Notes on Statistical and Machine Learning

Support Vector Machines

Chapter: *16* **Prepared by:** *Chenxi Zhou*

This note is prepared based on

- *Chapter 12, Support Vector Machines and Flexible Discriminants* in Hastie, Tibshirani, and Friedman (2009),
- *Chapter 11, Support Vector Machines* in Izenman (2009), and
- *Chapter 9, Regression Estimation* in Schölkopf and Smola (2002).

I. Review of Support Vector Machines in Linearly Separable Case

1. **Setup:** The training data consists of n pairs $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)$, where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \{-1, +1\}$ for all $i = 1, 2, \cdots, n$. Define a hyperplane by

$$\Big\{\mathbf{x} \,\Big|\, f(\mathbf{x}) := \mathbf{x}^\top \boldsymbol{\beta} + \beta_0 = 0\Big\},$$

where $\boldsymbol{\beta}$ is a unit vector, i.e., $\|\boldsymbol{\beta}\|_2 = 1$. A classification rule induced by f is

$$G(\mathbf{x}) = \mathrm{sign}(\mathbf{x}^\top \boldsymbol{\beta} + \beta_0),$$

where $G : \mathbb{R}^p \to \{-1, 1\}$; in other words, G outputs the class of the point $\mathbf{x}$.

2. **Linearly Separable Case:** If the two classes are linearly separable, we can find a function $f(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\beta} + \beta_0$ with $y_i f_i(\mathbf{x}_i) > 0$ for all i. In this case, we can find the hyperplane that creates the *biggest* margin between the training points for Class $+1$ and -1. The associated optimization problem is of the following form

$$\begin{aligned} \underset{\boldsymbol{\beta}, \beta_0}{\text{maximize}} \quad & M \\ \text{subject to} \quad & y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \ge M \text{ for all } i = 1, \cdots, n, \\ & \|\boldsymbol{\beta}\|_2 = 1 \end{aligned} \tag{1}$$

We can get rid of the constraint $\|\boldsymbol{\beta}\|_2 = 1$ by modifying the constraint as

$$\frac{1}{\|\boldsymbol{\beta}\|_2} y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \widetilde{\beta}_0) \ge M, \tag{2}$$

where $\widetilde{\beta}_0 = \|\boldsymbol{\beta}\|_2 \beta_0$. Note that (2) is equivalent to

$$y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \widetilde{\beta}_0) \ge \|\boldsymbol{\beta}\|_2 M. \tag{3}$$

Since any positively scaled multiple of $\boldsymbol{\beta}$ and β_0 satisfies the constraint as well, we can arbitrarily set $M = 1/\|\boldsymbol{\beta}\|_2$ and reformulate the original optimization problem as

$$\begin{aligned} \min_{\boldsymbol{\beta}, \beta_0} \quad & \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 \\ \text{subject to} \quad & y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq 1 \text{ for all } i = 1, \cdots, n. \end{aligned} \tag{4}$$

This resulting optimization problem is a convex problem with a quadratic objective function and linear inequality constraints.

II. Support Vector Machines in Linearly Non-separable Case

1. **Introducing the Slack Variables:** Suppose the classes can *overlap* in feature space. We still maximize M and allow some points to reside on the *wrong* side via defining the slack variables
$$\boldsymbol{\xi} := (\xi_1, \cdots, \xi_n)^\top.$$
We can include the constraints in (1) of the following form:
$$\xi_i \geq 0 \text{ for all } i = 1, 2, \cdots, n, \qquad \text{and} \qquad \sum_{i=1}^n \xi_i \leq \text{some constant.}$$
There are two natural ways of specifying these constraints:
   (a) $y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq M - \xi_i$: this is a natural choice as it measures the overlap in *actual* distance from the margin; but this leads to a non-convex optimization problem;
   (b) $y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq M(1 - \xi_i)$: this measures the relative distance, which changes the width of the margin M; the resulting optimization problem is convex.

   Of the two choices, we use the second one.

   Notice that the value ξ_i in $y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq M(1 - \xi_i)$ is the proportional amount by which the prediction $f(\mathbf{x}_i) = \mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0$ is on the wrong side of its margin. By bounding $\sum_{i=1}^n \xi_i$, we bound the total proportional amount by which predictions fall on the wrong side of their margin.

   In this setup, the misclassification of the i-th observation occurs when $\xi_i > 1$, so bounding $\sum_{i=1}^n \xi_i$ at a value K bounds the total number of training misclassifications at K.

2. **Problem Formulation:** We drop the constraint $\|\boldsymbol{\beta}\|_2 = 1$ by defining $M = 1/\|\boldsymbol{\beta}\|_2$

and obtain the following formulation of the problem in linearly inseparable case as

$$
\begin{aligned}
\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \quad & \|\boldsymbol{\beta}\|_2 \\
\text{subject to} \quad & y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i, \qquad \text{for all } i = 1, \cdots, n, \\
& \xi_i \geq 0, \qquad \text{for all } i = 1, \cdots, n \\
& \sum_{i=1}^n \xi_i \leq K,
\end{aligned} \tag{5}
$$

where $K > 0$ is some constant to be specified.

3. **Computing the Support Vector Machine:** Notice that (5) is a *convex* optimization problem with a quadratic objective function and linear constraints. We can write it in the following equivalent form

$$
\begin{aligned}
\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^n \xi_i \\
\text{subject to} \quad & y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i, \qquad \text{for all } i = 1, \cdots, n, \\
& \xi_i \geq 0, \qquad \text{for all } i = 1, \cdots, n,
\end{aligned} \tag{6}
$$

where C is the "cost" parameter. In this formulation, the linearly separable case corresponds to $C = \infty$.

The primal Lagrangian function of minimizing with respect to $\boldsymbol{\beta}$, β_0 and ξ is

$$
L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\xi}) := \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i[y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^n \mu_i \xi_i,
$$

which we minimize with respect to β_0, $\boldsymbol{\beta}$ and $\boldsymbol{\xi}$.

Differentiating L_P with respect to β_0, $\boldsymbol{\beta}$ and $\boldsymbol{\xi}$ and setting the derivatives to 0 yield

$$
\begin{aligned}
\frac{\partial L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\xi})}{\partial \beta_0} &\overset{\text{set}}{=} -\sum_{i=1}^n \alpha_i y_i = 0 \qquad &&\Longleftrightarrow \qquad 0 = \sum_{i=1}^n \alpha_i y_i, \\
\frac{\partial L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\xi})}{\partial \boldsymbol{\beta}} &\overset{\text{set}}{=} \boldsymbol{\beta} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = 0 \qquad &&\Longleftrightarrow \qquad \boldsymbol{\beta} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i, \\
\frac{\partial L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\xi})}{\partial \xi_i} &\overset{\text{set}}{=} C - \alpha_i - \mu_i = 0 \qquad &&\Longleftrightarrow \qquad \alpha_i = C - \mu_i \text{ for all } i = 1, \cdots, n,
\end{aligned}
$$

with constraints $\alpha_i \geq 0$, $\mu_i \geq 0$ and $\xi_i \geq 0$ for all $i = 1, \cdots, n$.

Substituting the constraints back to the Lagrangian primal function yields the *dual function*

$$
L_D(\alpha_1, \cdots, \alpha_n) := \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n \sum_{i'=1}^n \alpha_i \alpha_{i'} y_i y_{i'} \mathbf{x}_i^\top \mathbf{x}_{i'},
$$

which gives a *lower bound* on the objective function in (6) for any feasible point. We maximize the dual function with the constraints $0 \leq \alpha_i \leq C$ for all $i = 1, \cdots, n$ and $\sum_{i=1}^n \alpha_i y_i = 0$.

4. **Karush-Kuhn-Tucker Conditions:** The complete set of Karush-Kuhn-Tucker (KKT) conditions is

   (a) Stationarity:

   $$\boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \qquad 0 = \sum_{i=1}^{n} \alpha_i y_i, \qquad \alpha_i = C - \mu_i \text{ for all } i = 1, \cdots, n;$$

   (b) Complementary slackness: for all $i = 1, \cdots, n$,

   $$\alpha_i \big[y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) - (1 - \xi_i) \big] = 0, \qquad \text{and} \qquad \mu_i \xi_i = 0;$$

   (c) Primal feasibility: for all $i = 1, \cdots, n$,

   $$y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i, \qquad \text{and} \qquad \xi_i \geq 0;$$

   (d) Dual feasibility: for all $i = 1, \cdots, n$,

   $$\alpha_i \geq 0, \qquad \text{and} \qquad \mu_i \geq 0.$$

5. **Support Vectors:** By condition (a) in KKT conditions, we see that

   $$\widehat{\boldsymbol{\beta}} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i,$$

   where $\widehat{\boldsymbol{\beta}}$ is the solution to $\boldsymbol{\beta}$ in (6), and $(\hat{\alpha}_1, \hat{\alpha}_2, \cdots, \hat{\alpha}_n)^\top$ is the solution to $(\alpha_1, \alpha_2, \cdots, \alpha_n)$ in (19).

   From (b), $\hat{\alpha}_i \neq 0$ only when $y_i(\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}} + \hat{\beta}_0) - (1 - \xi_i) = 0$. These observations $\mathbf{x}_i$'s are called the *support vectors*. Among these support vectors,

   - some of them lie *on* the edge of the margin, i.e., $\hat{\xi}_i = 0$, and then $0 < \hat{\alpha}_i < C$;
   - the remainders have $\hat{\xi}_i > 0$ and $\hat{\alpha}_i = C$.

6. **Computing β_0:** To solve for β_0, we choose the margin points with $\hat{\alpha}_i \geq 0$ and $\hat{\xi}_i = 0$ and utilize the condition (b) above. It is better to take the average over all observations to numerical stability instead of using just a single observation.

7. **Decision Boundary:** Given $\widehat{\boldsymbol{\beta}}$ and $\hat{\beta}_0$, the decision function can be written as

   $$\widehat{G}(\mathbf{x}) = \text{sign}(\hat{f}(\mathbf{x})) = \text{sign}(\widehat{\boldsymbol{\beta}}^\top \mathbf{x} + \hat{\beta}_0).$$

8. **Parameter Tuning:** In (6), we have the cost parameter $C > 0$ as the tuning parameter, the optimal choice of which can be determined by K-fold cross-validation.

   Remark. Notice that leaving one observation that is *not* a support vector out will *not* change the solution.

9. **Three States of Labeled Points $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$:** By the KKT conditions, all labeled points $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ fall into exactly one of three distinct groups:

   (a) Observations *correctly* classified and outside their margins with $y_i f(\mathbf{x}_i) > 1$ and Lagrange multipliers $\alpha_i = 0$;

   (b) Observations *sitting on* their margins with $y_i f(\mathbf{x}_i) = 1$ and Lagrange multipliers $\alpha_i \in [0, C]$;

   (c) Observations *inside* their margins have $y_i f(\mathbf{x}_i) < 1$ with $\alpha_i = C$.

III. Support Vector Machines and Kernels

1. **Introduction:** Note the support vector classifiers lead to linear boundaries in the input feature space. By enlarging the feature space using *basis expansions*, we can obtain non-linear boundaries in the original feature space.

   The *support vector machine classifier* is an extension of the idea above, and the dimension of the enlarged space is allowed to become *very large*, or even infinite. However, with sufficiently many basis functions, the data would be *linearly separable* and overfitting may occur.

2. **Main Idea:** Suppose that we choose basis functions h_m, for $m = 1, \cdots, M$, and fit the support vector classifier using input features

$$\mathbf{h}(\mathbf{x}_i) = \big(h_1(\mathbf{x}_i), \cdots, h_M(\mathbf{x}_i)\big)^\top \in \mathbb{R}^M,$$

   for all $i = 1, \cdots, n$. Then, we produce the (nonlinear) function $f(\mathbf{x}) = h(\mathbf{x})^\top \boldsymbol{\beta} + \beta_0$, and the classifier is $\widehat{G}(\mathbf{x}) = \text{sign}(\hat{f}(\mathbf{x}))$ as before.

3. **Computing the SVM for Classification:** We work with the transformed feature vectors $\mathbf{h}$ directly, and the resulting Lagrangian dual function is

$$L_D(\alpha_1, \cdots, \alpha_n) := \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{i'=1}^n \alpha_i \alpha_{i'} y_i y_{i'} \langle \mathbf{h}(\mathbf{x}_i), \mathbf{h}(\mathbf{x}_{i'}) \rangle. \tag{7}$$

   Then, the solution function f is of the form

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x})^\top \boldsymbol{\beta} + \beta_0 = \sum_{i=1}^n \alpha_i y_i \langle \mathbf{h}(\mathbf{x}), \mathbf{h}(\mathbf{x}_i) \rangle + \beta_0. \tag{8}$$

   As before, given α_i, β_0 can be determined by solving $y_i \cdot f(\mathbf{x}_i) = 1$ for any or all $\mathbf{x}_i$ for which $\alpha_i \in (0, C)$.

4. **Introducing Kernels:** Notice that the dual function (7) and the solution function (8) both involve $\mathbf{h}$ *only* through the inner products. As a consequence, we do *not* need to specify the transformation $\mathbf{h}$ and *only* require the kernel function defined by

$$K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{h}(\mathbf{x}), \mathbf{h}(\mathbf{x}') \rangle,$$

that is, the inner product in the *transformed* space. Here, the kernel function $K(\cdot, \cdot)$ is a symmetric, positive (semi-)definite function.

Some popular choices of the kernel functions in support vector machines include

(a) *d-th Degree Polynomial:* $K(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^d$;

(b) *Radial Basis:* $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|_2^2)$;

(c) *Neural Network:* $K(\mathbf{x}, \mathbf{x}') = \tanh(\kappa_1 \langle \mathbf{x}, \mathbf{x}' \rangle + \kappa_2)$.

5. **Example of Kernel:** We show an example of polynomial kernel function with degree of 2 and two inputs, x_1 and x_2, that is, letting $\mathbf{x} = (x_1, x_2)$ and $\mathbf{x}' = (x_1', x_2')$,

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^2 \\ &= (1 + x_1 x_1' + x_2 x_2')^2 \\ &= 1 + 2x_1 x_1' + 2x_2 x_2' + (x_1 x_1')^2 + (x_2 x_2')^2 + 2x_1 x_1' x_2 x_2' \\ &= \big\langle (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2), (1, \sqrt{2}x_1', \sqrt{2}x_2', {x_1'}^2, {x_2'}^2, \sqrt{2}x_1' x_2') \big\rangle, \end{aligned}$$

where $\mathbf{h}(\mathbf{x}) = \mathbf{h}(x_1, x_2) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)^\top \in \mathbb{R}^6$.

6. **Effects of Cost Parameter C:**

(a) A *large* value of $C > 0$ will discourage any positive ξ_i, leading to an overfit *wiggly* boundary in the original feature space; and

(b) A *small* value of C will encourage a small value of $\|\boldsymbol{\beta}\|_2$, leading to a *smoother* boundary.

7. **Support Vector Machines as a Penalized Method:** With $f(\mathbf{x}) = h(\mathbf{x})^\top \boldsymbol{\beta} + \beta_0$, consider the following optimization problem

$$\underset{\boldsymbol{\beta}, \beta_0}{\text{minimize}} \ \sum_{i=1}^n [1 - y_i f(\mathbf{x}_i)]_+ + \frac{\lambda}{2} \|\boldsymbol{\beta}\|_2^2, \tag{9}$$

where $[x]_+ = \max\{x, 0\}$. This has the form of a loss function plus a penalty term. We

Loss Function	$L(y, f(\mathbf{x}))$	Minimizing Function
Binomial Deviance	$\log(1+\exp(-yf(\mathbf{x})))$	$f(\mathbf{x}) = \log\big(\frac{\mathbb{P}(Y=+1 \mid \mathbf{x})}{\mathbb{P}(Y=-1 \mid \mathbf{x})}\big)$
SVM Hinge Loss	$[1-yf(\mathbf{x})]_+$	$f(\mathbf{x}) = \mathrm{sign}[\mathbb{P}(Y=+1 \mid \mathbf{x}) - \frac{1}{2}]$
Squared Error	$(y-f(x))^2 = (1-yf(x))^2$	$f(x) = 2\mathbb{P}(Y=+1 \mid \mathbf{x}) - 1$
"Huberized" Square Hinge Loss	$-4yf(\mathbf{x})$, if $yf(\mathbf{x}) < -1$ $[1-yf(\mathbf{x})]_+^2$, otherwise	$f(\mathbf{x}) = 2\mathbb{P}(Y=+1 \mid \mathbf{x}) - 1$

Table 1: Different loss functions used for the binary classification problem.

show that the optimization problem (9), with $\lambda = 1/C$, is the same as (6) as below

$$\begin{aligned}
&\operatorname*{arg\,min}_{\boldsymbol{\beta},\beta_0}\left\{\frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^n \xi_i, \text{ s.t } \xi_i \geq 0, y_i(\mathbf{x}_i^\top\boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i, i = 1, \cdots, n\right\} \\
=&\operatorname*{arg\,min}_{\boldsymbol{\beta},\beta_0}\left\{\frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^n \xi_i, \text{ s.t } \xi_i \geq 0, \xi_i \geq 1 - y_i(\mathbf{x}_i^\top\boldsymbol{\beta} + \beta_0), i = 1, \cdots, n\right\} \\
=&\operatorname*{arg\,min}_{\boldsymbol{\beta},\beta_0}\left\{\frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^n \xi_i, \text{ s.t } \xi_i = \max\big\{0, 1 - y_i(\mathbf{x}_i^\top\boldsymbol{\beta} + \beta_0)\big\}, i = 1, \cdots, n\right\} \\
=&\operatorname*{arg\,min}_{\boldsymbol{\beta},\beta_0}\left\{\frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^n \max\big\{0, 1 - y_i(\mathbf{x}_i^\top\boldsymbol{\beta} + \beta_0)\big\}\right\} \\
=&\operatorname*{arg\,min}_{\boldsymbol{\beta},\beta_0}\left\{\frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^n [1 - y_i(\mathbf{x}_i^\top\boldsymbol{\beta} + \beta_0)]_+\right\} \\
=&\operatorname*{arg\,min}_{\boldsymbol{\beta},\beta_0}\left\{\frac{\lambda}{2}\|\boldsymbol{\beta}\|^2 + \sum_{i=1}^n [1 - y_i(\mathbf{x}_i^\top\boldsymbol{\beta} + \beta_0)]_+\right\}.
\end{aligned}$$

Here,

$$L(y, f) = [1 - yf]_+$$

is called the *hinge loss function* and is reasonable for two-class classification problem. The formulation (9) exhibits the SVM as a regularized function estimation problem, where the coefficients of the linear expansion $f(\mathbf{x}) = \beta_0 + \mathbf{h}(\mathbf{x})^\top\boldsymbol{\beta}$ are shrunk to zero.

8. **A Comparison of Different Loss Functions Used in Classification:** A table of comparing different loss functions and the corresponding minimizing functions are given below:

   (a) The *negative log-likelihood loss* has similar tails as the hinge loss, giving zero penalty to points well inside the margin and a linear penalty to points on the wrong side and far away;

(b) Squared-error loss give quadratic penalty to points both well inside their margin and outside;

(c) The squared hinge loss $[1 - yf(\mathbf{x})]_+^2$ is like the squared-error loss, except it is zero for points inside the margin;

(d) The "Huberized" squared hinge loss is a squared version of hinge loss but converts smoothly to a linear loss at $yf = -1$.

9. **Margin Maximizing Loss-function:** All the loss functions listed in Table 1 except the squared-error loss are so-called *margin maximizing loss-function*, meaning that if the data are separable, then the limit of $\widehat{\boldsymbol{\beta}}_\lambda$ in (9) as $\lambda \to 0$ defines the optimal separating hyperplane.

10. **Function Estimation and Reproducing Kernels:** Suppose that the basis $\mathbf{h}$ arises from the eigen-expansion of a positive definite kernel K,

$$K(\mathbf{x}, \mathbf{x}') = \sum_{m=1}^{\infty} \delta_m \phi_m(\mathbf{x}) \phi_m(\mathbf{x}'),$$

and

$$h_m(\mathbf{x}) = \sqrt{\delta_m} \phi_m(\mathbf{x}).$$

Then, letting $\theta_m = \sqrt{\delta_m} \beta_m$, we can re-write (9) as

$$\underset{\beta_0, \boldsymbol{\theta}}{\text{minimize}} \sum_{i=1}^{n} \left[1 - y_i \left(\beta_0 + \sum_{m=1}^{\infty} \theta_m \phi_m(\mathbf{x}_i) \right) \right]_+ + \frac{\lambda}{2} \sum_{m=1}^{\infty} \frac{\theta_m^2}{\delta_m},$$

where $\boldsymbol{\theta} := (\theta_1, \theta_2, \cdots)^\top$.

By the theory of reproducing kernel Hilbert spaces, the solution is of *finite dimensions* and takes on the following form

$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i).$$

In this sense, we can rewrite the optimization problem (9) as

$$\underset{\beta_0, \boldsymbol{\alpha}}{\text{minimize}} \sum_{i=1}^{n} [1 - y_i f(\mathbf{x}_i)]_+ + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha},$$

where $\boldsymbol{\alpha} := (\alpha_1, \cdots, \alpha_n)^\top \in \mathbb{R}^n$ and $\mathbf{K}$ is the $n \times n$ matrix of kernel evaluations for all pairs of training features.

11. **Relationship Between Loss Function and Function Space:** Consider the optimization problem

$$\underset{f \in \mathcal{H}}{\text{minimize}} \sum_{i=1}^{n} [1 - y_i f(\mathbf{x}_i)]_+ + \lambda J(f),$$

where $\mathcal{H}$ is the structured space of functions and J is an appropriate regularizer on $\mathcal{H}$. With a specified $\mathcal{H}$ and an appropriate J, we can characterize the solution.

Example: Let $\mathcal{H}$ be the space of additive functions $f(\mathbf{x}) = \sum_{j=1}^{p} f_j(x_j)$ and $J(f) = \sum_{j=1}^{p} \int \{f_j''(x_j)\}^2 \mathrm{d}x_j$. The solution is an additive cubic spline with the kernel $K(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{p} K_j(x_j, x_j')$, where each K_j is the kernel appropriate for the univariate smoothing spline in x_j, for all $j = 1, 2, \cdots, p$.

Conversely, any kernel functions can be used with any convex loss function and will lead to a finite-dimensional representation of solution.

Example: Suppose we use the binomial log-likelihood as the loss function, and the fitted function is of the form

$$\hat{f}(\mathbf{x}) = \log\left(\frac{\widehat{\mathbb{P}}(Y = +1 \,|\, \mathbf{x})}{\widehat{\mathbb{P}}(Y = -1 \,|\, \mathbf{x})}\right) = \hat{\beta}_0 + \sum_{i=1}^{n} \hat{\alpha}_i K(\mathbf{x}, \mathbf{x}_i),$$

and therefore,

$$\widehat{\mathbb{P}}(Y = +1 \,|\, \mathbf{x}) = \frac{1}{1 + \exp\left(-\hat{\beta}_0 - \sum_{i=1}^{n} \hat{\alpha}_i K(\mathbf{x}, \mathbf{x}_i)\right)}.$$

12. A Path Algorithm for the SVM Classifier: Consider the problem (9) and the solution for $\boldsymbol{\beta}$ at a given value of λ is

$$\boldsymbol{\beta}_\lambda = \frac{1}{\lambda} \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i.$$

The following is the path algorithm.

(a) Initially, set λ large, and the margin $\frac{1}{\|\boldsymbol{\beta}_\lambda\|_2}$ is wide. All points are inside the margin with $\alpha_i = C$;

(b) Decrease λ, and correspondingly, $\frac{1}{\|\boldsymbol{\beta}_\lambda\|_2}$ also decreases, and the margin becomes narrower. Consequently, some points move from inside of the margin to outside of the margin, and their $\alpha_i(\lambda)$ values will change from C to 0.

As λ decreases, all that changes are $\alpha_i \in [0, C]$ of those of points on the margin. Since all these points have $y_i f(\mathbf{x}_i) = 1$, this results in a small set of linear equations that prescribe how $\alpha_i(\lambda)$ and, hence, $\boldsymbol{\beta}_\lambda$ changes during these transitions.

IV. Multi-class Support Vector Machines

1. Setup: We consider the multi-class support vector machines, and let $y_i \in \{1, 2, \cdots, W\}$, for all $i = 1, \cdots, n$.

2. Multi-class SVM as a Series of Binary Problems:

(a) *One-versus-rest:* Divide the W-class problem into W binary classification sub-problems of the type "w-th class" vs. "not w-th class", for all $w = 1, 2, \cdots, W$.
A new observation $\mathbf{x}_0$ is then assigned to the class with the largest value of $\hat{f}_w(\mathbf{x}_0)$, for all $w = 1, 2, \cdots, W$, where $\hat{f}_w$ is the optimal SVM solution for the binary problem of the w-th class versus the rest;

(b) *One-versus-one:* Divide the W-class problem into $\binom{W}{2}$ comparisons of all 2 pairs of classes. A classifier $\hat{f}_w$ is constructed by coding the w-th class as positive and the u-th class as negative, for all $w, u = 1, 2, \cdots, W$, $w \neq u$. Then, for a new $\mathbf{x}_0$, aggregate the votes for each class and assign $\mathbf{x}_0$ to the class having the most votes.

3. **A True Multi-class SVM:** This part is based on Lee, Lin, and Wahba (2004) and *Section 11.4, Multiclass Support Vector Machines* in Izenman (2009).

(a) *Main Idea:* We need to consider all W classes *simultaneously*, and the classifier has to reduce to the binary SVM classifier if $W = 2$.

(b) *Relabeling:* Let $\mathbf{v}_1, \cdots, \mathbf{v}_W$ be a sequence of W-vectors, where $\mathbf{v}_w$ has the entry 1 in the w-th position and whose elements sum to zero, $w = 1, 2, \cdots, W$, i.e.,

$$\begin{aligned}
\mathbf{v}_1 &= \left(1, -\frac{1}{W-1}, -\frac{1}{W-1}, \cdots, -\frac{1}{W-1}\right), \\
\mathbf{v}_2 &= \left(-\frac{1}{W-1}, 1, -\frac{1}{W-1}, \cdots, -\frac{1}{W-1}\right), \\
&\vdots \\
\mathbf{v}_W &= \left(-\frac{1}{W-1}, -\frac{1}{W-1}, -\frac{1}{W-1}, \cdots, 1\right).
\end{aligned}$$

We let $\mathbf{x}_i$ has the label $\mathbf{y}_i = \mathbf{v}_w$ if $\mathbf{x}_i$ belongs to Class w, for all $i = 1, \cdots, n$ and all $w = 1, \cdots, W$.

(c) *Separating Hyperplane:* We generalize the separating function f to a W-vector of separating functions, i.e.,

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_W(\mathbf{x}))^\top \in \mathbb{R}^W,$$

where

$$f_w(\mathbf{x}) = \beta_{w,0} + h_w(\mathbf{x}), \qquad \text{for all } w = 1, 2, \cdots, W,$$

and h_w belongs to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$.
In order to ensure the uniqueness of the solution, we require

$$\sum_{w=1}^{W} f_w(\mathbf{x}) = 0. \tag{10}$$

(d) *Optimization Problem Formulation:* We find the function

$$\mathbf{f}(\mathbf{x}) := (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_W(\mathbf{x}))^\top \in \mathbb{R}^W$$

that minimizes

$$L_\lambda(\mathbf{f}) := \frac{1}{n}\sum_{i=1}^{n}[\mathbf{L}(\mathbf{y}_i)]^\top(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ + \frac{\lambda}{2}\sum_{w=1}^{W}\|h_w\|_{\mathcal{H}}^2, \tag{11}$$

where $\|\cdot\|_{\mathcal{H}}$ denotes the RKHS norm,

$$[f(\mathbf{x}_i) - \mathbf{y}_i]_+ = \big([f_1(\mathbf{x}_i) - y_{i,1}]_+, \cdots, [f_G(\mathbf{x}_i) - y_{i,G}]_+\big)^\top,$$

and $\mathbf{L}(\mathbf{y}_i)$ is a W-vector with 0 in the w-th component if $\mathbf{x}_i$ belongs to the w-th class, and 1 in all other components. In particular, the vector $\mathbf{L}(\mathbf{y}_i)$ represents no cost if it is correctly classified and has a cost of 1 if it is misclassified.

Remark. We can also use an unequal misclassification cost structure if appropriate.

(e) *Example:* Let $W = 2$. Then, we have $\mathbf{v}_1 = (1, -1)^\top$ and $\mathbf{v}_2 = (-1, 1)^\top$. If $\mathbf{x}_i$ belongs to Class 1, then

$$\begin{aligned}[\mathbf{L}(\mathbf{y}_i)]^\top(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ &= \begin{pmatrix}0\\1\end{pmatrix}^\top\begin{pmatrix}[f_1(\mathbf{x}_i) - 1]_+\\ [f_2(\mathbf{x}_i) - (-1)]_+\end{pmatrix}\\ &= [f_2(\mathbf{x}_i) - (-1)]_+\\ &= [1 - f_1(\mathbf{x}_i)]_+.\end{aligned}$$

Similarly, if $\mathbf{x}_i$ belongs to Class 2, then

$$\begin{aligned}[\mathbf{L}(\mathbf{y}_i)]^\top(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ &= \begin{pmatrix}1\\0\end{pmatrix}^\top\begin{pmatrix}[f_1(\mathbf{x}_i) - (-1)]_+\\ [f_2(\mathbf{x}_i) - 1]_+\end{pmatrix}\\ &= [f_1(\mathbf{x}_i) - (-1)]_+\\ &= [f_1(\mathbf{x}_i) + 1]_+.\end{aligned}$$

(f) *Characterization of the Solution to (11):* We now characterize the solution to (11). Since $h_w \in \mathcal{H}$, we can write it as

$$h_w = \sum_{\ell=1}^{n}\beta_{w,\ell}K(\mathbf{x}_\ell, \cdot) + h_w^\perp,$$

where $\{\beta_{w,\ell}\}$ are constants and $h_w^\perp$ is an element in $\mathcal{H}$ that is orthogonal to the linear span of $\{K(\mathbf{x}_1, \cdot), K(\mathbf{x}_2, \cdot), \cdots, K(\mathbf{x}_n, \cdot)\}$.

Due to (10), we must have

$$f_W = -\sum_{w=1}^{W-1}\beta_{w,0} - \sum_{w=1}^{W-1}\sum_{\ell=1}^{n}\beta_{w,\ell}K(\mathbf{x}_i, \cdot) + \sum_{w=1}^{W-1}h_w^\perp.$$

In addition, by the reproducing property of K, we have

$$\langle h_w, K(\mathbf{x}_i, \cdot)\rangle_{\mathcal{H}} = h_w(\mathbf{x}_i), \qquad \text{for all } i = 1, 2, \cdots, n,$$

and, hence,

$$\begin{aligned} f_w(\mathbf{x}_i) &= \beta_{w,0} + h_w(\mathbf{x}_i) \\ &= \beta_{w,0} + \langle h_w, K(\mathbf{x}_i, \cdot)\rangle_{\mathcal{H}} \\ &= \beta_{w,0} + \left\langle \sum_{\ell=1}^{n} \beta_{w,\ell} K(\mathbf{x}_\ell, \cdot) + h_w^\perp, K(\mathbf{x}_i, \cdot) \right\rangle_{\mathcal{H}} \\ &= \beta_{w,0} + \sum_{\ell=1}^{n} \Big[\beta_{w,\ell} \langle K(\mathbf{x}_\ell, \cdot), K(\mathbf{x}_i, \cdot)\rangle_{\mathcal{H}} + \langle h_w^\perp, K(\mathbf{x}_i, \cdot)\rangle_{\mathcal{H}} \Big] \\ &= \beta_{w,0} + \sum_{\ell=1}^{n} \beta_{w,\ell} \langle K(\mathbf{x}_\ell, \cdot), K(\mathbf{x}_i, \cdot)\rangle_{\mathcal{H}} \\ &= \beta_{w,0} + \sum_{\ell=1}^{n} \beta_{w,\ell} K(\mathbf{x}_\ell, \mathbf{x}_i). \end{aligned}$$

Thus, for all $w = 1, 2, \cdots, W-1$, we have

$$\begin{aligned} \|h_w\|_{\mathcal{H}}^2 &= \left\| \sum_{\ell=1}^{n} \beta_{w,\ell} K(\mathbf{x}_\ell, \cdot) + h_w^\perp \right\|_{\mathcal{H}}^2 \\ &= \sum_{\ell=1}^{n} \sum_{\ell'=1}^{n} \beta_{w,\ell}\beta_{w,\ell'} K(\mathbf{x}_\ell, \mathbf{x}_{\ell'}) + \|h_w^\perp\|_{\mathcal{H}}^2, \end{aligned}$$

and for $w = W$, we have

$$\|h_W\|_{\mathcal{H}}^2 = \left\| \sum_{w=1}^{W-1} \sum_{\ell=1}^{n} \beta_{W,\ell} K(\mathbf{x}_\ell, \cdot) \right\|_{\mathcal{H}}^2 + \left\| \sum_{w=1}^{W-1} h_w^\perp \right\|_{\mathcal{H}}^2.$$

Therefore, in order to minimize (11), we must set $h_w^\perp = 0$ and

$$f_w = \beta_{w,0} + \sum_{\ell=1}^{n} \beta_{w,\ell} K(\mathbf{x}_i, \cdot),$$

for all $w = 1, \cdots, W$.

(g) *Notation:* We adopt the following notation

i. $\boldsymbol{\beta}_w := (\beta_{w,1}, \beta_{w,2}, \cdots, \beta_{w,n})^\top \in \mathbb{R}^n$,

ii. $\mathbf{Z} = (\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \cdots, \boldsymbol{\zeta}_n)^\top = (\boldsymbol{\zeta}_{\bullet,1}, \boldsymbol{\zeta}_{\bullet,2}, \cdots, \boldsymbol{\zeta}_{\bullet,W}) \in \mathbb{R}^{n\times W}$, where

$$\boldsymbol{\zeta}_i = (\zeta_{i,1}, \zeta_{i,2}, \cdots, \zeta_{i,W})^\top \in \mathbb{R}^W$$

is the i-th row of $\mathbf{Z}$, $\zeta_{i,w} = [f(\mathbf{x}_i) - y_{i,w}]_+$ and $y_{i,w}$ is the w-th component of the W-vector $\mathbf{y}_i$, and $\boldsymbol{\zeta}_{\bullet,w}$ is the w-th column of $\mathbf{Z}$, for all $i = 1, \cdots, n$ and $w = 1, \cdots, W$,

iii. $\mathbf{L} = (\mathbf{L}_1, \mathbf{L}_2, \cdots, \mathbf{L}_W) = (\mathbf{L}(\mathbf{y}_1), \mathbf{L}(\mathbf{y}_2), \cdots, \mathbf{L}(\mathbf{y}_n))^\top \in \mathbb{R}^{n\times W}$, where $\mathbf{L}_w \in \mathbb{R}^n$ denotes the w-th column of $\mathbf{L}$ and $\mathbf{L}(\mathbf{y}_i) \in \mathbb{R}^W$ is the i-th row of $\mathbf{L}$, for all $i = 1, \cdots, n$ and all $w = 1, \cdots, W$,

iv. $(\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n)^\top = (\mathbf{y}_{\bullet,1}, \mathbf{y}_{\bullet,2}, \cdots, \mathbf{y}_{\bullet,W}) \in \mathbb{R}^{n\times W}$ be the matrix whose i-th row is $\mathbf{y}_i$ and w-th column is $\mathbf{y}_{\bullet,w}$, for all $i = 1, \cdots, n$ and $w = 1, \cdots, W$, and

v. $\mathbf{K} = [K(\mathbf{x}_i, \mathbf{x}_j)] \in \mathbb{R}^{n\times n}$ is the Gram matrix.

(h) *Equivalent Condition to* (10)*:* First note that (10) can be written as

$$\bar{\beta}_0 + \sum_{i=1}^{n} \bar{\beta}_i K(\mathbf{x}_i, \cdot\,) = 0,$$

where $\bar{\beta}_0 = \frac{1}{W}\sum_{w=1}^{W} \beta_{w,0}$ and $\bar{\beta}_\ell = \frac{1}{W}\sum_{w=1}^{W} \beta_{w,\ell}$.
At the n data points, (10) becomes

$$\left(\sum_{w=1}^{W} \beta_{w,0}\right)\mathbf{1}_n + \mathbf{K}\left(\sum_{w=1}^{W} \boldsymbol{\beta}_w\right) = \mathbf{0}_n.$$

If we let

$$\beta^*_{w,0} = \beta_{w,0} - \bar{\beta}_0, \qquad \text{and} \qquad \beta^*_{w,\ell} = \beta_{w,\ell} - \bar{\beta}_\ell,$$

then, we have

$$\begin{aligned} f^*_w(\mathbf{x}_i) &:= \beta^*_{w,0} + \sum_{\ell=1}^{n} \beta^*_{w,\ell} K(\mathbf{x}_\ell, \mathbf{x}_i) \\ &= (\beta_{w,0} - \bar{\beta}_0) + \sum_{\ell=1}^{n} (\beta_{w,\ell} - \bar{\beta}_\ell) K(\mathbf{x}_\ell, \mathbf{x}_i) \\ &= \beta_{w,0} + \sum_{\ell=1}^{n} \beta_{w,\ell} K(\mathbf{x}_\ell, \mathbf{x}_i) - \left(\bar{\beta}_0 + \sum_{\ell=1}^{n} \bar{\beta}_\ell K(\mathbf{x}_\ell, \mathbf{x}_i)\right) \\ &= f_w(\mathbf{x}_i). \end{aligned}$$

In addition, if we let $h^*_w = \sum_{\ell=1}^{n} \beta^*_{w,\ell} K(\mathbf{x}_i, \cdot\,) = \sum_{\ell=1}^{n} (\beta_{w,\ell} - \bar{\beta}_\ell) K(\mathbf{x}_i, \cdot\,)$, we have

$$\sum_{w=1}^{W} \|h^*_w\|^2_{\mathcal{H}} = \sum_{w=1}^{W} \boldsymbol{\beta}_w^\top \mathbf{K} \boldsymbol{\beta}_w - W \bar{\boldsymbol{\beta}}^\top \mathbf{K} \bar{\boldsymbol{\beta}} \le \sum_{w=1}^{W} \boldsymbol{\beta}_w^\top \mathbf{K} \boldsymbol{\beta}_w = \sum_{w=1}^{W} \|h_w\|^2_{\mathcal{H}},$$

where $\bar{\boldsymbol{\beta}} := (\bar{\beta}_1, \bar{\beta}_2, \cdots, \bar{\beta}_n)^\top \in \mathbb{R}^n$. If $\mathbf{K}\bar{\boldsymbol{\beta}} = \mathbf{0}_n$, we must have

$$\sum_{w=1}^{W} \|h^*_w\|^2_{\mathcal{H}} = \sum_{w=1}^{W} \boldsymbol{\beta}_w^\top \mathbf{K} \boldsymbol{\beta}_w - W \bar{\boldsymbol{\beta}}^\top \mathbf{K} \bar{\boldsymbol{\beta}} = \sum_{w=1}^{W} \boldsymbol{\beta}_w^\top \mathbf{K} \boldsymbol{\beta}_w = \sum_{w=1}^{W} \|h_w\|^2_{\mathcal{H}},$$

and $\sum_{w=1}^{W} \beta_{w,0} = 0$. Therefore,

$$0 = W^2 \bar{\boldsymbol{\beta}}^\top \mathbf{K} \bar{\boldsymbol{\beta}} = \left\|\sum_{\ell=1}^{n}\left(\sum_{w=1}^{W} \beta_{w,\ell}\right) K(\mathbf{x}_\ell, \cdot\,)\right\|^2_{\mathcal{H}} = \left\|\sum_{w=1}^{W}\sum_{\ell=1}^{n} \beta_{w,\ell} K(\mathbf{x}_\ell, \cdot\,)\right\|^2_{\mathcal{H}},$$

and, hence,

$$\sum_{w=1}^{W} \sum_{\ell=1}^{n} \beta_{w,\ell} K(\mathbf{x}_\ell, \mathbf{x}) = 0, \qquad \text{for all } \mathbf{x},$$

and

$$\sum_{w=1}^{W} \left(\beta_{w,0} + \sum_{\ell=1}^{n} \beta_{w,\ell} K(\mathbf{x}_\ell, \mathbf{x}) \right) = 0, \qquad \text{for all } \mathbf{x}.$$

As a conclusion, minimizing (11) under the constraint (10) *only* at the n data points is equivalent to minimizing it under (10) for every $\mathbf{x}$.

(i) *Primal Optimization Problem:* The primal problem is

$$\begin{aligned}
\text{minimize} \quad & \frac{1}{n} \sum_{w=1}^{W} \mathbf{L}_w^\top \boldsymbol{\zeta}_{\bullet,w} + \frac{\lambda}{2} \sum_{w=1}^{W} \boldsymbol{\beta}_w^\top \mathbf{K} \boldsymbol{\beta}_w \\
\text{subject to} \quad & \beta_{w,0} \mathbf{1}_n + \mathbf{K}\boldsymbol{\beta}_w - \mathbf{y}_{\bullet,w} \le \boldsymbol{\zeta}_{\bullet,w}, \text{ for all } w = 1, 2, \cdots, W, \\
& \boldsymbol{\zeta}_{\bullet,w} \ge \mathbf{0}_n, \text{ for all } w = 1, \cdots, W, \\
& \left(\sum_{w=1}^{W} \beta_{w,0} \right) \mathbf{1}_n + \mathbf{K} \left(\sum_{w=1}^{W} \boldsymbol{\beta}_w \right) = \mathbf{0}_n.
\end{aligned} \tag{12}$$

The corresponding primal Lagrangian function is

$$\begin{aligned}
& L_P(\{\beta_{w,0}, \boldsymbol{\beta}_w, \boldsymbol{\zeta}_{\bullet w}, \boldsymbol{\alpha}_w, \boldsymbol{\gamma}_w\}_{w=1}^{W}, \boldsymbol{\delta}) \\
& \quad = \frac{1}{n} \sum_{w=1}^{W} \mathbf{L}_w^\top \boldsymbol{\zeta}_{\bullet,w} + \frac{\lambda}{2} \sum_{w=1}^{W} \boldsymbol{\beta}_w^\top \mathbf{K} \boldsymbol{\beta}_w \\
& \qquad + \sum_{w=1}^{W} \boldsymbol{\alpha}_w^\top \big(\beta_{w,0} \mathbf{1}_n + \mathbf{K}\boldsymbol{\beta}_w - \mathbf{y}_{\bullet,w} - \boldsymbol{\zeta}_{\bullet,w} \big) - \sum_{w=1}^{W} \boldsymbol{\gamma}_w^\top \boldsymbol{\zeta}_{\bullet,w} \\
& \qquad + \boldsymbol{\delta}^\top \left(\left(\sum_{w=1}^{W} \beta_{w,0} \right) \mathbf{1}_n + \mathbf{K} \left(\sum_{w=1}^{W} \boldsymbol{\beta}_w \right) \right),
\end{aligned}$$

where $\boldsymbol{\alpha}_w \in \mathbb{R}^n$, $\boldsymbol{\gamma}_w \in \mathbb{R}^n$ and $\boldsymbol{\delta} \in \mathbb{R}^n$ are the nonnegative Lagrangian multipliers.

(j) *KKT Conditions:* The complete set of the Karush-Kuhn-Tucker conditions are

i. Stationarity: for all $w = 1, \cdots, W$,

$$\begin{aligned}
\frac{\partial L_P}{\partial \beta_{w,0}} &= (\boldsymbol{\alpha}_w + \boldsymbol{\delta})^\top \mathbf{1}_n = \mathbf{0}_n, \\
\frac{\partial L_P}{\partial \boldsymbol{\beta}_w} &= \lambda \mathbf{K} \boldsymbol{\beta}_w + \mathbf{K} \boldsymbol{\alpha}_w + \mathbf{K} \boldsymbol{\delta} = \mathbf{0}_n, \\
\frac{\partial L_P}{\partial \boldsymbol{\zeta}_{\bullet,w}} &= \frac{1}{n} \mathbf{L}_w - \boldsymbol{\alpha}_w - \boldsymbol{\gamma}_w = \mathbf{0}_n;
\end{aligned}$$

ii. Primal feasibility:

$$\begin{aligned}\beta_{w,0}\mathbf{1}_n + \mathbf{K}\boldsymbol{\beta}_w - \mathbf{y}_{\bullet,w} &\le \boldsymbol{\zeta}_{\bullet,w}, \text{ for all } w = 1, 2, \cdots, W, \\ \boldsymbol{\zeta}_{\bullet,w} &\ge \mathbf{0}_n, \text{ for all } w = 1, \cdots, W, \\ \left(\sum_{w=1}^{W} \beta_{w,0}\right)\mathbf{1}_n + \mathbf{K}\left(\sum_{w=1}^{W} \boldsymbol{\beta}_w\right) &= \mathbf{0}_n;\end{aligned}$$

iii. Dual feasibility:

$$\boldsymbol{\alpha}_w \ge \mathbf{0}_n, \qquad \text{and} \qquad \gamma_w \ge \mathbf{0}_n, \qquad \text{for all } w = 1, \cdots, W;$$

iv. Complementary slackness:

$$\begin{aligned}\boldsymbol{\alpha}_w^\top\big(\beta_{w,0}\mathbf{1}_n + \mathbf{K}\boldsymbol{\beta}_w - \mathbf{y}_{\bullet,w} - \boldsymbol{\zeta}_{\bullet,w}\big) &= 0, \qquad \text{for all } w = 1, \cdots, W, \\ \boldsymbol{\gamma}_w^\top\boldsymbol{\zeta}_{\bullet,w} &= 0, \qquad \text{for all } w = 1, \cdots, W.\end{aligned}$$

From the KKT conditions above, we have the following

i. $\mathbf{0}_n \le \boldsymbol{\alpha}_w \le \frac{1}{n}\mathbf{L}_w$, for all $w = 1, 2, \cdots, W$;
ii. $\boldsymbol{\delta} = -\frac{1}{W}\sum_{w=1}^{W} \boldsymbol{\alpha}_w =: -\bar{\boldsymbol{\alpha}}$, and $(\boldsymbol{\alpha}_w - \bar{\boldsymbol{\alpha}})^\top\mathbf{1}_n = 0$;
iii. If $\mathbf{K}$ is positive definite, $\boldsymbol{\beta}_w = -\lambda^{-1}(\boldsymbol{\alpha}_w - \bar{\boldsymbol{\alpha}})$. If $\mathbf{K}$ is *not* positive definite, $\boldsymbol{\beta}_w$ is *not* uniquely determined.

(k) *Dual Problem:* The dual problem is

$$\begin{aligned}\text{minimize } & L_D(\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_W) := \frac{1}{2\lambda}\sum_{w=1}^{W}(\boldsymbol{\alpha}_w - \bar{\boldsymbol{\alpha}})^\top\mathbf{K}(\boldsymbol{\alpha}_w - \bar{\boldsymbol{\alpha}}) + \sum_{w=1}^{W}\boldsymbol{\alpha}_w^\top\mathbf{y}_{\bullet,w} \\ \text{subject to } & \mathbf{0}_n \le \boldsymbol{\alpha}_w \le \frac{1}{n}\mathbf{L}_w \text{ for all } w = 1, 2, \cdots, W \\ & (\boldsymbol{\alpha}_w - \bar{\boldsymbol{\alpha}})^\top\mathbf{1}_n = 0 \text{ for all } w = 1, 2, \cdots, W.\end{aligned} \tag{13}$$

(l) *Solution to* $\{\boldsymbol{\beta}_w\}_{w=1}^W$*:* Let $(\widehat{\boldsymbol{\alpha}}_1, \widehat{\boldsymbol{\alpha}}_2, \cdots, \widehat{\boldsymbol{\alpha}}_W)$ be the minimizer of L_D. Then, we have

$$\widehat{\boldsymbol{\beta}}_w = -\frac{1}{\lambda}(\widehat{\boldsymbol{\alpha}}_w - \widehat{\bar{\boldsymbol{\alpha}}}),$$

where $\widehat{\bar{\boldsymbol{\alpha}}} := \frac{1}{W}\sum_{w=1}^{W}\widehat{\boldsymbol{\alpha}}_w$.

(m) *Classifying a New Observation:* The multi-class classification of a new observation $\mathbf{x}_0$ is

$$\underset{w=1,2,\cdots,W}{\arg\max}\,\{\hat{f}_w(\mathbf{x})\},$$

where

$$\hat{f}_w = \hat{\beta}_{w,0} + \sum_{\ell=1}^{n}\hat{\beta}_{w,\ell}K(\mathbf{x}_\ell, \cdot), \qquad \text{for all } w = 1, \cdots, W.$$

V. Support Vector Machines for Regression

1. **Generalizing the Concept of "Margin":**

   (a) In SVM classification, the "margin" is used to determine the amount of separation between two non-overlapping classes of points: the bigger the margin, the more confident we are that the optimal separating hyperplane is a superior classifier;

   (b) A regression analogue for the margin would entail forming a "band" around the true regression function that contains *most* of the points. Points *not* contained within the tube would be described through slack variables.

2. **ε-Insensitive Loss Function:** Let $\mu(\mathbf{x}) = \beta_0 + \mathbf{x}^\top \boldsymbol{\beta}$ be the true regression function. We consider a loss function that *ignores* errors associated with points falling within a certain distance of μ, denoted by $\varepsilon > 0$: for a point $(\mathbf{x}, y)$,

   (a) if $|y - \mu(\mathbf{x})| \leq \varepsilon$, then the loss is taken to be zero;

   (b) if $|y - \mu(\mathbf{x})| > \varepsilon$, then the loss is $|y - \mu(\mathbf{x})| - \varepsilon$.

   In particular, we consider the following *linear ε-insensitive loss function*

   $$\begin{aligned} V_\varepsilon(y, \mu(\mathbf{x})) &:= \max\big\{0, |y - \mu(\mathbf{x})| - \varepsilon\big\} \\ &= \begin{cases} 0, & \text{if } |y - \mu(\mathbf{x})| < \varepsilon \\ |y - \mu(\mathbf{x})| - \varepsilon, & \text{otherwise} \end{cases}. \end{aligned}$$

   Remark. An alternative choice is the *quadratic ε-insensitive loss function* defined as

   $$\begin{aligned} V_\varepsilon(y, \mu(\mathbf{x})) &= \max\big\{0, (y - \mu(\mathbf{x}))^2 - \varepsilon\big\} \\ &= \begin{cases} 0, & \text{if } |y - \mu(\mathbf{x})| < \varepsilon \\ (y - \mu(\mathbf{x}))^2 - \varepsilon, & \text{otherwise} \end{cases}. \end{aligned}$$

3. **Optimization Problem for SVM Regression:**

   (a) *Introducing the Slackness Variables:* Define the slack variables ξ_i and ξ_i' in the following way:

      i. If the point $(\mathbf{x}_i, y_i)$ lies above the ε-tube, then

      $$\xi_i' := y_i - \mu(\mathbf{x}_i) - \varepsilon \geq 0;$$

      ii. if the point $(\mathbf{x}_i, y_i)$ lies below the ε-tube, then

      $$\xi_i := \mu(\mathbf{x}_j) - y_i - \varepsilon \geq 0.$$

(b) *Problem Formulation:* The primal optimization problem is

$$
\begin{aligned}
\underset{\beta_0,\boldsymbol{\beta},\boldsymbol{\xi},\boldsymbol{\xi}'}{\text{minimize}}\ & \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i') \\
\text{subject to}\ & y_i - (\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}) \le \varepsilon + \xi_i', \\
& (\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}) - y_i \le \varepsilon + \xi_i, \\
& \xi_i' \ge 0, \xi_i \ge 0, \qquad \text{for all } i = 1, \cdots, n.
\end{aligned} \tag{14}
$$

The hyper-parameter $C > 0$ is used to balance the flatness of the function μ against our tolerance of deviations larger than ε.

(c) *Primal Lagrangian Function:* Let $\boldsymbol{\xi} := (\xi_1, \xi_2, \cdots, \xi_n)^\top$ and $\boldsymbol{\xi}' := (\xi_1', \xi_2', \cdots, \xi_n')^\top$. The primal Lagrangian function is

$$
\begin{aligned}
L_P(\beta_0, \boldsymbol{\beta}, \boldsymbol{\xi}, \boldsymbol{\xi}') := & \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i') - \sum_{i=1}^{n}\alpha_i(y_i - (\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}) - \varepsilon - \xi_i') \\
& - \sum_{i=1}^{n}\gamma_i(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta} - y_i - \varepsilon - \xi_i) \\
& - \sum_{i=1}^{n}\nu_i \xi_i' - \sum_{i=1}^{n}\zeta_i \xi_i,
\end{aligned}
$$

where $\{\alpha_i\}_{i=1}^n$, $\{\gamma_i\}_{i=1}^n$, $\{\nu_i\}_{i=1}^n$ and $\{\zeta_i\}_{i=1}^n$ are Lagrangian multipliers and are all nonnegative.

Differentiating L_P with respect to β_0, $\boldsymbol{\beta}$, ξ_i and ξ_i' and setting the results to 0 yield

$$
\frac{\partial L_P}{\partial \beta_0} = \sum_{i=1}^{n}(\alpha_i - \gamma_i) = 0, \tag{15}
$$

$$
\frac{\partial L_P}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta} + \sum_{i=1}^{n}(\alpha_i - \gamma_i)\mathbf{x}_i = \mathbf{0}_p, \tag{16}
$$

$$
\frac{\partial L_P}{\partial \xi_i'} = C + \alpha_i - \nu_i = 0, \qquad \text{for all } i = 1, 2, \cdots, n, \tag{17}
$$

$$
\frac{\partial L_P}{\partial \xi_i} = C + \gamma_i - \zeta_i = 0, \qquad \text{for all } i = 1, 2, \cdots, n. \tag{18}
$$

(d) *KKT Conditions:* The full set of KKT conditions for (14) is the following:

i. Stationarity: see (15) - (18);

ii. Primal feasibility:

$$
\begin{aligned}
y_i - (\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}) \le \varepsilon + \xi_i', \qquad & \text{for all } i = 1, 2, \cdots, n, \\
(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}) - y_i \le \varepsilon + \xi_i, \qquad & \text{for all } i = 1, 2, \cdots, n, \\
\xi_i' \ge 0, \xi_i \ge 0, \qquad & \text{for all } i = 1, \cdots, n;
\end{aligned}
$$

iii. Dual feasibility:

$$\alpha_i \geq 0, \quad \nu_i \geq 0, \quad \gamma_i \geq 0, \quad \zeta_i \geq 0, \qquad \text{for all } i = 1, \cdots, n;$$

iv. Complementary Slackness:

$$\begin{aligned}
\alpha_i(y_i - (\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}) - \varepsilon - \xi_i') &= 0, && \text{for all } i = 1, 2, \cdots, n, \\
\gamma_i((\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}) - y_i - \varepsilon - \xi_i) &= 0, && \text{for all } i = 1, 2, \cdots, n, \\
\nu_i \xi_i' &= 0, && \text{for all } i = 1, \cdots, n, \\
\zeta_i \xi_i &= 0, && \text{for all } i = 1, \cdots, n.
\end{aligned}$$

(e) *Dual Problem:* The corresponding dual problem is

$$\underset{\boldsymbol{\alpha}_i, \boldsymbol{\gamma}_i}{\text{maximize}} \; L_D(\boldsymbol{\alpha}, \boldsymbol{\gamma}) \tag{19}$$

where

$$L_D(\boldsymbol{\alpha}, \boldsymbol{\gamma}) := \sum_{i=1}^n y_i(\alpha_i - \gamma_i) - \varepsilon \sum_{i=1}^n (\alpha_i + \gamma_i) - \frac{1}{2} \sum_{i,i'=1}^n (\alpha_i - \gamma_i)(\alpha_{i'} - \gamma_{i'}) \langle \mathbf{x}_i, \mathbf{x}_{i'} \rangle,$$

subject to the constraints

$$\begin{aligned}
& 0 \leq \alpha_i, \; \gamma_i \leq C \text{ for all } i = 1, \cdots, n, \\
& \sum_{i=1}^n (\alpha_i - \gamma_i) = 0, \\
& \alpha_i \gamma_i = 0 \qquad \text{for all } i = 1, 2, \cdots, n.
\end{aligned}$$

Remark. Since $\alpha_i \gamma_i = 0$ for all $i = 1, 2, \cdots, n$, we can never have a set of dual variables α_i, γ_i which are both simultaneously nonzero.

(f) *SVM Regression Function:* If we let $(\widehat{\boldsymbol{\alpha}}^\top, \widehat{\boldsymbol{\gamma}}^\top)^\top$ be the maximizer of L_D in (19), the coefficient vector in the SVM regression function is

$$\widehat{\boldsymbol{\beta}} = \sum_{i=1}^n (\hat{\gamma}_i - \hat{\alpha}_i) \mathbf{x}_i,$$

$\hat{\beta}_0$ can be obtained using the complementary slackness conditions above, and the resulting regression function is

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^n (\hat{\gamma}_i - \hat{\alpha}_i) \langle \mathbf{x}_i, \mathbf{x} \rangle + \hat{\beta}_0,$$

Typically, in $\widehat{\boldsymbol{\beta}}$, there is only a subset of the solution values $(\hat{\gamma}_i^* - \hat{\alpha}_i)$ are nonzero, and the associated data values are called the *support vectors*.

Remark 1. The solution depends on the input values only through the inner products $\langle \mathbf{x}_i, \mathbf{x}_{i'} \rangle$. Hence, we can generalize the methods to richer function spaces by defining an appropriate inner product.

Remark 2. There are two parameters in support vector machine for regression, namely, ε and λ.

(a) ε is a parameter in the loss V_ε, and depends on the scales of y and r. One suggestion is to scale the response and use preset values of ε.

(b) λ can be chosen by cross validation.

4. **Regression and Kernels:** Consider approximating the regression function in terms of a set of basis function $\{h_m\}_{m=1}^M$,

$$f(\mathbf{x}) = \sum_{m=1}^{M} \beta_m h_m(\mathbf{x}) + \beta_0.$$

To estimate $\boldsymbol{\beta} := (\beta_1, \cdots, \beta_M)^\top$ and β_0, we minimize

$$H(f) := \sum_{i=1}^{n} V(y_i - f(\mathbf{x}_i)) + \frac{\lambda}{2} \sum_{m=1}^{M} \beta_m^2$$

for some general error measure V. For any choice of V, the solution $\hat{f} := \arg\min_f H(f)$ has the form

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} \hat{\alpha}_i K(\mathbf{x}, \mathbf{x}_i),$$

where $K(\mathbf{x}, \mathbf{y}) = \sum_{m=1}^{M} h_m(\mathbf{x}) h_m(\mathbf{y})$.

5. **Example:** Let $V(r) = r^2$ and assume $\beta_0 = 0$. Estimate $\boldsymbol{\beta}$ by the penalized least squares criterion

$$H(\boldsymbol{\beta}) = (\mathbf{Y} - \mathbf{H}\boldsymbol{\beta})^\top (\mathbf{Y} - \mathbf{H}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_2^2.$$

The solution $\widehat{\boldsymbol{\beta}}$ satisfies the equation

$$-\mathbf{H}^\top (\mathbf{Y} - \mathbf{H}\widehat{\boldsymbol{\beta}}) + \lambda \widehat{\boldsymbol{\beta}} = \mathbf{0}_M$$

and the fitted values are

$$\hat{\mathbf{Y}} = \mathbf{H}\widehat{\boldsymbol{\beta}}.$$

Note that

$$\mathbf{H}\widehat{\boldsymbol{\beta}} = (\mathbf{H}\mathbf{H}^\top + \lambda \mathbf{I})^{-1} \mathbf{H}\mathbf{H}^\top \mathbf{Y},$$

and the matrix $\mathbf{H}\mathbf{H}^\top \in \mathbb{R}^{n\times n}$ consists of inner products between pairs of observations i, i', i.e.,

$$[\mathbf{H}\mathbf{H}^\top]_{i,i'} = K(\mathbf{x}_i, \mathbf{x}_{i'}).$$

It follows that the predicted value at an arbitrary $\mathbf{x}_0$ satisfy

$$\hat{f}(\mathbf{x}_0) = \mathbf{h}(\mathbf{x}_0)^\top \widehat{\boldsymbol{\beta}} = \sum_{i=1}^{n} \hat{\alpha}_i K(\mathbf{x}_0, \mathbf{x}_i),$$

where $\widehat{\boldsymbol{\alpha}} := (\hat{\alpha}_1, \hat{\alpha}_2, \cdots, \hat{\alpha}_n)^\top = (\mathbf{H}\mathbf{H}^\top + \lambda \mathbf{I})^{-1}\mathbf{Y}$.

Remark. Similar to the support vector machine for classification, we don't need to specify or evaluate the large set of functions $h_1(\mathbf{x}_i), \cdots, h_M(\mathbf{x}_i)$ for all $i = 1, \cdots, n$. We *only* need to evaluate the inner product kernel $K(\mathbf{x}, \mathbf{x}_{i'})$ at n training data points.

References

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. Vol. 1. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Izenman, Alan J (Mar. 2009). *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. en. Springer Science & Business Media.

Lee, Yoonkyung, Yi Lin, and Grace Wahba (2004). "Multicategory Support Vector Machines". In: *Journal of the American Statistical Association* 99.465, pp. 67–81. DOI: 10.1198/016214504000000098. URL: https://doi.org/10.1198/016214504000000098.

Schölkopf, Bernhard and Alexander J. Smola (2002). *Learning with kernels : support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press. URL: http://www.worldcat.org/oclc/48970254.