1 The Basic Goal

The main idea is to transform a given constrained problem into an equivalent unconstrained problem. The theory and methods for unconstrained optimization can then be applied to the new problem.

2 Overview of methods for constrained NLP

2.1 Quadratic Programs
2.2 Separable Programs
2.3 Gradient Descent Style Methods
2.4 Newton Style Methods
2.5 Penalty Function
2.6 Barrier Function
2.7 Sequential Approximation

• all these methods take a constrained to unconstrained approach to solve the problem
  eg: penalty method penalize the function and assumes unconstrained

• most of these methods don’t guarantee but allow to approximate a solution
3 General form of constrained NLP

\[
\begin{align*}
\text{min (or max) } & f(x) \\
\text{Subject to } & g_i(x) = 0 \ (i \in E) \\
& g_i(x) \geq 0 \ (i \in I)
\end{align*}
\]

- Here E is an index set for the equality constraints and I is an index set for inequality constraints.
- \( f \) and \( g_i \) are twice continuously differentiable.

\textit{exempli gratia:}

\[
\begin{align*}
\text{min } & x_1^2 - 2x_1 + x_2^2 - x_3^2 + 4x_3 \quad (1) \\
\text{s.t. } & x_1 - x_2 + 2x_3 = 2 \quad (2)
\end{align*}
\]

...we can substitute \( x_1 = 2 + x_2 - 2x_3 \) into the objective and now the problem is unconstrained!

\[
\begin{align*}
\text{min } & 2x_2^2 + 3x_3^2 - 4x_2x_3 + 2x_2 \quad (3)
\end{align*}
\]

- A general iterative method for this is:
  \( x_{k+1} = x_k + p \); then \( x_{k+1} \) must also satisfy \( Ax = b \), and
  \( A(x_{k+1} + p) = b \rightarrow A_p = 0 \)
- The problem now becomes \( \text{min } f(x + Zv) \), where \( Z \) is the null space matrix for \( A \), and \( p = Zv \), where \( v \) is any vector. Now the problem is unconstrained.

4 A Matrix Approach

- Recall that for \( Ax = 0 \), all solutions \( x \) form the null space for \( A \), and \( Z \) is a basis for that null space. \( A \) is a plane in three dimensions, and \( Z \) is everything orthogonal to that plane.
- \( A \) is size \( m \times n \), \( m \leq n \), and if \( A \) has rank \( m \), then \( Z \) has rank \( (n - m) \).
- Back to the example, repeated from the prior section:
  \textit{e.g.}

\[
\begin{align*}
\text{min } & x_1^2 - 2x_1 + x_2^2 - x_3^2 + 4x_3 \quad (4) \\
\text{s.t. } & x_1 - x_2 + 2x_3 = 2 \quad (5)
\end{align*}
\]
• Assume we have a way to find the null space $Z$. One such choice is $Z = \begin{pmatrix} 1 & -2 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$

• Then any feasible new $x_{k+1}$ can be written

$$x + Zv = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & -2 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}v = x_k + Zv$$

(6)

• We seek to optimize the term $Zv$. Optimality conditions involve derivatives of a reduced function.

$$\phi(v) = f(x+Zv)$$

(7)

$$\nabla\phi(v) = Z^T \nabla f(x+Zv) = Z^T \nabla f(x)$$

(8)

$$\nabla^2\phi(v) = Z^T \nabla^2 f(x+Zv)Z = Z^T \nabla^2 f(x)Z$$

(9)

• The term $\nabla\phi(v) = Z^T \nabla f(x)$ is called the “reduced gradient.”
• The term $\nabla^2\phi(v) = Z^T \nabla^2 f(x)Z$ is called the ”reduced Hessian.”

5 Necessary Condition, Linear Equality Constraints

$$Z^T \nabla f(x) = 0$$

(10)

$$Z^T \nabla^2 f(x)Z \to \text{positive semidefinite.}$$

(11)

6 Sufficient Condition, Linear Equality Constraints

$$Ax = b$$

(12)

$$Z^T \nabla f(x) = 0$$

(13)

$$Z^T \nabla^2 f(x)Z \to \text{positive definite.}$$

(14)

e.g. our previous example:
\[ x = \begin{pmatrix} 2.5 \\ -1.5 \\ -1 \end{pmatrix} \]  \hspace{1cm} (15)

\[ \nabla f(x) = \begin{pmatrix} 2x_1 - 2 \\ 2x_2 \\ -2x_3 + 4 \end{pmatrix} \]  \hspace{1cm} (16)

\[ \nabla f(x) = \begin{pmatrix} 3 \\ -3 \\ 7 \end{pmatrix} \]  \hspace{1cm} (17)

\[ Z^T \nabla f(x) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \]  \hspace{1cm} (18)

The reduced Hessian is positive definite at \( x \), so we know that \( x \) is optimal.

7 Lagrange Multipliers

- Lagrange Multiplier express the gradient at the optimum as a linear combination of the rows of the constraint matrix \( A \)
- It indicates the sensitivity of the optimal objective value to the changes in the data
- Most applications where data is approximate, the only choice is to solve the problem using the best available estimates. Once a solution is obtained, the next step is to assess the quality of the resulting solution
- Question: how sensitive is the solution to variation in the data?

7.1 Derivation of \( \lambda \)

Take a look at the necessary conditions again. We can represent the gradient as \( \nabla f(x) = A^T \lambda \) by breaking it into null and range spaces. Note that \( \nabla f(x^*) = 0 \) at \( x^* \) for some \( m \)-vector \( \lambda \).
\[ \nabla f(x^*) = 0 \quad (21) \]
\[ \nabla f(x^*) = Zv + A^T \lambda \quad (22) \]
\[ Z^T \nabla f(x^*) = Z^T Zv + Z^T A^T \lambda \quad (23) \]
\[ \nabla f(x^*) = A^T \lambda \quad \text{...since } Z^T Z \text{ is nonzero, so } Zv = 0. \quad (24) \]

At a local minimum, the gradient of the objective function is a linear combination of the gradients of the constraints. We call these \( \lambda \) values "Lagrange multipliers." Why do we care about \( \lambda \)? Say we perturb the right hand side of the constraints, \( b \), to a new value \( b + \delta \). Using a Taylor series, we find that:

\[
\begin{align*}
 f(x) &\approx f(x) + (x_{\text{row}} - x)^T \nabla f(x) \\
 &\quad = f(x) + (x_{\text{new}} - x) A^T \lambda \\
 &\quad = f(x) + \delta \lambda \\
 &\quad = f(x) + \sum_{i=1}^{m} \delta_i \lambda_i
\end{align*} \quad (25) \]

In other words, the rate of change of \( f \) changes at the rate of the Lagrange parameters. So the Lagrange parameters represent shadow prices, or dual variables. Now we can develop a Lagrangian function:

\[
\nabla f(x) - A^T \lambda = 0 \quad (29) \\
A x = b \quad (30)
\]

This is a set of \( (n+m) \) equations, with \( (n+m) \) unknowns \( x \) and \( \lambda \). We can consider the dual variables as unknowns. Other representations are:

\[
\begin{align*}
 L(x, \lambda) &= f(x) - \lambda^T (A x - b) \\
 L(x, \lambda) &= f(x) - \sum \lambda_i (A x - b)_i
\end{align*} \quad (31) \quad (32) \]

The gradient w.r.t. both \( x \) and \( \lambda \) gives our optimality conditions, \( \nabla L(x, \lambda) = 0 \). The local minimizer must be a stationary point of the Lagrangian equation, \( L \).
7.2 Lagrangian optimality conditions

In the case of linear inequalities, we have the following conditions for optimality. Let \( x \) be a local minimum of \( f \) over \( Ax \geq b \), and let \( \lambda \) be a vector of Lagrange multipliers.

Necessary conditions:

\[
\nabla f(x) = A^T \lambda \quad \text{(34)}
\]

or

\[
Z^T \nabla f(x) = 0 \quad \text{(35)}
\]

\[
\lambda \geq 0 \quad \text{(36)}
\]

\[
\lambda^T (Ax - b) = 0 \quad \text{(37)}
\]

Sufficient conditions:

\[
Ax \geq b \quad \text{(39)}
\]

\[
\nabla f(x) = A^T \lambda \quad \text{(40)}
\]

or

\[
Z^T \nabla f(x) = 0 \quad \text{(41)}
\]

\[
\lambda \geq 0 \quad \text{(42)}
\]

\[
\lambda^T = 0 \text{ or } (Ax - b) = 0 \quad \text{ (Strict complimentarity.)} \quad \text{(43)}
\]

\[
Z^T \nabla^2 f(x) Z \rightarrow \text{positive definite} \quad \text{(44)}
\]

8 Descent methods to find the optimum

8.1 Gradient descent

Consider again the equality constraints, \( Ax = b \). Any vector can be written \( x = \hat{A}x_A + Zx_z \), where \( \hat{A} \) is the range space, and \( Z \) is the null space.

\[
Ax = b \quad \text{(45)}
\]

\[
A(\hat{A}x_A + Zx_z) = b \quad \text{(46)}
\]

\[
\hat{A}A^T x = b \quad \text{(47)}
\]

Bypass technical problems by saying that \( A \) has full rank. Then we can claim "\( x + p \) satisfies \( Ap = 0 \)." We want to search the feasible region for the optimal \( x \) in a step-by-step manner. This is known as a line search method. The idea is to move from place to place in the feasible region in the vector direction \( p = Zp_z \), which is the set of directions where the equality constraints are not violated.
Here \( p \) is a descent direction if \( p^T \nabla f(x) = p_z^T Z^T \nabla f(x) \leq 0 \), or \( p_z^T f(x) \leq 0 \). Previously our expression for steepest descent was: \( \min \frac{y^T f}{\|y\|} \). Now we have

\[
p = Zp_z = -Z(Z^T Z)^{-1}Z^T \nabla f(x) \quad (48)
\]

\[
p = -ZZ^T \nabla f(x) \quad \text{...the reduced gradient.} \quad (49)
\]

The \((Z^T Z)^{-1}\) term disappears if you find an orthogonal null space. This method is like steepest descent, but slightly easier. It is still a valid way to search the space. All you need to know is a local gradient and a basis for the null space, \(Z\). The prior work involved is to find that \(Z\).

### 8.2 Newton-type methods

Newton-type methods use the \( \nabla^2 f \) term as well, in the form "\( \nabla^2 f = G \) term." \( p \) turns out to be the solution of

\[
Z^T G(x)Zp_z = -Z^T \nabla f(x) \quad (50)
\]

This is the same as before, but everything has to be projected in the null space direction. "Quasi-Newton methods" approximate \( G \) at each step using various tricks, because \( G \) and \( G^{-1} \) are difficult to calculate.

### 8.3 Active set methods

We can convert inequality (nonbinding) constraints to equality (binding) constraints as follows:

\[
\begin{align*}
\left\{ \begin{array}{ll}
\min & f(x) \\
\text{s.t.} & Ax \geq b
\end{array} \right. \\
\rightarrow & \left\{ \begin{array}{ll}
\min & f(x) \\
\text{s.t.} & \hat{A}x = \hat{b}
\end{array} \right.
\end{align*}
\quad (51)
\]

...where \( \hat{A} \) is the correct active set of the solution (the binding set at the final solution).

**Active Set Methods** iterate both where you are, \(x\), and what constraints are binding. The concepts for active set methods are:

1. Vector \( x \) is the current point, and \( \hat{A} \) are binding constraints at \( x \).
2. Check whether \( x \) is a solution to the corresponding equality constraints \( \nabla f(x) = \hat{A}^T \lambda \) for some vector \( \lambda \).
3. If the answer to 2. is yes, and \( \lambda \geq 0 \), then \( x \) is a solution.
4. If the answer to 2. is yes, but $\lambda_j < 0$ for some $j$, then find direction $\alpha$:
\[
\alpha : \nabla f(x)^T p < 0 \quad (52)
\]
\[
a_j^T \alpha > 0 \quad (53)
\]
\[
\tilde{a}_i^T \alpha = 0 \quad (54)
\]
\[
i \neq j \quad (55)
\]
(a_j is dropped from the working set) 5. If $\nabla f(x) \neq \tilde{A}^T \lambda$, then construct a search direction $a$ with $A_a = 0$, and $\nabla f(x)^T p < 0$.

### 8.4 Quadratic Objective

**See full working example in textbook under Quadratic Programming**

A more special case - constraints are linear and objective is quadratic.

\[
\text{max } f(x) = cx - 0.5 x^T Q x
\]
such that $Ax \leq b$
\[
x \geq 0
\]

Q is a matrix in objective
\[
e.g. \ c= 15 \ 30 \ x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \ Q = \begin{bmatrix} 4 & -4 \\ -4 & 8 \end{bmatrix}
\]

\[
A = \begin{bmatrix} 1 & 2 \end{bmatrix} \ b = [30]
\]
\[
x^T Q x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} 4 & -4 \\ -4 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} =
\]
\[
4 x_1^2 - 4 x_1 x_2 - 4 x_2^2 + 8 x_3^2
\]
This is the special case where Q is positive semi definite. This leads to a modified simplex method. Consider the KKT conditions:
\[
15 + 4 x_2 - 4 x_1 - u_1 \leq 0
\]
\[
x_1 (15 + 4 x_2 - 4 x_1 - u_1) = 0
\]
\[
30 + 4 x_1 - 8 x_2 - u_1 \leq 0
\]
\[
x_2 (30 + 4 x_1 - 8 x_2 - u_1) = 0
\]
\[
x_1 + 2 x_2 - 30 \leq 0
\]
\[
u_1 (x_1 + 2 x_2 - 30) = 0
\]
Introduce slack variables to get a set of equality constraints.

Some variables are 'complimentary'.
\[
x_1 y_1 + x_2 y_2 + u_1 v_1 = 0
\]
\[
x_1, x_2, y_1, y_2, v_1 \geq 0
\]
\[
=> \text{ either } x_1 = 0 \text{ or } y_1 = 0 \text{ etc.}
\]
Leads to the set of equations:
\[4x_1 - 4x_2 + u_1 - y_2 = 15\]
\[-4x_1 + 8x_2 + 2u_1 - y_2 = 30\]
\[x_1 + 2x_2 + v_1 = 30\]
\[x_1 \geq 0, x_2 \geq 0, u_1 \geq 0, y_1 \geq 0, y_2 \geq 0, v_1 \geq 0\]
\[x_1 y_1 + x_2 y_2 + u_1 v_1 = 0\]
\[i.e. \text{Almost an LP problem.}\]

Optimal to original = feasible to this set of linear equations plus complexity constraints.

Modified simplex - uses phase 1 idea to find solution.
i.e. min \[Z = \sum_i Z_i\]
such that satisfying constraints \[\begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} x \\ Z \end{bmatrix} = \text{startingsolution where} x=0, u=0, y=-c^T, v = b\]
Use the simplex method with restricted entry rule:
Exclude any entering variable whose complimentary variable is already basic (non zero).
Solution to phase 1 ⇒ Feasible solution to original problem ⇒ optimal solution to the real problem.

Summary

- quadratic programming uses simplex method, objective is to find a solution to the KKT condition. These are linear because the objective is quadratic.
- set up the problem, find a feasible solution to linear equation by keeping the complemenarity condition. Set this up just as e would to find an initial solution to start the simplex method i.e. minimize sum of dummy variables
- apply simplex, keeping complementary condition satisfied by not allowing 2 complementary variables to both be in the basis
- if the final solution has objective value zero, we have found a solution to the KKT conditions - therefore solved the original optimization

- End of October 14 2010 lecture -