Constrained Optimization

Conner DiPaolo[1]

1 Introduction

We have already covered methods used to solve convex optimization problems of the form

$$\text{minimize: } f(\mathbf{x})$$

where f is convex. This problem is called *unconstrained*, since the place on constraints on $\mathbf{x}$, besides that it must live in $\mathbb{R}^n$. Namely, one of the quickest and most consistent algorithms for solving such problems (ignoring some aspects of scaling) is *Newton's Method*, where we approximate our objective as a quadratic using the second order Taylor approximation $f(\mathbf{x}) \approx f(\mathbf{x}_i) + \nabla f(\mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^\top \nabla^2 f(\mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)$ and solve that in closed form to find the next iterate $\mathbf{x}_{i+1}$. Newton's method is *fast*, obeying a property called *quadratic convergence*. Once $\mathbf{x}_i$ gets close enough (for some unknown measure of 'enough') to the optimal solution, every Newton iteration will double the number of significant digits in your current approximation of the optimal $\mathbf{x}^\star$. When you are not close to the optimal solution, Newton's method linearly converges to the optimal $\mathbf{x}^\star$ (under some conditions including convexity). We have never addressed, up until this point, how to solve more general convex optimization problems which can be *constrained*

$$\begin{aligned} \text{minimize: } & f_i(\mathbf{x}) && \text{(objective function)} \\ \text{subj. to: } & f_i(\mathbf{x}) \leq 0 && \text{(inequality constraints)} \\ & A\mathbf{x} = \mathbf{b}. && \text{(equality constraints)} \end{aligned}$$

For this problem to be convex we require that the feasible set (the region of $\mathbb{R}^n$ where the constraints hold) be a convex set, and that the objective function be convex. This implies that all inequality constraints f_i must be convex, and that equality constraints must be affine. We will first attack how to solve equality constrained convex optimization problems before moving onto inequality constrained problems. Combining the two methods is trivial once you have both, and we will briefly address this.

[1]Much of this material originates from Boyd & Vandenberghe's *Convex Optimization*. If you're looking for a more detailed analysis of these methods, consult that text. It's free online at `http://stanford.edu/~boyd/cvxbook/`

2 Equality Constrained Minimization

2.1 Equality Constrained Newton's Method

For unconstrained Newton's method, at every iteration we approximated our objective as a quadratic about our current estimate for the optimal $\mathbf{x}^\star$, and found a new approximate optimum under that quadratic approximation. It would make sense, then, for equality constrained optimization problems of the form

$$\begin{aligned} \text{minimize: } & f(\mathbf{x}) \\ \text{subj. to: } & A\mathbf{x} = b \end{aligned}$$

to just approximate the objective as a quadratic at each iteration and solve that problem in closed form. Is a quadratic program with equality constraints solvable in closed form? Yes!

Example 2.1 (Equality Constrained Convex Quadratic Minimization)**.** *Consider the task of solving the problem*

$$\begin{aligned} \textit{minimize: } & \frac{1}{2}\mathbf{x}^\top P\mathbf{x} + \mathbf{q}^\top \mathbf{x} + \mathbf{r} \\ \textit{subj. to: } & A\mathbf{x} = \mathbf{b}. \end{aligned}$$

where $P \succeq 0$. From this condition on P we know the problem is convex. From the KKT conditions we know at optimality

$$A\mathbf{x}^\star = \mathbf{b}$$

to ensure primal feasibility. Further, we can see that this has a Lagrangian

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\nu}) = \frac{1}{2}\mathbf{x}^\top P\mathbf{x} + \mathbf{q}^\top \mathbf{x} + \mathbf{r} + (A\mathbf{x} - \mathbf{b})^\top \boldsymbol{\nu}.$$

From the KKT conditions, again, we know that $\nabla L = 0$ at optimality, so

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^\star, \boldsymbol{\nu}^\star) = P\mathbf{x}^\star + \mathbf{q} + A^\top \boldsymbol{\nu}^\star = 0.$$

Thus we have two equations in two unknowns, and can solve this system analytically in the block form

$$\begin{bmatrix} P & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^\star \\ \boldsymbol{\nu}^\star \end{bmatrix} = \begin{bmatrix} -\mathbf{q} \\ \mathbf{b} \end{bmatrix}$$

This is known as the KKT system, and can be solved as any linear system would.

So now let's attack the general equality constrained problem in a Newton-esque algorithm that we'll call Equality Constrained Newton's Method. First, create our approximate problem

$$\begin{aligned} \text{minimize: } & f(\mathbf{x}_i) + \nabla f(\mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^\top \nabla^2 f(\mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i) \\ \text{subj. to: } & A\mathbf{x} = \mathbf{b}. \end{aligned}$$

This is close to the form of the example above, but we need $A\mathbf{x} = \mathbf{b}$ to be of the form $A(\mathbf{x} - \mathbf{x}_i) = \mathbf{c}$ for some $\mathbf{c}$. Assuming $A\mathbf{x} = \mathbf{b}$ to begin with, this gives the problem

$$\begin{aligned} \text{minimize: } & f(\mathbf{x}_i) + \nabla f(\mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^\top \nabla^2 f(\mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i) \\ \text{subj. to: } & A(\mathbf{x} - \mathbf{x}_i) = \mathbf{0}. \end{aligned}$$

Denote $\Delta\mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$. Then at each step we solve the KKT system

$$\begin{bmatrix} \nabla^2 f(\mathbf{x}_i) & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_i \\ \boldsymbol{\nu}^\star \end{bmatrix} = \begin{bmatrix} -\nabla f(\mathbf{x}_i) \\ \mathbf{0} \end{bmatrix}$$

for $\Delta\mathbf{x}_i$ and give the next iterate $\mathbf{x}_{i+1} = \Delta\mathbf{x}_i + \mathbf{x}_i$, repeating this process. Note that this algorithm reduces to a regular Newton step if we have no equality constraints. In a cleaner form we have the following algorithm.

Equality Constrained Newton's Method
input : f, ∇f, $\nabla^2 f$, starting point $\mathbf{x}_0$ such that $A\mathbf{x}_0 = \mathbf{b}$, tolerance ϵ
output: optimal point $\mathbf{x}^\star$
$i \leftarrow 0$
while $||\Delta\mathbf{x}_i||_2 \geq \epsilon$ **do**
  solve the KKT system for $\Delta\mathbf{x}_i$
  $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \Delta\mathbf{x}_i$
  $i \leftarrow i + 1$
end
return $\mathbf{x}^\star = \mathbf{x}_{i+1}$

2.2 Infeasible Start Equality Constrained Newton's Method

One drawback of using the above algorithm is that we must start with some $\mathbf{x}_0$ such that $A\mathbf{x}_0 = \mathbf{b}$, which could be nontrivial to compute. It would be better if that step of generating feasibility is encapsulated in our algorithm. Now suppose that $A\mathbf{x}_0 \neq b$. Then to ensure primal feasibility in each step our original equality constraint $A(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b}$ gives $A\Delta\mathbf{x} = \mathbf{b} - A\mathbf{x}$. Note that if we have feasible $\mathbf{x}$ the equality constraint reduces to $A\Delta\mathbf{x} = \mathbf{0}$ as seen above in the feasible start method. Substituting this into the optimality conditions gives the revised KKT system

$$\begin{bmatrix} \nabla^2 f(\mathbf{x}_i) & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_i \\ \boldsymbol{\nu}^\star \end{bmatrix} = - \begin{bmatrix} \nabla f(\mathbf{x}_i) \\ A\mathbf{x} - \mathbf{b} \end{bmatrix}.$$

Using this new system to solve for $\Delta\mathbf{x}_i$ gives a feasible point $\mathbf{x}_i + \Delta\mathbf{x}_i$ after any iteration. This, then, gives a slightly more general algorithm, Infeasible Start Newton's Method.

Infeasible Start Newton's Method
input : f, ∇f, $\nabla^2 f$, starting point $\mathbf{x}_0$, tolerance ϵ
output: optimal point $\mathbf{x}^\star$
$i \leftarrow 0$
while $||\Delta \mathbf{x}_i||_2 \geq \epsilon$ **do**
  solve the infeasible start KKT system for $\Delta \mathbf{x}_i$
  $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \Delta \mathbf{x}_i$
  $i \leftarrow i + 1$
end
return $\mathbf{x}^\star = \mathbf{x}_{i+1}$

3 General Constrained Minimization

Consider the convex problem

$$\begin{aligned} \text{minimize: } & f_0(\mathbf{x}) \\ \text{subj. to: } & f_i(\mathbf{x}) \leq 0 \qquad i = 1, 2, \ldots, m \\ & A\mathbf{x} = \mathbf{b}. \end{aligned}$$

One way to turn this into a unconstrained optimization problem, taking motivation from the Lagrangian, would be to minimize the objective, but when the constraints aren't satisfied to bump the objective f_0 to infinity. We can do this with the indicator function

$$I_-(u) = \begin{cases} 0 & u \leq 0 \\ \infty & \text{otherwise.} \end{cases}$$

This gives the equivalent optimization problem

$$\begin{aligned} \text{minimize: } & f_0(\mathbf{x}) + \sum_{i=1}^{m} I_-(f_i(\mathbf{x})) \\ \text{subj. to: } & A\mathbf{x} = \mathbf{b} \end{aligned}$$

This is great, but we can't differentiate the objective and therefore can't use Infeasible Start Newton Method shown above to solve this problem. Instead, we approximate

$$I_-(u) \approx -\frac{1}{t}\log(-u).$$

As t approaches ∞, this approximate converges to I_- pointwise. This gives the *differentiable* optimization problem (assuming f_0 is differentiable, of course)

$$\begin{aligned} \text{minimize: } & f_0(\mathbf{x}) - \frac{1}{t}\sum_{i=1}^{m} \log(-f_i(\mathbf{x})) \\ \text{subj. to: } & A\mathbf{x} = \mathbf{b} \end{aligned}$$

Since the negative log of a convex function is convex, this problem also retains convexity, and we can solve it using Infeasible Start Newton Method! Call the solution to the above problem $\mathbf{x}_t^\star$. We can show (and Boyd and Vandenberghe do!) that $\mathbf{x}_t^\star$ is m/t suboptimal. Then if we want ϵ accuracy we could just set $t = m/\epsilon$ and solve that problem:

$$\begin{aligned} \text{minimize: } & f_0(\mathbf{x}) - \frac{\epsilon}{m}\sum_{i=1}^{m}\log(-f_i(\mathbf{x})) \\ \text{subj. to: } & A\mathbf{x} = \mathbf{b} \end{aligned}$$

This would work in theory, but practical issues with ϵ being too small, for example, causes this to not work well. We can change this slightly to work very well in practice. Instead of just starting with some large t, we solve a sequence of smaller problems $\mathbf{x}_{t_0}^\star, \mathbf{x}_{t_1}^\star, \ldots$ for increasing t_i until $t_i \geq m/\epsilon$, ensuring an accuracy ϵ. This algorithm is outlined as follows.

Barrier Method
input : strictly feasible $\mathbf{x}$, $t := t^{(0)} > 0$, $\mu > 1$, tolerance $\epsilon > 0$
output: optimal point $\mathbf{x}^\star$
$i \leftarrow 0$
while $m/t < \epsilon$ **do**
  compute $\mathbf{x}_{t(i)}^\star$ using $\mathbf{x}_{t^{i-1}}^\star$ as a starting point
  set $t \leftarrow \mu t$
end
return $\mathbf{x}^\star = \mathbf{x}_{t^{(i+1)}}^\star$

Boyd and Vandenberghe go into a detailed analysis of why this is true, but in practice setting μ between 3 and 100 or so seems to work well. Basically, if μ is small, each problem iterate is likely to be close to the previous problem, and we will hug the quadratic convergence of Newton's Method closely, resulting in few inner iterations but many outer iterations. If μ is large, the problem iterates are likely to be quite different from the previous iterates, which results in many inner iterations to solve the problems but few outer iterations. Empirically, when μ is between 3 and 100, these effects tend to cancel one another out.

The Barrier Method is what is known as an *Interior Point Method*, because it traverses the interior of the feasible set to find the optimal solution (when iterating over t) instead of hugging the boundary of the region, like the Simplex algorithm for linear programs does. With this algorithm, you should be able to solve all optimization problems we would deal ever deal with in the course.