PGMO Lecture: Vision, Learning and Optimization

3. Proximal gradient methods

Thomas Pock

Institute of Computer Graphics and Vision, TU Graz

February 12, 2020

Overview

Implicit descent

- Recall the (explicit) gradient method for minimizing a convex function $f(x)$ with L-Lipschitz continuous gradient $\nabla f(x)$:

$$x^{k+1} = x^k - \tau \nabla f(x^k), \quad \tau \in (0, 2/L)$$

- A more desirable scheme would be to make the above iteration more "implicit":

$$x^{k+1} = x^k - \tau \nabla f(x^{k+1}), \quad \tau > 0$$

- We see that x^{k+1} is a critical point of the function

$$x \mapsto f(x) + \frac{1}{2\tau} \left\| x - x^k \right\|^2 .$$

- Hence x^{k+1} is exactly the proximal map wrt the function f of the point x^k and step size parameter τ.

Gradient descent on the Moreau envelope

- Let us denote by $f_\tau(x)$ the Moreau envelope of f.
- A gradient descent step with step size τ on the Moreau envelope yields

$$\begin{aligned} x^{k+1} &= x^k - \tau \nabla f_\tau(x^k) \\ &= x^k - \tau \left(\frac{1}{\tau}(I - \mathrm{prox}_{\tau f})(x^k) \right) \\ &= \mathrm{prox}_{\tau f}(x^k) \end{aligned}$$

- Hence an explicit gradient descent step on the Moreau envelope is equivalent to an implicit descent step, i.e. a proximal step.
- This algorithm is called the proximal point algorithm.
- Proposed by [Minty '62], [Martinet '70], [Rockafellar '76].
- It plays a major role in convex optimization, as many existing algorithms are just special cases of it.

Application to composite problems

- Let us consider composite problems of the form

$$\min_x f(x) := \frac{1}{2} \|Ax - b\|^2 + g(x),$$

  where we assume that we know how to compute $\mathrm{prox}_{\tau g}(x)$.
- An important observation is that in this case, the Moreau envelope and its gradient can be computed, provided we choose the right metric.
- Let

$$M = \frac{1}{\tau} I - A^* A,$$

  which is positive if $\tau \|A\|^2 < 1$.

Explicit solution

- In the M metric, the Moreau envelope is given by

$$f_M(x) := \min_y \frac{1}{2}\|y-x\|_M^2 + \frac{1}{2}\|Ay-b\|^2 + g(y)$$

- The point $\hat{x} = \mathrm{prox}_f^M(x)$ that solves the problem is given by

$$\hat{x} = \mathrm{prox}_{\tau g}(x - \tau A^*(Ax-b))$$

- The gradient is of the function f_M in the M-metric is given by

$$\nabla f_M(x) = x - \hat{x},$$

which is 1-Lipschitz and hence we can use a step size of 1 in the gradient descent.

- Finally, the iterations of the proximal point algorithm read

$$x^{k+1} = \mathrm{prox}_{\tau g}(x^k - \tau A^*(Ax^k - b))$$

Proximal point algorithm for more general problems

Definition

Let X be a Hilbert space. A multivalued function $T : X \rightrightarrows X$ is said to be a monotone operator if

$$\langle T(z) - T(z'), z - z' \rangle \geq 0 \,, \quad \forall z, z' \in X$$

Furthermore, it is maximal monotone, if its graph is not contained in any other monotone operator.

A fundamental problem is to find a zero of a maximal monotone operator T in a real Hilbert space X

$$\text{find } x \in X : \; 0 \in T(x)$$

The problem includes convex minimization problems but also convex-concave saddle point problems.

The proximal point algorithm

The proximal point algorithm can be used to solve the monotone inclusion problem by iterating

$$x^{k+1} = (I + \tau_k T)^{-1}(x^k),$$

where $\tau_k > 0$.

- Note that the iterates of the proximal point algorithm are not defined via proximal maps, which are computed by solving a minimization problem but rather directly based on the resolvent operator

$$J_{\tau_k T} = (I + \tau_k T)^{-1}$$

- We will later see that we can solve a class of saddle-point problems using the proximal point algorithm.

More general problems?

- We have seen that the application of the proximal point algorithm (in a well chosen metric) to composite problems of the form

$$\min_x f(x) := \frac{1}{2} \|Ax - b\|^2 + g(x),$$

with an easy to compute proximal map for the convex function g yields iterations of the form

$$x^{k+1} = \mathrm{prox}_{\tau g} \left(x^k - \tau A^*(Ax^k - b) \right)$$

- Looking closer to the iteration, one can see that it is a combination of an explicit gradient step w.r.t. the least squares term followed by an implicit step (proximal map) w.r.t. the function g.

Overview

Proximal gradient method

- It can be generalized to composite functions of the following form

$$\min_x F(x) := f(x) + g(x),$$

where f is a convex function with L-Lipschitz continuous gradient and g is a convex function with simple proximal map.

Algorithm 1 Proximal gradient method

Choose $x_0 \in \mathcal{X}$, $\tau > 0$.

for all $k \geq 0$ **do**

$$x^{k+1} = \text{prox}_{\tau g}(x^k - \tau \nabla f(x^k)).$$

end for

Discussion

- Let us observe that a fixed point of the proximal gradient method will satisfy

$$\begin{aligned} x =& \quad \mathrm{prox}_{\tau g}(x - \tau \nabla f(x)) \\ x =& \quad (I + \tau \partial g)^{-1}(x - \tau \nabla f(x)) \\ x + \tau \partial g(x) \ni& \quad x - \tau \nabla f(x) \\ \partial g(x) \ni& \quad -\nabla f(x) \\ 0 \in& \quad \partial g(x) + \nabla f(x) \end{aligned}$$

- The last line is exactly the optimality condition of our composite problem.
- In case $g = \delta_C$, the indicator function of a convex set C and hence $\mathrm{prox}_{\tau g} = \mathrm{proj}_C$, the algorithm reduces to the projected gradient method.

3 term inequality

We will now state an important inequality that will be used in deriving convergence rates for almost all methods.

Proposition

Let $F = f + g$ be a convex function with f μ_f-strongly convex with L-Lipschitz continuous gradient and g μ_g-strongly convex with simple to compute proximal map. Let $\bar{x}$ and $\hat{x}$ be the old and new iterations of the proximal gradient method, then one has for all $x \in \mathcal{X}$

$$F(x) + (1 - \tau\mu_f)\frac{\|x - \bar{x}\|^2}{2\tau}$$
$$\geq \frac{1 - \tau L}{\tau}\frac{\|\hat{x} - \bar{x}\|^2}{2} + F(\hat{x}) + (1 + \tau\mu_g)\frac{\|x - \hat{x}\|^2}{2\tau}$$

Observe that by choosing $x = \bar{x} = x^k$, $\hat{x} = x^{k+1}$ and $\tau \leq \frac{1}{L}$,

$$F(x^{k+1}) \leq F(x^k) - (1 + \tau\mu_g)\frac{\left\|x^{k+1} - x^k\right\|^2}{2\tau},$$

which shows that the proximal gradient method generates a non-incresing sequence of function values $F(x^k)$.

Convergence rate

We can show the following convergence rate for the proximal gradient method:

Theorem

Let $\{x^k\}$ be a sequence generated by the proximal gradient method with $\tau \leq \frac{1}{L}$. It has the following convergence rate:

$$F(x^k) - F(x^*) \leq \frac{\left\|x^* - x^0\right\|^2}{2k\tau}.$$

Moreover, if in addition f or g is strongly convex with parameters μ_f, μ_g with $\mu_f + \mu_g > 0$, we have

$$F(x^k) - F(x^*) + \frac{1 + \tau\mu_g}{2\tau}\left\|x^k - x^*\right\|^2 \leq \omega^k \frac{1 + \tau\mu_g}{2\tau}\left\|x^0 - x^*\right\|^2,$$

with $\omega = (1 - \tau\mu_f)/(1 + \tau\mu_g)$.

Remark convergence is guaranteed also for larger step sizes $\tau < 2/L$.

Backtracking linesearch

- In case the Lipschitz constant L of the smooth function f is unknown, the same convergence rates hold when implementing a backtracking linesearch.
- Starting with some initial guess $L_0 > 0$ one creates a non-decreasing sequence $\{L_k\}$ by letting $L_{k+1} = \eta^{i_k} L_k$ with $\eta > 1$ and i_k is the smallest integer such that

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L_{k+1}}{2} \left\| x^{k+1} - x^k \right\|^2,$$

  holds.
- In practice, one can also try to reduce the Lipschitz constant after each successful step, but the theoretical guarantees are lost.

Example: minimizing the dual ROF model

Recall that the dual ROF model is given by

$$\min_{\mathbf{p}} \underbrace{\frac{1}{2}\left\|\mathrm{D}^*\mathbf{p} - d\right\|^2}_{f(\mathbf{p})} + \underbrace{\delta_{\{\|\cdot\|_{q,\infty}\leq\lambda\}}(\mathbf{p}), \quad q \in \{2,\infty\}}_{g(\mathbf{p})}.$$

The function $f(\mathbf{p})$ has a Lipschitz continuous gradient

$$\nabla f(\mathbf{p}) = \mathrm{D}(\mathrm{D}^*\mathbf{p} - d),$$

with parameter $L = 8$, the function $g(\mathbf{p})$ has a easy to compute proximal map (projection)

$$\hat{\mathbf{p}} = \mathrm{proj}_{\{\|\cdot\|_{2,\infty}\leq\lambda\}}(\tilde{\mathbf{p}}) \Leftrightarrow \hat{\mathbf{p}}_{i,j} = \frac{\tilde{\mathbf{p}}_{i,j}}{\max\left\{1, |\tilde{\mathbf{p}}_{i,j}|_2/\lambda\right\}}.$$

The main iteration of the proximal gradient method is given by

$$\mathbf{p}^{k+1} = \mathrm{proj}_{\{\|\cdot\|_{q,\infty}\leq\lambda\}}(\mathbf{p}^k - \tau\mathrm{D}(\mathrm{D}^*\mathbf{p}^k - d)),$$

with $\tau \in (0, 2/L)$. The primal solution can be recovered via $u^k = d - \mathrm{D}^*\mathbf{p}^k$

Practical

`dual-rof.ipynb`

How good is this?

- What is an optimal first-order method?
- There are unbeatable lower bounds for first order methods of the form [Nemirovsky, Yudin 1983], [Nesterov 1994]

$$\boxed{x^{k+1} \in x^0 + \mathrm{span}\{\nabla f(x^0), \nabla f(x^1), ..., \nabla f(x^k)\}}$$

- Assume that the gradient of f is Lipschitz continuous with parameter L
- If f is μ-strongly convex with $\mu > 0$ then

$$f(x^k) - f(x^*) \geq \frac{\mu}{2}\left(\frac{\sqrt{L/\mu}-1}{\sqrt{L/\mu}+1}\right)^{2k}\left\|x^0 - x^*\right\|^2,$$

- If $\mu = 0$

$$f(x^k) - f(x^*) \geq \frac{3L\left\|x^0 - x^*\right\|^2}{32(k+1)^2}$$

Discussion

- From the lower bounds we see that the proximal gradient method is significantly slower compared (assuming $\tau = 1/L$) to the lower bounds:

- If $\mu > 0$ the proximal gradient method ($\mu_g = 0$) gives a linear rate of $(1 - \mu/L)^k/(2L)$ as compared to $(\mu/2)((\sqrt{L/\mu} - 1)/(\sqrt{L/\mu} + 1))^{2k}$ of the lower bound.
- Example: $L = 1, \mu = 0.001, k = 1000$. The proximal gradient method gives a reduction of 0.1838, however, the lower bound gives $5.5752e - 59$

- If $\mu = 0$, the proximal gradient method gives a sublinear rate of $L/(2k)$ versus a sublinear rate of $3L/(32(k+1)^2)$ of the lower bound.
- Example: $L = 1$, $k = 1000$. The proximal gradient reduces the right hand side by $1e - 03$, the lower bound however is $1e - 06$.

Overview

The heavy ball method

In [Polyak '64], the heavy ball algorithm is introduced for minimizing a μ-strongly convex function f with L-Lipschitz continuous gradient ∇f:

Algorithm 2 Heavy ball method

Choose $x_0 \in \mathcal{X}$, $\tau^k, \beta^k > 0$.
for all $k \geq 0$ **do**

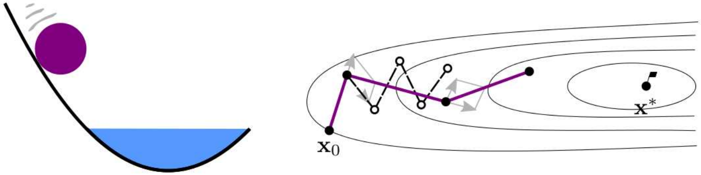
$$x^{k+1} = x^k - \tau^k \nabla f(x^k) + \beta^k (x^k - x^{k-1})$$

end for

- The additional term $\beta^k (x^k - x^{k-1})$ is called the momentum or inertial force.
- The heavy ball algorithm can be seen as a more general variant of the conjugate gradient (CG) method, with more freedom to choose τ^k and β^k.

Physical interpretation

- Can be seen as an explicit finite differences discretization of the heavy-ball with friction dynamical system
$$\ddot{x}(t) + \gamma \dot{x}(t) + \nabla f(x(t)) = 0\,.$$



Source: Stich et al.

- Consider the following finite differences approximation:
$$\ddot{x}(t) \approx \frac{x^{k+1} - 2x^k + x^{k-1}}{h^2},\; \dot{x}(t) \approx \frac{x^{k+1} - x^k}{h},\; \nabla f(x(t)) \approx \nabla f(x^k)$$

- Re-arranging the terms and properly defining the constants τ^k, β^k yields the heavy ball method.
- For quadratic problems and locally optimizing for τ^k, β^k, it is equivalent to the CG method.

Acceleration?

$\beta = 0.0$

Acceleration?

$\beta = 0.5$

Acceleration?

$\beta = 0.9$

Acceleration?

$$\beta = 1.1$$

Rate of convergence

Optimal t, β from the global properties of f [Polyak '64]

Theorem

If f is a twice continuously differentiable, μ-strongly convex function with L-Lipschitz continuous gradient, and τ, β are chosen according to

$$\tau = \frac{4}{(\sqrt{L} + \sqrt{\mu})^2}, \quad \beta = \left(\frac{\sqrt{L/\mu} - 1}{\sqrt{L/\mu} + 1} \right)^2$$

Then, for every $\varepsilon > 0$ there is $c > 0$ such that for all k

$$\left\| (x^{k+1} - x^*, x^k - x^*)^T \right\| \leq c \, (\sqrt{\beta} + \varepsilon)^k \left\| (x^k - x^*, x^{k-1} - x^*)^T \right\|$$

- The heavy ball method is optimal for smooth and strongly convex ($\mu > 0$) functions!
- It does not work so well for degenerate smooth ($\mu = 0$) functions.

Nesterov's algorithm

- The heavy ball method has the disadvantage that the optimal (linear) convergence is true only for strongly convex problems
- On smooth convex problem, it is not clear how to set the parameters and hence the method can be slow
- The heavy ball algorithm also requires the function to be twice continuously differentiable, which limits its applicability
- In 1983, Nesterov proposed a new algorithm, closely related to the heavy ball algorithm which requires the function to be only once continuously differentiable and yields the optimal convergence rate $O(1/k^2)$ for smooth (degenerate) problems.
- Major impact (only after 2005) to all optimization driven computational sciences such as vision, image/signal processing, machine learning, data mining, ...

Nesterov's "millenium" paper

Докл. Акад. Наук СССР
Том 269 (1983), № 3

Soviet Math. Dokl.
Vol. 27 (1983), No. 2

**A METHOD OF SOLVING
A CONVEX PROGRAMMING PROBLEM
WITH CONVERGENCE RATE $O(1/k^2)$**

UDC 51

YU. E. NESTEROV

1. In this note we propose a method of solving a convex programming problem in a Hilbert space E. Unlike the majority of convex programming methods proposed earlier, this method constructs a minimizing sequence of points $\{x_k\}_0^\infty$ that is not relaxational. This property allows us to reduce the amount of computation at each step to a minimum. At the same time, it is possible to obtain an estimate of convergence rate that cannot be improved for the class of problems under consideration (see [1]).

2. Consider first the problem of unconstrained minimization of a convex function $f(x)$. We will assume that $f(x)$ belongs to the class $C^{1,1}(E)$, i.e. that there exists a constant $L > 0$ such that for all $x, y \in E$

$$\|f'(x) - f'(y)\| \leq L\|x - y\|. \tag{1}$$

Nesterov's algorithm

In his paper, Nesterov introduced the following algorithm:

Algorithm 3 Nesterov's algorithm

Choose $x^0 = x^{-1} \in \mathcal{X}$, $\tau \leq 1/L$, $t_0 = 0$.

for all $k \geq 0$ **do**

$$\begin{aligned} t_{k+1} &= \tfrac{1+\sqrt{1+4t_k^2}}{2}, \ \beta_k = \tfrac{t_k - 1}{t_{k+1}} \\ y^k &= x^k + \beta_k(x^k - x^{k-1}) \\ x^{k+1} &= y^k - \tau \nabla f(y^k) \end{aligned}$$

end for

- Nesterov's algorithm uses a dynamic choice of the overrelaxation parameter with $\beta^k = \frac{t_k - 1}{t_{k+1}} \to 1$
- The gradient is evaluated at the extrapolated point $y^k = x^k + \beta^k(x^k - x^{k-1})$, while in the heavy ball method, the gradient is taken at the original point x^k

Convergence rate

Theorem

Consider a convex function $f : \mathcal{X} \to \mathbb{R}$ with L-Lipschitz continuous gradient. Let $\{x^k\}$ be a sequence generated by Nesterov's algorithm. Then, if x^* is a minimizer of $f(x)$, we have

$$f(x^k) - f(x^*) \leq \frac{2L \left\| x^0 - x^* \right\|^2}{(k+1)^2}$$

- This shows that (up to constants) Nesterov's algorithm is equivalent to the lower bound and hence an optimal method.
- In case the function is also strongly convex with parameter $\mu > 0$, the overrelaxation parameter can be set to the constant value

$$\beta_k = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}},$$

  such that Nesterov's algorithm also yields an optimal linear convergence rate.
- We will state the convergence rate in a more general proximal-gradient setting.

Overview

Proximal point algorithm

Proximal gradient method

Accelerated gradient methods

Accelerated proximal gradient methods

Nonlinear proximal methods

The FISTA algorithm

- Next, we will show how to generalize Nesterov's algorithm to composite problems of the form

$$\min_x F(x) := f(x) + g(x),$$

with ∇f L-Lipschitz continuous and g has a simple to compute proximal map.
- This leads to the famous Fast Iterative Shrinkage Thresholding Algorithm (FISTA) [Beck and Teboulle '09].
- Moreover, we will cover the case where $F(x)$ is $\mu = \mu_f + \mu_g$ strongly convex.

The FIST(A)lgorithm

Algorithm 4 FISTA

Given $0 < \tau \leq 1/L$, let $q = \tau\mu/(1+\tau\mu_g) < 1$. Choose $x^0 = x^{-1} \in \mathcal{X}$, and $t_0 \in \mathbb{R}$, $0 \leq t_0 \leq 1/\sqrt{q}$.

for all $k \geq 0$ **do**

$$\begin{aligned} y^k &= x^k + \beta_k(x^k - x^{k-1}) \\ x^{k+1} &= \mathrm{prox}_{\tau g}(y^k - \tau\nabla f(y^k)) \end{aligned}$$

where, for $\mu = 0$,

$$\begin{aligned} t_{k+1} &= \tfrac{1+\sqrt{1+4t_k^2}}{2}, \\ \beta_k &= \tfrac{t_k - 1}{t_{k+1}}, \end{aligned}$$

and if $\mu = \mu_f + \mu_g > 0$,

$$\begin{aligned} t_{k+1} &= \tfrac{1-qt_k^2+\sqrt{(1-qt_k^2)^2+4t_k^2}}{2}, \\ \beta_k &= \tfrac{t_k - 1}{t_{k+1}} \tfrac{1+\tau\mu_g - t_{k+1}\tau\mu}{1-\tau\mu_f}. \end{aligned}$$

end for

Convergence rate

The following result unifies Nesterov's algorithm [Nesterov '04] and the FISTA algorithm [Beck, Teboulle '09]

Theorem

Assume $t_0 = 0$ *and let* x^k *be generated by the FISTA algorithm, in either case* $\mu = 0$ *or* $\mu > 0$*. Then we have the decay rate*

$$F(x^k) - F(x^*) \leq \min\left\{(1+\sqrt{q})(1-\sqrt{q})^k, \frac{4}{(k+1)^2}\right\} \frac{1+\tau\mu_g}{2\tau} \left\|x^0 - x^*\right\|^2,$$

where $q = \tau\mu/(1+\tau\mu_g) < 1$*.*

Discussion

- The FISTA algorithm is an optimal algorithm!
- The fixed step size $\tau = 1/L$ can be replaced by a backtracking linesearch procedure to find the correct L.
- However, if $\mu > 0$ is unknown, the algorithm can be hard to tune.
- The FISTA algorithm is not a monotone algorithm, i.e. the function values can even increase. The convergence rate only ensures a certain quality after k steps.
- There also exists a monotone variant of the algorithm (MFISTA, [Beck, Teboulle '09]), preserving the same accelerated convergence rate.
- The iterates x^k can not be shown to converge to x^*, but a slight change of β^k ensures in addition the convergence of the iterates [Chambolle, Dossal '15].

Practical

`dual-rof.ipynb`

Adaptive FISTA

- Inspired by the conjugate gradient method, we proposed in [Ochs, P. '17] an adaptive method to compute the overrelaxation parameter β^k.
- Consider the overrelaxation step of the FISTA method $y^k(\beta) = x^k + \beta(x^k - x^{k-1})$ and solve in each step of the proximal gradient step

$$x^{k+1} = \arg\min_x \min_\beta g(x) + \left\langle \nabla f(y^k(\beta)), x - y^k(\beta) \right\rangle + \frac{L}{2} \|x - y(\beta)\|^2$$

- It turns out that in case $f(x)$ is a quadratic function with Hessian $H \preceq L \cdot I$, the above problem has an explicit solution:

$$\beta^* = \frac{\left\langle x^k - x^{k-1}, x - x^k \right\rangle_M}{\|x^k - x^{k-1}\|_M^2}, \quad M = \frac{1}{\tau} I - H,$$

and the proximal gradient step becomes

$$x^{k+1} = \arg\min_x g(x) + \frac{1}{2} \left\| x - x^k + Q^{-1} \nabla f(x^k) \right\|_Q^2,$$

where

$$u = \frac{M(x^k - x^{k-1})}{\|x^k - x^{k-1}\|_M}, \quad Q = \frac{1}{\tau} I - uu^T, \quad Q^{-1} = \tau I + \frac{\tau^2 uu^T}{1 - \tau u^T u}.$$

Discussion

- The adaptive FISTA method for quadratic f corresponds to a identify minus rank-1 proximal quasi-Newton method [Nocedal, Wright '06], for which most proximal maps are still tractable [Becker, Fadili '12].
- The adaptive FISTA method does not preserve the optimal convergence rate but works very well in practice.
- The optimal convergence rate can be preserved by integrating it for example in the MFISTA framework.

Overview

Proximal point algorithm

Proximal gradient method

Accelerated gradient methods

Accelerated proximal gradient methods

Nonlinear proximal methods

Mirror descent

- A natural generalization of the gradient methods is to replace the quadratic Euclidean distance function by other distances.
- Good reasons could be:
  - Act as a barier to incorporate constraints
  - Proximal map easier to solve in a different distance
  - Smaller Lipschitz constant
- The basic idea is to replace the gradient descent equation

$$\tau \nabla f(x^k) = x^k - x^{k+1} \quad \text{by} \quad \tau \nabla f(x^k) = \nabla \psi(x^k) - \nabla \psi(x^{k+1}),$$

  where ψ is a differentiable and strongly convex function.
- Introducing the Bregman ψ-distance

$$D_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla \psi(y), x - y \rangle ,$$

  we see that the generalized descent is a minimizer of

$$\min_x \frac{1}{\tau} D_\psi(x, x^k) + f(x^k) + \langle \nabla f(x^k), x - x^k \rangle .$$

- Observe that $\psi = \frac{1}{2} \|\cdot\|_2^2$ recovers the usual Euclidean setting.

Implicit mirror descent

- We can also consider an implicit variant, known as non-linear proximal point algorithm,

$$\min_x \frac{1}{\tau} D_\psi(x, x^k) + f(x)$$

  whose minimizer satisfies

$$\nabla\psi(x^{k+1}) - \nabla\psi(x^k) + \tau\partial f(x^{k+1}) \ni 0$$

- Thanks to the strong convexity of the Bregman distance, we can also derive the basic 3 term inequality

$$\frac{1}{\tau} D_\psi(x, x^k) + f(x) \geq \frac{1}{\tau} D_\psi(x^{k+1}, x^k) + f(x^{k+1}) + \frac{1}{\tau} D_\psi(x, x^{k+1}),$$

  from which convergence rates can be easily deduced.
- Can also be generalized to the forward-backward splitting setting.

Example

- Consider the simplex constrained Lasso problem

$$\min_{x \in S^{n-1}} f(x) := \frac{1}{2} \|Ax - b\|_2^2 ,$$

  whith the $n-1$ dimensional unit simplex defined as

$$S^{n-1} = \left\{ x : x_i \geq 0,\ i = 1...n,\ \sum_{i=1}^{n} x_i = 1 \right\}$$

- It is known that the entropy

$$\psi(x) := \sum_{i=1}^{n} x_i \ln x_i , \quad \nabla\psi(x) = (1 + \ln x_i)_{i=1}^{n})$$

  is 1-strongly convex w.r.t. the ℓ_1 norm $\|\cdot\|_1$
- Observe that the entropy acts as a barrier function for $x_i \geq 0$

Iterations

- We can drop the inequality constraints and hence the iteration takes the form

$$x^{k+1} = \arg\min_{\sum_i x_i = 1} f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{\tau} D_\psi(x, x^k)$$

- It turns out that we can explicitly solve the iteration as

$$x_i^{k+1} = \frac{e^{-\tau(\nabla f(x^k))_i}}{\sum_{j=1}^n x_j^k \, e^{-\tau(\nabla f(x^k))_j}} x_i^k, \quad \nabla f(x) = A^*(Ax - b)$$

- the step size is restricted as $0 < \tau \leq 1/L$, with L such that

$$\|\nabla f(x) - \nabla f(y)\|_\infty \leq L \|x - y\|_1$$

- The operator norm in the ℓ_1 norm is given by

$$\|A^* A\|_{1,\infty} = \sup_{\|x\|_1 \leq 1} \|A^* A x\|_\infty = \max_{i,j} \{ |A^* A|_{i,j} \},$$

  which is usually smaller than the 2-norm

Proximal gradient as nonlinear proximal gradient

- The proximal gradient method can also be written as an implicit mirror descent algorithm.
- Let us consider the following optimization problem:

$$\min_x f(x) + g(x),$$

  where f is smooth with L-Lipschitz continuous gradient and g has an easy prox operator.
- Let us choose the following kernel in the Bregman distance

$$\psi = \frac{L}{2} \|\cdot\|^2 - f(\cdot),$$

  which is convex as long as f has a L-Lipschitz continuous gradient.
- The corresponding Bregman distance is given by

$$D_\psi(x,y) = \frac{L}{2} \|x\|^2 - \frac{L}{2} \|y\|^2 + f(y) - f(x) - \langle Ly - \nabla f(y), x - y \rangle$$

- The implicit descent now reads

$$x^{k+1} = \arg\min_x f(x) + g(x) + D_\psi(x, x^k)$$

Solving the iterations

- The first-order-optimality condition for x^{k+1} is given by

$$
\begin{aligned}
0 &\in \nabla f(x^{k+1}) + \partial g(x^{k+1}) + Lx^{k+1} - \nabla f(x^{k+1}) - Lx^k + \nabla f(x^k) \\
0 &\in \partial f(x^k) + g(x^{k+1}) + L(x^{k+1} - x^k) \\
x^k - \frac{1}{L}\nabla f(x^k) &\in (I + \frac{1}{L}\partial g)(x^{k+1}) \\
x^{k+1} &= \mathrm{prox}_{\frac{1}{L}g}(x^k - \frac{1}{L}\nabla f(x^k))
\end{aligned}
$$

- This is exactly the proximal gradient method!