Statistical Computing (36-350)

Lecture 18: Optimization II: Unconstrained, Deterministic Optimization

Cosma Shalizi

28 October 2013

Agenda

- Approximation versus time
- Reminder: Newton's method
- Coordinate descent
- Derivative-free optimization: Nelder-Mead
- Optimizing statistical functionals

How Good vs. How Fast?

Given an **objective function** $f : \mathscr{D} \mapsto R$, find

$$\theta^* = \underset{\theta}{\operatorname{argmin}} f(\theta)$$

Approximation: How close can we get to θ^*, and/or $f(\theta^*)$?
Time complexity: How many computer steps does that take?
Typically, trade off approximation vs. time
Generally:

- Small approximation $\Rightarrow$ more time
- Smooth or specially structured f $\Rightarrow$ less time
- Larger $\mathscr{D}$ $\Rightarrow$ more time
- Higher-dimensional $\mathscr{D}$ $\Rightarrow$ more time

Newton's Method

Taylor expand $f(\theta^*)$ around a favorite point θ:

$$f(\theta^*) \approx f(\theta) + (\theta^* - \theta)\nabla f(\theta) + \frac{1}{2}(\theta^* - \theta)^T \mathbf{H}(\theta)(\theta^* - \theta)$$

$\mathbf{H} =$ **Hessian**, matrix of 2nd partial derivatives

Newton's Method

Taylor expand $f(\theta^*)$ around a favorite point θ:

$$f(\theta^*) \approx f(\theta) + (\theta^* - \theta)\nabla f(\theta) + \frac{1}{2}(\theta^* - \theta)^T \mathbf{H}(\theta)(\theta^* - \theta)$$

$\mathbf{H} =$ **Hessian**, matrix of 2nd partial derivatives

Set gradient with respect to θ^* to zero and solve:

$$\begin{aligned} 0 &= \nabla f(\theta) + \mathbf{H}(\theta)(\theta^* - \theta) \\ \theta^* &= \theta - (\mathbf{H}(\theta))^{-1}\nabla f(\theta) \end{aligned}$$

Newton's Method

Taylor expand $f(\theta^*)$ around a favorite point θ:

$$f(\theta^*) \approx f(\theta) + (\theta^* - \theta)\nabla f(\theta) + \frac{1}{2}(\theta^* - \theta)^T \mathbf{H}(\theta)(\theta^* - \theta)$$

$\mathbf{H} =$ **Hessian**, matrix of 2nd partial derivatives

Set gradient with respect to θ^* to zero and solve:

$$\begin{aligned} 0 &= \nabla f(\theta) + \mathbf{H}(\theta)(\theta^* - \theta) \\ \theta^* &= \theta - (\mathbf{H}(\theta))^{-1}\nabla f(\theta) \end{aligned}$$

Works *exactly* if f is quadratic

so that $\mathbf{H}^{-1}$ exists, etc.

If f isn't quadratic, keep pretending it is until we get close to θ^*, when it will be nearly true

Newton's Method: The Algorithm

1. Start with guess for θ
2. While ((not too tired) and (making adequate progress))
   1. Find gradient $\nabla f(\theta)$ and Hessian $\mathbf{H}(\theta)$
   2. Set $\theta \leftarrow \theta - \mathbf{H}(\theta)^{-1} \nabla f(\theta)$
3. Return final θ as approximation to θ^*

Like gradient descent, but with inverse Hessian giving the step-size

"This is about how far you can go with that gradient"

Advantages and Disadvantages of Newton's Method

Pro:

- Step-sizes chosen adaptively through 2nd derivatives, much harder to get zig-zagging, over-shooting, etc.
- Only $O(\epsilon^{-2})$ steps to get within ϵ of optimum
- Only $O(\log\log\epsilon^{-1})$ for very nice functions

Cons:

- Hopeless if $\mathbf{H}$ doesn't exist or isn't invertible
- Need to take $O(p^2)$ second derivatives *plus* p first derivatives
- Need to solve $\mathbf{H}\theta_{\text{new}} = \mathbf{H}\theta_{\text{old}} - \nabla f(\theta_{\text{old}})$ for θ_{new}
  inverting $\mathbf{H}$ is $O(p^3)$, but cleverness gives $O(p^2)$ for solving

Coordinate Descent

Newton's method adjusts all coordinates at once
Try this instead:

1. Start with initial guess θ
2. While ((not too tired) and (making adequate progress))
   - For $i \in (1:p)$
     1. do 1D optimization over i^{th} coordinate of θ, holding the others fixed
     2. Update i^{th} coordinate to this optimal value
3. Return final value of θ

Needs a good 1D optimizer, and can bog down for very tricky functions, but can also be extremely fast and simple

Nelder-Mead, a.k.a. the Simplex Method

Try to cage θ^* with a **simplex** of $p+1$ points
Order the trial points, $f(\theta_1) \leq f(\theta_2) \ldots \leq f(\theta_{p+1})$
θ_{p+1} is the worst guess — try to improve it
$\theta_0 = \frac{1}{n}\sum_{i=1}^{n} \theta_i =$ center of the not-worst

- **Reflection**: Try $x_0 - (x_{p+1} - x_0)$, across the center from x_{p+1}
  - if it's better than x_p but not than x_1, replace the old x_{p+1} with it
  - **Expansion**: if the reflected point is the new best, try $x_0 - 2(x_{p+1} - x_0)$; replace the old x_{p+1} with the better of the reflected and the expanded point
- **Contraction**: If the reflected point is worse that x_p, try $x_0 + \frac{x_{p+1} - x_0}{2}$; if the contracted value is better, replace x_{p+1} with it
- **Reduction**: If all else fails, $x_i \leftarrow \frac{x_1 + x_i}{2}$

Making Sense of Nedler-Mead

The Moves:

- Reflection: try the opposite of the worst point
- Expansion: if that really helps, try it some more
- Contraction: see if we overshot when trying the opposite
- Reduction: if all else fails, try being more like the best point

Pros:

- Each iteration ≤ 4 values of f, plus sorting (at most $O(p \log p)$, usually much better)
- No derivatives used, can even work for dis-continuous f

Con:

- Can need *many* more iterations than gradient methods

Optimizing Statistical Functionals

Optimizing for statistics is funny: we know our objective function is noisy

Have $\hat{f}_n$ (sample objective) but want to minimize f (population objective)

Why optimize $\hat{f}_n$ to $\pm 10^{-6}$ when $\hat{f}$ only matches f to ± 1?

If $\hat{f}_n$ is an average over data points, then (law of large numbers)

$$\mathbb{E}\left[\hat{f}_n(\theta)\right] = f(\theta)$$

and (central limit theorem)

$$\hat{f}_n(\theta) - f(\theta) = O(n^{-1/2})$$

Can use probability theory to analyze how closely the sample optimum matches the population optimum

$$\begin{aligned}
\hat{\theta}_n &= \underset{\theta}{\operatorname{argmin}} \hat{f}_n(\theta) \\
\nabla \hat{f}_n(\hat{\theta}_n) &= 0 \\
&\approx \nabla \hat{f}_n(\theta^*) + \widehat{\mathbf{H}}_n(\theta^*)(\hat{\theta}_n - \theta^*) \\
\hat{\theta}_n &\approx \theta^* - \widehat{\mathbf{H}}_n^{-1}(\theta^*) \nabla \hat{f}_n(\theta^*)
\end{aligned}$$

Opposite expansion to Newton's method

$$\hat{\theta}_n \approx \theta^* - \widehat{\mathbf{H}}_n^{-1}(\theta^*)\nabla\hat{f}_n(\theta^*)$$

$$\hat{\theta}_n \approx \theta^* - \widehat{\mathbf{H}}_n^{-1}(\theta^*)\nabla\hat{f}_n(\theta^*)$$

When does $\widehat{\mathbf{H}}_n^{-1}(\theta^*)\nabla\hat{f}_n(\theta^*) \rightarrow 0$?

$$\hat{\theta}_n \approx \theta^* - \widehat{\mathbf{H}}_n^{-1}(\theta^*)\nabla\hat{f}_n(\theta^*)$$

When does $\widehat{\mathbf{H}}_n^{-1}(\theta^*)\nabla\hat{f}_n(\theta^*) \to 0$?

$$\begin{aligned}\widehat{\mathbf{H}}_n(\theta^*) &\to \mathbf{H}(\theta^*) \text{ (by LLN)} \\ \nabla\hat{f}_n(\theta^*) - \nabla f(\theta^*) &= O(n^{-1/2}) \text{ (by CLT)}\end{aligned}$$

but $\nabla f(\theta^*) = 0$

$$\hat{\theta}_n \approx \theta^* - \widehat{\mathbf{H}}_n^{-1}(\theta^*)\nabla\hat{f}_n(\theta^*)$$

When does $\widehat{\mathbf{H}}_n^{-1}(\theta^*)\nabla\hat{f}_n(\theta^*) \rightarrow 0$?

$$\begin{aligned}
\widehat{\mathbf{H}}_n(\theta^*) &\rightarrow \mathbf{H}(\theta^*) \text{ (by LLN)} \\
\nabla\hat{f}_n(\theta^*) - \nabla f(\theta^*) &= O(n^{-1/2}) \text{ (by CLT)}
\end{aligned}$$

but $\nabla f(\theta^*) = 0$

$$\begin{aligned}
\therefore \nabla\hat{f}_n(\theta^*) &= O(n^{-1/2}) \\
\mathrm{Var}\left[\nabla\hat{f}_n(\theta^*)\right] &\rightarrow n^{-1}\mathbf{K}(\theta^*) \text{ (CLT again)}
\end{aligned}$$

How much noise is there in $\hat{\theta}_n$?

$$\begin{aligned}
\mathrm{Var}\left[\hat{\theta}_n\right] &= \mathrm{Var}\left[\hat{\theta}_n - \theta^*\right] \\
&= \mathrm{Var}\left[\widehat{\mathbf{H}}_n^{-1}(\theta^*)\nabla\hat{f}_n(\theta^*)\right] \\
&= \widehat{\mathbf{H}}_n^{-1}(\theta^*)\mathrm{Var}\left[\nabla\hat{f}_n(\theta^*)\right]\widehat{\mathbf{H}}_n^{-1}(\theta^*) \\
&\rightarrow n^{-1}\mathbf{H}^{-1}(\theta^*)\mathbf{K}(\theta^*)\mathbf{H}^{-1}(\theta^*) \\
&= O(pn^{-1})
\end{aligned}$$

How much noise is there in $f(\hat{\theta}_n)$?

$$\begin{aligned}
f(\hat{\theta}_n) - f(\theta^*) &\approx \frac{1}{2}(\hat{\theta}_n - \theta^*)^T \mathbf{H}(\theta^*)(\hat{\theta}_n - \theta^*) \\
\mathrm{Var}\left[f(\hat{\theta}_n) - f(\theta^*)\right] &\approx \mathrm{tr}\left(\mathbf{H}(\theta^*)\mathrm{Var}\left[\hat{\theta}_n - \theta^*\right]\mathbf{H}(\theta^*)\mathrm{Var}\left[\hat{\theta}_n - \theta^*\right]\right) \\
&\to n^{-2}\mathrm{tr}\left(\mathbf{K}(\theta^*)\mathbf{H}^{-1}(\theta^*)\mathbf{K}(\theta^*)\mathbf{H}^{-1}(\theta^*)\right) \\
&= O(pn^{-2})
\end{aligned}$$

What You Need to Remember

If everything works out ideally (maximum likelihood, correct model) $\mathbf{K} = \mathbf{H}$, and

$$\begin{aligned} \hat{\theta}_n &\approx \theta^* - \widehat{\mathbf{H}}_n^{-1}(\theta^*)\nabla \hat{f}_n(\theta^*) \\ \mathrm{Var}\left[\hat{\theta}_n\right] &\approx n^{-1}\mathbf{H}^{-1}(\theta^*) \approx n^{-1}\mathbf{H}(\hat{\theta}_n) \\ \mathrm{Var}\left[f(\hat{\theta}_n) - f(\theta^*)\right] &\approx n^{-2}p \end{aligned}$$

If $\mathbf{K} \neq \mathbf{H}$, do the algebra and deal with more noise

$\therefore$ Little point to optimizing $\hat{f}_n$ *much* more precisely than $\pm\sqrt{p/n^2}$

Summary

1. Trade-offs: complexity of iteration vs. number of iterations vs. precision of approximation
2. Noise limits how much optimization is worth doing
3. For smooth problems, we can calculate uncertainty from the Hessian and the gradient