

Machine Learning

(Học máy – IT3190E)

Khoat Than

School of Information and Communication Technology
Hanoi University of Science and Technology

2023

Contents

- Introduction to Machine Learning
- **Supervised learning**
 - **Support Vector Machines**
- Unsupervised learning
- Reinforcement learning
- Practical advice

Support Vector Machines (1)

- **Support Vector Machines (SVM)** (máy vector hỗ trợ) was proposed by Vapnik and his colleagues in 1970s. Then it became famous and popular in 1990s.
- Originally, SVM is a method for linear classification. It finds a hyperplane (also called *linear classifier*) to separate the two classes of data.
- For *non-linear classification* for which no hyperplane separates well the data, *kernel functions* (hàm nhân) will be used.
 - Kernel functions transform the data into another space, in which the classes are linearly separable.
- Sometimes, we call linear SVM when no kernel function is used. (in fact, linear SVM uses a linear kernel)

Support Vector Machines (2)

- SVM has a strong theory that supports its performance.
- It can work well with very high dimensional problems.
- It is now one of the most popular and strong methods.
- For text categorization, linear SVM performs very well.

1. SVM: the linearly separable case

■ Problem representation:

- Training data $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_r, y_r)\}$ with r instances.
- Each \mathbf{x}_i is a vector in an n -dimensional space, e.g., $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})^T$. Each dimension represents an attribute.
- Bold characters denote vectors.
- y_i is a class label in $\{-1; 1\}$. '1' is **positive** class, '-1' is **negative** class.

- ## ■ **Linear separability assumption:** there exists a hyperplane (of linear form) that well separates the two classes (giả thuyết tồn tại một siêu phẳng mà phân tách 2 lớp được)

Linear SVM

- SVM finds a hyperplane of the form:

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b$$

[Eq.1]

- \mathbf{w} is the weight vector; b is a real number (bias).
 - $\langle \mathbf{w} \cdot \mathbf{x} \rangle$ and $\langle \mathbf{w}, \mathbf{x} \rangle$ denote the inner product of two vectors (tích vô hướng của hai véctơ)
- Such that for each x_i :

$$y_i = \begin{cases} 1 & \text{if } \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 0 \\ -1 & \text{if } \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b < 0 \end{cases}$$

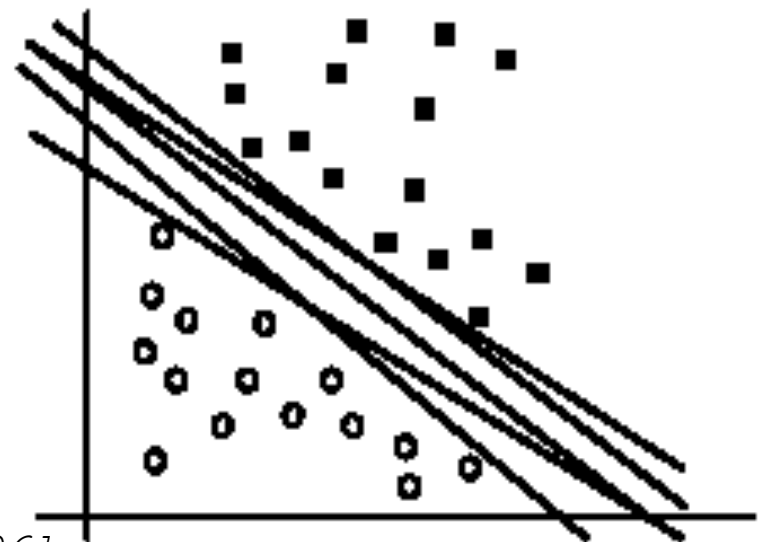
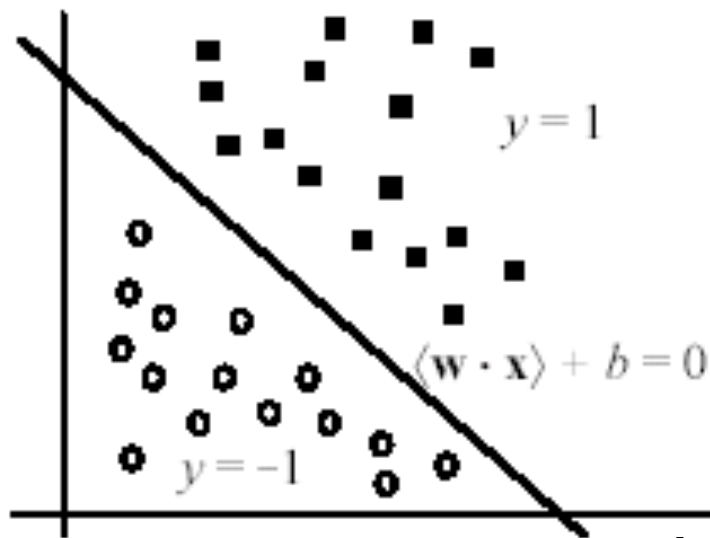
[Eq.2]

Separating hyperplane

- The hyperplane (H_0) which separates the positive from negative class is of the form:

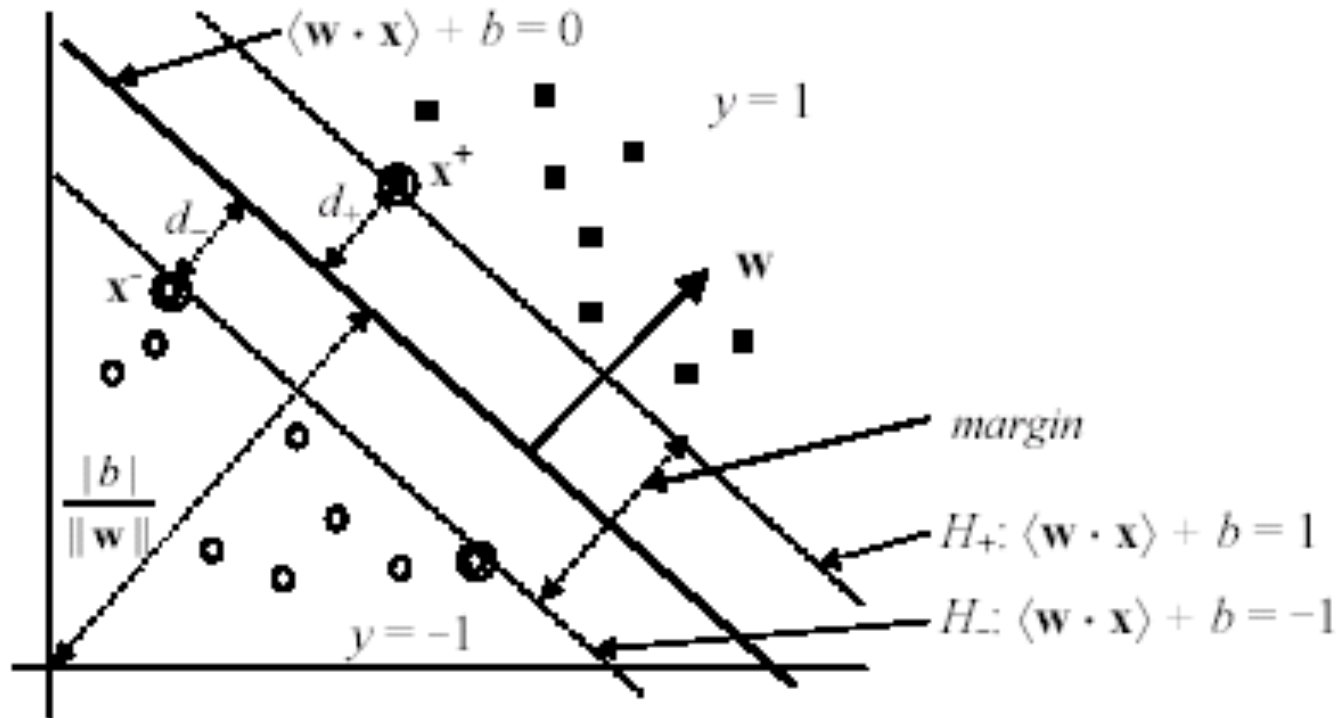
$$\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$$

- It is also known as the *decision boundary/surface*.
- But there might be infinitely many separating hyperplanes.
Which one should we choose?



Hyperplane with max margin

- SVM selects the hyperplane with **max margin**.
(SVM tìm siêu phẳng tách mà có lề lớn nhất)
- It is proven that *the max-margin hyperplane has minimal expected errors among all possible hyperplanes.*



Marginal hyperplanes

- Assume that the two classes in our data can be separated clearly by a hyperplane.
 - Denote $(\mathbf{x}^+, 1)$ in positive class and $(\mathbf{x}^-, -1)$ in negative class which are *closest* to the separating hyperplane H_0 ($\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$)
 - We define two parallel *marginal hyperplanes* as follows:
 - H_+ crosses \mathbf{x}^+ and is parallel with H_0 : $\langle \mathbf{w} \cdot \mathbf{x}^+ \rangle + b = 1$
 - H_- crosses \mathbf{x}^- and is parallel with H_0 : $\langle \mathbf{w} \cdot \mathbf{x}^- \rangle + b = -1$
 - No data point lies between these two marginal hyperplanes.
- And:

$$\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 1, \quad \text{if } y_i = 1$$

$$\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \leq -1, \quad \text{if } y_i = -1$$

[Eq.3]

The margin (1)

- **Margin** (mức lề) is defined as the distance between the two marginal hyperplanes.
 - Denote d_+ the distance from H_0 to H_+ .
 - Denote d_- the distance from H_0 to H_- .
 - $(d_+ + d_-)$ is the margin.
- Remember that the distance from a point \mathbf{x}_i to the hyperplane H_0 ($\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$) is computed as:

$$\frac{|\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b|}{\|\mathbf{w}\|} \quad [\text{Eq.4}]$$

- Where:

$$\|\mathbf{w}\| = \sqrt{\langle \mathbf{w} \cdot \mathbf{w} \rangle} = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2} \quad [\text{Eq.5}]$$

The margin (2)

- So the distance d_+ from \mathbf{x}^+ to H_0 is

$$d_+ = \frac{|\langle \mathbf{w} \cdot \mathbf{x}^+ \rangle + b|}{\|\mathbf{w}\|} = \frac{|1|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|} \quad [\text{Eq.6}]$$

- Similarly, the distance d_- from \mathbf{x}^- to H_0 is

$$d_- = \frac{|\langle \mathbf{w} \cdot \mathbf{x}^- \rangle + b|}{\|\mathbf{w}\|} = \frac{|-1|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|} \quad [\text{Eq.7}]$$

- As a result, the margin is:

$$\text{margin} = d_+ + d_- = \frac{2}{\|\mathbf{w}\|} \quad [\text{Eq.8}]$$

SVM: learning with max margin (1)

- *SVM learns a classifier H_0 with a maximum margin*, i.e., the hyperplane that has the greatest margin among all possible hyperplanes.

- This learning principle can be formulated as the following quadratic optimization problem:

- *Find \mathbf{w} and b that maximize*

$$\text{margin} = \frac{2}{\|\mathbf{w}\|}$$

- *and satisfy the below conditions for any training instance \mathbf{x}_i :*

$$\begin{cases} \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 1, & \text{if } y_i = 1 \\ \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \leq -1, & \text{if } y_i = -1 \end{cases}$$

SVM: learning with max margin (2)

- Learning SVM is equivalent to the following minimization problem:

- Minimize $\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2}$ [Eq.9]

- Conditioned on
$$\begin{cases} \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 1, & \text{if } y_i = 1 \\ \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \leq -1, & \text{if } y_i = -1 \end{cases}$$

- Note, it can be reformulated as:

- Minimize $\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2}$ [Eq.10]

- Conditioned on $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1, \quad \forall i = 1..r$ (P)

- This is a *constrained optimization problem*.

Constrained optimization (1)

- Consider the problem:

Minimize $f(\mathbf{x})$ conditioned on $g(\mathbf{x}) = 0$

- *Necessary condition*: a solution \mathbf{x}_0 will satisfy

$$\begin{cases} \frac{\partial}{\partial \mathbf{x}} (f(\mathbf{x}) + \alpha g(\mathbf{x})) \Big|_{\mathbf{x}=\mathbf{x}_0} = 0 ; \\ g(\mathbf{x}) = 0 \end{cases}$$

□ Where α is a *Lagrange multiplier*.

- In the cases of many constraints ($g_i(\mathbf{x})=0$ for $i=1\dots r$), a solution \mathbf{x}_0 will satisfy:

$$\begin{cases} \frac{\partial}{\partial \mathbf{x}} \left(f(\mathbf{x}) + \sum_{i=1}^r \alpha_i g_i(\mathbf{x}) \right) \Big|_{\mathbf{x}=\mathbf{x}_0} = 0 ; \\ g_i(\mathbf{x}) = 0 \end{cases}$$

Constrained optimization (2)

- Consider the problem with inequality constraints:

Minimize $f(\mathbf{x})$ conditioned on $g_i(\mathbf{x}) \leq 0$

- Necessary condition: a solution \mathbf{x}_0 will satisfy

$$\begin{cases} \left. \frac{\partial}{\partial \mathbf{x}} \left(f(\mathbf{x}) + \sum_{i=1}^r \alpha_i g_i(\mathbf{x}) \right) \right|_{\mathbf{x}=\mathbf{x}_0} = 0; \\ g_i(\mathbf{x}) \leq 0 \end{cases}$$

- Where $\alpha_i \geq 0$ is a Lagrange multiplier.
- $L = f(\mathbf{x}) + \sum_{i=1}^r \alpha_i g_i(\mathbf{x})$ is known as the **Lagrange function**.
 - \mathbf{x} is called *primal variable* (biến gốc)
 - α is called *dual variable* (biến đối ngẫu)

SVM: learning with max margin (3)

- The Lagrange function for problem [Eq. 10] is

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^r \alpha_i [y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1] \quad [\text{Eq.11a}]$$

- Where each $\alpha_i \geq 0$ is a Lagrange multiplier.
- Solving [Eq. 10] is equivalent to the following minimax problem:

$$\begin{aligned} & \min_{\mathbf{w}, b} \max_{\alpha \geq 0} L(\mathbf{w}, b, \alpha) \quad [\text{Eq.11b}] \\ &= \min_{\mathbf{w}, b} \max_{\alpha \geq 0} \left(\frac{1}{2} \langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^r \alpha_i [y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1] \right) \end{aligned}$$

SVM: learning with max margin (4)

- The *primal problem* [Eq. 10] can be derived by solving:

$$\max_{\alpha \geq 0} L(\mathbf{w}, b, \alpha) = \max_{\alpha \geq 0} \left(\frac{1}{2} \langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^r \alpha_i [y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1] \right)$$

- Its *dual problem* (đối ngẫu) can be derived by solving:

$$\min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha) = \min_{\mathbf{w}, b} \left(\frac{1}{2} \langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^r \alpha_i [y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1] \right)$$

- It is known that the optimal solution to [Eq. 10] will satisfy some conditions which is called the **Karush-Kuhn-Tucker** (KKT) conditions.

SVM: Karush-Kuhn-Tucker

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^r \alpha_i y_i \mathbf{x}_i = 0$$

[Eq.12]

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^r \alpha_i y_i = 0$$

[Eq.13]

$$y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 \geq 0, \quad \forall \mathbf{x}_i \quad (i = 1..r)$$

[Eq.14]

$$\alpha_i \geq 0$$

[Eq.15]

$$\alpha_i (y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1) = 0$$

[Eq.16]

- The last equation [Eq. 16] comes from a nice result from the duality theory.
 - Note: any $\alpha_i > 0$ will imply that the associated point \mathbf{x}_i lies in a boundary hyperplane (H_+ or H_-).
 - Such a boundary point is named as a **support vector**.
 - A non-support vector will correspond to $\alpha_i = 0$.

SVM: learning with max margin (5)

- In general, the KKT conditions do not guarantee the optimality of the solution.
- Fortunately, due to the convexity of the primal problem [Eq.10], the *KKT conditions are both necessary and sufficient to assure the global optimality of the solution. It means a vector satisfying all KKT conditions provides the globally optimal classifier.*
 - Convex optimization is ‘easy’ in the sense that we always can find a good solution with a provable guarantee.
 - There are many algorithms in the literature, but most are iterative.
- In fact, problem [Eq.10] is pretty hard to derive an efficient algorithm. Therefore, its **dual problem** is more preferable.

SVM: the dual form (1)

- Remember that the dual counterpart of [Eq.10] is

$$\min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha) = \min_{\mathbf{w}, b} \left(\frac{1}{2} \langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^r \alpha_i [y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1] \right)$$

- By taking the gradient of $L(\mathbf{w}, b, \alpha)$ in variables (\mathbf{w}, b) and zeroing it, we can find the following dual function:

$$L_D(\alpha) = \sum_{i=1}^r \alpha_i - \frac{1}{2} \sum_{i,j=1}^r \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \quad [\text{Eq.17}]$$

SVM: the dual form (2)

- Solving problem [Eq.10] is equivalent to solving its dual problem below:

□ Maximize $L_D(\boldsymbol{\alpha}) = \sum_{i=1}^r \alpha_i - \frac{1}{2} \sum_{i,j=1}^r \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$ [Eq.18]

□ Such that
$$\begin{cases} \sum_{i=1}^r \alpha_i y_i = 0 \\ \alpha_i \geq 0, \forall i = 1..r \end{cases}$$

(D)

- The constraints in **(D)** is much simpler than those of the primal problem. Therefore, deriving an efficient method to solve this problem might be easier.
 - However, existing algorithms for this problem are iterative and complicated. Therefore, we will not discuss any algorithm in detail !

SVM: the optimal classifier

- Once the dual problem is solved for α , we can recover the optimal solution to problem [Eq.10] by using the KKT.

- Let SV be the set of all support vectors

- SV is a subset of the training data.
- $\alpha_i > 0$ suggests that \mathbf{x}_i is a support vector.

- We can compute \mathbf{w}^* by using [Eq.12]. So:

$$\mathbf{w}^* = \sum_{i=1}^r \alpha_i y_i \mathbf{x}_i = \sum_{\mathbf{x}_i \in SV} \alpha_i y_i \mathbf{x}_i; \quad (\text{due to } \alpha_j = 0 \text{ for any } \mathbf{x}_j \text{ not in SV})$$

- To find b^* , we take an index k such that $\alpha_k > 0$:

- It means $y_k(\langle \mathbf{w}^* \cdot \mathbf{x}_k \rangle + b^*) - 1 = 0$ due to [Eq.16].

- Hence,

- $b^* = y_k - \langle \mathbf{w}^* \cdot \mathbf{x}_k \rangle$

SVM: classifying new instances

- The decision boundary is

$$f(\mathbf{x}) = \langle \mathbf{w}^* \cdot \mathbf{x} \rangle + b^* = \sum_{\mathbf{x}_i \in SV} \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b^* = 0 \quad [\text{Eq.19}]$$

- For a new instance \mathbf{z} , we compute:

$$\text{sign}(\langle \mathbf{w}^* \cdot \mathbf{z} \rangle + b^*) = \text{sign} \left(\sum_{\mathbf{x}_i \in SV} \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{z} \rangle + b^* \right) \quad [\text{Eq.20}]$$

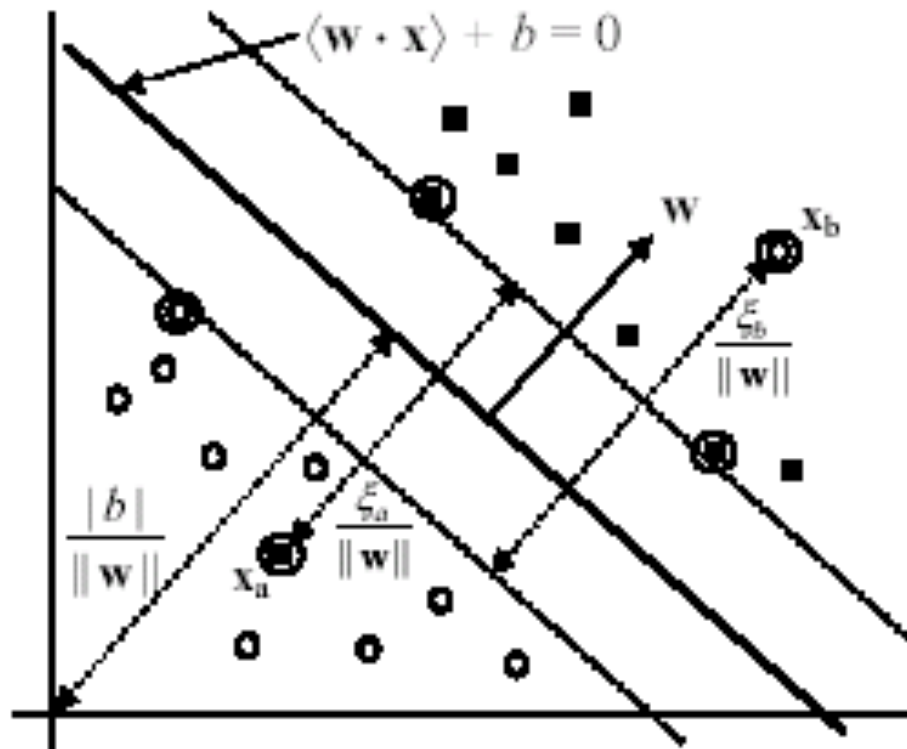
- If the result is 1, \mathbf{z} will be assigned to the positive class; otherwise \mathbf{z} will be assigned to the negative class.
- Note that this classification principle
 - Just depends on the support vectors.
 - Just needs to compute some dot products.

2. Soft-margin SVM

- *What if the two classes are not linearly separable?*
(Trường hợp 2 lớp không thể phân tách tuyến tính thì sao?)
 - Linear separability is ideal in practice.
 - Data are often noisy or erroneous, making two classes overlapping (nhiều/lỗi có thể làm 2 lớp giao nhau)
- In the case of linear separability:
 - Minimize
$$\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2}$$
 - Conditioned on $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1, \quad \forall i = 1..r$
- In the cases of **noises** or **overlapping**, those constraints may never meet simultaneously.
 - It means we cannot solve for \mathbf{w}^* and b^* .

Example of inseparability

- Noisy points \mathbf{x}_a and \mathbf{x}_b are mis-placed.



Relaxing the constraints

- To work with noises/errors, we need to relax the constraints about margin by using some slack variables $\xi_i (\geq 0)$:
(Ta sẽ mở rộng ràng buộc về lề bằng cách thêm biến bù)

$$\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 1 - \xi_i \quad \text{if } y_i = 1$$

$$\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \leq -1 + \xi_i \quad \text{if } y_i = -1$$

- For a noisy/erroneous point \mathbf{x}_i , we have: $\xi_i > 1$
- Otherwise $\xi_i = 0$.
- Therefore, we have the following constraints for the cases of linear inseparability:

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \quad \text{for all } i = 1 \dots r$$

$$\xi_i \geq 0 \quad \text{for all } i = 1 \dots r$$

Penalty on noises/errors

- We should enclose some information on noises/errors into the objective function when learning
(ta nên đính thêm thông tin về nhiễu/lỗi vào hàm mục tiêu)

- Otherwise, the resulting classifier easily underfits the data.

- A penalty term will be used so that learning is to minimize

$$\frac{\langle W, W \rangle}{2} + C \sum_{i=1}^r \xi_i^k$$

- Where $C (>0)$ is the penalty constant (hằng số phạt).
 - The greater C , the heavier the penalty on noises/errors.
- $k = 1$ is often used in practice, due to simplicity for solving the optimization problem.

The new optimization problem

■ *Minimize*
$$\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2} + C \sum_{i=1}^r \xi_i$$
 [Eq.21]

□ *Conditioned on*
$$\begin{cases} y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, & \forall i = 1..r \\ \xi_i \geq 0, & \forall i = 1..r \end{cases}$$

■ This problem is called **Soft-margin SVM**.

■ It is equivalent to minimize the following function

$$\left[\frac{1}{r} \sum_{i=1}^r \max(0, 1 - y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b)) \right] + \lambda \|\mathbf{w}\|_2^2$$

□ $\max(0, 1 - y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b))$ is called *Hinge loss*

□ Some popular losses: squared error, cross entropy, hinge

□ $\lambda > 0$ is a constant

The new optimization problem

- Its Lagrange function is

$$L = \frac{1}{2} \langle \mathbf{w} \cdot \mathbf{w} \rangle + C \sum_{i=1}^r \xi_i - \sum_{i=1}^r \alpha_i [y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 + \xi_i] - \sum_{i=1}^r \mu_i \xi_i$$

[Eq.22]

- Where $\alpha_i (\geq 0)$ and $\mu_i (\geq 0)$ are Lagrange multipliers.

Karush-Kuhn-Tucker conditions (1)

$$\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^r \alpha_i y_i \mathbf{x}_i = 0 \quad [\text{Eq.23}]$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^r \alpha_i y_i = 0 \quad [\text{Eq.24}]$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0, \quad \forall i = 1..r \quad [\text{Eq.25}]$$

Karush-Kuhn-Tucker conditions (2)

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 + \xi_i \geq 0, \quad \forall i = 1..r \quad [\text{Eq.26}]$$

$$\xi_i \geq 0 \quad [\text{Eq.27}]$$

$$\alpha_i \geq 0 \quad [\text{Eq.28}]$$

$$\mu_i \geq 0 \quad [\text{Eq.29}]$$

$$\alpha_i(y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 + \xi_i) = 0 \quad [\text{Eq.30}]$$

$$\mu_i \xi_i = 0 \quad [\text{Eq.31}]$$

The dual problem

- Maximize
$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^r \alpha_i - \frac{1}{2} \sum_{i,j=1}^r \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$
- Such that
$$\begin{cases} \sum_{i=1}^r \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \quad \forall i = 1..r \end{cases}$$
 [Eq.32]
- Note that neither ξ nor μ_i appears in the dual problem.
- This problem is almost similar with that [Eq.18] in the case of linearly separable classification.
- The only difference is the constraint: $\alpha_i \leq C$

Soft-margin SVM: the optimal classifier

- Once the dual problem is solved for α , we can recover the optimal solution to problem [Eq.21].

- Let SV be the set of all *support/noisy vectors*

- SV is a subset of the training data.
- $\alpha_i > 0$ suggests that \mathbf{x}_i is a support/noisy vector.

- We can compute \mathbf{w}^* by using [Eq.12]. So:

$$\mathbf{w}^* = \sum_{i=1}^r \alpha_i y_i \mathbf{x}_i = \sum_{\mathbf{x}_i \in SV} \alpha_i y_i \mathbf{x}_i; \quad (\text{due to } \alpha_j = 0 \text{ for any } \mathbf{x}_j \text{ not in SV})$$

- To find b^* , we take an index k such that $C > \alpha_k > 0$:

- It means $\xi_k = 0$ due to [Eq.25] and [Eq.31];
- And $y_k(\langle \mathbf{w}^* \cdot \mathbf{x}_k \rangle + b^*) - 1 = 0$ due to [Eq.30].
- Hence, $b^* = y_k - \langle \mathbf{w}^* \cdot \mathbf{x}_k \rangle$

Some notes

- From [Eq.25-31] we conclude that

If $\alpha_i = 0$	then $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1,$	and $\xi_i = 0$
If $0 < \alpha_i < C$	then $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) = 1,$	and $\xi_i = 0$
If $\alpha_i = C$	then $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) < 1,$	and $\xi_i > 0$

- The classifier can be expressed as a *linear combination* of few training points.
 - Most training points lie outside the margin area: $\alpha_i = 0$
 - The support vectors lie in the marginal hyperplanes: $0 < \alpha_i < C$
 - The noisy/erronous points will associate with $\alpha_i = C$
- Hence the optimal classifier is a *very sparse combination* of the training data.

Soft-margin SVM: classifying new instances

- The decision boundary is

$$f(\mathbf{x}) = \langle \mathbf{w}^* \cdot \mathbf{x} \rangle + b^* = \sum_{\mathbf{x}_i \in SV} \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b^* = 0 \quad [\text{Eq.19}]$$

- For a new instance \mathbf{z} , we compute:

$$\text{sign}(\langle \mathbf{w}^* \cdot \mathbf{z} \rangle + b^*) = \text{sign} \left(\sum_{\mathbf{x}_i \in SV} \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{z} \rangle + b^* \right) \quad [\text{Eq.20}]$$

- If the result is 1, \mathbf{z} will be assigned to the positive class; otherwise \mathbf{z} will be assigned to the negative class.

- *Note: it is important to choose a good value of C , since it significantly affects performance of SVM.*

- We often use a validation set to choose a value for C .

Linear SVM: summary

- Classification is based on a separating hyperplane.
- Such a hyperplane is represented as a combination of some support vectors.
- The determination of support vectors reduces to solve a quadratic programming problem.
- In the dual problem and the separating hyperplane, dot products can be used in place of the original training data.
 - This is the door for us to learn a nonlinear classifier.

3. Non-linear SVM

- Consider the case in which our data are not linearly separable

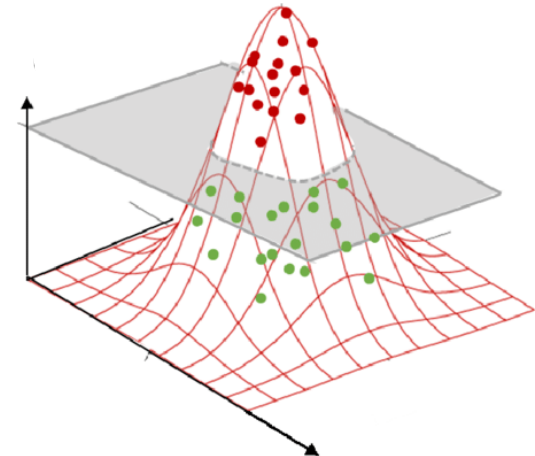
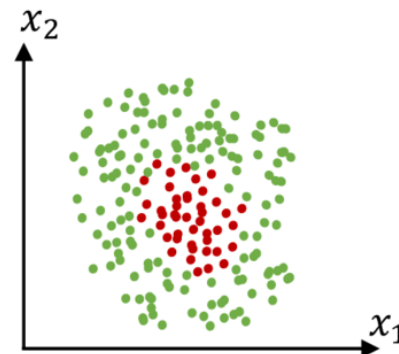
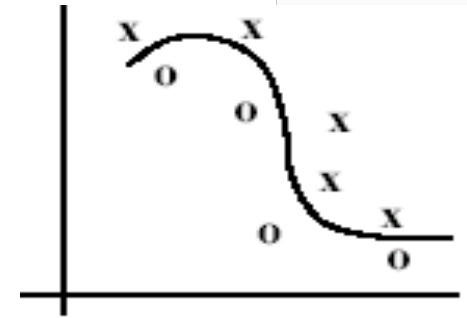
- This may often happen in practice

- How about using a non-linear function?

- **Idea of Non-linear SVM:**

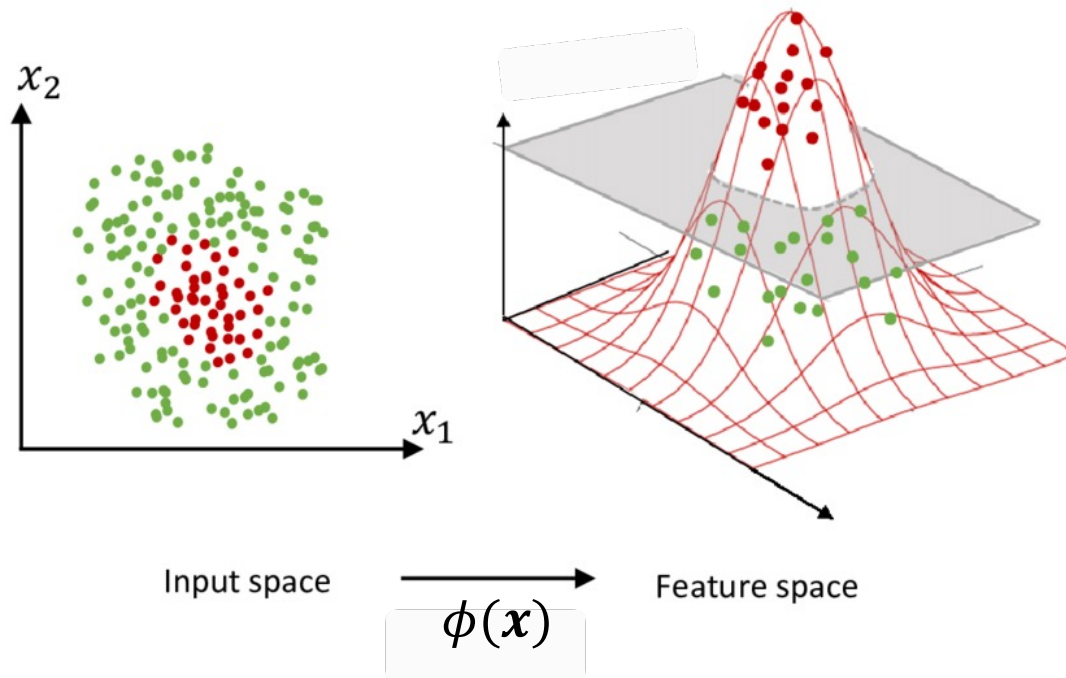
- **Step 1:** transform the input into another space, which often has *higher dimensions*, so that the projection of data is linearly separable

- **Step 2:** use linear SVM in the new space



Non-linear SVM

- **Input space:** initial representation of data
- **Feature space:** the new space after the transformation



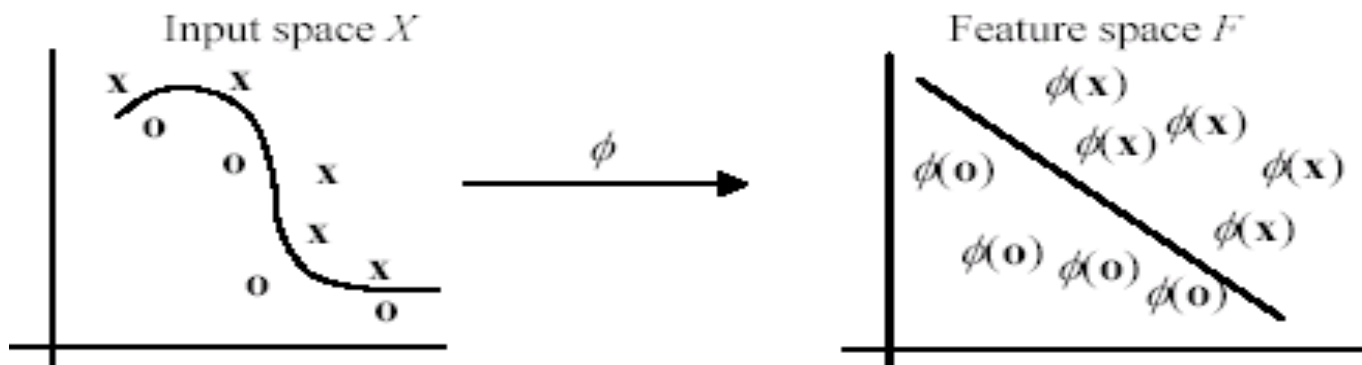
Non-linear SVM: transformation

- Our idea is to map the input \mathbf{x} to a new representation, using a non-linear mapping

$$\begin{aligned}\phi: X &\rightarrow F \\ \mathbf{x} &\mapsto \phi(\mathbf{x})\end{aligned}$$

- In the feature space, the original training data $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_r, y_r)\}$ are represented by

$$\{(\phi(\mathbf{x}_1), y_1), (\phi(\mathbf{x}_2), y_2), \dots, (\phi(\mathbf{x}_r), y_r)\}$$



Non-linear SVM: transformation

- Consider the input space to be 2-dimensional, and we choose the following map

$$\begin{aligned}\phi: X &\rightarrow F \\ (x_1, x_2) &\mapsto (x_1^2, x_2^2, \sqrt{2}x_1x_2)\end{aligned}$$

- So instance $\mathbf{x} = (2, 3)$ will have the representation in the feature space as

$$\phi(\mathbf{x}) = (4, 9, 8.49)$$

Non-linear SVM: learning & prediction

■ Training problem:

Minimize

$$L_P = \frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2} + C \sum_{i=1}^r \xi_i$$

[Eq.34]

Such that

$$\begin{cases} y_i (\langle \mathbf{w} \cdot \phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i, \quad \forall i = 1..r \\ \xi_i \geq 0, \quad \forall i = 1..r \end{cases}$$

■ The dual problem:

Maximize

$$L_D = \sum_{i=1}^r \alpha_i - \frac{1}{2} \sum_{i,j=1}^r \alpha_i \alpha_j y_i y_j \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \rangle$$

[Eq.35]

Such that

$$\begin{cases} \sum_{i=1}^r \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \quad \forall i = 1..r \end{cases}$$

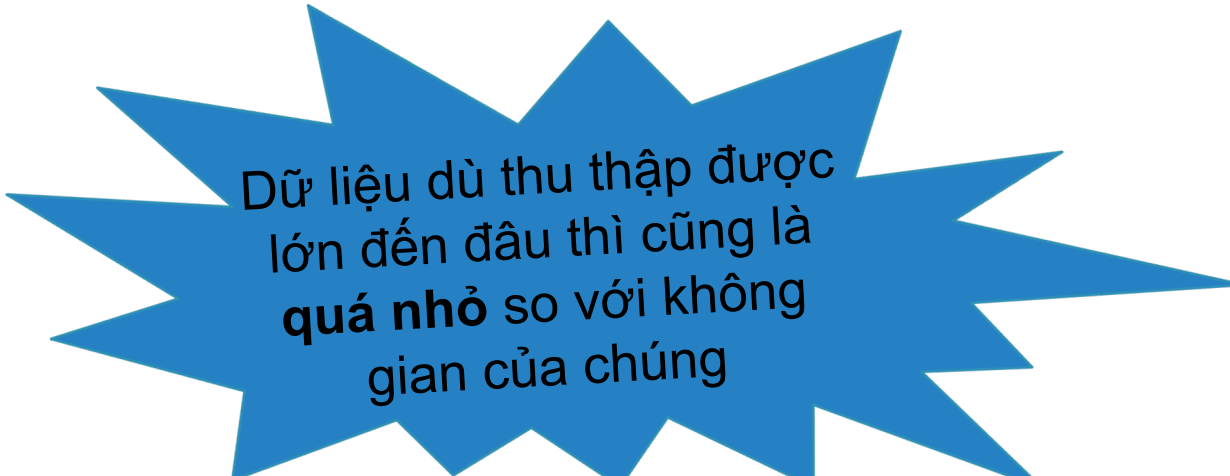
■ Classifier:

$$f(\mathbf{z}) = \langle \mathbf{w}^*, \phi(\mathbf{z}) \rangle + b^* = \sum_{\mathbf{x}_i \in SV} \alpha_i y_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{z}) \rangle + b^*$$

[Eq.36]

Non-linear SVM: difficulties

- How to find the mapping?
 - An intractable problem
- The curse of dimensionality
 - As the dimensionality increases, the volume of the space increases so fast that the available data become sparse.
 - This sparsity is problematic.
 - Increasing the dimensionality will require **exponentially** more training data.



Dữ liệu dù thu thập được
lớn đến đâu thì cũng là
quá nhỏ so với không
gian của chúng

Non-linear SVM: Kernel functions

- An explicit form of a transformation is not necessary

- **The dual problem:**

Maximize
$$L_D = \sum_{i=1}^r \alpha_i - \frac{1}{2} \sum_{i,j=1}^r \alpha_i \alpha_j y_i y_j \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \rangle$$

Such that
$$\begin{cases} \sum_{i=1}^r \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \quad \forall i = 1..r \end{cases}$$

- **Classifier:** $f(\mathbf{z}) = \langle \mathbf{w}^*, \phi(\mathbf{z}) \rangle + b^* = \sum_{\mathbf{x}_i \in SV} \alpha_i y_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{z}) \rangle + b^*$
- Both require only the inner product $\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$
- **Kernel trick:** Nonlinear SVM can be used by replacing those inner products by evaluations of some **kernel function**

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle \quad [\text{Eq.37}]$$

Kernel functions: example

- Polynomial

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^d$$

- Consider the polynomial with degree $d=2$. For any vectors $\mathbf{x}=(x_1, x_2)$ and $\mathbf{z}=(z_1, z_2)$

$$\begin{aligned} \langle \mathbf{x}, \mathbf{z} \rangle^2 &= (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\ &= \langle (x_1^2, x_2^2, \sqrt{2}x_1 x_2), (z_1^2, z_2^2, \sqrt{2}z_1 z_2) \rangle \\ &= \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = K(\mathbf{x}, \mathbf{z}) \end{aligned}$$

- Where $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$.
- Therefore the polynomial is the product of two vectors $\phi(\mathbf{x})$ and $\phi(\mathbf{z})$.

Kernel functions: popular choices

- Polynomial

$$K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x} \cdot \mathbf{z} \rangle + \theta)^d; \text{ trong đó: } \theta \in R, d \in N$$

- Gaussian radial basis function (RBF)

$$K(\mathbf{x}, \mathbf{z}) = e^{-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma}}; \text{ trong đó: } \sigma > 0$$

- Sigmoid

$$K(\mathbf{x}, \mathbf{z}) = \tanh(\gamma \langle \mathbf{x} \cdot \mathbf{z} \rangle + r), \text{ trong đó } \gamma, r \in R$$

- What conditions ensure a kernel function?

Mercer's theorem

SVM: summary

- SVM works with real-value attributes
 - Any nominal attribute need to be transformed into a real one
- The learning formulation of SVM focuses on 2 classes
 - How about a classification problem with > 2 classes?
 - One-vs-the-rest, one-vs-one: a multiclass problem can be solved by reducing to many different problems with 2 classes
- The decision function is simple, but may be hard to interpret
 - It is more serious if we use some kernel functions

SVM: some packages

■ LibSVM:

- <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

■ Linear SVM for large datasets:

- <http://www.csie.ntu.edu.tw/~cjlin/liblinear/>
- http://www.cs.cornell.edu/people/tj/svm_light/svm_perf.html

■ Scikit-learn in python:

- <http://scikit-learn.org/stable/modules/svm.html>

■ SVM^{light}:

- http://www.cs.cornell.edu/people/tj/svm_light/index.html

References

- B. Liu. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer, 2006.
- C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2): 121-167, 1998.
- Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." *Machine learning* 20.3 (1995): 273-297.

Exercises

- What is the main difference between SVM and KNN?
- How many support vectors are there in the worst case? Why?
- The meaning of the constant C in SVM? Compare the role of C in SVM with that of λ in Ridge regression.