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

Draft 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. 7.2 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.

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 $h_{w,b}(x) = \operatorname{sign}\left(\langle w, x \rangle + b\right)$, 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 $R_S(w, b) = \frac{1}{m}\sum_{i=1}^{m} \ell\left((w, b), z_i\right)$. The Empirical Risk Minimisation (ERM) halfspace (w^*, b^*) is

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

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.

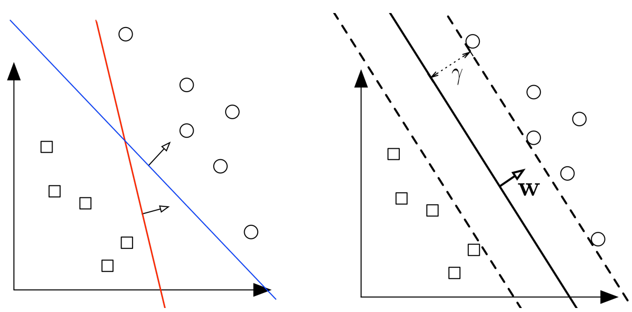


FIGURE 1.1

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

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

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

Scale

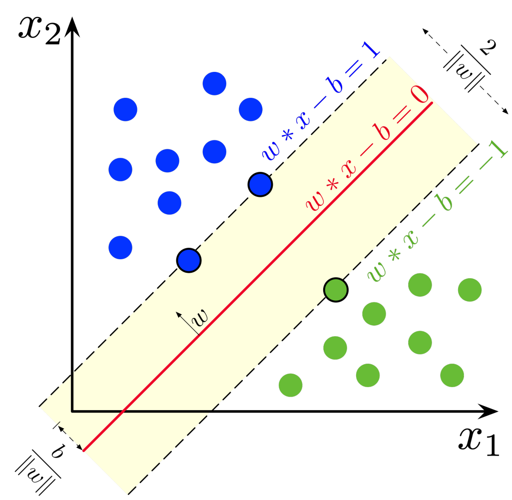
$$\hat{w}=\frac{\tilde{w}}{\|\tilde{w}\|},\ \textit{and } \hat{b}=\frac{\tilde{b}}{\left\|\tilde{b}\right\|}$$

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

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

(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 $\gamma^* := \min_i (|\langle w^*, x_i\rangle + b^*|)$. Firstly, because

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

$\left(\frac{w^*}{\gamma^*}, \frac{b^*}{\gamma^*}\right)$ 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

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

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

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

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

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

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*

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

and hence the Empirical Risk Function

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

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

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

Proof. In Algorithm 22, we consider

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

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 m\xi . 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 $\chi = \left(1, x^{(t)}\right)^\top$.

Appendix A. Recall

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

Definition 27. 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 28. 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 $\ell\left(\cdot, z\right)$ is convex and ρ-Lischitz function for all $z \in \mathcal{Z}$.

Definition 29. 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 $\ell\left(\cdot, z\right)$ is convex, nonnegative, and β-smooth function for all $z \in \mathcal{Z}$.