Lecture 12

Waldemar Hebisch

January 11, 2022

1 ADMM

1.1 Dual ascent

Let f be a convex function. Consider first problem of minimizing

$$f(x)$$

under constraint $Ax = b$. In dual approach we first write Lagrangian

$$L(x, \lambda) = f(x) + \langle \lambda, Ax - b \rangle$$

and dual function

$$g(\lambda) = \inf_x L(x, \lambda) = -f^*(-A^T\lambda) - \langle \lambda, b \rangle$$

where f^* is conjugate (Legendre transform) of f. Dual problem is to maximize $g(\lambda)$. Assuming that strong duality holds, the optimal values of the primal and dual problems are the same.

In the dual ascent method, we solve the dual problem using gradient ascent. Assuming that g is differentiable, the gradient $\nabla g(\lambda)$ can be evaluated as follows. We first find $v = \operatorname{argmin}_x L(x, \lambda)$, then we have $\nabla g(\lambda) = Av - b$, which is the residual for the equality constraint. The dual ascent method consists of iterating the updates

$$x_{i+1} = \operatorname{argmin}_x L(x, \lambda_i),$$
$$\lambda_{i+1} = \lambda_i + \alpha_i(Ax_{i+1} - b)$$

where $\alpha_i > 0$ is a step size.

Under rather restrictive conditions dual ascent converges to optimal x and λ.

Important benefit of the dual ascent method is that it can lead to a decentralized algorithm in some cases. Suppose, for example, that the objective f is separable (with respect to a partition or splitting of the variable into subvectors), meaning that

$$f(x) = \sum_j f_j(x_j)$$

where x_j are subvectors of $x = (x_1, \ldots, x_m)$. Then x-minimization can be split into separate minimizations over x_j (which can be performed in parallel). This approach is called *dual decomposition*.

Remark: Without constraints we could just optimize f_j separately. However, equality constraints introduce implicit dependence. Dual decomposition handles equality constraints.

1.2 Augmented Lagrangian

To improve robustness we may add penalty term to the Lagrangian, that is write:

$$L(x, \lambda) = f(x) + \langle \lambda, Ax - b \rangle + \frac{p}{2} \|Ax - b\|^2$$

where p is called penalty parameter (and L is now called augmented Lagrangian). Variant of dual ascent using augmented Lagrangian above and p as step size is called *method of multipliers* and has much better convergence properties than ordinary dual ascent.

Unfortunately, L above is no longer separable.

1.3 ADMM

ADMM (Alternating Direction Method of Multipliers) is intended to give good convergence properties of augmented Lagrangian and preserve separability.

Assume that f and g are convex. Consider problem of minimizing

$$f(x) + g(y)$$

subject to

$$Ax + By = c$$

Augmented Lagrangian is

$$L(x, y, \lambda) = f(x) + g(y) + \langle Ax + By - c, \lambda \rangle + \frac{p}{2} \|Ax + By - c\|^2$$

ADMM consists of the following iterations:

$$x_{i+1} = \operatorname{argmin}_x L(x, y_i, \lambda_i),$$

$$y_{i+1} = \operatorname{argmin}_y L(x_{i+1}, y, \lambda_i),$$

$$\lambda_{i+1} = \lambda_i + p(Ax_{i+1} + By_{i+1} - c).$$

Note that since x_{i+1} is computed without using x from previous iteration state of ADMM consists of y_i and λ_i.

In initial problem x and y play symmetric role, but in ADMM there is slight asymmetry.

Under rather weak regularity assumptions and assumption that L has saddle point we have

- goal function convergence to optimal value
- residual convergence, that is points converge to a feasible point
- dual variable convergence

There is convergence rate estimate in terms of optimal λ, main point is that error decreases at least as $O(\frac{1}{i})$.

For ADMM necessary and sufficient condition is primal feasibility

$$Ax + By - c = 0$$

and dual feasibility, that is (assuming differentiable f and g, in general need to consider subgradients)

$$\nabla f(x) + A^T\lambda = 0$$

and

$$\nabla g(y) + B^T\lambda = 0.$$

In fact g is minimized in second part of update, so if other conditions hold condition on g will be automatically satisfied. x_{i+1} is also obtained by minimization so

$$\nabla f(x_{i+1}) + A^T\lambda + pA^t(Ax_{i+1} + By_k - c) = 0.$$

Consequently we can treat

$$B(y_{i+1} - y_i)$$

as primary residual (measure of error). If this and

$$Ax + By - c$$

are small enough we are close to optimum.

Note that when $A = I$, then x minimization in ADMM is equivalent to computing proximal operator. If this operator is easy to compute, the that step will be quick. Similarly when $B = I$, then y minimization in ADMM is equivalent to computing proximal operator.

Example: Consensus problem. Minimize:

$$\sum_{j=1}^{m} f_j(x).$$

Equivalently, minimize

$$\sum_{j=1}^{m} f_j(x_j)$$

under constraints $x_j = x_k$ for all k, j. Let $V = \{x : \forall_{k,j} x_j = x_k\}$ and denote by I_V indicator function of V.

We can now rewrite problem as minimization of

$$\sum_{j=1}^{m} f_j(x_j) + I_V(y)$$

under constraint $x - y = 0$. Assuming $p = 1$ we can write

$$L(x, y, \lambda) = \sum_{j=1}^{m} f_j(x_j) + I_V(y) + \langle x - y, \lambda \rangle + \frac{1}{2}\|x - y\|^2$$

$$= \sum_{j=1}^{m} f_j(x_j) + I_V(y) + \frac{1}{2}(\|x - y + \lambda\|^2 - \|\lambda\|^2).$$

Now, x minimization is just computation of proximal operator and can be done separately for blocks:

$$(\text{argmin}_x\, L(x, y, \lambda))_j = (\text{prox}_{h(x)}(y - \lambda))_j = \text{prox}_{f_j}(y_j - \lambda_j)$$

where $h(x) = f_1(x_1) + \cdots + f_m(x_m)$.

The y minimization above is just projection onto V, which is easy to do: for x in V each x_j must be equal to average $\overline{x}$. More precisely

$$(\text{argmin}_y\, L(x, y, \lambda))_j = \text{Proj}_V(x + \lambda)_j = \overline{x + \lambda}.$$

The λ step is now

$$(\lambda_{i+1})_j = (\lambda_i)_j + (x_j - \overline{(x + \lambda_i)}).$$

Note that after that step we will have $\overline{\lambda_{i+1}} = 0$, so we can just start from λ_0 such that $\overline{\lambda_0} = 0$. Then y and λ update simplify to

$$(y_{i+1})_j = \overline{x},$$

$$(\lambda_{i+1})_j = (\lambda_i)_j + (x_j - \overline{x}).$$

Example: Low rank approximation. Consider problem of minimizing

$$\|A - S - L\|_{HS}^2 + c\|S\|_1 + d\|L\|_*$$

where A is known matrix, $\|\cdot\|_{HS}$ denotes elementwise L^2 norm (Hilbert-Schmidt norm), $\|\cdot\|_1$ denotes elementwise L^1 norm and $\|\cdot\|_*$ is nuclear norm. $\|\cdot\|_1$ encourages sparsity, $\|\cdot\|_*$ encourages low rank, so the problem is to approximate A by sum of sparse matrix and low rank matrix. In other words, when true matrix has low rank S allows small number of wrong entries (outliers) and $\|\cdot\|_2$ allows for some random noise. to simplify notation we will write $\|\cdot\|$ instead of $\|\cdot\|_{HS}$. We can rewrite the above as

$$\|C\|^2 + c\|S\|_1 + d\|L\|_* + g(C, S, L)$$

where g is indicator function of hyperplane V with equation $C + S + L = A$ and treat tuple (C, S, L) as x and use constraint $x = y$.

Then, writing $y = (\tilde{C}, \tilde{S}, \tilde{L})$ we get augmented Lagrangian

$$L(C, S, L, \tilde{C}, \tilde{S}, \tilde{L}, \lambda) = \|C\|^2 + c\|S\|_1 + d\|L\|_* + g(\tilde{C}, \tilde{S}, \tilde{L})$$
$$+\langle C - \tilde{C}, \lambda_C\rangle + \langle S - \tilde{S}, \lambda_S\rangle + \langle L - \tilde{L}, \lambda_L\rangle$$
$$+\frac{p}{2}(\|C - \tilde{C}\|^2 + \|S - \tilde{S}\|^2 + \|L - \tilde{L}\|^2)$$

where w denote by λ_C, λ_S and λ_L parts of λ corresponding to C, S and L. To simplify notation in the following we take $p = 1$.

Like in consensus problem we can simplify Lagrangian

$$L(C, S, L, \tilde{C}, \tilde{S}, \tilde{L}, \lambda) = \|C\|^2 + c\|S\|_1 + d\|L\|_* + g(\tilde{C}, \tilde{S}, \tilde{L})$$
$$\frac{1}{2}(\|C - \tilde{C} + \lambda_C\|^2 + \|S - \tilde{S} + \lambda_S\|^2 + \|L - \tilde{L} + \lambda_L\|^2$$
$$-(\|\lambda_C\|^2 + \|\lambda_S\|^2 + \|\lambda_L\|^2)).$$

The x minimization then can be done separately for C, S and L and reduces to proximal operators

$$\operatorname{argmin}_{(C,S,L)} L(C, S, L, \tilde{C}, \tilde{S}, \tilde{L}, \lambda)$$
$$= (\operatorname{prox}_{\|\cdot\|^2}(\tilde{C} - \lambda_C), \operatorname{prox}_{\|\cdot\|_1}(\tilde{S} - \lambda_S), \operatorname{prox}_{\|\cdot\|_*}(\tilde{L} - \lambda_L)).$$

Next, since g is indicator of hyperplane V, y minimization is just projection onto V. This projection is easy to compute: just subtract average and add $A/3$. Explicitly

$$\tilde{C} = C + \lambda_C - r,$$
$$\tilde{S} = S + \lambda_S - r,$$
$$\tilde{L} = L + \lambda_L - r,$$

where

$$r = \frac{1}{3}(C + S + L - (\lambda_C + \lambda_S + \lambda_L) - A)$$

Finally λ update is

$$\lambda_{i+1} = \lambda_i + ((C_{i+1}, S_{i+1}, L_{i+1}) - (\tilde{C}, \tilde{S}, \tilde{L}))$$
$$= \lambda_i + (C_{i+1}, S_{i+1}, L_{i+1}) - ((C_{i+1}, S_{i+1}, L_{i+1}) + \lambda_i - (r, r, r))$$
$$= (r, r, r).$$

Note that after first step $\lambda_C = \lambda_S = \lambda_L$, so we can assume that it also holds in initial state and just use single λ. This simplifies projection onto V, contribution from λ vanishes.

Now we can rewrite our formulas in simplified form

$$C_{i+1} = \operatorname{prox}_{\|\cdot\|^2}(\tilde{C}_i - \lambda_i),$$

$$S_{i+1} = \text{prox}_{\|\cdot\|_1}(\tilde{S}_i - \lambda_i),$$
$$L_{i+1} = \text{prox}_{\|\cdot\|_*}(\tilde{L}_i - \lambda_i),$$
$$r = \frac{1}{3}(C_{i+1} + S_{i+1} + L_{i+1} - A)$$
$$\tilde{C}_{i+1} = C_{i+1} - r,$$
$$\tilde{S}_{i+1} = S_{i+1} - r,$$
$$\tilde{L}_{i+1} = L_{i+1} - r,$$
$$\lambda_{i+1} = \lambda_i + r.$$

Above x minimization reduces to computation of proximal operators. Proximal operator for L^2 norm is trivial. Proximal operator for L^1 norm can be computed quite efficiently. Proximal operator for nuclear norm needs SVD, so is more expensive, but moderately so. Other operations are quite cheap.

1.4 Further reading 1

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Foundations and Trends in Machine Learning Vol. 3, No. 1 (2010) 1–122

2 Subgradient methods

When goal function is non-differentiable then gradient descent may have trouble at point where derivative does not exist. Worse, even if derivative exists at all point visited during descent gradient descent with exact line search may converge to non-optimal point.

Recall special case of example from lecture 4:

$$f(x) = \frac{1}{2}(x_1^2 + \frac{1}{2}x_2^2).$$

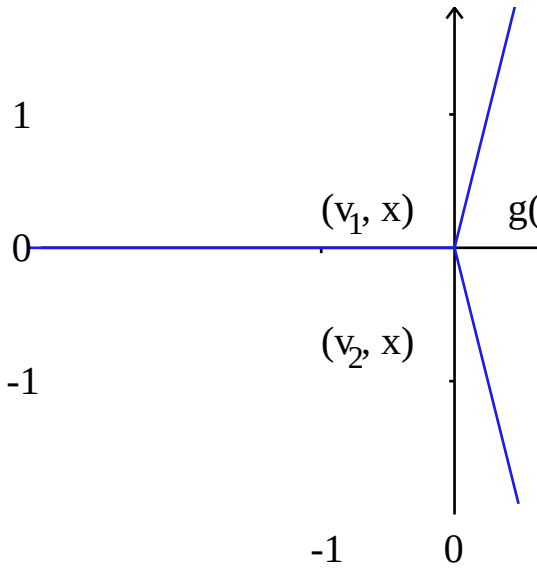
Starting from $x_0 = (\frac{1}{2}, 1)$ all x_i stay in sector $|x_1| \leq 2x_2$. Let $S = \{x : |x_1| \leq 4x_2\}$, $v_1 = (\frac{1}{2}, \frac{1}{16})$, $v_2 = (-\frac{1}{2}, \frac{1}{16})$. Let

$$g(x) = \frac{1}{8}\sqrt{33f(x)}.$$

Put $h(x) = g(x)$ for $x \in S$ and

$$h(x) = \max(\langle v_1, x\rangle, \langle v_2, x\rangle)$$

otherwise. Note that we glued h from pieces in such way that derivatives agree, except for x-es on half-line $l = \{x : x_1 = 0, x_2 \leq 0\}$. Consequently h has continuous derivative on $\mathbb{R}^2 - l$.



The following picture shows where we use various definitions:

h is convex. To check this we look at restriction of h to lines. On each line derivative of restriction is non-decreasing. This is clear in neighbourhood of points not in l (close to such point g is glued from two convex pieces and due to continuity of derivative we can add increments coming from both pieces). In neighbourhood of points from l with $x_2 < 0$ our h as maximum of two convex functions is convex, so also derivative can not decrease. Finally, h restricted to a line passing trough 0 consists of two linear pieces and one easily checks that restriction is convex.

Now, for $x \in S$ we have

$$\nabla h(x) = c\frac{\nabla f(x)}{2\sqrt{cf(x}}$$

so derivative of h points in the same direction as derivative of f. Consequently we do line search for h on the same line as for f and exact line search gives the same points. So, starting from $x_0 = (\frac{1}{2}, 1)$ all x_i will stay in S and converge to 0. But h is unbounded from below. Taking maximum of h and appropriate affine function we can produce convex function which agrees with h on S and attains minimum at some point different from 0.

Problems with gradient descent:

- have trouble at points where derivative does not exist
- with exact line search can converge to nonoptimal point
- with constant step size can diverge
- can not use line search to choose step size

Solution:

- use subgradient instead of gradient
- use predetermined step sizes α_i which decay to zero when i goes to infinity.

Definition: we say that vector v is in subgradient set of f at x when

$$\liminf_{w\to 0} \frac{f(x+w)-f(x)-\langle v,w\rangle}{\|w\|} \geq 0.$$

In such case we write $v \in \partial f(x)$. If f is differentiable at x, then clearly $\partial f(x) = \{\nabla f(x)\}$.

When f is defined on a subset S then in formula above x and $x+w$ are restricted to elements of S.

Note that in general $\partial f(x)$ is a closed convex set.

We say that f is subdifferentiable at x if x is in domain of f and $\partial f(x) \neq \emptyset$.

Example: Let $f(x) = \sin(1/x^2) + 1 - x^2$ for $x \neq 0$ and $f(0) = 0$ on $\mathbb{R}$. We have

$$f(x) - f(0) \geq -x^2.$$

Consequently

$$\liminf_{w\to 0} \frac{f(0+w)-f(0)-0\cdot w}{|w|} \geq \liminf_{w\to 0} \frac{-w^2}{|w|} = 0$$

so $0 \in \partial f(0)$.

When $v > 0$ we take $w_n = ((3/2+2n)\pi)^{-1/2}$ we have $w_n \to 0$, $\sin(1/w_n^2) = -1$ and $f(w_n) = -w_n^2$ so

$$\liminf \frac{f(0+w_n)-f(0)-vw_n}{|w_n|} = \liminf \frac{-w_n^2 - vw_n}{w_n} = -v < 0.$$

Consequently $v \notin \partial f(0)$. Similarly, when $v < 0$ then $v \notin \partial f(0)$. So $\partial f(0) = \{0\}$.

Example: Let $f(x) = \sin(1/x^2)$ for $x \neq 0$ and $f(0) = 0$. f is not subdifferentiable at 0.

Lemma 2.1 *When f is convex and $x \in \mathrm{dom}(f)$, then $v \in \partial f(x)$ if and only if plane parameterized by $\phi(y) = \langle v, y-x\rangle + f(x)$ is supporting plane for* $\mathrm{epi}(f)$.

Proof: When ϕ parametrizes supporting plane for $\mathrm{epi}(f)$ we have

$$f(x+w) \geq f(x) + \langle v, x+w-x\rangle = f(x) + \langle v, w\rangle$$

so

$$\frac{f(x+w)-f(x)-\langle v,w\rangle}{\|w\|} \geq 0$$

and consequently

$$\liminf_{w\to 0} \frac{f(x+w)-f(x)-\langle v,w\rangle}{\|w\|} \geq 0,$$

that is $v \in \partial f(x)$.

Conversely let $v \in \partial f(x)$. Consider u such that $\|u\| = 1$. Let $\varepsilon > 0$. Put $w = tu$. Since $v \in \partial f(x)$, by definition of lim inf there exists δ such that

$$\frac{f(x+tu) - f(x) - \langle v, tu\rangle}{\|tu\|} \geq -\varepsilon$$

for $0 < t < \delta$.

That is

$$f(x+tu) - f(x) - \langle v, tu\rangle \geq -\varepsilon t$$

for $0 < t < \delta$. Now, by convexity this holds for all $t > 0$. Since $\varepsilon > 0$ were arbitrary we get

$$f(x+tu) - f(x) - \langle v, tu\rangle \geq 0.$$

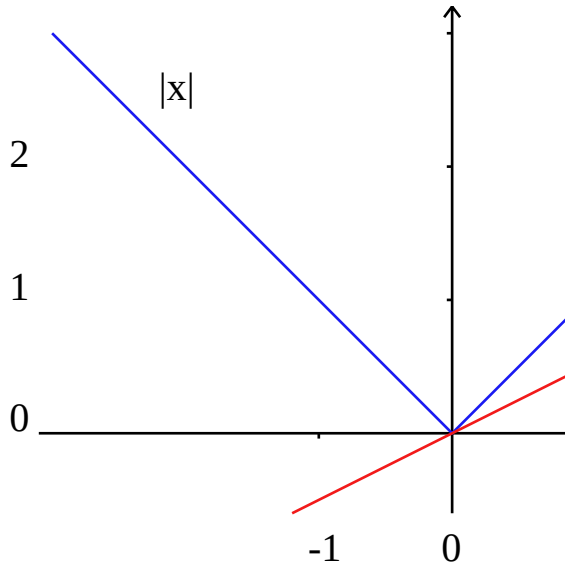
Now, for arbitrary $w \neq 0$ we put $u = \frac{w}{\|w\|}$ and write $w = tu$ with $t = \|w\|$. Applying inequality above we get

$$f(x+w) - f(x) - \langle v, w\rangle \geq 0.$$

Equivalently, with $w = y - x$

$$f(y) \geq f(x) - \langle v, y - x\rangle$$

But this means that ϕ parametrizes supporting plane for epi(f). □



Picture showing one of supporting lines for epigraph of $|x|$:

Note that lemma implies that for convex f and $x \in \text{dom}(f)$ set $\partial f(x)$ is always nonempty.

Example. Let $f(y) = |y|$ and $x = 0$. Clearly, $\phi(y) = vy$ parametrizes supporting line for epi(f) if and only if $v \in [-1, 1]$, so $\partial f(0) = [-1, 1]$. For $x \neq 0$ our f is differentiable and $\partial f(x) = \{\text{sign}(x)\}$.

We can formulate optimality condition in terms of subgradient:

Lemma 2.2 *If f has local minimum at $x \in \mathrm{dom}(f)$ then $0 \in \partial f(x)$. Conversely, if f is convex and $0 \in \partial f(x)$, then f has global minimum at x.*

Proof: If f has local minimum at x then

$$f(x+w) - f(x) \geq 0$$

for w close enough to 0 which like in previous lemma means $0 \in \partial f(x)$. Converse for convex functions follows from previous lemma. □

Remark: We have the following property analogous to optimality lemma used to derive KKT conditions:

Lemma 2.3 *If f has local minimum at $x \in \mathrm{dom}(f)$, then there exists $w \in \partial f(x)$ such that for all $v \in TS_x$ we have*

$$\langle w, v \rangle \geq 0$$

Namely, $w = 0 \in \partial f(x)$ satisfies conclusion above.

This is different than differentiable case inside interior of domain of f: if x in $\mathrm{Int}(\mathrm{dom}(f))$ and f is differentiable at x then $\partial f(x)$ has one element. However, if x is a boundary point of $\mathrm{dom}(f)$, than $\partial f(x)$ may be bigger, in particular it may contain both zero vector and a nonzero one.

We mention three properties that we will not use:

Lemma 2.4 *Convex and finite function is almost everywhere differentiable.*

In other words set of points where convex f is finite but not differentiable is of Lebesgue measure 0. Consequently, almost everywhere $\partial f(x)$ is a one point set.

Lemma 2.5 *Let f be convex and G be interior of the set where f is finite. The subgradient ∂f is upper semicontinuous on G that is for open U set*

$$\{x \in G : \partial f(x) \subset U\}$$

is open. Equivalently, if $x_i \to x \in G$, $w_i \in \partial f(x_i)$, $w_i \to w$, then $w \in \partial f(x)$.

Lemma 2.6 *Let f be strictly convex and G be interior of the set where f is finite. The subgradient ∂f is invertible on G, that is for $x_1, x_2 \in G$ we have $\partial f(x_1) \cap \partial f(x_2) = \emptyset$*

Rules for computation:

- $\partial(f+g)(x) = \partial f(x) + \partial g(x)$
- when g is differentiable then $\partial(f \circ g)(x) = \{h \cdot g'(x) : h \in \partial f(g(x))\}$

- when g is differentiable and nondecreasing, then $\partial(g \circ f)(x) = \{g'(f(x)) \cdot h : h \in \partial f(x)\}$
- when $f(x) = \max(f_1(x), \ldots, f_n(x))$, then

$$\partial f(x) = \text{conv}(\bigcup_{i:f(x)=f_i(x)} \partial f_i(x))$$

- for supremum there is much more complicated rule, when $f(x) = \sup_\alpha f_\alpha(x)$, then

$$\partial f(x) = \bigcap_{\varepsilon>0} \text{closure}(\text{conv}(\bigcup_{\alpha \in I(x,\varepsilon)} \partial f_\alpha(x)))$$

  where $I(x, \varepsilon) = \{\alpha : f(x) < f_\alpha(x) + \varepsilon\}$
- when $f_\alpha(x)$ is continuous in (x, α) and set of indices is compact for $f(x) = \sup_\alpha f_\alpha(x)$ we have

$$\partial f(x) = \text{closure}(\text{conv}(\bigcup_{\alpha:f(x)=f_\alpha(x)} \partial f_\alpha(x)))$$

  Remark: Note that in rule for sum we have addition of sets: $A + B = \{x + y : x \in A, y \in B\}$.

Example: $|x| = \max(x, -x)$.

Example: Put $S(x) = \partial|x|$. Then, by the formula for the sum $\partial\|x\|_1 = \prod S(x_i)$ (since we have sum of terms in disjoint variables sum of sets can be replaced by cartesian product, as we did).

Note: When $-v \in \partial f(x)$, then v need not to be descent direction for f at x. Let $f(x) = x_1 + 2|x_2|$. At $x = (1, 0)$ we have $\partial f(x) = 1 \times [-2, 2]$. In particular $v = (-1, 0)$ in satisfies $-v \in \partial f(x)$ and is a descent direction at x. But $v = (-1, -2)$ also satisfies $-v \in \partial f(x)$ and is not a descent direction at x.

Subgradient algorithm, given N and α_i:

- Step 1. Take arbitrary x_0. Put $y_{\max} = f(x_0)$, $x_{\max} = x_0$.
- Step 2. Take arbitrary v from $\partial f(x_i)$. Put $x_{i+1} = x_i - \alpha_i v$.
- Step 3. Increment i by 1.
- Step 4. If $f(x_i) < y_{\max}$, then put $y_{\max} = f(x_i)$, $x_{\max} = x_i$.
- Step 5. If $i = N$, then stop, otherwise go to step 2.

Remark: can also use proximal version when $f(x) = g(x) + h(x)$, $v \in \partial g(x)$ and

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

It remains to choose N and α_i. To have reasonable chance for convergence $\sum \alpha_i$ must be divergent.

Lemma 2.7 *If f is convex and Lipschitz continuous with Lipschitz constant M, f attains minimum at x_∞, then*

$$f(x_{\max}) - f(x_\infty) \leq \frac{\|x_0 - x_\infty\|^2 + M \sum_{i=0}^{N-1} \alpha_i^2}{2 \sum_{i=0}^{N-1} \alpha_i}$$

Consequence: For constant step size we may get only limited accuracy.

When M and $f(x_\infty)$ are known Polyak have found "optimal" choice of α_i, it the sense that they optimize estimate above. Since usually we want to find $f(x_\infty)$, Polyak rule is not practical. However $\alpha_i = \frac{1}{\sqrt{i+1}}$ is a reasonable choice, not far from Polyak rule. We get

$$f(x_{\max}) - f(x_\infty) \leq \frac{\|x_0 - x_\infty\|^2 + M \log(N)}{\sqrt{N}}$$

that is we need of order $\frac{1}{\varepsilon^2}$ (up to logarithmic factor) steps to attain accuracy ε.

2.1 Further reading

Yurii Nesterov, Introductory lectures on convex optimization, Springer 2004, chapter 3.