10-725/36-725: Convex Optimization **Fall 2016**

Lecture 14: Newton's Method

Lecturer: Javier Pena *Scribes: Varun Joshi, Xuan Li*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

14.1 Review of previous lecture

Given a function $f : \mathbb{R}^n \to \mathbb{R}$, we define its conjugate $f^* : \mathbb{R}^n \to \mathbb{R}$ as,

$$f^*(y) = \max_x (y^T x - f(x))$$

Some properties of the convex conjugate of a function are as follows:

- Conjugate f^* is always convex (regardless of the convexity of f)
- When f is quadratic and $Q \succ 0$ then f^* is quadratic in Q^{-1} i.e., for $f(x) = \frac{1}{2}x^T Qx + b^T x$, with $Q \succ 0$, $f^*(y) = \frac{1}{2}(y-b)^T Q^{-1}(y-b)$.
- When f is a norm, f^* is the indicator of the dual norm unit ball
- When f is closed and convex $x \in \partial f^*(y) \iff y \in \partial f(x)$

A key result that helps us write down the dual in terms of the conjugate is the Fenchel duality:

$$Primal : \min_x f(x) + g(x)$$
$$Dual : \max_u -f^*(u) - g^*(-u)$$

14.2 Introduction

In this section, we present the Newton's method and show that it can be interpreted as minimizing a quadratic approximation to a function at a point. We also briefly discuss the origin of Newton's method and how it can be used for finding the roots of a vector-valued function.

14.2.1 Newton's Method

Newton's method is a second-order method in the setting where we consider the unconstrained, smooth convex optimization problem

$$\min_x f(x)$$

where f is convex, twice differentiable and $dom(f) = \mathbb{R}^n$.

Newton's method: choose initial $x^{(0)} \in \mathbb{R}^n$, and

$$x^{(k)} = x^{(k-1)} - \left(\nabla^2 f(x^{(k-1)})\right)^{-1} \nabla f(x^{(k-1)}), \quad k = 1, 2, 3, \ldots$$

This is called pure Newton's method since there is no concept of a step-size involved. In Newton's method, we move in the direction of the negative Hessian inverse times the gradient. Compare this to gradient descent where we move in the direction of the negative gradient: choose initial $x^{(0)} \in \mathbb{R}^n$, and

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

14.2.2 Newton's method interpretation

Newton's method can be interpreted as minimizing a quadratic approximation to a function at a given point. The step $x^+ = x - \big(\nabla^2 f(x)\big)^{-1} \nabla f(x)$ can be obtained by minimizing over y the following quadratic approximation:

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

On the other hand, the gradient descent step $x^+ = x - t\nabla f(x)$ can be obtained by minimizing over y the following quadratic approximation:

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2t}\|y - x\|_2^2$$

As we can see, Newton's method minimizes a finer quadratic approximation to a function as compared to gradient descent. For example, for minimizing the function $f(x) = (10x_1^2 + x_2^2)/2 + 5\log(1 + \exp(-x_1 - x_2))$ a comparison of the steps taken by Newton's method and gradient descent is provided in figure 14.1. The figure

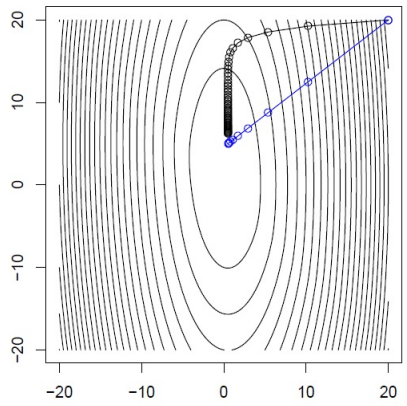

Figure 14.1: Comparison of Newton's Method (blue) with Gradient Descent (black)

shows a contrast between the behaviour of Newton's method and gradient descent. In gradient descent the direction of steps is always perpendicular to the level curves while that is not the case in Newton's method (due to the hessian term).
For a quadratic one step of Newton's method minimizes the function directly because the quadratic approximation to the quadratic function will be the function itself.

14.2.3 Newton's method for root finding

Newton's method was originally developed by Newton (1685) and Raphson (1690) for finding roots of polynomials. This was later generalized to minimization of nonlinear equations by Simpson (1740). Suppose $F : \mathbb{R}^n \to \mathbb{R}^n$ is a differentiable vector-valued function and consider the system of equations

$$F(x) = 0$$

Then, the Newton's method for finding the solution to this system of equations is: choose initial $x^{(0)} \in \mathbb{R}^n$, and

$$x^{(k)} = x^{(k-1)} - \left(F^{'}(x^{(k-1)})\right)^{-1} F(x^{(k-1)}), \quad k = 1, 2, 3, \ldots$$

where $F^{'}(x)$ is the Jacobian matrix of F at x.
The Newton step $x^{+} = x - F^{'}(x)^{-1}F(x)$ can be obtained by solving over y the linear approximation

$$F(y) \approx F(x) + F^{'}(x)(y - x) = 0$$

Newton's method for root finding is directly related to the Newton's method for convex minimization. In particular, newton's method for

$$\min_x f(x)$$

is the same as Newton's method for finding the roots of

$$\nabla f(x) = 0.$$

14.3 Properties

In this section, we present two key properties of Newton's method which distinguish it from first order methods.

14.3.1 Affine Invariance

Assume $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable and $A \in \mathbb{R}^{n \times n}$ is nonsingular. Let $g(y) := f(Ay)$. Then, Newton step for g at the point y is given by

$$y^{+} = y - \left(\nabla^2 g(y)\right)^{-1} \nabla g(y)$$

For the affine transformation $x = Ay$, it turns out that the Newton step for f at the point x is $x^{+} = Ay^{+}$. This means that the progress of Newton's method is independent of linear scaling. This property is not true for gradient descent.

14.3.2 Local Convergence

Newton's method has the property of local convergence. The formal statement of the property is as follows.

Theorem 14.1 *Assume $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable and $x^\star \in \mathbb{R}^n$ is a root of F, that is, $F(x^\star) = 0$ such that $F'(x^\star)$ is non-singular. Then*
(a) There exists $\delta > 0$ such that if $\|x^{(0)} - x^\star\| < \delta$ then Newton's method is well defined and

$$\lim_{k \to \infty} \frac{\|x^{(k+1)} - x^\star\|}{\|x^{(k)} - x^\star\|} = 0.$$

(b) If F' is Lipschitz continuous in a neighbourhood of $x^\star$ then there exists $K > 0$ such that

$$\|x^{(k+1)} - x^\star\| \leq K\|x^{(k)} - x^\star\|^2.$$

Part (a) of the theorem says that Newton's method has super-linear local convergence. Note that this is stronger than linear convergence: $x^{(k)} \to x^\star$ linearly $\iff \|x^{(k+1)} - x^\star\| \leq c\|x^{(k)} - x^\star\|$ for some $c \in (0, 1)$. If we further assume that F' is Lipschitz continuous then from part (b) we get that Newton's method has local quadratic convergence which is even stronger than super-linear convergence.

Note that the above theorem talks only about local convergence so it holds only when we are close to the root. Newton's method does not necessarily converge in the global sense.

14.4 Newton Decrement

For a smooth, convex function f the Newton decrement at a point x is defined as

$$\lambda(x) = \left(\nabla f(x)^T \left(\nabla^2 f(x)\right)^{-1} \nabla f(x)\right)^{1/2}$$

For an unconstrained convex optimization problem

$$\min_x f(x)$$

there are two ways to interpret the Newton Decrement.

Interpretation 1: Newton decrement relates the difference between $f(x)$ and the minimum of its quadratic approximation:

$$\begin{aligned} f(x) - \min_y \left(f(x) + \nabla f(x)^T (y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(x)(y-x)\right) \\ = \frac{1}{2} \nabla f(x)^T \left(\nabla^2 f(x)\right)^{-1} \nabla f(x) = \frac{1}{2}\lambda(x)^2 \end{aligned}$$

Thus, we can think of $\lambda(x)^2/2$ as an approximate bound on the suboptimality gap $f(x) - f^\star$. The bound is approximate because we are considering only the minimum of the quadratic approximation, not the actual minimum of $f(x)$.

Interpretation 2: Suppose the step in Newton's method is denoted by $v = -\left(\nabla^2 f(x)\right)^{-1} \nabla f(x)$, then

$$\lambda(x) = \left(v^T \nabla^2 f(x) v\right)^{1/2} = \|v\|_{\nabla^2 f(x)}$$

Thus, $\lambda(x)$ is the length of the Newton step in the norm defined by the Hessian.

Fact: Newton decrement is affine invariant i.e., for $g(y) = f(Ay)$ for a nonsingular A, $\lambda_g(y) = \lambda_f(x)$ at $x = Ay$.

14.5 Convergence Analysis for backtracking line search

14.5.1 Introduction to algorithm

The pure Newton's Method does not always converge, depending on the staring point. Thus, damped Newton's method is introduced to work together with pure Newton Method. With $0 < \alpha \leq \frac{1}{2}$ and $0 < \beta < 1$, at each iteration we start with $t = 1$, and while

$$f(x + tv) <= f(x) + \alpha t \nabla f(x)^T v$$

we perform the the Newton update, else we shrink $t = \beta t$. Here

$$v = -\left(\nabla^2 f(x)\right)^{-1} \nabla f(x)$$

14.5.2 Example: logistic regression

In lecture we are given a logistic regression example with $n = 500$ and $p = 100$. With backtracking, the Newton's Method is compared with gradient descent and the covergence curve is shown in 14.2. It is seen that Newton's Method has a different regime of convergence. Notice that the comparison might be unfair since the computation cost in these two methods might vary significantly.

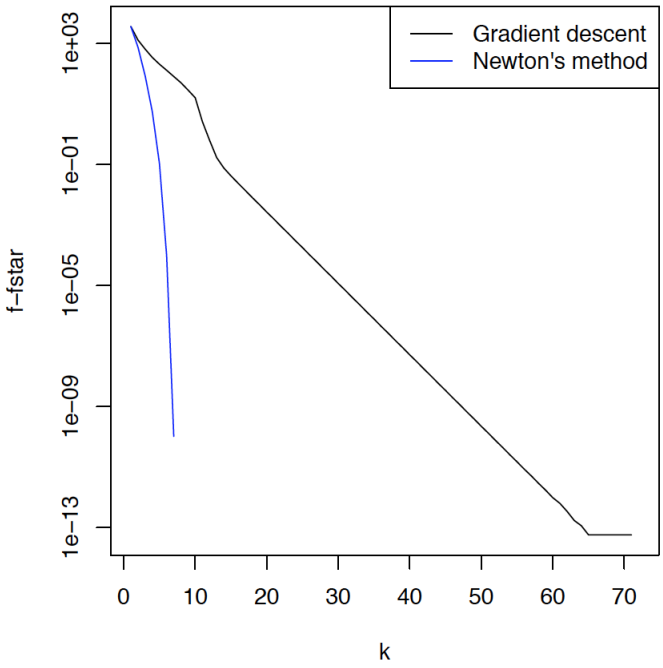

Figure 14.2: Comparison of Newton's Method with Gradient Descent (backtracking)

14.5.3 Convergence analysis

Given the assumption that

- f is strongly convex with parameter L, twice differentiable, and $\text{dom}(f) = \mathbb{R}^n$
- $\nabla^2 f$ Lipschitz with parameter M

Newton's Method with backtracking line search satisfies the following convergence bounds

$$f(x^{(k)}) - f^\star \leq \begin{cases} \left(f(x^{(0)}) - f^\star\right) - \gamma k & \text{if } k \leq k_0 \\ \frac{2m^3}{M^2}(\frac{1}{2})^{2^k - k_0 + 1} & \text{if } k > k_0 \end{cases}$$

where $\gamma = \alpha\beta^2\eta^2 m/L^2$, $\eta = \min\{1, 3(1-2\alpha)\}m^2/M$, and k_0 is the number of steps till $||\nabla f(x^{(k_0+1)})||_2 < \eta$. More precisely, the results indicates that in damped phase, we have

$$f(x^{(k+1)}) - f(x^{(k)}) \leq \gamma$$

In pure phase, backtracking selects $t = 1$, we have

$$\frac{M}{2m^2}||\nabla f(x^{(k+1)})||_2 \leq \left(\frac{M}{2m^2}||\nabla f(x^k)||_2\right)^2$$

Also, **once we enter pure phase, we won't leave.**
Finally, to reach $f(x^{(k)}) - f^\star \leq \epsilon$, at most

$$\frac{f(x^{(k)}) - f^\star}{\gamma} + \log\log(\epsilon_0/\epsilon)$$

iterations are need, where $\epsilon_0 = \frac{2m^3}{M^2}$.
The "log log" term in the convergence result makes the convergence quadratic. However, the quadratic convergence result is only local, it is guaranteed in the second or pure phase only. Finally, the above bound depends on L, m, M, but the algorithm itself does not.

14.6 Convergence Analysis for self concordant functions

14.6.1 Definition

To achieve a scale-free analysis we introduce self-concordant functions. A function is self-concordant if it convex on an open segment of $\mathbb{R}$ and satisfies

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

Two example would be $f(x) = -\sum_{i=1}^{n} \log(x_j)$ and $f(X) = -\log(\det(X))$.

14.6.2 Property

If g is self-concordance and A, b are of the right dimension, then

$$f(x) := g(Ax - b)$$

is also self-concordant.

14.6.3 Convergence Analysis

For self-concordant function f, Newton's method with backtracking line search needs at most

$$C(\alpha, \beta)(f(x^{(0)}) - f^{\star}) + \log\log(1/\epsilon)$$

iterations to achieve $f(x^k) - f^{\star} \leq \epsilon$ where α, β are constants.

14.7 Comparison to first order methods

14.7.1 High-level comparison

- **Memory** : Each iteration of Newton's method requires $O(n^2)$ storage due to the $n \times n$ Hessian whereas each gradient iteration requires $O(n)$ storage for the n-dimensional gradient.
- **Computation** : Each Newton iteration requires $O(n^3)$ flops as it solves a dense $n \times n$ linear system. Each gradient descent iteration requires $O(n)$ flops attributed to scaling/adding n-dimensional vectors.
- **Backtracking** : Backtracking line search has roughly the same cost for both methods, which use $O(n)$ flops per inner backtracking step.
- **Conditioning** : Newton's method is not afected by a problem's conditioning(due to affine invariance), but gradient descent can seriously degrade, since it depends adversely on the condition number.
- **Fragility** : Newton's method may be empirically more sensitive to bugs/numerical errors, whereas gradient descent is more robust.

We can see that even though Newton's method has quadratic convergence as compared to linear convergence of gradient descent, however, computing the Hessian might make the method a lot slower. If the Hessian is sparse and structured(e.g. banded), then both memory and computation are O(n).

14.8 Equality-constrained Newton's method

14.8.1 Introduction

Suppose now we have problems with equality constraints

$$\min_x f(x) \quad \text{subject to } Ax = b$$

Here we have three options: eliminating the equality constraints by writing $x = Fy + x_0$, where F spans null space of A, and $Ax_0 = b$; deriving the dual; or use the most straightforward option equality- constrained Newton's Method.

14.8.2 Definition

In equality-constrained Newton's Method, we take Newton steps which are confined to a region satisfied by the constraints. The Newton update is now $x^+ = x + tv$ where

$$v = \underset{A(x+z)=b}{\operatorname{argmin}} \left(f(x) + \nabla f(x)^T z + \frac{1}{2} z^T \nabla^2 f(x) z \right)$$

From KKT condition it follows that for some u, v we have

$$\begin{bmatrix} \nabla^2 & f(x)A^T \\ A & 0 \end{bmatrix} \cdot \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} \nabla f(x) \\ Ax - b \end{bmatrix}$$

The latter is the root-finding Newton step for KKT conditions of the origin equality-constrained problem that

$$\begin{bmatrix} \nabla f(x) + A^T y \\ Ax - b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

References

- S. Boyd and L. Vandenberghe (2004), "Convex optimization", Chapters 9 and 10
- Guler (2010),"Foundations of Optimization", Chapter 14.
- Y. Nesterov (1998), "Introductory lectures on convex optimization: a basic course", Chapter 2
- Y. Nesterov and A. Nemirovskii (1994), "Interior-point polynomial methods in convex programming", Chapter 2
- J. Nocedal and S. Wright (2006), "Numerical optimization", Chapters 6 and 7
- L. Vandenberghe, Lecture notes for EE 236C, UCLA, Spring 2011-2012