Optimization in Data Science: Algorithms

Stephen Wright (UW-Madison)

MoCaO, July, 2022

Outline: Algorithms

Optimization algorithms — mostly elementary — that can exploit the structure of data science applications.

- Gradient algorithms (first-order)
  - prox-gradient for regularization terms
- Accelerated gradient
- Stochastic gradient
  - and "variance reduced" hybrids with full-gradient
- Coordinate descent
- Primal-dual methods (for min-max problems)
- Augmented Lagrangian / ADMM

Structures of ML Optimization Problems

- Finite sum: $f(x) = \frac{1}{m}\sum_{j=1}^{m} f_j(x)$, $x \in \mathbb{R}^n$.
  - m and d may both be huge!
  - f_j sometimes convex (kernel learning, logistic regression) but nonconvex problems (NN, low-rank matrix) recently interesting.
- Regularization: $f(x) + \lambda\psi(x)$, where $\lambda > 0$ is regularization parameter and ψ is the regularization function or regularizer.
  - ψ usually nonsmooth, possibly nonconvex.
  - Introduces structure explicitly into the solution x e.g. sparsity, column selection.
  - Adjust λ to control fit to observations vs structure.
- min $f(x)$ subject to $x \in \Omega$ for closed convex Ω.
  - Sometimes constraint $x \in \Omega$ replaces regularization $\lambda\psi(x)$.
  - Arises naturally in some formulations e.g. kernel SVM dual form, nonnegative matrix factorization.

Everything Old is New Again

"Old" approaches from the optimization literature have found many applications in data analysis.

- Nesterov acceleration of gradient methods [Nesterov, 1983].
- Alternating direction method of multipliers (ADMM) [Eckstein and Bertsekas, 1992].
- Parallel coordinate descent and incremental gradient algorithms [Bertsekas and Tsitsiklis, 1989]
- Stochastic gradient [Robbins and Monro, 1951]
- Frank-Wolfe / conditional gradient [Frank and Wolfe, 1956, Dunn, 1979].

Many extensions have been made to these methods and their convergence analysis. Many variants and adaptations proposed.

Preliminaries: Convexity

f is convex if

$$f(\alpha x + (1 - \alpha y)) \leq \alpha f(x) + (1 - \alpha) f(y).$$

When f is convex and *smooth*, we have

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

Strongly convex is there is $\gamma > 0$ such that

$$f(\alpha x + (1 - \alpha y)) \leq \alpha f(x) + (1 - \alpha) f(y) - \frac{\gamma}{2} \alpha (1 - \alpha) \|x - y\|_2^2.$$

For f strongly convex and *smooth*, we have

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} \gamma \|y - x\|^2$$

Preliminaries: Optimality and Subgradients

For smooth f, first-order optimal (stationary) point satisfies $\nabla f(x^*) = 0$.
For convex f, this condition is sufficient for x^* to be optimal.

For smooth f, second-order necessary condition for optimality is $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succeq 0$

When ψ is convex but not necessarily smooth, define subgradient of ψ at x to be vector g such that

$$\psi(z) \geq \psi(x) + g^T(z - x), \quad \text{for all } z.$$

The subdifferential $\partial\psi(x)$ is the set of all subgradients of ψ at x.

A sufficient condition for x^* to be a minimizer of convex ψ is $0 \in \partial\psi(x^*)$.

Preliminaries: Smoothness and Consequences

A common assumption is that $f : \mathbb{R}^n \to \mathbb{R}$ has Lipschitz continuous gradients:

$$\|\nabla f(x) - \nabla f(z)\| \leq L\|x - z\|, \quad \text{for some } L > 0.$$

A consequence is that we can bound f above by a simple quadratic:

$$f(z) \leq f(x) + \nabla f(x)^T(z - x) + \frac{1}{2}L\|z - x\|^2.$$

Moreau Envelope, Prox Operations

For a closed proper convex function h and a positive scalar λ, the Moreau envelope is

$$M_{\lambda,h}(x) := \inf_u \left\{ h(u) + \frac{1}{2\lambda}\|u - x\|^2 \right\} = \frac{1}{\lambda} \inf_u \left\{ \lambda h(u) + \frac{1}{2}\|u - x\|^2 \right\}.$$

The prox operation for a function λh is the value that achieves the min in the Moreau envelope definition:

$$\text{prox}_{\lambda h}(x) := \arg\min_u \left\{ \lambda h(u) + \frac{1}{2}\|u - x\|^2 \right\}.$$

Moreau envelope is a smoothed version of h, with finite value for all x. Differentiable everywhere:

$$\nabla M_{\lambda,h}(x) = \frac{1}{\lambda}(x - \text{prox}_{\lambda h}(x)).$$

x^* is a minimizer of h if and only if it is a minimizer of $M_{\lambda,h}$.

First-Order Method: Steepest Descent

For $\min_x f(x)$, steepest descent steps in the negative gradient direction:

$$x^{k+1} \leftarrow x^k - \alpha_k \nabla f(x^k).$$

Classical analysis applies for f with Lipschitz gradients.

Setting $\alpha_k \equiv 1/L$, we have using the quadratic upper bound that

$$\begin{aligned} f(x^{k+1}) &= f(x^k - (1/L)\nabla f(x^k)) \\ &\leq f(x^k) - \frac{1}{L}\|\nabla f(x^k)\|^2 + \frac{1}{2}L\frac{1}{L^2}\|\nabla f(x^k)\|^2 \\ &\leq f(x^k) - \frac{1}{2L}\|\nabla f(x^k)\|^2, \end{aligned}$$

so we get a guaranteed decrease in f whenever $\nabla f(x_k) \neq 0$, i.e. when x_k is not stationary.

Convergence: General f

When f is bounded below by a constant $\bar{f}$, then steepest descent with $\alpha_k \equiv 1/L$ satisfies for any $T \geq 1$ that

$$\min_{0 \leq k \leq T-1} \|\nabla f(x^k)\| \leq \sqrt{\frac{2L[f(x^0) - f(x^T)]}{T}} \leq \sqrt{\frac{2L[f(x^0) - \bar{f}]}{T}}.$$

Convergence to stationary point at a $1/\sqrt{T}$ rate.

Proof: Rearrange expression above and use telescoping sum:

$$\sum_{k=0}^{T-1} \|\nabla f(x^k)\|^2 \leq 2L \sum_{k=0}^{T-1} [f(x^k) - f(x^{k+1})] = 2L[f(x^0) - f(x^T)].$$

Use

$$\min_{0 \leq k \leq T-1} \|\nabla f(x^k)\| = \sqrt{\min_{0 \leq k \leq T-1} \|\nabla f(x^k)\|^2} \leq \sqrt{\frac{1}{T} \sum_{k=0}^{T-1} \|\nabla f(x^k)\|^2}.$$

Convergence: Convex f

For f convex, same method yields a $1/T$ rate (in a different measure):

$$f(x^T) - f^* \leq \frac{L}{2T}\|x^0 - x^*\|^2.$$

Proof: By convexity of f, we have $f(x^*) \geq f(x^k) + \nabla f(x^k)^T(x^* - x^k)$, so

$$\begin{aligned} f(x^{k+1}) &\leq f(x^*) + \nabla f(x^k)^T(x^k - x^*) - \frac{1}{2L}\|\nabla f(x^k)\|^2 \\ &= f(x^*) + \frac{L}{2}\left(\|x^k - x^*\|^2 - \left\|x^k - x^* - \frac{1}{L}\nabla f(x^k)\right\|^2\right) \\ &= f(x^*) + \frac{L}{2}\left(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2\right). \end{aligned}$$

Sum over $k = 0, 1, 2, \ldots, T-1$, and telescope: we have

$$\begin{aligned}\sum_{k=0}^{T-1}(f(x^{k+1}) - f^*) &\leq \frac{L}{2}\sum_{k=0}^{T-1}\left(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2\right) \\ &= \frac{L}{2}\left(\|x^0 - x^*\|^2 - \|x^T - x^*\|^2\right) \\ &\leq \frac{L}{2}\|x^0 - x^*\|^2.\end{aligned}$$

Since $\{f(x^k)\}$ is a nonincreasing sequence, we have

$$f(x^T) - f(x^*) \leq \frac{1}{T}\sum_{k=0}^{T-1}(f(x^{k+1}) - f^*) \leq \frac{L}{2T}\|x^0 - x^*\|^2,$$

as required.

Convergence: Strongly Convex f

If f is strongly convex with modulus γ and smooth, we have linear convergence:

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\gamma}{L}\right)(f(x^k) - f(x^*)).$$

Proof: Minimize both sides of the definition of strong convexity w.r.t. y to obtain

$$\begin{aligned}
&\min_y f(y) \geq \min_y f(x) + \nabla f(x)^T(y-x) + \frac{\gamma}{2}\|y-x\|^2 \\
\Rightarrow \quad & f(x^*) \geq f(x) - \nabla f(x)^T\left(\frac{1}{\gamma}\nabla f(x)\right) + \frac{\gamma}{2}\left\|\frac{1}{\gamma}\nabla f(x)\right\|^2 \\
\Rightarrow \quad & f(x^*) \geq f(x) - \frac{1}{2\gamma}\|\nabla f(x)\|^2.
\end{aligned}$$

By rearrangement, we obtain

$$\|\nabla f(x)\|^2 \geq 2\gamma[f(x) - f(x^*)]. \tag{1}$$

Substitute into the basic decrease condition to obtain

$$\begin{aligned} f(x^{k+1}) &= f\left(x^k - \frac{1}{L}\nabla f(x^k)\right) \\ &\leq f(x^k) - \frac{1}{2L}\|\nabla f(x^k)\|^2 \\ &\leq f(x^k) - \frac{\gamma}{L}(f(x^k) - f^*). \end{aligned}$$

Subtracting f^* from both sides of this inequality, we obtain

$$f(x^{k+1}) - f^* \leq \left(1 - \frac{\gamma}{L}\right)(f(x^k) - f^*).$$

Convergence and Complexity

The global convergence rates $(1/\sqrt{T},\ 1/T,\ (1-\delta)^T)$ can be expressed alternatively as complexities: The number of iterations required to get a factor of ϵ reduction in the quantity at hand.

- $1/\sqrt{T}$ convergence rate $\Rightarrow$ $1/\epsilon^2$ iterations;
- $1/T$ convergence $\Rightarrow$ $1/\epsilon$ iterations;
- $(1-\delta)^T \Rightarrow \log\epsilon/\log(1-\delta)$ iterations. Since for small positive δ we have $\log(1-\delta) \approx -\delta$, this complexity is approximately $|\log\epsilon|/\delta$.

Regularized Optimization: Prox-Gradient Methods

$$\min_{x \in \mathbb{R}^n} \phi(x) := f(x) + \lambda \psi(x),$$

where f is a smooth convex function, ψ is a convex regularizer, $\lambda \geq 0$ is aregularization parameter.

- Take gradient descent step in f;
- Use prox operator to take $\lambda\psi$ explicitly into account.

$$x^{k+1} := \text{prox}_{\alpha_k \lambda \psi}(x^k - \alpha_k \nabla f(x^k)),$$

or equivalently

$$x^{k+1} := \arg\min_z \nabla f(x^k)^T (z - x^k) + \frac{1}{2\alpha_k} \|z - x^k\|^2 + \lambda \psi(z).$$

Convergence of Prox-Gradient

The details are technical, but the convergence behavior is similar to steepest descent on a smooth function. With $\alpha_k \equiv 1/L$, we have

$$\phi(x^k) - \phi^* \leq \frac{L\|x^0 - x^*\|^2}{2k},$$

which is the same $1/k$ rate as steepest descent.

The prox operation can be added to other algorithms for minimization of smooth f, such as coordinate descent and accelerated gradient.

Prox-Gradient for the Constrained Problem

We can express the constrained problem as a regularized optimization problem:

$$\min_{x \in \Omega} f(x) \quad \Leftrightarrow \quad \min_{x} f(x) + I_\Omega(x),$$

where $I_\Omega(x)$ is the indicator function that takes the value 0 when $x \in \Omega$ and ∞ otherwise.

The operation $\text{prox}_{I_\Omega}(z)$ is identical to projection onto Ω, which is denoted by $P_\Omega(z)$ and defined by

$$P_\Omega(z) := \arg\min_{y \in \Omega} \tfrac{1}{2}\|y - z\|_2^2.$$

Thus each step of prox-gradient in this case is

$$x^{k+1} = P_\Omega(x^k - \alpha_k \nabla f(x^k)),$$

which is the well known gradient projection algorithm.

Accelerated Gradient and Momentum

Accelerated gradient methods [Nesterov, 1983] highly influential.

Fundamental idea: **Momentum!** Search direction at iteration k depends on the latest gradient $\nabla f(x^k)$ and also the search direction at iteration $k-1$, which encodes gradient information from earlier iterations.

Heavy-ball & conjugate gradient (incl. nonlinear CG) also use momentum.

Heavy-Ball for $\min_x f(x)$:

$$x^{k+1} = x^k - \alpha \nabla f(x^k) + \beta(x^k - x^{k-1}).$$

Nesterov's optimal method:

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k + \beta_k(x^k - x^{k-1})) + \beta_k(x^k - x^{k-1}).$$

Accelerated Gradient and Momentum

Accelerated gradient methods [Nesterov, 1983] highly influential.

Fundamental idea: **Momentum!** Search direction at iteration k depends on the latest gradient $\nabla f(x^k)$ and also the search direction at iteration $k-1$, which encodes gradient information from earlier iterations.

Heavy-ball & conjugate gradient (incl. nonlinear CG) also use momentum.

Heavy-Ball for $\min_x f(x)$:

$$x^{k+1} = x^k - \alpha \nabla f(x^k) + \beta(x^k - x^{k-1}).$$

Nesterov's optimal method:

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k + \beta_k(x^k - x^{k-1})) + \beta_k(x^k - x^{k-1}).$$

Nesterov Acceleration

Set $\alpha_k = 1/L$ and introduce an intermediate variable $y^k \in \mathbb{R}^n$ (with $y^0 = x^0$) to rewrite the Nesterov optimal update as

$$x^{k+1} = y^k - \frac{1}{L}\nabla f(y^k),$$
$$y^{k+1} = x^{k+1} + \beta_{k+1}(x^{k+1} - x^k), \quad k = 0, 1, 2, \ldots$$

For general convex f, β_k is defined via another scalar sequence λ_k:

$$\lambda_0 = 0, \quad \lambda_{k+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4\lambda_k^2}\right), \quad \beta_k = \frac{\lambda_k - 1}{\lambda_{k+1}}.$$

Since $\lambda_k \geq 1$ for $k \geq 1$, we have $\beta_{k+1} \geq 0$.

Convergence:

$$f(x^T) - f^* \leq \frac{2L\|x^0 - x^*\|^2}{(T+1)^2}, \quad T = 1, 2, \ldots .$$

A $1/T^2$ rate! Rather than the $1/T$ rate of steepest descent.

Proof: Technical! See the sources.

Nesterov Acceleration: Strongly Convex

When f is strongly convex with modulus γ, another choice of β_k is used:

$$\beta_{k+1} \equiv \frac{\sqrt{L} - \sqrt{\gamma}}{\sqrt{L} + \sqrt{\gamma}}.$$

Convergence:

$$f(x^T) - f(x^*) \leq \frac{L+\gamma}{2}\|x^0 - x^*\|^2 \left(1 - \sqrt{\frac{\gamma}{L}}\right)^T, \quad T = 1, 2, \ldots.$$

This yields a complexity of $O(\sqrt{L/\gamma}\log\epsilon)$ to achieve an ϵ-optimal solution.

Improvement over steepest descent, which has complexity $O((L/\gamma)\log\epsilon)$.

Conjugate gradient applied to convex quadratic f has similar asymptotic rates.

Full Gradient: Does It Make Sense?

The methods above, based on full gradients, are useful for some problems in which X is a matrix, in which full gradients are practical to compute.

- Matrix completion, including explicitly parametrized problems with $X = LR^T$ or $X = ZZ^T$; nonnegative matrix factorization.
- Subspace identification;
- Sparse covariance estimation;

They are less appealing when the objective is the sum of m terms, with m large. To calculate

$$\nabla f(x) = \frac{1}{m} \sum_{j=1}^{m} \nabla f_j(x),$$

generally need to make a full pass through the data.

Often not practical for massive data sets. But can be hybridized with stochastic gradient methods (see below).

Stochastic Gradient (SG)

For $f(x) = (1/m)\sum_{j=1}^{m} f_j(x)$, iteration k has the form:

- Choose $j_k \in \{1, 2, \ldots, m\}$ uniformly at random;
- Set $x^{k+1} \leftarrow x^k - \alpha_k \nabla f_{j_k}(x^k)$.

$\nabla f_{j_k}(x^k)$ is a proxy for $\nabla f(x^k)$ but it depends on just one data item a_{j_k} and is much cheaper to evaluate.

Unbiased — $\mathbb{E}_j \nabla f_j(x) = \nabla f(x)$ — but the variance may be very large.

- Average the iterates for more robust convergence:

$$\bar{x}^k = \frac{\sum_{\ell=1}^{k} \gamma_\ell x^\ell}{\sum_{\ell=0}^{k} \gamma_\ell}, \text{where } \gamma_\ell \text{ are positive weights.}$$

- Minibatch: Use a set $J_k \subset \{1, 2, \ldots, m\}$ rather than a single item. (Smaller variance in the gradient estimate.)

Stochastic Gradient (SG)

Convergence results for h_j convex require bounds on the variance of the gradient estimate:

$$\frac{1}{m}\sum_{j=1}^{m}\|\nabla f_j(x)\|_2^2 \leq B^2 + L_g\|x - x^*\|^2.$$

Analyze expected convergence, e.g. $\mathbb{E}(f(x^k) - f^*)$ or $\mathbb{E}(f(\bar{x}^k) - f^*)$, where the expectation is over the sequence of indices $j_0, j_1, j_2, \ldots$.

Sample results:

- f strongly convex, $\alpha_k \sim 1/k$: $\mathbb{E}(f(x^k) - f^*) = O(1/k)$;
- h convex, $\alpha_k \sim 1/\sqrt{k}$: $\mathbb{E}(f(\bar{x}^k) - H^*) = O(1/\sqrt{k})$.
- $B = 0$, f strongly convex, $\alpha_k = \text{const}$: $\mathbb{E}(\|x^k - x^*\|_2^2) = O((1-\delta)^k)$ for some $\delta \in (0, 1)$.

Generalizes beyond finite sums, to $f(x) = \mathbb{E}_\xi f(x; \xi)$, where ξ is random.

Stochastic Gradient Variant: Adam

[Kingma and Ba, 2015] describe a modification of SGD which applies a diagonal scaling to the update g_k.

The scaling is computed from a weighted average of components of g_k, and their squares, from previous iterations.

Echoes of a diagonal scaling of the gradient direction $\nabla f(x_k)$: *scaled steepest descent*, which can also be thought of as a Newton-type method with a diagonal approximation to the Hessian $\nabla^2 f(x_k)$.

Convergence theory (of variations of Adam) has been the subject of much study.

The paper has 115,000 citations to date.

This method has become a juggernaut. Versions of it are used almost universally in practice.

Hybrids of Full Gradient and Stochastic Gradient

Stabilize SG by hybridizing with steepest descent (full-gradient). Get *linear convergence* for strongly convex functions, sublinear for weakly convex.

SAG: [LeRoux et al., 2012] Maintain approximations g_j to ∇h_j, use search direction $-(1/m)\sum_{j=1}^m g_j$. At iteration k, choose j_k at random, and update $g_{j_k} = \nabla h_{j_k}(x^k)$.

SAGA: [Defazio et al., 2014] Similar to SAG, but use search direction

$$-\nabla h_{j_k}(x^k) + g_{j_k} - \frac{1}{m}\sum_{j=1}^m g_j.$$

SVRG: [Johnson and Zhang, 2013] Similar again, but periodically do a full gradient evaluation to refresh all g_j.

Too much storage, **BUT** when h_j have the "ERM" form $h_j(a_j^T x)$ (linear least squares, linear SVM), all gradients can be stored in a scalar:

$$\nabla_x h_j(a_j^T x) = a_j h_j'(a_j^T x).$$

Coordinate Descent (CD) Framework

... for smooth unconstrained minimization: $\min_x H(x)$:

Set Choose $x^1 \in \mathbb{R}^n$;
for $\ell = 0, 1, 2, \ldots$ (epochs) **do**
  for $j = 1, 2, \ldots, n$ (inner iterations) **do**
    Define $k = \ell n + j$
    Choose index $i = i(\ell, j) \in \{1, 2, \ldots, n\}$;
    Choose $\alpha_k > 0$;
    $x^{k+1} \leftarrow x^k - \alpha_k \nabla_i H(x^k) e_i$;
  end for
end for

where

- $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^T$: the ith coordinate vector;
- $\nabla_i H(x) = i$th component of the gradient $\nabla H(x)$;
- $\alpha_k > 0$ is the step length.

CD Variants

- CCD (Cyclic CD): $i(\ell, j) = j$.
- RCD (Randomized CD a.k.a. Stochastic CD): $i(\ell, j)$ is chosen uniformly at random from $\{1, 2, \ldots, n\}$.
- RPCD (Randomized Permutations CD):
  - At the start of epoch ℓ, we choose a random permutation of $\{1, 2, \ldots, n\}$, denoted by π_ℓ.
  - Index $i(\ell, j)$ is chosen to be the jth entry in π_ℓ.

Important quantities in analysis:

- $L_{\max}$: componentwise Lipschitz constant for ∇H:

$$|\nabla_i H(x + te_i) - \nabla_i H(x)| \leq L_i |t|, \quad L_{\max} = \max_{i=1,2,\ldots,n} L_i.$$

- L: usual Lipschitz constant: $|\nabla H(x + d) - \nabla H(x)| \leq L\|d\|$.
- Lojasiewicz constant μ: $\|\nabla H(x)\|^2 \geq 2\mu[H(x) - H^*]$

Randomized CD Convergence

Of the three variants, convergence of the randomized form has by far the most elementary analysis [Nesterov, 2012].

Get convergence rates for quantity $\phi_k := \mathbb{E}(H(x^k) - x^*)$.

$$\mu > 0: \quad \phi_{k+1} \leq \left(1 - \frac{\mu}{nL_{\max}}\right)\phi_k, \quad k = 1, 2, \ldots,$$

$$\mu = 0: \quad \phi_k \leq \frac{2nL_{\max}R_0^2}{k}, \qquad k = 1, 2, \ldots,$$

where R_0 bounds distance from x^0 to solution set.

If the economics of evaluating gradient components are right, this can be a factor $L/L_{\max}$ faster than full-gradient steepest descent!

This ratio is in range $[1, n]$. Maximized by $H(x) = (\mathbf{1}\mathbf{1}^T)x$.

Functions like this are good cases for RCD and RPCD, which are much faster than CCD or steepest descent.

Cyclic and Random-Permutations CD Convergence

Analysis of [Beck and Tetruashvili, 2013] treats CCD as an approximate form of Steepest Descent, bounding improvement in f over one cycle in terms of the gradient at the start of the cycle.

Get linear and sublinear rates that are slower than both RCD and Steepest Descent. This analysis is fairly tight — recent analysis of [Sun and Ye, 2016] confirms slow rates on a worst-case example.

Same analysis applies to Randomized Permutations (RPCD), but practical results for RPCD are much better, and usually at least as good as RCD.

We can explain good behavior of RPCD now, showing that it tracks RCD not CCD. [Lee and Wright, 2018, Wright and Lee, 2020]

CD Extensions

- Block CD: Replace single component i by block $I \subset \{1, 2, \ldots, n\}$.
- Dual nonlinear SVM [Platt, 1999]. Choose *two* components of α per iteration, to stay feasible w.r.t. constraint $y^T \alpha = 0$.
- Can be accelerated (efficiently) using "Nesterov" techniques: [Nesterov, 2012, Lee and Sidford, 2013].
- Adaptable to the separable regularized case $H(x) + \lambda \Omega(x)$.
- Parallel asynchronous variants, suitable for implementation on shared-memory multicore computers, have been proposed and analyzed. [Bertsekas and Tsitsiklis, 1989, Liu and Wright, 2015, Liu et al., 2015]

Conditional Gradient / "Frank-Wolfe"

$$\min_{x \in \Omega} f(x),$$

where f is a convex function and Ω is a closed, bounded, convex set.

Start at $x_0 \in \Omega$. At iteration k:

$$v_k := \arg\min_{v \in \Omega} v^T \nabla f(x_k);$$

$$x_{k+1} := x_k + \alpha_k (v_k - x_k), \quad \alpha_k = \frac{2}{k+2}.$$

- Potentially useful when it is easy to minimize a linear function over the *original* constraint set Ω;
- Admits an elementary convergence theory: $1/k$ sublinear rate.
- Same convergence rate holds if we use a line search for α_k.

Revived by [Jaggi, 2013].

Augmented Lagrangian

Consider the linearly constrained problem,

$$\min f(x) \text{ s.t. } Ax = b,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex.

Define the Lagrangian function:

$$\mathcal{L}(x, \lambda) := f(x) + \lambda^T (Ax - b).$$

x^* is a solution if and only if there exists a vector of Lagrange multipliers $\lambda^* \in \mathbb{R}^m$ such that

$$-A^T \lambda^* \in \partial f(x^*), \quad Ax^* = b,$$

or equivalently:

$$0 \in \partial_x \mathcal{L}(x^*, \lambda^*), \quad \nabla_\lambda \mathcal{L}(x^*, \lambda^*) = 0.$$

Augmented Lagrangian

The augmented Lagrangian is (with $\rho > 0$)

$$\mathcal{L}(x, \lambda; \rho) := \underbrace{f(x) + \lambda^T(Ax - b)}_{\text{Lagrangian}} + \underbrace{\frac{\rho}{2}\|Ax - b\|_2^2}_{\text{“augmentation”}}.$$

Basic Augmented Lagrangian (a.k.a. method of multipliers) is

$$x_k = \arg\min_x \mathcal{L}(x, \lambda_{k-1}; \rho);$$
$$\lambda_k = \lambda_{k-1} + \rho(Ax_k - b);$$

[Hestenes, 1969, Powell, 1969]

Some constraints on x (such as $x \in \Omega$) can be handled explicitly:

$$x_k = \arg\min_{x \in \Omega} \mathcal{L}(x, \lambda_{k-1}; \rho);$$
$$\lambda_k = \lambda_{k-1} + \rho(Ax_k - b);$$

Alternating Direction Method of Multipliers (ADMM)

$$\min_{(x\in\Omega_x, z\in\Omega_z)} f(x) + h(z) \quad \text{s.t.} \quad Ax + Bz = c,$$

for which the Augmented Lagrangian is

$$\mathcal{L}(x, z, \lambda; \rho) := f(x) + h(z) + \lambda^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax - Bz - c\|_2^2.$$

Standard AL would minimize $\mathcal{L}(x, z, \lambda; \rho)$ w.r.t. (x, z) jointly. However, since coupled in the quadratic term, separability is lost.

In ADMM, minimize over x and z separately and sequentially:

$$x_k = \arg\min_{x\in\Omega_x} \mathcal{L}(x, z_{k-1}, \lambda_{k-1}; \rho);$$

$$z_k = \arg\min_{z\in\Omega_z} \mathcal{L}(x_k, z, \lambda_{k-1}; \rho);$$

$$\lambda_k = \lambda_{k-1} + \rho(Ax_k + Bz_k - c).$$

Extremely useful framework for many data analysis / learning settings. Major references: [Eckstein and Bertsekas, 1992, Boyd et al., 2011]

Convex-Concave Min-Max: Primal-Dual Algorithms

Formulation (reminder):

$$\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^n} L(x, y) \qquad \text{(Min-Max)}$$

$$\begin{aligned} L(x, y) &= \sum_{i=1}^{n} \left[\langle A_i x, y^{(i)} \rangle - h_i^*(y^{(i)}) \right] + g(x) \\ &= \langle Ax, y \rangle - h^*(y) + g(x), \end{aligned}$$

Gradient Ascent-Descent (GDA):

$$\begin{aligned} \bar{x}_{k+1} &= \text{prox}_{\tau, g}(\bar{x}_k - \tau A^\top \bar{y}_k) \\ \bar{y}_{k+1} &= \text{prox}_{\sigma, h^*}(\bar{y}_k + \sigma A \bar{x}_{k+1}), \end{aligned} \qquad \text{(GDA)}$$

for positive step sizes τ and σ. [1]

[1] $\mathrm{T} = \tau I$ in our earlier definition of prox.

Primal-Dual Algorithms

Primal-Dual Hybrid Gradient (PDHG) [Chambolle and Pock, 2011] uses extrapolation in the x step:

$$\begin{aligned} \bar{x}_{k+1} &= \text{prox}_{\tau,g}(\bar{x}_k - \tau A^\top (2\bar{y}_k - \bar{y}_{k-1})) \\ \bar{y}_{k+1} &= \text{prox}_{\sigma,h^*}(\bar{y}_k + \sigma A \bar{x}_{k+1}), \end{aligned} \tag{PDHG}$$

Equivalent form of PDHG:

$$\bar{x}_{k+1} = \text{prox}_{\tau,g}(\hat{x}_k - \tau A^\top \bar{y}_k) \tag{2a}$$

$$\bar{y}_{k+1} = \text{prox}_{\sigma,h^*}(\bar{y}_k + \sigma A \bar{x}_{k+1}) \tag{2b}$$

$$\hat{x}_{k+1} = \bar{x}_{k+1} - \tau A^\top (\bar{y}_{k+1} - \bar{y}_k). \tag{2c}$$

[Chambolle and Pock, 2011] discusses connections to Douglas-Rachford, Extrapolated gradient, ADMM.

[Aragón-Artacho et al., 2020] show application of DR to finding intersection of sets, pictures show the benefits of extrapolation.

Algorithms: Additional Features

Theoretical convergence / complexity properties of these algorithms can be improved (in some cases, including strong convexity / concavity and sparsity) by adding extra features.

- Coordinate descent: e.g. update random element(s) of y in (2b) instead of the whole vector.
- Variance Reduction: Adjust the update formula for x to account for noise arising from coordinate update of y.
- Dual Averaging: At step k, use a gradient term that is a weighted average over all previous iterations.
- Importance sampling: Apply different weights to different components of each update (e.g. weight matrix T in definition of prox).
- Iterate averaging: Output a weighted average of iterates, rather than the final iterate for x.

Some are used by PURE-CD [Alacaoglu et al., 2022] and VRPDA^2 [Song et al., 2021].

Not Discussed!

Many interesting topics not mentioned, including

- Newton methods.
- quasi-Newton methods.
- Linear equations $Ax = b$: Kaczmarz algorithms.
- Image and video processing: denoising and deblurring.
- Graphs: detect structure and cliques, consensus optimization, ...
- Integer and combinatorial formulations.
- Parallel variants: synchronous and asynchronous.
- Online learning.

References I


Alacaoglu, A., Cevher, V., and Wright, S. J. (2022).
On the complexity of a practical primal-dual coordinate method.


Aragón-Artacho, F. J., Campoy, R., and Tam, M. K. (2020).
The Douglas–Rachford algorithm for convex and nonconvex feasibility problems.
Mathematical Methods of Operations Research, 91:201–240.


Beck, A. and Tetruashvili, L. (2013).
On the convergence of block coordinate descent type methods.
SIAM Journal on Optimization, 23(4):2037–2060.


Bertsekas, D. P. and Tsitsiklis, J. N. (1989).
Parallel and Distributed Computation: Numerical Methods.
Prentice-Hall, Inc., Englewood Cliffs, New Jersey.


Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011).
Distributed optimization and statistical learning via the alternating direction methods of multipliers.
Foundations and Trends in Machine Learning, 3(1):1–122.


Chambolle, A. and Pock, T. (2011).
A first-order primal-dual algorithm for convex problems with applications to imaging.
Journal of Mathematical Imaging and Vision, 40(1):120–145.

References II

Defazio, A., Bach, F., and Lacoste-Julien, S. (2014).
SAGA: a fast incremental gradient method with support for non-strongly composite convex objectives.
In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 1646–1654. Curran Associates, Inc.

Dunn, J. C. (1979).
Rates of convergence for conditional gradient algorithms near singular and nonsingular extremals.
SIAM Journal on Control and Optimization, 17(2):187–211.

Eckstein, J. and Bertsekas, D. P. (1992).
On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators.
Mathematical Programming, 55:293–318.

Frank, M. and Wolfe, P. (1956).
An algorithm for quadratic programming.
Naval Research Logistics Quarterly, 3:95–110.

Hestenes, M. R. (1969).
Multiplier and gradient methods.
Journal of Optimization Theory and Applications, 4:303–320.

References III


Jaggi, M. (2013).
Revisiting Frank-Wolfe: Projection-free sparse convex optimization.
In *Proceedings of the 30th International Conference on Machine Learning*.


Johnson, R. and Zhang, T. (2013).
Accelerating stochastic gradient descent using predictive variance reduction.
In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 315–323. Curran Associates, Inc.


Kingma, D. P. and Ba, J. (2015).
Adam: A method for stochastic optimization.
In *Proceedings of International Conference on Learning Representations*.


Lee, C.-p. and Wright, S. J. (2018).
Random permutations fix a worst case for cyclic coordinate descent.
IMA Journal of Numerical Analysis, 39:1246–1275.


Lee, Y. T. and Sidford, A. (2013).
Efficient accelerated coordinate descent methods and faster algorihtms for solving linear systems.
In *54th Annual Symposium on Foundations of Computer Science*, pages 147–156.

References IV


LeRoux, N., Schmidt, M., and Bach, F. (2012).
A stochastic gradient methods with an exponential convergence rate for finite training sets.
In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 25*, pages 2663–2671. Curran Associates, Inc.


Liu, J. and Wright, S. J. (2015).
Asynchronous stochastic coordinate descent: Parallelism and convergence properties.
SIAM Journal on Optimization, 25(1):351–376.


Liu, J., Wright, S. J., Ré, C., Bittorf, V., and Sridhar, S. (2015).
An asynchronous parallel stochastic coordinate descent algorithm.
Journal of Machine Learning Research, 16:285–322.
arXiv:1311.1873.


Nesterov, Y. (1983).
A method for unconstrained convex problem with the rate of convergence $O(1/k^2)$.
Doklady AN SSSR, 269:543–547.


Nesterov, Y. (2012).
Efficiency of coordinate descent methods on huge-scale optimization problems.
SIAM Journal on Optimization, 22:341–362.

References V

Platt, J. C. (1999).
Fast training of support vector machines using sequential minimal optimization.
In Schölkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA. MIT Press.

Powell, M. J. D. (1969).
A method for nonlinear constraints in minimization problems.
In Fletcher, R., editor, *Optimization*, pages 283–298. Academic Press, New York.

Robbins, H. and Monro, S. (1951).
A stochastic approximation method.
Annals of Mathematical Statistics, 22(3).

Song, C., Wright, S. J., and Diakonikolas, J. (2021).
Variance reduction via primal-dual accelerated dual averaging for nonsmooth convex finite-sums.
In *International Conference on Machine Learning*, pages 9824–9834. PMLR.

Sun, R. and Ye, Y. (2016).
Worst-case complexity of cyclic coordinate descent: $o(n^2)$ gap with randomized version.
Technical Report arXiv:1604.07130, Department of Management Science and Engineering, Stanford University, Stanford, California.

References VI

Wright, S. J. and Lee, C.-p. (2020).
Analyzing random permutations for cyclic coordinate descent.
Mathematics of Computation, 89:2217–2248.