CS 357: Numerical Methods

Constrained Optimization

Eric Shaffer
Optimization Problems

Given function $f : \mathbb{R}^n \to \mathbb{R}$, and set $S \subseteq \mathbb{R}^n$, find $x^* \in S$ such that $f(x^*) \leq f(x)$ for all $x \in S$

$x^*$ is called minimizer or minimum of $f$

It suffices to consider only minimization, since maximum of $f$ is minimum of $-f$

Objective function $f$ is usually differentiable, and may be linear or nonlinear

Constraint set $S$ is defined by system of equations and inequalities, which may be linear or nonlinear

Points $x \in S$ are called feasible points

If $S = \mathbb{R}^n$, problem is unconstrained
Optimization Problems

- General continuous optimization problem:

\[
\min f(x) \quad \text{subject to} \quad g(x) = 0 \quad \text{and} \quad h(x) \leq 0
\]

where \( f: \mathbb{R}^n \rightarrow \mathbb{R}, \ g: \mathbb{R}^n \rightarrow \mathbb{R}^m, \ h: \mathbb{R}^n \rightarrow \mathbb{R}^p \)

- Linear programming: \( f, g, \) and \( h \) are all linear

- Nonlinear programming: at least one of \( f, g, \) and \( h \) is nonlinear
Newton’s Method (for optimization)

- Another local quadratic approximation is truncated Taylor series
  \[ f(x + h) \approx f(x) + f'(x)h + \frac{f''(x)}{2}h^2 \]

- By differentiation, minimum of this quadratic function of \( h \) is given by \( h = -\frac{f'(x)}{f''(x)} \)

- Suggests iteration scheme
  \[ x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} \]

  which is *Newton’s method* for solving nonlinear equation \( f'(x) = 0 \)

- Newton’s method for finding minimum normally has quadratic convergence rate, but must be started close enough to the solution.
Newton’s Method: Example

- Use Newton’s method to minimize \( f(x) = 0.5 - x \exp(-x^2) \)
- First and second derivatives of \( f \) are given by
  \[
  f'(x) = (2x^2 - 1) \exp(-x^2)
  \]
  and
  \[
  f''(x) = 2x(3 - 2x^2) \exp(-x^2)
  \]
- Newton iteration for zero of \( f' \) is given by
  \[
  x_{k+1} = x_k - \frac{(2x_k^2 - 1)}{(2x_k(3 - 2x_k^2))}
  \]
- Using starting guess \( x_0 = 1 \), we obtain
  \[
  \begin{array}{c|c}
    x_k & f(x_k) \\
    \hline
    1.000 & 0.132 \\
    0.500 & 0.111 \\
    0.700 & 0.071 \\
    0.707 & 0.071
  \end{array}
  \]
Newton’s Method: Example

- Use Newton’s method to minimize \( f(x) = 0.5 - x \exp(-x^2) \)
- First and second derivatives of \( f \) are given by
  \[
  f'(x) = (2x^2 - 1) \exp(-x^2)
  \]
  and
  \[
  f''(x) = 2x(3 - 2x^2) \exp(-x^2)
  \]
- Newton iteration for zero of \( f' \) is given by
  \[
  x_{k+1} = x_k - \frac{(2x_k^2 - 1)}{(2x_k(3 - 2x_k^2))}
  \]
- Using starting guess \( x_0 = 1 \), we obtain

<table>
<thead>
<tr>
<th>( x_k )</th>
<th>( f(x_k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000</td>
<td>0.132</td>
</tr>
<tr>
<td>0.500</td>
<td>0.111</td>
</tr>
<tr>
<td>0.700</td>
<td>0.071</td>
</tr>
<tr>
<td>0.707</td>
<td>0.071</td>
</tr>
</tbody>
</table>
Steepest Descent

- Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be real-valued function of $n$ real variables
- At any point $x$ where gradient vector is nonzero, negative gradient, $-\nabla f(x)$, points downhill toward lower values of $f$
- In fact, $-\nabla f(x)$ is locally direction of steepest descent: $f$ decreases more rapidly along direction of negative gradient than along any other
- *Steepest descent* method: starting from initial guess $x_0$, successive approximate solutions given by
  \[ x_{k+1} = x_k - \alpha_k \nabla f(x_k) \]
  where $\alpha_k$ is *line search* parameter that determines how far to go in given direction
Steepest Descent

- Given descent direction, such as negative gradient, determining appropriate value for \( \alpha_k \) at each iteration is one-dimensional minimization problem

\[
\min_{\alpha_k} f(x_k - \alpha_k \nabla f(x_k))
\]

that can be solved by methods already discussed

- Steepest descent method is very reliable: it can always make progress provided gradient is nonzero

- But method is myopic in its view of function's behavior, and resulting iterates can zigzag back and forth, making very slow progress toward solution

- In general, convergence rate of steepest descent is only linear, with constant factor that can be arbitrarily close to 1
Steepest Descent: Example

- Use steepest descent method to minimize
  \[ f(x) = 0.5x_1^2 + 2.5x_2^2 \]

- Gradient is given by \( \nabla f(x) = \begin{bmatrix} x_1 \\ 5x_2 \end{bmatrix} \)

- Taking \( x_0 = \begin{bmatrix} 5 \\ 1 \end{bmatrix} \), we have \( \nabla f(x_0) = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \)

- Performing line search along negative gradient direction,
  \[ \min_{\alpha_0} f(x_0 - \alpha_0 \nabla f(x_0)) \]

  exact minimum along line is given by \( \alpha_0 = 1/3 \), so next approximation is
  \( x_1 = \begin{bmatrix} 3.333 \\ -0.667 \end{bmatrix} \)
Steepest Descent: Example

\[ x_1 \]
\[ x_2 \]

<table>
<thead>
<tr>
<th>( x_k )</th>
<th>( f(x_k) )</th>
<th>( \nabla f(x_k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.000</td>
<td>1.000</td>
<td>15.000 5.000 5.000</td>
</tr>
<tr>
<td>3.333</td>
<td>0.667</td>
<td>6.667 3.333 3.333</td>
</tr>
<tr>
<td>2.222</td>
<td>0.444</td>
<td>2.963 2.222 2.222</td>
</tr>
<tr>
<td>1.481</td>
<td>-0.296</td>
<td>1.317 1.481 -1.481</td>
</tr>
<tr>
<td>0.988</td>
<td>0.198</td>
<td>0.585 0.988 0.988</td>
</tr>
<tr>
<td>0.658</td>
<td>-0.132</td>
<td>0.260 0.658 -0.658</td>
</tr>
<tr>
<td>0.439</td>
<td>0.088</td>
<td>0.116 0.439 0.439</td>
</tr>
<tr>
<td>0.293</td>
<td>-0.059</td>
<td>0.051 0.293 -0.293</td>
</tr>
<tr>
<td>0.195</td>
<td>0.039</td>
<td>0.023 0.195 0.195</td>
</tr>
<tr>
<td>0.130</td>
<td>-0.026</td>
<td>0.010 0.130 -0.130</td>
</tr>
</tbody>
</table>
Multi-Dimensional Optimization: Newton’s Method

- Broader view can be obtained by local quadratic approximation, which is equivalent to Newton’s method

- In multidimensional optimization, we seek zero of gradient, so *Newton iteration* has form

\[ \mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_f^{-1}(\mathbf{x}_k) \nabla f(\mathbf{x}_k) \]

where \( \mathbf{H}_f(\mathbf{x}) \) is *Hessian* matrix of second partial derivatives of \( f \),

\[
\{ \mathbf{H}_f(\mathbf{x}) \}_{ij} = \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}
\]
Multi-Dimensional Optimization: Newton’s Method

- Do not explicitly invert Hessian matrix, but instead solve linear system

\[
H_f(x_k) s_k = -\nabla f(x_k)
\]

for Newton step \( s_k \), then take as next iterate

\[
x_{k+1} = x_k + s_k
\]

- Convergence rate of Newton’s method for minimization is normally quadratic

- As usual, Newton’s method is unreliable unless started close enough to solution to converge
Multi-Dimensional Optimization: Newton’s Method

- Use Newton’s method to minimize
  \[ f(x) = 0.5x_1^2 + 2.5x_2^2 \]

- Gradient and Hessian are given by
  \[ \nabla f(x) = \begin{bmatrix} x_1 \\ 5x_2 \end{bmatrix} \quad \text{and} \quad H_f(x) = \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix} \]

- Taking \( x_0 = \begin{bmatrix} 5 \\ 1 \end{bmatrix} \), we have \( \nabla f(x_0) = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \)

- Linear system for Newton step is \[ \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix} s_0 = \begin{bmatrix} -5 \\ -5 \end{bmatrix}, \] so

  \[ x_1 - x_0 + s_0 = \begin{bmatrix} 5 \\ 1 \end{bmatrix} + \begin{bmatrix} -5 \\ -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \] which is exact solution for this problem, as expected for quadratic function.
Constrained Optimization: Example
First-Order Optimality Condition

- For function of one variable, one can find extremum by differentiating function and setting derivative to zero.

- Generalization to function of \( n \) variables is to find critical point, i.e., solution of nonlinear system

\[
\nabla f(x) = 0
\]

where \( \nabla f(x) \) is gradient vector of \( f \), whose \( i \)th component is \( \partial f(x)/\partial x_i \).

- For continuously differentiable \( f: S \subseteq \mathbb{R}^n \to \mathbb{R} \), any interior point \( x^* \) of \( S \) at which \( f \) has local minimum must be critical point of \( f \).

- But not all critical points are minima: they can also be maxima or saddle points.
Second-Order Optimality Condition

- For twice continuously differentiable \( f : S \subseteq \mathbb{R}^n \rightarrow \mathbb{R} \), we can distinguish among critical points by considering the **Hessian matrix** \( H_f(x) \) defined by

\[
\{H_f(x)\}_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}
\]

which is symmetric.

- At critical point \( x^* \), if \( H_f(x^*) \) is
  - positive definite, then \( x^* \) is minimum of \( f \)
  - negative definite, then \( x^* \) is maximum of \( f \)
  - indefinite, then \( x^* \) is saddle point of \( f \)
  - singular, then various pathological situations are possible
Constrained Optimality

- If problem is constrained, only feasible directions are relevant
- For equality-constrained problem

\[
\min f(x) \quad \text{subject to} \quad g(x) = 0
\]

where \( f : \mathbb{R}^n \to \mathbb{R} \) and \( g : \mathbb{R}^n \to \mathbb{R}^m \), with \( m \leq n \), necessary condition for feasible point \( x^* \) to be solution is that negative gradient of \( f \) lie in space spanned by constraint normals,

\[
-\nabla f(x^*) = J_g^T(x^*) \lambda
\]

where \( J_g \) is Jacobian matrix of \( g \), and \( \lambda \) is vector of Lagrange multipliers

- This condition says we cannot reduce objective function without violating constraints
Constrained Optimality

- **Lagrangian function** $\mathcal{L}: \mathbb{R}^{n+m} \rightarrow \mathbb{R}$, is defined by
  \[ \mathcal{L}(x, \lambda) = f(x) + \lambda^T g(x) \]

- Its gradient is given by
  \[ \nabla \mathcal{L}(x, \lambda) = \begin{bmatrix} \nabla f(x) + J_g^T(x) \lambda \\ g(x) \end{bmatrix} \]

- Its Hessian is given by
  \[ H_{\mathcal{L}}(x, \lambda) = \begin{bmatrix} B(x, \lambda) & J_g^T(x) \\ J_g(x) & O \end{bmatrix} \]

  where
  \[ B(x, \lambda) = H_f(x) + \sum_{i=1}^{m} \lambda_i H_{g_i}(x) \]
Constrained Optimality

- Together, necessary condition and feasibility imply critical point of Lagrangian function,

\[ \nabla \mathcal{L}(x, \lambda) = \begin{bmatrix} \nabla f(x) + J_g^T(x)\lambda \\ g(x) \end{bmatrix} = 0 \]

- Hessian of Lagrangian is symmetric, but not positive definite, so critical point of \( \mathcal{L} \) is saddle point rather than minimum or maximum

- Critical point \((x^*, \lambda^*)\) of \( \mathcal{L} \) is constrained minimum of \( f \) if \( B(x^*, \lambda^*) \) is positive definite on null space of \( J_g(x^*) \)

- If columns of \( Z \) form basis for null space, then test projected Hessian \( Z^T B Z \) for positive definiteness
Constrained Optimization: Example
Constrained Optimization: Example