Lecture 5

W. Hebisch

March 22, 2023

1 Convex optimization

When $g_i : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ are convex functions, then $S = \{x : g_i(x) \leq 0\}$ is a convex set. When f is defined and convex on S then problem

$$\begin{array}{cl} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0 \end{array}$$

is called (constrained) convex optimization problem. For linear (more precisely affine) function g_i we can use equality constraint $g_i(x) = 0$, namely we write equality as conjunction of two inequalities $g_i(x) \leq 0$ and $-g_i(x) \geq 0$ (for affine g_i both g_i and $-g_i$ is convex, otherwise w could not do this).

Example: linear programming problem, LASSO.

Frequently constraints and goal function have very special form, for example quadratic goal function and linear constraints (quadratic optimization).

1.1 More examples of convex problems

Example: SVM. We want to find plane that best separates two finite sets A and B. We can write equation of separating plane P as $P = \{x : f(x) = 0\}$ where

$$f(x) = \langle x, \beta \rangle + \beta_0.$$

Distance to the plane is

$$d(x, P) = \frac{1}{\|\beta\|} |f(x)|$$

with sign of $f(x)$ deciding which half-space contains x. So, finding plane which maximizes which separates A and B and maximizes distance to A and B leads to problem

$$\begin{array}{cl} \text{minimize} & \|\beta\|^2 \\ \text{subject to} & \langle x, \beta \rangle + \beta_0 \geq 1, \quad \text{for } x \in A \\ \text{and} & \langle x, \beta \rangle + \beta_0 \leq -1, \quad \text{for } x \in B \end{array}$$

Here we minimize quadratic function with linear constrants.

Note: this problem has solution only when separating plane exists.

Example: robust linear programming. Suppose that constraints are known only approximately and we want to make sure that problem is feasible for all possible constraints. Reasonable assumption is that in constraint $\langle a_i, x\rangle \le b_i$ we know that a_i belongs to some ellipsoid. That is $a_i = w_i + P_i u_i$ where $\|u_i\| \le 1$. Here w_i is center and P_i is a positive definite matrix. Then

$$\langle a_i, x\rangle = \langle w_i, x\rangle + \langle P_i u_i, x\rangle = \langle w_i, x\rangle + \langle u_i, P_i x\rangle$$

and maximal value of $\langle u_i, P_i x\rangle$ term (as a function of u_i) is clearly $\|P_i x\|_2$. So we can rewrite problem as: minimize $\langle c, x\rangle$ under constraints

$$\langle w_i, x\rangle + \|P_i x\|_2 - b_i \le 0.$$

This is so called second order cone constraint.

Example: geometric programming. Consider problem of minimizing f_0 under constraints $f_i \le 1$ for $i = 1, \ldots, m$, $g_i = 1$ for $i = 1, \ldots, l$ and $x_i > 0$, $i = 1, \ldots, n$ where each f_i is of form

$$f_i(x) = \sum c_{i,\alpha} x^\alpha$$

and g_i is of form

$$d_i x_i^\beta$$

where α and β_i have real coordinates and $d_i > 0$, $c_{i,\alpha} > 0$. This usually is non-convex problem. However, replacing x_i by $\exp(y_i)$ we can write in new coordinates:

$$f_i(y) = \sum c_{i,\alpha} \exp(\langle \alpha, y\rangle)$$

$$g_i(y) = d_i \exp(\langle \beta_i, y\rangle).$$

Taking logarithms we get new problem: minimize $\log(f_0)$ under constraints

$$\log(f_i) \le 0,$$

and $\langle \beta_i, y\rangle + \log(d_i) = 0$. One can check that $\log(f_i)$ is convex, so this is convex problem.

Example: In IBM Model 1 we are given set of pairs of sentences in native language N and foreign language F. Sentence F is assumed to be good translation of sentence N. For technical reasons we add a fictional empty word at start of N. We assume that a word from F may be translated from any word in N, with probability that depends on word in N, but does not depend on position. We want to estimate probabilities $P(f|n)$ where f is foreign word and n is native word. Our assumptions lead to formula

$$P(F|N) = \frac{\epsilon}{(1 + l_N)^{l_F}} \prod_{j=1}^{l_F} \sum_{i=0}^{l_N} P(F_j|N_i).$$

where l_F is length of F, l_N is length of N and ϵ is a normalizing parameter.

In IBM Model 1 we maximize

$$\prod_{(F,N)\in T} P(F|N)$$

where T is set of pairs used for training.

Passing to logarithms, we maximize

$$L = \sum_{(F,N)\in T} \log(P(F|N)).$$

We have

$$P(F|N) = c_{F,N} \prod_{j=1}^{l_F} \sum_{i=0}^{l_N} P(F_j|N_i)$$

which gives

$$\log(P(F|N)) = d_{F_N} + \sum_{j=1}^{l_F} \log(\sum_{i=0}^{l_N} P(F_j|N_i))$$

and

$$L = c + \sum_{(F,N)\in T} \sum_{j=1}^{l_F} \log(\sum_{i=0}^{l_N} P(F_j|N_i)).$$

Since log is strictly concave this is equivalent to minimizing convex function $-L$. Note: in older literature there is wrong claim that this is strictly convex problem and solution is unique.

Maximizing f in general is different (non convex) problem.

Many important problems, in particular problems appearing in training neural nets are non convex. Still, methods developed for convex problems frequently work (but there is no warranty).

1.2 Unconstrained optimization, optimality conditions

Consider case when there is no side conditions, that is feasible set is whole $\mathbb{R}^n$. Recall multivariate calculus:

Lemma 1.1 *Assume that x_0 is interior point of feasible set. If f attains local minimum at x_0 and f is differentiable at x_0, then $f'(x_0) = 0$. If f is twice differentiable at x_0, then $f''(x_0)$ is positive definite. If f is convex and $f'(x_0) = 0$, then x_0 is global minimum of f.*

Remark: Since we allow ∞ as a value, set of points where f is finite may be smaller than whole $\mathbb{R}^n$. So while we have no explicit constraints, we obtain equivalent effect by making f infinite where it would be otherwise undefined. In other words, our results will be applicable also to some problems with constraints. In particular the lemma above works when feasible set is convex and open.

1.3 Quadratic problem

Unconstrained linear problem is either trivial (that is f is constant) or unbounded. So the simplest nontrivial example is quadratic.

Consider minimization of quadratic function

$$f(x) = \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle + c.$$

If x_0 is an optimal solution, then

$$0 = \nabla f(x_0) = Ax_0 + b$$

so

$$Ax_0 = -b.$$

If A is strictly positive definite, then A is invertible and

$$x_0 = -A^{-1}b$$

is an optimal solution.

If A is only weakly positive definite, then any solution to $Ax_0 = -b$ is optimal. If $Ax_0 = -b$ has no solution or A is not positive definite, then problem is unbounded from below and there is no optimal solution. In all cases solution reduces to numerical linear algebra: equation solving and possibly checking if A is positive definite.

However, for large problems exactly solving linear equations may be too expensive. In fact, optimization methods lead to one widely used method for approximate solving of linear equations (conjugate gradient method).

1.4 Example: least squares regression

We are given approximate values y_i, $i = 1, \dots, m$ of an unknown function at points x_i respectively. We want to express our function as a linear combination of known functions ϕ_j, $j = 0, \dots, r$:

$$f(x) = \sum_{j=0}^{r} \beta_j \phi_j.$$

In experimental setup usually there is some error, so we want to minimize sum of squares of errors:

$$\sum_{i=1}^{m} \|y_i - f(x_i)\|^2 = \sum_{i=1}^{m} \left\langle y_i - \sum_{j=0}^{r} \beta_j \phi_j(x_i), y_i - \sum_{l=0}^{r} \beta_l \phi_l(x_i) \right\rangle$$

$$= \sum_{j=0}^{r} \sum_{l=0}^{r} \left(\langle \sum_{i=1}^{m} \langle \phi_j(x_i), \phi_l(x_i) \rangle \right) \beta_j \beta_l +$$

$$-2\sum_{j=0}^{r}\sum_{i=1}^{m}\langle y_i,\phi_j(x_i)\rangle\beta_j+\sum_{i=1}^{m}\langle y_i,y_i\rangle$$
$$=\langle A\beta,\beta\rangle+\langle b,\beta\rangle+c$$

where

$$A_{j,l}=\sum_{i=1}^{m}\langle\phi_j(x_i),\phi_l(x_i)\rangle,$$
$$b_j=-2\sum_{i=1}^{m}\langle y_i,\phi_j(x_i)\rangle$$
$$c=\sum_{i=1}^{m}\langle y_i,y_i\rangle$$

In particular, when $\phi_0 = 1$ and $\phi_j(x) = x_j$ we have linear least squares regression.

Variation: we may add penalty for large β, that is minimize

$$\langle A\beta,\beta\rangle+\langle b,\beta\rangle+c+\lambda\langle\beta,\beta\rangle=\langle\tilde{A}\beta,\beta\rangle+\langle b,\beta\rangle+c.$$

which only differs that we have now different matrix $\tilde{A}$. When penalty term omits β_0 this is called ridge regression.

L^1 penalty gives non-quadratic problem (LASSO).

1.5 Quadratic approximation

Recall Taylor theorem in integral form:

$$f(x)=f(x_0)+f'(x_0)(x-x_0)+\int_0^1 tf''(x_0+t(x-x_0))(x-x_0,x-x_0)dt.$$

This may be obtained by integrating by parts

$$f(x)=f(x_0)+\int_0^1 f'(x+t(x-x_0))(x-x_0)dt.$$

If f has minimum at x_0, then $f''(x_0)$ is positive definite. If $f''(x_0)$ is strictly positive definite and f is regular then there is some neighbourhood V of x_0 such that f is convex in V. So convex methods are useful for local convergence.

1.6 Descent, basic idea

Typical optimization methods are iterative. Large class of methods can be described as below.

Iteratively form points x_i, starting from some x_0. Put

$$x_{i+1}=x_i+\alpha_i h_i$$

where h_i is called *search direction* and $\alpha_i > 0$ is called *step size* (or learning rate in machine learning context). We want this to be descent method, that is

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

except when x_i is optimal. In convex case we have

$$f(x_{i+1}) \geq f(x_i) + \alpha_i \langle \nabla f(x_i), h_i \rangle$$

so we must have $\langle \nabla f(x_i), h_i \rangle < 0$.

In general $\langle \nabla f(x_i), h_i \rangle < 0$ means that for α small enough we get descent. We call such h_i *descent direction*.

Expression

$$\frac{\langle \nabla f(x_i), h_i \rangle}{\|h_i\|}$$

measures how fast f decays in direction h_i. It is natural to choose h_i so that it gives fastest decay. By property of scalar product this means

$$h_i = \frac{-\nabla f(x_i)}{\|\nabla f(x_i)\|}$$

which is called *steepest descent* direction. Descent methods using multiples of $-\nabla f(x_i)$ are called steepest descent.

After choice of h_i we still need to choose α_i. This is called line search. Exact line search means that we solve one dimensional problem of minimizing $f(x_i + \alpha h_i)$ exactly. Sometimes this can be done easily, but in most cases exact line search is too expensive and we use an approximate one.

Choice of α_i requires some care. We need sufficient descent, otherwise descent may converge to non-optimal point. Also, step size must be big enough.

Below we will present theoretically good method to choose α_i. However, in many practical problem fixed α_i works well (actual value is frequently determined in experimental way).

1.7 Descent, Armijo's condition

One rule is to accept only steps giving sufficient percentage of decay expected from derivative (sufficient decay):

$$f(x_i + \alpha h_i) - f(x_i) \leq \rho \alpha \langle \nabla f(x_i), h_i \rangle$$

where $\rho \in (0, 1)$ is a fixed parameter. To avoid too small steps we require that multiplying step by fixed $\eta > 1$ should give unacceptable step:

$$f(x_i + \eta \alpha h_i) - f(x_i) > \rho \eta \alpha \langle \nabla f(x_i), h_i \rangle.$$

For convex functions, when α gives sufficient decay, then any smaller value also gives sufficient decay.

1.8 Descent, backtracking line search

This leads to simple algorithm:

- Choose some initial step size.
- If step size is too large, then while step size is too large keep dividing it by η, return first acceptable value.
- If step size is acceptable, then while step size is acceptable keep multiplying it by η, return last acceptable value.

called *backtracking line search*

Example: Put $f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2)$ and $x_0 = (\gamma, 1)$ where $\gamma > 0$. Then, using exact line search in i-th iteration we get

$$(\gamma \left(-\frac{1-\gamma}{1+\gamma}\right)^i, \left(\frac{1-\gamma}{1+\gamma}\right)^i).$$

that is iterates converge geometrically with rate

$$\frac{1-\gamma}{1+\gamma}$$

so for small γ convergence is very slow.

Remark: Using backtracking line search we can expect slightly slower convergence.

1.9 Descent, condition number

For good behaviour need extra assumptions, namely that

$$m\|h\|^2 \leq f(x)''(h,h) \leq M\|h\|^2$$

When m is biggest possible and M smallest possible quotient $\frac{m}{M}$ is called condition number. Using exact or backtracking line search can prove linear convergence with rate $1 - c\frac{m}{M}$. Example above shows that estimate of rate of convergence can not be essentially improved.

1.10 Unconstrained optimization, convergence

Lemma 1.2 *Assume f has continuous derivative. Let x_n be a sequence produced by gradient descent with exact or backtracking (Armijo) line search. Then every limit point x_∞ of $\{x_n\}$ is a stationary point, that is $\nabla f(x_\infty) = 0$.*

Idea of the proof: $f(x_n)$ is nonincreasing, so it converges to $f(x_\infty)$. By contradiction, if x_∞ was nonstationary, then gradient descent would decrease value of f by a fixed amount for all y in a neighbourhood of x_∞. This gives contradiction with convergence. $\square$

Remark: With fixed α_i there is no warranty of descent. However, if fixed α_i gives descent, then the argument above works and proves that every limit point is a stationary point.

Remarks:

- If sublevel set $\{x : f(x) \leq f(x_0)\}$ is compact, then there exist limit point, otherwise gradient descent may diverge.
- There may be multiple limit points.
- Numerically gradient descent typically converges to local minimum, but theoretically can converge to a saddle point (due to descent can not converge to maximum).
- No claim about rate of convergence.
- The same argument works whenever descent is uniform in some neighbourhood of x_∞.

Example: Let $f(x_1, x_2) = \frac{1}{2}(x_1^2 + \frac{1}{2}x_2^2)$ for $x_2 \geq 0$ and $f(x_1, x_2) = \frac{1}{2}(x_1^2 - \frac{1}{2}x_2^2)$ for $x_2 < 0$. It is easy to check that f has Lipschitz continuous derivative: $\|\nabla f(x) - \nabla(y)\| \leq \|x - y\|$. We saw that started from $(\frac{1}{2}, 1)$ gradient descent with exact line search will keep $x_2 > 0$, so it will converge to non-optimal stationary point $(0, 0)$. Here by exact we mean line search that will find local minimum closest to starting point. In the example above f is unbounded from below on lines used in line search, but it is possible to construct function with Lipschitz continuous derivative such that it agrees with quadratic $\frac{1}{2}(x_1^2 + \frac{1}{2}x_2^2)$ on all lines used in line search and $(0, 0)$ is not a local minimum.

1.11 Unconstrained optimization, rate of convergence

To say anything about rate of convergence we need extra assumptions. We already had assumption that

$$f''(x)(h, h) \leq M\|h\|^2.$$

For nonconvex f we need symmetric inequality

$$-M\|h\|^2 \leq f''(x)(h, h).$$

As long as f is smooth and domain of f is convex this is equivalent to

$$\|\nabla f(y) - \nabla f(x)\| \leq M\|y - x\|,$$

that is Lipschitz continuity of derivative of f.

Comparing f with quadratic function we get inequality

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

Taking $y = x - \alpha\nabla f(x)$, $\alpha = \frac{1}{M}$ we get

$$f(y) \le f(x) - \frac{1}{2M}\|\nabla f(x)\|^2.$$

Moreover, for $\phi(\alpha) = f(x_i - \alpha\nabla f(x_i))$ when $0 < \alpha < \frac{1}{M}$, then $\phi(\alpha)' < 0$. Namely,

$$\phi(\alpha)' = -\langle \nabla f(x_i - \alpha\nabla f(x_i)), \nabla f(x_i)\rangle.$$

We write

$$\nabla f(x_i - \alpha\nabla f(x_i)) = (\nabla f(x_i - \alpha\nabla f(x_i)) - \nabla f(x_i)) + \nabla f(x_i).$$

so

$$\begin{aligned} &-\langle \nabla f(x_i - \alpha\nabla f(x_i)), \nabla f(x_i)\rangle \\ &= -\langle \nabla f(x_i - \alpha\nabla f(x_i)) - \nabla f(x_i), \nabla f(x_i)\rangle - \langle \nabla f(x_i), \nabla f(x_i)\rangle \\ &\le -\|\nabla f(x_i)\|^2 + \|\nabla f(x_i - \alpha\nabla f(x_i)) - \nabla f(x_i)\|\|\nabla f(x_i)\|. \end{aligned}$$

Consequently, if

$$\|\nabla f(x_i - \alpha\nabla f(x_i)) - \nabla f(x_i)\| < \|\nabla f(x_i)\|$$

then $\phi(\alpha)' < 0$. By Lipschitz continuity of ∇f we have

$$\|\nabla f(x_i - \alpha\nabla f(x_i)) - \nabla f(x_i))\| \le M\|\alpha\nabla f(x_i)\|$$

so it is enough to have

$$M\|\alpha\nabla f(x_i)\| < \|\nabla f(x_i)\|$$

which means $|\alpha| < \frac{1}{M}$.

Now, given x_i we see that exact line search will choose x_{i+1} so that

$$f(x_{i+1}) \le f(x_i) - \frac{1}{2M}\|\nabla f(x_i)\|^2.$$

That is

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

hence adding over i:

$$f(x_0) - f(x_{m+1}) \ge \frac{1}{2M}\sum_{i=0}^{m}\|\nabla f(x_i)\|^2$$

which means that there $i \le m$ such that

$$\|\nabla f(x_i)\|^2 \le \frac{2M(f(x_0) - f(x_{m+1}))}{m+1}.$$

If problem is bounded from below this means that we can decrease gradient to ε in $O(\frac{1}{\varepsilon^2})$ steps.

Above we handled fixed α and exact line search. Recall Armijo's rule:

$$f(x_i + \alpha h_i) - f(x_i) \le \rho\alpha\langle\nabla f(x_i), h_i\rangle.$$

By previous calculation $\alpha \le \frac{1}{M}$ is acceptable when $\rho \le \frac{1}{2}$. Together with second part of the rule, this means that we will choose $\alpha > \frac{1}{\eta M}$. If we choose $\alpha > \frac{1}{M}$, then we get at least $\frac{\rho}{2M}\|\nabla f(x_i)\|^2$ of decay, otherwise at least $\frac{1}{2\eta M}\|\nabla f(x_i)\|^2$ of decay. Then, proceeding as before we get

$$\|f'(x_i)\|^2 \le \frac{CM(f(x_0) - f(x_{m+1}))}{m+1}.$$

with $C = 2\max(\frac{1}{\rho}, \eta)$.

More generally, if h_i is arbitrary search direction such that $\langle\nabla f(x_i), h_i\rangle < 0$, then write

$$\cos(\theta_i) = -\frac{\langle\nabla f(x_i), h_i\rangle}{\|\nabla f(x_i)\|\|h_i\|}.$$

Repeating previous reasoning we get Zoutendijk inequality

$$\sum_{i=0}^{m}\cos(\theta_i)\|\nabla f(x_i)\|^2 \le \frac{CM(f(x_0) - f(x_{m+1}))}{m+1}$$

with similar conclusions as before.

$O(\frac{1}{\varepsilon^2})$ steps to decrease gradient to ε may look bad, but in fact is best possible estimate for gradient descent and several other methods. There is better method needing $O(\varepsilon^{\frac{-3}{2}})$ steps and this in general is best possible.

For convex functions situation is better. To simplify arguments we will consider constant step size $\alpha \le \frac{2}{M}$ where as before M is Lipschitz constant of gradient of f.

To prove the results below we need a useful estimate:

Lemma 1.3

$$f(x) + \langle\nabla f(x), y - x\rangle + \frac{2}{M}\|\nabla f(x) - \nabla f(y)\|^2 \le f(y),$$

$$\frac{1}{M}\|\nabla f(y) - \nabla f(x)\|^2 \le \langle\nabla f(y) - \nabla f(x), y - x\rangle$$

Proof: To prove first inequality consider $\phi(y) = f(y) - \langle\nabla f(x), y - x\rangle$. $\nabla\phi(x) = 0$ so ϕ attains minimal value at x, so

$$f(x) = \phi(x) \le \phi(y - \frac{1}{M}\nabla\phi(y)).$$

Since gradient of ϕ has the same Lipschitz constant as f we have descent estimate

$$\phi(y - \frac{1}{M}\nabla\phi(y)) \le \phi(y) - \frac{1}{2M}\|\nabla\phi(y)\|^2.$$

Since $\nabla\phi(y) = \nabla f(y) - \nabla f(x)$ this gives first estimate.

Adding first estimate for x and y and with reversed order gives second estimate. □

Now, we will prove that under conditions above distance to optimal point can not increase.

Lemma 1.4 *With assumptions as above*

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

Proof.

$$\begin{aligned}
\|x_{i+1} - x_\infty\|^2 &= \|x_i - x_\infty - \alpha\nabla f(x_i)\|^2 \\
&= \|x_i - x_\infty\|^2 - 2\alpha\langle\nabla f(x_i), x_i - x_\infty\rangle + \alpha^2\|\nabla f(x_i)\|^2 \\
&\le \|x_i - x_\infty\|^2 - \alpha\frac{2}{M}\|\nabla f(x_i)\|^2 + \alpha^2\|\nabla f(x_i)\|^2 \\
&\le \|x_i - x_\infty\|^2
\end{aligned}$$

as long as $\alpha \le \frac{2}{M}$. □

Now, we can prove decay of goal function:

Lemma 1.5 *Let f be convex such that gradient of f is Lipschitz continuous with constant M. Gradient descent using constant step size $\alpha = \frac{1}{M}$ satisfies*

$$f(x_m) - f(x_\infty) \le \frac{2M\|x_0 - x_\infty\|^2}{m+4}$$

where x_∞ is a minimizer of f.

Proof. By convexity

$$f(x_\infty) - f(x_i) \ge \langle\nabla f(x_i), x_\infty - x_i\rangle \ge -\|\nabla f(x_i)\|\|x_\infty - x_i\|$$

so

$$\|\nabla f(x_i)\| \ge \frac{f(x_i) - f(x_\infty)}{\|x_i - x_\infty\|} \ge \frac{f(x_i) - f(x_\infty)}{\|x_0 - x_\infty\|}$$

Combining this with descent estimate we get

$$f(x_{i+1}) - f(x_i) \le -\frac{(f(x_i) - f(x_\infty))^2}{2M\|x_0 - x_\infty\|^2}$$

Writing $\delta_i = f(x_i) - f(x_\infty)$ this means

$$\delta_{i+1} - \delta_i \le -\frac{\delta_i^2}{2M\|x_0 - x_\infty\|^2}$$

so

$$\delta_{i+1} \le \delta_i - \frac{\delta_i^2}{2M\|x_0 - x_\infty\|^2}$$

Now the we get result by induction. It is clearly true for $i = 0$. The right hand side is a quadratic in δ_i, which attains maximal value at $\delta_{max} = M\|x_0 - x_\infty\|^2$. By inductive assumption $\delta_i \le \frac{2M\|x_0 - x_\infty\|^2}{i+4}$, which is smaller than δ_{max}, so our quadratic is increasing for relevant δ_i and we get estimate from above plugging in upper estimate for δ_i.

Consequently

$$\delta_{i+1} \le \frac{2M\|x_0 - x_\infty\|^2}{i+4} - \frac{2M\|x_0 - x_\infty\|^2}{(i+4)^2}$$

$$= 2M\|x_0 - x_\infty\|^2 \left(\frac{1}{i+4} - \frac{1}{(i+4)^2}\right)$$

$$\le 2M\|x_0 - x_\infty\|^2 \left(\frac{1}{i+4} - \frac{1}{(i+4)(i+5)}\right)$$

$$= \frac{2M\|x_0 - x_\infty\|^2}{i+5}.$$

□

Remark. We get similar result for smaller steps. However, when $\alpha > \frac{2}{M}$ our proof that $\|x_i - x_\infty\|$ is nonincreasing no longer works. For $\frac{1}{M} < \alpha \le \frac{2}{M}$ we get worse estimate of descent. In practice larger steps are likely to give faster convergence, but theory suggests small steps. So there is a discrepancy. We will see that similar discrepancy appears in different situations.

We also have decay of gradient:

Lemma 1.6 *With assumptions as above we have*

$$\min_{m/2 \le i < m} \|\nabla f(x_i)\| \le \frac{4M\|x_0 - x_\infty\|}{m+1}.$$

Remark: Clearly, this is better then non-convex gradient estimate.

Proof: Applying our previous general gradient estimate with x_0 replaced by x_{m_0} where $m_0 = m/2$ (rounded up) we get

$$\min_{m_0 \le i < m} \|\nabla f(x_i)\|^2 \le \frac{2M(f(x_{m_0}) - f(x_\infty))}{m - m_0 + 1}.$$

Using estimate for $f(x_{m_0}) - f(x_\infty)$ this is

$$\le \frac{2M(2M\|x_0 - x_\infty\|^2)}{(m_0+4)(m - m_0 + 1)}$$

$$\leq \frac{(4M\|x_0 - x_\infty\|)^2}{(m+1)^2}$$

which gives the claim. □

1.12 Further reading

David G. Luenberger, Yinyu Ye, Linear and Nonlinear Programming, chapters 7 and 8.

Stephen Boyd, Lieven Vandenberghe, Convex Optimization, chapter 9.