

# Machine learning theory

## Support vector machines

Hamid Beigy

Sharif university of technology

April 24, 2023





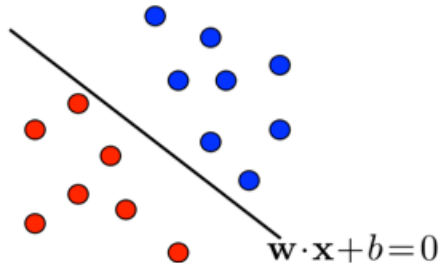
1. Linear classifier
2. Support vector machine
3. Margin theory
4. Summary

## Linear classifier

---



1. In this session, we will study the family of linear classifiers, one of the most useful families of hypothesis classes.
2. Many learning algorithms that are being widely used in practice rely on linear predictors because of
  - the ability to learn them efficiently in many cases,
  - linear predictors are intuitive,
  - are easy to interpret, and
  - fit the data reasonably well in many natural learning problems.
3. A linear classifier separates different classes by a linear separator.

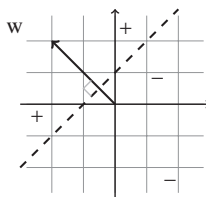




1. **Training data:** sample drawn iid from set  $\mathcal{X} \subseteq \mathbb{R}^n$  according to some distribution  $\mathcal{D}$ .

$$S = \{(x_1, y_1), \dots, (x_m, y_m)\} \in \mathcal{X} \times \{-1, +1\}.$$

2. **Problem:** find hypothesis  $h : \mathcal{X} \mapsto \{-1, +1\}$  in  $H_n$  with small generalization error  $\mathbf{R}(h)$ .
3. **Hypothesis space:**  $H_n = \{x \mapsto \text{sgn}(\langle w, x \rangle + b) \mid w \in \mathbb{R}^n, b \in \mathbb{R}\}$ .
4. A linear classifier is defined as  $h(x) = \text{sgn}(\langle w, x \rangle + b)$ .
5. Vector  $w$  is orthogonal to the separator.



6. We shown that  $VC(H_n) = n + 1$ .
7. We can learn this space using the ERM paradigm, as long as the sample size is  $\frac{(n+1) + \log(1/\delta)}{\epsilon}$ .
8. Implementing the ERM rule in the nonseparable case is known to be **computationally hard**.



1. Linear programs are problems that can be expressed as maximizing a linear function subject to linear inequalities. That is

$$\begin{aligned} & \max_{w \in \mathbb{R}^n} \langle u, w \rangle \\ & \text{subject to } Aw \geq v. \end{aligned}$$

where

- $w \in \mathbb{R}^n$  is the vector of variables we wish to determine.
  - $A$  is an  $m \times n$  matrix.
  - $u \in \mathbb{R}^n$  and  $v \in \mathbb{R}^m$  are vectors.
2. Linear programs can be solved efficiently.



1. Suppose that the training data is linearly separable.
2. We are interested to find  $w$  and  $b$  that results in zero training error.
3. Let  $w = (b, w_1, w_2, \dots, w_n)$  and  $\mathbf{x} = (1, x_1, \dots, x_n)$ .
4. Hence, we are looking for  $w \in \mathbb{R}^{n+1}$  such that for all  $i$

$$\text{sign}(\langle w, \mathbf{x}_i \rangle) = y_i$$

5. Equivalently, we are looking for  $w \in \mathbb{R}^{n+1}$  such that for all  $i$

$$y_i \langle w, \mathbf{x}_i \rangle > 0.$$

6. Let  $w^*$  be a vector that satisfies this condition.
7. Define  $\gamma = \min_i (y_i \langle w^*, \mathbf{x}_i \rangle)$  and let  $\bar{w} = \frac{w^*}{\gamma}$ . Therefore, for all  $i$  we have

$$y_i \langle \bar{w}, \mathbf{x}_i \rangle = \frac{1}{\gamma} y_i \langle w^*, \mathbf{x}_i \rangle \geq 1.$$

8. We have thus shown that there exists a vector that for all  $i$  satisfies

$$y_i \langle w, \mathbf{x}_i \rangle > 1.$$



1. We have thus shown that there exists a vector that for all  $i$  satisfies

$$y_i \langle w, x_i \rangle > 1.$$

2. To find a vector that satisfies the above inequality,
  - Set  $A$  to be  $m \times (n + 1)$  matrix whose rows are the instances multiplied by  $y_i$ :  $A_{ij} = y_i \times x_{ij}$ .
  - Set  $v$  to be  $(1, 1, \dots, 1) \in \mathbb{R}^{n+1}$ .
3. Then the above inequality becomes

$$Aw > v.$$

4. The LP form requires a maximization objective, thus, we set a **dummy** objective,  $u = (0, \dots, 0) \in \mathbb{R}^{n+1}$ .
5. There are other algorithm for finding the linear classifier such as Perceptron.

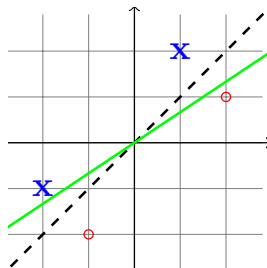


## Support vector machine

---



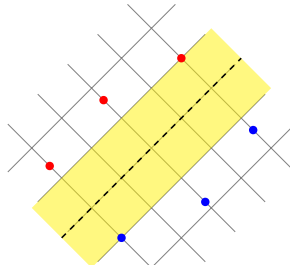
1. Consider the problem of finding a separating hyperplane for a linearly separable dataset  $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$  with  $\mathbf{x}_i \in \mathbb{R}^n$  and  $y_i \in \{-1, +1\}$ .
2. Which of the infinite hyperplanes should we choose?



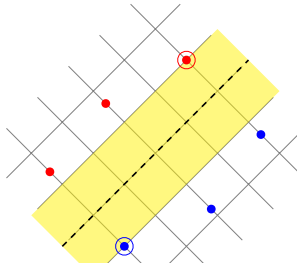
- Hyperplanes that pass too close to the training examples will be sensitive to noise and, therefore, less likely to generalize well for data outside the training set.
  - It is reasonable to expect that a hyperplane that is farthest from all training examples will have better generalization capabilities.
3. We can find the maximum margin linear classifier by first identifying a classifier that correctly classifies all the examples and then increasing the geometric margin until we cannot increase the margin any further.
  4. We can also set up an optimization problem for directly maximizing the geometric margin.



1. What is the margin of a classifier?



2. The goal of SVM is to maximize the margin.



3. The closest examples are called **support vectors**.



1. We will need the classifier to be correct on all the training examples ( $y_k \langle \mathbf{w}, \mathbf{x}_k \rangle \geq \gamma$  for all  $k = 1, 2, \dots, m$ ) subject to these constraints, we would like to maximize the geometric margin ( $\frac{\gamma}{\|\mathbf{w}\|}$ ). Hence, we have (Let  $b = 0$ )

$$\text{Maximize } \frac{\gamma}{\|\mathbf{w}\|} \quad \text{subject to } y_k \langle \mathbf{w}, \mathbf{x}_k \rangle \geq \gamma \text{ for all } k = 1, 2, \dots, m$$

2. We can alternatively minimize the inverse  $\frac{\|\mathbf{w}\|}{\gamma}$  or the inverse squared  $\frac{\|\mathbf{w}\|^2}{\gamma^2}$  subject to the same constraints.

$$\text{Minimize } \frac{1}{2} \frac{\|\mathbf{w}\|^2}{\gamma^2} \quad \text{subject to } y_k \langle \mathbf{w}, \mathbf{x}_k \rangle \geq \gamma \text{ for all } k = 1, 2, \dots, m$$

Factor  $\frac{1}{2}$  is included merely for later convenience.

3. The above problem can be written as

$$\text{Minimize } \frac{1}{2} \left\| \frac{\mathbf{w}}{\gamma} \right\|^2 \quad \text{subject to } y_k \left\langle \left( \frac{\mathbf{w}}{\gamma} \right), \mathbf{x}_k \right\rangle \geq 1 \text{ for all } k = 1, 2, \dots, m$$

4. This problem tells the dependency on the ratio  $\frac{\mathbf{w}}{\gamma}$  not  $w$  or  $\gamma$  separately.
5. Scaling  $\mathbf{w}$  by a constant also doesn't change the decision boundary. We can therefore fix  $\gamma = 1$  and solve for  $\mathbf{w}$ .



1. By fixing  $\gamma = 1$  and solving for  $\mathbf{w}$ , we obtain

$$\text{Minimize } \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{subject to } y_k \langle \mathbf{w}, \mathbf{x}_k \rangle \geq 1 \text{ for all } k = 1, 2, \dots, m$$

2. This optimization problem is in the standard SVM form and is a quadratic programming problem.
3. We will modify the linear classifier here slightly by adding an offset term so that the decision boundary does not have to go through the origin. In other words, the classifier that we consider has the form

$$h(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$$

$\mathbf{w}$  is the weight vector  $b$  is the bias of the separating hyperplane. The hyperplane is shown by  $(\mathbf{w}, b)$ .

4. The bias parameter changes the optimization problem to

$$\text{Minimize } \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{subject to } y_k (\langle \mathbf{w}, \mathbf{x}_k \rangle + b) \geq 1 \text{ for all } k = 1, 2, \dots, m$$

5. Note that the bias only appears in the constraints. This is different from simply modifying the linear classifier through origin by feeding it with examples that have an additional constant component, i.e.,  $\mathbf{x}' = [1; \mathbf{x}]$ .



1. The optimization problem for SVM is defined as

$$\text{Minimize } \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{subject to } y_k (\langle \mathbf{w}, \mathbf{x}_k \rangle + b) \geq 1 \text{ for all } k = 1, 2, \dots, m$$

2. In order to solve this constrained optimization problem, we introduce Lagrange multipliers  $\alpha_k \geq 0$ , with one multiplier  $\alpha_k$  for each of the constraints giving the Lagrangian function

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{k=1}^m \alpha_k [y_k (\langle \mathbf{w}, \mathbf{x}_k \rangle + b) - 1]$$

where  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)^T$ .

3. Note the minus sign in front of the Lagrange multiplier term, because we are minimizing with respect to  $\mathbf{w}$  and  $b$ , and maximizing with respect to  $\alpha$ . **please read Appendix E of Bishop<sup>1</sup>.**
4. Setting the derivatives of  $L(\mathbf{w}, b, \alpha)$  with respect to  $\mathbf{w}$  and  $b$  equal to zero, we obtain the following two equations

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} &= \sum_{k=1}^m \alpha_k y_k \mathbf{x}_k \\ \frac{\partial L}{\partial b} = 0 \Rightarrow 0 &= \sum_{k=1}^m \alpha_k y_k \end{aligned}$$

<sup>1</sup>Christopher M. Bishop (2006). *Pattern Recognition and Machine Learning*. Berlin, Heidelberg: Springer-Verlag.



1.  $L$  has to be minimized with respect to the primal variables  $w$  and  $b$  and maximized with respect to the dual variables  $\alpha_k$ . Eliminating  $\mathbf{w}$  and  $b$  from  $L(\mathbf{w}, b, \alpha)$  using these conditions then gives the dual representation of the problem in which we maximize

$$\psi(\alpha) = \sum_{k=1}^m \alpha_k - \frac{1}{2} \sum_{k=1}^m \sum_{j=1}^m \alpha_k \alpha_j y_k y_j \langle \mathbf{x}_k, \mathbf{x}_j \rangle$$

2. We need to maximize  $\psi(\alpha)$  subject to the following constraints

$$\begin{aligned} \alpha_k &\geq 0 \quad \forall k \\ \sum_{k=1}^m \alpha_k y_k &= 0 \end{aligned}$$

3. The constrained optimization of this form satisfies the Karush-Kuhn-Tucker (KKT) conditions, which in this case require that the following three properties hold

$$\begin{aligned} \alpha_k &\geq 0 \\ y_k g(\mathbf{x}_k) &\geq 1 \\ \alpha_k [y_k g(\mathbf{x}_k) - 1] &= 0 \end{aligned}$$



1. For optimal  $\alpha_k$ 's,

$$\alpha_k [1 - y_k (\langle \mathbf{w}, \mathbf{x}_k \rangle + b)] = 0$$

2.  $\alpha_k$  is **non-zero** only if  $\mathbf{x}_k$  lies on one of the **two margin boundaries**, i.e., for which  $y_k (\langle \mathbf{w}, \mathbf{x}_k \rangle + b) = 1$
3. These examples are called **support vectors**.
4. To classify a data  $\mathbf{x}$  using the trained model, we evaluate the following function

$$h(x) = \text{sgn} \left( \sum_{k=1}^m \alpha_k y_k \langle \mathbf{x}_k, \mathbf{x} \rangle \right)$$





1. Leave-one-out is a method to estimate true error of a classifier and is defined as

**Definition (Leave-one-out error)**

Let  $h$  be the hypothesis output by learning algorithm  $A$  after receiving sample  $S$  of size  $m$ . Then, the leave-one-out error of  $A$  over  $S$  is:

$$\hat{R}_{LOO}(A) = \frac{1}{m} \sum_{i=1}^m \mathbb{I} [h_{S-\{\mathbf{x}_i\}}(\mathbf{x}_i) \neq y_i]$$

2. Thus, for each  $i \in \{1, 2, \dots, m\}$ ,  $A$  is trained on all the points in  $S$  except for  $\mathbf{x}_i$ , and its error is then computed using  $\mathbf{x}_i$ . The leave-one-out error is the average of these errors.
3. In general, computing the leave-one-out error may be **costly** since it requires training  $m$  times on samples of size  $m - 1$ .



Leave-one-out error has the following property.

### Lemma

The average leave-one-out error for samples of size  $m \geq 2$  is an unbiased estimate of the average generalization error for samples of size  $m - 1$ :

$$\mathbb{E}_{S \sim \mathcal{D}^m} [\hat{\mathbf{R}}_{LOO}(A)] = \mathbb{E}_{S' \sim \mathcal{D}^{m-1}} [\mathbf{R}(h_{S'})]$$

where  $\mathcal{D}$  denotes the distribution according to which points are drawn.

### Proof.

By the linearity of expectation, we can write

$$\begin{aligned} \mathbb{E}_{S \sim \mathcal{D}^m} [\hat{\mathbf{R}}_{LOO}(A)] &= \frac{1}{m} \sum_{i=1}^m \mathbb{E}_{S \sim \mathcal{D}^m} [\mathbb{I}[h_{S-\{x_i\}}(\mathbf{x}_i) \neq y_i]] \\ &= \mathbb{E}_{S \sim \mathcal{D}^m} [\mathbb{I}[h_{S-\{x_i\}}(\mathbf{x}_i) \neq y_i]] \\ &= \mathbb{E}_{S' \sim \mathcal{D}^{m-1}, \mathbf{x}_1 \sim \mathcal{D}} [\mathbb{I}[h_{S'}(\mathbf{x}_1) \neq y_1]] \\ &= \mathbb{E}_{S' \sim \mathcal{D}^{m-1}} [\mathbf{R}(h_{S'})]. \end{aligned}$$

□

**Theorem**

Let  $h_S$  be the optimal hyperplane for a sample  $S$  and let  $N_{SV}(S)$  be the number of support vectors defining  $h_S$ . Then,

$$\mathbb{E}_{S \sim \mathcal{D}^m} [\mathbf{R}(h_S)] \leq \mathbb{E}_{S \sim \mathcal{D}^{m+1}} \left[ \frac{N_{SV}(S)}{m+1} \right]$$

**Proof.**

1. Let  $S$  be a linearly separable sample of  $m+1$ .
2. If  $\mathbf{x}$  is not a support vector for  $h_S$ , removing it does not change the SVM solution. Thus,  $h_{S-\{\mathbf{x}\}} = h_S$  and  $h_{S-\{\mathbf{x}\}}$  correctly classifies  $\mathbf{x}$ .
3. By contraposition, if  $h_{S-\{\mathbf{x}\}}$  misclassifies  $\mathbf{x}$ ,  $\mathbf{x}$  must be a support vector, which implies

$$\hat{\mathbf{R}}_{LOO}(SVM) \leq \frac{N_{SV}(S)}{m+1}.$$

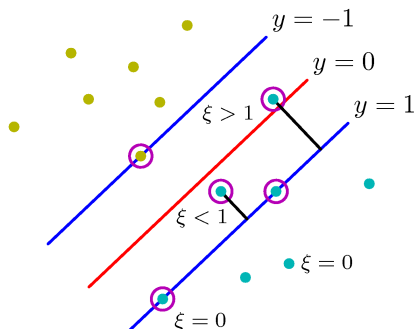
4. Taking the expectation of both sides and using the previous lemma yields the result. □



1. This Theorem gives a sparsity argument in favor of SVMs: **the average error of the algorithm is upper bounded by the average fraction of support vectors.**
2. We hope that for many distributions seen in practice, a relatively small number of the training points will lie on the marginal hyperplanes.
3. The solution will then be sparse in the sense that a small fraction of the dual variables  $\alpha_i$  will be non-zero.
4. This bound is relatively weak since it applies only to the average generalization error of the algorithm over all samples of size  $m$ .
5. It provides no information about the **variance of the generalization error.**



1. We have assumed that the training data are linearly separable in the feature space.
2. The resulting SVM will give exact separation of the training data.
3. In the practice, the class-conditional distributions may overlap, in which the exact separation of the training data can lead to poor generalization.
4. We need a way to modify the SVM so as to allow some training examples to be miss-classified.
5. To do this, we introduce slack variables ( $\xi_k \geq 0$ ) (distance by which it violates the margin); one slack variable for each training example.
6. The slack variables are defined by  $\xi_k = 0$  for examples that are inside the correct boundary margin and  $\xi_k = |y_k - g(x_k)|$  for other examples, where  $g(x) = \langle w, x \rangle + b$ .
7. Thus for data point that is on the decision boundary  $g(x_k) = 0$  will have  $\xi_k = 1$  and the data points with  $\xi_k \geq 1$  will be misclassified.





1. The exact classification constraints will be

$$y_k g(x_k) \geq 1 - \xi_k \quad \text{for } k = 1, 2, \dots, m$$

2. Our goal is now to maximize the margin while softly penalizing points that lie on the wrong side of the margin boundary. We therefore minimize

$$C \sum_{k=1}^m \xi_k + \frac{1}{2} \|w\|^2$$

$C > 0$  controls the trade-off between the slack variable penalty and the margin.

3. We now wish to solve the following optimization problem.

$$\text{Minimize } \frac{1}{2} \|w\|^2 + C \sum_{k=1}^m \xi_k \quad \text{subject to } y_k g(x_k) \geq 1 - \xi_k \text{ for all } k = 1, 2, \dots, m$$

4. The corresponding Lagrangian is given

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + C \sum_{k=1}^m \xi_k - \sum_{k=1}^m \alpha_k [y_k g(x_k) - 1 + \xi_k] - \sum_{k=1}^m \beta_k \xi_k$$

where  $\alpha_k \geq 0$  and  $\beta_k \geq 0$  are Lagrange multipliers.

## Margin theory

---

1. The **geometric margin** of a point for a linear classifier is defined as

### Definition (Geometric margin of a point)

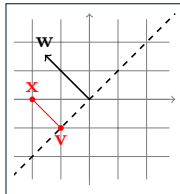
The **geometric margin (margin)** of a point  $\mathbf{x}$  for a linear classifier  $h : \mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle + b$ , denoted by  $\rho(\mathbf{x})$ , is **its distance to the hyperplane**  $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$ ,

$$\rho(\mathbf{x}) = \frac{|\langle \mathbf{w}, \mathbf{x} \rangle + b|}{\|\mathbf{w}\|}$$

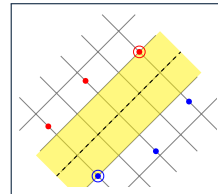
2. The **geometric margin** of a linear classifier is defined as

### Definition (Geometric margin of a classifier)

The **geometric margin (margin)** of a linear classifier  $h : \mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle + b$  for a sample  $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ , denoted by  $\rho_g$ , is **the minimum margin of the points in that sample**:



$$\rho_g = \min_{1 \leq i \leq m} \frac{|\langle \mathbf{w}, \mathbf{x}_i \rangle + b|}{\|\mathbf{w}\|}$$







1. The VC-dimension of the family of hyperplanes or linear hypotheses in  $\mathbb{R}^n$  is  $n + 1$ .
2. This result yields that for any  $\delta > 0$ , with probability at least  $1 - \delta$ , for any  $h \in H$ , we have

$$\mathbf{R}(h) \leq \hat{\mathbf{R}}(h) + \sqrt{\frac{2(n+1) \log\left(\frac{em}{n+1}\right)}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}}$$

3. When the dimension of the feature space  $n$  is large compared to the sample size  $m$  ( $n \gg m$ ), this bound is uninformative.
4. The theorem (**VC-dimension of canonical hyperplanes**) presents a bound on the VC-dimension of canonical hyperplanes.
5. This bound does not depend on the dimension of feature space  $n$ , but only on the margin and the radius  $r$  of the sphere containing the data.



### Lemma (Cauchy-Schwarz inequality)

For all  $x, y \in \mathbb{R}_+^n$ ,

$$|\langle x, y \rangle| \leq \|x\|_2 \|y\|_2$$

with equality iff  $x$  and  $y$  are collinear.

### Lemma (Jensen's inequality)

Let  $x$  be a random variable taking values in a non-empty convex set  $C \subseteq \mathbb{R}^n$  with a finite expectation  $\mathbb{E}[x]$ , and  $f$  a measurable convex function defined over  $C$ . Then,  $\mathbb{E}[x]$  is in  $C$ ,  $\mathbb{E}[f(x)]$  is finite, and we have  $f(\mathbb{E}[x]) \leq \mathbb{E}[f(x)]$ .

### Theorem (VC-dimension of canonical hyperplanes)

Let  $S \subseteq \{x \mid \|x\| \leq r\}$ . Then, the VC-dimension  $d$  of the set of canonical hyperplanes  $\{x \mapsto \text{sgn}(\langle w, x \rangle) \mid \min_{x \in S} |\langle w, x \rangle| = 1 \wedge \|w\| \leq \Lambda\}$  verifies

$$d \leq (r\Lambda)^2$$

**Proof.**

Let  $\{x_1, \dots, x_d\}$  be a set fully shattered. Then, for all  $y \in \{-1, +1\}$ , there exists  $w$  such that for all  $i \in \{1, 2, \dots, m\}$ , we have  $1 \leq \langle w, x_i \rangle$ . Summing up the inequalities gives

$$d \leq \left\langle w, \sum_{i=1}^d y_i x_i \right\rangle \leq \|w\| \left\| \sum_{i=1}^d y_i x_i \right\| \leq \Lambda \left\| \sum_{i=1}^d y_i x_i \right\|. \quad \text{By using Cauchy-Schwarz ineq.}$$

Taking the expectation over  $y \sim U$  (uniform) yields

$$\begin{aligned} d &\leq \Lambda \mathbb{E}_{y \sim U} \left[ \left\| \sum_{i=1}^d y_i x_i \right\| \right] \leq \Lambda \sqrt{\mathbb{E}_{y \sim U} \left[ \left\| \sum_{i=1}^d y_i x_i \right\|^2 \right]} \quad \text{By using Jensen's ineq.} \\ &= \Lambda \sqrt{\sum_{i,j=1}^d \mathbb{E} [y_i y_j] \langle x_i, x_j \rangle} \\ &= \Lambda \sqrt{\sum_{i=1}^d \langle x_i, x_i \rangle} \\ &= \Lambda \sqrt{dr^2} = \Lambda r \sqrt{d} \end{aligned}$$

Thus  $\sqrt{d} \leq \Lambda r$  and  $d \leq (\Lambda r)^2$ . □

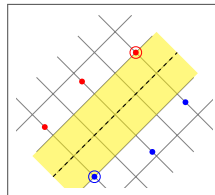
1. In this session, learning guarantees are presented, which are independent of dimension  $n$ .
2. These guarantees are the notion of **confidence margin** and hold for **real-valued functions**.

### Definition (Confidence margin)

The **confidence margin** of a real-valued function  $h$  at  $(x, y) \in \mathcal{X} \times \mathcal{Y}$  is  $\rho_h(x, y) = yh(x)$ .

- When  $yh(x) > 0$ ,  $h$  **classifies  $x$  correctly** but we interpret the magnitude of  $|h(x)|$  as the confidence of the prediction made by  $h$ .
- The relationship with geometric margin for linear functions  $h : x \mapsto \langle w, x \rangle + b$  is

$$|\rho_h(x, y)| \geq \rho_g \|w\|.$$





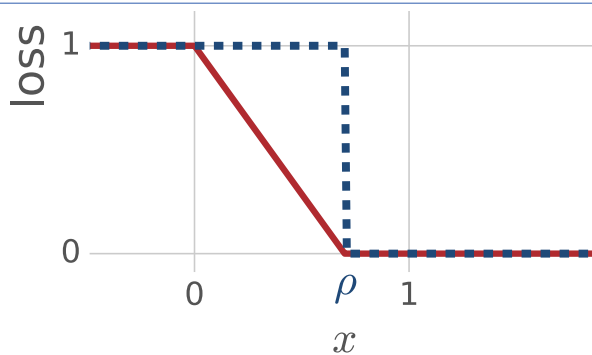
- In view of definition of **confidence margin**, for any parameter  $\rho > 0$ , we will define a  $\rho$ -margin loss that,
  - penalizes  $h$  with the cost of 1 when it misclassifies point  $x$  i.e.  $yh(x) \leq 0$
  - penalizes  $h$  (linearly) when it correctly classifies  $h$  with confidence less than or equal to  $\rho$  i.e.  $yh(x) \leq \rho$ .

### Definition

Margin loss function For any  $\rho > 0$ , the  $\rho$ -margin loss is the function  $L_\rho : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}_+$  defined for all  $y, y' \in \mathbb{R}$  by  $L_\rho(y, y') = \Phi_\rho(yy')$  with,

$$\Phi_\rho(x) = \min \left( 1, \max \left( 0, 1 - \frac{x}{\rho} \right) \right) = \begin{cases} 1 & \text{if } x \leq 0 \\ 1 - \frac{x}{\rho} & \text{if } 0 \leq x \leq \rho \\ 0 & \text{if } \rho \leq x \end{cases}$$

The parameter  $\rho > 0$  can be interpreted as the **confidence margin** demanded from a hypothesis  $h$ .





1. The **empirical margin loss** is similarly defined as the margin loss over the training sample.

### Definition (Empirical margin loss function)

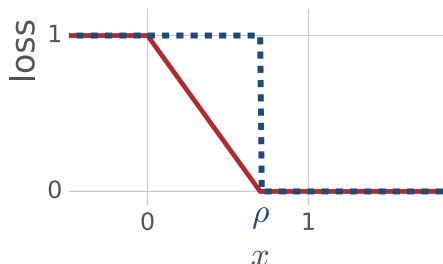
Given a sample  $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$  and a hypothesis  $h$ , the **empirical margin loss** is defined by

$$\hat{R}_\rho(h) = \frac{1}{m} \sum_{i=1}^m \Phi_\rho(y_i h(x_i))$$

2. It is clear that  $\Phi_\rho(x)$  is  $1/\rho$  – Lipschitz.
3. Note that, for any  $i \in \{1, 2, \dots, m\}$ , we have  $\Phi_\rho(y_i h(x_i)) \leq \mathbb{I}[y_i h(x_i) \leq \rho]$  and we have

$$\frac{1}{m} \sum_{i=1}^m \Phi_\rho(y_i h(x_i)) \leq \frac{1}{m} \sum_{i=1}^m \mathbb{I}[y_i h(x_i) \leq \rho]$$

4. In generalization bounds, the empirical margin loss can be replaced by this upper bound.



**Lemma (Talagrand's Lemma)**

Let  $\Phi : \mathbb{R} \mapsto \mathbb{R}$  be an  $L$ -Lipschitz function. Then, for any hypothesis set  $H$  of real-valued functions,

$$\hat{\mathcal{R}}_S(\Phi \circ H) \leq L \times \hat{\mathcal{R}}_S(H)$$

**Proof.**

Please read the proof of Lemma 5.7 of Mehryar Mohri and Afshin Rostamizadeh and Ameet Talwalkar Book<sup>a</sup>. □

---

<sup>a</sup>Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar (2018). *Foundations of Machine Learning*. Second Edition. MIT Press.



The proof of the following Theorem given in class.

### Theorem (Generalization bound based on Rademacher complexity)

Let  $\mathcal{G}$  be a family of functions mapping from  $\mathcal{Z}$  to  $[0, 1]$ . Then, for any  $\delta > 0$ , with probability at least  $1 - \delta$  over the draw of an IID sample  $S$  of size  $m$ , each of the following holds for all  $g \in \mathcal{G}$ :

$$\mathbb{E}[g(z)] \leq \frac{1}{m} \sum_{i=1}^m g(z_i) + 2\mathcal{R}_m(\mathcal{G}) + O\left(\sqrt{\frac{\ln \frac{1}{\delta}}{m}}\right)$$

$$\mathbb{E}[g(z)] \leq \frac{1}{m} \sum_{i=1}^m g(z_i) + 2\hat{\mathcal{R}}_S(\mathcal{G}) + O\left(\sqrt{\frac{\ln \frac{2}{\delta}}{m}}\right)$$

### Theorem (General margin bound for linear classifiers)

Let  $H$  be a set of real-valued functions. Fix  $\rho > 0$ . For any  $\delta > 0$ , with probability at least  $1 - \delta$ , the following hold for all  $h \in H$ :

$$\mathbf{R}(h) \leq \hat{\mathbf{R}}_\rho(h) + \frac{2}{\rho} \mathcal{R}_m(H) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

$$\mathbf{R}(h) \leq \hat{\mathbf{R}}_\rho(h) + \frac{2}{\rho} \hat{\mathcal{R}}_S(H) + 3\sqrt{\frac{\log(1/\delta)}{2m}}$$





### Proof of general margin bound for linear classifiers.

Let  $\tilde{H} = \{z = (x, y) \mapsto yh(x) \mid h \in H\}$ . Consider the family of functions taking values in  $[0, 1]$

$$\tilde{H} = \{\Phi_\rho \circ h \mid h \in H\}.$$

By the theorem given in the previous slide (**generalization bound based on Rademacher complexity**), with probability at least  $1 - \delta$  for all  $g \in \tilde{H}$ , we have

$$\mathbb{E} [g(z)] \leq \frac{1}{m} \sum_{i=1}^m g(z_i) + \frac{2}{\rho} \mathcal{R}_m(H) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

Thus

$$\mathbb{E} [\Phi_\rho(yh(x))] \leq \hat{\mathbf{R}}_\rho(h) + \frac{2}{\rho} \mathcal{R}_m(\tilde{H}) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

Since  $\Phi_\rho$  is  $\frac{1}{\rho}$ -Lipschitz by **Talagrand's lemma**, Then

$$\mathcal{R}_m(\tilde{H}) \leq \frac{1}{\rho} \mathcal{R}_m(H) = \frac{1}{\rho m} \mathbb{E}_{\sigma, S} \left[ \sup_{h \in H} \sum_{i=1}^m \sigma_i h(x_i) \right] = \frac{1}{\rho} \mathcal{R}_m(H)$$

Since  $\mathbb{I}[yh(x) < 0] \leq \Phi_\rho(yh(x))$ , this shows the first statement, and similarly the second one.  $\square$



1. This generalization bounds  $\left( \mathbf{R}(h) \leq \hat{\mathbf{R}}_\rho(h) + \frac{2}{\rho} \mathcal{R}_m(H) + \sqrt{\frac{\log(1/\delta)}{2m}} \right)$  suggest a trade-off: a larger value of  $\rho$  decreases the complexity term, but tends to increase the empirical margin-loss  $\hat{\mathbf{R}}_\rho(h)$  by requiring from a hypothesis  $h$  a higher confidence margin.
2. If for a relatively large value of  $\rho$  the empirical margin loss of  $h$  remains relatively small, then  $h$  benefits from a very favorable guarantee on its generalization error.
3. For the above theorem, the margin parameter  $\rho$  must be selected beforehand. But, the bounds of the theorem can be generalized to hold uniformly for all  $\rho \in (0, 1]$  at the cost of a modest additional term  $\sqrt{\frac{\log \log(2/\rho)}{m}}$  as shown in the following Theorem.

### Theorem

Let  $H$  be a set of real-valued functions. Fix  $r > 0$ . Then, for any  $\delta > 0$ , with probability at least  $1 - \delta$ , each of the following hold for all  $h \in H$  and  $\rho \in (0, r]$

$$\mathbf{R}(h) \leq \hat{\mathbf{R}}_\rho(h) + \frac{4}{\rho} \mathcal{R}_m(H) + \sqrt{\frac{\log \log(2r/\rho)}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}}$$

$$\mathbf{R}(h) \leq \hat{\mathbf{R}}_\rho(h) + \frac{4}{\rho} \hat{\mathcal{R}}_S(H) + \sqrt{\frac{\log \log(2r/\rho)}{m}} + 3\sqrt{\frac{\log(1/\delta)}{2m}}$$


**Theorem (Rademacher complexity of linear hypotheses)**

Let  $S \subseteq \{x \mid \|x\| \leq r\}$  be a sample of size  $m$  and let  $H = \{x \mapsto \langle w, x \rangle \mid \|w\| \leq \Lambda\}$ . Then

$$\hat{\mathcal{R}}_S(H) \leq \sqrt{\frac{r^2 \Lambda^2}{m}}$$

**Proof.**

$$\begin{aligned} \hat{\mathcal{R}}_S(H) &= \frac{1}{m} \mathbb{E}_\sigma \left[ \sup_{\|w\| \leq \Lambda} \sum_{i=1}^m \sigma_i \langle w, x_i \rangle \right] = \frac{1}{m} \mathbb{E}_\sigma \left[ \sup_{\|w\| \leq \Lambda} \left\langle w, \sum_{i=1}^m \sigma_i x_i \right\rangle \right] \\ &\leq \frac{\Lambda}{m} \mathbb{E}_\sigma \left[ \left\| \sum_{i=1}^m \sigma_i x_i \right\| \right] \leq \frac{\Lambda}{m} \sqrt{\mathbb{E}_\sigma \left[ \left\| \sum_{i=1}^m \sigma_i x_i \right\|^2 \right]} = \frac{\Lambda}{m} \sqrt{\mathbb{E}_\sigma \left[ \sum_{j=1}^m \sigma_j \sigma_j \langle x_i, x_j \rangle \right]} \\ &\leq \frac{\Lambda}{m} \sqrt{\sum_{i=1}^m \|x_i\|^2} \\ &\leq \frac{\Lambda \sqrt{mr^2}}{m} = \sqrt{\frac{r^2 \Lambda^2}{m}} \end{aligned}$$

□



### Corollary (Margin bound (linear classifier))

Let  $\rho > 0$  and  $H = \{x \mapsto \langle w, x \rangle \mid \|w\| \leq \Lambda\}$ . Let also  $S \subseteq \{x \mid \|x\| \leq r\}$  be a sample of size  $m$ . Then for any  $\delta > 0$  with probability at least  $1 - \delta$  for any  $h \in H$ ,

$$R(h) \leq \hat{R}_\rho(h) + 2\sqrt{\frac{r^2 \Lambda^2 / \rho^2}{m}} + 3\sqrt{\frac{\log(2/\delta)}{2m}}$$

### Proof.

Follows directly general margin bound and bound on  $\hat{R}_S(H)$  for linear classifiers. □

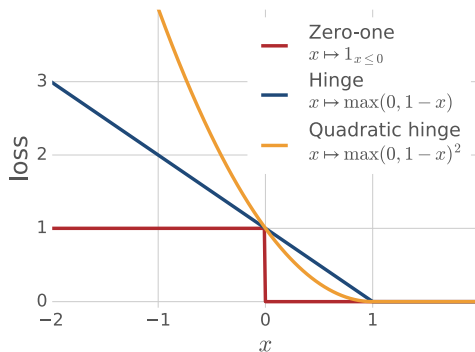
1. This bound for linear hypotheses is **remarkable**, it does not depend directly on  $n$ , but only on  $\rho$ .
2.  $R(h)$  can be small when  $\frac{\rho}{r\Lambda}$  is large; while  $\hat{R}_\rho(h)$  is relatively small.
3.  $\hat{R}_\rho(h)$  is small when few points are either classified incorrectly or correctly, but with **margin**  $\leq \rho$ .
4. When  $S$  is **linearly separable**, for a linear hypothesis with geometric margin  $\rho_g$  and the choice of the confidence margin parameter  $\rho = \rho_g$ ,  $\hat{R}_\rho(h)$  is zero.
5. Thus, if  $\rho_g$  is **relatively large**, this provides a strong guarantee for the generalization error of the corresponding linear hypothesis.

## Summary

---

1. Generalization bound does not depend on the dimension but on the margin.
2. This suggests seeking a large-margin separating hyperplane in a higher-dimensional feature space.
3. Taking dot products in a high-dimensional feature space can be very costly.
4. For any  $\rho > 0$ , the  $\rho$ -margin loss function is upper bounded by the  $\rho$ -hinge loss.

$$\Phi_{\rho}(x) = \min \left( 1, \max \left( 0, 1 - \frac{x}{\rho} \right) \right) \leq \max \left( 0, 1 - \frac{x}{\rho} \right)$$



5. The bounds given in these slides can be extended for other loss functions such as [hinge loss function](#). Hence,

$$\mathbf{R}(h) \leq \frac{1}{m} \sum_{i=1}^m \max \left( 0, 1 - \frac{y_i \langle \mathbf{w}, \mathbf{x}_i \rangle}{\rho} \right) + \frac{4}{\rho} \sqrt{\frac{r^2 / \rho^2}{m}} + \sqrt{\frac{\log \log(2r/\rho)}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}}$$



1. Since for any  $\rho > 0$ ,  $h/\rho$  admits the same generalization error as  $h$  with probability of at least  $(1 - \delta)$ , then the following holds for all  $h \in \{x \mapsto \langle w, x \rangle \mid \|w\| \leq 1/\rho\}$  and for all  $\rho > 0$ :

$$\mathbf{R}(h) \leq \frac{1}{m} \sum_{i=1}^m \max(0, 1 - y_i \langle w, x_i \rangle) + 4\sqrt{\frac{r^2/\rho^2}{m}} + \sqrt{\frac{\log \log(2r/\rho)}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}}$$

2. This inequality can be used to derive an algorithm that selects  $w$  and  $\rho > 0$  to minimize the right-hand side.
3. The minimization with respect to  $\rho > 0$  does not lead to a convex optimization, which may not be optimal.
4. Thus, instead,  $\rho > 0$  is left as a free parameter of the algorithm, typically determined via **cross-validation**.



1. Only the first term of the right-hand side depends on  $\mathbf{w}$  for any  $\rho > 0$ , Thus we have:

$$\min_{\|\mathbf{w}\|^2 \leq \frac{1}{\rho^2}} \frac{1}{m} \sum_{i=1}^m \max(0, 1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle)$$

2. Introducing a Lagrange variable  $\lambda \geq 0$ , we obtain

$$\min_{\mathbf{w}} \lambda \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{i=1}^m \max(0, 1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle)$$



3. For any  $\rho > 0$ , there exists an equivalent dual variable  $\lambda \geq 0$  that achieves the same optimal  $\mathbf{w}$ ,  $\lambda$  can be freely selected via **cross-validation**.
4. The resulting algorithm precisely coincides with SVMs using an alternative objective function.
5. The advantage of the **hinge loss is that it is convex**, while the **margin loss is not**.
6. This bound suggests seeking a large-margin hyperplane in a higher-dimensional feature space.
7. Taking dot products in a high-dimensional feature space can be very costly and solution is based on **kernels**.





1. Appendix E of [Christopher M. Bishop Book](#) (Bishop 2006)
2. Section 9.1 and Chapter 15 of [Shai Shalev-Shwartz and Shai Ben-David Book](#) (Shalev-Shwartz and Ben-David 2014)
3. Chapter 4 of [Mehryar Mohri and Afshin Rostamizadeh and Ameet Talwalkar Book](#) (Mohri, Rostamizadeh, and Talwalkar 2018).



-  Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Berlin, Heidelberg: Springer-Verlag.
-  Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar (2018). *Foundations of Machine Learning*. Second Edition. MIT Press.
-  Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

## Lagrangian optimization

---

- Assume that we have a primal optimization problem of the form,

$$\min_x \phi(x) \quad \text{subject to} \quad g_i(x) \geq 0 \quad \text{for } i = 1, 2, \dots, l$$

- Assume that  $\phi$  is **convex** and the constraints  $g_i$  are **linear**.
- We can construct the **Lagrangian optimization problem** as follows,

$$\max_{\alpha} \min_x L(x, \alpha) = \max_{\alpha} \min_x \left( \phi(x) - \sum_{i=1}^l \alpha_i g_i(x) \right)$$

such that

$$\alpha_i \geq 0 \quad \text{for } i = 1, 2, \dots, l$$

- The values  $\alpha_1, \dots, \alpha_l$  are called the **Lagrangian multipliers**.
- We call  $x$  the **primal variable** and  $\alpha$  the **dual variable**.

- We have

$$\max_{\alpha} \min_x L(x, \alpha) = \max_{\alpha} \min_x \left( \phi(x) - \sum_{i=1}^l \alpha_i g_i(x) \right)$$

- Let  $x = x^*$  be an **optimum** then

$$\max_{\alpha} L(x^*, \alpha) = \max_{\alpha} \left( \phi(x^*) - \sum_{i=1}^l \alpha_i g_i(x^*) \right)$$

- Let  $\alpha = \alpha^*$  be an **optimum** then

$$\min_x L(x, \alpha^*) = \min_x \left( \phi(x) - \sum_{i=1}^l \alpha_i^* g_i(x) \right)$$

- This implies that our solutions are **saddle points** on the graph of the function  $L(x, \alpha)$
- An important observation is that at the saddle point the identity

$$\frac{\partial L}{\partial x} = 0$$

- Here, the point  $x^*$  represents an **optimum of  $L$**  with **respect to  $x$** .

## Lagrangian optimization (cont.)

- Let  $\alpha^*$  and  $x^*$  be a solution to the Lagrangian such that,

$$\max_{\alpha} \min_x L(x, \alpha) = L(x^*, \alpha^*) = \phi(x^*) - \sum_{i=1}^l \alpha_i^* g_i(x^*)$$

- Then  $x^*$  is a solution to the primal objective function if and only if the following conditions hold

$$\frac{\partial}{\partial x} L(x^*, \alpha^*) = 0,$$

$$\alpha_i^* g_i(x^*) = 0,$$

$$g_i(x^*) \geq 0,$$

$$\alpha_i^* \geq 0,$$

for  $i = 1, 2, \dots, l$ .

- These conditions are collectively referred to as the **Karush-Kuhn-Tucker (KKT)** conditions and if satisfied ensure that (Why? Please verify it.)

$$L(x^*, \alpha^*) = \phi(x^*)$$

- The KKT conditions are always satisfied for convex optimization problems.**

- Assume that  $x^*$  be an optimum, that is,

$$\frac{\partial}{\partial x} L(x^*, \alpha) = 0,$$

- Then we can rewrite our Lagrangian as an objective function of only the dual variable,

$$L(x^*, \alpha) = \psi(\alpha),$$

the function  $\psi$  the **Lagrangian dual**.

- This gives us our new, **dual optimization problem**

$$\max_{\alpha} \psi(\alpha) \quad \text{subject to} \quad \alpha_i \geq 0 \quad \text{for } i = 1, 2, \dots, l$$

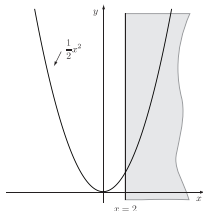
- If the KKT conditions are satisfied

$$\max_{\alpha} \psi(\alpha) = \psi(\alpha^*) = L(\alpha^*, x^*) = \phi(x^*).$$

## Lagrangian optimization (Example)

- Consider the convex optimization problem,

$$\min_x \phi(x) = \min_x \frac{1}{2}x^2 \quad \text{subject to } g(x) = x - 2 \geq 0$$



- The Lagrangian is

$$L(x, \alpha) = \frac{1}{2}x^2 - \alpha(x - 2).$$

- This saddle point occurs where the gradient of the Lagrangian with respect to  $x$  is equal to zero,

$$\frac{\partial L}{\partial x}(\alpha, x^*) = x^* - \alpha = 0$$

- Solving for  $x^*$  gives  $x^* = \alpha$ . Now, substituting  $x^* = \alpha$  into the Lagrangian gives

$$L(\alpha, x^*) = \frac{1}{2}\alpha^2 - \alpha^2 + 2\alpha = 2\alpha - \frac{1}{2}\alpha^2$$



- We can write **dual optimization form** with  $\psi(\alpha) = L(\alpha, x^*)$  as

$$\max_{\alpha} \psi(\alpha) = \max_{\alpha} \left( 2\alpha - \frac{1}{2}\alpha^2 \right) \quad \text{subject to } \alpha \geq 0$$

- Since  $L(x, \alpha)$  is convex, we can write

$$\frac{\partial \psi}{\partial \alpha}(\alpha^*) = 2 - \alpha = 0$$

- This means that  $x^* = \alpha^* = 2$ .

Questions?