EQ2415 – Machine Learning and Data Science HT22

Tutorial 3

A. Honoré, A. Ghosh

1 Kernel substitution & Support Vector Machine

Material: Bishop's Book, chapter 7.1.1, 7.1.2, 7.1.3

Typically, we encounter problems involving parametric models where we have to learn the mapping from an input $\mathbf{x}$ to an output vector $\mathbf{y}$, and we assume that this mapping is parameterized by an unknown parameter vector $\mathbf{w}$. The idea is that during the learning process, we have to learn $\mathbf{w}$ s.t. the output provided by the parametric model (denoting this by $\mathbf{y}(\mathbf{x};\mathbf{w})$) and the meausured output (from given data) is *as close* as possible (this is mathematically realized in terms of a cost function).

But sometimes, we maybe interested in learning another class of methods that are *memory-based.* These methods, crudely speaking, are based on defining a metric that quantifies the similarity between two vectors in the input space and require storing (memorizing) the entire training dataset. For making predictions on unseen data (test data), they require some form of similarity comparison with some / all of the stored training data points, making the inference process quite tedious.

1.1 Linear setup

We consider the classical two-class classification problem using a linear setup [1]

$$y(\mathbf{x};\mathbf{w}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) + b \tag{1}$$

where, a nonlinear feature transformation on the original input $\mathbf{x}$ is applied. Using the idea of kernels that has been introduced in the previous tutorial, we can make a dual representation of (1) that enables us to avoid working directly in the high-dimensional feature space characterized by $\boldsymbol{\phi}(\mathbf{x})$ (we will see this later).

In support vector machines (SVMs), the problem in (1) is solved by constructing a **maximum-margin classifier**. A **margin** is defined as the smallest distance (usually this is the perpendicular distance) between the decision boundary and any of the samples in the training data. The location of this decision boundary is given by a subset of the training data samples known as **support vectors.**

1.2 Formulation of the optimization problem (without slack)

Recall, that the maximum margin solution to (1) is found by solving

$$\mathbf{w}^\star, b^\star = \arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|_2} \min_n \left[t_n y(\mathbf{x}_n;\mathbf{w})\right] \right\} \tag{2}$$

Direct solution to this problem is difficult and using the scale invariance relation of $\mathbf{w}$ and b, we have the alternative problem as

$$\mathbf{w}^\star, b^\star = \arg\min_{\mathbf{w},b} \left\{ \frac{1}{2}\|\mathbf{w}\|_2^2 \right\} \text{ s.t. } t_n y(\mathbf{x}_n;\mathbf{w}) \geq 1, \forall n = 1,2,\dots,N. \tag{3}$$

To solve this constrained optimization problem, formulate the Lagrangian as

$$\mathcal{L}\left(\mathbf{w}, b, \{a_n\}_{n=1}^N\right) = \frac{1}{2}\|\mathbf{w}\|_2^2 - \sum_{n=1}^N a_n \left(t_n \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_n) + b\right) - 1\right) \tag{4}$$

Question 1a. Solve the unconstrained problem in (4) and express the final learning solution in terms of the multipliers $\mathbf{a}$ (this will result in the dual formulation).

Solution: We differentiate (4) w.r.t. to both $\mathbf{w}$ and b. First differentiating w.r.t. $\mathbf{w}$ gives us

$$\begin{aligned}&\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0\\ &\implies 2\mathbf{w} - \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n) = 0\\ &\implies \mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n)\end{aligned} \tag{5}$$

Next, differentiating w.r.t. b gives us

$$\begin{aligned}&\frac{\partial \mathcal{L}}{\partial b} = 0\\ &\implies \sum_{n=1}^{N} a_n t_n = 0\end{aligned} \tag{6}$$

Substituting the results of (5),(6) into (4), we have

$$\begin{aligned}\mathcal{L}_{dual}(\mathbf{a}) &= \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n,m=1}^{N} a_n a_m t_n t_m \boldsymbol{\phi}(\mathbf{x}_n)^\top \boldsymbol{\phi}(\mathbf{x}_m)\\ &= \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n,m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \text{ s.t. } a_n \geq 0, \quad \sum_{n=1}^{N} a_n t_n = 0\end{aligned} \tag{7}$$

The advantage of formulating the problem using the dual formulation is: Kernel trick helps to work in high-dimensional feature spaces without dealing with transformed vectors directly. Also, (7) can be solved using quadratic programming. ■

Question 1b. Can you formulate (4) in terms of a quadratic programming problem using the dual formulation?
Solution: In popular solvers like `cvxpy`, quadratic programming problems are usually formulated as

$$\begin{aligned}\underset{\mathbf{a}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{a}^T P \mathbf{a} + \mathbf{q}^T \mathbf{a}\\ \text{s.t.} \quad & G\mathbf{a} <= \mathbf{h}\\ & \mathbf{A}\mathbf{a} = \mathbf{b}.\end{aligned} \tag{8}$$

This can be done by simply inspection, which gives us $P_{mn} = t_n t_m k(\mathbf{x}_n, \mathbf{x}_m), \mathbf{q} = -\mathbf{1}, \mathbf{h} = 0, \mathbf{G} = -\mathbf{I}, \mathbf{A} = \left[\mathbf{t}_{\boldsymbol{n}}{}^\top\right], \mathbf{b} = \mathbf{0}$. ■

1.3 Classifying new points

We can classify new points as

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \tag{9}$$

Any data point in the training set for which $a_n = 0$ will not contribute to the prediction for new points. From the condition $a_n\left(t_n\left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_n) + b\right) - 1\right) \geq 0$, we have that remaining points should satisfy $t_n\left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_n) + b\right) = 1$. These points *lie on the maximum margin hyperplane* and are referred to as **support vectors**.

Question 1c. Now, that we know this, what is the problem with equation (9)?
Solution: Avoid usage of all the N test points. In practice, we only need points from $\mathcal{S} = \{(\mathbf{x}_n, t_n) : a_n \neq 0, t_n y_n = 1\}$ or equivalently the index set $\mathcal{I}_{\mathcal{S}} = \{n : a_n \neq 0, t_n y_n = 1\}$. ■

1.4 Computing the intercept

Question 1d. How can you compute the intercept b in (1) in a stable way?
Solution: The intercept can be found by using mean of the support vectors as they satisfy $t_n y_n = 1$, so that

$$b = \frac{1}{|\mathcal{I}_\mathcal{S}|} \sum_{n \in \mathcal{I}_\mathcal{S}} \left(t_n - \sum_{m \in \mathcal{I}_\mathcal{S}} a_m t_m k\left(\mathbf{x}_n, \mathbf{x}_m\right) \right) \tag{10}$$

This outer averaging results in a more numerically stable solution, where $|\mathcal{I}_\mathcal{S}|$ is the cardinality of the support set. ■

1.5 Formulation of the optimization problem (with slack)

In practice, data distributions maybe overlapping to some extent, so exact separation doesn't guarantee a good generalization using SVMs. The trick here is to allow some points to be **misclassified** by introducing **slack variables** $\xi_n \geq 0$ s.t. $n = 1, 2, \ldots, N$ with one slack variable per training data point.

1.5.1 Defining slack variables

Slack variables are defined as

$$\xi_n = \begin{cases} 0, & \text{on or inside the correct } \textbf{margin} \text{ boundary} \\ \left| t_n - y\left(\mathbf{x}_n; \mathbf{w}\right) \right|, & \text{otherwise} \end{cases} \tag{11}$$

Question 2a. What will be the value of ξ_n on

1. the wrong side of the decision boundary?

2. on the decision boundary?

3. the correct side of the decision boundary?

Solution: 1. On the wrong side of the decision boundary both t_n and $y\left(\mathbf{x}_n; \mathbf{w}\right)$ will have opposite signs, and as $t_n \in \{-1, 1\}$, so $\xi > 1$.

2. On the decision boundary, $y\left(\mathbf{x}_n; \mathbf{w}\right)$ will be 0, so $\xi = 1$.

3. On the correct side of the decision boundary, the value of $0 \leq \xi < 1$ since both t_n and $y\left(\mathbf{x}_n; \mathbf{w}\right)$ will have same signs, and if exactly equal then they are on the margin boundary. ■

1.5.2 Solving the minimization

Using slack variables, the cost function analogous to (3) now becomes

$$\mathbf{w}^\star, b^\star = \underset{\mathbf{w}, b}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{n=1}^{N} \xi_n \right\} \text{ s.t. } t_n y\left(\mathbf{x}_n; \mathbf{w}\right) \geq 1 - \xi_n, \xi_n \geq 0, \forall n = 1, 2, \ldots, N. \tag{12}$$

The Lagrangian is then analogously formulated as To solve this constrained optimization problem, formulate the Lagrangian as

$$\begin{aligned} \mathcal{L}\left(\mathbf{w}, b, \{a_n\}_{n=1}^N, \{\xi_n\}_{n=1}^N, \{\mu_n\}_{n=1}^N\right) &= \frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{n=1}^{N} \xi_n \\ &\quad - \sum_{n=1}^{N} a_n \left(t_n \left(\mathbf{w}^\top \boldsymbol{\phi}\left(\mathbf{x}_n\right) + b \right) - 1 + \xi_n \right) - \sum_{n=1}^{N} \mu_n \xi_n \end{aligned} \tag{13}$$

Question 2b. Can you formulate (4) in terms of a quadratic programming problem using the dual formulation?
Question 2c. Solve the unconstrained problem in (13) and express the final learning solution in terms of the multipliers **a** (this will result in the dual formulation).

Solution: Use the KKT conditions on the constraints, and then find derivatives w.r.t. $\mathbf{w}, b, \boldsymbol{\xi}$. The KKT conditions give

$$a_n \left(t_n \left(\mathbf{w}^\top \boldsymbol{\phi}\left(\mathbf{x}_n\right) + b\right) - 1 + \xi_n\right) \geq 0 \implies \begin{cases} a_n \geq 0 \\ \left(t_n \left(\mathbf{w}^\top \boldsymbol{\phi}\left(\mathbf{x}_n\right) + b\right) - 1 + \xi_n\right) \geq 0 \\ a_n \left(t_n \left(\mathbf{w}^\top \boldsymbol{\phi}\left(\mathbf{x}_n\right) + b\right) - 1 + \xi_n\right) = 0 \end{cases} \tag{14}$$

and for slack variables

$$\mu_n \xi_n \geq 0 \implies \begin{cases} \mu_n \geq 0 \\ \xi_n \geq 0 \\ \mu_n \xi_n = 0 \end{cases} \tag{15}$$

The results obtained are similar to (8), with additionally $a_n = C - \mu_n$. Now, putting all these back in (12), we have the dual formulation with slack as

$$\begin{aligned} \mathcal{L}_{dual}\left(\mathbf{a}\right) &= \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n,m=1}^{N} a_n a_m t_n t_m \boldsymbol{\phi}^\top\left(\mathbf{x}_n\right) \boldsymbol{\phi}\left(\mathbf{x}_m\right) \\ &= \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n,m=1}^{N} a_n a_m t_n t_m k\left(\mathbf{x}_n, \mathbf{x}_m\right) \text{ s.t. } 0 \leq a_n \leq C, \ \sum_{n=1}^{N} a_n t_n = 0 \end{aligned} \tag{16}$$

The advantage of formulating the problem using the dual formulation is: Kernel trick helps to work in high-dimensional feature spaces without dealing with transformed vectors directly. Also, (16) can be solved using quadratic programming. The formulation into the quadratic programming problem is left as an exercise. ■

Question 2d. How do you

1. predict for new data and characterize the set of support vectors?
2. Compute the intercept term?

Solution: Avoid usage of all the N test points. In practice, we can use (9), but we only need points from $\mathcal{S}_{slack} = \{(\mathbf{x}_n, t_n) : a_n \neq 0, t_n y_n = 1 - \xi_n\}$ or equivalently the index set $\mathcal{I}_{\mathcal{S}_{slack}} = \{n : a_n \neq 0, t_n y_n = 1 - \xi_n\}$.

The intercept can be found by using mean of the support vectors as they satisfy $t_n y_n = 1 - \xi_n$, so that

$$b = \frac{1}{|\mathcal{I}_{\mathcal{S}_{slack}}|} \sum_{n \in \mathcal{I}_{\mathcal{S}_{slack}}} \left(t_n - \sum_{m \in \mathcal{I}_{\mathcal{S}}} a_m t_m k\left(\mathbf{x}_n, \mathbf{x}_m\right)\right) \tag{17}$$

This outer averaging results in a more numerically stable solution, where $|\mathcal{I}_{\mathcal{S}_{slack}}|$ is the cardinality of the support set $\mathcal{I}_{\mathcal{S}_{slack}}$. ■

Question 3. Bishop Prob. 7.2
Solution: We know the original constraint was:

$$t_n y\left(\mathbf{x}_n; \mathbf{w}\right) \geq 1, \forall n = 1, 2, \ldots, N \tag{18}$$

The modified constraint in this question:

$$t_n y\left(\mathbf{x}_n; \mathbf{w}\right) \geq \gamma, \forall n = 1, 2, \ldots, N, \gamma > 0 \tag{19}$$

The loss function is given as

$$\mathcal{L}\left(\mathbf{w}, b, \{a_n\}_{n=1}^{N}\right) = \frac{1}{2}\|\mathbf{w}\|_2^2 - \sum_{n=1}^{N} a_n \left(t_n \left(\mathbf{w}^\top \boldsymbol{\phi}\left(\mathbf{x}_n\right) + b\right) - \gamma\right) \tag{20}$$

Solving for $\mathbf{w}, b$ yields same results as in problem 1a. But how does the dual formulation look like?

$$\begin{aligned}\mathcal{L}_{dual}(\mathbf{a}) &= \gamma \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n,m=1}^{N} a_n a_m t_n t_m \boldsymbol{\phi}(\mathbf{x}_n)^\top \boldsymbol{\phi}(\mathbf{x}_m) \\ &= \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n,m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \text{ s.t. } a_n \geq 0, \quad \sum_{n=1}^{N} a_n t_n = 0\end{aligned} \tag{21}$$

Then we substitute variables $a_n\gamma \rightarrow a'_n$ and we get

$$\begin{aligned}\mathcal{L}_{dual}(\mathbf{a}') &= \sum_{n=1}^{N} a'_n - \frac{1}{2}\sum_{n,m=1}^{N} a'_n a'_m \frac{1}{\gamma^2} t_n t_m \boldsymbol{\phi}(\mathbf{x}_n)^\top \boldsymbol{\phi}(\mathbf{x}_m) \\ &= \sum_{n=1}^{N} a'_n - \frac{1}{2}\sum_{n,m=1}^{N} a'_n a'_m t'_n t'_m k(\mathbf{x}_n, \mathbf{x}_m) \text{ s.t. } a'_n \geq 0, \quad \sum_{n=1}^{N} a'_n t'_n = 0, t'_n = \frac{t_n}{\gamma}\end{aligned} \tag{22}$$

Hence, the form of the optimization problem still remains the same, with the labels being rescaled. Since $\gamma > 0$, this doesn't affect the problem. ■

Question 4. Bishop Prob. 7.4
Solution: We know the margin is the perpendicular distance from the decision boundary to the closest point $\mathbf{x}_n$ from the data set, so we have to show that

$$\frac{1}{\rho^2} = \|\mathbf{w}\|_2^2 \tag{23}$$

We know from results in Problem 1b-c that the optimal conditions for SVM ensure that $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n)$ and that $\sum_{n=1}^{N} a_n t_n = 0$. We also know that for a point closest to the dec. boundary $t_n y_n = 1$. Using these facts, we have

$$\begin{aligned}\|\mathbf{w}\|_2^2 &= \mathbf{w}^\top \mathbf{w} \\ &= \mathbf{w}^\top \left(\sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n)\right) \\ &= \sum_{n=1}^{N} a_n t_n \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_n) \\ &= \sum_{n=1}^{N} a_n t_n \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_n) + b - b\right) \\ &= \sum_{n=1}^{N} a_n t_n \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_n) + b\right) - b\left(\sum_{n=1}^{N} a_n t_n\right) \\ &= \sum_{n=1}^{N} a_n t_n \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_n) + b\right) \\ &= \sum_{n=1}^{N} a_n \quad \left(\text{ for points on the margin } t_n\left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_n) + b\right) = 1\right)\end{aligned} \tag{24}$$

■

References

[1] Bishop, Christopher M., and Nasser M. Nasrabadi. Pattern recognition and machine learning. Vol. 4. No. 4. New York: springer, 2006.