First-Order Optimization Algorithms for Machine Learning

Proximal-Gradient

Mark Schmidt

University of British Columbia

Summer 2020

Solving Problems with Simple Regularizers

- We were discussing how to solve non-smooth L1-regularized objectives like

$$\underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2}\|Xw - y\|^2 + \lambda\|w\|_1.$$

- Use our trick to formulate as a quadratic program?
  - $O(d^2)$ or worse.
- Make a smooth approximation to the L1-norm?
  - Destroys sparsity (we'll again just have one subgradient at zero).
- Use a subgradient method?
  - Needs $O(1/\epsilon)$ iterations even in the strongly-convex case.
- Transform to "smooth f with simple constraints" and use projected-gradient?
  - Works well, but increases problem size and destroys strong-convexity.

- For "simple" regularizers, proximal-gradient methods don't have these drawbacks

Should we use projected-gradient for non-smooth problems?

- Some non-smooth problems can be turned into smooth problems with simple constraints.

- But transforming might make problem harder:
  - For L1-regularization least squares,

$$\operatorname*{argmin}_{w \in \mathbb{R}^d} \frac{1}{2}\|Xw - y\|^2 + \lambda \|w\|_1,$$

  we can re-write as a smooth problem with bound constraints,

$$\operatorname*{argmin}_{w_+ \geq 0,\ w_- \geq 0} \|X(w_+ - w_-) - y\|^2 + \lambda \sum_{j=1}^{d} (w_+ + w_-).$$

  - Doubles the number of variables.
  - Transformed problem is not strongly convex even if the original was.

Outline

1. Proximal-Gradient
2. Active-Set Complexity

Quadratic Approximation View of Gradient Method

- We want to solve a smooth optimization problem:

$$\operatorname*{argmin}_{w \in \mathbb{R}^d} f(w).$$

- Iteration w^k works with a quadratic approximation to f:

$$f(v) \approx f(w^k) + \nabla f(w^k)^\top (v - w^k) + \frac{1}{2\alpha_k}\|v - w^k\|^2,$$

$$w^{k+1} \in \operatorname*{argmin}_{v \in \mathbb{R}^d} \left\{ f(w^k) + \nabla f(w^k)^\top (v - w^k) + \frac{1}{2\alpha_k}\|v - w^k\|^2 \right\}.$$

We can equivalently write this as the quadratic optimization:

$$w^{k+1} \in \operatorname*{argmin}_{v \in \mathbb{R}^d} \left\{ \frac{1}{2}\|v - (w^k - \alpha_k \nabla f(w^k))\|^2 \right\},$$

and the solution is the gradient algorithm:

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

Quadratic Approximation View of Proximal-Gradient Method

- We want to solve a smooth plus non-smooth optimization problem:

$$\operatorname*{argmin}_{w\in\mathbb{R}^d} f(w)+r(w).$$

- Iteration w^k works with a quadratic approximation to f:

$$f(v)+r(v) \approx f(w^k) + \nabla f(w^k)^\top (v-w^k) + \frac{1}{2\alpha_k}\|v-w^k\|^2 + r(v),$$

$$w^{k+1} \in \operatorname*{argmin}_{v\in\mathbb{R}^d} \left\{ f(w^k) + \nabla f(w^k)^\top (v-w^k) + \frac{1}{2\alpha_k}\|v-w^k\|^2 + r(v)\right\}.$$

  We can equivalently write this as the proximal optimization:

$$w^{k+1} \in \operatorname*{argmin}_{v\in\mathbb{R}^d} \left\{ \frac{1}{2}\|v-(w^k-\alpha_k\nabla f(w^k))\|^2 + \alpha_k r(v)\right\},$$

  and the solution is the proximal-gradient algorithm:

$$w^{k+1} = \text{prox}_{\alpha_k r}[w^k - \alpha_k \nabla f(w^k)].$$

Proximal-Gradient for L1-Regularization

- The proximal operator for L1-regularization when using step-size α_k,

$$\text{prox}_{\alpha_k\lambda\|\cdot\|_1}[w^{k+\frac{1}{2}}] \in \underset{v\in\mathbb{R}^d}{\text{argmin}}\left\{\frac{1}{2}\|v - w^{k+\frac{1}{2}}\|^2 + \alpha_k\lambda\|v\|_1\right\},$$

  involves solving a simple (strongly-convex) 1D problem for each variable j:

$$w_j^{k+1} \in \underset{v_j\in\mathbb{R}}{\text{argmin}}\left\{\frac{1}{2}(v_j - w_j^{k+\frac{1}{2}})^2 + \alpha_k\lambda|v_j|\right\}.$$

- We can find the argmin by finding the unique v_j with 0 in the sub-differential.
- The solution is given by applying "soft-threshold" operation:
  1. If $|w_j^{k+\frac{1}{2}}| \leq \alpha_k\lambda$, set $w_j^{k+1} = 0$.
  2. Otherwise, shrink $|w_j^{k+\frac{1}{2}}|$ by $\alpha_k\lambda$.

Proximal-Gradient for L1-Regularization

- An example sof-threshold operator on absolute value with $\alpha_k \lambda = 1$:

Input	Threshold	Soft-Threshold
0.6715	0	0
−1.2075	−1.2075	−0.2075
0.7172	0	0
1.6302	1.6302	0.6302
0.4889	0	0

- Symbolically, the soft-threshold operation computes

$$w_j^{k+1} = \underbrace{\text{sign}(w^{k+\frac{1}{2}})}_{-1 \text{ or } +1} \max\left\{0, |w_j^{k+\frac{1}{2}}| - \alpha_k \lambda\right\}.$$

- Has the nice property that iterations w^k are sparse.
  - Compared to subgradient method which wouldn't give exact zeroes.

Proximal-Gradient Method

- So proximal-gradient step takes the form:

$$w^{k+\frac{1}{2}} = w^k - \alpha_k \nabla f(w^k)$$
$$w^{k+1} = \operatorname*{argmin}_{v \in \mathbb{R}^d} \left\{ \frac{1}{2}\|v - w^{k+\frac{1}{2}}\|^2 + \alpha_k r(v) \right\}.$$

- Second part is called the proximal operator with respect to a convex $\alpha_k r$.
  - We say that r is simple if you can efficiently compute proximal operator.

- Very similar properties to projected-gradient when ∇f is Lipschitz-continuous:
  - Guaranteed improvement for $\alpha < 2/L$, practical backtracking methods work better.
  - Solution is a fixed point, $w^* = \text{prox}_r[w^* - \alpha \nabla f(w^*)]$ for any $\alpha > 0$.
  - If f is strongly-convex then

$$F(w^k) - F^* \leq \left(1 - \frac{\mu}{L}\right)^k \left[F(w^0) - F^*\right],$$

where $F(w) = f(w) + r(w)$.

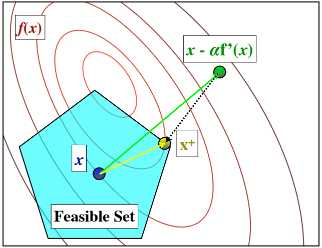
Projected-Gradient is Special case of Proximal-Gradient

- Projected-gradient methods are a special case:

$$r(w) = \begin{cases} 0 & \text{if } w \in \mathcal{C} \\ \infty & \text{if } w \notin \mathcal{C} \end{cases}, \quad (\text{indicator function for convex set } \mathcal{C})$$

gives

$$w^{k+1} \in \underbrace{\operatorname*{argmin}_{v \in \mathbb{R}^d} \frac{1}{2}\|v - w^{k+\frac{1}{2}}\|^2 + \alpha_k r(v)}_{\text{proximal operator}} \equiv \operatorname*{argmin}_{v \in \mathcal{C}} \frac{1}{2}\|v - w^{k+\frac{1}{2}}\|^2 \equiv \underbrace{\operatorname*{argmin}_{v \in \mathcal{C}} \|v - w^{k+\frac{1}{2}}\|}_{\text{projection}}$$



Properties of Proximal-Gradient

- Two convenient properties of proximal-gradient:
  - Proximal operators are non-expansive,

$$\|\text{prox}_r(w) - \text{prox}_r(v)\| \leq \|w - v\|,$$

    it only moves points closer together (easy to see for special case of projection).

    (including w^k and w^*)
  - For convex f, only fixed points are global optima,

$$w^* = \text{prox}_r(w^* - \alpha \nabla f(w^*)),$$

    for any $\alpha > 0$.

    (can test $\|w^k - \text{prox}_r(w^k - \nabla f(w^k))\|$ for convergence)
- Proximal gradient has two line-searches (generalizes projected variants):
  - Fix α_k and search along direction to w^{k+1} (1 proximal operator, non-sparse iterates).
  - Vary α_k values (multiple proximal operators per iteration, gives sparse iterations).

Proximal-Gradient Linear Convergence Rate

- Simplest linear convergence proofs are based on the proximal-PL inequality,

$$\frac{1}{2}\mathcal{D}_r(w, L) \geq \mu(F(w) - F^*),$$

where compared to PL inequality we've replaced $\|\nabla f(w)\|^2$ with

$$\mathcal{D}_r(w, \alpha) = -2\alpha \min_v \left[\nabla f(w)^\top (v - w) + \frac{\alpha}{2}\|v - w\|^2 + r(v) - r(w)\right],$$

and recall that $F(w) = f(w) + r(w)$.

- This non-intuitive property holds for many important problems:
  - L1-regularized least squares.
  - Any time f is strong-convex (i.e., add an L2-regularizer as part of f).
  - Any $f = g(Aw)$ for strongly-convex g and r being indicator for polyhedral set.
- But it can be painful to show that functions satisfy this property.

Proximal-Gradient Convergence under Proximal-PL

- Linear convergence if ∇f is Lipschitz and F is proximal-PL:

$$\begin{aligned} F(w_{k+1}) &= f(w^{k+1}) + r(w^{k+1}) \\ &= f(w_{k+1}) + r(w_k) + r(w_{k+1}) - r(w_k) \\ &\leq f(w_k) + \langle \nabla f(w_k), w_{k+1} - w_k \rangle + \frac{L}{2}||w_{k+1} - w_k||^2 + r(w_k) + r(w_{k+1}) - r(w \\ &= F(w_k) + \langle \nabla f(w_k), w_{k+1} - w_k \rangle + \frac{L}{2}||w_{k+1} - w_k||^2 + r(w_{k+1}) - r(w_k) \\ &\leq F(w_k) - \frac{1}{2L}\mathcal{D}_r(w_k, L) \\ &\leq F(w_k) - \frac{\mu}{L}[F(w_k) - F^*], \end{aligned}$$

and then we can take our usual steps.

Proximal-Newton

- We can define accelerated proximal-gradient in a straightforward way.
- We can define proximal-Newton methods using

$$w^{k+\frac{1}{2}} = w^k - \alpha_k [H_k]^{-1} \nabla f(w^k) \qquad \text{(Newton step)}$$

$$w^{k+1} = \underset{v \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ \frac{1}{2} \|v - w^{k+\frac{1}{2}}\|_{H_k}^2 + \alpha_k r(v) \right\} \qquad \text{(proximal step)}$$

- This is expensive even for simple r like L1-regularization.
- But there are analogous tricks to projected-Newton methods:
  - Diagonal or Barzilai-Borwein Hessian approximation.
  - "Orthant-wise" methods are analogues of two-metric projection.
  - Inexact methods use approximate proximal operator.
    - Orthant-wise and inexact methods often combined with L-BFGS or Hessian-free.

Outline

Active-Set Identification

- For L1-regularization, proximal-gradient "identifies" active set in finite time:

  (under mild assumptions)

  - For all sufficiently large k, sparsity pattern of x^k matches sparsity pattern of x^*.

$$w^0 = \begin{pmatrix} w_1^0 \\ w_2^0 \\ w_3^0 \\ w_4^0 \\ w_5^0 \end{pmatrix} \xrightarrow[\text{after finite } k \text{ iterations}]{} w^k = \begin{pmatrix} w_1^k \\ 0 \\ 0 \\ w_4^k \\ 0 \end{pmatrix}, \quad \text{where} \quad w^* = \begin{pmatrix} w_1^* \\ 0 \\ 0 \\ w_4^* \\ 0 \end{pmatrix}$$

- Useful if we are only interested in finding the sparsity pattern.
- Convergence rate will be faster once this happens (optimizing over subspace).
  - You could also apply Newton-like methods on the non-zero variables.

Related Work and More-General Results

- Idea of finitely identifying non-zeroes dates back (at least) to Bertskeas [1976].
  - For projected-gradient applied to smooth functions with non-negative constraints.
- Has been shown for a variety of convex/non-convex problems.
  - Burke & Moré [1988], Wright [1993], Hare & Lewis [2004], Hare [2011].
- These prior works only show that identification happens asymptotically.
  - For some finite but unknown k.
- Recent works consider "active-set complexity" of an algorithm:
  - The number of iterations before it is guaranteed to have reached the active set.

Special Case: Optimizing with Non-Negative Constraints

- We will first consider optimization with non-negative constraints,

$$\underset{w \geq 0}{\text{argmin}}\, f(w),$$

using the projected-gradient method with a step-size of $1/L$,

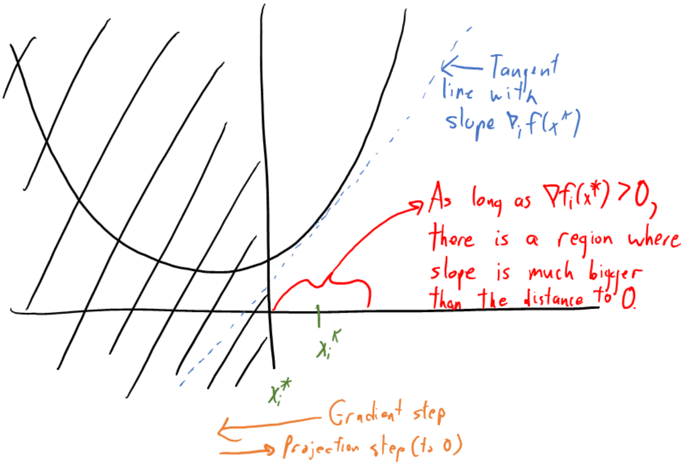
$$w^{k+1} = \left[w^k - \frac{1}{L} \nabla f(w^k) \right]^+.$$

- This also leads to sparsity, and we use $\mathcal{Z}$ as the indices i where $w_i^* = 0$.

- We'll assume:
  1. Gradient ∇f is L-Lipschitz continuous.
  2. Function f is μ-strongly convex.
  3. Non-degeneracy condition: for all $i \in \mathcal{Z}$ we have $\nabla f(w_i^*) \geq \delta$ for some $\delta > 0$.
     - "You can't have $\nabla_i f(w^*) = 0$ for variables i that are supposed to be zero."
     - This condition is standard: prevents reaching solution through interior.

Active-Set Identification for Non-Negative Constraints

- Let's show that we set $i \in \mathcal{Z}$ to zero when we're "close" to the solution.



Active-Set Identification for Non-Negative Constraints

- Let's show that we set $i \in \mathcal{Z}$ to zero when we're "close" to the solution.
  - Implies "for large 'k', if w_i^* is zero then the algorithm sets w_i^k to 0".

- Consider an iteration k where we have $\|w^k - w^*\| \leq \frac{\delta}{2L}$.

- In this region we have two useful properties for all $i \in \mathcal{Z}$:
  1. The value of the variable must be small: $w_i^k \leq \frac{\delta}{2L}$.
     - Since $w_i^* = 0$ and w_i^k is within $\delta/2L$ of w_i.
  2. The value of the gradient must be large: $\nabla_i f(w^k) \geq \delta/2$.
     - Since $\nabla_i f(w^*) \geq \delta$ and ∇f is Lipschitz.

- Plugging these into the projected-gradient update gives for $i \in \mathcal{Z}$ that

$$w_i^{k+1} = \left[w_i^k - \frac{1}{L}\nabla_i f(w^k)\right]^+ \leq \left[\frac{\delta}{2L} - \frac{\delta}{2L}\right]^+ = 0.$$

Active-Set Complexity for Non-Negative Constraints

- If ∇f is Lipschitz and f is strongly-convex then iterates converge linearly,

$$\|w^k - w^*\| \leq (1 - \kappa^{-1})^k \|w^0 - w^*\|,$$

  where the condition number κ is L/μ.

- Thus, for all sufficiently large k we have $\|w^k - w^*\| \leq \frac{\delta}{2L}$.
  - For these k the algorithm will have the correct active set.

- Using $(1 - \kappa^{-1})^k \leq \exp(-k/\kappa)$ and solving for k gives

$$\kappa \log(2L\|w^0 - w^*\|/\delta),$$

  so we find the sparsity pattern after this many iterations ("active-set complexity").

Active-Set Complexity for Non-Smooth Regularizers

- Can be generalized to lower/upper bounds and non-smooth but separable,

$$\mathop{\text{argmin}}_{l \leq w \leq u} f(w) + \sum_{i=1}^{n} g_i(w_i).$$

- Key differences:
  - The set $\mathcal{Z}$ will be variables occuring at bounds or non-smooth points.
    - For L1-regularization this is again the variables with $w_i^* = 0$.
  - The quantity δ will be the "minimum distance to the sub-differential boundary",

$$\delta = \min_{i \in \mathcal{Z}} \{\min\{-\nabla_i f(w^*) - \min\{\partial g_i(w_i^*)\}, \max\{\partial g_i(w_i^*)\} + \nabla_i f(x^*)\}\}.$$

    - For L1-regularization this is $\delta = \lambda - \max_{i \in \mathcal{Z}}\{|\nabla f_i(w^*)|\}$.
  - The non-degeneracy condition is that $\delta > 0$.
    - For L1-regularization we require $|\nabla_i f(w^*)| \neq \lambda$ for $i \in \mathcal{Z}$.
  - Proof needs to bound w_i^k from above and below based on $\partial g_i(w_i^*)$.
    - For other problems/algorithms, see "Wiggle Room Lemma".

Superlinear Convergence

- In a typical setting, we might hope that $|\mathcal{Z}^c| << d$.
  - Here we have the potential for faster algorithms by doing Newton steps on $\mathcal{Z}$.

- Some possibilities:
  - At some point, switch from proximal-gradient to Newton on the manifold.
    - Unfortunately, hard to decide when to switch.

  - Each iteration checks progress of proximal-gradient and Newton [Wright, 2012].
  - Two-metric projection [Gafni & Bertsekas, 1984].
    - May require expensive Newton steps before we're on the manifold.

  - There remains some theoretical and experimental work to do here.

Summary

- Simple regularizers are those that allow efficient proximal operator.
- Proximal-gradient: linear rates for sum of smooth and simple non-smooth.
- Manifold identification: identify the sparsity pattern in finite iterations.
- Active-set complexity is the number of iterations needed to find manifold.

- Next time: going beyond L1-regularization to "structured sparsity".

Indicator Function for Convex Sets

- The indicator function for a convex set is
$$r(w) = \begin{cases} 0 & \text{if } w \in \mathcal{C} \\ \infty & \text{if } w \notin \mathcal{C} \end{cases}.$$
- This is a function with "extended-real-valued" output: $r : \mathbb{R}^d \rightarrow \{\mathbb{R}, \infty\}$.
- The convention for convexity of such functions:
  - The "domain" is defined as the w values where $r(w) \neq \infty$ (in this case $\mathcal{C}$).
  - We need this domain to be convex.
  - And the function should to be convex on this domain.

Implicit subgradient viewpoint of proximal-gradient

- The proximal-gradient iteration is

$$w^{k+1} \in \operatorname*{argmin}_{v \in \mathbb{R}^d} \frac{1}{2}\|v - (w^k - \alpha_k \nabla f(w^k))\|^2 + \alpha_k r(v).$$

- By non-smooth optimality conditions that 0 is in subdifferential, we have that

$$0 \in (w^{k+1} - (w^k - \alpha_k \nabla f(w^k)) + \alpha_k \partial r(w^{k+1}),$$

which we can re-write as

$$w^{k+1} = w^k - \alpha_k (\nabla f(w^k) + \partial r(w^{k+1})).$$

- So proximal-gradient is like doing a subgradient step, with
  1. Gradient of the smooth term at w^k.
  2. A particular subgradient of the non-smooth term at w^{k+1}.
     - "Implicit" subgradient.

Proximal-Gradient for L0-Regularization

- There are some results on proximal-gradient for non-convex r.
- Most common case is L0-regularization,

$$f(w) + \lambda \|w\|_0,$$

  where $\|w\|_0$ is the number of non-zeroes.
  - Includes AIC and BIC from 340.
- The proximal operator for $\alpha_k \lambda \|w\|_0$ is simple:
  - Set $w_j = 0$ whenver $|w_j| \leq \alpha_k \lambda$ ("hard" thresholding).
- Analysis is complicated a bit because discontinuity of prox as function of α_k.
  - If step size is too small then you may not be able to move.

Faster Rate for Proximal-Gradient

- It's possible to show a slightly faster rate for proximal-gradient using $\alpha_t = 2/(\mu + L)$.
- See http://www.cs.ubc.ca/~schmidtm/Documents/2014_Notes_ProximalGradient.pdf

Equivalent Conditions to Proximal-PL

- When ∇f is L-Lipschitz continuous, the following 3 conditions are equivalent:
  1. Proximal-PL for some $\mu > 0$:
     $$\frac{1}{2}\mathcal{D}_r(w, L) \geq \mu(F(w) - F^*),$$
  2. Error bounds for some $c > 0$:
     $$\|w - w_p\| \leq c\left\|w - \text{prox}_{\frac{1}{L}r}\left(w - \frac{1}{L}\nabla f(w)\right)\right\|,$$
     where w_p is the projection of x onto the set of solution.
  3. Kurdyka-Lojasiewicz for some $\mu > 0$:
     $$\min_{s \in \partial F(w)} \frac{1}{2}\|s\|^2 \geq \mu(F(w) - F^*),$$
     where $\partial F(w)$ is the "local" sub-differential.

(Same as usual sub-differential for convex)