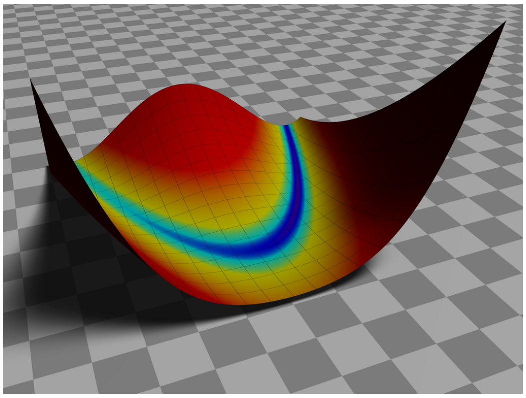
Sequential Quadratic Programming 2

Lecture 19



ME EN 575
Andrew Ning
aning@byu.edu

Outline

Inequality Constrained (Sequential) Quadratic Programming

Merit Functions

Inequality Constrained (Sequential) Quadratic Programming

Pull out a sheet of paper, and try to solve the following analytically (don't look ahead):

$$
\begin{aligned}
\text{minimize} \quad & x_1^2 - x_2 \\
\text{subject to} \quad & x_2 - 2x_1 \leq 0 \\
& 1 - x_1 - x_2 \leq 0
\end{aligned}
$$

We already know how to solve a QP with equality constraints. Let's change the inequality constraints to equality constraints.

$$
\begin{aligned}
x_2 - 2x_1 + s_1^2 &= 0 \\
1 - x_1 - x_2 + s_2^2 &= 0
\end{aligned}
$$

Form the Lagrangian:

$\mathcal{L}(x_1, x_2, \lambda_1, \lambda_2, s_1, s_2) =$
$x_1^2 - x_2 + \lambda_1(x_2 - 2x_1 + s_1^2) + \lambda_2(1 - x_1 - x_2 + s_2^2)$

Take partial derivatives and set equal to 0.

$$
\begin{aligned}
\nabla_{x_1}\mathcal{L} = 0 \Rightarrow &\quad 2x_1 - 2\lambda_1 - \lambda_2 = 0 \\
\nabla_{x_2}\mathcal{L} = 0 \Rightarrow &\quad -1 + \lambda_1 - \lambda_2 = 0 \\
\nabla_{\lambda_1}\mathcal{L} = 0 \Rightarrow &\quad x_2 - 2x_1 + s_1^2 = 0 \\
\nabla_{\lambda_2}\mathcal{L} = 0 \Rightarrow &\quad 1 - x_1 - x_2 + s_2^2 = 0 \\
\nabla_{s_1}\mathcal{L} = 0 \Rightarrow &\quad \lambda_1 s_1 = 0 \\
\nabla_{s_2}\mathcal{L} = 0 \Rightarrow &\quad \lambda_2 s_2 = 0 \\
\lambda_1 \geq 0, \lambda_2 \geq 0
\end{aligned}
$$

We don't have a full rank set of linear equations. How to solve?

$$
\begin{bmatrix} 2 & 0 & -2 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ -2 & 1 & 0 & 0 & 1 & 0 \\ -1 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \lambda_2 \\ s_1^2 \\ s_2^2 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}
$$

* If you're bothered by the appearance of s^2 just pretend that it is k where $k = s^2$.

We haven't yet used the complementarity conditions $\lambda_i s_i = 0$. There are four possibilities:

1. C1 and C2 are both inactive.
2. C1 and C2 are both active.
3. C1 is inactive and C2 is active.
4. C1 is active and C2 is inactive.

Let's try all four possibilities.

Assume both constraints are inactive and try to solve the system of equations. This means $\lambda_1 = \lambda_2 = 0$

If we move the corresponding columns the linear system is singular (no solution).

We can tell this right away because of this equation:

$$-1 + \lambda_1 - \lambda_2 = 0$$

Assume both constraints are active: $s_1 = s_2 = 0$.

This results in equation

$$\begin{bmatrix} 2 & 0 & -2 & -1 \\ 0 & 0 & 1 & -1 \\ -2 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}$$

Solution: $(x_1^*, x_2^*, \lambda_1^*, \lambda_2^*) = (1/3, 2/3, 5/9, -4/9)$

Cannot have negative Lagrange multipliers at solution, so this assumption is not possible.

Assume constraint 1 is inactive and constraint 2 is active: $\lambda_1 = 0, s_2 = 0$

Solution: $(x_1^*, x_2^*, \lambda_2^*, s_1^{2*}) = (-0.5, 1.5, -1, -2.5)$

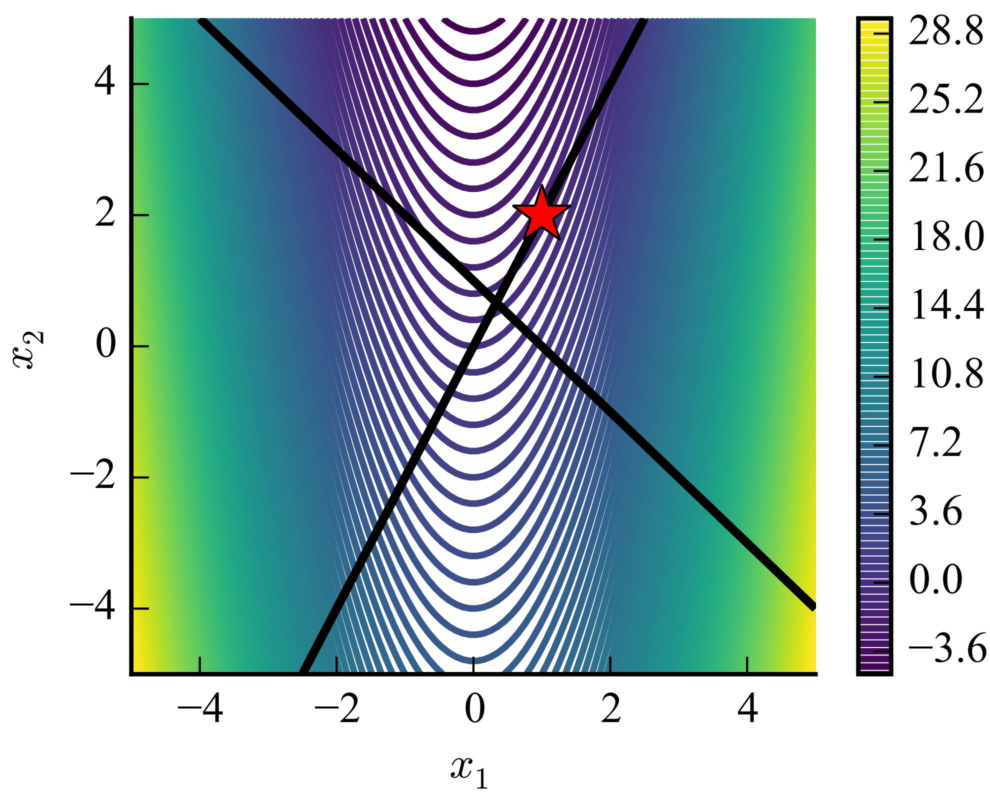
Not possible: slack and Lagrange multiplier are negative. We can also clearly see from this equation,

$$-1 + \lambda_1 - \lambda_2 = 0$$

Assume constraint 2 is active, and constraint 1 is inactive: $s_1 = 0, \lambda_2 = 0$

$$\begin{bmatrix} 2 & 0 & -2 & 0 \\ 0 & 0 & 1 & 0 \\ -2 & 1 & 0 & 0 \\ -1 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \lambda_1 \\ s_2^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}$$

Solution: $(x_1^*, x_2^*, \lambda_1^*, s_2^{2*}) = (1, 2, 1, 2)$



Clearly, this combinatorial approach is too difficult for a large number of constraints.

There are two main strategies:

- Keep track of which constraints are active: Active Set SQP methods.
- Keep the constraints feasible: Interior point methods (modern IP methods enforce feasibility only for $\lambda \geq 0$ and $s \geq 0$ and not the actual constraints).

Unfortunately we don't know which constraints are active a priori, it requires some guessing and updating as we go. Various techniques are used to handle this, but we won't get into those details.

Merit Functions

Recall, that the SQP methods gives a predicted search direction and step length: p_k.

However, we still need to perform a line search (or use a trust-region based approach). Update step: $x_{k+1} = x_k + \alpha p_k$

If a line search: what merit function do we use? Obviously, the Lagrangian would be ideal, but we don't yet know the values for the Lagrange multipliers.

The merit function for a line search need not be differentiable. We just need to find a sufficient decrease.

l_1 penalty:

$$\phi(x;\mu) = f(x) + \mu \|c_{vio}(x)\|_1$$

l_2 penalty:

$$\phi(x;\mu) = f(x) + \frac{1}{2}\mu \|c_{vio}(x)\|_2$$

Augmented Lagrangian:

$$\phi(x;\mu) = f(x) + \lambda^T c(x) + \frac{1}{2}\mu \|c_{vio}(x)\|_2$$

Filter approach (multiobjective):

$$\text{minimize:} f(x) \text{ and } h(x) = \|c_{vio}(x)\|_1$$

We have only taken a high level overview of SQP. There are lots of important details required to make these effective that we won't have time to discuss.