Dual formulation and Kernel Support Vector Machine

Xin Bing

Department of Statistical Sciences
University of Toronto

Computation of the hard-margin SVM

Primal-formulation:

$$\begin{aligned} &\min_{\mathbf{w},b} \|\mathbf{w}\|_2^2 \\ &\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad i = 1, \ldots, n \end{aligned}$$

- Convex, in fact, a quadratic program. (Stochastic) Gradient descent can be directly used.
- In practice, it is more common to solve the optimization problem based on its dual formulation.

Dual-formulation of the hard-margin SVM

For $\alpha_i \geq 0$ for all $i = 1, \ldots, n$, write the Lagrangian function

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \|\mathbf{w}\|_2^2 + \sum_{i=1}^n \alpha_i \left[1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\right],$$

Taking the derivative w.r.t. $\mathbf{w}$ and b yields

$$\mathbf{w} = \frac{1}{2} \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i, \qquad \sum_{i=1}^n \alpha_i y_i = 0.$$

Plugging into $L(\mathbf{w}, b, \boldsymbol{\alpha})$ yields

$$\begin{aligned} &\frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j - b \sum_{i=1} \alpha_i y_i \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j. \end{aligned}$$

Dual-formulation of the hard-margin SVM

The dual problem is

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0, \ \alpha_i \geq 0, \quad i = 1, \ldots, n. \end{aligned}$$

The K.K.T. conditions ensure the following relationships between the primal and dual formulations.

- Their optimal objective values are equal.
- The optimal solutions $\hat{\mathbf{w}}$ and $\hat{\boldsymbol{\alpha}}$ satisfy

$$\hat{\mathbf{w}} = \frac{1}{2} \sum_{i=1}^{n} \hat{\alpha}_{\mathbf{i}} \, y_i \mathbf{x}_i, \qquad \begin{array}{ll} \hat{\alpha}_i > 0, & \text{if } y_i(\hat{\mathbf{w}}^\top \mathbf{x}_i + \hat{b}) = 1 \\ \hat{\alpha}_i = 0, & \text{if } y_i(\hat{\mathbf{w}}^\top \mathbf{x}_i + \hat{b}) > 1 \end{array}.$$

- The predicted label for any $\mathbf{x}$ is

$$\text{sign}(\hat{\mathbf{w}}^\top \mathbf{x} + \hat{b}).$$

Prime-formulation of the soft-margin SVM

Soft-margin SVM is equivalent to, for some $C = C(K)$,

$$\min_{\mathbf{w},b,\zeta} \ \|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\zeta_i$$

$$\text{s.t. } y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0, \quad i = 1, \dots, n.$$

Dual-formulation of the soft-margin SVM

It can be shown[1] that the dual-formulation of the soft-margin SVM is

$$\max_{\boldsymbol{\alpha}} \ \sum_{i=1}^{n} \alpha_i - \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \ \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{s.t.} \ \sum_{i=1}^{n} \alpha_i y_i = 0, \ 0 \le \alpha_i \le \mathbf{C}, \quad i = 1, \ldots, n.$$

Here $C > 0$ is the tuning parameter.

[1] Chapter 12.2.1 in ESL.

Kernel SVM: extension to non-linear boundary

Recall

$$\max_{\boldsymbol{\alpha}} \ \sum_{i=1}^{n} \alpha_i - \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \ \mathbf{x_i}^\top \mathbf{x_j}$$

$$\text{s.t.} \ \sum_{i=1}^{n} \alpha_i y_i = 0, \ 0 \le \alpha_i \le \mathbf{C}, \quad i = 1, \dots, n.$$

Represent $\mathbf{x}_i$ in different bases, $h(\mathbf{x}_i)$, to have non-linear boundary (in $\mathbf{x}_i$).

The only change is the objective function

$$\max_{\boldsymbol{\alpha}} \ \sum_{i=1}^{n} \alpha_i - \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \ \mathbf{h}(\mathbf{x_i})^\top \mathbf{h}(\mathbf{x_j}).$$

Kernel trick

- We can represent the inner-product $h(\mathbf{x}_i)^\top h(\mathbf{x}_j)$ by using

  $$K(\mathbf{x}_i, \mathbf{x}_j) = h(\mathbf{x}_i)^\top h(\mathbf{x}_j), \qquad \forall i \neq j \in \{1, \ldots, n\}.$$

  The function K is called **kernel** that quantifies the similarity of two feature vectors.

- Regardless how large the space of $h(\mathbf{x}_i)$ is, all we need to compute is the pairwise kernel

  $$K(\mathbf{x}_i, \mathbf{x}_j), \qquad \forall i \neq j \in \{1, \ldots, n\}.$$

  This is known as the **kernel trick**.

Examples of kernel SVM

- Linear:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$$

  with the corresponding $h(\mathbf{x}_i) = \mathbf{x}_i$.

- dth-Degree polynomial:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \mathbf{x}_i^\top \mathbf{x}_j\right)^d .$$

  The corresponding h would be polynomials. For example, consider $d = 2$, $\mathbf{x}_i = x_i$ and $h(\mathbf{x}_i) = [1, \sqrt{2}x_i, x_i^2]$, then

$$K(\mathbf{x}_i, \mathbf{x}_j) = h(\mathbf{x}_i)^\top h(\mathbf{x}_j) = 1 + 2x_i x_j + x_i^2 x_j^2 = \left(1 + \mathbf{x}_i^\top \mathbf{x}_j\right)^2 .$$

- Radial basis: for some $\gamma > 0$,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right).$$

  The corresponding $h(\mathbf{x}_i)$ has **infinite** dimensions!

SVMs with More than Two Classes

One-Versus-One: Let $C = \{1, 2, \ldots, K\}$.

- Construct $\binom{K}{2}$ SVMs for each pair of classes.
  - For classes {1,2}, consider data $(\mathbf{x}_i, y_i)$ with $y_i \in \{1, 2\}$. Let

    $$z_i = -1\{y_i = 1\} + 1\{y_i = 2\}.$$

    Fit SVM by using $(\mathbf{x}_i, z_i)$ with $y_i \in \{1, 2\}$.
  - For classes {1,3}, consider data $(\mathbf{x}_i, y_i)$ with $y_i \in \{1, 3\}$. Let

    $$z_i = -1\{y_i = 1\} + 1\{y_i = 3\}.$$

    Fit SVM by using $(\mathbf{x}_i, z_i)$ with $y_i \in \{1, 3\}$.
  - Repeat for all pairs.
- For each test point $\mathbf{x}_0$, assign it to the majority class predicted by $\binom{K}{2}$ SVMs.

SVMs with More than Two Classes

One-Versus-All

- Construct K SVMs by choosing each class one at a time.
  - For class $\{1\}$, consider ALL data $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$. Let

    $$z_i = 2 \cdot 1\{y_i = 1\} - 1.$$

    Fit SVM and let its parameter be $(\hat{b}^{(1)}, \hat{\mathbf{w}}^{(1)})$.
  - For class $\{2\}$, consider ALL data $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$. Let

    $$z_i = 2 \cdot 1\{y_i = 2\} - 1.$$

    Fit SVM and let its parameter be $(\hat{b}^{(2)}, \hat{\mathbf{w}}^{(2)})$.
  - Repeat for all classes.
- For each test point $\mathbf{x}_0$, assign it to the class

$$\arg\max_{k \in C} \left(\hat{b}^{(k)} + \mathbf{x}_0^\top \hat{\mathbf{w}}^{(k)} \right).$$