Computational Methods

Constrained Optimization

Constrained Optimization

- Unconstrained Optimization finds a minimum of a function under the assumption that the parameters can take on any possible value.
- In a range of problems additional constraints exist that limit the range of feasible parameters.
  - Unconstrained optimization techniques would often find solutions that are not feasible (parameters do not fulfill constraints)
  - Simply limiting iterative methods at the constraints will often lead to suboptimal solutions

Constrained Optimization

- Constrained optimization problems can be defined using an objective function and a set of constraints.
  - Objective function: $\min_x f(x)$
  - Equality constraints: $g_i(x)=0$
  - Inequality constraints: $h_i(x)\leq 0$
- A feasible point is any point that fulfills all the constraints.
- An optimal point is one that locally optimizes the value function given the constraints.

Constrained Optimization

- A number of special cases for constrained optimization problems can be defined based on the types of functions used
  - Linear programming:
    - Linear objective function: $f(x)=\alpha^T x$
    - Linear constraints: $g(x)=Bx-c$; $h(x)=Cx-d$
  - Quadratic programming:
    - Quadratic objective function: $f(x)=1/2\ x^T A x + \alpha^T x$
    - Linear constraints: $g(x)=Bx-c$; $h(x)=Cx-d$

Constrained Optimization

- Solving constrained optimization problems is substantially harder than solving unconstrained optimization problems
  - If possible, convert the constrained optimization problem into an unconstrained optimization problem
    - For constraints on individual parameters, replace them with differentiable functions of an unconstrained parameter

$$a \le x_i \le b \quad \Rightarrow \quad x_i = \frac{(b-a)}{1+e^{-\tilde{x}_i}} + a$$

$$x_i \le a \quad \Rightarrow \quad x_i = e^{\tilde{x}_i} + a$$

$$x_i + \alpha x_j = 0 \quad \Rightarrow \quad x_i = -\alpha x_j$$

Constrained Optimization

- Another differentiation can be made in terms of the types of constraints present
  - Optimization problems with only equality constraints
  - Optimization problems with inequality constraints
- Equality constraints are generally easier to deal with than inequality constraints

Problems with Equality Constraints

- A problem with only equality constraints is defined through and objective function *f(x)* and a set of constraints, *g(x)=0*
  - Feasible points have to fulfill the constraints
  - Constrained optimal points have to fulfill the necessary condition that the negative gradient of the objective function can not have any component that falls within the space spanned by the constraints
    - Otherwise there would be a way to lower the objective function without violating the constraints

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

Problems with Equality Constraints

- Lagrange multipliers λ introduce the equality constraints into the optimality condition
  - Instead of optimizing *f(x)* optimize the Lagrangian function *L(x, λ)*

$$L(x,\lambda) = f(x) + \lambda^T g(x)$$

    - At the constrained minimum the Lagrangian has a minimum and therefore constrained optimization solution has to be a solution to the nonlinear system

$$\nabla L(x,\lambda) = \begin{pmatrix} \nabla f(x) + J_g^T(x)\lambda \\ g(x) \end{pmatrix} = 0$$

Equality Constraint Optimization

- Problems with equality constraints can be solved by finding a critical point of the Lagrangian using Newton's method

$$\nabla L(x,\lambda) = \begin{pmatrix} \nabla f(x) + J_g^T(x)\lambda \\ g(x) \end{pmatrix} = 0$$

  - To find this point requires the Hessian

$$H_L(x,\lambda) = \begin{pmatrix} B(x,\lambda) & J_g^T(x) \\ J_g(x) & 0 \end{pmatrix}$$

$$B(x,\lambda) = H_f(x) + \sum_{i=1}^{m} H_{g_i}(x)$$

Equality Constraint Optimization

- The step for Newton's method can be determined (as generally in Newton's method) as the solution to the linear system

$$\begin{pmatrix} B(x,\lambda) & J_g^T(x) \\ J_g(x) & 0 \end{pmatrix} \begin{pmatrix} s \\ \delta \end{pmatrix} = -\begin{pmatrix} \nabla f(x) + J_g^T(x)\lambda \\ g(x) \end{pmatrix}$$

  - In the case of pure equality constraints this can be solved directly by solving the *(m+n)x(m+n)* linear system
- Solution has to be verified as a minimum since it could be a saddle

Equality Constraint Optimization

- We need a means to judge the progress to evaluate convergence and feasibility
  - *x* in each step does not necessarily fulfill the constraints
    - Only when converged do the constraints have to be fulfilled
  - Merit functions can be used to measure progress
    - E.g. Penalty function $\phi_p(x) = f(x) + \frac{1}{2}\rho g(x)^T g(x)$
      - ρ represents tradeoff between optimality and feasilbity
      - Penalty function for a large ρ has the same minimum as the original function

Equality Constraint Optimization

- Penalty function provides another means to solve an equality constrained optimization problem
  - Constrained minimum of *f(x)* is the same as the unconstrained minimum of the penalty function for a sufficiently large ρ
    - Penalty function allows to convert a constrained optimization problem into an unconstrained problem
    - Unconstrained problem becomes ill-conditioned for large ρ
      - Penalty term starts to dominate and *f(x)* disappears
      - Range from which optimization converges correctly becomes small
    - Use of iterative approach can solve problem
      - Incrementally solve unconstrained optimization problems for increasing values of ρ

Inequality Constraints

- With inequality constraints, the matrix for Newton's method can no longer be solved directly.
  - Sometimes inequality constraints can be (partially) converted into equality constraints
    - Active set methods replace inequality constraints with equality constraints for a subset of the constraints
      - Inequality constraints that are not currently violated can be temporarily dropped as the next step is unlikely to violate them
      - Inequality constraints that are violated can be replaced by equality constraints since the equality is the closes value to fulfilling the constraint
    - Active set method requires to reevaluate the set of required constraints at each step

Extended Form

- Inequality constraints can be replaced by equality constraints and an additional, simpler inequality constraint
  - Introducing slack variables

    $$h(x) \le 0 \quad \Rightarrow \quad h(x) + x_s = 0 \ , \ x_s \le 0$$

  - System with slack variables can be represented as an extended system (m_h additional variables) with solely constraints on the slack variables
    - Allows more efficient solution methods since all constraints are uniform

Barrier Methods

- As for equality constraints, optimization problems with inequality constraints can be solved by converting them into a sequence of unconstrained optimization problems with appropriate penalties
  - Barrier functions form one-sided penalty function

$$\phi_\mu(x) = f(x) - \mu \sum_{i=1}^{m_p} \frac{1}{h_i(x)}$$

    - For sufficiently small μ the barrier function has the same minimum as the constrained optimization problem for *f(x)*
      - Again ill conditioned for small μ
    - Iterated unconstrainted optimization using barrier function with decreasing μ can address problem
      - Basis for interior point methods

Sequential Quadratic Programming

- Lagrange multipliers λ can also be used with inequality constraints but have to be 0 for all inactive constraints
  - Instead of optimizing *f(x)* we can again optimize the Lagrangian function *L(x, λ)*
    - At the constrained minimum the Lagrangian has a minimum and therefore constrained optimization solution has to be a solution to the nonlinear system

$$\nabla L(x,\lambda) = \begin{pmatrix} \nabla f(x) + J_g^T(x)\lambda \\ g(x) \end{pmatrix} = 0$$

Sequential Quadratic Programming

- The step for Newton's method can be determined (as generally in Newton's method) as the solution to the linear system with the added constraints for active set variables

$$\begin{pmatrix} B(x,\lambda) & J_g^T(x) \\ J_g(x) & 0 \end{pmatrix}\begin{pmatrix} s \\ \delta \end{pmatrix} = -\begin{pmatrix} \nabla f(x) + J_g^T(x)\lambda \\ g(x) \end{pmatrix}$$

  - This can no longer be solved directly but represents a quadratic programming problem for

$$f_q(s) = \frac{1}{2} s^T B(x,\lambda) s + s^T (\nabla f(x) + J_{g,h}^T(x)\lambda)$$
$$J_g(x)s + g(x) = 0$$
$$J_h(x)s + h(x) \le 0$$

Linear Programming

- Linear programs are an important special class of constrained optimization problems
  - Linear objective function *f(x)*
  - Linear constraints *g(x), h(x)*
    - Often with the additional constraints $x \geq 0$
- In linear programming problems (as in quadratic programming problems), the feasible region is convex
  - Since objective function is linear the optimum has to occur at a vertex of the constrained feasible region
    - Solution is at an intersection point of *n* constraints
  - Simplex algorithm uses extended form to navigate a search through the space of vertices

Simplex Algorithm

- In augmented form a vertex is characterized as a point in which at least n variables are 0

$$\begin{pmatrix} 1 & -c^T & 0 \\ 0 & A & I \end{pmatrix} \begin{pmatrix} Z \\ x \\ x_s \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix} \quad , \quad x \geq 0 \;\; \leq \;\; x_s \geq 0$$

  - Starting from one vertex, navigate to a neighbor with a lower value of Z
    - Neighbors are nodes in which one of the n variables that were set to 0 become non-zero
    - Selection of neighbor is the one that requires the smallest change (i.e. along steepest “gradient”)

Simplex Algorithm

- In the worst case the complexity of the simplex algorithm is exponential
  - Problems can be constructed where it moves through every single vertex
  - In practice it is usually very efficient and most of the time outperforms interior point methods (which are polynomial)
- Simplex algorithm can solve problems with large numbers of variables

Simplex Algorithm - Example

- Linear program:
  - Objective function: $P=-2x-3y-4z$
  - Constraints: $3x+2y+z \leq 10$, $2x+5y+3z \leq 15$, $x,y,z \geq 0$
- Convert linear program into extended form:
  - Objective function: $P+2x+3y+4z=0$
  - Constraints: $3x+2y+z+s=10$, $2x+5y+3z+t=15$, $x,y,z,s,t \geq 0$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 0 & 0 \\ 0 & 3 & 2 & 1 & 1 & 0 \\ 0 & 2 & 5 & 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} P \\ x \\ y \\ z \\ s \\ t \end{pmatrix} = \begin{pmatrix} 0 \\ 10 \\ 15 \end{pmatrix}$$

Simplex Algorithm - Example

- Determine initial vertex by identifying 3 variables that can simultaneously be 0 under the constraints
  - Convert the matrix such that first row contains 0 for all elements except the first column and the 3 variables that are equal to 0
    - In this case this is already the case when selecting *x, y, z*. Thus starting with the vertex described by *x=y=z=0*
    - To find next vertex, find the variable among the current ones that has the steepest downhill gradient in the objective function (i.e. highest coefficient in the first row) – in this case *z*

$$\begin{pmatrix} 1 & 2 & 3 & \boxed{4} & 0 & 0 \\ 0 & 3 & 2 & \boxed{1} & 1 & 0 \\ 0 & 2 & 5 & \boxed{3} & 0 & 1 \end{pmatrix} \begin{pmatrix} P \\ x \\ y \\ z \\ s \\ t \end{pmatrix} = \begin{pmatrix} 0 \\ 10 \\ 15 \end{pmatrix}$$

Simplex Algorithm - Example

- Identify the next variable to set to 0 (instead of *z*) by finding the constraint row that requires the minimal change in z (computed by dividing the right hand side of the constraint equations by the corresponding value in the z column and then selecting the constraint equation with the smallest value – in this case the second constraint (third row) and thus the variable t

$$\begin{pmatrix} 1 & 2 & 3 & \boxed{4} & 0 & 0 \\ 0 & 3 & 2 & \boxed{1} & 1 & 0 \\ 0 & 2 & 5 & \boxed{3} & 0 & 1 \end{pmatrix} \begin{pmatrix} P \\ x \\ y \\ z \\ s \\ t \end{pmatrix} = \begin{pmatrix} 0 \\ 10 \\ 15 \end{pmatrix} \quad \begin{matrix} \\ 10/1=10 \\ 15/3=5 \end{matrix}$$

Simplex Algorithm - Example

- Eliminate the entries in the z column for all but the selected constraint row and eliminate all entries corresponding to non-zero variables in the first row.

$$\begin{pmatrix} 1 & -2/3 & -11/3 & 0 & 0 & -4/3 \\ 0 & 8/3 & 1/3 & 0 & 1 & -1/3 \\ 0 & 2 & 5 & 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} P \\ x \\ y \\ z \\ s \\ t \end{pmatrix} = \begin{pmatrix} -20 \\ 5 \\ 15 \end{pmatrix}$$

- Find the direction towards the next vertex by identifying the strongest downhill direction for the selected variables (*x,y,t*)
  - Since all entries in the first row are negative, there is now way to improve the function and we have reached the optimal vertex with *x=y=t=0*

Simplex Algorithm - Example

- Find the values for all the variables in the vertex

$$\begin{pmatrix} 1 & -2/3 & -11/3 & 0 & 0 & -4/3 \\ 0 & 8/3 & 1/3 & 0 & 1 & -1/3 \\ 0 & 2 & 5 & 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} P \\ x \\ y \\ z \\ s \\ t \end{pmatrix} = \begin{pmatrix} -20 \\ 5 \\ 15 \end{pmatrix}$$

  - From the first row we obtain the objective function *P=2/3x+11/3y+4/3t-20* which is (as it has to be) minimal at *x=y=t=0* with a value of *-20*
  - Using the second constraint equation and the already known variable values we obtain *2x+5y+3z+t = 3z = 15* and thus *z=5*
    - Solution is *x=y=0 , z=5 , P=-20*

Constrained Optimization

- Constrained Optimization allows to find the best parameters for arbitrary objective functions while obeying constraints
  - Equality and inequality constraints
- Special cases of Linear programming can be addressed using the constraints explicitly
- Simplex algorithm
- Most general solution methods transform the constrained problem into an unconstrained one with the same solution
  - Lagrangian function
  - Penalty or barrier functions
    - Convergence is enhanced by iterating over increasingly strict barrier functions