International Statistical Review
Revue Internationale de Statistique

International Statistical Review (2014), 82, 1, 71–75 doi:10.1111/insr.12033

Discussions

Yves Atchade and George Michailidis

Department of Statistics, University of Michigan, Ann Arbor, MI 48109, USA
E-mail: gmichail@umich.edu

Optimization methods have always played a key role in the development of statistical methodology, but have become of critical importance for modern methods to analyse large size high-dimensional data. Lange, Choi and Zhua are to be commended for providing a comprehensive overview of optimization methods widely used in statistics. The paper discusses classical unconstrained optimization algorithms (steepest descent and variants), the majorization–maximisation framework that has proved very useful in devising novel algorithms for a variety of statistical problems, and also provides a flavour of constrained optimization problems arising in high-dimensional statistics (e.g. regularisation, matrix completion, etc.). The topics discussed and their accompanying examples focus on important classes of algorithms that have helped statisticians develop and solve complex models.

In this note, we focus on a class of algorithms suitable for high-dimensional constrained optimization problems. The class of interest is that of *proximal* algorithms, that are very generally applicable in constrained non-smooth optimization, but are in particular very well suited to recent statistical techniques developed for the analysis of high-dimensional data Bach *et al.* (2011), Lee *et al.* (2010), & Ravikumar *et al.* (2010). In our discussion, we make connections to a number of topics discussed in Lange, Choi and Zhua, including connections to majorization–maximisation and classical gradient descent algorithms and acceleration schemes.

1 Proximal Algorithms

We start by providing some definitions and then discuss special cases arising in high-dimensional statistics.

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a closed proper convex function; that is, its epigraph is a non-empty closed convex set. The *proximal operator* of f is defined as

$$\operatorname{prox}_f(u) = \operatorname{argmin}_x \left(f(x) + \frac{1}{2\lambda} ||x - u||_2^2 \right), \quad x, u \in \mathbb{R}^n, \tag{1}$$

where $|| \cdot ||_2^2$ denotes the usual Euclidean norm, and $\lambda > 0$ is a parameter that controls the degree to which the proximal operator maps point closer to the minimum of f. As shown in Boyd & Vandenberghe (2004), the function minimised is strongly convex and not everywhere infinite and hence admits a unique minimiser. Note that the definition implies that $\operatorname{prox}_f(u)$ is a point that approximately minimises f, but remains close to u.

In general, proximal mapping requires solving the minimisation problem posited in (1), but there are instances where the proximal operator admits an explicit expression.

(1) $f(x) = 0$, then $\text{prox}_f(u) = u$.
(2) $f(x) = I_C(x)$ where I_C denotes the indicator function of a convex set C; then, $\text{prox}_f(u) = P_C(u)$ corresponds to the *projection* P_C on C.
(3) $f(x) = \psi||x||_1$, where $||\cdot||_1$ denotes the ℓ_1 norm, then the i-th coordinate of $\text{prox}_f(u)$ is given by: (i) $\text{prox}_f(u)_i = 0$ if $|u_1| \le \psi$; (ii) $\text{prox}_f(u)_i = u_i - \psi$ if $u_i > \psi$; and (iii) $\text{prox}_f(u)_i = u_i + \psi$ if $u_i < -\psi$. It can be seen that the proximal mapping corresponds to a soft-thresholding operation.

A proximal minimisation algorithm is defined at the k-th iteration step by

$$x^{k+1} := \text{prox}_f\left(x^k\right).$$

As presented in Bauschke & Combettes (2011), if f has a minimum, then x^k will converge to the set of minimisers of f and $f(x^k)$ to its optimal value. Similarly to steepest descent algorithms, convergence is guaranteed for values of the parameter λ that satisfy $\lambda^k > 0$ and $\sum_{k=1}^{\infty} \lambda^k = \infty$.

This basic algorithm has not found many applications but is instructive from a theoretical point of view. One important application though is on solving the quadratic function

$$f(x) = \frac{1}{2}x'Ax - b'x, \quad A \in \mathbb{R}^{n\times n}, \quad x, b \in \mathbb{R}^n$$

with $A \succeq 0$ (positive semidefinite). The proximal mapping can be written down explicitly as

$$\text{prox}_f(u) = \left(A + \frac{1}{\lambda}I\right)^{-1}\left(b + \frac{1}{\lambda}u\right),$$

which gives rise to the update of Golub & Wilkinson (1996)

$$x^{k+1} = \left(A + \frac{1}{\lambda}I\right)^{-1}\left(b + \frac{1}{\lambda}x^k\right).$$

The power of proximal mappings is revealed in the following setting.

2 Proximal Gradient Method

Consider the following optimization problem:

$$\min_x f(x) + g(x), \tag{2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are closed proper convex functions, and f is *differentiable*. The function g can be used to encode constraints on x as discussed in the 'Augmented Lagrangians' section of Lange, Choi and Zhua.

Then, the proximal gradient method is given by

$$x^{k+1} = \text{prox}_g\left(x^k - \lambda^k \nabla f(x^k)\right), \tag{3}$$

where $\lambda^k > 0$ denotes the step size. If $\nabla f(x)$ is Lipschitz continuous with constant M, then this algorithm converges in $\mathcal{O}(1/k)$ steps, for fixed step size $\lambda^k \equiv \lambda \in (0, 1/M)$. If M is unknown, then various line search strategies (akin to those used in gradient descent methods) can be employed.

Note that using the special form of the proximal operator, it can be seen that when $g = 0$, this algorithm reduces to the standard gradient descent one, when $g = \psi||x||_1$ (a lasso penalty), it leads to soft-thresholding and for $g = I_C$, it reduces to the projected gradient method in Bertsekas (1999).

In Beck & Tebouille (2012), the proximal gradient algorithm is interpreted as a majorization–minimisation algorithm. Specifically, consider the upper bound on $f(x)$ given by

$$Q_\lambda(x, y) = f(y) + \nabla f(y)'(x - y) + \frac{1}{2\lambda}||x - y||_2^2, \ \ \lambda > 0.$$

For fixed λ, $Q_\lambda(x, y)$ is convex and satisfies $Q_\lambda(x, x) = f(x)$ and is an upper bound on f when ∇f is Lipschitz continuous with constant M and $\lambda \in (0, 1/M]$. Then, the algorithm that uses updates of the form

$$x^{k+1} = \text{argmin}_x Q_\lambda\left(x, x^k\right)$$

is a majorization–minimisation one. Analogously, for the function $f(x) + g(x)$, we can use $\tilde{Q}_\lambda(x, y) = Q_\lambda(x, y) + g(x)$, and some algebra shows that the majorization–minimisation-based update

$$x^{k+1} = \text{argmin}_x \tilde{Q}_\lambda\left(x, x^k\right)$$

is equivalent to (3).

3 Accelerated Proximal Gradient Method

Nesterov (Nesterov, 1983) introduced a sequence of updates that *accelerate* the convergence rate from linear ($\mathcal{O}(1/k)$) to quadratic ($\mathcal{O}(1/k^2)$) in convex programming problems. The sequence extrapolates between previous updates of the algorithm. Specifically, for the proximal gradient method, it takes the form

$$y^{k+1} = x^k + \omega^k\left(x^k - x^{k-1}\right), \ \ x^{k+1} = \text{prox}_g\left(y^{k+1} - \lambda^k \nabla f\left(y^{k+1}\right)\right), \tag{4}$$

where $\omega^k \in [0, 1)$ is an extrapolation parameter and λ^k the usual step size. A simple choice for the extrapolation parameter is $\omega^k = k/(k + 3)$.

Then, it can be established that for ∇f Lipschitz continuous with constant M, the acceleration scheme guarantees convergence in $\mathcal{O}(1/k^2)$ steps with a fixed step size $\lambda^k \equiv \lambda \in (0, 1/M]$. As before, if M is unknown, the step sizes can be determined through a line search.

4 Stochastic Proximal Gradient Algorithms

An area where we believe more research is needed to specifically suit the needs of statistics is stochastic optimization. This corresponds for example to (2), where the function f is given as an intractable integral of the form $f(x) = \int F(x, \zeta)\Pi(d\zeta)$. This is a well-known problem in stochastic programming and online learning and has generated various stochastic extensions of the algorithms presented earlier (Nemirovski *et al.*, 2009; Xiao, 2010; Duchi *et al.*, 2012; Lan, 2012). But in statistics, this problem takes a more challenging form. Latent variables abound in statistics and lead to log-likelihood functions and their derivatives that are intractable: $f(x) = \log \int F(x, \zeta)\Pi(d\zeta)$, $\nabla f(x) = \int \nabla_x \{\log F(x, \zeta)\} \check{\Pi}_x(d\zeta)$, where $\check{\Pi}_x(d\zeta) \propto F(x, \zeta)\Pi(d\zeta)$. The important point here is that the distribution $\check{\Pi}_x$ is typically very difficult to simulate, and unlike most examples in online learning stochastic optimization, $\check{\Pi}_x$ depends on x. Nevertheless, one can easily adapt the proximal gradient algorithm and its accelerated version presented earlier, by replacing $\nabla f(x)$ by a Monte Carlo approximation $\widehat{\nabla} f(x)$ obtained by simulating from $\check{\Pi}_x$ (possibly using Markov chain Monte Carlo methods). The approximation can be obtained with a fixed number of Monte Carlo steps, or with an increasing number of Monte Carlo steps.

For illustration purposes, consider the following random effects logistic regression example. We have n statistical units with repeated binary responses $\{y_{it},\ 1 \le t \le T_i\}$, $y_{it} \in \{0, 1\}$. For the covariate matrix $Z \in \mathbb{R}^{n \times p}$, with i-th row denoted by z_i, we assume that

$$y_{it}|u_i \sim \text{Ber}\left(\frac{e^{z_i\beta+u_i}}{1+e^{z_i\beta+u_i}}\right), \quad 1 \le i \le n, \quad 1 \le t \le T_i,$$

where $u_i \overset{i.i.d.}{\sim} N(0, \sigma^2)$. The log-likelihood of β is intractable and requires integrating out the random effect u. We estimate β by an ℓ_1-penalised likelihood approach; thus, $g(\beta) = \psi\|\beta\|_1$.

We implemented the proximal gradient algorithm, and its accelerated version by replacing in (3) and (4), $\nabla f(x^k)$ by $\widehat{\nabla} f(x^k)$ obtained using a Markov chain Monte Carlo scheme. Figure 1 depicts the relative error $\|\beta_k - \beta_\star\|/\|\beta_\star\|$, as a function of the iterations k for both algorithms, where $\beta_\star$ denotes the true value of the parameter vector.

This simulation example suggests that stochastic versions of the proximal gradient algorithm and its accelerated versions can be designed to deal with high-dimensional statistical models with intractable log-likelihood functions. However, more work is required to gain a clear and deep understanding of the conditions needed so that these extensions work and exhibit convergence properties analogous to their deterministic counterparts.

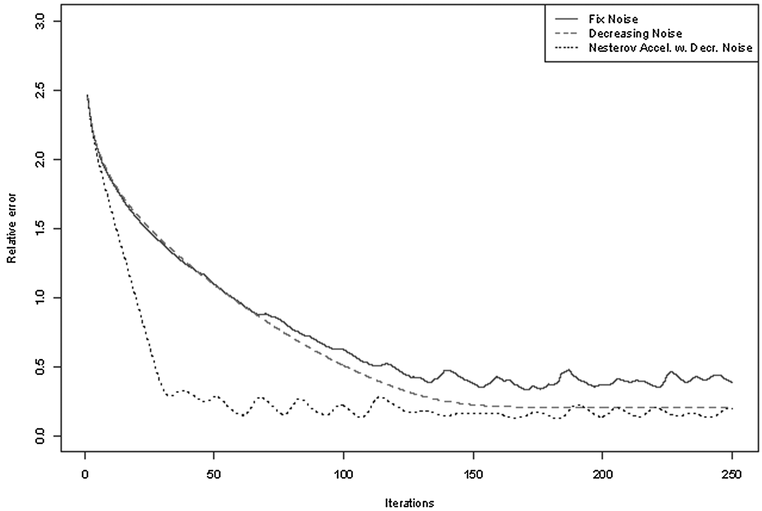

Figure 1. *Relative errors for stochastic proximal gradient algorithm for a random effects logistic model with* $p = 100$, $T = 10$ *and* $k = 200$.

5 Alternating Direction Method of Multipliers

In the proximal gradient algorithm, the function f was assumed to be smooth. However, the method can be adapted to handle the following variant of the minimisation problem:

$$\min_{x} f(x) + g(x),$$

where $f, g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are closed proper convex functions, and both f, g can be *non-differentiable*. Then, the alternating direction method of multipliers employs the following updates:

$$x^{k+1} = \mathrm{prox}_f(z^k - u^k), \ \ z^{k+1} = \mathrm{prox}_g\left(x^{k+1} + u^k\right), \ \ u^{k+1} = u^k + x^{k+1} - z^{k+1}.$$

It can be seen that this method handles the two functions in the objective function completely separately through their proximal operators. It is most useful when the proximal operators of f and g can be easily computed, but that of the objective function $f + g$ is not. The convergence theory for this method is discussed in detail in Boyd *et al.* (2011).

Acknowledgement

The authors' research was supported in part by the NSF grant DMS-1228164.

References

Bach, F., Jenatton, R., Mairal, J. & Obozinski, G. (2011). Optimization with sparsity-inducing penalties. *Found. Trends Mach. Learn.*, **4**, 1–106.

Bauschke, H. & Combettes, P. (2011). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces.* New York: Springer.

Beck, A. & Tebouille, M. (2012). Smoothing and first order methods: a unified framework. *SIAM J. Optim.*, **22**, 557–580.

Bertsekas, D. (1999). *Nonlinear Programming.* Belmont, MA: Athena Scientific.

Boyd, S. & Vandenberghe, L. (2004). *Convex Optimization.* New York: Cambridge University Press.

Duchi, J. C., Agarwal, A., Johansson, M. & Jordan, M. (2012). Ergodic mirror descent. *SIAM J. Optim.*, **22**, 1549–1578.

Boyd, S., Parikh, N., Chu, E., Peleato, B. & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, **3**, 1–122.

Golub, G. & Wilkinson, J. (1996). Note on the iterative refinement of least squares solution. *Numer. Math.*, **9**, 139–148.

Lan, G. (2012). An optimal method for stochastic composite optimization. *Math. Program. Ser. A*, **133**, 365–397.

Lee, J., Recht, B., Salakhutdinov, R., Srebro, N. & Tropp, J. (2010). Practical large scale optimization for max-norm regularization. *Adv. Neural Inf. Process. Syst.*, **23**, 1297–1305.

Nemirovski, A., Juditsky, A., Lan, G. & Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, **19**, 1574–1609.

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate O(1/k2). *Soviet Mathematics Doklady*, **27**, 372–376.

Ravikumar, P., Agarwal, A. & Wainwright, M. (2010). Message-passing for graphstructured linear programs: Proximal methods and rounding schemes. *J. Mach. Learn. Res.*, **11**, 1043–1080.

Xiao, L. (2010). Dual averaging methods for regularized stochastic learning and online optimization. *J. Mach. Learn. Res.*, **11**, 2543–2596.

[*Received June 2013, accepted June 2013*]