Lecture 24: Algorithms for Constrained Optimization

Zizhuo Wang

Institute of Data and Decision Analytics (iDDA)
Chinese University of Hong Kong, Shenzhen

Dec 5, 2018

Announcements

- Homework 9 due next Wednesday (12/12)

Recap: Algorithms for Unconstrained Optimization

We have introduced four methods for unconstrained optimization problems

For single-variable problems

- Bisection method
- Golden section method

For multi-variable problems

- Gradient descent method
- Newton's method

Today, we introduce methods for constrained optimization problems

- Gradient projection method

Constrained Optimization

We consider the following constrained optimization problem:

$$\begin{array}{ll} \text{minimize}_{\mathbf{x}} & f(\mathbf{x}) \\ \text{subject to} & h_i(\mathbf{x}) = 0, \quad \forall i \\ & g_j(\mathbf{x}) \leq 0, \quad \forall j \end{array}$$

In unconstrained method, the main idea is:

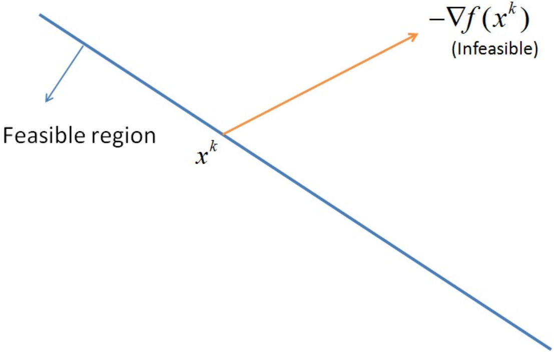
- At each $\mathbf{x}^k$, compute a descent direction, say $\mathbf{d}^k$
- Then find an appropriate step-size α_k and go to $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k$
- Both gradient and Newton's methods are based on this idea

The problem when we have constraints

- $\mathbf{x}^{k+1}$ may become infeasible

In the following, we assume we have an initial feasible solution

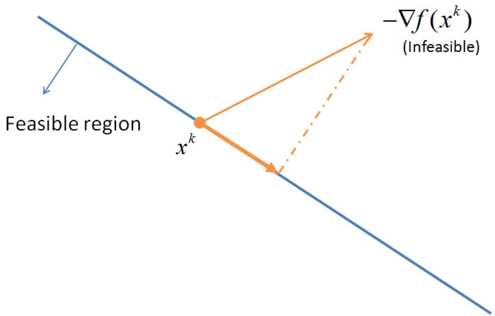
Illustration



- One of the solution to this problem is to use the gradient projection method

Main Idea of Gradient Projection Method

Project the descent direction onto the feasible set



We still use the general framework, however, the $\mathbf{d}^k$ chosen has to take into account feasibility.

Gradient Projection Method

We first consider a simpler case when the problem has linear equality constraints

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \end{array}$$

Remember at $\mathbf{x}^k$, the feasible direction set is

$$S_F(\mathbf{x}^k) = \{\mathbf{d} | A\mathbf{d} = 0\}$$

Idea: We select a search direction that maintains feasibility while achieving maximum descent.

Projection

We solve the following *projection* problem:

$$\begin{array}{ll} \text{minimize}_{\mathbf{d}} & \nabla f(\mathbf{x}^k)^T \mathbf{d} \\ \text{subject to} & A\mathbf{d} = 0, ||\mathbf{d}||_2 \leq 1 \end{array} \quad (1)$$

If we don't have the constraint $A\mathbf{d} = 0$, then the optimal solution to the above problem will be

$$\mathbf{d}^* = \frac{-\nabla f(\mathbf{x}^k)}{||\nabla f(\mathbf{x}^k)||}$$

which is the same as in the gradient method.

With the constraint $A\mathbf{d} = 0$, one can view (1) as projecting the vector $-\nabla f(\mathbf{x}^k)$ to the set $\{\mathbf{d} | A\mathbf{d} = 0\}$

Projection

Theorem (Projection Theorem)

Given any vector $\mathbf{d}$, the projection of $\mathbf{d}$ onto the region $A\mathbf{d} = 0$ is

$$(I - A^T(AA^T)^{-1}A)\mathbf{d}$$

We call $P_A = I - A^T(AA^T)^{-1}A$ the projection matrix.

Theorem

Let $\mathbf{y} = P_A\mathbf{x}$. Then $\mathbf{y}$ solves the following optimization problem:

$$\begin{array}{rl} \textit{minimize}_{\mathbf{y}} & ||\mathbf{y} - \mathbf{x}||_2^2 \\ \textit{s.t.} & A\mathbf{y} = 0 \end{array}$$

Proof Using KKT Conditions

Consider

$$\begin{array}{rl} \text{minimize}_{\mathbf{y}} & ||\mathbf{y}-\mathbf{x}||_2^2 \\ \text{s.t.} & A\mathbf{y}=0 \end{array}$$

Associate the constraints with multiplier λ, the KKT condition is:

$$2(\mathbf{y}-\mathbf{x})+A^T\lambda=0, \quad A\mathbf{y}=0$$

By multiplying A we get $\lambda = 2(AA^T)^{-1}A\mathbf{x}$.

Plug it back in the KKT condition, we get

$$\mathbf{y}=\mathbf{x}-A^T(AA^T)^{-1}A\mathbf{x}=(I-A^T(AA^T)^{-1}A)\mathbf{x}=P_A\mathbf{x}$$

It is a convex optimization problem, thus $\mathbf{y}$ is the unique optimal solution. $\square$

Properties of the Projection Matrix

There are several properties of P_A

- $AP_A = 0$ (therefore for any $\mathbf{x}$, $\mathbf{y} = P_A\mathbf{x}$ satisfies $A\mathbf{y} = 0$)
- $P_A^2 = P_A$
- All eigenvalues of P_A are either 0 or 1
- P_A is positive semi-definite, and $\mathbf{x}^T P_A \mathbf{x} = 0$ if and only if $P_A\mathbf{x} = 0$

Corollary

The optimal solution to the projection problem is

$$\bar{\mathbf{d}}^k = \frac{-P_A \nabla f(\mathbf{x}^k)}{||P_A \nabla f(\mathbf{x}^k)||_2}$$

In the gradient projection method, we will choose $\mathbf{d}^k = -P_A \nabla f(\mathbf{x}^k)$ as the search direction in each iteration.

Search Direction

We can verify that

$$\mathbf{d}^k = -P_A \nabla f(\mathbf{x}^k)$$

is a descent direction since

$$\nabla f(\mathbf{x}^k)^T \mathbf{d}^k = -\nabla f(\mathbf{x}^k)^T P_A \nabla f(\mathbf{x}^k) \leq 0$$

due to the positive semi-definiteness of P_A (it is < 0 as long as $P_A \nabla f(\mathbf{x}^k) \neq 0$)

- It is also a feasible direction since $A\mathbf{d}^k = 0$

Procedure

Procedure for gradient projection method with linear equality constraints:

Start with any point $\mathbf{x}^0$. Set $k = 0$ and stopping criterion $\epsilon > 0$. Define $P_A = I - A^T(AA^T)^{-1}A$

1. Check $||P_A \nabla f(\mathbf{x}^k)||$. If $||P_A \nabla f(\mathbf{x}^k)|| \leq \epsilon$, stop and output $\mathbf{x}^k$. Otherwise, continue to Step 2
2. Define $\mathbf{d}^k = -P_A \nabla f(\mathbf{x}^k)$ as the search direction
3. Use backtracking line search to find the step size α_k
4. Let $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k$, let $k = k + 1$. Go back to step 1

- Again, the only difference with the gradient or Newton's method is the search direction
- The reason for the difference is that we need to take care of feasibility in this case

Convergence Rate

Theorem (Convergence of Gradient Projection Method)

Under proper conditions, the gradient projection method converges to a local minimizer in a linear convergence rate.

Theorem (Gradient Projection for Convex Problems)

If the problem is convex, then under proper conditions, the gradient project method converges to a global minimizer in a linear convergence rate.

Linear Inequality Constraints

If the constraints are:

$$\begin{aligned} \mathbf{a}_i^T\mathbf{x} &\le b_i \quad i \in I_1 \\ \mathbf{a}_i^T\mathbf{x} &= b_i \quad i \in I_2 \end{aligned}$$

Then for any feasible solution $\mathbf{x}^k$, we first construct the feasible direction set:

$$\begin{aligned} \mathbf{a}_i^T\mathbf{d} &\le 0 \quad \text{if } i \in I_1 \text{ and } a_i^T\mathbf{x}^k = b_i \\ \mathbf{a}_i^T\mathbf{d} &= 0 \quad\quad\quad\quad i \in I_2 \end{aligned}$$

Linear Inequality Constraints

We consider a modified projection problem:

$$\begin{array}{lll} \text{minimize}_{\mathbf{d}} & \nabla f(\mathbf{x}^k)^T \mathbf{d} & \\ \text{subject to} & \mathbf{a}_i^T \mathbf{d} \le 0 & \text{if } i \in I_1 \text{ and } a_i^T \mathbf{x}^k = b_i \\ & \mathbf{a}_i^T \mathbf{d} = 0 & i \in I_2 \\ & \|\mathbf{d}\|_2 \le 1 & \end{array}$$

The projection in this case is more complicated, but the idea is the same.

- However, the projection matrices will be different at each iteration, depending on which constraint is active or not
- The overall procedure is still the same, except the search directions at each step are different

Nonlinear Constraints

If the constraints are

$$g_j(\mathbf{x}) \leq 0 \quad \forall j$$

Then we can use a linear approximation for the feasible direction set:

$$\nabla g_j(\mathbf{x})^T \mathbf{d} \leq 0 \quad \text{if } g_j(\mathbf{x}) = 0$$

The feasible set is a linear function of **d** and therefore, we can solve a similar projection problem

Newton's Method with Linear Equality Constraints

The above discussion is all about extending the gradient descent methods to the constrained cases.

- We can also extend Newton's method to problems with constraints

The idea is similar

- Approximate the objective function by a quadratic function (using second-order Taylor expansion)
- Find the minimizer of the quadratic expansion, however, in this case, with the constraints

Newton's Method with Linear Equality Constraints

We consider

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \end{aligned}$$

At $\mathbf{x}^k$, compute the Newton's step by

$$\begin{aligned} \text{minimize}_{\mathbf{v}} \quad & f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T \mathbf{v} + \frac{1}{2}\mathbf{v}^T \nabla^2 f(\mathbf{x}^k)\mathbf{v} \\ \text{s.t.} \quad & A\mathbf{v} = 0 \end{aligned}$$

By KKT conditions, this equality constrained quadratic program can be solved explicitly:

$$\begin{bmatrix} \nabla^2 f(\mathbf{x}^k) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \lambda \end{bmatrix} = \begin{bmatrix} -\nabla f(\mathbf{x}^k) \\ 0 \end{bmatrix}$$

where λ is the Lagrangian multiplier.

Newton's Method with Linear Equality Constraints

Therefore one can use $\mathbf{v}$ as the search direction, and use the same iterative framework

- Use backtracking search to find α and update

Similar idea can be applied to solve inequality constrained optimizations

- There might not be an explicit solution to the Newton's step
- However, it is just a quadratic program, which can be solved easily

To summarize, this approach projects the descent direction to the feasible direction set at each iteration, and otherwise proceeds the same as before.

Example

$$\begin{array}{rc} \text{minimize} & e^{x_1+x_2} + x_1^2 + 3x_2^2 - x_1x_2 \\ \text{subject to} & x_1 + 2x_2 = 1 \end{array}$$

The projection matrix is $I - A^T(AA^T)^{-1}A$ where $A = [1, 2]$. Thus, we have

$$P_A = \begin{bmatrix} 4/5 & -2/5 \\ -2/5 & 1/5 \end{bmatrix}$$

Use MATLAB to find the solution. Use the initial feasible solution $(1, 0)$

Example 2

$$\begin{aligned} \text{minimize}_{\mathbf{x}} \quad & ||A\mathbf{x} - \mathbf{b}||_2^2 \\ \text{subject to} \quad & W\mathbf{x} = \mathbf{z} \end{aligned}$$

Summary of Nonlinear Optimization

Optimality conditions

- KKT conditions
- Be able to formulate and analyze KKT conditions

Convexity

- Know the concepts: convex set, convex function, convex constraints, and convex optimization problem (also for concave)
- Can identify them
- Can convert problems into convex optimization problem (if possible)

Summary of Nonlinear Optimization

Algorithms

- Bisection, Golden section method
- Gradient descent, Newton's method
- Gradient projection method
- Understand the idea and procedures, can write some codes to implement them