Descent methods

CE 377K

April 28, 2015

ANNOUNCEMENTS

Last HW posted (due last day of class)

Each group: please email me with group members and title of topic.

REVIEW

KKT conditions

Descent methods

Assume that we are given a convex optimization problem.

A *descent* method is an iterative algorithm consisting of the following steps:

1. Choose an initial feasible solution $\mathbf{x} \leftarrow \mathbf{x^0}$.
2. Identify a feasible "target" solution $\mathbf{x}^*$ in a "downhill direction."
3. Choose a step size $\lambda \in [0, 1]$ and set $\mathbf{x} \leftarrow \lambda \mathbf{x}^* + (1 - \lambda)\mathbf{x}$
4. Test for termination, and return to step 2 if we need to improve further.

First, we'll go through two ways to choose the target solution. Then we will go through three ways to choose the step size.

Target selection

Both of the target selection rules are based on the gradient at the current point, $\nabla f(\mathbf{x})$.

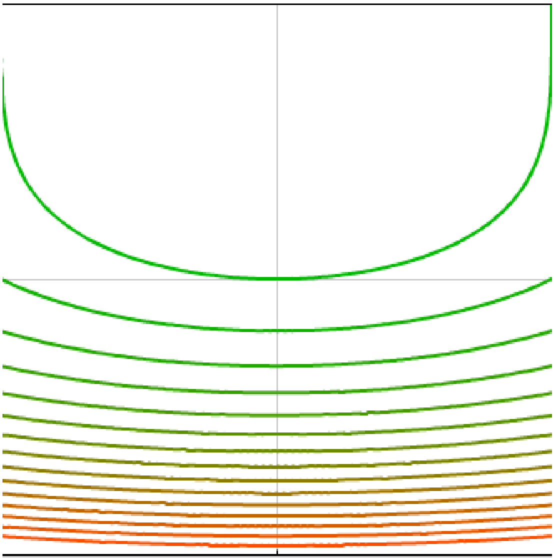
Remember that the gradient is a vector pointing in the direction of steepest *ascent*. Since our standard form is a minimization problem, we will want to move in the opposite direction as the gradient.

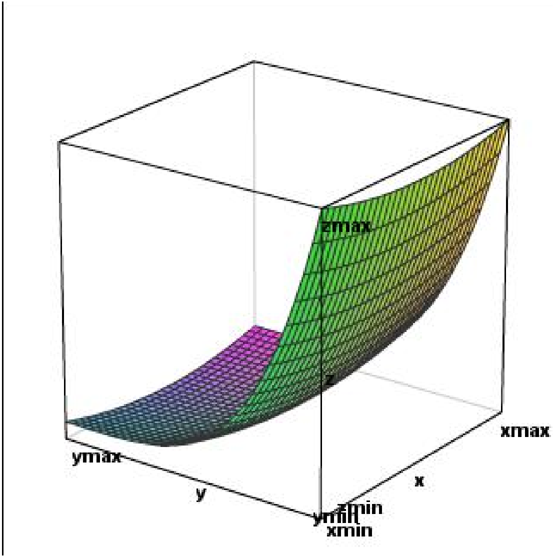
However, just using the gradient as a search direction creates problems, because it fails to account for the constraints.

The conditional gradient and gradient projection methods are designed to provide reasonable target points while accounting for constraints.

DESCENT METHODS

As an example, this week we will be minimizing $(x_1 - 1)^2 + (x_2 - 2)^4$ over the set $0 \leq x_1, x_2 \leq 2$.




zmax
z
ymax
y
ymin
zmin
xmin
x
xmax

Conditional gradient

In the conditional gradient rule, we assume that the slope of the objective is the *same* throughout the feasible region. (This is not true, but gives us a direction to move in.)

This is equivalent to replacing the objective function with its linear approximation at the current point $\mathbf{x}$.

We then solve the optimization problem for the approximate objective function $f(\mathbf{x}^*) = f(\mathbf{x}) + \nabla f(\mathbf{x})(\mathbf{x}^* - \mathbf{x})$ with the same constraints, and use the optimal $\mathbf{x}^*$ value for the target.

If all of the constraints are linear, the conditional gradient method is nothing more than solving a linear program.

Example

Starting with the initial solution $\mathbf{x} = \begin{bmatrix} 0 & 0 \end{bmatrix}$, the gradient at this point is $\begin{bmatrix} -2 & -8 \end{bmatrix}$.

The linearized objective $f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot (\mathbf{x}^* - \mathbf{x})$ is

$$17 + \begin{bmatrix} -2 & -4 \end{bmatrix} \cdot \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix}$$

The solution to $\min -2x_1 - 8x_2$ subject to $0 \leq x_1, x_2 \leq 2$ is $(2, 2)$. This is the target $\mathbf{x}^* = \begin{bmatrix} 2 & 2 \end{bmatrix}$

Notice that we can always simplify $f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot (\mathbf{x}^* - \mathbf{x})$ by removing constants. We can just as well minimize $\nabla f(\mathbf{x}) \cdot \mathbf{x}^*$

If we had chosen $\mathbf{x} = \begin{bmatrix} 2 & 0 \end{bmatrix}$ as the initial solution, the gradient is $\begin{bmatrix} 2 & -8 \end{bmatrix}$.

The linearized objective is then $\min 2x_1 - 8x_2$, and the target is $\mathbf{x}^* = \begin{bmatrix} 0 & 2 \end{bmatrix}$.

If we had chosen $\mathbf{x} = \begin{bmatrix} 1 & 1 \end{bmatrix}$ as the initial solution, the gradient is $\begin{bmatrix} 0 & -4 \end{bmatrix}$.

The linearized objective is then $\min -4x_2$, and any vector wih $x_2 = 0$ can be chosen as the target.

Gradient projection

In the gradient projection method, the target is found by calculating the point $\mathbf{x} - s\nabla f(\mathbf{x})$ (where s is another step size), and then calculating the *projection* of that point onto the feasible region.

The **projection** of a point $\mathbf{x}$ onto a set X is the point in X closest to $\mathbf{x}$, and is denoted by $\Pi_X \mathbf{x}$.

Examples of projection

Project the point $(5, 10)$ onto the set defined by $0 \leq x_1 \leq 7$, $0 \leq x_2 \leq 7$.

Project the point $(4, 2)$ onto the set defined by $x_1 + x_2 = 5$

Assume that $s = 1/2$ (this is another parameter that can be tuned)

At the point $\mathbf{x} = \begin{bmatrix} 0 & 0 \end{bmatrix}$, the gradient is $\begin{bmatrix} -2 & -8 \end{bmatrix}$.

The target is selected to be the projection of $\begin{bmatrix} 0 & 0 \end{bmatrix} - 1/2 \begin{bmatrix} -2 & -8 \end{bmatrix}$ onto the feasible set.

The projection of $\begin{bmatrix} 1 & 4 \end{bmatrix}$ onto the feasible set is $\begin{bmatrix} 1 & 2 \end{bmatrix}$, so this is the target.

As a rule of thumb, the gradient projection target tends to be “better”, but calculating the projection of a point onto complicated feasible regions is not easy.

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.

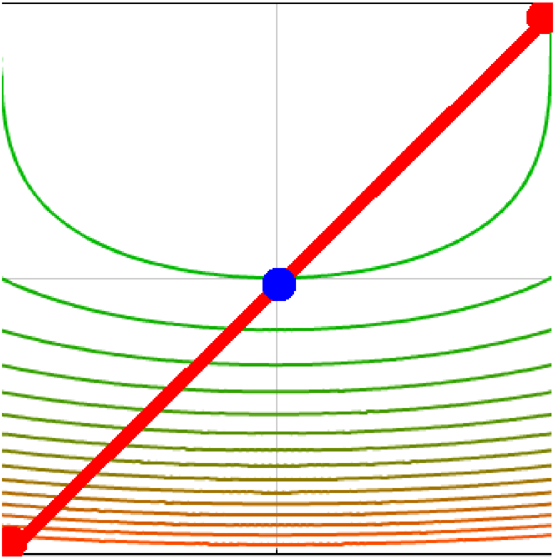
A typical sequence of λ values is 1/2, 1/3, 1/4, etc.

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

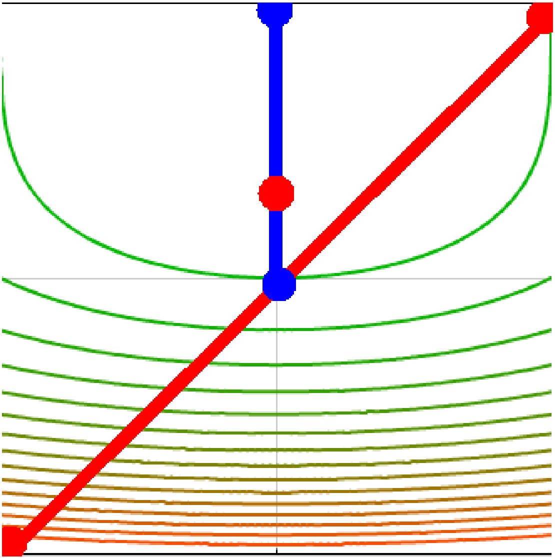
(Conditional gradient method plus MSA).

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 $\mathbf{x}$ 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.



Line minimization rule

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

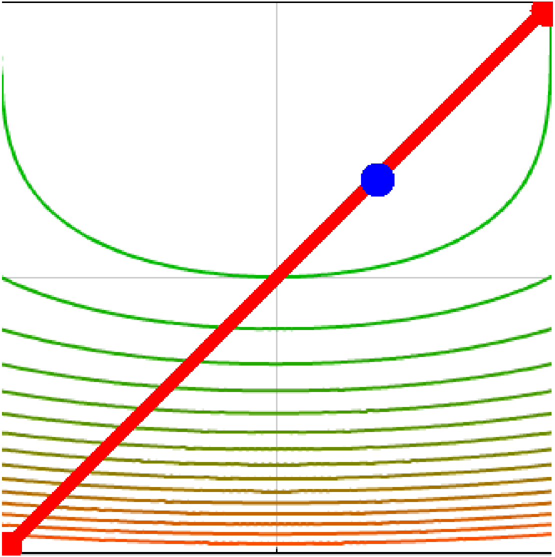
This value can be found using the bisection method or Newton's method.

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

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 $\begin{bmatrix} 1.41 & 1.41 \end{bmatrix}$. (Objective reduced from 17 to 0.289)



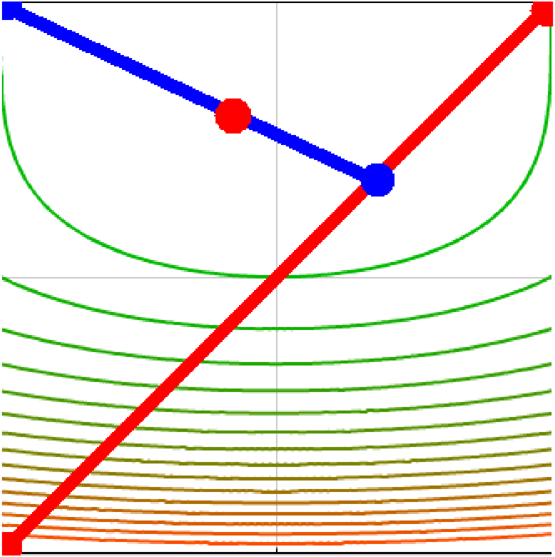
Example

(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).

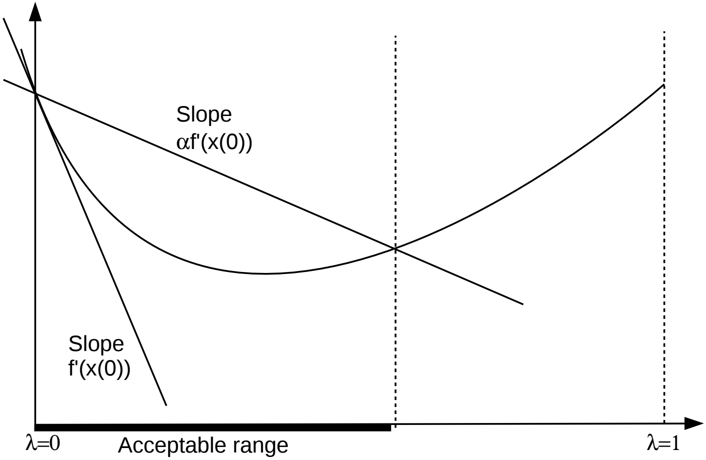
The Armijo rule does not try to find the value of λ which minimizes f, but is content to find a value of λ which reduces f “enough.”

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

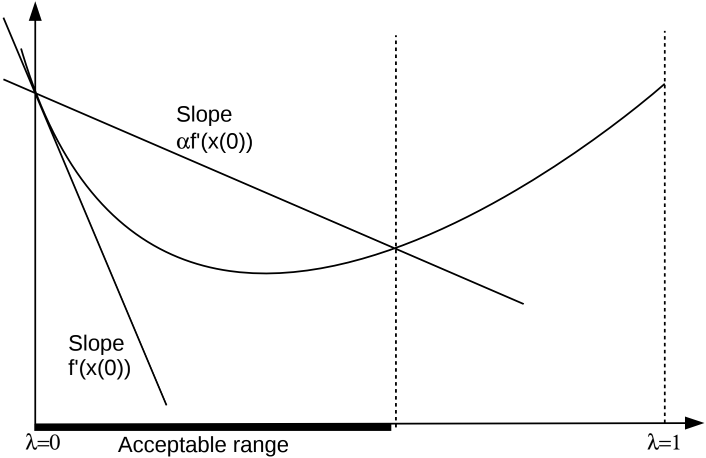
An "acceptable" λ is defined as one for which

$$\frac{f(\mathbf{x}) - f(\mathbf{x}(\lambda))}{\lambda} \geq \alpha |f'(\mathbf{x}(0))|$$

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 $\mathbf{x}^*$.



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

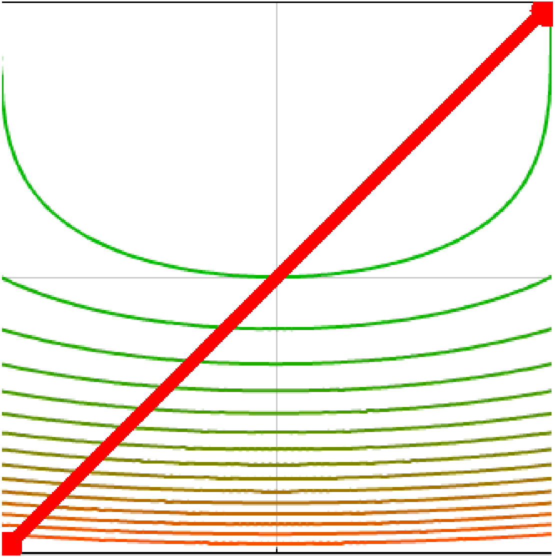
(Conditional gradient plus Armijo rule with $\alpha = 0.1$). At the initial point $\begin{bmatrix} 0 & 0 \end{bmatrix}$, the target point was $\begin{bmatrix} 2 & 2 \end{bmatrix}$, so

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

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.