Machine Learning and Neural Networks (MATH3431) Epiphany term, 2023

Handout 6: Support Vector Machines

Lecturer & author: Georgios P. Karagiannis georgios.karagiannis@durham.ac.uk

Aim. To introduce the Support Vector Machines as a procedure. Motivation, set-up, description, computation, and implementation. We focus on the classical treatment.

Reading list & references:

(1) Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
- Ch. 15 (pp. 167-170, 171-172, 176-177) Support Vector Machine

(2) Bishop, C. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: Springer.
- Ch. 7.1 Sparse Kernel Machines/Maximum marginal classifiers

1. Intro and motivation

Note 1. Support Vector Machines (SVM) is a ML procedure for learning linear predictors in high-dimensional feature spaces with regards the sample complexity challenges. Due to a duality property, SVM have sparse solutions, so that predictions for new inputs depend only on quantities evaluated at a subset of the training data points.

Definition 2. Let $w \neq 0$. Hyperplane in space $\mathcal{X} \subseteq \mathbb{R}^d$ is called the sub-set

$$S = \left\{ x \in \mathbb{R}^d : \langle w, x \rangle + b = 0 \right\}.$$

It separates $\mathcal{X}$ in two half-spaces

$$S_+ = \left\{ x \in \mathbb{R}^d : \langle w, x \rangle + b > 0 \right\}$$

and

$$S_- = \left\{ x \in \mathbb{R}^d : \langle w, x \rangle + b < 0 \right\}$$

Definition 3. Halfspace (hypothesis space) is hypotheses class $\mathcal{H}$ designed for binary classification problems, $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{-1, +1\}$ defined as

$$\mathcal{H} = \left\{ x \longmapsto \operatorname{sign}\left(\langle w, x \rangle + b \right) : w \in \mathbb{R}^d, b \in \mathbb{R} \right\},$$

where b is called bias.

Definition 4. Each **halfspace hypothesis** $h \in \mathcal{H}$ has form

$$h_{w,b}(x) = \operatorname{sign}\left(\langle w, x \rangle + b \right) \tag{1.1}$$

, it takes an input in $\mathcal{X} \subseteq \mathbb{R}^d$ and returns an output in $\mathcal{Y} = \{-1, +1\}$. We may refer to it as halfspace (w, b) as this setting determines it.

Note 5. Let $S = \{(x_i, y_i)\}_{i=1}^m$ be a training set of examples with $x_i \in \mathbb{R}^d$ the features and $y_i \in \{-1, +1\}$ the labels.

Definition 6. The training set S is **linearly separable** if there exists a halfspace (w, b) such that for all $i = 1, ..., n$

$$y_i = \text{sign}\left(\langle w, x_i \rangle + b\right)$$

or equivalently

$$y_i \left(\langle w, x_i \rangle + b\right) > 0$$

Note 7. Let the loss be $\ell\left((w, b), z\right) = 1\left(y_i \neq \text{sign}\left(\langle w, x_i \rangle + b\right)\right)$, and hence the Empirical Risk Function be $R_S\left(w, b\right) = \frac{1}{m}\sum_{i=1}^m \ell\left((w, b), z_i\right)$. The Empirical Risk Minimisation (ERM) halfspace (w^*, b^*) is

$$(w^*, b^*) = \arg\min_{w,b}\left(R_S\left(w, b\right)\right) = \arg\min_{w,b}\left(\frac{1}{m}\sum_{i=1}^m \ell\left((w, b), z_i\right)\right)$$

Definition 8. Margin of a hyper-plane with respect to a training set is defined to be the minimal distance between a point in the training set and the hyper-plane.

Note 9. Support Vector Machines (SVM) aims at learning the maximum margin separating hyper-plane Figure (1.1; Right). The rational is that if a hyperplane has a large margin, then it will still separate the training set even if we slightly perturb each instance.

Example 10. Figure (1.1; Left) shows two different separating hyper-planes for the same data set, Figure (1.1; Right) shows the maximum margin hyper-plane: the margin γ is the distance from the hyper-plane (solid line) to the closest points in either class (which touch the parallel dotted lines). It is reasonable to prefer as a predictive rule the hyperplane on the right.

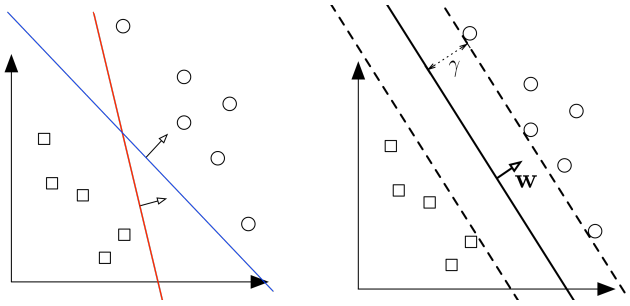


FIGURE 1.1

2. Hard Support Vector Machine

Note 11. Hard Support Vector Machine (Hard-SVM) is the learning rule in which we return an ERM hyperplane that separates the training set with the largest possible margin.

Assumption 12. *Assume the training sample $S = \{(x_i, y_i)\}_{i=1}^m$ is linearly separable.*

Algorithm 13. *(Hard-SVM) Given a linearly separable training sample $S = \{(x_i, y_i)\}_{i=1}^m$ the Hard-SVM rule for the binary classification problem is:*

Solve[1]

$$\left(\tilde{w}, \tilde{b}\right) = \arg\min_{(w,b)} \|w\|_2^2 \tag{2.1}$$

$$\text{subject to: } y_i \left(\langle w, x_i \rangle + b\right) \geq 1, \ \forall i = 1, ..., m \tag{2.2}$$

Scale

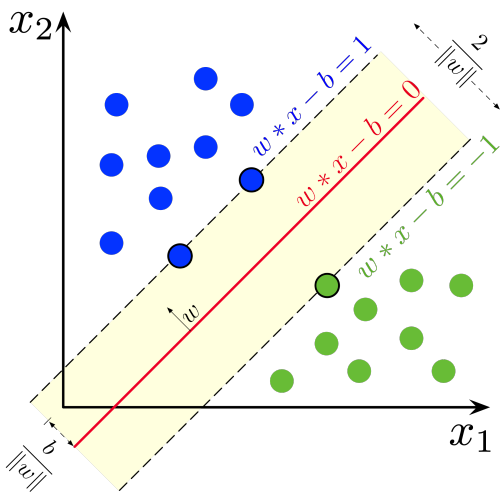
$$\hat{w} = \frac{\tilde{w}}{\|\tilde{w}\|}, \text{ and } \hat{b} = \frac{\tilde{b}}{\|\tilde{w}\|}$$

Note 14. Following we show why Algorithm 13 produces a Hard-SVM hyperplane stated in Note 11.

Fact 15. *The distance between a point x and the hyperplane defined by (w, b) with $\|w\| = 1$ is $|\langle w, x \rangle + b|$.*

Proof. We skip it. □

Note 16. On the right, see the geometry of Algorithm 13.



Note 17. Hard-SVM selects two parallel hyperplanes that separate the two classes of data so that the distance between them is as large as possible. The predictive hyperplane (rule) is the hyperplane that lies halfway between them.

Note 18. Hard-SVM in Algorithm 13 searches for the hyperplane with minimum norm w among all those that separate the data and have distance greater or equal to 1.

Proof. (Sketch of the proof of Algorithm 13)

(1) Based on Note 11, and Fact 15, the closest point in the training set to the separating hyperplane has distance

$$\min_i \left(|\langle w, x_i \rangle + b|\right)$$

hence, by definition, the Hard-SVM hypothesis should be such as

$$(w^*, b^*) = \arg\max_{(w,b):\|w\|=1} \left(\min_i \left(|\langle w, x_i \rangle + b|\right)\right) \tag{2.3}$$

$$\text{subject to } y_i \left(\langle w, x_i \rangle + b\right) > 0, \ \forall i = 1, ..., m \tag{2.4}$$

(2) If there is a solution in (2.3) then (2.3) is equivalent to

$$(w^*, b^*) = \arg\max_{(w,b):\|w\|=1} \left(\min_i \left(y_i \left(\langle w, x_i \rangle + b\right)\right)\right) \tag{2.5}$$

[1] It is a quadratic programming problem.

(3) Next we show that 2.5 is equivalent to the output of Algorithm 13; i.e. $(w^*, b^*) = \left(\hat{w}, \hat{b}\right)$.
Let $\gamma^* := \min_i \left(|\langle w^*, x_i\rangle + b^*|\right)$. Firstly, because

$$y_i \left(\langle w^*, x_i\rangle + b^*\right) \geq \gamma^* \;\Leftrightarrow\; y_i \left(\langle \frac{w^*}{\gamma^*}, x_i\rangle + \frac{b^*}{\gamma^*}\right) \geq 1$$

$\left(\frac{w^*}{\gamma^*}, \frac{b^*}{\gamma^*}\right)$ satisfies condition (2.2). Secondly, I have $\|w_0\| \leq \left\|\frac{w^*}{\gamma^*}\right\| = \frac{1}{\gamma^*}$ because of (2.1) and because of $\|w^*\| = 1$. Hence, for all $i = 1, ..., m$, it is

$$y_i \left(\langle \hat{w}, x_i\rangle + \hat{b}\right) = \frac{1}{\|w^*\|} y_i \left(\langle w_0, x_i\rangle + b_0\right) \geq \frac{1}{\|w^*\|} \geq \gamma^*$$

Hence $\left(\hat{w}, \hat{b}\right)$ is the optimal solution of (2.5).

□

Definition 19. Homogeneous halfspaces in SVM is the case where the halfspaces pass from the origin; that is when the bias term in 2.2 is zero $b = 0$.

3. Soft Support Vector Machine

Note 20. Hard-SVM assumes the strong Assumption 12 that the training set is linearly separable, that might not always the case, and hence there is need to derive a procedure that weakens this assumption.

Note 21. Soft Support Vector Machine (Soft-SVM) aims to relax the strong assumption of Hard-SVM that the training set is linearly separable (2.4) with purpose to be extend the scope of application. Soft-SVM is given below. I.e., Soft-SVM does not assume Assumption 12.

Algorithm 22. *(Soft-SVM) Given a training sample $S = \{(x_i, y_i)\}_{i=1}^m$ the Soft-SVM rule for the binary classification problem is:*

Solve

$$(w^*, b^*, \xi^*) = \underset{(w,b,\xi)}{\arg\min} \left(\lambda \|w\|_2^2 + \frac{1}{m} \sum_{i=1}^m \xi_i\right) \tag{3.1}$$

$$\text{subject to: } y_i \left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i, \;\forall i = 1, ..., m \tag{3.2}$$

$$\xi_i \geq 0, \;\forall i = 1, ..., m \tag{3.3}$$

Note 23. To relax the linearly separable training set assumption, Soft-SVM relies on replacing the "harder" constraint (2.2) with the "softer" one in 3.2 through the introduction of non-negative unknown quantities $\{\xi_i\}_{i=1}^m$ controlling how much the separability assumption (2.2) is violated. Soft-SVM learns all (w, b, ξ) via the minimization part in (3.1) where the trade off between the two terms is controlled via the user specified parameter λ.

Proposition 24. *Consider the hinge loss function*

$$\ell\left((w, b), z\right) = \max\left(0, 1 - y\left(\langle w, x\rangle + b\right)\right)$$

and hence the Empirical Risk Function

$$R_S((w,b)) = \frac{1}{m}\sum_{i=1}^{m}\max\left(0, 1 - y_i\left(\langle w, x_i\rangle + b\right)\right)$$

Then the solution of Algorithm 22 is equivalent to the regularization problem

$$(w^*, b^*) = \arg\min_{(w,b)}\left(R_S((w,b)) + \lambda\|w\|_2^2\right)$$

Proof. In Algorithm 22, we consider

$$\arg\min_{(w,b)}\left(\min_{\xi}\left(\lambda\|w\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i\right)\right) \tag{3.4}$$

Consider (w,b) fixed and focus on the inside minimization. From (3.2), it is $\xi_i \geq 1 - y_i\left(\langle w^*, x_i\rangle + b^*\right)$, and from (3.3), it is $\xi_i \geq 0$. If $y_i\left(\langle w, x_i\rangle + b\right) \geq 1$, the best assignment in 3.4 is $\xi_i = 0$ because it is $\xi_i \geq 0$ from (3.3) and I need to minimize (3.4) wrt ξ_i's. If $y_i\left(\langle w, x_i\rangle + b\right) \leq 1$, the best assignment in (3.4) is $\xi_i = 1 - y_i\left(\langle w, x_i\rangle + b\right)$ because I need to minimize w.r.t ξ. Hence $\xi_i = \max\left(0, 1 - y_i\left(\langle w, x_i\rangle + b\right)\right)$. □

Note 25. Hence the Soft-SVM is a binary classification problem with hinge loss function and regularization term biasing toward low norm separators.

Note 26. Given Proposition 24, Soft-SVM in Algorithm 22 can be learned via any variation of SGD, eg online SGD (batch size $m = 1$) with recursion

$$\varpi^{(t+1)} = \varpi^{(t)} - \eta_t v_t$$

where $v_t = \begin{cases} y^{(t)}\langle\varpi^{(t)}, \chi^{(t)}\rangle & \text{if } y^{(t)}\langle\varpi, \chi^{(t)}\rangle \geq 1 \\ -y^{(t)}\chi^{(t)} & \text{otherwise} \end{cases}$, $\varpi = \left(b^{(t)}, w^{(t)}\right)^\top$ and $\chi = \left(1, x^{(t)}\right)^\top$.

4. Support vectors

Lemma 27. *(Fritz John optimality conditions) Suppose that*

$$\begin{aligned} w^* &= \arg\min_{w} f(w) \\ s.t. g_i(w) &\leq 0, \ \ \forall i = 1, ..., m \\ h_j(x) &= 0, \ \ \forall j = 1, ..., n \end{aligned}$$

where f, g_1, ..., g_m are diffentiable. Then there exists $\alpha_i \geq 0$ for $i = 1, ..., m$ and $\beta_j \in \mathbb{R}$ such that

$$\nabla_w f(w^*) + \sum_{i\in I}\alpha_i\nabla_w g_i(w^*) + \sum_{j\in\mathcal{J}}\beta_j\nabla_w h_j(w^*) = 0$$

where $I = \{i : g_i(w^*) = 0\}$.

Theorem 28. *Let $\hat{w}$ such that*

$$\hat{w} = \arg\min_{w} \|w\|_2^2 \tag{4.1}$$

$$\text{subject to: } y_i\langle w, x_i\rangle \geq 1, \ \forall i = 1, ..., m \tag{4.2}$$

then there exists coefficients $\alpha_i \in \mathbb{R}$ for $i = 1, ..., m$ such that [2]

$$\hat{w} = \sum_{i \in I} \alpha_i x_i$$

where $I = \{i : \langle \hat{w}, x_i\rangle = 1\}$.

Definition 29. Support vectors are called the examples $\{x_i : i \in I\}$ in the training data set.

Note 30. The support vectors have distance $1/\|\hat{w}\|$ from the separating hyperplane. See Figure 4.1.

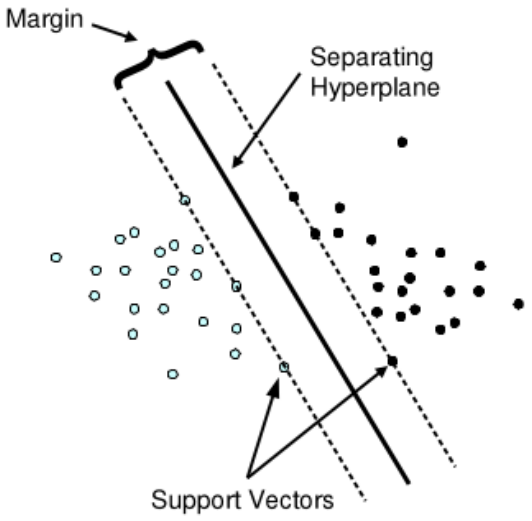


FIGURE 4.1

5. DUALITY

Fact 31. *(Karush–Kuhn–Tucker (KKT) conditions) Let $f : \mathbb{R}^n \to \mathbb{R}$ be an objective convex function, let $\{g_i : \mathbb{R}^n \to \mathbb{R}\}_{i=1}^m$ inequality constraint convex functions, and let $\{h_j : \mathbb{R}^n \to \mathbb{R}\}_{j=1}^n$ equality constraint functions.*

Primal problem: *Consider the following convex nonlinear minimization problem, with solution x^**

$$\begin{aligned} p^* = \underset{x}{minimize}\,(f(x)) \\ s.t. g_i(x) \leq 0, \ \forall i = 1, ..., m \\ h_j(x) = 0, \ \forall j = 1, ..., n \end{aligned} \tag{5.1}$$

[2] y_i is absorbed in α_i's

Lagrangian function: *To the problem we associate the Lagrangian* $L : \mathbb{R}^d \times \mathbb{R}^m \times \mathbb{R}^n$ *with*

$$L(x, \alpha, \beta) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x) + \sum_{j=1}^{n} \beta_j h_j(x)$$

Lagrangian dual problem: *The associated Lagrangian dual problem is*

$$\begin{aligned} d^* = \underset{\alpha}{maximize} \left(\min_x (L(x, \alpha, \beta)) \right) \\ s.t. \alpha_i \geq 0 \\ \beta_j \in \mathbb{R} \end{aligned}$$

where

$$\tilde{L}(\alpha, \beta) := \min_x (L(x, \alpha, \beta))$$

is called dual function

(Weak duality): *In general it is* $p^* \geq d^*$.

(Strong duality via Slater condition): *If the primal problem (5.1) is convex, and satisfies the weak Slater's condition, i.e.*

$$(\exists x_0 \in \mathcal{D}) \ : \ (g_i(x_0) < 0, \ \forall i = 1, ..., n) \text{ and } (h_i(x_0) = 0, \ \forall i = 1, ..., n)$$

then strong duality holds, that is: $p^* = d^*$. *It other words*

$$\min_x \max_{\alpha \geq 0, \beta} (L(x, \alpha, \beta)) = \max_{\alpha \geq 0, \beta} \min_x (L(x, \alpha, \beta))$$

Solution: *With an optimal vector* x^* *in (5.1) there is associated a non-negative vector* α^* *and vector* β^* *such that*

$$0 = \nabla_{(x, \alpha, \beta)} L(x^*, \alpha^*, \beta^*)$$

KKT conditions: *Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient conditions for* x^* *to be local minimum*

$$\begin{aligned} 0 =& \nabla f(x^*) + \sum_{j=1}^{n} \beta_j \nabla h_j(x^*) + \sum_{i=1}^{m} \alpha_i \nabla g_i(x^*) && \text{Stationarity} \\ & g_i(x^*) \leq 0, \ \forall i = 1, ..., m && \text{Primal feasibility} \\ & h_j(x^*) = 0, \ \forall j = 1, ..., n \\ & \alpha_i \geq 0 \ \forall i = 1, ..., m && \text{Dual feasibility} \\ & \alpha_i g_i(x^*) = 0, \ \forall i = 1, ..., m && \text{Complementary slackness} \end{aligned} \tag{5.2}$$

5.1. **Hand SVM.**

Note 32. The problem in Hard-SVM Algorithm 13 (2.2) and (2.2), can be re-writen in the form

$$\min_w \left(\frac{1}{2} \|w\|_2^2 + g(w) \right) \tag{5.3}$$

for

$$g(w) = \max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left(\langle w, x_i \rangle + b\right)\right) = \begin{cases} 0 & \text{if } y_i \left(\langle w, x_i \rangle + b\right) \geq 1 \\ \infty & \text{else} \end{cases}$$

Hence (5.3) is equivalent to

$$\min_{w} \max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \left(\frac{1}{2} \|w\|_2^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left(\langle w, x_i \rangle + b\right)\right) \right)$$

$$\geq \max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \min_{w,b} \left(\underbrace{\frac{1}{2} \|w\|_2^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left(\langle w, x_i \rangle + b\right)\right)}_{= L(w,b;\alpha)} \right) \tag{5.4}$$

where (5.4) is called the weak duality. Strong duality is when the equality hold. In our case the equality holds, but we do not go into details. Here, keeping α fixed, $L(w, \alpha, b)$ is minimized when

$$0 = \nabla_w L(w, \alpha, b)\big|_{(w^*, b^*)} \implies w^* = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{5.5}$$

$$0 = \nabla_b L(w, \alpha, b)\big|_{(w^*, b^*)} \implies 0 = \sum_{i=1}^{m} \alpha_i y_i \tag{5.6}$$

The dual function is

$$\begin{aligned} \tilde{L}(\alpha) = \min_{w,b} \left(L(w, \alpha, b)\right) &= \frac{1}{2} \left\| \sum_{i=1}^{m} \alpha_i y_i x_i \right\|_2^2 - \sum_{i=1}^{m} \alpha_i \left(y_i \left(\langle \sum_{j=1}^{m} \alpha_j y_j x_j, x_i \rangle + b \right) - 1 \right) \\ &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_j x_i \rangle \end{aligned}$$

Then dual problem is

$$\max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_j x_i \rangle \right)$$

Note 33. The Dual problem of the (Primal) problem in Hard-SVM Algorithm 13 is

$$\alpha^* = \arg \max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_j, x_i \rangle \right) \tag{5.7}$$

$$\text{subject to } 0 = \sum_{i=1}^{m} \alpha_i y_i$$

Note 34. Once the optimal α^* are computed from (5.7), the optimal weights w^* can be computed (from (5.5)) as

$$w^* = \sum_{i=1}^{m} \alpha_i^* y_i x_i \tag{5.8}$$

Note 35. If $\alpha_i^* = 0$, the example (x_i, y_i) does not contribute to (5.8). If $\alpha_i^* \neq 0$, (x_i, y_i) contributes to (5.8). Moreover if $\alpha_i^* \neq 0$ KKT condition (5.2)

$$\alpha_i^* \left(y_i \left(\langle w^*, x_i \rangle + b^* \right) - 1 \right) = 0$$

implies that $y_i \left(\langle w^*, x_i \rangle + b^* \right) = 1$ meaning that x_i is on the boundary of the margin. Features x_i associated to $\alpha_i^* \neq 0$ which contribute to the evaluation of the optimal weights w^* and hence the evaluation of the separating predictive rule $h_{w,b}$ are called **supporting vectors** (see Figure 4.1).

Note 36. Given optimal α^*, and letting $\mathcal{I} = \left\{ i : y_i \left(\langle w^*, x_i \rangle + b^* \right) - 1 = 0 \right\}$, 5.8 becomes

$$w^* = \sum_{i \in \mathcal{I}} \alpha_i^* y_i x_i \tag{5.9}$$

Note 37. The bias can be computed by KKT condition (5.2), it is

$$\alpha_i^* \left(y_i \left(\langle w^*, x_i \rangle + b^* \right) - 1 \right) = 0$$

Consider $\mathcal{I} = \left\{ i : y_i \left(\langle w^*, x_i \rangle + b^* \right) - 1 = 0 \right\}$ then, for $i \in \mathcal{I}$ it is

$$y_i^2 \left(\langle w^*, x_i \rangle + b^* \right) - y_i = 0 \overset{y_i^2 = 1}{\Longrightarrow} \langle w^*, x_i \rangle + b^* - y_i = 0$$

and summing up all $i \in \mathcal{I}$, I get

$$\sum_{i \in \mathcal{I}} \left(\langle w^*, x_i \rangle + b^* - y_i \right) = 0 \implies b^* = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \left(y_i - \langle w^*, x_i \rangle \right) \tag{5.10}$$

Note 38. Therefore the predictive halfspace hypothesis is given as

$$\begin{aligned} h_{w,b}(x) &= \operatorname{sign} \left(\langle w^*, x \rangle + b^* \right) \\ &= \operatorname{sign} \left(\sum_{i \in \mathcal{I}} \alpha_i^* y_i \langle x_i, x \rangle + b^* \right) \end{aligned} \tag{5.11}$$

with $\mathcal{I} = \left\{ i : y_i \left(\langle w^*, x_i \rangle + b^* \right) - 1 = 0 \right\}$.

Note 39. Compared to the Primal problem (Algorithm 13), the dual problem (Note 33) is computationally desirable in cases that the training data size m is smaller that the number of the dimentionality of the feature space d; i.e. $d \gg m$. The solution of the Primal quadratic programming problem in $d + 1$ and complexity $O(d + 1)$, while the dual quadratic programming problem has m variables and complexity $O(m)$.

Note 40. Compared to the Primal problem, the dual problem (Note 33), can provide sparse solutions relying on a small number of support vectors (see 5.9, 5.10, and 5.11).

Note 41. The importance of the existence of the dual problem is that eg 5.7 and (5.11) involves the inner products between instances and does not require the direct access to specific elements of features within instance. For instance, if we consider the hypothesis (1.1) as an expansion of bases $h_{w,b}(x) = \operatorname{sign} \left(\langle w, \phi(x) \rangle + b \right)$ with $\phi(x) = \left(\phi_1(x), ..., \phi_d(x) \right)$ with a large d such as $d \gg m$ then,

based on (5.11), the separation rule is

$$h_{w,b}(x) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i k\left(x', x\right) + b\right)$$

and 5.7 becomes

$$\alpha^* = \arg \max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j k\left(x', x\right)\right)$$

with $k(x', x) = \langle \phi(x'), \phi(x) \rangle$. As we will see in the "kernel methods" lecture, in specific case, one can specify the "kernel" function $k(\cdot, \cdot)$ (having with specific desirable properties) avoiding the direct specification of a possibly high-dimensional dictionary of features $\{\phi_j(\cdot)\}$.

Note 42. In the Soft-SVM case, the solution is the same as in the Hard-SVM (5.8) and (5.11), the only difference is that in the quadratic programming problem (5.7) it is subject to $\alpha_j \in [0, C]$ where $C = 1/2\lambda$, it is called "cost" and controls the cost of having the constriant violation by adding the $\xi_i' s$.

Appendix A. Recall

The following is part of " Handout 1: Elements of convex learning problems"

Definition 43. Convex learning problem is a learning problem $(\mathcal{H}, \mathcal{Z}, \ell)$ that the hypothesis class $\mathcal{H}$ is a convex set, and the loss function ℓ is a convex function for each example $z \in \mathcal{Z}$.

Definition 44. Convex-Lipschitz-Bounded Learning Problem $(\mathcal{H}, \mathcal{Z}, \ell)$ with parameters ρ, and B, is called the learning problem whose the hypothesis class $\mathcal{H}$ is a convex set, for all $w \in \mathcal{H}$ it is $\|w\| \leq B$, and the loss function $\ell(\cdot, z)$ is convex and ρ-Lischitz function for all $z \in \mathcal{Z}$.

Definition 45. Convex-Smooth-Bounded Learning Problem $(\mathcal{H}, \mathcal{Z}, \ell)$ with parameters β, and B, is called the learning problem whose the hypothesis class $\mathcal{H}$ is a convex set, for all $w \in \mathcal{H}$ it is $\|w\| \leq B$, and the loss function $\ell(\cdot, z)$ is convex, nonnegative, and β-smooth function for all $z \in \mathcal{Z}$.