Computational Methods

Optimization

Unconstrained Optimization
Optimization

- Optimization problems are concerned with finding the minimum or maximum of an objective function.
  - Find \( x^* \) such that \( f(x^*) \leq f(x) \) for all \( x \) in \( S \)
    - Maximization of \( f(x) \) is the same as minimization of \( -f(x) \)
  - Least squares problem is a special case where the function to be minimized is the residual.

- Optimization problems can also include a set of constraints that limit the set of feasible points, \( S \).
  - Unconstrained optimization does not have any constraints.
  - Equality constraints are of the form \( g(x) = 0 \).
  - Inequality constraints are of the form \( h(x) \leq 0 \).
Optimization

- General continuous optimization problem is defined by the objective function and the constraints

  Find \( \min f(x) \)
  Subject to \( g_i(x) = 0 \) and \( h_j(x) \leq 0 \)

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

- Linear programming characterizes optimization problems where the objective function and the constraints are linear

- Nonlinear programming characterizes optimization problems where at least one of the constraints or the objective function are nonlinear
Global vs. Local Optimization

- A global optimum is a point that is an optimum for all feasible points (points matching the constraints).
- A local optimum is a point that is an optimum only for the feasible points in some neighborhood around the point.
Global vs. Local Optimization

- Global optimization is in general a very difficult problem and existing methods for global optimality are limited to specific function types
  - Even verifying that an optimum is a global optimum is very difficult in general

- Most optimization methods are designed to find local optima
  - To increase the chance of finding global optima, local optimization methods can be run multiple times from different starting points
Existence of Solution

- Not every function has an optimum
  - E.g. Polynomials of odd order do not have global optima since they tend to $\pm\infty$ for $x$ towards $\pm\infty$

- Some conditions exist under which the existence of a global minimum can be ensured
  - If objective function $f(x)$ is continuous on a closed and bounded set $S$ of feasible points, then $f(x)$ has a global minimum on $S$
  - If $f(x)$ is coercive on a closed, unbounded set $S$ of feasible points, then it has a global minimum on $S$
    - Coercive: $\lim_{\|x\| \to \infty} f(x) = \infty$
Uniqueness of Solution

- If function $f(x)$ is convex on the set of feasible points, $S$, then any minimum on $S$ is a global minimum on $S$
  - In a convex function every minimum must have the same function value (all minima form a “plateau”)
- If function $f(x)$ is strictly convex on the set of feasible points, $S$, then it has a unique global minimum on $S$
  - A strictly convex function can only have one minimum
Optimality Conditions

- First-Order optimality condition
  - Any extremum of a continuous, differentiable function $f(x)$ has to be either on the boundary of the feasible set or a critical point, i.e. a solution to the nonlinear system
  $$\nabla f(x) = 0$$
  - Not every critical point is an extrema (e.g. saddle points)

- Second-Order optimality condition
  - If $f(x)$ is twice differentiable, the Hessian matrix (matrix of partial second derivatives) permits to identify extrema
    - Critical point $x^*$ is a minimum if $H_f(x^*)$ is positive definite
    - Critical point $x^*$ is a maximum if $H_f(x^*)$ is negative definite
Sensitivity and Conditioning

- Sensitivity analysis for the scalar case using Taylor series expansion
  
  - Absolute forward error: \( \Delta x = \hat{x} - x^* \)
  
  - Absolute backward error:
    \[
    f(x^* + \Delta x) = f(x^*) + f'(x^*)\Delta x + \frac{1}{2} f''(x^*)\Delta x^2 + O(\Delta x^3)
    \approx f(x^*) + \frac{1}{2} f''(x^*)\Delta x^2
    \]
    \[
    f(\hat{x}) - f(x^*) \approx \frac{1}{2} f''(x^*)\Delta x^2
    \]
  
  - Sensitivity of optimization:
    \[
    |\Delta x| \approx \sqrt{2} \frac{|f(\hat{x}) - f(x^*)|}{|f''(x^*)|}
    \]
Unconstrained Optimization

- Unconstrained optimization has many similarities to the problem of solving equations and solution methods are similar
  - Direct search methods iteratively narrow down the neighborhood of the solution (like bisection method for equation solving
  - Iterated approximation methods use a fixed-point formulation and the derivative (or an approximation of it) to achieve the solution
Direct Search Methods for One-Dimensional Optimization

- Golden Section search iteratively narrows down the interval within which the solution has to exist
  - To ensure existence of the solution within the interval, the function is assumed to be unimodal in the interval
    - $f(x)$ is unimodal in an interval if it has a minimum, $x^*$, in the interval and is strictly increasing in both directions from this point
    - Any continuous, twice differentiable function has a (potentially small) interval around each minimum for which the function is unimodal
  - To divide the interval two points, $x_1$, $x_2$, within the interval are used and their function values indicate which end of the interval can be discarded
    - The side of the point with the higher function value is discarded
Golden Section Search

- Golden Section achieves a reduction of the interval by a constant factor and requires only one function evaluation in each iteration by carefully choosing the locations of the interior points.

- Interior points in interval \([a,b]\) are chosen at

\[
\begin{align*}
x_1 &= a + \frac{3 - \sqrt{5}}{2} (b - a) \quad \text{, } x_2 = a + \frac{\sqrt{5} - 1}{2} (b - a)
\end{align*}
\]

- If one side of the interval is discarded the other point stays at a correct location in the new interval. E.g. if left is discarded:

\[
\begin{align*}
x_2 &= a + \frac{\sqrt{5} - 1}{2} (b - a) = x_1 + \frac{2\sqrt{5} - 4}{2} \frac{1}{1 - (3 - \sqrt{5})/2} (b - x_i) = x_1 + \frac{(2\sqrt{5} - 4)2}{2(-1 + \sqrt{5})} (b - x_i) \\
&= x_1 + \frac{-3 + 4\sqrt{5} - 5}{2(-1 + \sqrt{5})} (b - x_i) = x_1 + \frac{(3 - \sqrt{5})(-1 + \sqrt{5})}{2(-1 + \sqrt{5})} (b - x_i) = x_1 + \frac{3 - \sqrt{5}}{2} (b - x_i)
\end{align*}
\]
Example

- Golden Section search on $f(x) = 0.5 - xe^{-x^2}$ and initial interval $[0,2]$
Successive Parabolic Interpolation

- Golden Section search is safe but converges only at a linear rate with constant 0.618
- Successive parabolic interpolation uses only one point in the interval, calculating the next (and which interval point to remove)
  - Take 3 points and their function values and interpolate them using a parabola
    - Minimum of parabola is added as a new point
    - Oldest of the points is dropped
  - Achieves superlinear convergence with $r \approx 1.324$
Newton’s Method

- As in the case of solving nonlinear equations, a better convergence rate can be achieved using information about the function

\[ f(x + \Delta x) \approx f(x) + f'(x)\Delta x + \frac{f''(x)}{2} \Delta x^2 \]

- Necessary first-order condition yields

\[ \frac{\partial f(x + \Delta x)}{\partial \Delta x} = f'(x) + f''(x)\Delta x = 0 \quad \Rightarrow \quad \Delta x = -\frac{f'(x)}{f''(x)} \]

- Yields Newton’s method for \( f'(x) = 0 \)

\[ x_{t+1} = x_t - \frac{f'(x_t)}{f''(x_t)} \]

- Has quadratic convergence rate and converges if started close enough to the solution
Safeguard Methods

- As in nonlinear equation solving there are interval search methods that are guaranteed to converge slowly and methods with higher convergence rates but without guarantees that they will converge.

- Safeguard methods combine multiple methods to achieve both guaranteed convergence and a good convergence rate.
  - Golden selection search and successive parabolic interpolation can be combined if no derivatives are available.
  - Golden selection search and Newton’s method can be combined if derivatives are available.
Multi-Dimensional Optimization

- As for the one-dimensional case, two basic methods for optimization of multi-dimensional functions exist
  - Direct search methods
  - Iterative descent methods
- Direct search methods for multi-dimensional data introduce additional problems
  - Definition of an equivalent to a bracket is not easily possible in the multi-dimensional case
Nelder-Mead Method

- Nelder-Mead is a direct search method for the multi-variate case
  - Does not use a “bracket” but a simplex with \( n+1 \) points for a function with \( n \) variables
    - Points of the simplex form the points of interest defining a region
      - No guarantee that the solution lies within this region
    - Next search volume can lie partially outside the previous one
  - Next simplex is created by replacing the worst point of the existing simplex with a new point
    - New point is improved point on the line connecting the old point and the centroid of the remaining points in the simplex
    - If no better point is found, simplex is shrunk towards best point
Nelder-Mead Method

- Nelder-Mead does not require any information about the function to be minimized
  - Only evaluation of the function is necessary
  - Convergence is only guaranteed if the function within the simplex region has a unique minimum
- Operations change location and shape of simplex
  - Reflection: mirrors simplex away from worst point
  - Expansion: expands simplex in direction of new point
  - Contraction: contracts simplex away from worst point
  - Reduction: shrinks simplex around best point
Nelder-Mead Method

- Variations of the algorithm exists which apply slightly different rules to select operation
- A common set of rules is:
  - Precomputation:
    - Sort the vertices of the simplex by function value \( f(x_i) \)
    - Compute centroid \( x_c \) of the best \( n \) vertices
  - Start by applying reflection to worst vertex to get reflected vertex \( x_r \)
    \[
    x_r = x_c + \alpha(x_c - x_{n+1})
    \]
    - Often \( \alpha = 1 \)
    - If reflected vertex is not the worst of the remaining and not the best vertex, replace previously worst vertex with reflected vertex
Nelder-Mead Method

- Else, if reflected vertex is the best vertex apply expansion
  \[ x_e = x_c + \beta (x_c - x_{n+1}) \quad , \quad \text{often} \quad \beta = 2 \]
  - If expansion point is better than the reflected point, replace the worst point with the expanded point
  - Else replace the worst point with the reflected point

- Else, if reflected point is the worst point apply contraction
  \[ x_o = x_c - \gamma (x_c - x_{n+1}) \quad , \quad \text{often} \quad \gamma = 1/2 \]
  - If contraction point is better than the original worst point, replace the worst point with the contraction point
  - Else, if the contraction point is no better than the original worst point apply reduction by shrinking all points towards the best point
  \[ x_i = x_1 + \eta (x_i - x_1) \quad , \quad \text{often} \quad \eta = 1/2 \]
Nelder-Mead Method

- Nelder-Mead can be applied to smooth and non-smooth functions
  - Does not need derivatives
- Computational cost of the algorithm increases fast as the number of variables increases
- No guaranteed convergence
  - The choice of the initial simplex is essential to finding the desired minimum
Iterative Descent Methods

- When derivative information of the function is available, this information can be used to accelerate optimization on smooth functions
  - Steepest descent methods
    - Strictly follow the gradient direction of the function
  - Newton’s method
    - Take into account the derivative of the gradient (the Hessian)
  - Quasi-Newton methods
    - Use a local approximation of the Hessian to reduce computation and be potentially more robust
Steepest Descent with Line Search

- In steepest descent methods the direction of the update step for the iterative solution is always given by the negative gradient at the current point.

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

- Pick of step size \( \alpha \) is very important for convergence towards a solution

  - Line search can be used to determine the best \( \alpha \) for the current point

\[ \alpha_t = \arg \min_{\alpha} f(x_t - \alpha \nabla f(x_t)) \]

Line search can be solved a a one-dimensional minimization problem
Steepest Descent with Line Search

- Steepest descent with line search is very robust and reliable
  - Always makes progress
  - Convergence is only linear
    - Ignoring of second derivatives makes it inefficient
Newton’s Method

Rather than relying on the gradient alone (a first order approximation), Newton’s method again uses a local second order approximation.

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

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

To avoid the inversion, this can again be broken into a linear equation solution for the step size followed by an update

\[ H_f(x_t) s_t = -\nabla f(x_t) \]

\[ x_{t+1} = x_t + s_t \]
Newton’s Method

If it converges, Newton’s method converges faster towards the solution

- Quadratic convergence
- Convergence is assured only when started close enough to a solution
  - In principle a step size is no longer necessary
  - Convergence can be improved by adding line search in the direction of the Newton step to ensure decrease in every step (damped Newton)
    - Incurs significant additional complexity
Quasi-Newton Methods

- Newton’s method needs calculation of the Hessian and its inversion (solution of linear system).
  - $O(n^3)$ computational complexity
  - Requires knowledge of the Hessian
- Quasi-Newton methods use an approximation of the Hessian (similar to Boyden’s method)
  - $x_{t+1} = x_t - \alpha_t B_f^{-1}(x_t) \nabla f(x_t)$
  - Broyden–Fletcher–Goldfarb–Shannon (BFGS) Method
  - Conjugate Gradient Methods
BFGS Method

- Broyden–Fletcher–Goldfarb–Shannon (BFGS) is an extension of Broyden’s Method for equation solving that maintains symmetry of approximate Hessian

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

- The approximate Hessian is updated in each iteration starting with an initial estimate (often \( B_0=I \))

\[ B_t s_t = -\nabla f(x_t) \]

\[ x_{t+1} = x_t + s_t \]

\[ \Delta f'_{t+1} = \nabla f(x_{t+1}) - \nabla f(x_t) \]

\[ B_{t+1} = B_t + \left( \Delta f'_{t+1}^T \Delta f'_{t+1} \right) / \left( \Delta f'_{t}^T s_t \right) - \left( B_t s_t s_t^T B_t^T \right) \left( s_t^T B_t s_t \right) \]

- Sherman–Morrison formula can be used to avoid inversion
BFGS Method

- BFGS method does not require second derivatives and is computationally much less expensive
  - Methods can be used to directly update factorization of $B$, making the method $O(n^2)$
  - Converges superlinearly
  - More robust than Newton’s method
- Line search can be used to add a step size
  - Can increase the convergence radius for BFGS
    - Incurs additional cost
Conjugate Gradient Method

- The Conjugate gradient method further simplifies the approximation of the Hessian by explicitly estimating its effect on the gradient

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

\[ s_t = \nabla f(x_t) - \beta_t s_{t-1} \]

\[ \beta_t = \frac{(\nabla f(x_t)^T \nabla f(x_t))}{(\nabla f(x_{t-1})^T \nabla f(x_{t-1}))} \]

- Conjugate gradient is exact for a quadratic objective function after at most n iterations
  - Also works usually well for general unconstrained optimization
- The step parameter can be formed using line search
Unconstrained Optimization

- Unconstrained Optimization allows to find the best parameters for arbitrary objective functions
  - Least squares is a special case of unconstrained optimization
- Two basic approaches exist
  - Direct search techniques
  - Iterative improvement algorithms
    - Newton’s method if gradient and Hessian information is available
    - Quasi-Newton methods if no such information is to be used.