MIT, 2.098/6.255/15.093J
Optimization Methods
Final Review, Fall 2009

December 11, 2009

Pre Mid-term

- Geometry of LP
- Simplex Method
- Duality Theory
- Sensitivity Analysis
- Large Scale Optimization
- Networks

Post Mid-term

- Integer Optimization
  - Formulations
  - Algorithms : Cutting Planes, Branch and Bound, Heuristics
  - Lagrangean Duality
- Dynamic Programming
- Non-linear Optimization
  - Convex functions
  - Unconstrained Optimization : Steepest Descent, Newton's Method and Conjugate Gradient
  - Constrained Optimization : KKT Conditions

- Interior Point Methods
  - Affine Scaling
  - Barrier Methods
- Semi-definite Optimization

1 Integer Optimization

1.1 Formulations

- Binary choice (x is 0 or 1 depending on the chosen alternative)
- Forcing constraints ($x \leq y$, decision x is made only if decision y is made)
- Relations between variables ($\sum_{i=1}^{n} x_i \leq 1$, at most one of the variables can be one)
- Disjunctive constraints (given m constraints, need that atleast k are satisfied - $\sum_{i=1}^{k} y_i \geq k,\ \ a_i' x \geq b_i y_i,\ \ y_i \in \{0,1\}$)
- Arbitrary piecewise linear cost functions

Guidelines for Strong Formulations

The quality of a formulation of an integer optimization problem with feasible solution set $\mathcal{F}$ can be judged by the closeness of the feasible set of its linear relaxation to the convex hull of $\mathcal{F}$.

- Facility Location - two formulations : P_{FL} and P_{AFL} but P_{FL} is closer to the convex hull although it has many more constraints than P_{AFL}.
- In general, good formulations can have exponential number of constraints.

1.2 Cutting Planes - Gomory Cuts

Key idea is to cut off parts of the feasible region, so that all integer solutions are still feasible, but the solution from the LP relaxation is violated.
Let x^* be an optimal basic feasible solution and let B be an associated optimal basis. We partition x into a subvector x_B of basic variables and a subvector x_N of nonbasic variables. WLOG, let the first m variables are basic, so that $x_{B(i)} = x_i, i = 1, ..., m$. Let N be the set of indices of nonbasic variables. Let A_N be the submatrix of A with columns $A_i, i \in N$. From the optimal tableau, we obtain the coefficients of the constraints:

$$x_B + B^{-1} A_N x_N = B^{-1} b$$

Let $\overline{a}_{ij} = (B^{-1}A_j)_i$ and $\overline{a}_{i0} = (B^{-1}b)_i$. We consider one equality from the optimal tableau, in which $\overline{a}_{i0}$ is fractional:

$$x_i + \sum_{j \in N} \overline{a}_{ij} x_j = \overline{a}_{i0}$$

Since, $x_j \geq 0, \ \forall j$, we have

$$x_i + \sum_{j \in N} \lfloor \overline{a}_{ij} \rfloor x_j \leq x_i + \sum_{j \in N} \overline{a}_{ij} x_j = \overline{a}_{i0}$$

Since x_j should be integer, we obtain

$$x_i + \sum_{j \in N} \lfloor \overline{a}_{ij} \rfloor x_j \leq \lfloor \overline{a}_{i0} \rfloor$$

This inequality is valid for all ineteger solutions, but it is not satisfied by x^*.

1.3 Branch and Bound

The idea is to divide the entire feasible region into smaller regions, such that it is easier to compute lower bounds on these subproblems so that some of them can be eliminated given information on other parts of the tree. Let $b(\mathcal{F}_i)$ denote the lower bound to the optimal cost of subproblem $\mathcal{F}_i$. Let U be an upper bound on the optimal cost, which could be the cost of the best feasible solution encountered thus far.

- Select an active subproblem $\mathcal{F}_i$.
- If the subproblem is infeasible, delete it; otherwise, compute $b(\mathcal{F}_i)$ for the corresponding subproblem.
- If $b(\mathcal{F}_i) \geq U$, delete the subproblem.
- If $b(\mathcal{F}_i) < U$, either obtain an optimal solution to the subproblem, or break the corresponding subproblem into further subproblems, which are added to the list of active subproblems.

1.4 Lagrangean Duality

Integer Programming problems are hard to solve in general. However, there are some sets of constraints that can be sovled *more efficiently* than others. The main idea is to relax difficult constraints by adding *penalties* to the cost function instead of imposing them as constraints.

Consider the primal problem:

$$\begin{array}{ll} \min & c'x \\ \text{s/t.} & Ax \geq b \\ & x \in X \end{array}$$

where $X = \{x \in \mathcal{Z}^n | Dx \geq d\}$. Relax the difficult constraints $Ax \geq b$ with Lagrange multipliers $p \geq 0$ (if the constraints are equalities, then p is unconstrained), we obtain the problem:

$$\begin{array}{ll} \text{Z(p)} = & \min\ c'x + p'(b - Ax) \\ \text{s.t.} & x \in X \end{array}$$

Property: Z(p) is a piecewise concave function.

The Lagrange dual problem is written as follows:

$$Z_D = \max_{p \geq 0} Z(p)$$

Theorem:

$$\begin{array}{ll} Z_D = & \min\ c'x \\ \text{s.t.} & Ax \geq b \\ & x \in \text{conv}(X) \end{array}$$

N.b. X is a discrete set, with a finite number of points, while $\text{conv}(X)$ is a *continuous* set, in fact a *polyhedron*. Why? Try drawing these sets for a small example.

We have that the following integer programming weak duality inequalities always hold (for a minimization problem, of course):

$$Z_{LP} \leq Z_D \leq Z_{IP}.$$

2 Dynamic Programming

1. State x_k;
2. Control u_k;
3. Dynamics: $x_{k+1} = f_k(x_k, u_k)$;
4. Boundary Conditions: $J_N(x_N), \quad \forall x_N$.
5. Recursion: $J_k(x_k) = \min\limits_{u_k \in \mathcal{U}_k} \big[g_k(x_k, u_k) + J_k(x_{k+1})\big]$.

3 Non-linear Optimization

How to determine whether a function is convex

If a function is differentiable, we can use the following facts:

- $\nabla^2 f(x)$ is PSD $\forall x \implies$ f is convex
- $\nabla^2 f(x)$ is PD (positive definite) $\forall x \implies$ f is strictly convex

Once we know a few basic classes of convex functions, we can use the following facts:

- Linear functions $f(x) = a^\top x + b$ are convex
- Quadratic functions $f(x) = \frac{1}{2}x^\top Qx + b^\top x$ are convex if Q is PSD (positive semi-definite)
- Norms are convex functions (the proof is left an exercise, using the properties of norms defined above)
- $g(x) = \sum_{i=1}^{k} a_i f_i(x)$ is convex if $a_i \geq 0$, f_i convex, $\forall i \in \{1, \ldots, k\}$

4 Unconstrained Optimization

4.1 Optimality Conditions

Consider the unconstrained problem: $min_{x \in \mathbb{R}^n} f(x)$, where $f(x)$ is twice differentiable, the optimality conditions are:

1. Necessary conditions:
   If x^* is a local minimum, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is PSD.
2. Sufficient conditions:
   If $\nabla f(\overline{x}) = 0$ and $\exists \epsilon > 0$: $\nabla^2 f(x)$ is PSD for all $x \in B(\overline{x}, \epsilon)$, then $\overline{x}$ is a local optimum.

For a continuously differentiable convex function f, the sufficient and necessary conditions for x^* to be a global minimum is $\nabla f(x^*) = 0$.

4.2 Gradient Methods

We are interested in solving the following nonlinear unconstrained problem: $min_{x \in \mathbb{R}^n} f(x)$. In general, gradient methods generate a sequence of iterates x_k that converge to an optimal solution x^*.
Generic algorithm elements:

1. Iterative update $x^{k+1} = x^k + \lambda^k d^k$
2. Descent direction $\nabla f(x^k)' d^k < 0$; for example, $d^k = -D^k \nabla f(x^k)$, where D^k is PSD.
3. Best step length $\lambda^k = argmin_{\lambda > 0} f(x^k + \lambda d^k)$.

4.3 Methods of Unconstrained Optimization

Steepest Descent

The unnormalized direction $-\nabla f(x)$ is called the direction of steepest descent at x.

For the steepest descent method, we set D_k to be the identity matrix I for all k. Thus the iterative step is just

$$x_{k+1} = x_k - \lambda_k \nabla f(x_k).$$

The algorithm stops when $\nabla f(x_k) = 0$, or when $\|\nabla f(x_k)\|$ is very small. The only unspecified parameter in this algorithm is the stepsize λ_k. There are various methods for choosing a stepsize. If $f(x)$ is a convex function, then one way to pick a stepsize is an exact line search. Since we already determined that the new point will be $x_k + \lambda_k d_k$, where $d_k = -\nabla f(x_k)$, we just want to find λ_k to minimize $f(x_k + \lambda_k d_k)$. Let $h(\lambda) = f(x_k + \lambda d_k)$. We want to find λ such that $h'(\lambda) = \nabla f(x_k + \lambda d_k)^T d_k = 0$. In some cases, we can find an analytical solution to this equation. If not, recognize that $h(\lambda)$ is convex since it is the composition of a convex function with a linear function. Thus $h''(\lambda) \geq 0$ for all λ, which implies $h'(\lambda)$ is increasing. Notice that $h'(0) = \nabla f(x_k)^T d_k = -\nabla f(x_k)^T \nabla f(x_k) = -\|\nabla f(x_k)\|^2 < 0$. Since $h'(\lambda)$ is increasing, we can find some $\bar{\lambda} > 0$ such that $h'(\bar{\lambda}) > 0$. Then we can keep bisecting the interval $\left[0, \bar{\lambda}\right]$ until we find λ^* such that $h'(\lambda^*) = 0$.

Newton's Method

Suppose we are at a point x and move to $x + d$. The second-order approximation of f at $x + d$ is

$$h(d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T H(x) d,$$

where $H(x)$ is the Hessian of f at x. We minimize h by finding d such that $\nabla h(d) = \nabla f(x) + H(x)d = 0$, i.e., $d = -H(x)^{-1} \nabla f(x)$, which is called the Newton direction or Newton step at x. This motivates Newton's method, in which the iterative step is

$$x_{k+1} = x_k - H(x_k)^{-1} \nabla f(x_k).$$

Here the stepsize is $\lambda_k = 1$ in every iteration, and $D_k = H(x_k)^{-1}$. Note that the Newton direction is not necessarily a descent direction, though it is as long as $H(x_k)^{-1}$ is positive definite.

Conjugate Gradient Method

Consider minimizing the quadratic function $f(x) = \frac{1}{2} x^\top Q x + c^\top x$.

1. $d_1, d_2, \ldots, d_m$ are **Q-conjugate** if

$$d_i^\top Q d_j = 0, \ \forall i \neq j$$

2. Let x_0 be our initial point.

3. Direction $d_1 = -\nabla f(x_0)$.

4. Direction $d_{k+1} = -\nabla f(x_{k+1}) + \lambda_k d_k$, where $\lambda_k = \frac{\nabla f(x_{k+1})^\top d_k}{d_k^\top Q d_k}$ in the quadratic case (and $\lambda_k = \frac{||\nabla f(x_{k+1})||^2}{||\nabla f(x_k)||^2}$ in the general case). It turns out that with each d_k constructed in this way, $d_1, d_2, \ldots, d_k$ are **Q**-conjugate.

5. By Expanding Subspace Theorem, x_{k+1} minimizes $f(x)$ over the affine subspace $S = x_0 + \text{span}\{d_1, d_2 \ldots, d_k\}$.

6. Hence finite convergence (n steps).

Rates of Convergence

We want to analyze the convergence rate, or the rate at which the error $e_k = \|x_k - x^*\|$ is decreasing, for the two methods described above. Suppose, for example, that the error was $e_k = 0.1^k$ in iteration k. Then we would have errors $10^{-1}, 10^{-2}, 10^{-3}, \ldots$. This error is decreasing linearly. As another example, suppose the error was $e_k = 0.1^{2^k}$. In this case, the errors would be $10^{-2}, 10^{-4}, 10^{-8}, \ldots$ (much faster!). This error is decreasing quadratically.

- Linear convergence rate for steepest descent.
- Quadratic (locally) for Newton's method.
- Newton's method typically converges in fewer iterations than steepest descent, but the computation can be much more expensive because Newton's method requires second derivatives.
- Conjugate Gradient converges in less than n steps.

5 Karush Kuhn Tucker Necessary Conditions

$$\begin{array}{lllll} \text{P:} & \min & f(x) & & \\ & \text{s.t.} & g_j(x) & \leq & 0, \quad j = 1, \ldots, p \\ & & h_i(x) & = & 0, \quad i = 1, \ldots, m \end{array}$$

(KKT Necessary Conditions for Optimality)
If $\hat{x}$ is local minimum of P and the following *Constraint Qualification Condition (CQC)* holds:

- The vectors $\nabla g_j(\hat{x}),\ j \in \mathcal{I}(\hat{x})$ and $\nabla h_i(\hat{x}),\ i = 1, \ldots, m$, are linearly independent, where $\mathcal{I}(\hat{x}) = \{j \ : g_j(\hat{x}) = 0\}$ is the set of indices corresponding to active constraints at $\hat{x}$.

Then, there exist vectors (u, v) s.t.:

1. $\nabla f(\hat{x}) + \sum_{j=1}^{p} u_j \nabla g_j(\hat{x}) + \sum_{i=1}^{m} v_i \nabla h_i(\hat{x}) = 0$
2. $u_j \geq 0,\ j = 1, \ldots, p$
3. $u_j g_j(\hat{x}) = 0,\ j = 1, \ldots, p$ (or equivalently $g_j(\hat{x}) < 0 \Rightarrow u_j = 0,\ j = 1, \ldots, p$)

(KKT + Slater)
If $\hat{x}$ is local minimum of P and the following *Slater Condition* holds:

- There exists some feasible solution $\bar{x}$ such that $g_j(\bar{x}) < 0,\ \forall j \in \mathcal{I}(\hat{x})$, where $\mathcal{I}(\hat{x}) = \{j\ : g_j(\hat{x}) = 0\}$ is the set of indices corresponding to active constraints at $\hat{x}$.

Then, there exist vectors (u, v) s.t.:

1. $\nabla f(\hat{x}) + \sum_{j=1}^{p} u_j \nabla g_j(\hat{x}) + \sum_{i=1}^{m} v_i \nabla h_i(\hat{x}) = 0$
2. $u_j \geq 0,\ j = 1, \ldots, p$
3. $u_j g_j(\hat{x}) = 0,\ j = 1, \ldots, p$ (or equivalently $g_j(\hat{x}) < 0 \Rightarrow u_j = 0,\ j = 1, \ldots, p$)

Example.
Solve

$$\begin{array}{lll} \min & x_1^2 + x_2^2 + x_3^2 & \\ \text{s.t.} & x_1 + x_2 + x_3 & \leq \quad -18 \end{array}$$

6 Interior Point Methods

6.1 Affine Scaling Methods

6.1.1 The Algorithm

Inputs: (A, b, c), an initial primal feasible solution $x^0 > 0$, an optimality tolerance $\epsilon > 0$ and parameter $\beta \in (0, 1)$

1. Initialization
2. Computation of dual estimates and reduced costs
3. Optimality Check, Unbounded Check
4. Update of primal solution

Theorem

If we apply the long-step affine scaling algorithm with $\epsilon = 0$, the following hold:
(a) For the long-step variant and under assumptions A and B, and if $0 < \beta < 1$, x^k and p^k converge to the optimal primal and dual solutions.
(b) For the second long-step variant, and under assumption A and if $0 < \beta < 2/3$, the sequences x^k and p^k converge to some primal and dual optimal solutions, respectively.

- simple mechanics.
- imitates simplex near the boundary.

6.2 Barrier Methods

A barrier function $G_(x)$, is a continuous function with the property that it approaches ∞ as one of the $g_j(x)$ approaches 0 from below.

Examples:

$$-\sum_{j=1}^{p} \log[-g_j(x)] \quad \text{and} \quad -\sum_{j=1}^{p} \frac{1}{g_j(x)}$$

Consider the primal/dual pair of linear optimization problems

$$\begin{array}{llrcl} \text{P:} & \min & c^\top x & & \\ & \text{s.t.} & Ax & = & b \\ & \text{s.t.} & x & \geq & 0 \end{array} \qquad \begin{array}{llrcl} \text{D:} & \max & b^\top p & & \\ & \text{s.t.} & A^\top p + s & = & c \\ & \text{s.t.} & s & \geq & 0 \end{array}$$

To solve P, we define the following barrier problem:

$$\begin{array}{llll} \text{BP:} & \min & B_\mu(x) \triangleq & c^\top x - \mu \sum_{j=1}^{n} \log x_j \\ & \text{s.t.} & & Ax = b \end{array}$$

Assume that for all $\mu > 0$, BP has an optimal solution $x(\mu)$. This optimum will be unique. Why?

As μ varies, the $x(\mu)$ form what is called the *central path*. $\lim_{\mu \to 0} x(\mu)$ exists and $x^* = \lim_{\mu \to 0} x(\mu)$ is an optimal solution to P.
Then the barrier problem from the dual problem is

$$\begin{array}{llll} \text{BD:} & \max & b^\top p + \mu \sum_{j=1}^{n} \log s_j \\ & \text{s.t.} & A^\top p + s = c \end{array}$$

Let $\mu > 0$. Then $x(\mu), s(\mu), p(\mu)$ are optimal solutions to BP and BD if and only if the following hold:

$$\begin{aligned} Ax(\mu) &= b \\ x(\mu) &\geq 0 \\ A^\top p(\mu) + s(\mu) &= c \\ s(\mu) &\geq 0 \\ x_j(\mu) s_j(\mu) &= \mu, \ \forall j \end{aligned}$$

To solve BP using the *Primal path following algorithm*, we:

1. Start with a feasible interior point solution $x_0 > 0$
2. Step in the Newton direction $d(\mu) = \left(I - X^2 A^\top (A X^2 A^\top)^{-1} A\right)\left(Xe - \frac{1}{\mu} X^2 c\right)$
3. Decrement μ

4. Iterate until convergence is obtained (complementary slackness above is ϵ-satisfied)

Note if we were to fix μ and carry out several Newton steps, then x would converge to $x(\mu)$. By taking a single step in the *Newton direction* we can guarantee that x stays "close to" $x(\mu)$, i.e. the *central path*. Hence following the iterative Primal path following algorithm we will converge to an optimal solution by this result and the first theorem above.

15.093J / 6.255J Optimization Methods

Fall 2009