Optimization

6. Convex optimization: algorithms

Andrew Lesniewski

Baruch College
New York

Fall 2018

Outline

1. Unconstrained problems

2. Equality constrained problems

3. Inequality constrained problems

Unconstrained convex problems

- We consider first an unconstrained convex optimization problem:

$$\min\ f(x), \tag{1}$$

where $f(x)$ is twice continuously differentiable. By x^* we denote its solution, and by $f^* = f(x^*)$ its optimal value.

- The necessary and sufficient condition for x^* is

$$\nabla f(x^*) = 0. \tag{2}$$

- As in the case of general nonlinear optimization problems, the solution methods are iterative, and start with an initial guess x_0 such that
  (i) $x_0 \in \mathrm{dom}(f)$,
  (ii) the *sublevel set* $S = \{x \in \mathrm{dom}(f) :\ f(x) \leq f(x_0)\}$ is closed.
- The second condition is usually hard to verify. Cases when it is true include:
  (i) if $\mathrm{dom}(f) = \mathbb{R}^n$,
  (ii) if $f(x) \to \infty$, as x approaches the boundary of $\mathrm{dom}(f)$.

Unconstrained strong convex problems

- A function $f(x)$ is *strongly convex* on C, if there exists $\mu > 0$ such that

$$\nabla^2 f(x) \geq \mu I, \quad \text{for all } x \in C. \tag{3}$$

- An implication of strong convexity is:

$$f(y) \geq f(x) + \nabla f(x)^{\mathrm{T}}(y-x) + \frac{1}{2}\,\mu\|y-x\|^2, \quad \text{for all } x, y \in C. \tag{4}$$

- Indeed, from Taylor's theorem, there is a z on the line segment connecting x and y such that

$$f(y) = f(x) + \nabla f(x)^{\mathrm{T}}(y-x) + \frac{1}{2}\,(y-x)^{\mathrm{T}}\nabla^2 f(z)(y-x),$$

which, together with (3), implies (4).

Unconstrained strong convex problems

- We infer from (4) that
$$f^* \geq f(x) - \frac{1}{2\mu} \|\nabla f(x)\|^2. \tag{5}$$
- To see this, we note that the RHS of (4) has a minimum in y for $\bar{y} = x - \nabla f(x)/\mu$, and the minimum value is equal to $f(x) - \|\nabla f(x)\|^2/2\mu$.
- This inequality allows us to formulate the following exit criterion for the search: in order to be within ε from f^*, $f^* - f(x) \leq \varepsilon$, we terminate the search when
$$\|\nabla f(x)\| \leq \sqrt{2\mu\varepsilon}.$$

Descent methods

- The *descent method* consists in constructing a sequence

$$x_{k+1} = x_k + t_k \Delta x_k , \tag{6}$$

where the search direction $\Delta x_k \in \mathbb{R}^n$ and step size $t_k > 0$ are chosen so that

$$f(x_{k+1}) < f(x_k) \tag{7}$$

- From convexity, Δx_k must satisfy

$$\nabla f(x_k)^{\mathrm{T}} \Delta x_k < 0, \tag{8}$$

i.e. the angle between $\nabla f(x_k)$ and Δx_k is acute.
- General descent method: Choose an initial guess x_0 and iterate the following steps:

```
while( exit criterion not satisfied )
    determine a descent direction $\Delta x_k$
    choose a step size $t_k$
    update $x_{k+1} = x_k + t_k \Delta x_k$
```

Backtracking line search

- The second step of the algorithm, the *line search*, determines where on the ray

$$\{x + t\Delta x : \; t > 0\} \tag{9}$$

  the next iterate will be.
- We choose t to minimize the objective function along the ray (9):

$$t = \arg\min_{s>0} f(x + s\Delta x). \tag{10}$$

- It is usually sufficient to solve this problem approximately. One inexact line search method that is very simple and quite effective is the *backtracking line search*. It depends on two constants $\alpha \in (0, 1/2)$ and $\beta \in (0, 1)$.
- Given a descent direction Δx and $t_0 = 1$, iterate the following steps:

$$\begin{aligned} &\text{while } f\big(x + t_k\Delta x\big) \geq f(x) + \alpha t_k \nabla f(x)^{\mathrm{T}} \Delta x \\ &t_{k+1} = \beta t_k \end{aligned}$$

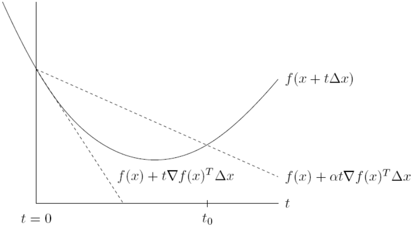
- The line search is called backtracking because it starts with unit step size and then reduces it by the factor β until the exit condition holds.

Backtracking line search

- Since Δx is a descent direction, we have $\nabla f(x)^{\mathrm{T}}\Delta x < 0$, and so

$$\begin{aligned} f(x + t\Delta x) &\approx f(x) + \nabla f(x)^{\mathrm{T}}\Delta x \\ &< f(x) + \alpha t \nabla f(x)^{\mathrm{T}}\Delta x, \end{aligned}$$

  which shows that the backtracking line search eventually terminates. The constant α can be interpreted as the fraction of the decrease in $f(x)$ predicted by linear extrapolation that we will accept.
- In the figure below, the lower dashed line shows the linear extrapolation of $f(x)$, and the upper dashed line has a slope a factor of α smaller. The backtracking condition is that $f(x)$ lies below the upper dashed line, i.e., $0 < t < t_0$.


$f(x + t\Delta x)$
$f(x) + t\nabla f(x)^T\Delta x$
$f(x) + \alpha t\nabla f(x)^T\Delta x$
t
$t = 0$
t_0

Gradient descent methods

- In the gradient descent method we choose $\Delta x = -\nabla f(x)$.
- Choose an initial guess $x_0 \in \mathrm{dom}(f)$ and iterate:

```
while( exit criterion not satisfied )
$\Delta x_k = -\nabla f(x_k)$
choose step size $t_k$ via exact or backtracking line search
update $x_{k+1} = x_k + t_k \Delta x_k$
```

- As already discussed in Lecture Notes #1, this method tends to be slow.

Steepest descent method

- The first order Taylor expansion of $f(x)$ is

$$f(x+\xi) \approx f(x) + \nabla f(x)^{\mathrm{T}} \xi .$$

  From calculus, $\nabla f(x)^{\mathrm{T}} \xi$ is the directional derivative of $f(x)$ in the direction of the vector ξ.
- We choose Δx to point in the direction of ξ, but we have to bound the magnitude of ξ.
- To this end, we choose a norm $\|x\|$ on $\mathbb{R}^n$ (for example, Euclidean), and define the *normalized steepest descent direction* as:

$$\Delta x_{\mathrm{nsd}} = \arg\min\{\nabla f(x)^{\mathrm{T}} \xi : \ \|\xi\| = 1\}. \tag{11}$$

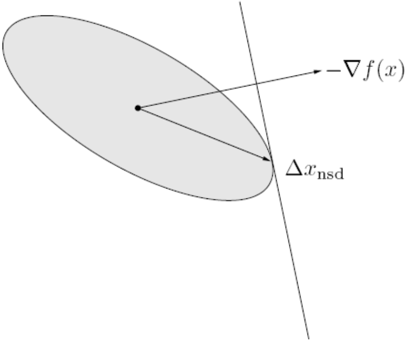
- For example, if the norm $\|\xi\|$ is the Euclidean norm, then $\Delta x_{\mathrm{nsd}} = -\nabla f(x)$, and the method reduces to the gradient descent.

Steepest descent method: quadratic norm

- Given a positive definite matrix H, we define the quadratic H-weighted norm on $\mathbb{R}^n$ by
$$\|\xi\|_H = \sqrt{\xi^{\mathrm{T}} H \xi}. \tag{12}$$
- The normalized steepest descent direction is given by
$$\Delta x_{\mathrm{nsd}} = -H^{-1} \nabla f(x). \tag{13}$$

Steepest descent method: quadratic norm

- The ellipsoid shown in the figure below is the unit ball of the norm, translated to the point x. The normalized steepest descent direction Δx_{nsd} at x extends as far as possible in the direction $\nabla f(x)$ while staying in the ellipsoid.



Steepest descent method: L^1-norm

- We consider the steepest descent method for the L^1-norm. A normalized steepest direction,

$$\Delta x_{\mathrm{nsd}} = \arg\min\{\nabla f(x)^{\mathrm{T}}\xi : \ \|\xi\|_1 = 1\}, \tag{14}$$

  can be characterized as follows.

- Let i be an index for which $\|\nabla f(x)\|_\infty = |\partial f(x)/\partial x_i|$. Then

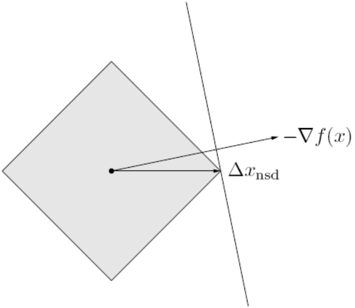
$$\Delta x_{\mathrm{nsd}} = -\mathrm{sign}\Big(\frac{\partial f(x)}{\partial x_i}\Big)\, e_i, \tag{15}$$

  where e_i is the i-th standard basis vector.

- Thus, the normalized steepest descent step in L^1-norm can always be chosen to be a standard basis vector (or its negative). It is the coordinate axis direction along which the approximate decrease in $f(x)$ is greatest.

Steepest descent method: L^1 norm

- The diamond is the unit ball of the L^1-norm, translated to the point x. The normalized steepest descent direction can always be chosen in the direction of a standard basis vector. In the figure below, we have $\Delta x_{\mathrm{nsd}} = e_1$.



- The steepest descent method in the L^1-norm has a natural interpretation. At each iteration we select a component of $\nabla f(x)$ with maximum absolute value, and decrease or increase the component x_i, according to the sign of $\big(\nabla f(x)\big)_i$.

Newton's methods

- For $x \in \mathrm{dom}(f)$, the step

$$\Delta x_{\mathrm{nt}} = -\nabla^2 f(x)^{-1} \nabla f(x) \tag{16}$$

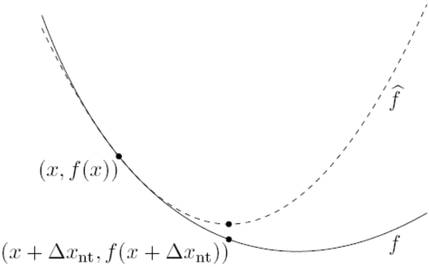
  is called the Newton step.
- The second order Taylor approximation to $f(x)$ is

$$\hat{f}(x+\xi) = f(x) + \nabla f(x)^{\mathrm{T}} \xi + \frac{1}{2} \xi^{\mathrm{T}} \nabla^2 f(x) \xi . \tag{17}$$

- This is a quadratic function in ξ, and $\xi = \Delta x_{\mathrm{nt}}$ is its minimizer!
- Since $f(x)$ is twice continuously differentiable, this *quadratic model* should be accurate for x near x^*.

Newton's method

- The figure below shows the function $f(x)$ (solid line) and its second order approximation $\hat{f}(x+\xi)$ (dashed). The Newton step Δx_{nt} is what must be added to x to give the minimizer of $\hat{f}(x+\xi)$.

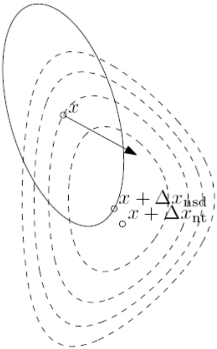


Newton's method

- We can get another insight into the workings of Newton's method by interpreting it in terms of the quadratic norm introduced in (12).
- Namely, the direction of the Newton step at x is the steepest descent direction for the norm (12) with $H = \nabla^2 f(x)$.

Newton's method

- The dashed lines are level curves of a convex function. The ellipsoid (solid line) is $\{x + \xi : \xi^{\mathrm{T}} \nabla^2 f(x) \xi \leq 1\}$. The arrow shows $-\nabla f(x)$, the gradient descent direction. The Newton step Δx_{nt} is the steepest descent direction in the norm (12). The figure also shows Δx_{nsd}, the normalized steepest descent direction for the same norm.

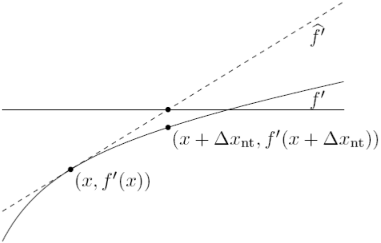


Newton's method

- The first order condition $\nabla f(x^*) = 0$ for x near x^* reads:

$$\nabla f(x) + \nabla^2 f(x)\xi \approx 0, \tag{18}$$

  and its solution is $\xi = \Delta x_{\mathrm{nt}}$.
- Thus the Newton step is what must be added to x so that the linearized optimality condition holds. This is illustrated the figure below.
- The solid curve is the derivative $f'(x)$ of the function $f(x)$ shown in the previous figure, and $\hat{f}'(x)$ is the linear approximation of $f'(x)$. The Newton step Δx_{nt} is the difference between the root of $f'(x)$ and x.


$\hat{f}'$
f'
$(x + \Delta x_{\mathrm{nt}}, f'(x + \Delta x_{\mathrm{nt}}))$
$(x, f'(x))$

Newton's method

- The quantity
$$\lambda(x)=\sqrt{\nabla f(x)^{\mathrm{T}}\nabla^2 f(x)^{-1}\nabla f(x)} \tag{19}$$
is called the *Newton decrement* at x.
- This is an important concept, because
$$\begin{aligned} f(x)-\min_{\xi}\hat{f}(x+\xi) &= f(x)-\hat{f}(x+\Delta x_{\mathrm{nt}})\\ &=\frac{1}{2}\,\lambda(x)^2, \end{aligned}$$
and so it is a measure of distance between $f(x)$ and the minimum of its quadratic model.
- It can thus be used as an estimate of $f(x)-f^*$.

Newton's method

- We can summarize Newton's method in the following way.
- Choose an initial guess $x_0 \in \mathrm{dom}(f)$ and iterate the following steps:

```
while( exit criterion not satisfied )
  compute the Newton step $\Delta x_k$ and decrement $\lambda(x_k)$
  exit if $\lambda(x_k) < \varepsilon$
  choose a step size $t_k$ by exact or backtracking line search
  update $x_{k+1} = x_k + t_k \Delta x_k$
```

Equality constrained problems: KKT conditions

- Consider now an equality constrained convex optimization problem:

$$\min\ f(x), \quad \text{subject to } Ax = b, \tag{20}$$

  where $f(x)$ is twice continuously differentiable. Without loss of generality we assume that $p = \mathrm{rank}(A) < n$ (so that the constraints are independent).

- If x^* is a solution to (20), the first oder conditions read:

$$\begin{aligned} \nabla f(x^*) + A^{\mathrm{T}}\lambda^* &= 0, \\ Ax^* &= b, \end{aligned} \tag{21}$$

  where λ^* is the vector of Lagrange multipliers.

- Solving (20) is equivalent to solving the system (21).

Equality constrained problems: KKT conditions

- For example, consider the quadratic problem:

$$\min \frac{1}{2} x^{\mathrm{T}} H x + q^{\mathrm{T}} x + r, \quad \text{subject to } Ax = b, \tag{22}$$

  where H is positive semidefinite.
- The KKT conditions (21) read

$$\begin{pmatrix} H & A^{\mathrm{T}} \\ A & 0 \end{pmatrix} \begin{pmatrix} x^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} -q \\ b \end{pmatrix}. \tag{23}$$

- This system may or may not have solutions depending on the problem. It is nonsingular, if the matrix $P + A^{\mathrm{T}} A$ is (strictly) positive definite.
- The system (23) is an example of a *KKT system*.

Newton's method

- Newton's method can be used to solve equity constraint convex problems! Key differences with unconstraint problems:
  (i) the initial guess x_0 has to be feasible, i.e. $Ax_0 = b$,
  (ii) the Newton step Δx_k has to satisfy the feasibility condition $A\,\Delta x_k = 0$.
- In order to find the appropriate step, we assume that x is feasible, and consider the second order Taylor approximation at x to the problem (20):

$$\min_{\xi}\ f(x) + \nabla f(x)^{\mathrm{T}}\xi + \frac{1}{2}\,\xi^{\mathrm{T}}\nabla^2 f(x)\xi, \quad \text{subject to } A\xi = 0. \tag{24}$$

- This is a quadratic problem (in ξ) of the form (22), and its solution reduces to solving the linear system:

$$\begin{pmatrix} \nabla^2 f(x) & A^{\mathrm{T}} \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla f(x) \\ 0 \end{pmatrix}. \tag{25}$$

Newton's method

- Newton's method for equality constraint convex problems can be formulated as follows.
- Choose an initial guess $x_0 \in \mathrm{dom}(f)$, such that $Ax_0 = b$ and iterate the following steps:

```
while( exit criterion not satisfied )
   compute the Newton step $\Delta x_k$ decrement $\lambda(x_k)$
   exit if $\lambda(x_k) < \varepsilon$
   choose the step size $t_k$ by the backtracking line search
```

- With some effort, it is possible to extend this algorithm to infeasible starting points x_0. A detailed presentation can be found in [1].

Solving KKT systems

- We now describe methods for solving the (linear) KKT systems that arise in the process of determining the Newton step. We write this system in the general form

$$\begin{pmatrix} H & A^{\mathrm{T}} \\ A & 0 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} = - \begin{pmatrix} q \\ p \end{pmatrix} . \tag{26}$$

- *Solving the full system.* This is the most straightforward approach. The KKT matrix is symmetric, but not necessarily positive definite, and so the preferred approach is the LDL^{T}-decomposition. This is a reasonable approach when the dimension of the problem is small.
- *Elimination.* Assuming that H is positive definite, we have

$$\begin{aligned} v &= H^{-1}(p + A^{\mathrm{T}} w), \\ Av &= -p. \end{aligned}$$

and so

$$w = (AH^{-1}A^{\mathrm{T}})^{-1}(p - AH^{-1}q).$$

Solving KKT systems

- This solution can be expressed in the block matrix form:

$$\begin{pmatrix} v \\ w \end{pmatrix} = -\begin{pmatrix} 0 & H^{-1}A^{\mathrm{T}}(AH^{-1}A^{\mathrm{T}})^{-1} \\ (AH^{-1}A^{\mathrm{T}})^{-1}AH^{-1} & -(AH^{-1}A^{\mathrm{T}})^{-1} \end{pmatrix} \begin{pmatrix} q \\ p \end{pmatrix} \tag{27}$$

  The square block matrix on the right hand side is a special case of the *Schur inversion formula* of the block matrix on the left hand side of (28). The negative definite matrix $S = -(AH^{-1}A^{\mathrm{T}})^{-1}$ is called the *Schur complement* of H.

- In some cases (e.g. diagonal H), the Schur complement can be calculated efficiently, in which case this method is faster (linear in n rather than cubic) than the LDL^{T}-decomposition method.

Solving KKT systems

- *Elimination with singular H.* Problem 2 of Assignment #4 shows that the KKT matrix is nonsingular if and only if $H + A^{\mathrm{T}} QA > 0$, for some $Q > 0$.
- If $H \geq 0$ is singular, we can always find a matrix $Q \geq 0$ so that $H + A^{\mathrm{T}} QA > 0$. Thus the KKT matrix with H replaced by $H + A^{\mathrm{T}} QA > 0$ is nonsingular.
- But the system (28) is equivalent to the system:

$$\begin{pmatrix} H + A^{\mathrm{T}} QA & A^{\mathrm{T}} \\ A & 0 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} = - \begin{pmatrix} q + A^{\mathrm{T}} Qp \\ p \end{pmatrix}, \tag{28}$$

which is nonsingular and can be solved by elimination!

Formulation of the problem

- Consider now a general convex optimization problem

$$\min f(x), \quad \text{subject to } \begin{cases} c_i(x) \leq 0, \; i = 1, \ldots, m, \\ Ax = b. \end{cases} \tag{29}$$

  with $c_i(x)$ convex and twice continuously differentiable, and $p = \operatorname{rank}(A) < n$.

- We assume that
  (i) the optimal solution x^* exists,
  (iii) Slater's condition holds.

  As a result, λ^* exists and, along with x^*, satisfies the KKT conditions.

- The goal is to solve an inequality constrained problem by means of *interior point methods*, which reduce it to a sequence of linear constrained problems. We had a glimpse of this method in Lecture Notes #2.

Formulation of the problem

- We start by reformulating the problem as a logarithmic barrier problem:

$$\min f(x) + \frac{1}{t}\,B(x), \quad \text{subject to } Ax = b, \tag{30}$$

where $B(x)$ is the barrier function:

$$B(x) = -\sum_{i=1}^{m} \log\big(-c_i(x)\big) \tag{31}$$

- Note that this objective function is convex and twice continuously differentiable on its domain:

$$\begin{aligned} \nabla B(x) &= -\sum_{i=1}^{m} \frac{1}{c_i(x)}\,\nabla c_i(x), \\ \nabla^2 B(x) &= \sum_{i=1}^{m} \frac{1}{c_i(x)^2}\,\nabla c_i(x)\nabla c_i(x)^{\mathrm{T}} - \sum_{i=1}^{m} \frac{1}{c_i(x)}\,\nabla^2 c_i(x). \end{aligned} \tag{32}$$

Formulation of the problem

- As a result, Newton's method should be applicable!
- For example, in case of an inequality constrained LP problem, we are led to the following approximate optimization problem:

$$\min c^{\mathrm{T}} x - \frac{1}{t} \sum_{i=1}^{m} \log \left(b_i - a_i^{\mathrm{T}} x\right), \quad \text{subject to } Ax = b. \tag{33}$$

Choosing t

- One might expect that choosing large t right away might be a good idea. As we already mentioned in Lecture Notes #2, this is not necessarily the case.
- Newton's method works well if the Hessian of the objective function is not too large (Taylor's expansion!).
- From the explicit expression (32) we see, however, that the Hessian explodes as x approaches the boundary of the feasible set.
- For this reason, we will consider a sequence of problems with gradually increasing t, where each of the problems starts with the solution of the previous problem (warm start).

Central path

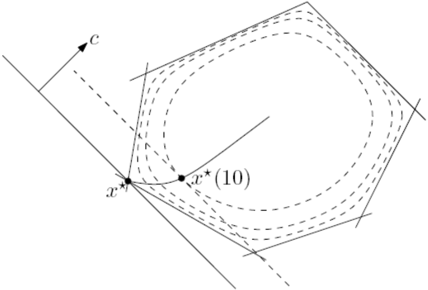
- For $t > 0$ we define the *central point* $x^*(t)$ as the solution to the optimization problem:
$$\min tf(x) + B(x), \quad \text{subject to } Ax = b, \tag{34}$$
and assume that it exists.
- The *central path* is the set $\{x^*(t),\ t > 0\}$ of central points.
- The central path exists, provided the following conditions are satisfied for all $t > 0$:
  (i) Strict feasibility:
$$\begin{aligned} c_i\big(x^*(t)\big) &< 0,\ \text{ for } i = 1\ldots, m,\\ Ax^*(t) &= b. \end{aligned} \tag{35}$$
  (ii) First order condition (*centrality condition*):
$$t\nabla f\big(x^*(t)\big) - \sum_{i=1}^{m} \frac{1}{c_i\big(x^*(t)\big)}\,\nabla c_i\big(x^*(t)\big) + A^{\mathrm{T}}\lambda = 0. \tag{36}$$

Central path

- In the LP example discussed above, the centrality condition reads:

$$tc + \sum_{i=1}^{m} \frac{1}{b_i - a_i^{\mathrm{T}} x}\, a_i + A^{\mathrm{T}} \lambda = 0. \tag{37}$$

- The figure below shows the central path for an LP with $n = 2$ and $m = 6$. The dashed curves show three contour lines of $B(x)$. The central path converges to x^* as $t \to \infty$. Also shown is the point on the central path with $t = 10$. The optimality condition (37) at this point can be verified geometrically. The line $c^{\mathrm{T}} x = c^{\mathrm{T}} x^*(10)$ is tangent to the contour line of $B(x)$ through $x^*(10)$.



Central path

- From the centrality condition (36) we can derive an important property of the central path: Every central point yields a dual feasible point, and hence a lower bound on the optimal value f^*.
- Specifically, define

$$\begin{aligned} \lambda_i^*(t) &= -\frac{1}{tc_i\big(x^*(t)\big)}\,, \text{ for } i = 1, \ldots, m, \\ \nu_i^*(t) &= \frac{1}{t}\,\lambda_{m+i}, \text{ for } i = 1, \ldots, p. \end{aligned} \tag{38}$$

- We claim that the pair $\lambda^*(t), \nu^*(t)$ is dual feasible, and so it yields a lower bound for the optimal value f^*.
- First of all, $\lambda_i^*(t) > 0$, since $c_i\big(x^*(t)\big) < 0$.

Central path

- Next, rewriting (36) as

$$\nabla f\big(x^*(t)\big) + \sum_{i=1}^{m} \lambda_i^*(t)\,\nabla c_i\big(x^*(t)\big) + A^{\mathrm{T}}\nu^*(t) = 0, \tag{39}$$

  we infer that $x^*(t)$ minimizes the Lagrange function:

$$L(x,\lambda,\nu) = f(x) + \sum_{i=1}^{m} \lambda_i c_i(x) + \nu^{\mathrm{T}}(Ax - b), \tag{40}$$

  for $\lambda_i = \lambda_i^*(t), \nu_i = \nu_i^*(t)$. This means that $\lambda^*(t), \nu^*(t)$ is indeed dual feasible.

- Let us now consider the dual Lagrange function $q\big(\lambda^*(t), \nu^*(t)\big)$:

$$\begin{aligned} q\big(\lambda^*(t), \nu^*(t)\big) &= f\big(x^*(t)\big) + \sum_{i=1}^{m} \lambda_i^*(t) c_i\big(x^*(t)\big) + \nu^*(t)^{\mathrm{T}}\big(Ax^*(t) - b\big) \\ &= f\big(x^*(t)\big) - \frac{m}{t}\,. \end{aligned}$$

Central path

- This identity says that the duality gap for $x^*(t)$ and $\big(\lambda^*(t), \nu^*(t)\big)$ is m/t.
- As $t \to \infty$, the duality gap goes to zero, i.e.

$$f\big(x^*(t)\big) - f^* \to 0. \tag{41}$$

- As a consequence, $x^*(t) \to x^*$, as $t \to \infty$.

The barrier method

- Algorithmically, the barrier method can be formulated as follows.
- Choose a strictly feasible initial guess $x_0 \in \mathrm{dom}(f)$, $t_0 > 0$, $\mu > 1$ and iterate the following steps:

```
while( exit criterion not satisfied )
    compute $x^*(t_k)$ by minimizing $t_k f(x) + B(x)$ subject to $Ax = b$
    exit if $m/t_k < \varepsilon$
    define $t_{k+1} = \mu t_k$, and choose $x^*(t_k)$ to be the initial guess
```

- The first step of the algorithm is called the *centering step* or an *outer iteration*.
- The Newton iterations or steps executed during the centering step are referred to as *inner iterations*. At each inner step, we have a primal feasible point; we have a dual feasible point, however, only at the end of each outer (centering) step.

The barrier method

- We can estimate the number of outer iterations after which the algorithm will stop.
- Since it starts at t_0, and it exits when

$$\frac{m}{t_0\mu^k} < \varepsilon\,,$$

the number of iterations is given by

$$k = \left\lceil \frac{\log\left(m/(\varepsilon t_0)\right)}{\log(\mu)} \right\rceil.$$

- While this analysis shows that the barrier method does converge (under reasonable assumptions), it does not address a basic question: As the parameter t increases, do the centering problems become more difficult? Numerical evidence suggests that for a wide variety of problems, this is not the case; the centering problems appear to require a nearly constant number of Newton steps to solve, even as t increases.

The barrier method

- Computing $x^*(t)$ exactly is not necessary, since the central path has no significance beyond the fact that it leads to a solution of the original problem in the limit $t \to \infty$.
- On the other hand, the cost of computing an extremely accurate minimizer of $tf(x) + B(x)$, as compared to the cost of computing a good minimizer, is only marginally more, i.e., a few Newton steps at most. For this reason it is not unreasonable to assume exact centering.

The barrier method

- The choice of the parameter involves a trade-off in the number of inner and outer iterations required.

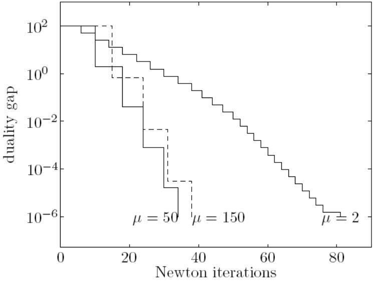
  (i) If μ is close to 1 then, at each outer iteration, t_k increases by a small factor. As a result, the initial guess $x(t_k)$ for the next Newton search is a very good starting point, and the number of Newton steps needed to compute the next iterate is small. However, we expect a large number of outer iterations, since each outer iteration reduces the duality gap by only a small amount. In this case the iterates closely follow the central path.

  (ii) If μ is large, after each outer iteration t_k increases by a large amount. Thus $x(t_k)$ may not be a very good guess for the next Newton search, and we expect many inner iterations. This results in fewer outer iterations, since the duality gap is reduced by the large factor μ at each outer iteration, but more inner iterations. With μ large, the iterates are widely separated on the central path.

- In practise, the two effects really offset each other. The total number of inner iterations are constant for sufficiently large μ. Values $10 \lesssim \mu \lesssim 20$ seem to work well.

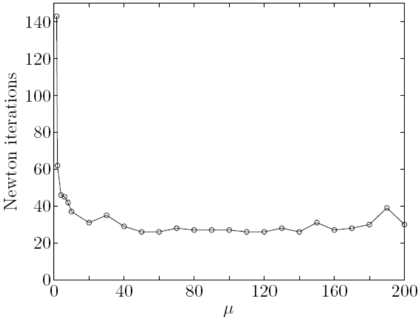
The barrier method

- The figure below shows the progress of barrier method for a small LP problem, showing duality gap versus cumulative number of Newton steps. Three plots are shown, corresponding to three values of the parameter μ: 2, 50, and 150. In each case, we have approximately linear convergence of duality gap.



The barrier method

- The figure below examines the trade-off in the choice of the parameter μ, for a small LP program. The vertical axis shows the total number of Newton steps required to reduce the duality gap from 100 to 10^{-3}, and the horizontal axis shows μ. The barrier method works well for values of μ larger than ≈ 3, but is otherwise not sensitive to the value of μ.



The barrier method

- An important issue is the choice of the initial value t_0 of t.
  (i) If t_0 is chosen too large, the first outer iteration will require too many inner iterations.
  (ii) If t_0 is chosen too small, the algorithm will require extra outer iterations, and possibly too many inner iterations in the first centering step.
- Since m/t_0 is the duality gap that results from the first centering step, it is reasonable to choose t_0 so that m/t_0 is approximately of the same order as $f\big(x^*(0)\big) - f^*$, or μ times this amount.
- For example, if a dual feasible point (λ, ν) is known, with duality gap $\eta = f\big(x^*(0)\big) - q(\lambda, \nu)$, then we can take $t_0 = m/\eta$. Thus, in the first outer iteration we simply compute a pair with the same duality gap as the initial primal and dual feasible points.

Final remarks

- The methods discussed in this section (central paths, barrier method, etc) can be extended to include conic optimization problems such as as SOCP and SDP discussed in Lecture Notes #4 and #5.
- *Primal-dual interior point methods* are often more efficient than the barrier method, especially when high accuracy is required.
  (i) They update primal and dual variables at each iteration; there is no distinction between inner and outer iterations.
  (ii) They often exhibit superlinear asymptotic convergence.
  (iii) Iterations can start at infeasible points.
  (iv) The cost per iteration same as for the barrier method.
- These topics are discussed in [1].

Recap

- Surprisingly many problems in finance can be expressed as convex optimization problems.
- Roughly speaking, tractability in optimization requires convexity: local optima are global.
- Unlike convex problems, algorithms for nonconvex optimization find local (often suboptimal) solutions (Levenberg-Marquardt, BFGS, ...), or are very expensive (differential evolution, ...).
- Interior-point methods require a small number of steps (20 80 steps in practice).
- Basic algorithms (Newton, barrier, . . .) are easy to implement and work well for small and medium size problems (and larger problems if the structure is exploited).

Credits

- Lecture Notes #4 – #6 follow largely the presentation of [1]. All figures have been copied from that book.

References

Boyd, S., and Vanderberghe, L.: *Convex Optimization*, Cambridge University Press (2004).