Constrained Optimization

Consider the following optimization problem:

$$\min f(z), \quad \text{subj to } c(z) \geq 0. \tag{1}$$

Assume both f and c are differentiable. A point z is called a **feasible point** for (1), if it satisfies the constraint, i.e., $c(z) \geq 0$. We need to differentiate two types of feasible points: $c(z) > 0$ and $c(z) = 0$.

- If $c(z) > 0$, then we say that the inequality constraint is **inactive**. Due to the continuity of $c(z)$, there exists a small neighborhood (e.g., a circle) around z and all points in that neighborhood are feasible points. That is, z is an interior point of the feasible region.
- If $c(z) = 0$, then we say that the inequality constraint is **active**. That is, z is on the boundary of the feasible region.

Recall that $-\nabla_z f(z)$ represents a direction that can decrease the value of f. Suppose z^* is a local minimizer (or one of the local minimizers) of (1).

- If $c(z^*) > 0$, then $-\nabla_z f(z^*) = 0$, i.e., at z^* there is no direction that can decrease the value of f.
- If $c(z^*) = 0$, then $-\nabla_z f(z^*) = -\lambda \nabla_z c(z^*)$ where $\lambda \geq 0$ is any non-negative number, i.e., the only direction that can decrease the value of f is a *forbidden* direction since it's parallel to the direction that would decrease the value of $c(\cdot)$. Since z^* is at the boundary of $c(z) \geq 0$, moving toward a direction that decreases $c(\cdot)$ would leave the feasible region.

We can summarize the above conditions for a local minimizer z^* as

$$\nabla_z f(z^*) = \lambda \nabla_z c(z^*), \tag{2}$$

$$c(z^*) \geq 0, \quad \lambda \geq 0, \quad \text{and} \quad \lambda c(z^*) = 0. \tag{3}$$

Equation (2) can be replaced by $\nabla_z L(z, \lambda) = 0$, where

$$L(z, \lambda) = f(z) - \lambda c(z)$$

is the *Lagrangian* associated with problem (1), and λ is the *Lagrange multiplier*. The last equality in (3) is the *Complementarity Condition*: λ and $c(z^*)$ cannot be non-zero simultaneously.

Consider a more general constrained optimization problem where we have multiple inequality constraints:

$$\begin{array}{ll} \min & f(z) \\ \text{subj to} & c_i(z) \geq 0, \quad i \in I, \end{array}$$

Then the KKT Conditions, the first order necessary conditions for a local minimizer z^*, are

$$\begin{aligned} \nabla_z f(z^*) &= \sum_i \lambda_i c_i(z^*), \\ c_i(z^*) &\geq 0, \ \forall i \\ \lambda_i &\geq 0, \ \forall i \\ \lambda_i c_i(z^*) &= 0, \ \forall i. \end{aligned}$$

The first equation (above) can be replaced by $\nabla_z L(z, \lambda) = 0$, where

$$L(z, \lambda) = f(z) - \sum_i \lambda_i c_i(z)$$

is the *Lagrangian* and λ_i's are the *Lagrange multipliers*. The Complementarity Condition implies that λ_i and $c_i(z^*)$ can't be non-zero simultaneously.

Convex Programming Problem

When $f(z)$ is convex and $c_i(z)$ is concave, then

$$\begin{array}{ll} \min\limits_z & f(z) \\ \text{subj to} & c_i(z) \geq 0, \quad i \in I, \end{array} \tag{4}$$

is a convex programming problem (minimizing a convex function on a convex set).

Some facts about convex programming problems:

- Every local solution z^* is a global solution and the set of global solutions is convex.
- The KKT conditions are sufficient and necessary for a global solution.
- The following duality holds: if (z^*, λ^*) solve the **primal** problem (4), they also solve the **dual** problem

$$\begin{array}{ll} \max\limits_{z,\lambda} & L(z,\lambda) \\ \text{subject to} & \nabla_z L(z,\lambda) = 0, \quad \lambda_i \geq 0. \end{array}$$

Linear SVM

The separable case.

- The Primal:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\beta},\beta_0} & \frac{1}{2}\|\boldsymbol{\beta}\|^2 \\ \text{subject to} & y_i(\mathbf{x}_i \cdot \boldsymbol{\beta} + \beta_0) - 1 \geq 0 \end{array}$$

- Lagrange function $L(\boldsymbol{\beta}, \beta_0, \lambda_1, \ldots, \lambda_n)$:

$$L = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_i \lambda_i y_i(\mathbf{x}_i^t \boldsymbol{\beta} + \beta_0) + \sum_i \lambda_i$$

- KKT conditions:

$$\begin{aligned} \nabla_{\boldsymbol{\beta}} L = \mathbf{0} \Longrightarrow \quad \boldsymbol{\beta} &= \sum_i \lambda_i y_i \mathbf{x}_i & (5) \\ \nabla_{\beta_0} L = 0 \Longrightarrow \quad \sum \lambda_i y_i &= 0 & (6) \\ \lambda_i &\geq 0 \\ y_i(\mathbf{x}_i \cdot \boldsymbol{\beta} + \beta_0) - 1 &\geq 0 \\ \lambda_i \Big[y_i(\mathbf{x}_i \cdot \boldsymbol{\beta} + \beta_0) - 1 \Big] &= 0 \end{aligned}$$

Using (5) and (6), we can rewrite the Lagrange function L as

$$\begin{aligned} L &= \frac{1}{2}\Big(\sum_i \lambda_i y_i \mathbf{x}_i\Big)^2 - \sum_i \lambda_i y_i \mathbf{x}_i \Big(\sum_j \lambda_j y_j \mathbf{x}_j\Big) + \sum_i \lambda_i \\ &= \sum_i \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j), \end{aligned}$$

where $\mathbf{x}_i \cdot \mathbf{x}_j$ denotes the inner product between two vectors, which is equal to $\mathbf{x}_i^t \mathbf{x}_j$.

- The Dual

$$\begin{aligned} \max_{\lambda_{1:n}} \quad & \sum \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ \text{subj to} \quad & \sum \lambda_i y_i = 0, \ \lambda_i \geq 0 \end{aligned}$$

The affine constraint $\sum \lambda_i y_i = 0$ can be eliminated. Define $\tilde{\mathbf{x}} = (\mathbf{x}, 1)$. Then

$$\begin{aligned} \sum_{i,j} \lambda_i \lambda_j y_i y_j (\tilde{\mathbf{x}}_i \cdot \tilde{\mathbf{x}}_j) &= \sum_{i,j} \lambda_i \lambda_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j + 1) \\ &= \sum_{i,j} \lambda_i \lambda_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) + \big(\sum_i \lambda_i y_i\big)^2 \\ &= \sum_{i,j} \lambda_i \lambda_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j). \end{aligned}$$

The dual can be expressed as

$$\min_{\boldsymbol{\lambda}} \Big[\frac{1}{2}\boldsymbol{\lambda}^t K \boldsymbol{\lambda} - \mathbf{1}^t \boldsymbol{\lambda}\Big], \quad \text{subject to } \lambda_i \geq 0,$$

where K is an $n \times n$ matrix with $K_{ij} = y_i y_j (\tilde{\mathbf{x}}_i \cdot \tilde{\mathbf{x}}_j)$. Although both are convex quadratic optimization problems, the dual is easier to solve than the primal since the constraints are just bound constraints. For example, the dual can be solved by a coordinate descent algorithm[1]. Note that the optimization for a single λ_i is in closed form.

[1] "A Dual Coordinate Descent Method for Large-scale Linear SVM" https://www.csie.ntu.edu.tw/~cjlin/papers/cddual.pdf.

Another advantage of working with the dual is that we can use the kernel trick to solve for nonlinear SVM using the same algorithm.

The non-separable case.

- The Primal

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \gamma \sum \xi_i \\ \text{subject to} \quad & y_i(\mathbf{x}_i \cdot \boldsymbol{\beta} + \beta_0) - 1 + \xi_i \geq 0, \\ & \xi_i \geq 0 \end{aligned}$$

- The Dual

$$\begin{aligned} \max_{\lambda_i} \quad & \sum \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ \text{subject to} \quad & \sum \lambda_i y_i = 0, \; 0 \leq \lambda_i \leq \gamma. \end{aligned}$$

Kernels

Consider a symmetric bivariate function $K(\mathbf{x}, \mathbf{x}')$ defined on $\mathbb{R}^p \times \mathbb{R}^p$. Can any symmetric bivariate function $K(\mathbf{x}, \mathbf{x}')$ be a kernel? **The answer is No.** Not every symmetric bivariate function $K(\mathbf{x}, \mathbf{x}')$ can be written as $K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$.

A symmetric bivariate function $K(\cdot, \cdot)$ is said to be positive semidefinite (psd), if for any $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^p$ and any real numbers $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$,

$$\sum_i \sum_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \tag{7}$$

where m is any positive integer. Then by *Mercer's theorem*, K has the following eigen decomposition

$$K(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{\infty} d_j \phi_j(\mathbf{x}) \phi_j(\mathbf{x}'),$$

where d_j's are decreasing non-negative eigenvalues and ϕ_j's are a set of orthonormal eigenfunctions. So if we define a mapping

$$\Phi(\mathbf{x}) = \left(\sqrt{d_1}\phi_1(\mathbf{x}), \sqrt{d_2}\phi_2(\mathbf{x}), \cdots\right),$$

then $K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$.

Given $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^p$, we can construct an $m \times m$ matrix $\mathbf{K}$ (known as the Gram matrix) with its (i, j)th entry being $K(\mathbf{x}_i, \mathbf{x}_j)$. Then condition (7) implies that for any m points in $\mathbb{R}^p$, the corresponding Gram matrix $\mathbf{K}_{m\times m}$ must be psd - for any vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^T$,

$$\boldsymbol{\alpha}^t \mathbf{K} \boldsymbol{\alpha} \geq 0.$$

When extending linear SVM to non-linear SVM, we use the so-called Kernel trick: if an algorithm only uses the inner product between $\mathbf{x}_i$'s, then we can operate this algorithm in a new feature space, which is constructed by embedding a point $\mathbf{x}$ to a new feature vector $\Phi(\mathbf{x})$, without explicitly constructing the mapping Φ; all we need to do is to replace any inner product $\mathbf{x}_i^t \mathbf{x}_j$ by $K(\mathbf{x}_i, \mathbf{x}_j)$. For example, we can have kernel PCA or kernel FDA.