An optimization survey

CE 377K

April 30, 2015

ANNOUNCEMENTS

Last HW posted (due last day of class)

Scheduling group presentations

STEP SIZE SELECTION

There are a number of ways to choose the step size λ after the target has been chosen. We'll go through three:

- The **method of successive averages** is simplest and fastest, but not very intelligent.
- The **line minimization rule** tends to work well in practice, but can be slower.
- The Armijo rule uses trial and error to quickly find a "reasonably good" λ value.

Method of successive averages

The method of successive averages uses a *fixed sequence* of λ values, rather than trying to customize λ at each step of the algorithm.

There are two risks with using fixed values of λ : if λ is too small, convergence will be very slow. If λ is too large, the algorithm may not converge at all.

The method of successive averages tries to avoid both of these difficulties by starting with larger values of λ and moving to smaller ones.

A typical sequence of λ values is 1/2, 1/3, 1/4, etc.

Example

(Conditional gradient method plus MSA).

When we started the conditional gradient method with $\begin{bmatrix} 0 & 0 \end{bmatrix}$ the target point was $\begin{bmatrix} 2 & 2 \end{bmatrix}$.

Using the method of successive averages the new point is $1/2 \begin{bmatrix} 2 & 2 \end{bmatrix} + (1 - 1/2) \begin{bmatrix} 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix}$ (objective reduced from 17 to 1.16).



Example

(Conditional gradient method plus MSA).

When we started the conditional gradient method with $\begin{bmatrix} 0 & 0 \end{bmatrix}$ the target point was $\begin{bmatrix} 2 & 2 \end{bmatrix}$.

Using the method of successive averages the new point is $1/2 \begin{bmatrix} 2 & 2 \end{bmatrix} + (1 - 1/2) \begin{bmatrix} 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix}$ (objective reduced from 17 to 1.16).

At this point the gradient is $\begin{bmatrix} 0 & -4 \end{bmatrix}$; the new target is any point where $x_2 = 2$. If (1,2) is the new target, then **x** is updated to $1/3 \begin{bmatrix} 12 \end{bmatrix} + 2/3 \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 14/3 \end{bmatrix}$. (Objective reduced from 1.16 to 0.358)

Here you see one downside of MSA — the global optimum would have been reached if we had chosen $\lambda = 1$. MSA does not have the ability to detect such cases, it always follows the pre-set sequence of step sizes.



The line minimization rule chooses the value of $\lambda \in [0, 1]$ which minimizes the objective function along the line connecting **x** to **x**^{*}.

This value can be found using the bisection method or Newton's method.

Specifically, we want to choose λ to minimize $f(\lambda \mathbf{x}^* + (1 - \lambda)\mathbf{x})$ subject to $0 \le \lambda \le 1$.

Example

(Conditional gradient plus line minimization). When we started the conditional gradient method with $\begin{bmatrix} 0 & 0 \end{bmatrix}$ the target point was $\begin{bmatrix} 2 & 2 \end{bmatrix}$.

Using the line minimization rule, $\lambda = 0.705$ and the new point is [1.41 1.41]. (Objective reduced from 17 to 0.289)



Example

(Conditional gradient plus line minimization). When we started the conditional gradient method with $\begin{bmatrix} 0 & 0 \end{bmatrix}$ the target point was $\begin{bmatrix} 2 & 2 \end{bmatrix}$.

Using the line minimization rule, $\lambda = 0.705$ and the new point is $\begin{bmatrix} 1.41 & 1.41 \end{bmatrix}$. (Objective reduced from 17 to 0.289)

At this point the gradient is $\begin{bmatrix} 0.82 & -2.36 \end{bmatrix}$. The new target minimizes $0.82x_1 - 2.36x_2$, so $\mathbf{x}^* = \begin{bmatrix} 0 & 2 \end{bmatrix}$.

Using the line minimization rule, $\lambda = 0.328$ and the new point is $\begin{bmatrix} 0.948 & 1.60 \end{bmatrix}$. (Objective reduced to 0.027)



Armijo rule

The Armijo rule tries to overcome disadvantages of both MSA (not very smart) and line minimization (it takes too long).

The Armijo rule does not try to find the value of λ which minimizes f, but is content to find a value of λ which reduces f "enough."

This rule is a "trial-and-error" technique, where we try different values of λ until we find an acceptable one.

An "acceptable" λ is defined as one for which

$$rac{f(\mathbf{x}) - f(\mathbf{x}(\lambda))}{\lambda} \geq lpha |f'(\mathbf{x}(0))|$$

where $\mathbf{x}(\lambda)$ is the new point as a function of λ , and f' is the derivative of f at \mathbf{x} , in the direction of \mathbf{x}^* .



Optimization survey

A good rule of thumb is to set $\alpha = 0.1$, and to try the sequence $\{1, 1/2, 1/4, 1/8, ...\}$ of λ values until one of them is acceptable.



Example

(Conditional gradient plus Armijo rule with $\alpha = 0.1$). At the initial point $\begin{bmatrix} 0 & 0 \end{bmatrix}$, the target point was $\begin{bmatrix} 2 & 2 \end{bmatrix}$, so

$$\mathbf{x}(\lambda) = \lambda \begin{bmatrix} 2 & 2 \end{bmatrix} + (1 - \lambda) \begin{bmatrix} 0 & 0 \end{bmatrix} = \begin{bmatrix} 2\lambda & 2\lambda \end{bmatrix}$$

So,
$$f(\mathbf{x}(\lambda)) = (2\lambda - 1)^2 + (2\lambda - 2)^4$$
 and
 $f'(\mathbf{x}(0)) = 4(2(0) - 1) + 8(2(0) - 2) = -20$

For a point to be acceptable in the Armijo rule, the decrease in the objective function (from 17) divided by λ must be at least $2 = 0.1 \times 20$.

Start by trying $\lambda = 1$. In this case, the objective function decreaes to 1; 16/1 > 2 so this point is acceptable and we move to $\begin{bmatrix} 2 & 2 \end{bmatrix}$.



The new point is $\begin{bmatrix} 2 & 2 \end{bmatrix}$, the gradient is $\begin{bmatrix} 2 & 0 \end{bmatrix}$, so the new target point minimizes $2x_1$; say $\mathbf{x}^* = \begin{bmatrix} 0 & 2 \end{bmatrix}$.

So,
$$\mathbf{x}(\lambda) = \begin{bmatrix} 2 - 2\lambda & 2 \end{bmatrix}$$
, $f(\mathbf{x}(\lambda)) = (1 - 2\lambda)^2$, and $f'(\mathbf{x}(0)) = -4(1 - 2(0)) = -4$.

For a point to be acceptable in the Armijo rule, the decrease in the objective function (from 1) divided by λ must be at least 0.4.

If $\lambda = 1$, the new point is $\begin{bmatrix} 0 & 2 \end{bmatrix}$ and the objective function is 1, so there is no decrease... unacceptable.

If $\lambda = 1/2$, the new point is $\begin{bmatrix} 1 & 2 \end{bmatrix}$ and the objective function is 0; since (1-0)/2 > 0.4 this point is acceptable.



A SURVEY OF OPTIMIZATION

We started by giving a few examples of optimization problems:

- Transit frequency-setting problem
- Roadway maintenance scheduling
- Shortest path
- Facility location

Since then we've also seen toll-setting and signal timing problems.

We then looked at techniques for solving simple kinds of problems:

- Stationary points for unconstrained problems generalized with Lagrange multipliers and KKT conditions
- Line search (bisection and Newton's) for single dimension problems
 also used for descent method "subproblems" and with a single Lagrange multiplier

At the other end of the scale were heuristics for solving very complicated problems:

- Simulated annealing how does it relate to the descent methods we just learned?
- Genetic algorithms mimicing biological evolution

Another place where heuristics have been used is for locating real-time travel information



Placing these signs in appropriate locations can help keep drivers away from congestion or accidents, but there is a limited budget for locating these signs.

This can also be framed as providing personalized alternate routes through a mapping service

(A) 7201 wood hollow drive 78731



VS.

N Lamar Blvd Add Destination - Show options • Byca Get Directions m Also available: Also available: Also available: Also available: Driving directions to N Lamar Blvd 4.1 mi - about 9 mins A 7201 Wood Hollow Dr 1 Austin, TX 78731 1. Head southwest on Wood Hollow Dr toward 🗭 0.1 mi Far W Blvd 2. Turn left at Far W Blvd 🗭 0.2 m 3. Turn right at N Me Pac Expy 4. Take the ramp on the left onto TX.1 S 🗭 1.8 mi 5. Take the 45th St exit 😰 0.2 mi Camp Mabry 6. Turn right at Fairview Dr. 7. Turn left at W45th St 8. Turn left at N Lamar Blvd 🗭 0.2 mi B N Lamar Blvd (1) Map data 62008 LeadDog Consulting, Tele Adlas

Decision variables: 0-1 decision variable for each intersection

- Constraint: Budget
- Objective: More complicated has to account for travel time probability distributions; potentially multiple drivers with different origins and destinations; and potentially congestion issues from re-routing.

The objective function was complicated enough that heuristic approaches (simulated annealing and some tailor-made heuristic) performed much better.



Network optimization problems were discussed next, because the network structure shows up often in practice and leads to specialized algorithms.

- Minimum spanning tree
- Shortest path problem
- Maximum flow problem
- Minimum cost flow problem

Note that every network optimization problem we saw could also be solved as a linear (or even nonlinear) program. Variants of the shortest path problem show up very frequently in transportation. Some fancier versions include:

- Link costs can change over time
- Link costs are random variables which take some probability distribution
 - Minimize expected travel time
 - Minimize travel time standard deviation
 - Maximize probability of on-time arrival
 - Path can update based on real-time information
- More than one "cost" (e.g., time, distance, and cost)

Searching for on-streetparking is a probabilistic shortest path problem.



How can we better manage parking?

Optimization survey

Electric vehicles have the potential to reduce noise, pollution, and energy consumption.



However, their range is significantly shorter than gasoline vehicles. What can we do about "range anxiety?"

Optimization survey

From here, we moved to the general linear programming problem. These are useful both because they can be solved exactly and efficiently (simplex method) and as the subproblem in the conditional gradient method.

They also lend themselves very easily towards a sensitivity analysis based on reduced costs and the simplex tableau. In transportation, traffic congestion can be approximated by linear equations



(Cell Transmission Model)

The problem of determining the most efficient way to move a certain vehicle load through a congestible traffic system can be written as a linear program which encodes the CTM.

$$\operatorname{Minimize}_{\forall t \in \mathcal{T}} \sum_{\forall i \in \mathfrak{C} \setminus \mathfrak{C}_{\mathrm{S}}} x_{i}^{t}$$
(12)

subject to

$$\begin{split} \mathbf{x}_{i}^{t} - \mathbf{x}_{i}^{t-1} - \sum_{k \in \Gamma^{-1}(i)} \mathbf{y}_{ki}^{t-1} + \sum_{j \in \Gamma(i)} \mathbf{y}_{ij}^{t-1} = \mathbf{0}, \\ & \forall i \in \mathcal{C} \backslash \{\mathcal{C}_{\mathbf{R}}, \mathcal{C}_{\mathbf{S}}\}, \quad \forall t \in \mathcal{T} \quad (13) \\ \mathbf{y}_{ij}^{t} - \mathbf{x}_{i}^{t} \leq \mathbf{0}, \quad \mathbf{y}_{ij}^{t} \leq \mathbf{Q}_{i}^{t}, \quad \mathbf{y}_{ij}^{t} \in \mathbf{A}_{i}^{t} \mathbf{x}_{j}^{t} \leq \mathbf{A}_{j}^{t} N_{j}^{t}, \\ & \forall (i, j) \in \mathcal{C}_{\mathbf{0}} \cup \mathcal{C}_{\mathbf{R}}, \quad \forall t \in \mathcal{T} \quad (14) \\ \mathbf{y}_{ij}^{t} - \mathbf{x}_{i}^{t} \leq \mathbf{0}, \quad \mathbf{y}_{ij}^{t} \leq \mathbf{Q}_{i}^{t}, \quad \forall (i, j) \in \mathcal{C}_{\mathbf{S}}, \quad \forall t \in \mathcal{T} \quad (15) \\ \mathbf{y}_{ij}^{t} \in \mathbf{Q}_{j}^{t}, \quad \mathbf{y}_{ij}^{t} + \mathbf{b}_{j}^{t} \mathbf{x}_{j}^{t} \leq \mathbf{b}_{j}^{t} N_{j}^{t} \quad \forall (i, j) \in \mathcal{C}_{\mathbf{D}}, \quad \forall t \in \mathcal{T} \quad (16) \\ \sum_{\forall j \in \Gamma(i)} \mathbf{y}_{ij}^{t} - \mathbf{x}_{i}^{t} \leq \mathbf{0}, \quad \sum_{\forall j \in \Gamma(i)} \mathbf{y}_{ij}^{t} \leq \mathbf{Q}_{i}^{t} \\ & \forall i \in \mathcal{C}_{\mathbf{D}}, \quad \forall t \in \mathcal{T} \quad (17) \end{split}$$

$$y_{ij}^{t} - x_{i}^{t} \leq 0, \quad y_{ij}^{t} \leq Q_{i}^{t}, \quad \forall (i, j) \in \mathcal{E}_{\mathrm{M}}, \quad \forall t \in \mathcal{T}$$
(18)

$$\sum_{\forall i \in \Gamma^{-1}(j)} \boldsymbol{y}_{ij}^t \leqslant \boldsymbol{Q}_j^t, \quad \sum_{\forall i \in \Gamma^{-1}(j)} \boldsymbol{y}_{ij}^t + \delta_j^t \boldsymbol{x}_j^t \leqslant \delta_j^t N_j^t,$$

$$\forall j \in \mathscr{C}_{\mathrm{M}}, \quad \forall t \in \mathcal{T} \quad (19)$$

$$\begin{aligned} x_i^t - x_i^{t-1} + y_{ij}^{t-1} &= d_i^{t-1}, \\ j \in \Gamma(i), \quad \forall i \in \mathcal{C}_{\mathbb{R}}, \quad \forall t \in \mathcal{T}, \quad x_i^0 = \zeta_i, \quad \forall i \in \mathcal{C}, \end{aligned}$$
(20)

$$y_{ij}^0 = 0, \quad \forall (i,j) \in \mathscr{C}$$

$$(21)$$

$$x_i^t \ge 0, \quad \forall i \in \mathscr{C}, \quad \forall t \in \mathscr{T}$$
 (22)

$$y_{ij}^t \ge 0, \quad \forall (i,j) \in \mathscr{C}, \quad \forall t \in \mathcal{T}$$
 (23)

Finally, we discussed the most general class of nonlinear optimization problems. When the objective function and feasible region are convex, we can use descent methods to solve the problem.

Solving nonlinear problems drew on a lot of concepts from elsewhere in the course: stationary points, line search methods, linear programs, sensitivity analysis



How should signals be timed over a network (not just a single intersection)?

Optimization survey

What is the best way to manage an intersection when there are autonomous vehicles?