

Math Camp Notes: Constrained Optimization

Consider the following general constrained optimization problem:

$$\begin{aligned} \max_{x_i \in \mathbb{R}} f(x_1, \dots, x_n) \text{ subject to :} \\ g_1(x_1, \dots, x_n) \leq b_1, \dots, g_k(x_1, \dots, x_n) \leq b_k, \\ h_1(x_1, \dots, x_n) = c_1, \dots, h_k(x_1, \dots, x_n) = c_k. \end{aligned}$$

The function $f(\mathbf{x})$ is called the **objective** equation, $g(\mathbf{x})$ is called an **inequality constraint**, and $h(\mathbf{x})$ is called an **equality constraint**. Notice that this problem differs from the regular unconstrained optimization problem in that instead of finding the extrema of the curve $f(\mathbf{x})$, we are finding the extrema of $f(\mathbf{x})$ only at points which satisfy the constraints.

Example:

Maximize $f(x) = x^2$ subject to $0 \leq x \leq 1$.

Solution: We know that $f(x)$ is strictly monotonically increasing over the domain, therefore the maximum (if it exists) must lie at the largest number in the domain. Since we are optimizing over a compact set, the point $x = 1$ is the maximal number in the domain, and therefore it is the maximum.

This problem was easy because we could visualize the graph of $f(x)$ in our minds and see that it was strictly monotonically increasing over the domain. However, we see a method to find constrained extrema of functions even when we can't picture them in our minds.

Equality Constraints

One Constraint

Consider a simple optimization problem with only one constraint:

$$\begin{aligned} \max_{x \in \mathbb{R}} f(x_1, \dots, x_n) \text{ subject to :} \\ h(x_1, \dots, x_n) = c. \end{aligned}$$

Now draw level sets of the function $f(x_1, \dots, x_n)$. Since we might not be able to achieve the unconstrained extrema of the function due to our constraint, we seek to find the value of \mathbf{x} which gets us onto the highest level curve of $f(\mathbf{x})$ while remaining on the function $h(\mathbf{x})$. Notice also that the function $h(\mathbf{x})$ will be just tangent to the level curve of $f(\mathbf{x})$.

Call the point which maximizes the optimization problem \mathbf{x}^* . Since at \mathbf{x}^* the level curve of $f(\mathbf{x})$ is tangent to the curve $g(\mathbf{x})$, it must also be the case that the gradient of $f(\mathbf{x}^*)$ must be in the same direction as the gradient of $h(\mathbf{x}^*)$, or

$$\nabla f(\mathbf{x}^*) = \lambda \nabla h(\mathbf{x}^*),$$

where λ is merely a constant.

Expanding this formula, we have

$$\frac{df}{dx_1} = \lambda \frac{dh}{dx_1} \Rightarrow \frac{df}{dx_1} - \lambda \frac{dh}{dx_1} = 0$$

⋮

$$\frac{df}{dx_n} = \lambda \frac{dh}{dx_n} \Rightarrow \frac{df}{dx_n} - \lambda \frac{dh}{dx_n} = 0$$

Integrate each equation to get

$$f - \lambda h - C = 0$$

Therefore, when we derivate this equation, we get our previously stated condition on the gradient. Since this equation yields the proper first order condition for all C , it yields the proper first order condition when $C = \lambda c$. Call the equation when $C = -\lambda c$ the **Lagrangean**.

$$L(x_1, \dots, x_n, \lambda) = f(x_1, \dots, x_n) - \lambda [h(x_1, \dots, x_n) - c]$$

Taking the partial derivatives of this function will yield the first order conditions for an extrema. The reason we have $C = -\lambda c$ is so that taking the derivative of the Lagrangean will yield our equality constraint.

Example:

Maximize $f(x_1, x_2) = x_1 x_2$ subject to $h(x_1, x_2) \equiv x_1 + 4x_2 = 16$.

Solution: Form the lagrangean

$$L(x_1, x_2) = x_1 x_2 - \lambda (x_1 + 4x_2 - 16)$$

The first order conditions are

$$\frac{dL}{dx_1} = x_2 - \lambda = 0$$

$$\frac{dL}{dx_2} = x_1 - 4\lambda = 0$$

$$\frac{dL}{d\lambda} = x_1 + 4x_2 - 16 = 0$$

From the first two equations we have

$$x_1 = 4x_2.$$

Plugging this into the last equation we have that $x_2 = 2$, which in turn implies that $x_1 = 8$, and that $\lambda = 2$. This states that the only candidate for the solution is $(x_1, x_2, \lambda) = (8, 2, 2)$.

Remember that points obtained using this formula may or may not be a maximum or minimum, since the first order conditions are only necessary conditions. They only give us candidate solutions.

There is another more subtle way that this process may fail, however. Consider the case where $\nabla h(\mathbf{x}^*) = 0$, or in other words, the point which maximizes $f(\mathbf{x})$ is also a critical point of $h(\mathbf{x})$. Remember our necessary condition for a maximum

$$\nabla f(\mathbf{x}^*) = \lambda \nabla h(\mathbf{x}^*),$$

Since $\nabla h(\mathbf{x}^*) = 0$, this implies that $\nabla f(\mathbf{x}^*) = 0$. However, this the necessary condition for an unconstrained optimization problem, not a constrained one! In effect, when $\nabla h(\mathbf{x}^*) = 0$, the constraint is no longer taken into account in the problem, and therefore we arrive at the wrong solution.

Many Constraints

Consider the problem

$$\max_{x_i \in \mathbb{R}} f(x_1, \dots, x_n) \text{ subject to :}$$

$$h_1(x_1, \dots, x_n) = c_1, \dots, h_m(x_1, \dots, x_n) = c_m.$$

Lets first talk about how this might fail. As we saw for one constraint, if $\nabla h(\mathbf{x}^*) = 0$, then the constraint drops out of the equation. Now consider the Jacobian matrix, or a vector of the gradients of the different $h_i(\mathbf{x}^*)$.

$$Dh(\mathbf{x}^*) = \begin{pmatrix} \nabla h_1(\mathbf{x}^*) \\ \vdots \\ \nabla h_m(\mathbf{x}^*) \end{pmatrix} = \begin{pmatrix} \frac{dh_1(\mathbf{x}^*)}{dx_1} & \dots & \frac{dh_1(\mathbf{x}^*)}{dx_n} \\ \vdots & \ddots & \vdots \\ \frac{dh_m(\mathbf{x}^*)}{dx_1} & \dots & \frac{dh_m(\mathbf{x}^*)}{dx_n} \end{pmatrix}$$

Notice that if and of the $\nabla h_i(\mathbf{x}^*)$'s is zero, then that constraint will not be taken into account in the analysis. Also, there will be a row of zeros in the Jacobian, and therefore the Jacobian will not be full rank. The generalization of the condition that $\nabla h(\mathbf{x}^*) \neq 0$ for the case when $m = 1$ is that the Jacobian matrix must

be full rank. Otherwise, one of the constraints is not being taken into account, and the analysis fails. We call this condition the **non-degenerate constraint qualification (NDCQ)**.

The lagrangean for the multi-constraint optimization problem is

$$L(x_1, \dots, x_n, \lambda) = f(x_1, \dots, x_n) - \sum_{i=1}^m \lambda_i [h_i(x_1, \dots, x_n) - c_i]$$

And therefore the necessary conditions for a maximum are

$$\begin{aligned} \frac{dL}{dx_1} = 0, \dots, \frac{dL}{dx_n} = 0 \\ \frac{dL}{d\lambda_1} = 0, \dots, \frac{dL}{d\lambda_n} = 0 \end{aligned}$$

Example:

Maximize $f(x, y, z) = xyz$ subject to $h_1(xyz) \equiv x^2 + y^2 = 1$ and $h_2(xyz) \equiv x + z = 1$.

Solution: First let us form the Jacobian matrix

$$D\mathbf{h}(x, y, z) = \begin{pmatrix} 2x & 2y & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Notice that this is only singular if both $x = y = 0$. However, if this is the case, then our first constraint fails to hold. Therefore, the matrix is nonsingular for all points in the domain, and so we don't need to worry about the NDCQ. Form the lagrangean

$$L(x, y, z) = xyz - \lambda_1 (x^2 + y^2 - 1) - \lambda_2 (x + z - 1)$$

The first order conditions are

$$\begin{aligned} \frac{\partial L}{\partial x} &= yz - 2\lambda_1 x - \lambda_2 = 0 \\ \frac{\partial L}{\partial y} &= xz - 2\lambda_1 y = 0 \\ \frac{\partial L}{\partial z} &= xy - \lambda_2 = 0 \\ \frac{\partial L}{\partial \lambda_1} &= x^2 + y^2 - 1 = 0 \\ \frac{\partial L}{\partial \lambda_2} &= x + z - 1 = 0 \end{aligned}$$

We can solve the second and third equations for λ_1 and λ_2

$$\lambda_1 = \frac{xz}{2y}, \quad \lambda_2 = xy$$

and plug these into the first equation to get

$$yz - 2\frac{x^2z}{2y} - xy = 0$$

Multiply both sides by y

$$y^2z - x^2z - xy^2 = 0$$

Now we want to solve the two constraints for y and z in terms of x , plug them into this equation, and get one equation in terms of x

$$(1 - x^2)(1 - x) - x^2(1 - x) - x(1 - x^2) = 0$$

$$(1-x)[(1-x^2)-x^2-x(1+x)] = 0$$

$$(1-x)[1-3x^2-x] = 0$$

This yields $x = \left\{1, \frac{-1+\sqrt{13}}{6}, \frac{-1-\sqrt{13}}{6}\right\}$. Let's analyze $x = 1$ first. From the second constraint we have that $z = 0$, and from the first constraint we have that $y = \{1, -1\}$. However, this violates the equation

$$y^2z - x^2z - xy^2 = 0,$$

and therefore $x = 1$ cannot be a solution.

Plugging in the other values, we get four candidate solutions

$$x = .4343, y = \pm 0.9008, z = 0.5657$$

$$x = -.7676, y = \pm 0.6409, z = 1.7676$$

Inequality Constraints

One Constraint

Consider the problem

$$\max_{x_i \in \mathbb{R}} f(x_1, \dots, x_n) \text{ subject to :}$$

$$g_1(x_1, \dots, x_n) \leq b_1.$$

In this case the solution is not *constrained* to the curve, but merely *bounded* by it. Solving the constrained optimization problem with inequality constraints is the same as solving them with equality constraints, but with more conditions.

In order to understand the new conditions, imagine the graph of the level sets which we talked about before. Instead of being constrained to the function $g(\mathbf{x})$, the domain is now bounded by it instead. However, the *boundary* of the function is still the same as before. Notice that there is still a point where the boundary is tangent to the highest level set on $g(\mathbf{x})$. The question now is whether the boundary is *binding* or *not binding*.

Case 1: Binding This is the case where the unconstrained extrema lies outside of the domain. In other words, the inequality constrains us from reaching the extrema of $f(\mathbf{x})$. In this case, the gradient of $f(\mathbf{x})$ is going to point in the steepest direction up the graph. The gradient of $g(\mathbf{x})$ points to the set $g(\mathbf{x}) \geq b$ (since it points in the direction of increasing $g(\mathbf{x})$). Therefore, if the constraint is binding, then $\nabla g(\mathbf{x})$ is pointing in the same direction as $\nabla f(\mathbf{x})$, which implies that $\lambda \geq 0$.

Case 2: Not Binding This is the case where the unconstrained extrema lies inside the domain. In other words, the inequality does not constrain us from reaching the extrema of $f(\mathbf{x})$. Remember again that the gradient of $f(\mathbf{x})$ is going to point in the steepest direction up the graph. The gradient of $g(\mathbf{x})$ points to the set $g(\mathbf{x}) \geq b$ (since it points in the direction of increasing $g(\mathbf{x})$). This time, however, the gradient of $g(\mathbf{x})$ will point away from the gradient of $f(\mathbf{x})$ (since we have passed the optimum, the gradient of $f(\mathbf{x})$ is now pointing in the opposite direction). Therefore, if the constraint is not binding, then $\nabla g(\mathbf{x})$ is pointing in opposite direction as $\nabla f(\mathbf{x})$, which implies that $\lambda \leq 0$.

As we can see, it does not matter whether the constraint binds or does not bind, the lagrangean multiplier must always be greater than or equal to 0. Therefore, a new set of conditions are

$$\lambda_i \geq 0 \forall i.$$

If the constraint binds, then that means the solution will lie somewhere on the constraint. Therefore, we can replace the inequality constraint with an equality constraint:

$$\max_{x_i \in \mathbb{R}} f(x_1, \dots, x_n) \text{ subject to :}$$

$$g_1(x_1, \dots, x_n) = b_1.$$

This implies that $g_1(x_1, \dots, x_n) - b_1 = 0$.

If the constraint is not binding, then there is no use in having it in the analysis. Notice that if $\lambda = 0$, then the constraint drops out just as we wish. Therefore, we have a new set of conditions called the **complementary slackness conditions**

$$[g_1(x_1, \dots, x_n) - b_1] \lambda = 0.$$

This works because if the constraint is binding, then $g_1(x_1, \dots, x_n) - b_1 = 0$, and if the constraint is not binding, then $\lambda = 0$.

Many Constraints

Consider the problem:

$$\max_{x_i \in \mathbb{R}} f(x_1, \dots, x_n) \text{ subject to :}$$

$$g_1(x_1, \dots, x_n) \leq b_1, \dots, g_m(x_1, \dots, x_n) \leq b_m.$$

In order to understand the new NDCQ, we must realize that if a constraint does not bind, we don't care if it drops out of the equation. The point of the NDCQ was to ensure that binding constraints do not drop out. Therefore, the NDCQ with inequality constraints is the same as the equality constraints, except for the fact that we only care about the Jacobian matrix of the binding constraints, or the Jacobian for the constraints with $\lambda_i > 0$.

The following first order conditions will tell us the candidate points for extrema

$$\frac{dL}{dx_1} = 0, \dots, \frac{dL}{dx_n} = 0$$

$$g_1(x_1, \dots, x_n) \leq b_1, \dots, g_m(x_1, \dots, x_n) \leq b_m$$

$$\lambda_1 \geq 0, \dots, \lambda_m \geq 0$$

$$[g_1(x_1, \dots, x_n) - b_1] \lambda_1 = 0, \dots, [g_m(x_1, \dots, x_n) - b_m] \lambda_m = 0$$

Homework

1. In chapter 18 of Simon and Blume, numbers 3, 6, 7, 10, 11, and 12.