Constrained Minimization

In this lecture, we will focus on minimizing an objective function subject to constraints. There are numerous ways to introduce constraints; we will focus on the so-called *functional form*, in which the constraints are level sets of functions. The general constrained optimization problem we will consider is

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad \text{s.t.} \quad \begin{cases} g_i(\mathbf{x}) \leq 0 : i = 1, \ldots, m \\ h_j(\mathbf{x}) = 0 : j = 1, \ldots, k. \end{cases}$$

"s.t." is spelled out as "subject to"; the set of constraints $g_i(\mathbf{x}) \leq 0$ is called *inequality constraints*, while the set $h_j(\mathbf{x}) = 0$ is called *equality constraints*. We will concentrate mainly on the former case, since every $h(\mathbf{x}) = 0$ can be translated into two inequality constraints of the form $h(\mathbf{x}) \leq 0$ and $-h(\mathbf{x}) \leq 0$.

The set of points in $\mathbb{R}^n$ satisfying all constraints is called the *feasible set*, and a point in it is called a *feasible point*. If at a solution point $\mathbf{x}^*$ an inequality constraints is satisfied with equality $g_i(\mathbf{x}^*) = 0$, it is said to be *active*; if the inequality is strict, $g_i(\mathbf{x}^*) < 0$, the constraint is said to be *inactive*. Equality constraints are, obviously, always active. A point for which all inequality constraints are passive is said *strictly feasible* (geometrically, it is a point in the interior of the feasible set).

1 Optimality conditions

For unconstrained minimization problems, we have seen the necessary first-order optimality condition $\nabla f(\mathbf{x}^*) = 0$. This is no longer true, in general, for constrained problems, as the unconstrained minimizer might be an infeasible point. If only one constraint $g(\mathbf{x})$ is active, the constrained minimizer will be a point on a level line of $f(\mathbf{x})$ that is tangent to the level line of the constraint $g(\mathbf{x}) = 0$. Geometrically, this means that at the constrained minimizer $\mathbf{x}^*$, the gradients of f and g are collinear, which can be expressed as

$$\nabla f(\mathbf{x}^*) + \lambda \nabla g(\mathbf{x}^*) = 0.$$

Furthermore, the gradient of the objective indicating an increase direction has to point to the interior of the feasible set; on the other hand, the gradient of the constraint always points in the opposite direction. This means that $\lambda \geq 0$. In a more general case, when several constraints are active, $\nabla f(\mathbf{x}^*)$ is collinear with a positive linear combination of the $\nabla g_i(\mathbf{x}^*)$'s.

In case there are equality constraints, we can write each of them as two inequality constraint $h_j(\mathbf{x}) \leq 0$ and $-h_j(\mathbf{x}) \leq 0$; since both are active, the linear combination of the

gradients $\lambda_1 \nabla h_j(\mathbf{x}^*) - \lambda_2 \nabla h_j(\mathbf{x}^*)$ can be simply written as $\lambda \nabla h_j(\mathbf{x}^*)$ without non-negativity restrictions on λ.

We can summarize these observations as the following theorem:

Theorem 1 (Karush-Kuhn-Tucker first-order necessary conditions)**.** *Let* $\mathbf{x}^*$ *be a* regular *constrained minimizer (regular means that the gradients of the active constraints are linearly independent). Then, there exist* $\boldsymbol{\lambda}^* \in \mathbb{R}^m_+$ *and* $\boldsymbol{\mu}^* \in \mathbb{R}^k$ *such that*

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(\mathbf{x}^*) + \sum_{j=1}^{k} \mu_i^* \nabla h_j(\mathbf{x}^*) = 0,$$

where

$$\sum_{i=1}^{m} \lambda_i^* g_i(\mathbf{x}) = 0.$$

The last condition is equivalent to saying that $\lambda_i^* = 0$ for inactive constraints (for which $g_i(\mathbf{x}) < 0$). This condition is usually known as *complementary slackness* (we defer the explanation of this name).

The KKT conditions are *sufficient* of the objective $f(\mathbf{x})$ is convex, the inequality constraints are C^1 and convex, and the equality constraints are affine. Otherwise, more complicated second-order sufficient conditions have to be used.

An equivalent form to write the KKT conditions is by constructing the *Lagrangian* function

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{k} \mu_i h_j(\mathbf{x}) = f(\mathbf{x}) + \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{g}(\mathbf{x}) + \boldsymbol{\mu}^{\mathrm{T}} \mathbf{h}(\mathbf{x})$$

The condition

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = 0$$

is equivalent to the first order KKT condition.

2 Penalty methods

Focusing on problems with inequality constraints, observe that there is a trivial way to convert them into unconstrained problems, by aggregating to the objective function a *penalty function* taking the value of 0 inside the feasible set and ∞ otherwise. This can be expressed as

$$\min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) \equiv f(\mathbf{x}) + \sum_{i=1}^{m} \varphi_{\text{ideal}}(g_i(\mathbf{x})),$$

where

$$\varphi_{\text{ideal}}(t) = \begin{cases} 0 & : \quad t \leq 0 \\ \infty & : \quad t > 0. \end{cases}$$

The function $F(\mathbf{x})$ is called a *penalty aggregate*.

Since the penalty φ_{ideal} is not tractable with the unconstrained optimization algorithms we have seen so far, we will substitute it with nicer smooth penalty functions. We define a prototype penalty function $\varphi(t)$ by requiring

1. φ is convex and monotonically increasing
2. Increasingly steep for infeasible points: $\lim_{t\to\infty} \varphi'(t) = \infty$
3. Increasingly flat for feasible points: $\lim_{t\to-\infty} \varphi'(t) = 0$
4. $\varphi(0) = 0$
5. (Arbitrary normalization) $\varphi'(0) = 1$.

Using this prototype function, we define a family of penalty functions $\varphi_p(t)$ with the parameter $p > 0$ such that in the limit $p \to \infty$, $\varphi_p \to \varphi_{\text{ideal}}$. For example, one way to define such a family is

$$\varphi_p(t) = \frac{1}{p}\varphi(pt).$$

Note that $\varphi_p'(t) = \varphi'(pt)$, meaning that for the same value of $t > 0$, φ_p becomes increasingly bigger as p grows, while for the same value of $t < 0$, φ_p becomes increasingly closer to 0.

The following choices are common for the prototype penalty function:

1. *Exponential*: $\varphi(t) = e^t - 1$. The disadvantage of this penalty is that standard floating point arithmetics will saturate at infinity for relatively modest values of $t \approx 100$. Special care has to be taken in the numerical optimization algorithms to avoid infinite values.

2. *Quadratic-logarithmic* is a penalty function growing quadratically for positive values, and decreasing very slowly as a logarithm for negative ones. An example of such a function is

$$\varphi(t) = \begin{cases} \dfrac{t^2}{2} + t & : \quad t \le -\dfrac{1}{2} \\ -\dfrac{1}{4}\log(-2t) - \dfrac{3}{8} & : \quad t > -\dfrac{1}{2}. \end{cases}$$

   The particular choice of the coefficients guarantees that the function is C^2 at $t = -\dfrac{1}{2}$.

Exercise 1. *Show that the above functions are C^2 satisfying the penalty function conditions.*

Equipped with the family of penalty functions, we can formulate the penalty aggregate for every p,

$$F_p(\mathbf{x}) \equiv f(\mathbf{x}) + \sum_{i=1}^{m} \varphi_p(g_i(\mathbf{x})).$$

Penalty methods start with a moderate value of p, minimize $F_p(\mathbf{x})$, increase p, and repeat the process. This is summarized as the following iterative procedure:

input : function f; inequality constraints $g_1, \ldots, g_m$, initial point $\mathbf{x}_0$; parameter $\beta > 1$
output: (approximate) constrained local minimizer $\mathbf{x}^*$ of f s.t. $g_i \leq 0$
Start with a small p_0
for $k = 1, 2, \ldots,$ *until convergence* **do**
  Find $\mathbf{x}_k = \arg\min_{\mathbf{x}} F_{p_{k-1}}(\mathbf{x})$ using any unconstrained minimization algorithm initialized with $\mathbf{x}_{k-1}$
  Increase penalty parameter $p_k = \beta p_{k-1}$
end
Return $\mathbf{x}^* = \mathbf{x}^k$

Algorithm 1: Penalty method

The typical choice for β is $2 \div 10$. Outer iterations are typically stopped when the constraint violation is below a preset threshold,

$$\max\{g_i(\mathbf{x})\} \leq \epsilon$$

and the change in the function value $f(\mathbf{x}_k)$ or the change in the argument $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|$ is sufficiently small. Another commonly used stopping condition is when p_k reaches some $p_{\max}$, around $10^5 \div 10^6$.

When the problem has equality constraints, they can be trivially converted into inequality constraints. However, it is more efficient to introduce them into the penalty aggregate using a special penalty for equality constraints. The ideal penalty for equality constraint is

$$\psi_{\text{ideal}}(t) = \begin{cases} 0 & : \quad t = 0 \\ \infty & : \quad t \neq 0. \end{cases}$$

As before, it can be replaced by a family of penalty functions $\psi_p(t) = \dfrac{1}{p}\psi(pt)$ derived from a prototype function, e.g., $\psi(t) = t^2$, growing increasingly fast to infinity for $t \neq 0$. Other constructions of the parametric family are, for example, $\psi_p(t) = p\psi(t)$ (this construction is equivalent to the previous one for the quadratic penalty).

3 Barrier methods

In some problems, the objective outside the feasible set is numerically ill-behaved or even undefined (think of $f(x) = -log(x)$ subject to $x > 0$). In such cases, a special types of penalties are built ensuring $\lim_{t \uparrow 0} \varphi'(t) = \infty$ and $\varphi(t) = \infty$ for $t \geq 0$. Such penalties are called *barrier functions*, and methods involving them *barrier methods*. An example of a barrier function is $\varphi(t) = -\log(-t)$.

The advantage of barrier methods is that they always produce a strictly feasible solution. The disadvantage is that special precaution has to be taken e.g. in the line search to

guarantee that no infeasible points are substituted into the barrier aggregate (for example, if Newton's method is used with inexact line search, one has to decrease the step sufficiently to ensure that the resulting point is feasible, and only then to start applying the standard Armijo rule). Also, barrier methods must be initialized with a feasible point.

4 Derivation of KKT conditions via penalty methods

There is a deep connection between penalty methods and KKT conditions, that we will now try to illustrate. Let us denote $\mathbf{x}_p^* = \arg\min_{\mathbf{x}} F_p(\mathbf{x})$. From the first-order necessary condition on this unconstrained problem, we have

$$0 = \nabla F_p(\mathbf{x}_p^*) = \nabla f(\mathbf{x}_p^*) + \sum_{i=1}^{m} \varphi_p'(g_i(\mathbf{x}_p^*))\nabla g_i(\mathbf{x}_p^*).$$

Denoting $\lambda_i^p = \varphi_p'(g_i(\mathbf{x}_p^*))$, we have

$$0 = \nabla F_p(\mathbf{x}_p^*) = \nabla f(\mathbf{x}_p^*) + \sum_{i=1}^{m} \lambda_i^p \nabla g_i(\mathbf{x}_p^*) = \nabla_{\mathbf{x}} L(\mathbf{x}_p^*, \boldsymbol{\lambda}^p).$$

Note that since φ_p is monotonically increasing $\lambda_i^p \geq 0$.

In the limit $p \to \infty$, $\varphi_p \to \varphi_{\text{ideal}}$, and $\mathbf{x}_p^* \to \mathbf{x}^*$. It is more elaborate to show that $\boldsymbol{\lambda}^p \to \boldsymbol{\lambda}^*$. First, observe that for inactive constraints, $g_i(\mathbf{x}_p^*) < 0$, $\lambda_i^p = \varphi_p'(g_i(\mathbf{x}_p^*))$ becomes flatter as p grows, resulting in $\lambda_i^p \to \lambda_i^* = 0$. For a sufficiently large p we can therefore neglect the sum $\lambda_i^p \nabla g_i(\mathbf{x}_p^*)$ over the inactive constraint, remaining with

$$-\nabla f(\mathbf{x}_p^*) = \mathbf{G}_{\text{a}}(\mathbf{x}_p^*)\boldsymbol{\lambda}_{\text{a}}^p,$$

where $\boldsymbol{\lambda}_{\text{a}}^p$ is subvector of $\boldsymbol{\lambda}^p$ corresponding to the active constraints, and $\mathbf{G}_{\text{a}}(\mathbf{x}_p^*)$ is a matrix whose columns are the gradients of the corresponding active constraints g_i at $\mathbf{x}_p^*$.

When the gradients of the active constraints are linearly independent, the matrix $\mathbf{G}_{\text{a}}$ is full rank and as the result a small change in $\mathbf{x}_p^*$ resulting in a small perturbation in $\mathbf{G}_{\text{a}}$ results in a small change in the solution $\boldsymbol{\lambda}_{\text{a}}^p$. As the result, the limit $\boldsymbol{\lambda}_{\text{a}}^p \to \boldsymbol{\lambda}^*$ exists.

Exercise 2. *Prove formally the existence of the above limit.*

5 Augmented Lagrangian

One of the major disadvantages of penalty methods is the need to increase the parameter p to very large values in order to obtain accurate solutions. This increase the derivatives of the penalty aggregate, making its unconstrained optimization challenging to most numerical methods. The family of *augmented Lagrangian* algorithms overcomes this limitation.

In order to construct augmented Lagrangian, let us first consider problems with inequality constraints only, for which we construct a new family of *penalty-multiplier functions*, $\varphi_{p,\lambda}(t)$, similar to our previous family of penalty function with the distinction that now we also demand $\varphi'(0) = \lambda$. The analog of the penalty aggregate is now played by the augmented Lagrangian function

$$F_p(\mathbf{x}; \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^{m} \varphi'_{p,\lambda_i}(g_i(\mathbf{x}))\nabla g_i(\mathbf{x}).$$

As before, unconstrained minimization is used to find

$$\mathbf{x}^*_{p,\boldsymbol{\lambda}} = \arg\min_{\mathbf{x}} F_p(\mathbf{x}; \boldsymbol{\lambda}).$$

If for some of the constraints we have $g_i(\mathbf{x}^*_p) \geq 0$, we will try to modify $F_p(\mathbf{x}; \boldsymbol{\lambda})$ in such a way to shift the value of $g_i(\mathbf{x}^*_p)$ as close as possible to zero and reduce the constraint violation. This can be done by changing λ_i to $\lambda'_i = \varphi'_{p,\lambda_i}(g_i(\mathbf{x}^*_{p,\boldsymbol{\lambda}}))$ in such a way that $\varphi'_{p,\lambda'_i}(0) = \varphi'_{p,\lambda_i}(g_i(\mathbf{x}^*_{p,\boldsymbol{\lambda}}))$. Note that this reduction in the constraint violation is achieved without changing p at all – in fact, augmented Lagrangian methods produce very accurate solutions with a constant or only moderately increasing p.

The procedure is summarized as the following algorithm:

input : function f; inequality constraints $g_1, \ldots, g_m$, initial point $\mathbf{x}_0$; parameter $\beta > 1$
output: (approximate) constrained local minimizer $\mathbf{x}^*$ of f s.t. $g_i \leq 0$
Start with a small p_0 and an initial estimate of the multipliers $\boldsymbol{\lambda}_0$ ($\boldsymbol{\lambda}_0 = 1$ if no better guess is available)
for $k = 1, 2, \ldots,$ *until convergence* **do**
  Find $\mathbf{x}_k = \arg\min_{\mathbf{x}} F_{p_{k-1}}(\mathbf{x}; \boldsymbol{\lambda}_{k-1})$
  Update multipliers $\boldsymbol{\lambda}_k = \varphi'_{p,\boldsymbol{\lambda}_{k-1}}(\mathbf{x}_k)$
  Optional safeguard: restrict $\boldsymbol{\lambda}_k = \min\left\{\max\left\{\boldsymbol{\lambda}_k, \frac{1}{3}\boldsymbol{\lambda}_{k-1}\right\}, 3\boldsymbol{\lambda}_{k-1}\right\}$
  Optional: update penalty parameter $p_k = \min\{\beta p_{k-1}, p_{\max}\}$
end
Return $\mathbf{x}^* = \mathbf{x}^k$

Algorithm 2: Augmented Lagrangian method

The update of the penalty parameter is not necessary at all, but increasing p mildly until it reaches $p_{\max} \approx 100 \div 1000$ (the exact setting is very problem-dependent!) usually improves convergence speed. The safeguard on the update of the multipliers prevents too rapid decrease of $\boldsymbol{\lambda}$ to zero that might negatively affect the convergence speed.

In case of equality constraints, we can trivially convert them into inequality constraints. However, it is numerically more efficient to handle equality constraints using an appropriate family of penalty-multiplier functions. Recall that in the penalty method we used a quadratic penalty function of the form $\psi_p(t) = pt^2$. In order to convert it to a penalty-multiplier function, we have to enforce $\psi'_{p,\mu}(0) = \mu$, which is impossible with this particular choice.

The typical solution to this problem is the addition of a linear term of the form

$$\psi'_{p,\mu}(0) = pt^2 + \mu t.$$

Let us assume to be given the optimal Lagrange multiplier $\boldsymbol{\lambda}^*$ at the solution $\mathbf{x}^*$ of the constrained problem. By virtue of our construction of the penalty-multiplier functions, for an active constraint $g_i(\mathbf{x}^*) = 0$, $\varphi'_{p,\lambda_i^*}(g_i(\mathbf{x}^*)) = \varphi'_{p,\lambda_i^*}(0) = \lambda_i^*$. In case of an inactive constraint, $\lambda_i^* = 0$, meaning that the slope of φ_{p,λ_i^*} is zero at $t \leq 0$. Since $g_i(\mathbf{x}^*) < 0$, we have $\varphi'_{p,\lambda_i^*}(g_i(\mathbf{x}^*)) = 0 = \lambda_i^*$. Combining these results,

$$F_p(\mathbf{x}^*; \boldsymbol{\lambda}^*) = f(\mathbf{x}^*) + \sum_{i=1}^{m} \varphi'_{p,\lambda_i^*}(g_i(\mathbf{x}^*)) \nabla g_i(\mathbf{x}^*) = \nabla L_{\mathbf{x}}(\mathbf{x}^*, \boldsymbol{\lambda}^*).$$

As the result,

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} F_p(\mathbf{x}; \boldsymbol{\lambda}^*)$$

for any p (not necessarily $p \to \infty$). This is a very strong property of augmented Lagrange algorithms, that makes it more advantageous over penalty methods.