SUPPORT VECTOR MACHINES

INTRODUCTION

- **Support Vector Machines (SVM)** (máy vectơ hỗ trợ) was proposed by Vapnik and his colleages 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 play the role to transform the data into another space, in which the data is linearly separable.

- Sometimes, we call linear SVM when no kernel function is used. (in fact, linear SVM uses a linear kernel)

INTRODUCTION

- 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.

CONTENTS AT A GLANCE

01

SVM: The linearly separable case

SVM: the linearly separable case

Problem representation

- Training data $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x_r}, y_r)\}$ with r instances
- Each $\mathbf{x}_i$ is a vector in an n-dimensional space,

  e.g., $\boldsymbol{x_i} = (x_{i1}, x_{i2}, \dots, x_{in})^\top$. Each dimension represents an attribute.
- Bold characters denote vectors.
- $\boldsymbol{y_i}$ 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*

LINEAR SVM

01

SVM finds a hyperplane of the form:

f(x) = <w .x> + b

- w is the weight of vector; b is a real number (bias)
- <w.x> and <w,x> denote the inner product of two vectors

02

Such that for each x_i

$y_i = 1\ if < w.x_i > + \mathrm{b} >= 0$

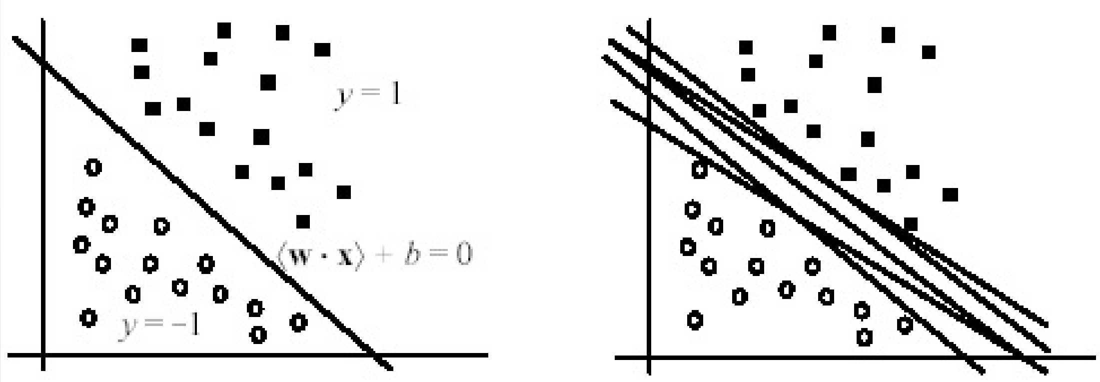
$y_i = -1\ if < w.x_i > + \mathrm{b} >= 0$

SEPERATING HYPERPLANE

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

$$<w . x> + b = 0$$

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



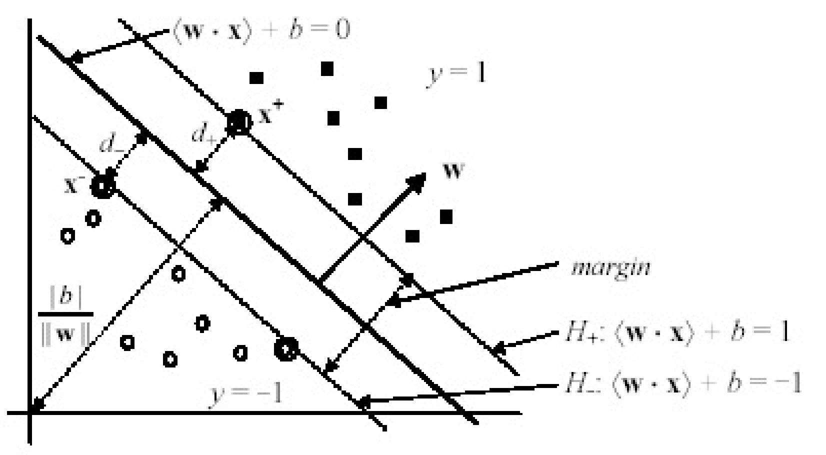
HYPERPLANE & MAX MARGIN OF HYPERPLANE

02

It is proven that *the max-margin hyperplane has minimal errors among all possible hyperplanes.*

01

SVM selects the hyperplane with **max margin**.



HYPERPLANE & MAX MARGIN OF HYPERPLANE

01

Assume that the two classes in our data can be separated clearly by a hyperplane.

02

Denote (x^+, 1) in possitive class and (x^-, -1) in negative class which are *closest* to the separating hyperplane H_0

(<w . x> + b = 0)

03

We define two parallel *marginal hyperplanes* as follows:

- H_+ crosses x^+ and is parallel with H_0: (<w . x^+> + b = 1
- H_- crosses x^- and is parallel with H_0: (<w . x^-> + b = -1

No data point lies between these two marginal hyperplanes. And satisfying:

<w . x_i> + b $\geq$ 1, if y_i = 1

<w . x_i> + b $\leq$-1, if y_i = -1

THE MARGIN

01

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.

02

Remember that the distance from a point x_i to the hyperplane H_0 (<**w** × **x**> + *b* = 0) is computed as:

$$\frac{\langle w \cdot x_i \rangle + b}{\|w\|}$$

Where:

$$\|w\| = \sqrt{\langle w \cdot w \rangle} = \sqrt{w_1^2 + w_{2+\dots}^2 + w_n^2}$$

THE MARGIN

01

So the distance d_+ from x^+ to H_0 is

$$d_+ = \frac{|\langle w \cdot x^+ \rangle + b|}{\|w\|} = \frac{|1|}{\|w\|} = \frac{1}{\|w\|}$$

02

So the distance d_- from x^- to H_0 is

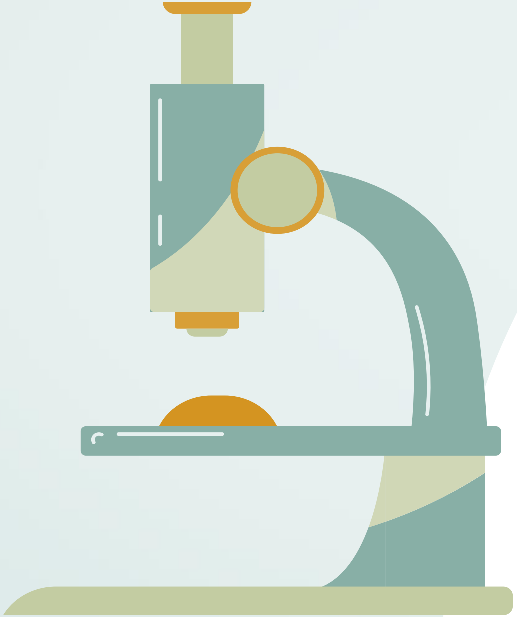
$$d_- = \frac{|\langle w \cdot x^- \rangle + b|}{\|w\|} = \frac{|-1|}{\|w\|} = \frac{1}{\|w\|}$$

03

As a result, the margin is:

$$\text{margin} = d_+ + d_- = \frac{2}{\|w\|}$$

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.

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

   - *Find **w** and b that maximize*

2

SVM: learning with max margin

Learning SVM is equivalent to the following minimization problem:

- Minimize: $\frac{\langle w \cdot w \rangle}{2}$

- Conditioned on:

  $<w \,.\, x_i> + b \geq 1$, if $y_i = 1$

  $<w \,.\, x_i> + b \leq -1$, if $y_i = -1$

Note, it can be reformulated as:

- Minimize: $\frac{\langle w \cdot w \rangle}{2}$ (*)

- Conditioned on:

  $y_i(<w \,.\, x_i> + b) \geq 1, \forall i = 1..r$

This is a ***constrained optimization*** *problem*.

THE MARGIN

01

So the distance d_+ from x^+ to H_0 is

$$d_+ = \frac{|\langle w \cdot x^+ \rangle + b|}{\|w\|} = \frac{|1|}{\|w\|} = \frac{1}{\|w\|}$$

02

So the distance d_- from x^- to H_0 is

$$d_- = \frac{|\langle w \cdot x^- \rangle + b|}{\|w\|} = \frac{|-1|}{\|w\|} = \frac{1}{\|w\|}$$

03

As a result, the margin is:

$$\text{margin} = d_+ + d_- = \frac{2}{\|w\|}$$

SVM: learning with max margin

01

The Lagrange function for problem (*) is

$$L(w,b,\alpha) = \frac{1}{2}\langle w \cdot w\rangle - \sum_{i=1}^{r} \alpha_i[y_i(\langle w \cdot x_i\rangle + b)$$

Where each $\alpha_i \geq 0$ is a Lagrange multiplier.

02

Solving (*) is equivalent to the following minimax problem:

$$\arg\min_{\mathbf{w},b}\max_{\alpha\geq 0} L(\mathbf{w},b,\alpha)$$

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

SVM: learning with max margin

01

The *primal problem* (*) can be derived by solving:

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

02

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

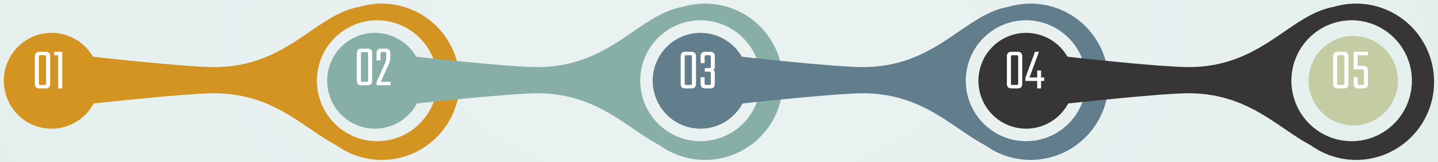
$$\min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha)$$
$$= \min_{\mathbf{w},b}\left(\frac{1}{2}\langle\mathbf{w}\cdot\mathbf{w}\rangle - \sum_{i=1}^{r}\alpha_i[y_i(\langle\mathbf{w}\cdot\mathbf{x}_i\rangle + b) - 1]\right)$$

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

SVM: Karush-Kuhn-Tucker

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

$$\alpha_i \geq 0$$

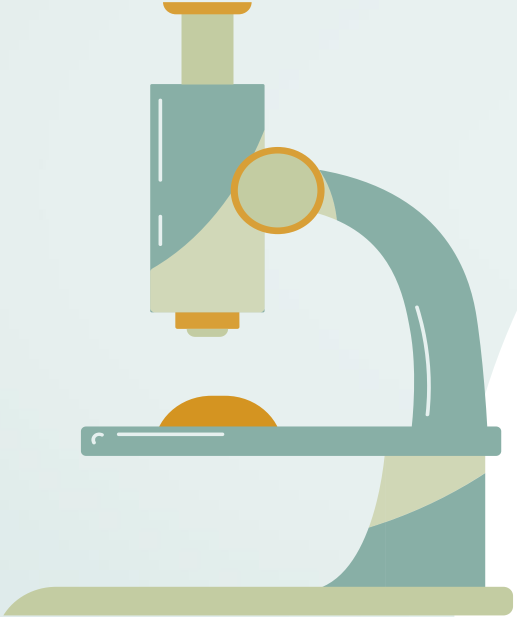


$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{r} \alpha_i y_i \mathbf{x_i} = 0$$

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

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

SVM: Karush-Kuhn-Tucker



- The last equation (5) : $\alpha_i(y_i(\langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b) - 1) = 0$ comes from a nice result from the duality theory.
- Note: any $\alpha_i > 0$ will imply that the associated point 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

- In general, the KKT conditions do not guarantee the optimality of the solution.

- Fortunately, due to the convexity of the primal problem (*), 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 (*) is pretty hard to derive an efficient algorithm. Therefore, its **dual problem** is more preferable.

SVM: the dual form

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(**w**,b,α) in variables (**w**,b) and zeroing it, we can find the following dual function:

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

SVM: the dual form

Solving problem (*) is equivalent to solving its dual problem below:

- Maximize $L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{r} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{r} \alpha_i \alpha_j y_i y_j \langle \mathbf{x_i} \cdot \mathbf{x_j} \rangle$
- Such that $\begin{cases} \sum_{i=1}^{r} \alpha_i y_i = 0 \\ \alpha_i \geq 0, \forall i = 1..r \end{cases}$

The constraints in **(D)** is much more 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 $\boldsymbol{\alpha}$, we can recover the optimal solution to problem (*) by using the KKT.

Let SV be the set of all support vectors

- SV is a subset of the training data.
- α_i> 0 suggests that x_i is a support vector.

We can compute **w*** by using (1). So:

- $\mathbf{w}^* = \sum_{i=1}^{r} \alpha_i y_i \mathbf{x}_i = \sum_{\mathbf{x}_i \in SV} \alpha_i y_i \mathbf{x}_i; \alpha_j = 0\ for\ any\ \mathbf{x_j}\ \boldsymbol{not\ in\ SV})$

To find b*, we take an index k such that α_k> 0:

- It means $y_k(\langle \mathbf{w}^* \cdot \mathbf{x}_k \rangle + b^*) - 1 = 0$ due to (5).
- Hence, $b^* = y_k - \langle \mathbf{w}^* \cdot \mathbf{x}_k \rangle$

SVM: classifying new instance

The decision boundary is

$$f(\mathbf{x}) = \langle \mathbf{w}^* \cdot \mathbf{x} \rangle + b^* = \sum_{\mathbf{x_i} \in SV} \alpha_i y_i \langle \mathbf{x_i} \cdot \mathbf{x} \rangle + b^* = 0$$

For a new instance **z**, we compute:

$$\text{sign}(\langle \mathbf{w}^* \cdot \mathbf{z} \rangle + b^*) = \text{sign}\left(\sum_{\mathbf{x_i} \in SV} \alpha_i y_i \langle \mathbf{x_i} \cdot \mathbf{z} \rangle + b^* \right)$$

- If the result is 1, **z** will be assigned to the possitive class; otherwise **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.

Soft-margin SVM

02

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 erronous, making two classes overlapping (nhiễu/lỗi có thể làm 2 lớp giao nhau)

01

02

In the case of linear separability:

- *Minimize* $\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2}$
- Conditioned on $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1, \forall i = 1..r$

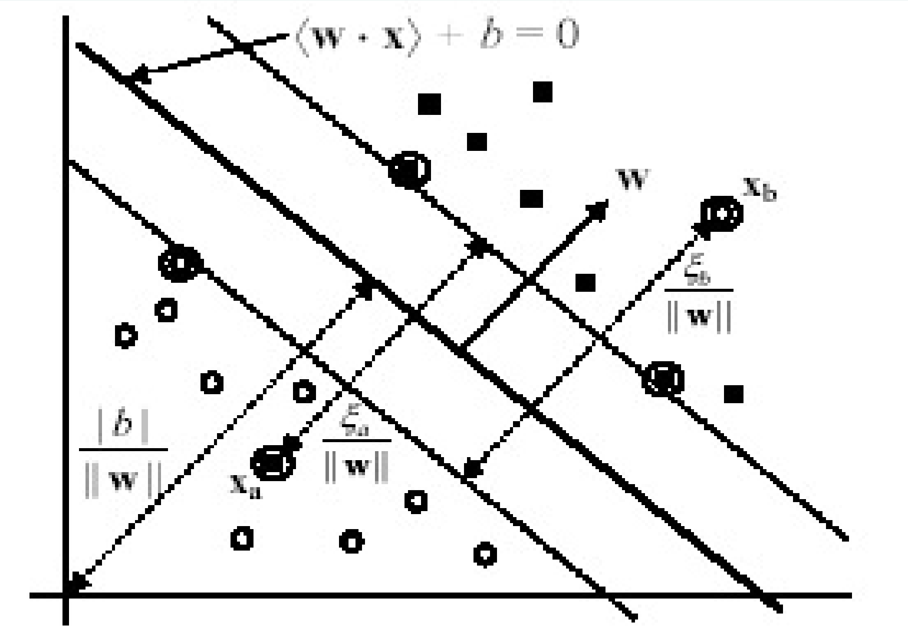
03

In the cases of **noises** or **overlapping**, those constraints may never meet simutaneously.

- It means we cannot solve for **w*** and b*.

Example of inseparability

- Noisy points x_a and x_b are mis-labeled.



Relaxing the constraints

01

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)

$$\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 1 - \xi_i \qquad y_i = 1$$
$$\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \leq -1 + \xi_i \qquad y_i = -1$$

- For a noisy/erronous point ξ_i, we have: ξ_i >1
- Otherwise ξ_i = 0.

02

Therefore, we have the following conditions for the cases of nonlinear separability:

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \qquad for\ all\ i = 1 \ldots r$$
$$\xi_i \geq 0 \qquad for\ all\ i = 1 \ldots r$$

Penalty of noises/errors

01

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 overfits the data.

02

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

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

Where C (>0) is the penalty constant (hằng số phạt). The greater C, the heavier the penalty on noises/errors.

03

k = 1 is often used in practice, due to simplicity for solving the optimization problem.

The new optimization problem

- Minimize $\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2} + C\sum_{i=1}^{r} \xi_i \quad \binom{*}{*}$

  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

$$\left[\frac{1}{r}\sum_{i=1}^{r} max\left(0,1 - y_i(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b)\right)\right] + \lambda \parallel \boldsymbol{w} \parallel_2^2$$

$max\left(0,1 - y_i(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b)\right)$ 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

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

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

Karush-Kuhn-Tucker conditions

$$\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{r} \alpha_i y_i \mathbf{x_i} = 0$$

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

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0, \forall i = 1..r$$

06

07

08

Karush-Kuhn-Tucker conditions

09

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

12

$$\mu_i \geq 0$$

10

$$\xi_i \geq 0$$

13

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

11

$$\alpha_i \geq 0$$

14

$$\mu_i \xi_i = 0$$

The dual problem

$$\text{Maximize} \quad L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{r} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{r} \alpha_i \alpha_j y_i y_j \langle \mathbf{x_i} \cdot \mathbf{x_j} \rangle$$

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

Note that neither ξ_i nor μ_i appears in the dual problem.

This problem is almost similar with that (**) in the case of linearly separable classification.

The only difference is the constraint: $\alpha_i \le C$

Soft-margin SVM: the optimal classifier

Once the dual problem is solved for $\boldsymbol{\alpha}$, we can recover the optimal solution to problem $\binom{*}{*}$

Let SV be the set of all support/noisy vectors

- SV is a subset of the training data.
- $\alpha_i > 0$ suggests that x_i is a support/noisy vector.

We can compute **w*** by using (1). So:

- $\mathbf{w}^* = \sum_{i=1}^{r} \alpha_i y_i \mathbf{x}_i = \sum_{\mathbf{x}_\mathrm{i} \in SV} \alpha_i y_i \mathbf{x}_i$ $(due\ to\ \alpha_i = 0\ for\ any\ \mathbf{x}_\mathrm{i}\ not\ in\ SV\ )$

To find b*, we take an index k such that C > α_k > 0:

- It means $\xi_k = 0$ due to (8) and (14).
- And $y_k(\langle \mathbf{w}^* \cdot \mathbf{x}_k \rangle + b^*) - 1 = 0$ due to (13).
- Hence, $b^* = \frac{1}{y_k} - \langle \mathbf{w}^* \cdot \mathbf{x}_k \rangle$

Some notes

From equations (8) to (14), we conclude that:

$$
\begin{aligned}
&If\ \alpha_i = 0\ then \quad y_i(\langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b) \geq 1 \quad and\ \xi_\mathrm{i} = 0 \\
&If\ 0 < \alpha_\mathrm{i} < C\ then \quad y_i(\langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b) = 1 \quad and\ \xi_\mathrm{i} = 0 \\
&If\ \alpha_\mathrm{i} = C\ then \quad y_i(\langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b) < 1 \quad and\ \xi_\mathrm{i} > 0
\end{aligned}
$$

The classifier can be expressed as a *linear combination* of few training points.

- Most training points lie outside the margin area: $\alpha_\mathrm{i} = 0$
- The support vectors lie in the marginal hyperplanes: $0 < \alpha_\mathrm{i} < C$
- The noisy/erronous points will associate with $\alpha_\mathrm{i} = C$

Hence the optimal classifier is a very sparse *combination* of the training data.

Soft-margin SVM: classifying new instances

The decision boundary is

$$f(\mathbf{x}) = \langle \mathbf{w}^* \cdot \mathbf{x} \rangle + b^* = \sum_{\mathbf{x_i} \in SV} \alpha_i y_i \langle \mathbf{x_i} \cdot \mathbf{x} \rangle + b^* = 0$$

For a new instance **z**, we compute:

$$\text{sign}(\langle \mathbf{w}^* \cdot \mathbf{z} \rangle + b^*) = \text{sign}\left(\sum_{\mathbf{x_i} \in SV} \alpha_i y_i \langle \mathbf{x_i} \cdot \mathbf{z} \rangle + b^* \right)$$

- If the result is 1, **z** will be assigned to the possitive class; otherwise **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.

Non-linear SVM

03

Non-linear SVM: Kernel functions

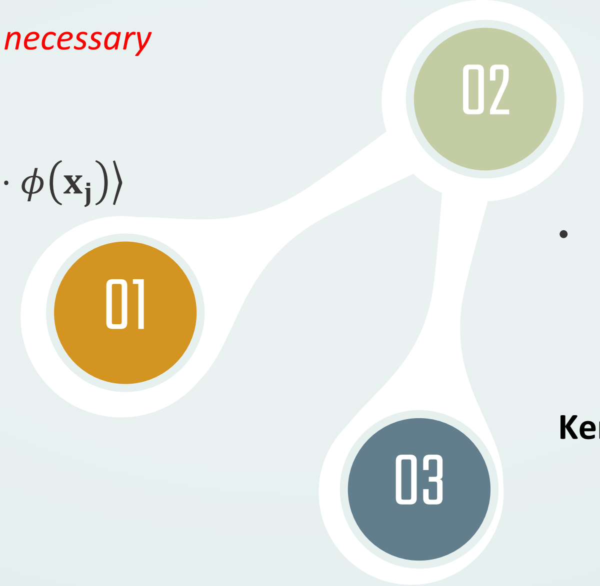
An explicit form of a transformation is not necessary

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



Classifier:

$$f(\mathbf{z}) = \langle \boldsymbol{w}^*, \phi(\mathbf{z}) \rangle + b^*$$

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

- Both require only the inner product

Kernel trick: Nonlinear SVM can be used by replacing those inner products by evaluations of some ***kernel function***

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

Kernel functions: example

01

Polynomial

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

02

Consider the polynomial with degree d=2.

For any vectors $x = (x_1, x_2)\ and\ z = (z_1, z_2)$

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

Where $\phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$

03

Therefore the polynomial is the product of two vectors $\phi(x)\ and\ \phi(z)$

Kernel functions: popular choices

Polynomial

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

$$where : \theta \in R, d \in N$$

Sigmoid

$$K(\mathbf{x}, \mathbf{z}) = \tanh(\beta \langle \mathbf{x} \cdot \mathbf{z} \rangle - \lambda) = \frac{1}{1 + e^{-(\beta \langle \mathbf{x}.\mathbf{z} \rangle - \lambda)}};$$

$$where : \ \beta, \lambda \in R$$

Gaussian radial basis function (RBF)

$$K(\mathbf{x}, \mathbf{z}) = e^{-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma}};$$

$$where : \ \sigma > 0$$

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

THANKS

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution