Chapter 9. Support Vector Machine

9.3 The Solution of Support Vector Machines & 9.4 Extension of Support Vector Machines Using a Kernel

Gayoung Moon

2025-05-29

Descendants of Lagrange
School of Mathematics, Statistics and Data Science
Sungshin Women's University

Outline

KKT(Karush-Kuhn-Tucker) conditions

- KKT conditions are necessary or sufficient conditions that an optimal solution must satisfy in a constrained nonlinear optimization problem.

- The KKT conditions are defined for optimization problems with the following constraints:

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f_0(x) \\ \text{subject to } & f_i(x) \le 0,\ i = 1, \cdots, m \\ & h_j(x) = 0,\ j = 1, \cdots, p. \end{aligned}$$

- $f_0(x)$: objective function,
  $f_i(x) \le 0$: inequality constraints,
  $h_j(x) = 0$: equality constraints.
- And we express with the Lagrange function as:

$$L(x, \alpha_i, \lambda_j) := f_0(x) + \sum_{i=1}^{m} \alpha_i f_i(x) + \sum_{j=1}^{p} \lambda_j h_j(x).$$

The Four Conditions of the KKT Conditions

- There are four conditions that must be satisfied for the solution x^* to be the optimal solution:

1) Stationarity condition

$$\nabla f_0(x^*) + \sum_{i=1}^{m} \alpha_i \nabla f_i(x^*) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x^*) = 0,$$

2) Primal feasibility condition

$$f_i(x^*) \leq 0,\ h_j(x^*) = 0,$$

3) Dual feasibility condition

$$\alpha_i \geq 0 \text{ for all } i,$$

4) Complementary slackness condition

$$\alpha_i f_i(x^*) = 0 \text{ for all } i.$$

Outline

The Optimal Problem and the KKT Conditions in Support Vector Machine

- In support vector machine, the following optimization problem is solved:

$$\min_{\beta,\beta_0,\epsilon} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\epsilon_i$$

$$\text{subject to } \; y_i(\beta_0 + x_i\beta) \geq 1 - \epsilon_i,$$

$$\epsilon_i \geq 0 \quad \forall i.$$

- And the following seven equations are KKT conditions:

$$y_i(\beta_0^* + x_i\beta^*) - (1 - \epsilon_i^*) \geq 0$$
$$\epsilon_i^* \geq 0$$

$$\alpha_i[y_i(\beta_0^* + x_i\beta^*) - (1 - \epsilon_i^*)] = 0$$
$$\mu_i\epsilon_i^* = 0$$

$$\beta^* = \textstyle\sum_{i=1}^{N} \alpha_i y_i x_i \in \mathbb{R}^p$$
$$\textstyle\sum_{i=1}^{N} \alpha_i y_i = 0$$
$$C - \alpha_i - \mu_i = 0.$$

The KKT Conditions in SVM: Primal Feasibility Condition

- The first two conditions can be derived from the KKT condition (2):

$$f_1(\beta^*), \cdots, f_m(\beta^*) \leq 0.$$

- In the optimal problem of SVM, the original constraints are as follows:

$$y_i(\beta_0 + x_i\beta) \geq 1 - \epsilon_i \quad \forall i = 1, \cdots, N$$
$$\epsilon_i \geq 0 \quad \forall i = 1, \cdots, N.$$

- Thus, when the above two conditions, which are existing constraints, are satisfied, the optimal solution can be found.

The KKT Conditions in SVM: Complementary Slackness Condition

- The two conditions in the second paragraph are derived from the KKT condition (4):
$$\alpha_1 f_1(\beta^*) = \cdots = \alpha_m f_m(\beta^*) = 0.$$
- Based on the optimization problem of SVM, the Lagrangian function is set as follows
$$L_P := \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\epsilon_i - \sum_{i=1}^{N}\alpha_i\{y_i(\beta_0 + x_i\beta) - (1-\epsilon_i)\} - \sum_{i=1}^{N}\mu_i\epsilon_i,$$
and α_i and μ_i are the Lagrange multipliers.
- The relationship between the original constraints and the Lagrange multipliers $\alpha_i,\ \mu_i$ is as follows
$$\alpha_i[y_i(\beta_0 + x_i\beta) - (1-\epsilon_i)] = 0$$
$$\mu_i\epsilon_i = 0,$$

then it means that one of the Lagrange multipliers and the constraints must have the value 0.

The KKT Conditions in SVM: Stationarity Condition

- The last three conditions can be derived from the KKT condition (1):

$$\nabla f_0(\beta^*) + \sum_{i=1}^{m} \alpha_i \nabla f_i(\beta^*) = 0.$$

- Differentiating L_P w.r.t $\beta, \beta_0, \epsilon_i$:

$$\frac{\partial L_P}{\partial \beta} = \beta - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \Longrightarrow \beta = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$\frac{\partial L_P}{\partial \beta_0} = -\sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\frac{\partial L_P}{\partial \epsilon_i} = C - \alpha_i - \mu_i = 0.$$

- Then we obtain:

$$\beta = \textstyle\sum_{i=1}^{N} \alpha_i y_i x_i$$
$$\textstyle\sum_{i=1}^{N} \alpha_i y_i = 0$$
$$C - \alpha_i - \mu_i = 0.$$

SVM Primal Form: Applying KKT Conditions for Simplification

- We can construct the dual problem of L_P from the primal problem.
- If we reconstruct the equation of L_P using the equations differentiated by β_0 and ϵ_i, we can write it as follows:

$$
\begin{aligned}
L_P &:= \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\epsilon_i - \sum_{i=1}^{N}\alpha_i\{y_i(\beta_0 + x_i\beta) - (1-\epsilon_i)\} - \sum_{i=1}^{N}\mu_i\epsilon_i \\
&= \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\epsilon_i - \sum_{i=1}^{N}\alpha_i\{y_i(\beta_0 + x_i\beta) - (1-\epsilon_i)\} - \sum_{i=1}^{N}(C-\alpha_i)\epsilon_i \\
&= \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\epsilon_i - C\sum_{i=1}^{N}\epsilon_i - \sum_{i=1}^{N}\alpha_i\{y_i(\beta_0 + x_i\beta)\} + \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i\epsilon_i + \sum_{i=1}^{N}\alpha_i\epsilon_i \\
&= \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N}\alpha_i\{y_i(\beta_0 + x_i\beta)\} + \sum_{i=1}^{N}\alpha_i \\
&= \frac{1}{2}\|\beta\|^2 - \beta_0\sum_{i=1}^{N}\alpha_i y_i - \sum_{i=1}^{N}\alpha_i y_i x_i\beta + \sum_{i=1}^{N}\alpha_i = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N}\alpha_i y_i x_i\beta + \sum_{i=1}^{N}\alpha_i, \\
\therefore L_P &:= \sum_{i=1}^{N}\alpha_i + \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N}x_i\beta\alpha_i y_i.
\end{aligned}
$$

Formulating the Dual using the Lagrange Multipliers

- The equation of L_P, reconstructed by the equation differentiate by β, change again as follows:

$$
\begin{aligned}
L_P &:= \sum_{i=1}^{N} \alpha_i + \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} x_i \beta \alpha_i y_i \\
&= \sum_{i=1}^{N} \alpha_i + \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y_i x_i^\top\right)^\top \left(\sum_{j=1}^{N} \alpha_j y_j x_j^\top\right) - \sum_{i=1}^{N} x_i \left(\sum_{i=1}^{N} \alpha_i y_i x_i^\top\right) \alpha_i y_i \\
&= \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y_i x_i^\top\right)^\top \left(\sum_{j=1}^{N} \alpha_j y_j x_j^\top\right) \\
&= \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i x_j^\top .
\end{aligned}
$$

- So, we construct the function with input as the Lagrange coefficients $\alpha_i \geq 0,\ i = 1, \cdots, N$:

$$
L_D := \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i x_j^\top .
$$

- In this case, α ranges over $\sum_{i=1}^{N} \alpha_i y_i$ and $0 \leq \alpha_i \leq C$.

Understanding Constraints in the Dual Problem

- In the dual objective function L_D, μ_i disappears, but the constraint that α_i has a range of $0 \leq \alpha_i \leq C$ still exists through the condition $\mu_i = C - \alpha_i$.

- By solving the dual problem under this constraint, we can obtain the values of α_i.
  Then, the primal variable β can be computed using:

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$$

- And depending on the range of α_i, it can be classified into three cases, which is the key to SVM interpretation.

Solution of the Dual Problem According to α_i Range

- We can distinguish where each sample point is located based on the following three cases of α_i:

$$\begin{cases} \alpha_i = 0 & \Longrightarrow y_i(\beta_0 + x_i\beta) > 1 \\ 0 < \alpha_i < C & \Longrightarrow y_i(\beta_0 + x_i\beta) = 1 \\ \alpha_i = C & \Longrightarrow y_i(\beta_0 + x_i\beta) < 1. \end{cases}$$

Solution of the Dual Problem According to α_i Range (Contd.)

- When $\alpha_i = 0$, applying $C - \alpha_i - \mu_i = 0,\ \mu_i \epsilon_i = 0$, and $y_i(\beta_0 + x_i\beta) - (1 - \epsilon_i) \geq 0$ in this order, we have:

$$\alpha_i = 0 \Longrightarrow \mu_i = C > 0 \Longrightarrow \epsilon = 0 \Longrightarrow y_i(\beta_0 + x_i\beta) \geq 1.$$

- When $0 < \alpha_i < C$, from $\mu_i \epsilon_i = 0,\ C - \alpha_i - \mu_i = 0$, we have $\epsilon_i = 0$. Moreover, applying $\alpha_i[y_i(\beta_0 + x_i\beta) - (1 - \epsilon_i)] = 0$, we have:

$$0 < \alpha_i < C \Longrightarrow y_i(\beta_0 + x_i\beta) - (1 - \epsilon_i) = 0 \Longrightarrow y_i(\beta_0 + x_i\beta) = 1.$$

- When $\alpha_i = C$, from $\epsilon_i \geq 0$, we have $\epsilon_i \geq 0$. Moreover, applying $\alpha_i[y_i(\beta_0 + x_i\beta) - (1 - \epsilon_i)] = 0$, we have:

$$\alpha_i = C \Longrightarrow y_i(\beta_0 + x_i\beta) - (1 - \epsilon_i) = 0 \Longrightarrow y_i(\beta_0 + x_i\beta) \leq 1.$$

The Proof of the Optimal Solution in SVM

- As we have seen above, we show that at least one i satisfies $y_i(\beta_0 + x_i\beta) = 1$.

- If there exists an i such that $0 < \alpha_i < C$ and at least one $\epsilon_i = 0$, then from the above, the i satisfies $y_i(\beta_0 + x_i\beta) = 1$.
  That is, an optimal solution exists, which means that at least one support vector exists. In this case, we can obtain β_0 as $\beta_0 = y_1 - x_i\beta$.

The Proof of the Optimal Solution in SVM (Contd.)

- Suppose $\alpha_1 = \cdots = \alpha_N = 0$, then we have $\beta = 0$ from $\beta = \sum \alpha_i y_i x_i$.
  Moreover, we have $\mu_i = C$ and $\epsilon_i = 0$, which means $y_i(\beta_0 + x_i\beta) \geq 1$ from $\mu_i \epsilon_i = 0$ and $C - \alpha_i - \mu_i = 0$.
  $\rightarrow$ Therefore, $\beta_0 = \pm 1$ satisfy $y_i(\beta_0 + x_i\beta) = 1$ when $y_i = \pm 1$. It means, there are not just one optimal solution.

- Next, we suppose that $\alpha_i = C$ for at least one i and $\epsilon_i > 0$ for all i.
  If we define $\epsilon_* := \min_i \epsilon_i$, and replace $\epsilon_i \rightarrow \epsilon - \epsilon_*$, $\beta_0 \rightarrow \beta_0' + y_i\epsilon_*$, respectively, the latter still satisfies the constraint:

$$y_i\{(\beta_0' + y_i\epsilon_*) + x_i\beta\} - \{1 - (\epsilon - \epsilon_*)\} \geq 0,$$
$$\epsilon - \epsilon_* \geq 0.$$

  $\rightarrow$ Then, we can obtain a smaller value of $f_0(\beta, \beta_0, \epsilon) = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N} \epsilon_i$, which contradicts the underlying assumption that β, β_0, ϵ was optimal.

Solving the Dual Problem Using the Package in R

- We solve the dual problem of L_D using a quadratic programming solver.
- In the R language, a package called **quadprog** is available for this purpose.
- To solve the dual problem in **R**, we transform $L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i x_j^\top$ into the following matrix form:

$$\begin{aligned} &\text{minimize} \quad -\frac{1}{2}\alpha^\top D_{\text{mat}}\alpha + d_{\text{vec}}^\top \alpha, \\ &\text{subject to} \quad A_{\text{mat}}\alpha \geq b_{\text{vec}},\ \alpha \in \mathbb{R}^N. \end{aligned}$$

- $D_{\text{mat}} = (x_i y_i)(x_j y_j)^\top \in \mathbb{R}^{N\times N}$,
  $d_{\text{vec}} \in \mathbb{R}^N$: linear term of objective function,
  $A_{\text{mat}} \in \mathbb{R}^{m\times N}$: constraint matrix,
  $b_{\text{vec}} \in \mathbb{R}^m (m \geq 1)$: constrain right side.

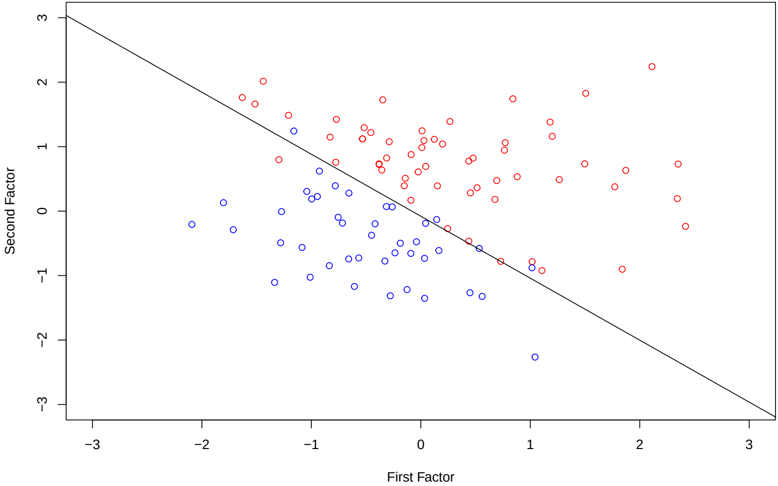
- In particular, in the formulation derived above, we take $m = 2N + 1,\ meq = 1,$

$$z = \begin{bmatrix} x_{1,1}y_1 & \cdots & x_{1,p}y_1 \\ \vdots & \vdots & \vdots \\ x_{N,1}y_N & \cdots & x_{N,p}y_N \end{bmatrix} \in \mathbb{R}^{N\times p},\ A_{\text{mat}} = \begin{bmatrix} y_1 & \cdots & y_N \\ -1 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & -1 \\ 1 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{(2N+1)\times N},$$

$D_{\text{mat}} = zz^\top \in \mathbb{R}^{N\times N}$,
$b_{\text{vec}} = [0, -C, \cdots, -C, 0, \cdots, 0]^\top \in \mathbb{R}^{2N+1},\ d_{\text{vec}} = [1, \cdots, 1]^\top \in \mathbb{R}^N$, and
$\alpha = [\alpha_1, \cdots, \alpha_N] \in \mathbb{R}^N$.

[Example 1] Drawing the border of the SVM

- We generate samples and draw the border of the support vector machine.



Outline

Using the Kernel Function in SVM

- The reason for solving the dual rather than the primal is that the dual objective L_D can be expressed using inner products $\langle \cdot, \cdot \rangle$ as

$$L_D := \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle.$$

- Let V be the vector space with $\phi : \mathbb{R}^p \to V$, then we may replace $\langle x_i, x_j \rangle$ by $k(x_i, x_j) := \langle \phi(x_i), \phi(x_j) \rangle$.

- In such a case, we construct a nonlinear classification rule from $(\phi(x_1), y_1), \cdots, (\phi(x_N), y_N)$.
  $\to$ In other words, even if the mapping $\phi(x) \to y$ is linear and the learning is performed via a linear model, the original mapping $x \to y$ can still be nonlinear.

- In the following, we construct a matrix K such that the (i,j)-th element is $K_{i,j} = \langle \phi(x_i), \phi(x_j) \rangle \in V$.

- In fact, for $z \in \mathbb{R}^N$, the matrix

$$\begin{aligned} z^\top K z &= \sum_{i=1}^{N} \sum_{j=1}^{N} z_i \langle \phi(x_i), \phi(x_j) \rangle z_j \\ &= \langle \sum_{i=1}^{N} z_i \phi(x_i), \sum_{j=1}^{N} z_j \phi(x_j) \rangle \\ &= \| \sum_{i=1}^{N} z_i \phi(x_i) \|^2 \geq 0 \end{aligned}$$

is symmetric and non-negative definite, and $k(\cdot,\cdot)$ is a kernel in the strict sense.

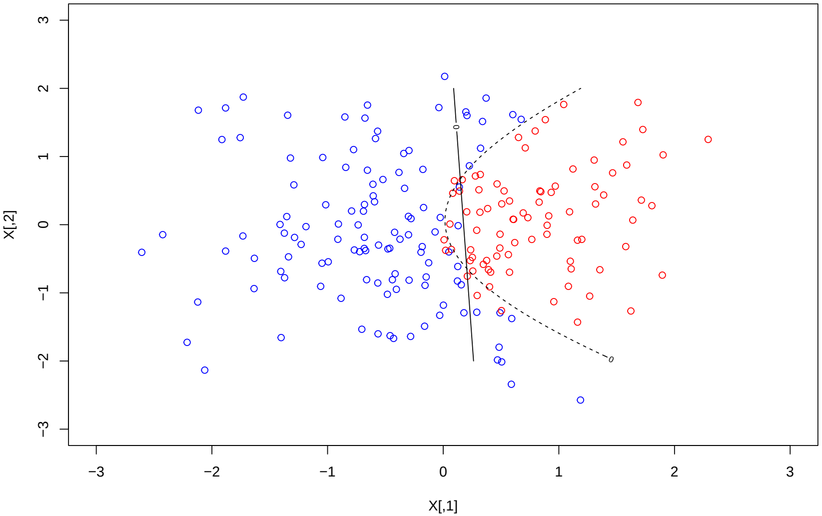
Definition New Function svm.2 using svm.1

- We modify the function **svm.1** as follows to define new function **svm.2**:

1. add argument **K** to the function definition,
2. replace sum(X[, i]*X[, j]) with **K(X[i,], X[j,])**,
3. replace **beta** in **return()** with **alpha**.

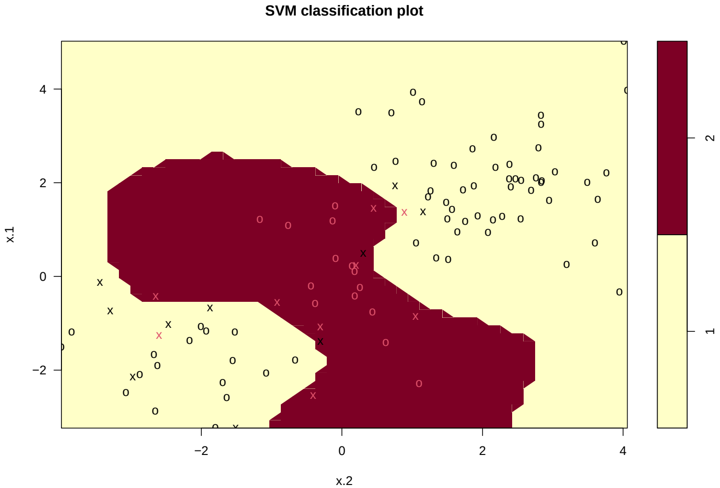
[Example 2] Comparison of Linear and Nonlinear Borders

- We generate samples and draw linear and nonlinear borders that are flat and curved surfaces.



[Example 3] Using the e1071 Package in R

- Using the **e1071** package with the radial kernel, we draw a nonlinear surface.
    - Radial kernel: $K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$
- We can use **svm** function to specify parameter values such as γ and kernel to be used for analysis.



Q & A

Thank you :)