AMSC 607 / CMSC 878o Advanced Numerical Optimization
Fall 2008
UNIT 3: Constrained Optimization
PART 3: Penalty and Barrier Methods
Dianne P. O'Leary

Penalty and Barrier Methods

Reference: N&S Chapter 16

Two disadvantages of feasible direction methods:

- the need to guess the active set.
- the need to get an initial feasible point.

So we again build on our unconstrained algorithms, but in a different way.

Idea:

- We have a collection of methods that work for unconstrained problems:
  - Newton's method
  - steepest descent
  - quasi-Newton
  - ...
- Can we modify them to work when we have constraints?

The plan

- Barrier Methods
- Overcoming Ill-Conditioning in Barrier Methods
- Penalty Methods
- Overcoming Ill-Conditioning in Penalty Methods: Exact Penalty Methods

---

Barrier Methods

Barrier Methods

Idea: Suppose we have an initial feasible point.

We want to change the function f to raise a barrier at the boundary.

Picture

The formalism

The problem:

$$\min_{\mathbf{x}} f(\mathbf{x})$$
$$\mathbf{c}(\mathbf{x}) \geq \mathbf{0}$$

The Lagrangian:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{x})$$

Barrier function:

$$B_\mu(\mathbf{x}) = f(\mathbf{x}) + \mu\phi(\mathbf{x})$$

Possible choices of ϕ:

- Log barrier:
$$\phi(\mathbf{x}) = -\sum_{i=1}^{m} \log(c_i(\mathbf{x}))$$
- Inverse barrier:
$$\phi(\mathbf{x}) = \sum_{i=1}^{m} \frac{1}{c_i(\mathbf{x})}$$

The severity of the barrier is controlled by the choice of μ:

- large μ: gradual barrier
- small μ: sharp barrier

Picture

The tradeoff

- If μ is large and the minimizer is near the boundary:
  - the barrier function looks very different from f.
- If μ is small and the minimizer is near the boundary:
  - steep gradients.
  - ill-conditioning and therefore hard to solve the problem.

  Nice example in N&S p.534.

Because of this, we usually solve a sequence of problems:

- Start with a large μ to guide us near the solution.
- Gradually reduce μ, using our previous solution as a starting point.

Typical Convergence Result

Theorem: Under the conditions in N&S pp540-541 (continuity of the functions, bounded level sets, nonempty feasible set with some regularity) , let $x(\mu)$ solve the barrier problem

$$\min_{\mathbf{x}} B_\mu(\mathbf{x}) .$$

Then given a sequence $\mu_1 > \mu_2 > \ldots$ converging to zero, there exists a subsequence of $\{\mathbf{x}_{\mu_i}\}$ that converges to a local solution of our problem.

Proof: See book.

Even more important:

If some technical conditions hold (a constraint qualification, and no "accidentally zero" Lagrange multiplier), then the points $x(\mu)$ define a path called the barrier trajectory.

And this path is differentiable!

(We will use this later when we develop interior point methods.)

For a proof of this result, see N&S p. 532.

A closer look at the log barrier formulation

$$B_\mu(\mathbf{x}) = f(\mathbf{x}) - \mu \sum_{i=1}^{m} \log(c_i(\mathbf{x}))$$

Notation: To eliminate clutter in the equations, I will abbreviate the summation:

$$\sum_{i=1}^{m} = \sum .$$

To minimize B_μ with respect to $\mathbf{x}$, we set the derivative of this barrier function equal to zero:

$$\begin{aligned} \mathbf{0} &= \mathbf{g}(\mathbf{x}) - \mu \sum \frac{\nabla c_i(\mathbf{x})}{c_i(\mathbf{x})} \\ &= \mathbf{g}(\mathbf{x}) - \sum \frac{\mu}{c_i(\mathbf{x})} \nabla c_i(\mathbf{x}) \end{aligned}$$

But in order to solve our problem, what we really want to do is set the derivative of the Lagrangian to zero:

$$\mathbf{0} = \mathbf{g}(\mathbf{x}) - \mathbf{A}(\mathbf{x})^T \boldsymbol{\lambda} = \mathbf{g}(\mathbf{x}) - \sum \lambda_i \nabla c_i(\mathbf{x})$$

So we see that we have estimates of the Lagrange multipliers:

$$\lambda_i \approx \frac{\mu}{c_i(\mathbf{x})} \geq 0 .$$

This is good, but we haven't solved our original problem, since

$$\lambda_i(\mu) c_i(\mathbf{x}) = \mu, \quad \text{not } 0.$$

So, as μ gets small, we haven't solved our original problem, but we have solved a nearby one formed by changing 0 to μ!

A closer look at the inverse barrier function

$$B_\mu(\mathbf{x}) = f(\mathbf{x}) + \mu \sum \frac{1}{c_i(\mathbf{x})}$$

Unquiz: Go through a similar derivation for this barrier function, showing that the Lagrange multiplier estimates are

$$\lambda_i(\mu) = \frac{\mu}{(c_i(\mathbf{x}))^2} .$$

[]

Summary of the Barrier formulation: Log barrier function

The solution $\mathbf{x}(\mu)$ to

$$\min_{\mathbf{x}} B_\mu(\mathbf{x}) = \min_{\mathbf{x}} f(\mathbf{x}) - \mu \sum \log(c_i(\mathbf{x}))$$

is the solution to

$$\begin{aligned} \mathbf{g}(\mathbf{x}) - \mathbf{A}(\mathbf{x})^T \boldsymbol{\lambda} &= \mathbf{0} \\ \boldsymbol{\lambda} &\geq \mathbf{0} \\ \lambda_i c_i(\mathbf{x}) &= \mu,\ i = 1, \ldots, m \end{aligned}$$

and this is a perturbation of order μ of the optimality conditions for our original problem.

Properties of this formulation:

- We don't need to choose an active set, so there is no combinatorial explosion as the number of constraints gets large.
- We have convergence under rather mild conditions.
- We get estimates of the Lagrange multipliers free.
- B is convex if f and $-c_i$ are.
- The Hessian matrix of B is ill-conditioned as $\mu \to 0$. This makes it hard to compute the Newton direction, but we'll fix this in a minute.
- The barrier function is rather ill behaved for small μ so the line search must be specially designed:
  - A quadratic model does not fit the function well.
  - We need to model the singularity in the function.
- We need a strictly feasible initial guess.

Overcoming Ill-Conditioning in Barrier Methods

Note: This covers material similar to N&S 16.3 but is somewhat different.

We'll use the log barrier function as an example, and consider what we would need to do in order to compute the Newton direction.

The log barrier function:

$$B_\mu(\mathbf{x}) = f(\mathbf{x}) - \mu \sum \log\left(c_i(\mathbf{x})\right)$$

Differentiate with respect to $\mathbf{x}$:

$$\nabla B_\mu = \mathbf{g}(\mathbf{x}) - \mu \sum \frac{\nabla c_i(\mathbf{x})}{c_i(\mathbf{x})}$$

Calculate the second derivative matrix, recalling that $\lambda_i = \mu / c_i(x)$:

$$\begin{aligned} \nabla^2 B_\mu &= \mathbf{H} - \mu \sum \left[\frac{\nabla^2 c_i(\mathbf{x})}{c_i(\mathbf{x})} - \frac{\nabla c_i(\mathbf{x}) \nabla c_i(\mathbf{x})^T}{c_i(\mathbf{x})^2} \right] \\ &= \mathbf{H} - \sum \lambda_i \nabla^2 c_i(\mathbf{x}) + \frac{1}{\mu} \sum \lambda_i^2 \nabla c_i(\mathbf{x}) \nabla c_i(\mathbf{x})^T \\ &\equiv \mathbf{H}_L + \frac{1}{\mu} \mathbf{A}^T \mathbf{D} \mathbf{A} \end{aligned}$$

where $\mathbf{D}$ is a diagonal matrix with entries λ_i^2.

The first term, $\mathbf{H}_L$, is the Hessian of the Lagrangian function, and this is independent of μ.

The second term is troublesome. If the ith constraint is active at the solution, then as $\mu \to 0$, λ_i will be nonzero, so $\lambda_i^2 / \mu \to \infty$.

This causes the Hessian matrix for B_μ to blow up in $k < n$ different directions if k constraints are active at the solution. This is not good for Newton's method....

Linear algebra to the rescue

For simplicity, assume that all constraints are active at the solution, and all $\lambda_i > 0$.

Factor

$$\mathbf{A}^T = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

where

- $\mathbf{A}^T : n \times m$
- $\mathbf{Q}_1 : n \times m$
- $\mathbf{Q}_2 : n \times (n - m)$
- $\mathbf{R} : m \times m$
- $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$.

Then

$$\bigtriangledown^2 B_\mu = \mathbf{H}_L + \frac{1}{\mu}\mathbf{A}^T\mathbf{D}\mathbf{A} = \mathbf{H}_L + \frac{1}{\mu}\mathbf{Q}_1\mathbf{R}\mathbf{D}\mathbf{R}^T\mathbf{Q}_1^T.$$

Now let

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q}_2 & \mathbf{Q}_1 \end{bmatrix}.$$

Then

$$\begin{aligned} \tilde{\mathbf{Q}}^T\mathbf{A}^T\mathbf{D}\mathbf{A}\tilde{\mathbf{Q}} &= \begin{bmatrix} \mathbf{Q}_2^T \\ \mathbf{Q}_1^T \end{bmatrix} \mathbf{Q}_1\mathbf{R}\mathbf{D}\mathbf{R}^T\mathbf{Q}_1^T \begin{bmatrix} \mathbf{Q}_2 & \mathbf{Q}_1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}\mathbf{D}\mathbf{R}^T \end{bmatrix}, \end{aligned}$$

so

$$\begin{aligned} \tilde{\mathbf{Q}}^T \bigtriangledown^2 B_\mu \tilde{\mathbf{Q}} &= \tilde{\mathbf{Q}}^T\mathbf{H}_L\tilde{\mathbf{Q}} + \frac{1}{\mu}\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}\mathbf{D}\mathbf{R}^T \end{bmatrix} \\ &\equiv \tilde{\mathbf{H}} + \frac{1}{\mu}\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}\mathbf{D}\mathbf{R}^T \end{bmatrix} \end{aligned}$$

Now let's factor this matrix as

$$\tilde{\mathbf{Q}}^T \bigtriangledown^2 B_\mu \tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{K} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{H}}_{11} & \tilde{\mathbf{H}}_{12} \\ \mathbf{0} & \mathbf{G} \end{bmatrix}$$

where $\tilde{\mathbf{H}}_{ij}$ refers to a block of $\tilde{\mathbf{H}}$.

The first block row of this is clearly ok.

Setting the (2,1) block on the left-hand side equal to the block on the right-hand side yields

$$\tilde{\mathbf{H}}_{21} = \mathbf{K}\tilde{\mathbf{H}}_{11}$$

so

$$\mathbf{K} = \tilde{\mathbf{H}}_{21}{\tilde{\mathbf{H}}_{11}}^{-1}$$

and the conditioning of ${\tilde{\mathbf{H}}_{11}}^{-1}$ is not affected by μ.

Next, look at the (2,2) block:

$$\tilde{\mathbf{H}}_{22} + \frac{1}{\mu}\mathbf{R}\mathbf{D}\mathbf{R}^T = \mathbf{K}\tilde{\mathbf{H}}_{12} + \mathbf{G}$$

so

$$\begin{aligned} \mathbf{G} &= \frac{1}{\mu}\mathbf{R}\mathbf{D}\mathbf{R}^T + \tilde{\mathbf{H}}_{22} - \tilde{\mathbf{H}}_{21}{\tilde{\mathbf{H}}_{11}}^{-1}\tilde{\mathbf{H}}_{12} \\ &\rightarrow \frac{1}{\mu}\mathbf{R}\mathbf{D}\mathbf{R}^T \\ &\equiv \hat{\mathbf{G}} \end{aligned}$$

as $\mu \to 0$. Note that the condition of $\hat{\mathbf{G}}$ is independent of μ!

So we will replace $\mathbf{G}$ by $\hat{\mathbf{G}}$ in the factorization to get an approximate Newton direction.

Where are we?

We want to solve

$$\bigtriangledown^2 B_\mu \mathbf{p} = \mathbf{y}$$

where $\mathbf{y}$ is the negative gradient of the barrier function.

This is equivalent to solving

$$\tilde{\mathbf{Q}}^T \bigtriangledown^2 B_\mu \tilde{\mathbf{Q}} (\tilde{\mathbf{Q}}^T \mathbf{p}) = \tilde{\mathbf{Q}}^T \mathbf{y}\,.$$

How can we (approximately) solve this?

Algorithm:

1. Let $\tilde{\mathbf{y}} = \tilde{\mathbf{Q}}^T \mathbf{y}$.
2. Solve
$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{K} & \mathbf{I} \end{bmatrix} \mathbf{z} = \tilde{\mathbf{y}}\,.$$
Since $\mathbf{K}\tilde{\mathbf{H}}_{11} = \tilde{\mathbf{H}}_{21}$, we can solve this by forming
$$\begin{aligned} \mathbf{z}_1 &= \tilde{\mathbf{y}}_1 \\ \tilde{\mathbf{H}}_{11} q &= \mathbf{z}_1 \\ \mathbf{z}_2 &= \tilde{\mathbf{y}}_2 - \tilde{\mathbf{H}}_{21}\mathbf{q} \end{aligned}$$
3. Solve
$$\begin{bmatrix} \tilde{\mathbf{H}}_{11} & \tilde{\mathbf{H}}_{12} \\ \mathbf{0} & \hat{\mathbf{G}} \end{bmatrix} \tilde{\mathbf{p}} = \mathbf{z}$$
by forming
$$\begin{aligned} \hat{\mathbf{G}}\tilde{\mathbf{p}}_2 &= \mathbf{z}_2 \\ \tilde{\mathbf{H}}_{11}\tilde{\mathbf{p}}_1 &= \mathbf{z}_1 - \tilde{\mathbf{H}}_{12}\tilde{\mathbf{p}}_2 \end{aligned}$$
4. Form $\mathbf{p} = \tilde{\mathbf{Q}}\tilde{\mathbf{p}}$.

Then **p** is the approximate Newton direction and ill-conditioning has been avoided!

Penalty Methods

We have already developed most of the machinery we need to understand these methods, and they are not as important in practice as the Barrier methods, so we'll cover them in somewhat less detail.

Penalty Methods

If we don't have an initial feasible point, then none of our previous algorithms (feasible direction methods, barrier methods) can be applied.

What can we do? Use a penalty method.

Idea: Let

$$\pi(\mathbf{x}, \rho) = f(\mathbf{x}) + \rho\psi(\mathbf{x})$$

where ρ is a scalar penalty parameter and

$$\begin{aligned} \psi(\mathbf{x}) &= 0 \quad \mathbf{x} \text{ feasible} \\ &> 0 \quad \text{otherwise} \end{aligned}$$

Again we solve a sequence of problems

$$\min_{\mathbf{x}} \pi(\mathbf{x}, \rho_i)\,,$$

but now $\rho_i \to \infty$.

Examples of penalty functions:

- Quadratic loss function for equality constraints:

$$\psi(\mathbf{x}) = \frac{1}{2}\mathbf{c}(x)^T\mathbf{c}(x)$$

- Quadratic loss function for inequalities:

$$\psi(\mathbf{x}) = \frac{1}{2}\mathbf{c}_+(\mathbf{x})^T\mathbf{c}_+(\mathbf{x})$$

  where

$$\begin{aligned} (\mathbf{c}_+)_i &= c_i \quad c_i < 0 \\ &= 0 \quad c_i \geq 0 \end{aligned}$$

- One other loss function:

$$\psi(\mathbf{x}) = \frac{1}{\gamma} \sum |(\mathbf{c}_+(\mathbf{x}))_i|^\gamma$$

for a fixed parameter $\gamma \geq 1$.

Properties

- The sequence of optimal points $\{\mathbf{x}(\rho_i)\}$ is convergent (under mild assumptions)
- The points $\mathbf{x}(\rho)$ define a trajectory and yield estimates of Lagrange multipliers.

Example: quadratic loss function for equality constraints

$$\pi(\mathbf{x}, \rho) = f(\mathbf{x}) + \frac{1}{2}\rho \sum c_i(\mathbf{x})^2$$

Differentiate:

$$\bigtriangledown \pi = \mathbf{g}(\mathbf{x}) + \rho \sum c_i(\mathbf{x}) \bigtriangledown c_i(\mathbf{x}) = \mathbf{0}$$

Compare with Lagrangian conditions:

$$\mathbf{0} = \mathbf{g}(\mathbf{x}) - \sum \lambda_i \bigtriangledown c_i(\mathbf{x}),$$

so our estimates of Lagrange multipliers are

$$\lambda_i(\rho) = -\rho c_i(\mathbf{x})\,.$$

[]

Properties:

- The conditioning of the Hessian of the penalty function is increasingly bad as $\rho \to \infty$. (See the example in N&S p.538.)
- For inequality constraints, the second derivative can be discontinuous. This gives Newton's method a lot of trouble!

Overcoming Ill-Conditioning in Penalty Methods: Exact Penalty Methods

Reference: N&S 16.5.

Idea: Construct a penalty problem that is equivalent to the original problem.

Then we don't need to solve a sequence of problems!

Choice 1

Sacrifice differentiability.

$$\pi(\mathbf{x}, \rho) = f(\mathbf{x}) + \rho \sum |(\mathbf{c}_+(\mathbf{x}))_i|$$

This function is continuous, but fails to be differentiable when we hit a constraint.

Lemma: If $\mathbf{x}^*$ satisfies the 2nd order sufficient conditions for optimality, then there exists a number $\bar{\rho}$ such that $\mathbf{x}^*$ is an isolated local minimizer of $\pi(\mathbf{x}, \rho)$ for any ρ greater than $\bar{\rho}$.

Idea of Proof: See picture. Choose ρ large enough so that $\pi(\mathbf{x}, \rho) > f(\mathbf{x}^*)$ for infeasible points in the neighborhood, and so that $\mathbf{x}^*$ is the local minimizer for feasible points. []

A subtle point: We can't make this work with a quadratic penalty function – it is too flat at the constraint. See the example in N&S p538.

Choice 2

Put the Lagrange multipliers into the function explicitly: Augmented Lagrangian

We'll use the original function, plus a penalty term, plus the Lagrangian term:

$$\min_{\mathbf{x}} f(\mathbf{x}) + \frac{1}{2}\rho \mathbf{c}(\mathbf{x})^T \mathbf{c}(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{x})$$

We need to choose ρ large enough to make $\mathbf{x}^*$ an unconstrained minimizer of this function.

The resulting algorithm:

Given initial guesses $\mathbf{x}^{(0)}$, $\boldsymbol{\lambda}^{(0)}$, and $\rho^{(0)}$, set $k = 0$.

We iterate until optimality:

- Determine $\mathbf{x}^{(k+1)}$ by minimizing the augmented Lagrangian, fixing $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(k)}$ and $\rho = \rho^{(k)}$, using your favorite method.
- Get new estimates of the multipliers $\boldsymbol{\lambda}^{(k+1)}$ and increase $\rho^{(k)}$ to get $\rho^{(k+1)}$.
- Set $k = k + 1$.

How to estimate the multipliers: Differentiate the augmented Lagrangian that we just minimized, and evaluate it at $\mathbf{x}^{(k+1)}$:

$$\mathbf{g}(\mathbf{x}^{(k+1)}) - \mathbf{A}(\mathbf{x}^{(k+1)})^T(\boldsymbol{\lambda}^{(k)} - \rho^{(k)}\mathbf{c}(\mathbf{x}^{(k+1)})) = \mathbf{0}\,,$$

so we set

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} - \rho^{(k)}\mathbf{c}(\mathbf{x}^{(k+1)})\,.$$

An alternate interpretation of the augmented Lagrangian method

The iteration is really driven by $\boldsymbol{\lambda}$, not ρ.

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} - \rho^{(k)}\mathbf{c}(\mathbf{x}^{(k+1)})$$

means that we change $\boldsymbol{\lambda}$ by using the search direction $\mathbf{c}(\mathbf{x}^{(k+1)})$ and the step length parameter $\rho^{(k)}$.

This is steepest ascent on the dual problem

$$\max_{\boldsymbol{\lambda}} \min_{\mathbf{x}} f(\mathbf{x}) - \boldsymbol{\lambda}^T\mathbf{c}(\mathbf{x}) + \frac{1}{2}\rho\mathbf{c}(\mathbf{x})^T\mathbf{c}(\mathbf{x})$$

since if we differentiate $f(\mathbf{x}) - \boldsymbol{\lambda}^T\mathbf{c}(\mathbf{x}) + \frac{1}{2}\rho\mathbf{c}(\mathbf{x})^T\mathbf{c}(\mathbf{x})$ respect to $\boldsymbol{\lambda}$ we get

$$-\mathbf{c}(\mathbf{x}(\boldsymbol{\lambda}))\,.$$

For inequality constraints:

$$\min_{\mathbf{x}} f(\mathbf{x}) - \mu^{(k)} \sum (\boldsymbol{\lambda}^{(k)})_i \log(\mu^{(k)-1} c_i(\mathbf{x}) + 1)$$

(See N&S p560)

How do we update $\boldsymbol{\lambda}$?

$$\mathbf{g}(\mathbf{x}^{(k+1)}) - \mu^{(k)} \sum (\boldsymbol{\lambda}^{(k)})_i \frac{1}{\mu^{(k)-1} c_i(\mathbf{x}^{(k+1)}) + 1} \mu^{(k)-1} \bigtriangledown c_i(\mathbf{x}^{(k+1)}) = 0$$

so

$$(\boldsymbol{\lambda}^{(k+1)})_i = \frac{(\boldsymbol{\lambda}^{(k)})_i}{\mu^{(k)-1} c_i(\mathbf{x}^{(k+1)}) + 1} \,.$$

For augmented Lagrangians ...

... there is a tension between conditioning and convergence:

- fast if $\rho^{(k)}$ large ,
- but ill-conditioned, so hard to get a good solution.

Final Words

These methods still have their place, but their main usefulness to us is as forerunners of interior point methods.