

Lecture 15 — 2/27

Lecturer: Mert Pilanci

Scribe: Hitha Revalla

15.1 Newton's method

Duality plays a very fundamental role in designing second-order methods for convex optimization. Newton's method is a second-order method in the simplest setting where we consider unconstrained smooth convex optimization (same as the setting for gradient descent).

$$\min_x f(x)$$

Recall that in gradient descent, the update in the k th iteration, $x^{(k)}$ moved in the direction of the negative gradient of the previous iteration

$$x^{(k)} = x^{(k-1)} - t_k \nabla f(x^{(k-1)}) \quad k = 1, 2, 3, \dots$$

where t_k is the step-size. In contrast, in Newton's method we move in the direction of negative Hessian inverse of the gradient.

$$x^{(k)} = x^{(k-1)} - (\nabla^2 f(x^{(k-1)}))^{-1} \nabla f(x^{(k-1)})$$

This is called the pure Newton's method, since there's no notion of a step size involved. As is evident from the update, Newton's method involves solving linear systems in the Hessian. To motivate Newton's method, consider the following quadratic approximation at x

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} (y - x)^T \nabla^2 f(x) (y - x)$$

The Newton update is obtained by minimizing the above w.r.t. y . This quadratic approximation is better than the approximation used in gradient descent (given by 14.1), since it uses more information about the function via the Hessian.

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} (y - x)^T \nabla^2 f(x) (y - x)$$

15.1.1 Convergence Analysis

The pure Newton method doesn't necessarily converge. Depending on where we start, the Newton method can either converge or diverge "quadratically quickly". In practice, backtracking line search is used with Newton's method, with parameters $0 < \alpha \leq 1/2$, $0 < \beta < 1$

just like first-order methods. In Newton's method with backtracking, we start with $t = 1$.

While $f(x + tv) > f(x) + \alpha t \nabla f(x)^T v$ we shrink $t = \beta t$, else we perform the Newton update. Here, $v = \nabla^2 f(x)^{-1} \cdot \nabla f(x)$. Note that $\nabla f(x)^T v = -\lambda^2(x)$.

Newton's method involves solving the Hessian ($\text{in } O(n^3)$ flops), which might be expensive itself.

Convergence Result

Let us assume that f is strongly convex with parameter m and twice differentiable and $\text{dom}(f) = \mathbb{R}^n$. Let us also assume that f is Lipschitz with parameter L . These conditions guarantee that gradient descent on the same problem will have linear convergence. Recall that the gradient descent convergence rate depends adversely on the condition number, L/m . Let us additionally assume that $\nabla^2 f(x)$ is Lipschitz with parameter H . Hence,

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_F \leq H \|x - y\|_2$$

where $\nabla^2 f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$.

The convergence results hold for Newton's method with backtracking (with parameters α, β) and depend on two parameters, $\gamma > 0$ and $0 < \eta \leq m^2/H$. Let k_0 be the number of steps till $\|\nabla f(x(k_0 + 1))\|_2 < \eta$. k_0 breaks the convergence theorem into two stages, where $\eta = \min\{1, 3(1 - 2)\}m^2/H$.

In the first stage, when $k \leq k_0$,

$$f(x(k)) - f \leq (f(x^{(0)}) - f^*)\gamma k$$

This might not be very fast, but we are guaranteed to decrease the criterion by γ in every step where $\gamma = \alpha\beta^2\eta^2m/L^2$. If a function is poorly conditioned, then γ is fairly small and we cannot decrease the criterion by much in each step. The above bound is very conservative and this phase of convergence is called the damped Newton phase. There exists a second regime of convergence when $k > k_0$

$$f(x^{(k)}) - f^* \leq \frac{2m^3}{H^2} \left(\frac{1}{2}\right)^{2k-k_0+1}$$

This rate of convergence is extremely fast, and hence is named quadratic convergence. This phase is called the pure Newton phase. In the pure Newton phase, $t = 1$ is always satisfied for backtracking, there is no decay in the step size t . This means that once we enter the pure Newton phase, we won't leave it. To reach an desired level of accuracy, where $f(k) - f \leq \epsilon$, the number of iterations required to leave the first phase are $\frac{f(x^{(0)}) - f}{\gamma}$. For the second phase, we can use strong convexity to prove that the number of iterations required is $\frac{f(x^{(0)}) - f}{\gamma} + \log(\log(\epsilon_0/\epsilon))$

Self-concordance

The convergence results for Newton's method might seem dissatisfying because we know that it is affine invariant but the results involve constants L, m and H . A scale-free analysis has been proposed by Nesterov and Nemirovskii for self-concordant functions, f . A function is self-concordant if it satisfies

$$|f'''(x)| \leq 2f''(x)^{3/2} \forall x$$

For example, $f(x) = \log x$ is self-concordant. If a function f is self-concordant, then Newton's method with backtracking line search requires at most $C(\alpha, \beta)(f(x^{(0)}) - f^*) + \log(\log(1/\epsilon))$ steps to reach level of accuracy. This result is interesting because it does not involve the Lipschitz or strong convexity constants and is a better way of characterizing Newton's method.

15.1.2 Log Barrier Method

Previously, we looked at Newton's method for minimizing twice differentiable convex functions with equality constraints. One of the limitations of this method is that we cannot deal with inequality constraints. The barrier method is a way to address this issue. Formally, given the following problem,

$$\min f(x)$$

subject to

$$h_i(x) \leq 0, i = 1, \dots, m$$

$$Ax = b$$

assuming that f, h_i are all convex and twice differentiable functions, all with domain \mathbb{R}^n , the log barrier is defined as

$$\theta(x) = - \sum_{i=1}^m \log(-h_i(x))$$

It can be seen that the domain of the log barrier is the set of strictly feasible points, $x : h_i(x) < 0, i = 1 \dots m$. Note that the equality constraints are ignored for the rest of this chapter, because those can be solved using the Newton's method directly.

Log Barrier Calculus

The gradient and hessian of the log-barrier, defined as follows will be useful in the Newton's method with log-barrier

$$\begin{aligned} \nabla \theta(x) &= - \sum_{i=1}^m \frac{1}{h_i(x)} \nabla h_i(x) \\ \nabla^2 \theta(x) &= \sum_{i=1}^m \frac{1}{h_i(x)^2} \nabla h_i(x) \nabla h_i(x)^T + \sum_{i=1}^m \frac{1}{h_i(x)} \nabla^2 h_i(x) \end{aligned}$$

15.2 Randomised Newton's Method

Newton's method converges in superlinear time, but Newton's method requires inverting the hessian, which is prohibitively expensive for large datasets.

The problem is that we have to solve linear system $Hx = \nabla f(x^t)$ at each iteration. Randomized newton reduce the cost per iteration by replacing the hessian with random matrix D_t such that

$$E[d_t] = H(x^t)$$

We will use a sketch matrix S to form a random vector d_t such that

$$E[d_t] = H(x^t)$$

Recall our setup:

$$x^{t+1} = \operatorname{argmin}_{x \in \mathbb{R}^d} f(x^t) + \nabla f(x^t)^T(x - x^t) + \frac{1}{2}(x - x^t)^T H(x^t)(x - x^t)$$

Now supposed we have some hessian square root matrix $L \in \mathbb{R}^{n \times d}$. ie. $L^T L = H(x)$. Consider $f(x) = g(Ax)$, where $g: \mathbb{R}^n \rightarrow \mathbb{R}$ has the separable form $g(Ax) = \sum_i^n g_i(a_i^T x)$. In this case, $L = \operatorname{diag}(g''(a_i^T x))_{i=1}^n A$

The standard Newton's Method now becomes:

$$x^{t+1} = \operatorname{argmin}_{x \in \mathbb{R}^d} f(x^t) + \nabla f(x^t)^T(x - x^t) + \frac{1}{2}(x - x^t)^T H(x^t)(x - x^t)$$

$$x^{t+1} = \operatorname{argmin}_{x \in \mathbb{R}^d} f(x^t) + \nabla f(x^t)^T(x - x^t) + \frac{1}{2} \|L(x - x^t)\|_2^2$$

To randomized:

$$x^{t+1} = \operatorname{argmin}_{x \in \mathbb{R}^d} f(x^t) + \nabla f(x^t)^T(x - x^t) + \frac{1}{2} S_t \|L(x - x^t)\|_2^2$$

Where $S_t \in \mathbb{R}^{m \times n}$ is an independent realization of a sketching matrix at iteration t . Solving for x :

$$\begin{aligned} x^{t+1} &= x^t - D_t^{-1} \nabla f(x^t) \\ D_t &= L^T S_t^T S_t L \end{aligned}$$

We see:

$$E[D_t] = E[L^T S_t^T S_t L] = L^T E[S_t^T S_t] L = L^T L = H(x)$$

Each step of the newton sketch algorithm can be computed in $O(md^2)$ using conjugate gradient instead of $O(nd^2)$ of standard newton.

Convergence

If m is chosen to satisfy certain conditions, the unconstrained newton sketch algorithm achieves linear convergence:

$$f(x^t) - f(x^*) \leq \frac{\beta\gamma}{8L} \left(\frac{1}{2} + \epsilon \frac{\beta}{\gamma}\right)^t$$

Where $\beta = \lambda \min(H(x))$, $\gamma = \lambda \max(H(x^*))$ and we assume the hessian is Lipschitz continuous, i.e. $\|H(x) - H(y)\| \leq L\|x - y\|_2$.

Bibliography

- [1] • S. Boyd and L. Vandenberghe(2004), “Convex Optimization,” Chapters 9 and 10.
- [2] • R. Tibshirani(2015), “Newton’s Method”.