Newton’s Method and Self-Concordance

• Differentiable convex optimization and acceleration
• Newton’s Method
• Armijo backtracking search
• self-concordant functions
• Interior Point Method
Unconstrained Differentiable Convex Optimization

\[ \min_x f(x) \]

- \( f(x) \) strongly convex and differentiable

\[ \partial f(x) = \{ \nabla f(x) \} \]

- subgradient descent = gradient descent
Gradient Descent for Strongly Convex Functions

• recall strong convexity

A convex function $f$ is called strongly convex if there exists two positive constants $\beta_- \leq \beta_+$ such that

$$\beta_- I \preceq \nabla^2 f(x) \preceq \beta_+ I$$

for every $x$ in the domain of $f$

• Equivalent to

$$\lambda_{\min}(\nabla^2 f(x)) \geq \beta_-$$
$$\lambda_{\max}(\nabla^2 f(x)) \leq \beta_+$$
Gradient Descent for Strongly Convex Functions

\[ x_{t+1} = x_t - \mu_t \nabla f(x_t) \]

- Suppose that \( f \) is strongly convex with parameters \( \beta_-, \beta_+ \)

define \( f^* := \min_x f(x) \)

**Convergence result:** Using constant step-size \( \mu_t = \frac{1}{\beta_+} \), we have

\[ f(x_{t+1}) - f^* \leq (1 - \frac{\beta_-}{\beta_+})(f(x_t) - f^*) \]

recursively applying we get

- \( f(x_k) - f^* \leq (1 - \frac{\beta_-}{\beta_+})^k (f(x_0) - f^*) \)
Gradient Descent for Strongly Convex Functions

- linear convergence

- rate depends on the curvature

\[ f(x_k) - f^* \leq (1 - \frac{\beta_-}{\beta_+})^k (f(x_0) - f^*) \]

- minimizing \( f(Ax) \) where \( A \in \mathbb{R}^{n \times d} \) via Gradient Descent takes

\[ O(\kappa nd \log(\frac{1}{\epsilon})) \] operations where \( \kappa = \frac{\beta_+}{\beta_-} \)
Gradient Descent with Momentum (Heavy Ball Method) for Strongly Convex Functions

\[ x_{t+1} = x_t - \mu \nabla f(x_t) + \beta (x_t - x_{t-1}) \]

• step-size parameter \( \mu = \frac{4}{(\sqrt{\beta_+} + \sqrt{\beta_-})^2} \)

• momentum parameter \( \beta = \max\left(|1 - \sqrt{\mu \beta_-}|, |1 - \sqrt{\mu \beta_+}|\right)^2 \)

• minimizing \( f(Ax) \) where \( A \in \mathbb{R}^{n \times d} \) via Gradient Descent with Momentum takes \( O(\sqrt{\kappa nd \log(\frac{1}{\epsilon})}) \) where \( \kappa = \frac{\beta_+}{\beta_-} \)
Newton’s Method

• Suppose $f$ is twice differentiable, and consider a second order Taylor approximation at a point $x_t$

\[
f(y) \approx f(x_t) + \nabla f(x_t)^T(y - x_t) + \frac{1}{2}(y - x^t)\nabla^2 f(x^t)(y - x^t)
\]

• minimizing the approximation yields $x_{t+1} = x_t - (\nabla^2 f(x))^\inv \nabla f(x)$

• Damped Newton updates: $x_{t+1} = x_t - t\Delta_t$ where
  \[\Delta_t := (\nabla^2 f(x))^{-1} \nabla f(x),\] where $t$ is a step-size parameter

• Hessian of $f(Ax)$ where $A \in \mathbb{R}^{n \times d}$ takes $O(n d^2)$ operations to calculate and $O(d^3)$ to invert. Alternatively, we can factorize in $O(nd^2)$ time (QR, Cholesky, SVD)
Choosing step-sizes: backtracking (Armijo) line search

given a descent direction $\Delta x$ for $f$ at $x \in \text{dom } f$, $\alpha \in (0, 0.5)$, $\beta \in (0, 1)$.

$t := 1.$
while $f(x + t\Delta x) > f(x) + \alpha t \nabla f(x)^T \Delta x$,  \[ t := \beta t. \]
Newton’s Method with Line Search

given a starting point \( x \in \text{dom}\, f \), tolerance \( \epsilon > 0 \).

repeat
1. Compute the Newton step and decrement.
   \[
   \Delta x_{nt} := -\nabla^2 f(x)^{-1} \nabla f(x); \quad \lambda^2 := \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x).
   \]
2. Stopping criterion. \text{quit} if \( \lambda^2 / 2 \leq \epsilon \).
3. Line search. Choose step size \( t \) by backtracking line search.
4. Update. \( x := x + t\Delta x_{nt} \).
Newton’s Method for Strongly Convex Functions

• Strong convexity with parameters \( \beta_-, \beta_+ \)

• Lipschitz continuity of the Hessian

\[ \| \nabla^2 f(x) - \nabla^2 f(y) \|_2 \leq L \| x - y \|_2^2 \]

for some constant \( L > 0 \)

• **Basic convergence result:** The number of iterations for \( \epsilon \) approximate solution in objective value is bounded by

\[ T := \text{constant} \times \frac{f(x_0) - f^*}{\beta_-/\beta_+^2} + \log_2 \log_2 \left( \frac{\epsilon_0}{\epsilon} \right) \]

where \( \epsilon_0 = 2\beta_-^3/L^2 \). Computational complexity: \( O((nd^2 + nd)T) \)
Affine Invariance of Newton’s Method

• The previous analysis can be improved

• The key insight is that Newton’s Method is invariant under linear transformations

• Newton’s Method for $f(x)$ is $x_{t+1} = x_t - (\nabla^2 f(x))^{-1} \nabla f(x)$

• Consider a linear invertible transformation $y = Ax$ and $g(y) = f(A^{-1}y)$. Then Newton’s Method for $g(y)$ is given by

$$
y_{t+1} = y_t - (\nabla^2 g(y_t))^{-1} \nabla g(y_t)
= Ax_t - (A^{-T} \nabla^2 f(x_t) A^{-1})^{-1} A^{-T} \nabla f(x)
= Ax_t - A \nabla^2 f(x_t)^{-1} \nabla f(x_t) = Ax_{t+1}
$$
Self-concordant Functions in $\mathbb{R}$

- A function $f : \mathbb{R} \rightarrow \mathbb{R}$ is self-concordant when $f$ is convex and

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

for all $x$ in the domain of $f$.

- Examples: linear and quadratic functions, negative logarithm

- One can use a constant $k$ other than 2 in the definition. The number 2 is used in the definition so that $-\log(x)$ is self-concordant
Self-concordant Functions in $\mathbb{R}^d$

- A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is self-concordant when it is self-concordant along every line, i.e.,
  
  (i) $f$ is convex
  (ii) $g(t) := f(x + tv)$ is self-concordant for all $x$ in the domain of $f$ and all $v$
Self-concordant Functions in $\mathbb{R}^d$

• Scaling with a positive factor of at least 1 preserves self-concordance:

$$f \text{ is self concordant} \implies \alpha f \text{ is self concordant for } \alpha \geq 1$$

• Addition preserves self-concordance

$$f_1 \text{ and } f_2 \text{ is self concordant} \implies f_1 + f_2 \text{ is self concordant}$$

• if $f(x)$ is self-concordant, affine transformations $g(x) := f(Ax + b)$ are also self-concordant

• $x^T Ax + b^T x$, $-\log(x)$ and $-\log \det(X)$ are self-concordant functions
Newton’s Method for Self-concordant Functions

• Suppose $f$ is a self-concordant function

• Theorem

Newton’s method with line search finds an $\epsilon$ approximate point in less than

$$T := \text{constant} \times (f(x_0) - f^*) + \log_2 \log_2 \frac{1}{\epsilon}$$

iterations.

• Computational complexity: $T \times (\text{cost of Newton Step})$
due to Nesterov and Nemirovski
Interior Point Programming

- Logarithmic Barrier Method

Goal:

\[
\min_x f_0(x) \text{ s.t. } f_i(x) \leq 0, \ i = 1, \ldots, n
\]

Indicator penalized form

\[
\min_x f_0(x) + \sum_{i=1}^{n} \mathbb{I}(f_i(x))
\]

where \( \mathbb{I} \) is a \( \{0, \infty\} \) valued indicator function
• Approximate the indicator via \( -\frac{\log(-f_i(x))}{t} \)

\[
x^*(t) = \arg\min_x f_0(x) - \frac{1}{t} \sum_{i=1}^n \log(-f_i(x))
\]

\[
= \arg\min_x tf_0(x) - \sum_{i=1}^n \log(-f_i(x))
\]

• \( t > 0 \) is the barrier parameter

• \( x^*(t), t > 0 \) is called the *central path*
Interior Point Programming
Example: Linear Programming

• LP in standard form where $A \in \mathbb{R}^{n \times d}$

$$
\min_{Ax \leq b} c^T x
$$

• Central path

$$
\arg \min_x t c^T x - \sum_{i=1}^{n} \log(b_i - a_i^T x)
$$

• self-concordant function

• Hessian $\nabla^2 f(x) = A^T \text{diag} \left( \frac{1}{(b_i - a_i^T x)^2} \right) A$ takes $O(nd^2)$ operations
Barrier Method for Constrained Convex Programs

\[ p^* = \min f_0(x) \text{ s.t. } f_i(x) \leq 0, \ i = 1, \ldots, n \]

Suppose that \( f_0, f_1, \ldots, f_n \) are twice differentiable. Define

\[ x^*(t) := \min_x t f_0(x) - \sum_{i=1}^n \log(-f_i(x)) \]

1. Centering step. Compute \( x^*(t) \) via Newton’s Method starting at \( x \)

2. Update \( x := x^*(t) \)

3. Stopping criterion. quit if \( n/t < \epsilon \)

4. Increase \( t. \ t := \mu t \)
Central path for an LP
Other Self-concordant (sc) Barrier Functions

- $-\log \det X$ is an sc barrier for the positive semidefinite cone

- $-\log(x^T Ax + b^T x + c)$ is an sc barrier for the convex set $x^T Ax + b^T x + c > 0$ when $A \succeq 0$

- $-\log(y^2 - x^x)$ is an sc barrier for the second order cone $\|x\|_2 \leq y$
Barrier Method for Constrained Convex Programs

- terminates with $f_0(x^*(t)) - p^* \leq \epsilon$

- choice of $\mu$ involves a trade-off: large $\mu$ means fewer outer iterations, more inner (Newton) iterations. Typical values of $\mu = 10 - 20$
Optimality gap of the central path

- Central path $x^*(t) = \arg \min_x t f_0(x) - \sum_{i=1}^{n} \log(-f_i(x))$

- Optimality conditions $x^*(t)$ (necessary and sufficient)

$$t \nabla f_0(x^*) + \sum_{i=1}^{n} \frac{1}{-f_i(x^*(t))} \nabla f_i(x^*(t)) = 0$$

- $x^*(t)$ minimizes the Lagrangian for the original problem for

$$\lambda = -\frac{1}{tf_i(x^*(t))}$$

$$\nabla_x L(x, \lambda) = \nabla f_0(x) + \sum_{i=1}^{n} \lambda_i \nabla f_i(x) = 0$$
• \( \lambda^*(t) = -\frac{1}{tf_i(x^*(t))} > 0 \) is dual feasible and provides a lower-bound

\[
\min_{x \text{ s.t. } f_i(x) \leq 0 \forall i} f_0(x) \geq \max_{\lambda \geq 0} \min_x f_0(x) + \sum_{i=1}^{n} \lambda_i f_i(x) \\
\geq \min_x f_0(x) + \sum_{i=1}^{n} \lambda^*(t) f_i(x) \\
= f_0(x^*(t)) + \sum_{i=1}^{n} \lambda^*(t) f_i(x^*(t)) \\
= f_0(x^*(t)) - \sum_{i=1}^{n} \frac{f_i(x^*(t))}{tf_i(x^*(t))} = f_0(x^*(t)) - \frac{n}{t}
\]

Therefore optimality gap is at most \( n/t \)
Complexity Analysis: Number of centering steps

• Assuming that we can find $x^*(t) = \arg\min_x t f_0(x) - \sum_{i=1}^n \log(-f_i(x))$ via Newton’s method for $t = t^0, \mu t^0, \mu^2 t^0, \ldots$, the optimality gap after $k$ centering steps is $\frac{n}{\mu^k t^0}$

• Accuracy $\epsilon$ is achieved after

$$\log \left( \frac{m}{(\epsilon t^0)} \right) \log \mu$$

centering steps, plus the initial centering step
Complexity Analysis: Number of Newton Iterations

• Number of Newton iterations per centering step is bounded by

\[ T := \text{constant} \times (f(x_0) - \min_x f(x)) + \log_2 \log_2 \frac{1}{\epsilon} \]

• Bound on the effort of computing \( x^*(\mu t) \) starting at \( x = x^*(t) \) depends on the initial optimality gap \( f(x_0) - \min f(x) \) where

\[ f(x) := t f_0(x) + \sum_{i=1}^{n} \log(-f_i(x)) \]

• it can be shown that (see Chapter 11.5 in Convex Optimization)

\[ T \leq \text{constant} \times \frac{n(\mu - 1 - \log \mu)}{\gamma} + \log_2 \log_2 \frac{1}{\epsilon} \]
• number of outer (centering) iterations is \( \frac{\log(n/(\epsilon t(0)))}{\log \mu} \)

• total number of Newton iterations \( N := \frac{\log(n/(\epsilon t(0)))}{\log \mu} n(\mu - 1 - \log \mu) \gamma \)

• confirms the trade-off in the choice of \( \mu \)

• for \( \mu = 1 + 1/\sqrt{n} \), total number of Newton iterations
\[
N = O(\sqrt{n} \log \left(\frac{n/t(0)}{\epsilon}\right))
\]

• this proves the polynomial-time complexity of barrier method for convex programming

• this choice of \( \mu \) optimizes worst-case complexity. In practice we choose \( \mu \) fixed, e.g., \( \mu = 10, \ldots, 20 \). The number of outer iterations is in the tens and not very sensitive for \( \mu \geq 10 \).
Numerical Example

• We solve a Second Order Cone Program

\[
\begin{align*}
    \text{min } & \quad f^T x \\
    \text{s.t. } & \quad \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \ i = 1, \ldots, n
\end{align*}
\]

using the sc barrier \(- \sum_{i=1}^{n} \log \left( (c_i^T x + d_i)^2 - \|A_i x + b\|_2^2 \right)\)

• The central path is given by

\[
x^*(t) = \arg \min_{x} t f^T x - \sum_{i=1}^{n} \log \left( (c_i^T x + d_i)^2 - \|A_i x + b\|_2^2 \right)
\]
Numerical Example

- Randomly generated problem instances where $n = 50$ and $x \in \mathbb{R}^{50}$
Reformulating Non-differentiable Objectives

• Example: Robust regression

\[
\min_x \|Ax - b\|_1
\]

• Reformulation

\[
\min_{x,y} \|y\|_1 = \min_{x,y,s} 1^T s
\]

\[
\text{s.t. } Ax - b = y \quad \text{s.t. } -s_i \leq y_i \leq s_i \forall i
\]

\[
Ax - b = y
\]
Conclusions

• Interior Point (barrier) methods run in provably polynomial-time for convex optimization when we have self-concordant barriers

• They are also very efficient in practice

• Main computational load is solving 20-30 linear systems for the Newton iterations

• There are also primal-dual interior methods which are more efficient