increase while ensuring overall convergence. Second, it uses the spectral steplength introduced by Barzilai and Borwein (1988). This gives an initial steplength based on a diagonal approximation to the Hessian that minimizes the squared error in the quasi-Newton interpolation conditions. In particular, we have

$$\alpha_{bb} = \frac{\langle y_{k-1}, y_{k-1} \rangle}{\langle s_{k-1}, y_{k-1} \rangle}. \tag{4}$$

Based on this, we set the initial steplength to

$$\alpha = \min\{\alpha_{\max}, \max\{\alpha_{\min}, \alpha_{bb}\}\},$$

where we set the upper limit $\alpha_{\max} = 10^{10}$ and the lower limit $\alpha_{\min} = 10^{-10}$. Due to its strong empirical

Algorithm 2: Spectral Projected Gradient Algorithm

```
Given $x_0$, step length bounds $0 < \alpha_{\min} < \alpha_{\max}$, initial step
length $\alpha_{bb} \in [\alpha_{\max}, \alpha_{\max}]$, and history length $h$.
while not converged do
    $\bar{\alpha}_k \leftarrow \min\{\alpha_{\max}, \max\{\alpha_{\min}, \alpha_{bb}\}\}$
    $d_k \leftarrow \mathcal{P}_{\mathcal{C}}(x_k - \bar{\alpha}_k \nabla q_k(x_k)) - x_k$
    Set bound $f_b \leftarrow \max\{f(x_k), f(x_{k-1}), \ldots, f(x_{k-h})\}$
    $\alpha \leftarrow 1$
    while $q_k(x_k + \alpha d_k) > f_b + \nu\alpha \nabla q_k(x_k)^T d_k$ do
        Choose $\alpha \in (0, \alpha)$ (cubic interpolation)
    $x_{k+1} \leftarrow x_k + \alpha d_k$
    $s_k \leftarrow x_{k+1} - x_k$
    $y_k \leftarrow \nabla q_k(x_{k+1}) - \nabla q_k(x_k)$
    $\alpha_{bb} \leftarrow y_k^T y_k / s_k^T y_k$
    $k \leftarrow k + 1$
```

performance and simplicity there has recently been a growing interest in the SPG method and it has been used successfully in several applications (e.g., van den Berg and Friedlander (2008); Dai and Fletcher (2005)).

In this paper we use SPG to solve the constrained, strictly-convex quadratic subproblems (2), where B_k is defined by the compact L-BFGS approximation. In the unconstrained case, Friedlander et al. (1999) show that SPG has a superlinear convergence rate for minimizing strictly convex quadratics under certain conditions. The SPG method is summarized in Algorithm 2. Note that the computational complexity is dominated by the cost of function and gradient evaluations (and thus by the efficient matrix-vector products with the Hessian approximation B_k, c.f. §3), and by the projections $\mathcal{P}_{\mathcal{C}}(\cdot)$ onto $\mathcal{C}$. For good overall performance it is thus essential to have an operator that efficiently projects onto the feasible set. Fortunately, such operators exist for a number of commonly encountered convex sets. We give examples in the next section.

In practice it may not be feasible or necessary to run the SPG subproblem until convergence; recall that the only requirement is that the SPG algorithm generates a direction of feasible descent for the quasi-Newton method. Therefore we may, for example, choose to initialize SPG with the projected steepest-descent iterate and perform a fixed number of iterations to improve the steepest descent direction towards the optimal projected quasi-Newton direction. Either way, solving the SPG subproblem can be expected to be more expensive than updating the compact L-BFGS matrix. This implies that the proposed quasi-Newton algorithm is most effective when the cost of evaluating the overall objective function dominates the cost of applying SPG to the subproblem.

5 Projection onto Norm-Balls

The Euclidean projection operator used in the previous section is defined as

$$\mathcal{P}_{\mathcal{C}}(c) = \arg\min_{x} \quad \|c - x\|_2 \quad \text{subject to} \quad x \in \mathcal{C}. \quad (5)$$

We are particularly interested in the case where $\mathcal{C}$ is a ball induced by a given norm:

$$\mathcal{C} \equiv \mathcal{B}_\tau = \{x \mid \|x\| \leq \tau\}.$$

For certain ℓ_p-norms (i.e., $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$), notably ℓ_2 and ℓ_∞, this projection is easily solved; for ℓ_2 the solution is given by $x = \beta c$, with $\beta = \min\{1, \tau/\|c\|_2\}$; for ℓ_∞ we have $x = \operatorname{sgn}(c) \cdot \min\{|c|, \tau\}$. A randomized algorithm with expected linear-time for projection onto the ℓ_1-norm ball is described by Duchi et al. (2008b).

In the context of group variable selection it is often beneficial to work with projection onto mixed p, q-norm balls, defined by

$$\|x\|_{p,q} = \Big(\textstyle\sum_i \|x_{\sigma_i}\|_q^p \Big)^{1/p}, \quad (6)$$

where $\{\sigma_i\}_{i=1}^g$ are g disjoint groups of indices. A special case of this mixed norm is the $\ell_{1,2}$-norm

$$\|x\|_{1,2} = \textstyle\sum_i \|x_{\sigma_i}\|_2,$$

which is used in group Lasso (Yuan and Lin, 2006). The $\ell_{1,\infty}$-norm also arises in group variable selection (Turlach et al., 2005); the $\ell_{\infty,1}$ and $\ell_{\infty,2}$-norms arise in dual formulations of group variable selection problems (Duchi et al., 2008a). Projection onto the $\ell_{\infty,p}$-norm balls reduces to independent projection of each group x_{σ_i} onto ℓ_p-norm balls.

As the following proposition (proved in the Appendix) shows, projection onto the $\ell_{1,2}$-norm ball is done by projecting the vector of group norms $\|x_{\sigma_i}\|_2$ onto the ℓ_1-ball, followed by scaling the elements in each group.

Proposition 1. *Consider $c \in \mathbb{R}^n$ and a set of g disjoint groups $\{\sigma_i\}_{i=1}^g$ such that $\cup_i \sigma_i = \{1, \ldots, n\}$. Then the Euclidean projection $\mathcal{P}_{\mathcal{C}}(c)$ onto the $\ell_{1,2}$-norm ball of radius τ is given by*

$$x_{\sigma_i} = \operatorname{sgn}(c_{\sigma_i}) \cdot w_i, \qquad i = 1, \ldots, g,$$

where $w = \mathcal{P}(v)$ is the projection of vector v onto the ℓ_1-norm ball of radius τ, with $v_i = \|c_{\sigma_i}\|_2$.

Using the efficient ℓ_1 projection algorithm, this immediately gives an expected linear time algorithm for $\ell_{1,2}$ projection.

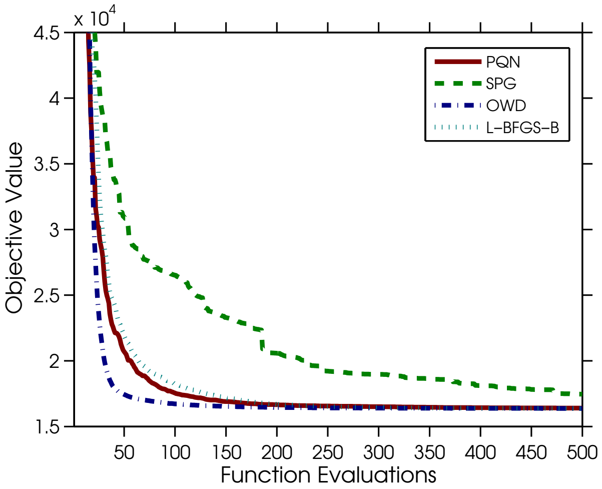

Figure 1: Comparison of objective value versus number of function evaluations for different methods (chain-structured CRF parameter estimation with ℓ_1-regularization).

6 Experiments

In this section we compare the running time of different optimization algorithms on three different convex objective functions subject to convex constraints. For problems with simple constraints, our projected quasi-Newton (PQN) method is competitive with state-of-the-art methods based on L-BFGS, and for problems with more complex constraints (for which L-BFGS cannot easily be applied), our method outperforms the current best approaches.

6.1 Sparse Conditional Random Field Parameter Estimation

We first consider a noun-phrase chunking task (Sang and Buchholz, 2000). The goal is to compute $p(y_i|x, w, v)$, where y_i is one of 22 possible labels (states) for word i, x is a set of features derived from the sentence, and (w, v) are the parameters of the model. A simple approach would be to use logistic regression. However, to exploit correlation amongst neighboring labels, we use a chain structured conditional random field (CRF), defined by

$$p(y \mid x, w, v) \propto \exp\left(\sum_{i=1}^{n} w_{y_i}^T x_i + \sum_{i=1}^{n-1} v_{y_i,y_{i+1}}\right),$$

where w are the parameters for the local evidence potentials, and v are the parameters controlling the 22×22 transition matrix. Note that evaluating this likelihood term (and its derivative) takes $O(vk^2)$ time per sentence, where $k = 22$ is the number of states, and v is the length of each sentence.

We follow Sha and Pereira (2003) and use a feature vector of size 1.8M, representing things such as "did word W occur at location i", for each English word W that occured three or more times in the training data. To perform feature selection from this large set, we use a sparsity-promoting ℓ_1 regularizer. Thus we have to solve the following convex and non-smooth problem

$$\underset{w,v}{\text{minimize}} \; -\sum_i p(y_i \mid x_i, w, v) + \lambda\|w\|_1 + \lambda\|v\|_1, \quad (7)$$

or equivalently the following constrained and smooth problem

$$\begin{array}{ll} \underset{w,v}{\text{minimize}} & -\sum_i p(y_i \mid x_i, w, v) \\ \text{subject to} & \|w\|_1 + \|v\|_1 \le \tau. \end{array} \quad (8)$$

Here λ (or τ) controls the amount of regularization/sparsity. (Note that here x_i represent observed features, rather than parameters to be optimized.)

We compared four optimizers on this data set: (i) applying a bound-constrained L-BFGS method to (7) after converting it into a bound-constrained problem with twice the number of variables (see, e.g., Andrew and Gao (2007)), (ii) applying the OWD algorithm (Andrew and Gao, 2007) directly to (7), (iii) applying SPG directly to (8), and finally (iv) applying our PQN method to (8). Figure 1 plots the value of the objective (7) against the number of evaluations of the function for $\lambda = 1$ (and τ set such that the problems give the same solution). With this value of λ, the model achieves a nearly identical prediction error to that achieved with ℓ_2-regularization, but only had $10,907$ non-zero parameters. In this plot, we see that the three methods that use the L-BFGS approximation (L-BFGS-B, OWD, and PQN) behave similarly and all find a very good solution after a small number of iterations. We also compared applying PQN to the bound constrained problem solved by L-BFGS-B, and found that it gave nearly identical performance to PQN applied to (8). In contrast, SPG takes substantially longer to approach this quality of solution. These trends also held for other values of λ. While the added iteration cost of PQN makes it unappealing for this problem in comparison to OWD or L-BFGS-B, we emphasize that L-BFSG-B and OWD can not handle more general constraints, while PQN can be used to give this level of performance for more general constraints.

6.2 Gaussian Graphical Model Structure Learning

We consider the problem of learning the structure (topology) of a Gaussian graphical model. Since absent edges in the graph correspond to zeros in the precision matrix, we can efficiently learn the structure by solving

$$\underset{K \succ 0}{\text{minimize}} \; -\log\det(K) + \operatorname{tr}(\hat{\Sigma}K) + \lambda\|K\|_1,$$

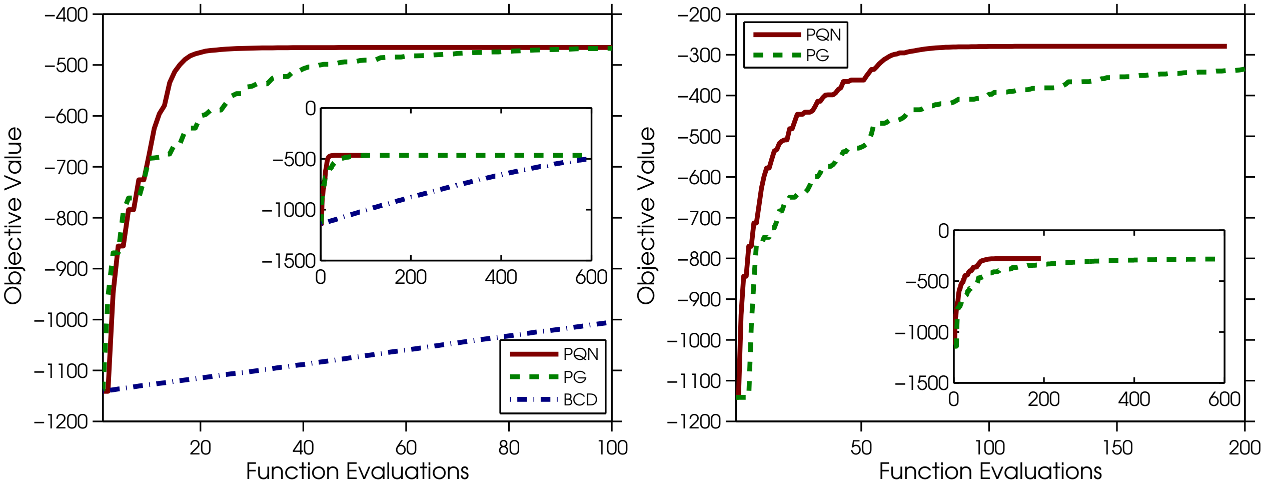

Figure 2: Comparison of dual objective value versus number of function evaluations for different methods (GGM structure learning with ℓ_1-regularization (left) and group ℓ_1-regularization (right)).

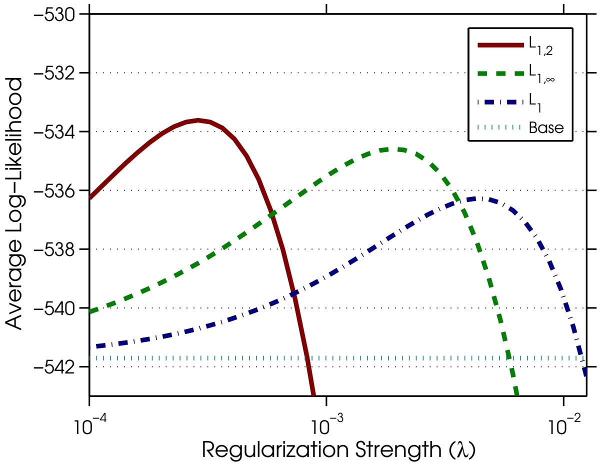

Figure 3: Average cross-validated log-likelihood against regularization strength under different regularization schemes applied to the regularized empricial covariance for the yeast gene expression data from (Duchi et al., 2008a).

where $\hat{\Sigma}$ is the empirical covariance, K is the precision matrix, the norm $\|K\|_1$ is applied element-wise, and $K \succ 0$ denotes symmetric positive definiteness of K; see Banerjee et al. (2008). This method is dubbed the "graphical lasso" by Friedman et al. (2007). Note that evaluating the objective takes $O(v^3)$ time.

A block coordinate descent (BCD) algorithm for optimizing this is presented in Banerjee et al. (2008) and Friedman et al. (2007). This method solves the corresponding dual problem

$$\begin{aligned} \underset{\Sigma, W}{\text{maximize}} \quad & \log\det(\hat{\Sigma} + W) \\ \text{subject to} \quad & \hat{\Sigma} + W \succ 0,\ \|W\|_\infty \leq \lambda, \end{aligned} \tag{9}$$

where W is the dual variable and $\|W\|_\infty$ is computed element-wise. More recently, Duchi et al. (2008a) describe a projected gradient (PG) method for solving (9). The PG method constructs a dual feasible $K = (\hat{\Sigma} + W)^{-1}$ at each iteration, and stops when the primal-dual gap is sufficiently small. While projection in the dual formulation is efficient, first-order methods can be slow. In Figure 2, we compare our PQN method to the PG method on the gene expression data from Duchi et al. (2008a) (we also plot the objective value at corresponding iterations of the BCD method).

Duchi et al. also consider an interesting extension to the problem, where sparsity on groups of variables is imposed using $\ell_{1,\infty}$-regularization, analogous to group lasso in the classification/regression setting. In Figure 2, we see that PQN is also faster than the PG method for this problem. Further, PQN can also easily be applied in the case of the $\ell_{1,2}$ group norm. Figure 3 compares the three regularization strategies over fifty random train/test splits following the scenario outlined in Duchi et al. (2008a), where we see that regularization with group $\ell_{1,2}$-norm provides a further improvement.

6.3 Markov Random Field Structure Learning

In the case of Markov random fields on binary variables (such as Ising models), there is a one-to-one correspondence between parameters and edges in the graph, as in the Gaussian case, and so one can again learn structure by optimizing an ℓ_1-penalized log likelihood (Lee et al., 2006). Note, however, that computing the objective now takes $O(k^v)$ time per example, where $k = 2$ and v is the number of nodes. Lee et al. (2006) proposed to use loopy belief propagation to approximate the objective, and to use an incremental grafting strategy to perform the optimization.

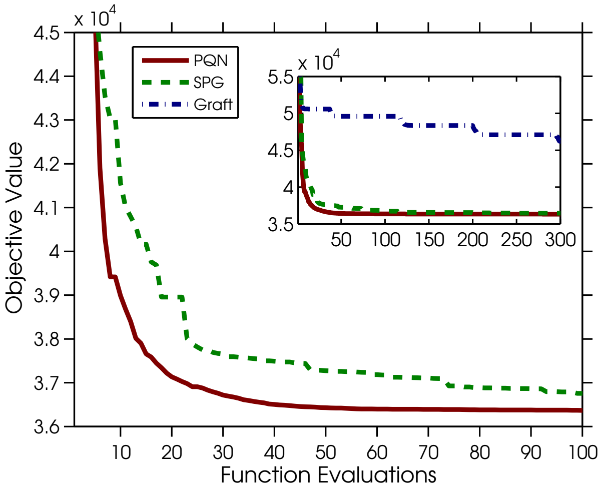

Figure 4: Comparison of objective value versus number of function evaluations for different methods (MRF structure learning with group ℓ_1-regularization).

If the variables are non-binary, there are multiple parameters per edge, and so one needs to use a group ℓ_1 penalty to learn sparse graphs (Schmidt et al., 2008). (This also happens when learning the structure of conditional random fields, where each edge is often associated with a weight vector.) Schmidt et al. (2008) convert the penalized problem into a constrained problem by introducing one variable per group, and applying SPG to the resulting second-order cone problem. Using our proposition regarding efficient projection for the $\ell_{1,2}$-norm balls, PQN can be applied directly to the constrained version of the problem.

Figure 4 compares the SPG method of Schmidt et al. (2008), the proposed PQN method, and a grouped variable extension of the grafting method of Lee et al. (2006), on an MRF structure learning problem. (The data set consists of 5400 measurements of 11 protein activity levels, that have been discretized into three states, representing under-expressed, normal, and over-expressed, as in Sachs et al. (2005), who modeled the data using a directed graphical model whose structure was estimated by heuristic search methods.) We see that, once again, the PQN method is much faster than the previous best (SPG) method.

7 Discussion

Although our experiments have focused on constraints involving norms of the parameter vectors, there is a wide variety of constraints where the projection can easily be computed. For example, projection onto a hyper-plane or half-space is a trivial calculation, while projection onto the probability simplex can be computed in $\mathcal{O}(n\ log\ n)$ (Duchi et al., 2008b). Projection of a symmetric matrix onto the cone of positive semi-definite matrices (in Frobenius norm) is given by setting negative eigenvalues in the spectral decomposition of the matrix to 0 (Boyd and Vandenberghe, 2004). Projection onto a second-order cone constraint of the form $||x||_2 \leq s$ (where x and s are both variables) can be computed in linear-time (Boyd and Vandenberghe, 2004), while projection onto a constraint of the form $||x||_\infty \leq s$ can be solved in $\mathcal{O}(n\ log\ n)$ (Schmidt et al., 2008). Further, if it is difficult to compute the projection onto the full constraint set but simple to project onto subsets of the constraints, Dykstra's algorithm (Dykstra, 1983) can be used to compute the projection (but the efficiency of this depends on the constraints).

This paper is concerned with the problem of minimizing an expensive objective that depends on a large number of variables, within the domain of a convex set where the projection is easily computed. The method presented here is a natural generalization of the L-BFGS method to this task, and takes advantage of the recent SPG method to make the iterations efficient. Our experiments indicate that the method achieves state of the art performance on problems with very simple constraints, while it represents a substantial improvement over state of the art methods for problems with more complex constraints.

Appendix

Proof of Proposition 1. Recall that the n-vector x is partitioned into g groups with pairwise disjoint index sets σ_i, $i = 1, \ldots, g$ whose union is $\{1, \ldots, n\}$. Write x as a g-vector $\widetilde{x} = (\widetilde{x}_1, \widetilde{x}_2, \cdots, \widetilde{x}_g)$, where each $\widetilde{x}_k = (x_j)_{j \in \sigma_k}$ denotes the tuple formed by x_{σ_k}, the components of x belonging to group k. With this notation we extend the signum function as

$$\operatorname{sgn}(\widetilde{x})_k = \operatorname{sgn}(\widetilde{x}_k) = \frac{\widetilde{x}_k}{|\widetilde{x}_k|}, \tag{10}$$

with $|\widetilde{x}_k| = \|x_{\sigma_k}\|_2$. It can be verified that this extended signum function satisfies the usual properties that

$$\operatorname{sgn}(\alpha\widetilde{x}) = \operatorname{sgn}(\widetilde{x}) \quad \text{for all } \alpha > 0, \tag{11a}$$

$$\|\operatorname{sgn}(\widetilde{x}_k)\|_2 \leq 1. \tag{11b}$$

We adopt the convention that $\operatorname{sgn}(0)$ can be taken to be any vector satisfying the second property. Finally, with the p-norm $\|\widetilde{x}\|_p := \|x\|_{p,2}$ based on (6), it can be shown that $\nabla\frac{1}{2}\|\widetilde{x}\|_2^2 = \widetilde{x}$, and $\nabla\|\widetilde{x}\|_1 = \operatorname{sgn}(\widetilde{x})$.

We now apply these results to derive the optimality conditions for the group projection leading to the extended soft-thresholding operator. Using $\|\widetilde{x}\|_2 = \|x\|_2$, and $\|\widetilde{x}\|_1 = \|x\|_{1,2}$ we can solve

$$\underset{\widetilde{x}}{\text{minimize}} \quad \tfrac{1}{2}\|\widetilde{c} - \widetilde{x}\|_2^2 + \lambda\|\widetilde{x}\|_1. \tag{12}$$

A vector $\widetilde{x}$ is a solution of this problem if and only if it satisfies

$$\nabla(\tfrac{1}{2}\|\widetilde{x}-\widetilde{c}\|_2^2+\lambda\|\widetilde{x}\|_1)=\widetilde{x}-\widetilde{c}+\lambda\,\mathrm{sgn}(\widetilde{x})=0. \quad (13)$$

We claim that the $\widetilde{x}$ satisfying this condition, and hence giving the solution of (12), is given by

$$\widetilde{x}_k=S_\lambda(\widetilde{c})_k=S_\lambda(\widetilde{c}_k), \quad (14)$$

where

$$S_\lambda(\widetilde{c}_k)=\begin{cases}\mathrm{sgn}(\widetilde{c}_k)(|\widetilde{c}_k|-\lambda) & \text{if } |\widetilde{c}_k|>\lambda;\\ 0 & \text{otherwise.}\end{cases}$$

To check this we separately consider the two cases $|\widetilde{c}|>\lambda$ and $|\widetilde{c}|\le\lambda$. In the case where $|\widetilde{c}_k|>\lambda$, substitute $\widetilde{x}_k=S_\lambda(\widetilde{c}_k)$ into (13) and use property (11a) to obtain:

$$\begin{aligned}&\widetilde{x}_k-\widetilde{c}_k+\lambda\,\mathrm{sgn}(\widetilde{x}_k)\\ &=\ \mathrm{sgn}(\widetilde{c}_k)(|\widetilde{c}_k|-\lambda)-\widetilde{c}_k+\lambda\,\mathrm{sgn}(\mathrm{sgn}(\widetilde{c}_k)(|\widetilde{c}_k|-\lambda))\\ &=\ \mathrm{sgn}(\widetilde{c}_k)(|\widetilde{c}_k|-\lambda)-\widetilde{c}_k+\lambda\,\mathrm{sgn}(\widetilde{c}_k)\\ &=\ \mathrm{sgn}(\widetilde{c}_k)\cdot|\widetilde{c}_k|-\widetilde{c}_k=0.\end{aligned}$$

In case $|\widetilde{c}_k|\le\lambda$, we substitute $\widetilde{x}=0$, giving $\widetilde{c}=\lambda\,\mathrm{sgn}(0)$. But because $\mathrm{sgn}(0)$ is arbitrary, we can choose it as $(1/\lambda)\widetilde{c}_k$, which clearly satisfies the condition.

Assuming $\|\widetilde{c}\|_1>\tau>0$, it remains to be shown how to find λ giving $\|S_\lambda(\widetilde{c})\|_1=\tau$, based on $v_i=|\widetilde{c}_i|$. Defining $\mathcal{I}_\lambda=\{j\mid|\widetilde{c}_j|>\lambda\}=\{j\mid v_j>\lambda\}$, and noting that $|\mathrm{sgn}(\widetilde{c}_i)|=1$ for all $i\in\mathcal{I}_\lambda$ we have

$$\begin{aligned}\|S_\lambda(\widetilde{c})\|_1&=\sum_{i\in\mathcal{I}_\lambda}|\mathrm{sgn}(\widetilde{c}_i)(|\widetilde{c}_i|-\lambda)|=\sum_{i\in\mathcal{I}_\lambda}||\widetilde{c}_i|-\lambda|\\ &=\sum_{i\in\mathcal{I}_\lambda}|\widetilde{c}_i|-\lambda=\sum_{i\in\mathcal{I}_\lambda}v_i-\lambda=S_\lambda(v).\end{aligned}$$

The last step follows from $v_i\ge 0$ and the definition of $\mathcal{I}_\lambda$, and shows that the λ corresponds to the value chosen for projection onto the ℓ_1-norm ball. The final part of the proposition constructs x from v and $w=P_\tau(v)$. In group notation this step can be derived as

$$\begin{aligned}\widetilde{x}_i&=\mathrm{sgn}(\widetilde{c}_i)\cdot w_i=\mathrm{sgn}(\widetilde{c}_i)\cdot\mathrm{sgn}(v_i)\cdot\max\{0,v_i-\lambda\}\\ &=\mathrm{sgn}(\widetilde{c}_i)\cdot\max\{0,v_i-\lambda\}\\ &=\mathrm{sgn}(\widetilde{c}_i)\cdot\max\{0,|\widetilde{c}_i|-\lambda\},\end{aligned}$$

where we used (10) and the fact that we can take $\mathrm{sgn}(v_i)=\mathrm{sgn}(|\widetilde{c}_i|)=1$. This exactly coincides with (14), as required. □

References

G. Andrew and J. Gao. Scalable training of ℓ_1-regularized log-linear models. *ICML*, 2007.

O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *JMLR*, 9:485–516, 2008.

J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8:141–148, 1988.

E. van den Berg and M. Friedlander. Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comp.*, 31(2):890–912, 2008.

D. Bertsekas. *Nonlinear programming.* Athena Scientific, 1999.

E. Birgin, J. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.*, 10(4):1196–1211, 2000.

S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

R. Byrd, J. Nocedal, and R. Schnabel. Representations of quasi-newton matrices and their use in limited memory methods. *Math. Program.*, 63(1):129–156, 1994.

R. Byrd, P. Lu, and J. Nocedal. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Stat. Comput.*, 16(5):1190–1208, 1995.

Y.-H. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numer. Math.*, 100:21–47, 2005.

J. Duchi, S. Gould, and D. Koller. Projected subgradient methods for learning sparse gaussians. *UAI*, 2008a.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. *ICML*, 2008b.

R. Dykstra. An algorithm for restricted least squares regression. *JASA*, 78(384):837–842, 1983.

A. Friedlander, J. M. Martínez, B. Molina, and M. Raydan. Gradient method with retards and generalizations. *Siam J. Numer. Anal.*, 36(1):275–289, 1999.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation the graphical lasso. *Biostatistics*, 2007.

S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of markov networks using l1-regularization. *NIPS*, 2006.

D. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Math. Program. B*, 45(3): 503–528, 1989.

J. Nocedal and S. Wright. *Numerical Optimization.* Springer, 1999.

K. Sachs, O. Perez, D. Pe'er, D. Lauffenburger, and G. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308, 2005.

E. Sang and S. Buchholz. Introduction to the conll-2000 shared task: Chunking. *CoNLL*, 2000.

M. Schmidt, K. Murphy, G. Fung, and R. Rosales. Structure learning in random fields for heart motion abnormality detection. *CVPR*, 2008.

F. Sha and F. Pereira. Shallow parsing with conditional random fields. *NACL-HLT*, 2003.

B. Turlach, W. Venables, and S. Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B*, 68(1):49–67, 2006.