Lecture 11

January 4, 2022

Proximal methods

Assume that $f(x) = g(x) + h(x)$ where g is smooth and h is convex.

Example:

$$g(x) + \|x\|_1$$

(l^1 norm promotes sparsity). In particular LASSO problem is of this form.

Proximal methods

Example: For matrices instead of l^1 norm it is natural to use nuclear norm. For positive definite matrices nuclear norm is just sum of eigenvalues. In general one can compute nuclear norm using singular value decomposition. We use

$$g(X) + \|X\|_*$$

where X goes over matrices and $\|X\|_*$ is nuclear norm to promote low rank matrices as solutions.

Proximal operator

For differentiable f steepest descent is

$$x_{i+1} = x_i - \alpha_i \nabla f(x_i).$$

We can think that steepest descent minimizes quadratic approximation to f:

$$x_{i+1} = \operatorname{argmin}_x f(x_i) + \langle \nabla f(x_i), x \rangle + \frac{1}{2\alpha_i} \|x - x_i\|^2.$$

Proximal operator

When h is nonsmooth natural idea is to take quadratic approximation to g and leave h as is. That is

$$x_{i+1} = \operatorname{argmin}_x g(x_i) + \langle \nabla g(x_i), x \rangle + \frac{1}{2\alpha_i}\|x - x_i\|^2 + h(x)$$

$$= \operatorname{argmin}_x \frac{\alpha_i}{2}\|\nabla g(x_i)\|^2 + \langle \nabla g(x_i), x - x_i \rangle + \frac{1}{2\alpha_i}\|x - x_i\|^2 + h(x)$$

$$= \operatorname{argmin}_x \frac{1}{2}\|x - x_i\|^2 + \langle \alpha_i \nabla g(x_i), x - x_i \rangle + \frac{1}{2}\|\alpha_i \nabla g(x_i\|^2 + \alpha_i h(x)$$

$$= \operatorname{argmin}_x \frac{1}{2}\|x - x_i + \alpha_i \nabla g(x_i)\|^2 + \alpha_i h(x).$$

where we obtained second equality by changing constant term and third equality multiplying by α_i.

Proximal operator

The last expression is called proximal operator. More formally for arbitrary convex function $h : \mathbb{R}^n \mapsto \mathbb{R} \cup \infty$ we define

$$\text{prox}_h(x) = \text{argmin}_y(\frac{1}{2}\|y - x\|^2 + h(x)).$$

With such notation we have

$$x_{i+1} = \text{prox}_{\alpha_i h(x)}(x_i - \alpha_i \nabla g(x_i)).$$

Resulting algorithm is called proximal gradient algorithm.

Proximal operator

Why this is good?

- con above each step requires minimization of auxiliary function which in principle can be expensive.
- pro there are examples where proximal operator can be computed in closed form and is quite cheap to evaluate.
- pro proximal operator has nice theoretical properties, in particular satisfies remarkable equalities.
- pro some other algorithms can be viewed as proximal gradient algorithm with appropriate *h*.

Proximal operator

Example: $h(x) = \|x\|_1 = \sum_{j=1}^{n} |x_j|$. Then

$$\operatorname{prox}_{\lambda h}(x) = \operatorname{argmin}_y(\frac{1}{2}\|y - x\|^2 + \lambda\|y\|_1)$$

$$= \operatorname{argmin}_y(\sum_{j=1}^{n}(\frac{1}{2}(y_j - x_j)^2 + \lambda|y_j|)).$$

This is sum of functions of separate variables, so it is enough to minimize each term separately, that is compute

$$\operatorname{argmin}_z \frac{1}{2}(z - x_j)^2 + \lambda|z|.$$

Now, $\phi_{t,\lambda}(z) = \frac{1}{2}(z - t)^2 + \lambda|z|$ is differentiable when $z \neq 0$, so optimum is attained either at $z = 0$ or at point where derivative is 0.

Proximal operator

Differentiating we get

$$\phi'_{t,\lambda}(z) = z - t + \lambda$$

for $z > 0$ and

$$\phi'_{t,\lambda}(z) = z - t - \lambda$$

for $z < 0$.
The first gives condition

$$z - t + \lambda = 0$$

that is $z = t - \lambda$. Similarly the second expression gives $z = t + \lambda$. Since $\phi_{t,\lambda}$ is convex when one of conditions above hold, than this gives minimum. Otherwise, minimum is a 0. So

$$\operatorname{argmin}_z \phi_{t,\lambda} = \begin{cases} t - \lambda & \text{when } t > \lambda \\ 0 & \text{when } t \in [-\lambda, \lambda] \\ t + \lambda & \text{when } t < -\lambda \end{cases}$$

Proximal operator

For original problem we get

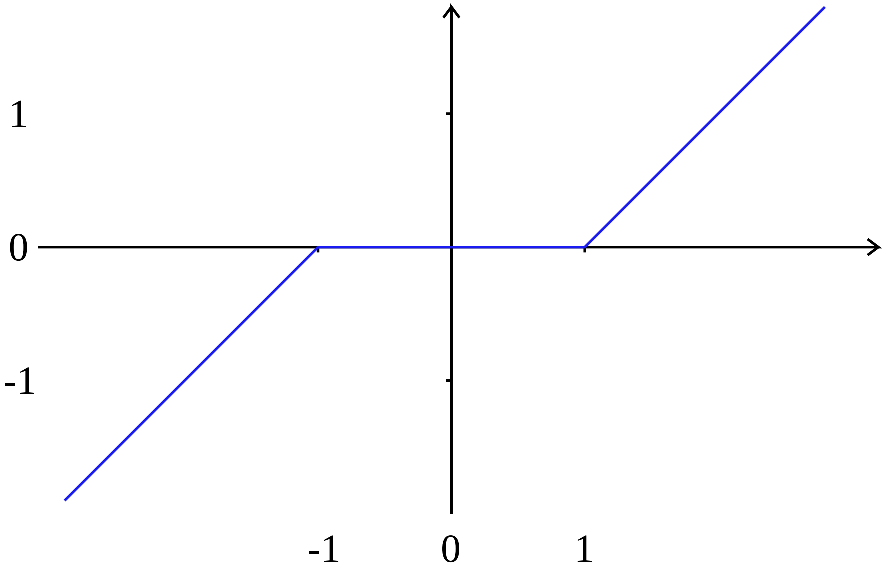
$$\operatorname{prox}_{\lambda h}(x) = S_\lambda(x)$$

where

$$S_\lambda(x)_j = \begin{cases} x_j - \lambda & \text{when } x_j > \lambda \\ 0 & \text{when } x_j \in [-\lambda, \lambda] \\ x_j + \lambda & \text{when } x_j < -\lambda \end{cases}$$

is called soft thresholding.

Graph of soft thresholding with $\lambda = 1$ in one variable:



ISTA

For LASSO problem, that is minimizing $f(x) + \lambda\|x\|_1$ where

$$f(x) = \frac{1}{2}\|Ax - y\|^2$$

proximal gradient algorithm now is:

$$x_{i+1} = S_\lambda(x_i - \alpha_i A^T(Ax - y)).$$

This is called ISTA (iterative soft thresholding).

Proximal operator

Example: When C is closed convex set and $h(x) = 0$ for $x \in C$ and $h(x) = \infty$ otherwise, we get

$$\operatorname{prox}_h(x) = \operatorname{argmin}_{y \in C} \frac{1}{2} \|y - x\|^2 = \operatorname{Proj}_C(x).$$

So corresponding proximal gradient algorithm is just projected gradient algorithm.

Proximal operator

Proximal operator is nonexpansive:

Lemma

$$\|\operatorname{prox}_h(x_2) - \operatorname{prox}_h(x_1)\|^2 \leq \langle \operatorname{prox}_h(x_2) - \operatorname{prox}_h(x_1), x_2 - x_1 \rangle,$$

$$\|\operatorname{prox}_h(x_2) - \operatorname{prox}_h(x_1)\| \leq \|x_2 - x_1\|.$$

Remark: In particular projection operator is nonexpansive.
Proof: Second inequality follows from the first using Schwartz inequality, so it is enough to prove the first.

Proximal operator

By definition

$$\text{prox}_h(x) = \text{argmin}_z \frac{1}{2}\|z-x\|^2 + h(z)$$

so when $z_0 = \text{prox}_h(x)$ than for any z we have

$$\frac{1}{2}\|z-x\|^2 + h(z) \geq \frac{1}{2}\|z_0-x\|^2 + h(z_0).$$

We have

$$\|z-x\| = \|z-z_0\|^2 + 2\langle z-z_0, z_0-x\rangle + \|z_0-x\|^2$$

so

$$\frac{1}{2}\|z-z_0\|^2 + \langle z-z_0, z_0-x\rangle + h(z) \geq h(z_0).$$

Since h is convex we can drop $\|z-z_0\|^2$ from the left hand side.

Proximal operator

Namely

$$\langle z - z_0, z_0 - x\rangle + h(z) \geq h(z_0) - \frac{1}{2}\|z - z_0\|^2.$$

Graph of convex function has supporting hyperplane at $z = z_0$, but due to inequality above the only possible supporting plane is $t = h(z_0)$. So, we have

$$\langle z - z_0, z_0 - x\rangle + h(z) \geq h(z_0).$$

Now, applying this to $x = x_1$, $z_0 = z_1 = \mathrm{prox}_h(x_1)$ and z_2 we get

$$\langle z_2 - z_1, z_1 - x_1\rangle + h(z_2) \geq h(z_1).$$

By symmetry, when $z_2 = \mathrm{prox}_h(x_2)$ we get

$$\langle z_1 - z_2, z_2 - x_2\rangle + h(z_1) \geq h(z_2).$$

Proximal operator

Adding both together we get

$$\langle z_2 - z_1, (z_1 - x_1) - (z_2 - x_2)\rangle + h(z_2) + h(z_1) \geq h(z_1) + h(z_2)$$

so

$$\langle z_2 - z_1, (z_1 - z_2) + (x_2 - x_1)\rangle \geq 0$$

so

$$\langle z_2 - z_1, x_2 - x_1\rangle - \|z_2 - z_1\| \geq 0$$

and indeed

$$\langle z_2 - z_1, x_2 - x_1\rangle \geq \|z_2 - z_1\|^2$$

□

Moreau decomposition

Lemma

Let $h^*(x) = \sup_z \langle x, z\rangle - h(z)$ is Legendre transform of h. Then

$$x = \mathrm{prox}_h(x) + \mathrm{prox}_{h^*}(x)$$

Proof: When $u = \mathrm{prox}_h(x)$, then as in previous lemma we have

$$\langle z - u, u - x\rangle + h(z) \geq h(x)$$

that is

$$\langle x - u, u\rangle - h(x) \geq \langle x - u, z\rangle - h(z)$$

so supremum in definition of $h^*(x - u)$ is attained at $z = u$.
Moreover

$$h^*(z) \geq \langle z, u\rangle - h(u) = \langle z - (x - u), u\rangle + h^*(x - u).$$

Moreau decomposition

We have $z - x = (z - (x - u)) - u$ so

$$\frac{1}{2}\|z - x\|^2 = \frac{1}{2}\|z - (x - u)\|^2 - \langle z - (x - u), u\rangle + \frac{1}{2}\|u\|^2.$$

Adding inequality for $h^*(z)$ we get

$$\frac{1}{2}\|z - x\|^2 + h^*(z) \geq \frac{1}{2}\|z - (x - u)\|^2 + \frac{1}{2}\|u\|^2 + h^*(x - u).$$

$$\geq \frac{1}{2}\|u\|^2 + h^*(x - u).$$

We have equality when $z = x - u$, so

$$\mathrm{prox}_{h^*}(x) = x - u$$

and

$$\mathrm{prox}_h(x) + \mathrm{prox}_{h^*}(x) = u + x - u = x.$$

□

Moreau decomposition

Example: When C is a convex cone, than $I_C^* = I_{-C^*}$ where C^* is dual cone: $C^* = \{z : \langle z, x\rangle \geq 0 \quad \text{for all } x \in C\}$. So we get

$$x = \mathrm{Proj}_C(x) + \mathrm{Proj}_{-C^*}(x).$$

Example: When C is closed and convex we define support function S_C as

$$S_C(x) = \sup_{z \in C} \langle x, z\rangle.$$

Then

$$\mathrm{prox}_{S_C}(x) = x - \mathrm{Proj}_C(x)$$

Namely, $S_C = I_C^*$ so

$$x = \mathrm{prox}_{S_C}(x) + \mathrm{prox}_{I_C}(x) = \mathrm{prox}_{S_C}(x) + \mathrm{Proj}_C(x).$$

Moreau decomposition

Example: $\text{prox}_{\|\cdot\|_1}(x) = x - \text{Proj}_{B_\infty}(x)$ where B_∞ is unit ball in l^∞ norm. This is easy to compute explicitly and gives another derivation of soft thresholding operator.
Example: $\text{prox}_{\|\cdot\|_\infty}(x) = x - \text{Proj}_{B_1}(x)$ where B_1 is unit ball in l^1 norm. Projection onto B_1 can be computed efficiently.
Note: in general when B is unit ball in some norm, then S_B is the norm.

Moreau decomposition

Example: Let $h(x) = \max\{x_1, \ldots, x_n\}$. Then

$$\operatorname{prox}_h(x) = x - \operatorname{Proj}_\Delta(x)$$

where

$$\Delta = \{t : t \geq 0, \sum_{j=1}^{n} t_j = 1\}$$

is probability simplex.
Indeed, again $h = S_\Delta$.

Example projections

Example: Let

$$\Delta = \{t : t \geq 0, \sum_{j=1}^{n} t_j = 1\}$$

be probability simplex. Recall that

$$\mathrm{Proj}_\Delta(x) = \mathrm{argmin}_{z \in \Delta} \frac{1}{2}\|z - x\|^2.$$

We claim that optimality condition in minimization above is

$$z = (x - \lambda 1)_+$$

where 1 is vector with all coordinates equal to 1 and $(x - \lambda 1)_+$ means that we replace negative coordinates of $x - \lambda 1$ by 0. λ can be determined from condition

$$\|(x - \lambda 1)_+\|_1 = 1.$$

Example projections

Namely, let $I = \{i : z_i = 0\}$. Write

$$f(z) = \frac{1}{2}\|z - x\|^2,$$

$$g_0(z) = -1 + \sum_{i=1}^{n} z_i = \langle z, 1\rangle - 1,$$

$$g_i(z) = -z_i.$$

Clearly, finding $\mathrm{Proj}_\Delta(x)$ is equivalent to finding z which minimizes $f(z)$ under constraints $g_0(z) = 0$, $g_i(z) \leq 0$ for $i = 1, \dots, n$. I is set of active inequality constraints. Since $\sum_{i=1}^{n} z_i = 1$ at least one inequality constraint is inactive. Hence, $\nabla g_i(z)$ with $i \in \{0\} \cup I$ are linearly independent. By KKT conditions we have

$$\nabla f(z) - c_0 \nabla g_0(z) + \sum_{i \in I} c_i \nabla g_i(z) = 0$$

Example projections

Since $\nabla f(z) = z - x$, in coordinates we have:

$$z_i - x_i - c_0 = 0$$

for $i \notin I$ and

$$z_i - x_i - c_0 - c_i = 0$$

for $i \in I$. Since $c_i \geq 0$ and $z_i = 0$ for $i \in I$ this implies

$$-x_i \geq c_0$$

so

$$x_i \leq -c_0.$$

Put $\lambda = -c_0$. By inequality above for $i \in I$ we have

$$(x_i - \lambda)_+ = 0 = z_i$$

For $i \notin I$ we have

$$z_i - x_i - c_0 = z_i - x_i + \lambda.$$

Example projections

Hence, for $i \notin I$

$$z_i = x_i - \lambda$$

For $i \notin I$ we have $z_i > 0$ so

$$z_i = (x_i - \lambda)_+.$$

So, we proved that

$$z = (x - \lambda 1)_+.$$

Note that

$$\sum_{i=1}^{n} (x_i - \lambda)_+$$

decreases when λ is increased, so λ is uniquely determined by condition

$$\sum_{i=1}^{n} (x_i - \lambda)_+ = 1$$

Example projections

It remains to give efficient method to find λ. Note that

$$\sum_{i=1}^{n}(x_i - \lambda)_+ = \sum_{i \notin I}(x_i - \lambda) = -\lambda(n - |I|) + \sum_{i \notin I} x_i$$

where $|I|$ denotes number of elements of I. So once I is known we can easily compute λ. To find I we use bisection. More precisely, we take trial value of λ. For correct λ we have $I = \{i : z_i \leq \lambda\}$. If trial λ gives too large sum, then we need to increase λ, when sum is too small we need to decrease λ. We take one of x_i as trial value, starting from median. Median and sums can be computed in linear time and after single trial problem size is reduced by half so whole computation can be done in linear time (one needs to reuse previously computed sums).

Example projections

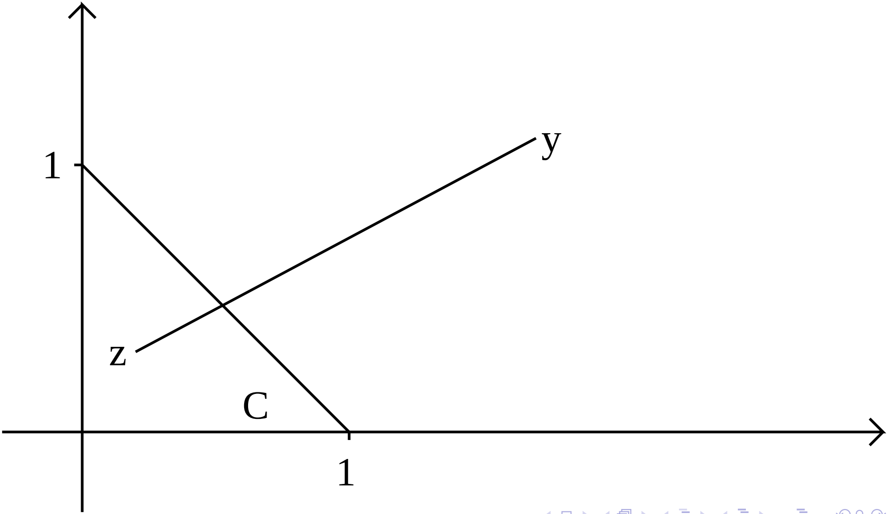
Example: Let $C = \{t : t_j \geq 0, \sum_{j=1}^{n} t_j \leq 1\}$. This can be reduced to previous one. Procedure is as follows:

1. Let y be x with negative coordinates replaced by 0
2. If y is in C then return it, otherwise return $\mathrm{Proj}_\Delta(y)$

To justify this, first note that when for some i coordinate x_i is negative, then $(Proj_C(x))_i = 0$. Namely, if $(Proj_C(x))_i$ were positive, then replacing $(Proj_C(x))_i$ by 0 we would get point in C which is closer to x. So $\mathrm{Proj}_C(x) = \mathrm{Proj}_C(y)$. Of course, when $y \in C$, then $\mathrm{Proj}_C(y) = y$. Otherwise line joining y and $\mathrm{Proj}_C(y)$ must have common point with hyperplane given by equation $\sum_{j=1}^{n} y_j = 1$. Since distance of this point to y is not bigger than distance between $\mathrm{Proj}_C(y)$ and y, the common point is $\mathrm{Proj}_C(y)$, that is $\mathrm{Proj}_C(y) \in \Delta$, so $\mathrm{Proj}_C(y) = \mathrm{Proj}_\Delta(y)$.

Example projections

Two dimensional situation is illustrated on the picture (where we minimize distance between $z \in C$ and fixed y):



Example projections

Example: Projection onto l^1 unit ball. Let $s_i = 1$ when $x_i \geq 0$ and $s_i = -1$ when $x_i < 0$. Let operator S be defined by equality $S(z)_i = s_i z_i$. We have $S(x) \geq 0$. But we also have $S(B_1) = B_1$ where B_1 is unit ball in l^1 norm. Since $\|S(x) - S(y)\|_2 = \|x - y\|_2$ we have

$$\text{Proj}_{B_1}(S(z)) = S(\text{Proj}_{B_1}(z))$$

Put $y = S(x)$. We have $x = S(y)$ so

$$\text{Proj}_{B_1}(x) = S(\text{Proj}_{B_1}(y)).$$

To compute $\text{Proj}_{B_1}(y)$ note that $\text{Proj}_{B_1}(y) = \text{Proj}_C(y)$ (this is similar to what happened for C).

Proximal gradient algorithm

Lemma

When gradient g is Lipschitz continuous with constant M, with constant step $\alpha_i = \frac{1}{M}$ we have

$$f(x_{i+1}) \leq f(x_i).$$

Assuming optimal point x_∞ exist we have

$$\|x_{i+1} - x_\infty\| \leq \|x_i - x_\infty\|$$

$$f(x_i) - f(x_\infty) \leq \frac{M\|x_0 - x_\infty\|^2}{2i}.$$

In other words both objective function and distance to optimum is nonincreasing and we have convergence.

Proximal gradient algorithm

Lemma

If $mI \leq \nabla^2 g(x) \leq MI$*, with constant step* $\alpha_i = \frac{1}{M}$ *we have*

$$\|x_i - x_\infty\|^2 \leq (1 - \frac{m}{M})^i \|x_0 - x_\infty\|^2$$

Remark: This estimate is essentially the same as for steepest descent.

Proximal gradient algorithm

Convergence analysis of proximal gradient algorithm is based on following lemma:

Lemma

If gradient of g is Lipschitz continuous with constant M, $z = \mathrm{prox}_{\frac{1}{M}h}(y - \frac{1}{M}\nabla g(y))$, *then for all x*

$$f(z) - f(x) \leq \frac{M}{2}\|x - y\|^2 - \frac{M}{2}\|x - z\|^2 - r(x, y)$$

where $r(x, y) = g(x) - g(y) - \langle \nabla g(y), x - y \rangle \geq 0.$

Proof: $r(x, y) \geq 0$ by convexity. Put

$$\phi(t) = g(y) + \langle \nabla g(y), t - y \rangle + \frac{M}{2}\|t - y\|^2 + h(t)$$

$$= g(y) - \frac{M}{2}\|\frac{1}{M}\nabla g(y)\|^2 + \frac{M}{2}\|t - (y - \frac{1}{M}\nabla g(y))\|^2 + h(t)$$

Proximal gradient algorithm

so $z = \operatorname{argmin}_t \phi(t)$. We can rewrite ϕ as

$$\phi(x) = \psi(x) + \frac{M}{2}\|x - z\|^2$$

with convex ψ. Note that ψ must have minimum at z. Namely, in opposite case there would be v such that

$$\psi(z + v) < \psi(z).$$

By convexity for $0 < t < 1$ we have

$$\psi(z + tv) \leq t\psi(z + v) + (1 - t)\psi(z)$$

$$= t(\psi(z + v) - \psi(z)) + \psi(z).$$

If $\psi(z + v) - \psi(z) < 0$, then for small positive t term $t(\psi(z + v) - \psi(z))$ has bigger absolute value than $\|z + tv - z\|^2 = t^2\|v\|^2$, so also $\phi(z + tv)$ would be smaller than $\phi(z)$.

Proximal gradient algorithm

Hence, we have

$$\phi(x) \geq \phi(z) + \frac{M}{2}\|x - z\|^2.$$

On the other hand

$$g(z) \leq g(y) + \langle \nabla g(y), t - y \rangle + \frac{M}{2}\|t - y\|^2$$

so

$$f(z) \leq \phi(z).$$

Together we get

$$\phi(x) \geq f(z) + \frac{M}{2}\|x - z\|^2.$$

Proximal gradient algorithm

Now, by definition of ϕ:

$$g(y) + (\langle \nabla g(y), x - y \rangle + \frac{M}{2}\|x - y\|^2 + h(x) \geq f(z) + \frac{M}{2}\|x - z\|^2.$$

But

$$\begin{aligned} & g(y) + \langle \nabla g(y), x - y \rangle + h(x) \\ = {} & g(x) + h(x) + (g(y) - g(x) + \langle \nabla g(y), x - y \rangle) = f(x) - r(x, y) \end{aligned}$$

so

$$f(x) - r(x, y) + \frac{M}{2}\|x - y\|^2 \geq f(z) + \frac{M}{2}\|x - z\|^2$$

and finally

$$f(z) - f(x) \leq \frac{M}{2}\|x - y\|^2 - \frac{M}{2}\|x - z\|^2 - r(x, y)$$

Proximal gradient algorithm

Taking $x = y = x_i$ in the lemma we get $z = x_{i+1}$ and

$$f(x_{i+1}) - f(x_i) \leq -\frac{M}{2}\|x_i - x_{i+1}\|^2 - r(x, y) \leq 0$$

so

$$f(x_{i+1}) \leq f(x_i)$$

Taking $x = x_\infty$, $y = x_i$ we get $z = x_{i+1}$ and

$$f(x_{i+1}) - f(x_\infty) \leq \frac{M}{2}\|x_i - x_\infty\|^2 - \frac{M}{2}\|x_{i+1} - x_\infty\|^2 - r(x, y)$$

so

$$0 \leq \frac{M}{2}\|x_i - x_\infty\|^2 - \frac{M}{2}\|x_{i+1} - x_\infty\|^2$$

so

$$\|x_{i+1} - x_\infty\| \leq \|x_i - x_\infty\|.$$

Proximal gradient algorithm

Using estimate above in better way we get

$$f(x_{i+1}) - f(x_\infty) \leq \frac{M}{2}\|x_i - x_\infty\|^2 - \frac{M}{2}\|x_{i+1} - x_\infty\|^2$$

Adding up inequalities for $j = 0, \ldots, i-1$ we get

$$\sum_{j=0}^{i-1}(f(x_{j+1}) - f(x_\infty)) \leq \frac{M}{2}\|x_0 - x_\infty\|^2 - \frac{M}{2}\|x_i - x_\infty\|^2.$$

But we know $f(x_i) \leq f(x_{j+1})$ so

$$f(x_i) - f(x_\infty) \leq \frac{M}{2i}\|x_0 - x_\infty\|^2$$

Proximal gradient algorithm

Using assumption $mI \leq \nabla^2 g(x)$ we see that

$$r(x,y) \geq \frac{m}{2}\|x-y\|^2$$

so main lemma gives

$$f(x_{i+1}) - f(x_\infty) \leq \frac{M}{2}\|x_i - x_\infty\|^2 - \frac{M}{2}\|x_{i+1} - x_\infty\|^2 - \frac{m}{2}\|x_i - x_\infty\|^2$$

so

$$0 \leq \frac{M-m}{2}\|x_i - x_\infty\|^2 - \frac{M}{2}\|x_{i+1} - x_\infty\|^2$$

that is

$$\|x_{i+1} - x_\infty\|^2 \leq \frac{M-m}{M}\|x_i - x_\infty\|^2 = (1 - \frac{m}{M})\|x_i - x_\infty\|^2$$

which proves rate of convergence in strictly convex case.

Proximal gradient algorithm

There are similar results for proximal gradient algorithm with line search. In line search we start with reasonably large α, check if

$$g(x_i - \alpha d_i) \leq g(x_i) - \alpha\langle \nabla g(x_i), d_i\rangle + \frac{\alpha}{2}\|d_i\|^2$$

where d_i is direction and otherwise we shrink α. The condition above allows proofs of convergence rate (with slightly worse constant) to go on.

Accelerated proximal gradient

We can improve convergence using variant of Nesterov acceleration. Put $x_{-1} = x_0$ and

$$v_i = x_i + \frac{i-1}{i+2}(x_i - x_{i-1})$$

$$x_{i+1} = \mathrm{prox}_{\alpha_i h}(v_i - \alpha_i \nabla g(v_i))$$

First two steps are just usual proximal proximal gradient steps, other contain momentum term.

FISTA

Accelerated proximal gradient algorithm applied to LASSO problem is called FISTA

Further reading

This material is relatively new and most is *not* present in the books we use. More examples of support functions, Legendre transform and projection is in:
Stephen Boyd, Lieven Vandenberghe, Convex Optimization, chapter 2, 3, 5 and 8.
Research level survey:
Neal Parikh, Stephen Boyd, Proximal Algorithms, Foundations and Trends in Optimization Vol. 1, No. 3 (2013) 123–231.