Elective Course

Course Code: CS4103

Autumn 2025-26

Lecture #44

Artificial Intelligence for Data Science

Week-12:

MACHINE LEARNING (Part XII)

Support Vector Machine (SVM)

Course Instructor:

Dr. Monidipa Das

Assistant Professor

Department of Computational and Data Sciences

Indian Institute of Science Education and Research Kolkata, India 741246

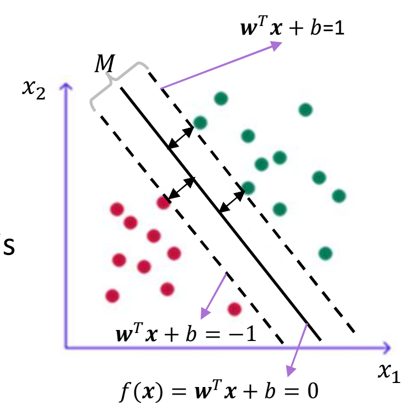
Linear SVM Mathematically

The linearly separable case [Revisited]

2

- Assume that all data is at least distance 1 from the hyperplane,
- Then the following two constraints follow for a training set $\{(\boldsymbol{x_i}, y_i)\}$

$$\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b \geq 1 \quad \text{if } y_i = 1$$
$$\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b \leq -1 \quad \text{if } y_i = -1$$



- For support vectors, the inequality becomes an equality
- Then, since each example's distance from the hyperplane is

$$r = y\frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}$$

- The margin is: $M = \dfrac{2}{\|\mathbf{w}\|}$

Linear SVMs Mathematically (cont.) [Revisited]

- Then we can formulate the *quadratic optimization problem:*

> Find $\mathbf{w}$ and b such that
> $M = \frac{2}{||\boldsymbol{w}||}$ is maximized;
> and for all $\{(\boldsymbol{x_i}, y_i)\}$: $\boldsymbol{w^T x_i} + b \geq 1$ if y_i=1; $\boldsymbol{w^T x_i} + b \leq -1$ if y_i = -1

- A better formulation (min $||\boldsymbol{w}||$ = max $\frac{1}{||w||}$):

> Find $\mathbf{w}$ and b such that
> $\boldsymbol{\Phi}(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$ is minimized;
> and for all $\{(\boldsymbol{x_i}, y_i)\}$: $y_i\,(\boldsymbol{w^T x_i} + b) \geq 1$

Solving the Optimization Problem [Revisited]

> Find $\mathbf{w}$ and b such that
> $\boldsymbol{\Phi}(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w^T w}$ is minimized;
> and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i\,(\mathbf{w^T x_i} + b) \geq 1$

➡

> Find $\mathbf{w}$ and b such that
> $\boldsymbol{\Phi}(\boldsymbol{w}) = \frac{1}{2}||\boldsymbol{w}||^2$ is minimized;
> and for all $\{(\mathbf{x_i}, y_i)\}$: $-y_i\,(\mathbf{w^T x_i} + b) + 1 \leq 0$

- This is now optimizing a *quadratic* function subject to *linear* constraints
- The solution involves ***constructing a dual problem*** where a ***Lagrange multiplier*** α_i is associated with every constraint in the primary problem.
  - **<u>Construct the Lagrangian:</u>** $\mathcal{L}(w, b, \alpha) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{\boldsymbol{i}} \alpha_i [y_i\,(\mathbf{w^T x_i} + b) - 1]$

$$\nabla_w \mathcal{L}(w, b, \alpha) = \boldsymbol{w} - \sum_{\boldsymbol{i}} \alpha_i y_i\, \mathbf{x_i} = \mathbf{0} \Rightarrow \boldsymbol{w} = \sum_{\boldsymbol{i}} \alpha_i y_i\, \mathbf{x_i} \qquad \nabla_b \mathcal{L}(w, b, \alpha) = -\sum_{\boldsymbol{i}} \alpha_i y_i = \mathbf{0} \Rightarrow \sum_{\boldsymbol{i}} \alpha_i y_i = \mathbf{0}$$

Solving the Optimization Problem [Revisited]

5

– **Plugging back w and b values obtained and simplifying:**

$$\mathcal{L}(w,b,\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\ \alpha_j y_i\ y_j {x_i}^T x_j - b\sum_i \alpha_i y_i$$

$$\Rightarrow\ \mathcal{L}(w,b,\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\ \alpha_j y_i\ y_j {x_i}^T x_j$$

- ***Dual optimization problem:***

Why do we use the dual formulation?

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathrm{Q}(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} y_i\ y_j\, \alpha_i\ \alpha_j {x_i}^T x_j$ *is maximized* and
s.t. $\alpha_i \geq 0$ for all α_i
$\sum_i \alpha_i y_i = 0$

The Optimization Problem Solution [Revisited]

6

- Solving the optimization problem involved computing the inner products $\mathbf{x_i}^T\mathbf{x_j}$ between all pairs of training points.
- The solution has the form:

$$\boldsymbol{w} = \Sigma\alpha_i y_i \boldsymbol{x_i} \qquad b = y_k - \mathbf{w}^T\mathbf{x_k} \quad \text{for any } \mathbf{x_k} \text{ such that } \alpha_k \neq 0$$

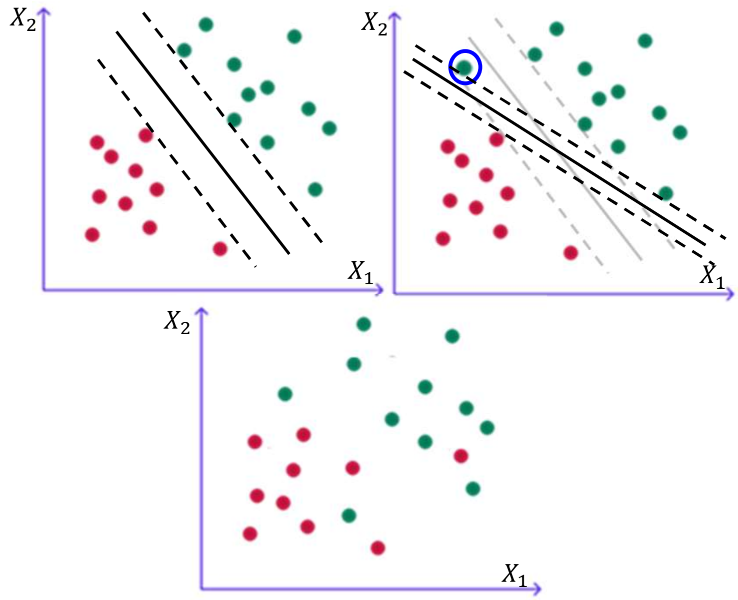
- Each non-zero α_i indicates that corresponding $\boldsymbol{x}_i$ is a support vector.
- Then the classifying function will have the form:

$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x_i}^T\mathbf{x} + b$$

- It relies on an *inner product* between the test point $\mathbf{x}$ and the support vectors $\mathbf{x_i}$

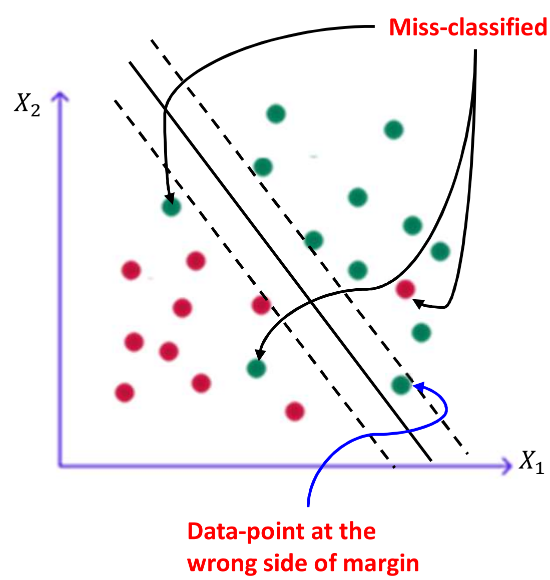
Maximal Margin Classifier: Limitations
[Revisited]

- Although the maximal margin classifier is often successful, it can also lead to overfitting when n is large.
- In many cases ***no separating hyperplane exists***, hence, no maximal margin classifier
- Remedy: Idea of soft margin

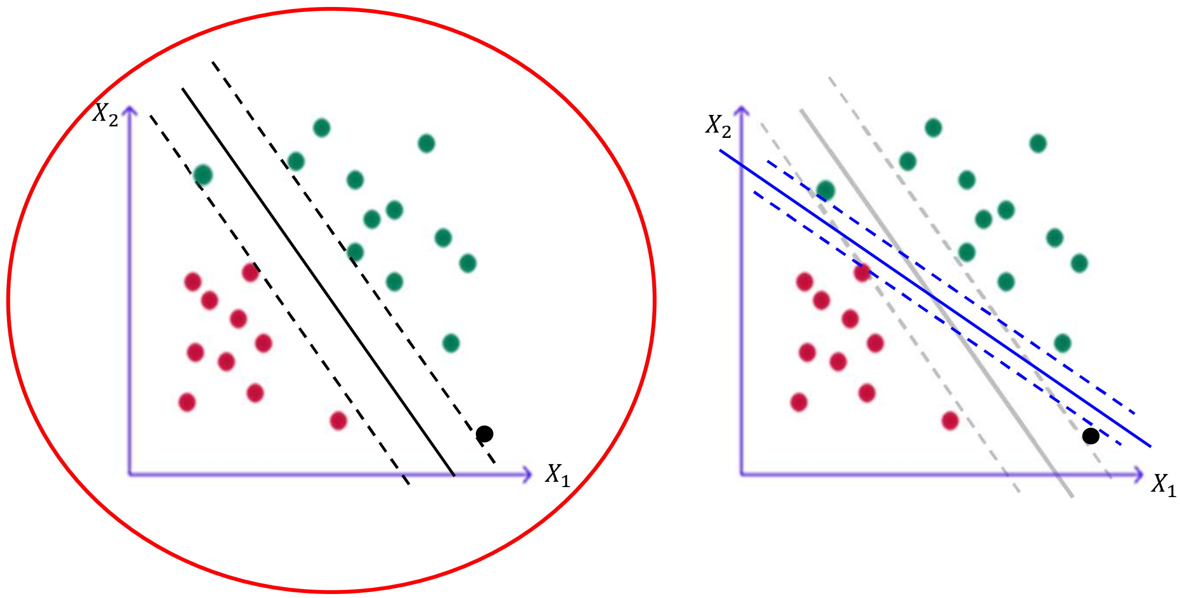


Soft Margin

- Allowing some data points to be misclassified or to violate the margin
  - Greater robustness to individual observations
  - Better classification of most of the training observations



Which one is more robust?



Support Vector Classifier

- The generalization of the maximal margin classifier to the non-separable case
- Soft margin classifier
  - The hyperplane is chosen to correctly separate most of the training observations into the two classes, but may misclassify a few observations

Soft Margin Classification Mathematically

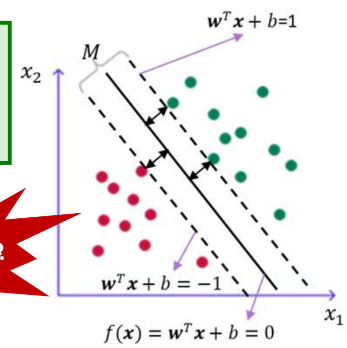
11

- The old formulation:

> Find $\mathbf{w}$ and b such that
> $\boldsymbol{\Phi}(\boldsymbol{w}) = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{w}$ is minimized and for all $\{(\boldsymbol{x_i}, y_i)\}$ $\quad y_i\,(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x_i} + b) \geq 1$

- The new formulation incorporating **slack variables** ξ_i:

> Find $\mathbf{w}$ and b such that
> $\boldsymbol{\Phi}(\boldsymbol{w}) = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{w} + C\Sigma\xi_i$ is minimized and
> for all $\{(\boldsymbol{x_i}, y_i)\}$ $\quad y_i\,(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x_i} + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

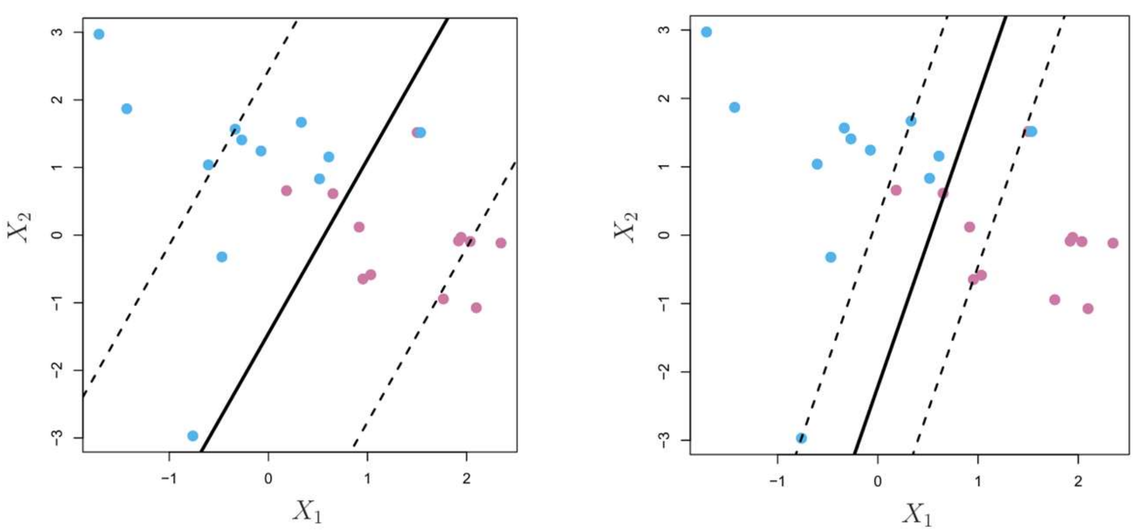


- Parameter C can be viewed as
  - a way to control weighting between the twin goals
  - a way to control overfitting
  - a regularization term

What would be the Lagrangian?

Soft Margin Classification: Impact of C

12



Soft Margin Classification – Solution

13

- The dual problem for soft margin classification:

How?

> Find $\alpha_1 \dots \alpha_N$ such that
> $\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j}$ is maximized and
> (1) $\Sigma\alpha_i y_i = 0$
> (2) $0 \leq \alpha_i \leq C$ for all α_i

- Neither slack variables ξ_i nor their Lagrange multipliers appear in the dual problem!
- Again, $\boldsymbol{x_i}$ with non-zero α_i will be support vectors.
- Solution to the dual problem is:

> $\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i}$
> $b = y_k(1 - \xi_k) - \mathbf{w}^{\mathbf{T}}\mathbf{x}_k$ where $k = \underset{k'}{\text{argmax}}\ \alpha_{k'}$

w is not needed explicitly for classification!

> $f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x_i}^{\mathbf{T}}\mathbf{x} + b$

Classification with SVMs

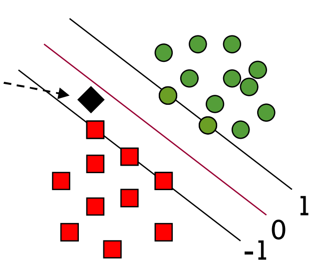
14

- Given a new point $\boldsymbol{x}$, we can score its projection onto the hyperplane normal:
  - I.e., compute score: $\boldsymbol{w}^T\boldsymbol{x} + b = \Sigma\alpha_i y_i \boldsymbol{x_i}^T\boldsymbol{x} + b$
    - Decide class based on whether < or > 0
  - Can set confidence threshold t.

Score > t: yes

Score < $-t$: no

Else: don't know



Linear SVMs: Summary

15

- The classifier is a *separating hyperplane.*
- The most "important" training points are the support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points $\mathbf{x}_i$ are support vectors with non-zero Lagrange multipliers α_i.
- Both in the dual formulation of the problem and in the solution, training points appear only inside ***inner products***:

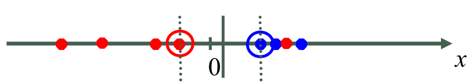
Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \boxed{\mathbf{x_i}^T\mathbf{x_j}}$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

$f(\mathbf{x}) = \Sigma\alpha_i y_i \boxed{\mathbf{x_i}^T\mathbf{x}} + b$

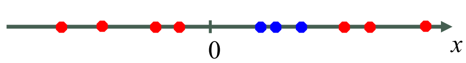
Non-linear SVMs

16

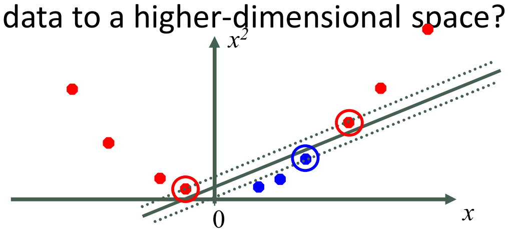
- Datasets that are linearly separable (with some noise) work out great:



- But if the dataset is just too hard?



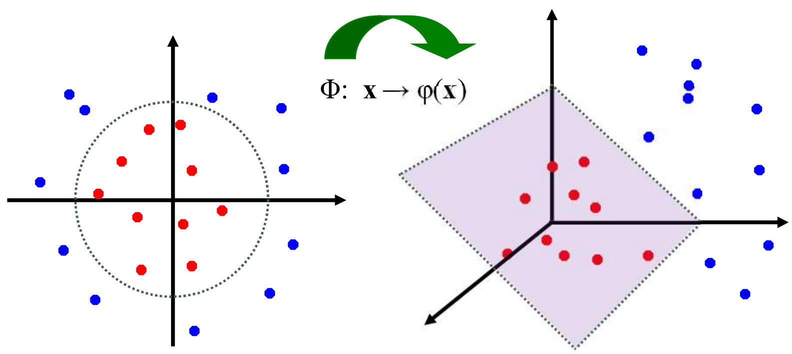
- What if we map data to a higher-dimensional space?:

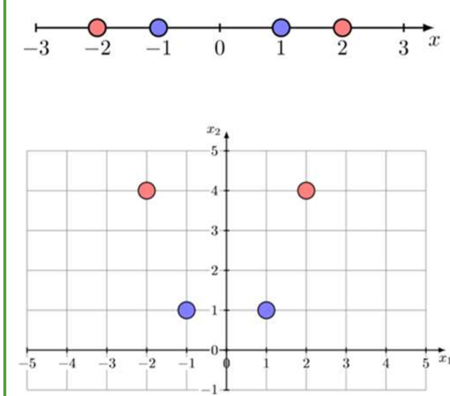


Non-linear SVMs: Feature spaces

17

- **General idea:** the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:





The "Kernel Trick"

18

- The linear classifier relies on an inner product between vectors $\boldsymbol{x_i}^T\boldsymbol{x_j}$
- If every datapoint is mapped into high-dimensional space via some transformation Φ: $\boldsymbol{x} \rightarrow \varphi(\boldsymbol{x})$, the inner product becomes:

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \varphi(\boldsymbol{x_i})^T\varphi(\boldsymbol{x_j})$$

- A *kernel function* is some function that corresponds to an inner product in some expanded feature space.

- **Example:**

2-dimensional vectors $\boldsymbol{x} = [x_1 \;\; x_2]$; let $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (1 + \boldsymbol{x_i}^T\boldsymbol{x_j})^2$,

Need to show that $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \varphi(\boldsymbol{x_i})^T\varphi(\boldsymbol{x_j})$:

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (1 + \boldsymbol{x_i}^T\boldsymbol{x_j})^2 = 1 + x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2} + x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} =$$

$$= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^T [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}]$$

$$= \varphi(\boldsymbol{x_i})^T\varphi(\boldsymbol{x_j}) \quad \textit{where} \quad \varphi(\boldsymbol{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$$

Kernels

19

- Why to use kernels?
  - Map data into better representational space
  - Make non-separable problem separable.

- Common kernels
  - Linear Kernel $K(x_i, x_j) = {x_i}^T x_j$
  - Polynomial Kernel $K(x_i, x_j) = (1 + {x_i}^T x_j)^d$
  - Radial basis function (RBF) Kernel / Gaussian Kernel (infinite dimensional space)

$$K(x_i, x_j) = e^{-\gamma ||x_i - x_j||^2} \qquad K(\mathbf{x}_i, \mathbf{x}_j) = e^{-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2}$$

20

Questions?