Machine Learning Principles

Class11 : October 13

Support Vector Machines and Kernels I

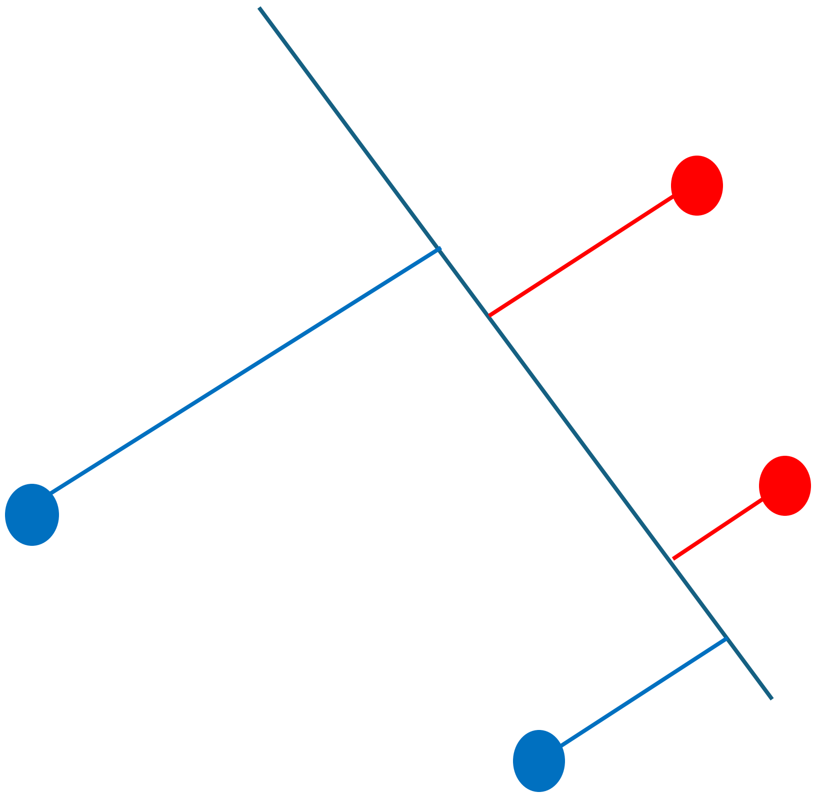
Instructor: Diana Kim

Today's Lecture

- Maximum Margin Classifier
  - Objective Function: Maximum Margin Classification
  - **S**upport **V**ector **M**achine (SVM) : hard margin SVM
  - Kernel Functions (Polynomial and Gaussian Kernel) and Kernel Tricks
  - SVM using Gaussian Kernels

**Concept of Margin

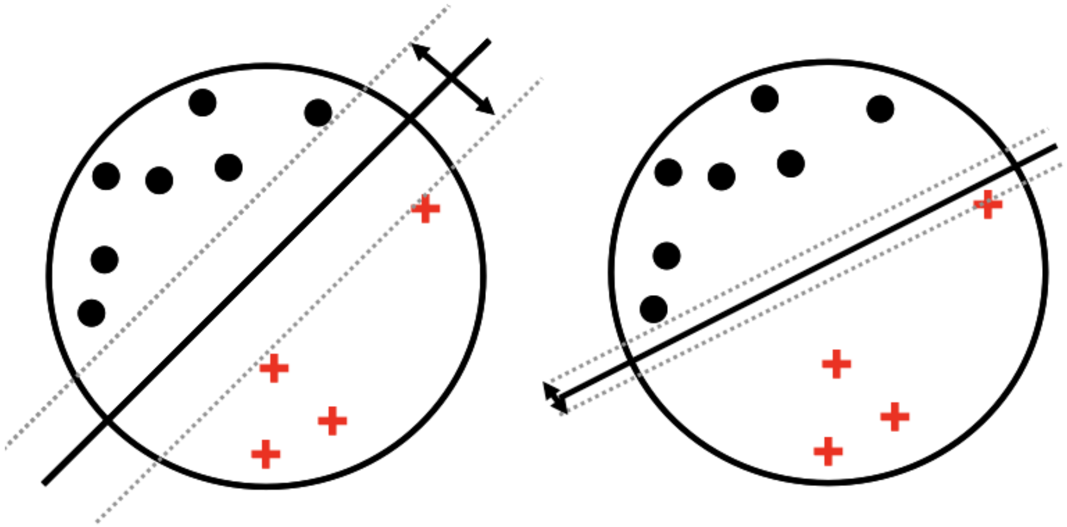
- margin: the smallest distance
  between the decision boundary and any of the samples



▪ Why do we need a maximum margin classifier?

In logistic regression, we pointed out that it promotes a large margin
but does not directly a maximum margin.
why margin matters?

[1] Why a large margin matter?

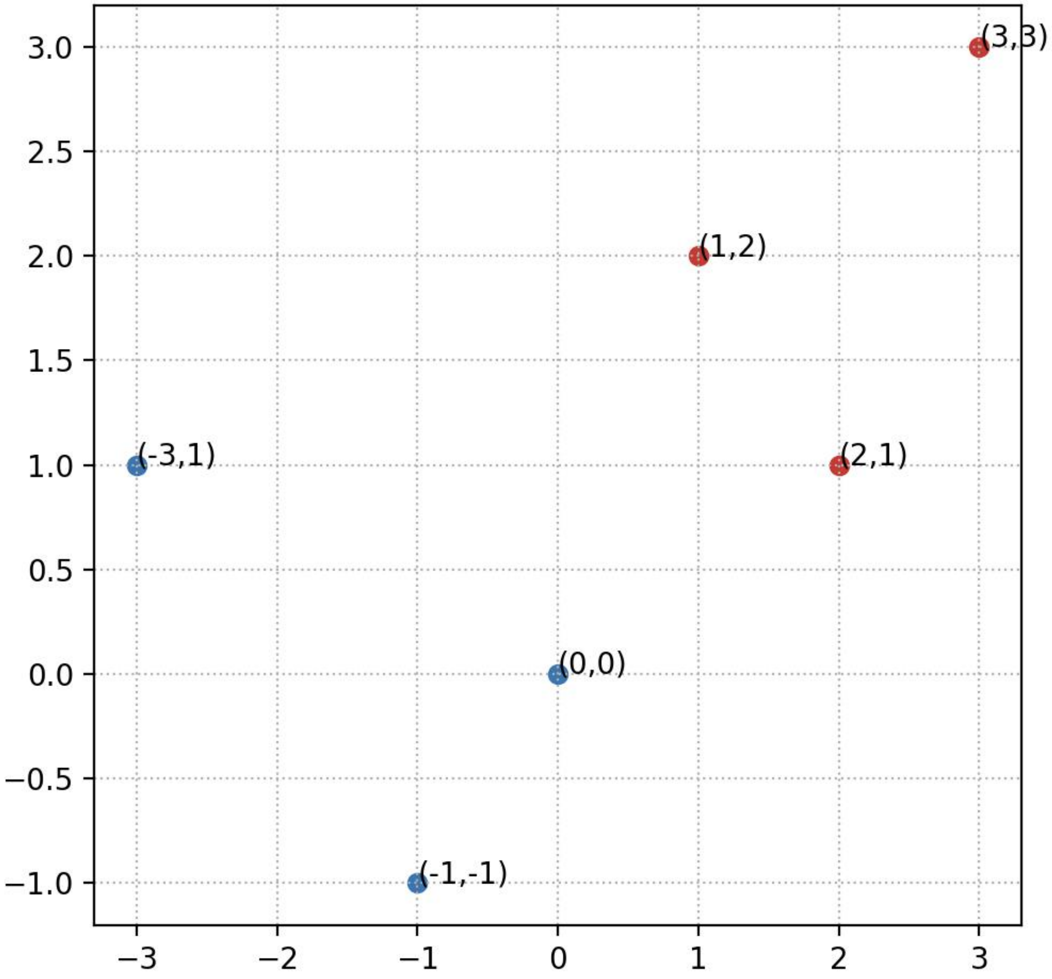


- when training data is linearly separable, there exist many hyperplanes to sperate the training data points. among them,
- a maximum margin classifier is desirable because

  (1) the gap between train & true distribution

  (2) need a robust classifier giving consistent results to a slight changes in data distribution.

[2] a naïve way to find a maximum margin classifier



1. pick one blue and one red
2. compute a hyper plane by the two points.
3. compute the margin for the plane
4. repeat 1.2,3 for other points combinations.
5. pick the plane gives the maximum margin

   and separates red and blue.

- Objective Function: Maximum Margin Classification

We start with this assumption:

(1) data is already projected to a feature space

(2) data is linearly separable in the feature space

[1] Objective for Maximum Margin

$$w*, b* = \arg\max_{w,b} \min_{n} \frac{t_n(w^t x_n + b)}{||w||} \quad (1)$$

$$\Leftrightarrow w*, b* = \arg\max_{w,b} \frac{\Delta}{||w||} \quad (2)$$

$$\text{subject to} \quad t_n(w^t x_n + b) \geq \Delta \quad \forall n$$

$$\Delta \geq 0$$

when $\Delta = min_n \, t_n \, (w^t x_n + b)$

Q: What is the t_n?
t_n is 1/ 0 ? or +1/ − 1?

[2] Objective for Maximum Margin (scaled version)

$$w'*, b'* = \arg\max_{w,b} \frac{1}{||w'||} \quad (3)$$

$$\text{subject to} \quad t_n(w'^t x_n + b') \geq 1 \quad \forall n$$

- The scaling factor $1/\Delta$ ($\vec{w}$ + b) does not change the hyperplane and also the margin $\frac{1}{||w/\Delta||}$ will be the same

[3] Objective for Maximum Margin (inverse version)

$$w*, b* = \arg\max_{w,b} \frac{1}{||w||} \tag{4}$$

subject to $\quad t_n(w^t x_n + b) \geq 1 \quad \forall n$

$$w*, b* = \arg\min_{w,b} ||w|| \tag{5}$$

subject to $\quad t_n(w^t x_n + b) \geq 1 \quad \forall n$

$$w*, b* = \arg\min_{w,b} ||w||^2 \tag{6}$$

subject to $\quad t_n(w^t x_n + b) \geq 1 \quad \forall n$

this will give the equivalent optimum
$|x^*| < |x| \leftrightarrow |x^*|^2 < |x|^2$

[4] Objective for Maximum Margin (primary)

[primary problem]

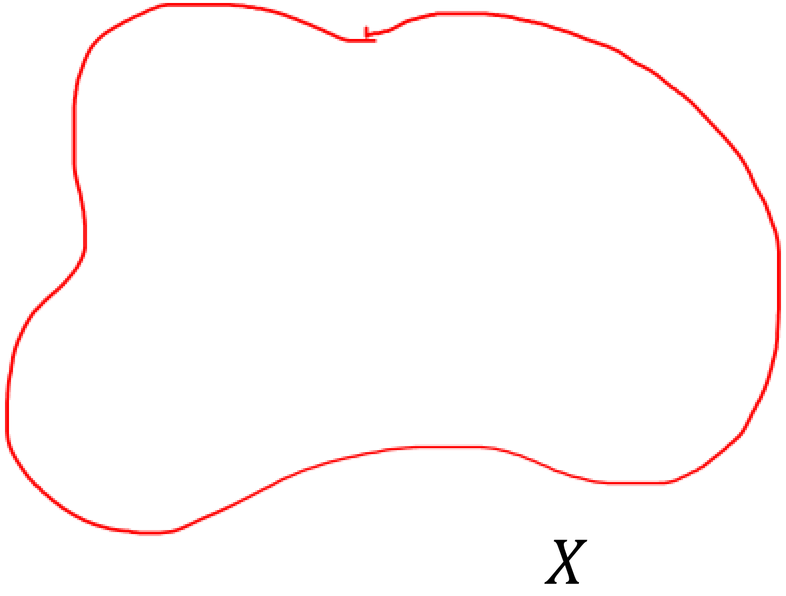
$$w*, b* = \arg\min_{w,b} \frac{1}{2}||w||^2$$

$$\text{subject to} \quad t_n(w^t x_n + b) \geq 1 \quad \forall n$$

- this is a convex & inequality constrained problem.
- Q: # constraints?åå

**Review: Inequality Constraint Problem (necessary conditions)

$$\min_x \quad f(x)$$
$$\text{s.t.} \quad g_i(x) \le 0 \quad i = 1, \ldots, m$$



- two possible cases for x^*

  (1) x^* inside of the manifold by $g_i(x) < 0$

  (2) x^* on the boundary of $g_i(x) = 0$

**Review: Inequality Constraint Problem (KKT necessary conditions)

Let x^* be a local minimum of the problem

$$\min_x \quad f(x)$$

$$\text{s.t.} \quad g_i(x) \leq 0 \quad i = 1, \ldots, m$$

important to set the inequality this form (less than or equal to)!

Then, there exist $\lambda_i, \; i = 1, \ldots, m$ such that

(1) $$\nabla f(x*) + \sum_{i=1}^{m} \lambda_i \nabla g_m(x*) = 0$$

- stationary condition

(2) $$\begin{cases} \lambda_j \geq 0 & j = 1, \ldots, r \\ \lambda_j = 0 & \forall j \notin A(x*) \end{cases}$$

- complementary slackness condition
  $\lambda_i \cdot g_i(x^*) = 0$

$A(x^*)$ is the set of active constraints at x^*

**Review: Inequality Constraint Problem (KKT necessary conditions)

Let x^* be a local minimum of the problem

$$\min_{x} \quad f(x)$$
$$\text{s.t.} \quad g_i(x) \leq 0 \quad i = 1, \ldots, m$$

Then, there exist $\lambda_i,\ i = 1, \ldots, m$ such that

$$(1) \quad \nabla f(x*) + \sum_{i=1}^{m} \lambda_i \nabla g_m(x*) = 0$$

$$(2) \quad \begin{cases} \lambda_j \geq 0 & j = 1, \ldots, r \\ \lambda_j = 0 & \forall j \notin A(x*) \end{cases}$$

$$(3) \quad g(x^*) \leq 0$$

- Primary feasibility

[5] Objective for Maximum Margin (Lagrangian Function)

[primary problem]

$$w*, b* = \arg\min_{w,b} \frac{1}{2}||w||^2$$

$$\text{subject to} \quad t_n(w^t x_n + b) \geq 1 \quad \forall n$$

[Lagrangian Function]

$$L(w, b, \lambda_{n=1}^N) = \frac{1}{2}||w||^2 + \sum_{n=1}^N \lambda_n - \sum_{n=1}^N \lambda_n \cdot t_n \cdot (w^t x_n + b)$$

Q: how many Lagrangian parameters are involved?

[6] Objective for Maximum Margin (KKT conditions)

[Lagrangian Function]

$$L(w, b, \lambda_{n=1}^{N}) = \frac{1}{2}||w||^2 + \sum_{n=1}^{N} \lambda_n - \sum_{n=1}^{N} \lambda_n \cdot t_n \cdot (w^t x_n + b)$$

[by the stationary condition, two conditions are derived below]

$$\nabla_w L(w, b) = \vec{w*} - \sum_{n=1}^{N} \lambda_n^* \cdot t_n \cdot \vec{x_n} = 0 \qquad (1) \longrightarrow \qquad \vec{w*} = \sum_{n=1}^{N} \lambda_n^* \cdot t_n \cdot \vec{x_n}$$

$$\nabla_b L(w, b) = \sum_{n=1}^{N} \lambda_n^* \cdot t_n = 0 \qquad (2)$$

[7] Objective for Maximum Margin (dual representation)

- [Lagrangian Function]

$$L(w, b, \lambda_{n=1}^{N}) = \frac{1}{2}||w||^2 + \sum_{n=1}^{N} \lambda_n - \sum_{n=1}^{N} \lambda_n \cdot t_n \cdot (w^t x_n + b)$$

[plug in optimal] $$\vec{w*} = \sum_{n=1}^{N} \lambda_n^* \cdot t_n \cdot \vec{x_n}$$

- [Dual Representation]

$$D(\lambda) = -\frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n \cdot \lambda_m \cdot t_n \cdot t_m \cdot \vec{x_n}^t \vec{x_m} + \sum_{n=1}^{N} \lambda_n$$

Q: maximize? minimize?

[8] Objective for Maximum Margin (primary vs. dual)

[Relationship between **dual** and **primal**]

$$D(\lambda) = L(\lambda, x*) = f(x*) + \lambda g(x*) \leq f(x*)$$

- the dual function is the lower bound of the primary objective function. why?
- we know that
  $L(\lambda*, x*) = f(x*) + \lambda^* g(x*) = f(x*)$
- hence, we need to maximize / minimize (?) to find the optimal λ^*!

[9] Objective for Maximum Margin (dual representation)

[dual problem]

$$\arg\max_{\lambda_{n=1}^{N}} -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \lambda_n \cdot \lambda_m \cdot t_n \cdot t_m \cdot \vec{x_n}^{t}\vec{x_m} + \sum_{n=1}^{N} \lambda_n$$

$$\text{subject to} \quad \lambda_n \geq 0$$

$$\sum_{n=1}^{N} \lambda_n \cdot t_n = 0$$

- this is the quadratic optimization problem we need to solve!

[10] Objective for Maximum Margin (dual solution)

when we found the dual solutions $\lambda_1, \lambda_2, \lambda_3 .... \lambda_N$,

by the complementary slackness condition

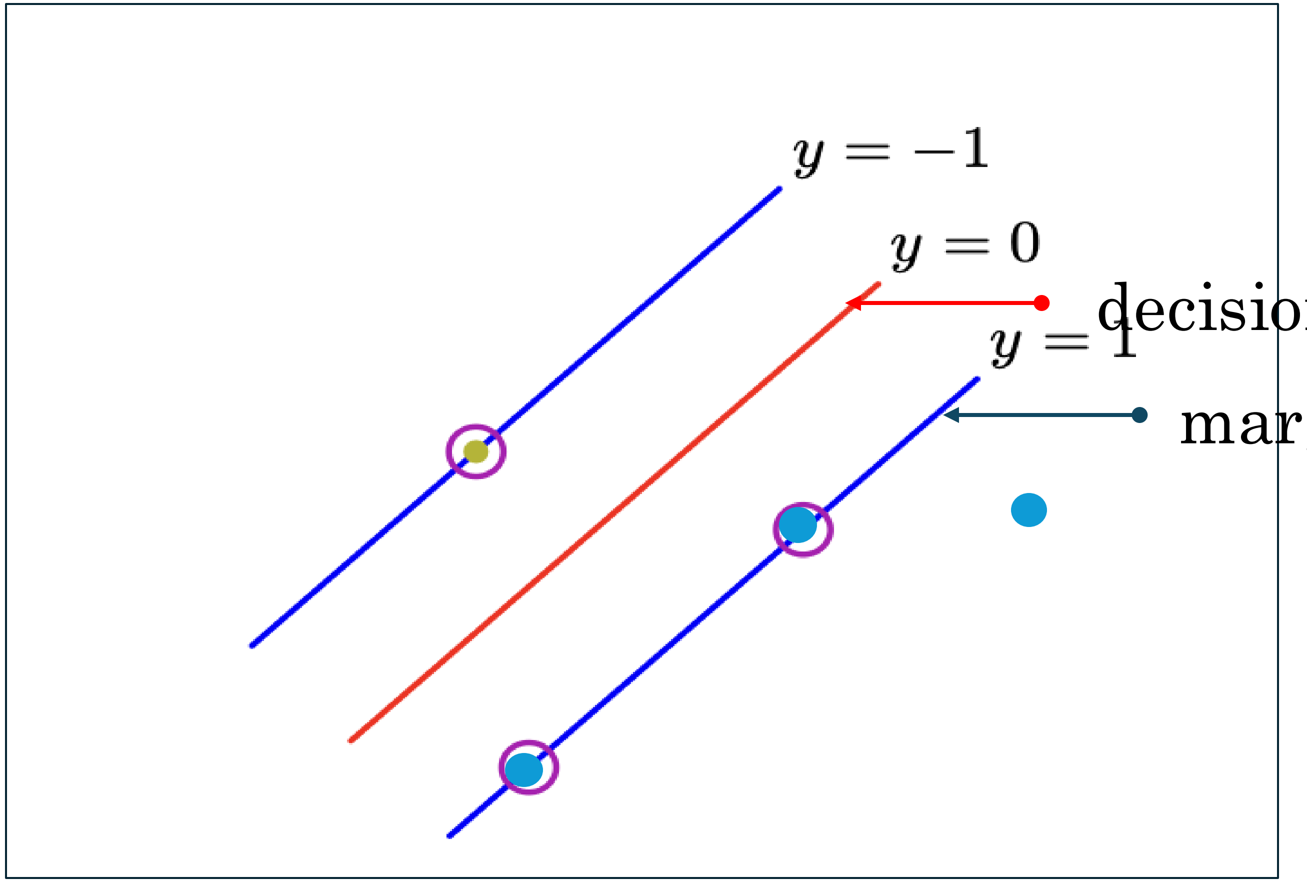
the following is true.

$$\begin{cases} \lambda_i = 0, & t_i(w^t x_i + b) > 1 \\ \lambda_i > 0, & t_i(w^t x_i + b) = 1 \end{cases}$$

- the data points on the correct side of margin.
- the data points on the margin.

[11] Objective for Maximum Margin (dual solution)

From Bishop Figure 7.1



decision boundary

margin boundary

- data points corresponding to the positive Lagrangian on the margin.

[12] Objective for Maximum Margin (**S**upport **V**ector **M**achine: SVM)

- decision boundary is defined by the data points on the margin.
- the data points on the margin ($\lambda^*{}_n$) are called "support vectors"

$$\vec{w}* = \sum_{n=1}^{N} \lambda_n^* t_n \phi(x)$$

[SVM classifier]

$$y(x) = \sum_{n=1}^{N} \lambda_n^* t_n \phi(x)^t \phi(x) + b \quad \begin{cases} y(x) \geq 0 & x \in + \\ y(x) < 0 & x \in - \end{cases}$$

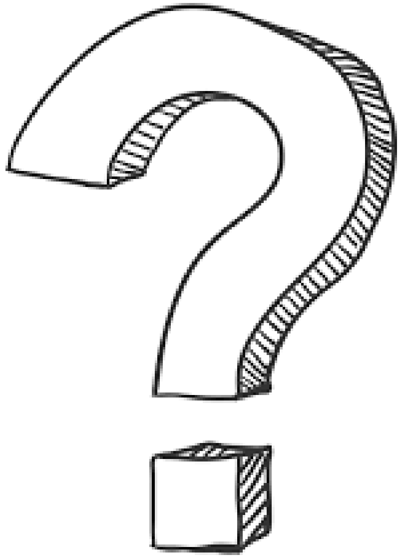
[13] Objective for Maximum Margin (dual representation)

[dual problem]

$$\arg\max_{\lambda_{n=1}^{N}} -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\lambda_n \cdot \lambda_m \cdot t_n \cdot t_m \cdot \vec{x_n}^t \vec{x_m} + \sum_{n=1}^{N}\lambda_n$$

$$\text{subject to} \quad \lambda_n \geq 0$$

$$\sum_{n=1}^{N}\lambda_n \cdot t_n = 0$$



- by the way,
- primary and dual are quadratic optimization; also dual is on N dimensional ($N >> M$: # feature dim in general)
- why do we want to solve dual?

[14] Objective for Maximum Margin (dual representation)

[dual problem]

$$\arg\max_{\lambda_{n=1}^{N}} -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\lambda_n \cdot \lambda_m \cdot t_n \cdot t_m \cdot \boxed{\phi(x_n)^t \phi(x_m)} + \sum_{n=1}^{N}\lambda_n$$

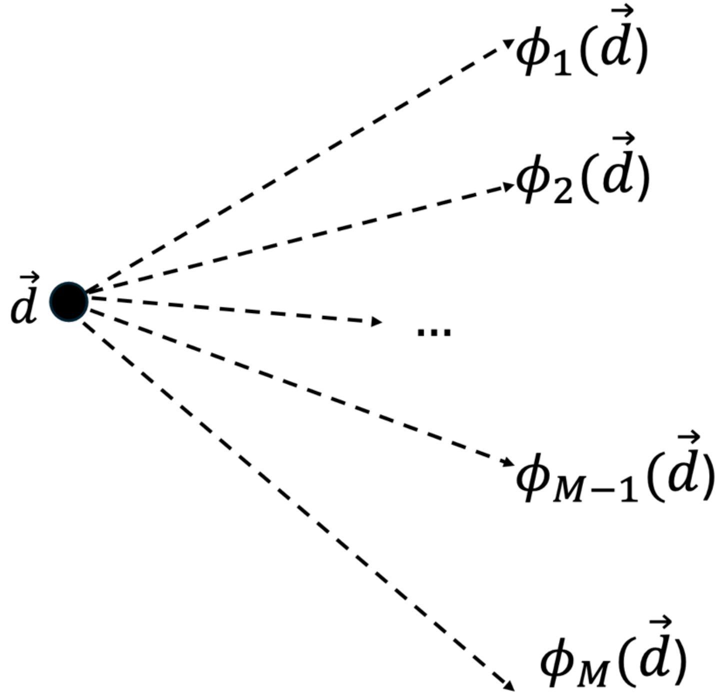
$$\text{subject to} \quad \lambda_n \geq 0$$

$$\sum_{n=1}^{N}\lambda_n \cdot t_n = 0$$

- we don't need explicit feature design, the dual function only needs inner product?

Q: what is the advantage of using this?

** well designed feature space makes data linearly separable!



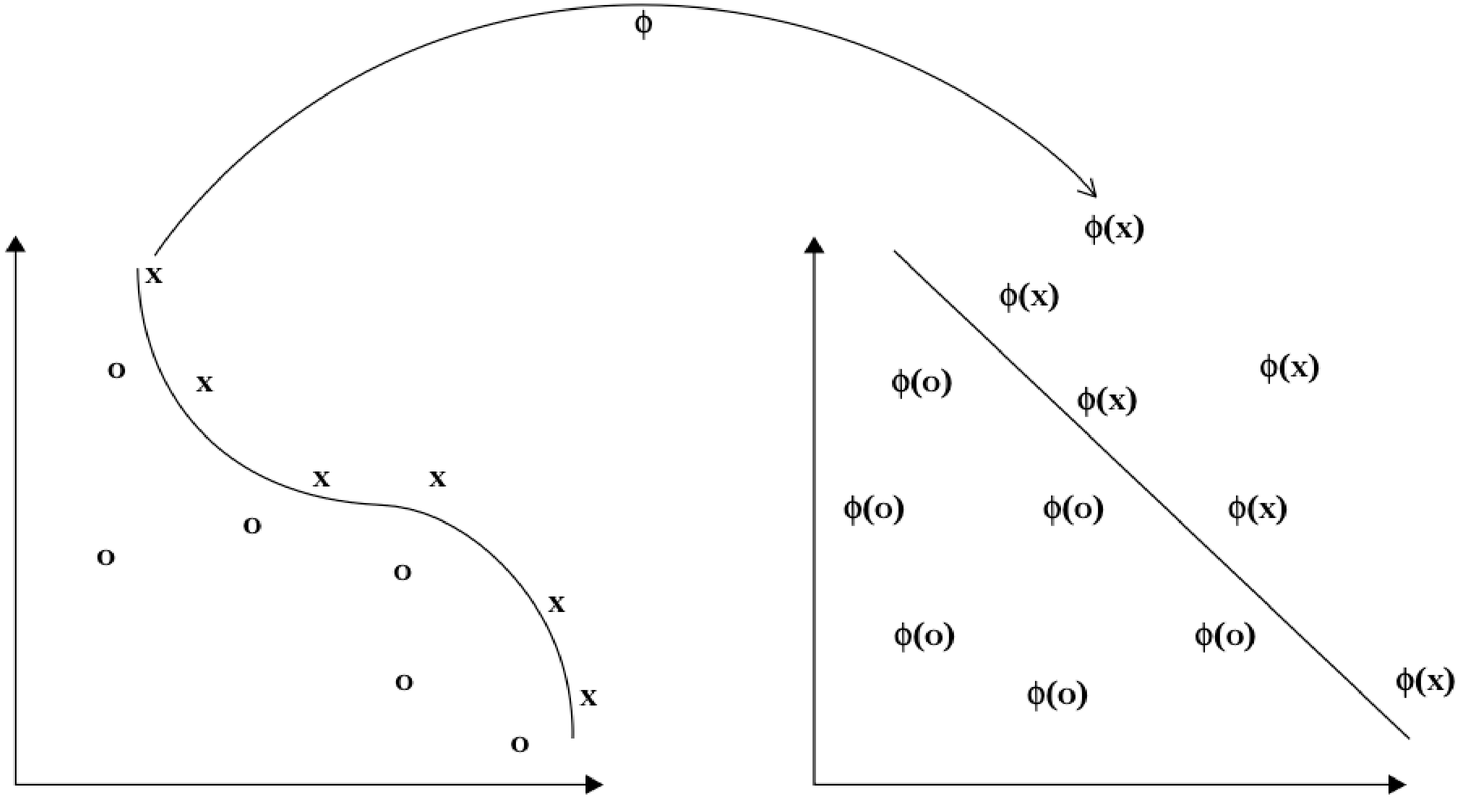


Fig. 2.1. The function ϕ embeds the data into a feature space where the nonlinear pattern now appears linear. The kernel computes inner products in the feature space directly from the inputs.

From Kernel Methods for Pattern Analysis by John Shawe-Talyor

[15] Objective for Maximum Margin (kernel trick)

[dual problem]

$$\operatorname*{arg\,max}_{\lambda_{n=1}^{N}} -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \lambda_n \cdot \lambda_m \cdot t_n \cdot t_m \cdot \phi(x_n)^t \phi(x_m) + \sum_{n=1}^{N} \lambda_n$$

$$\text{subject to} \quad \lambda_n \geq 0$$

$$\sum_{n=1}^{N} \lambda_n \cdot t_n = 0$$

- dual function only needs inner product values without explicitly designing the feature map. This is called “Kernel Trick”!

- Kernel Functions :

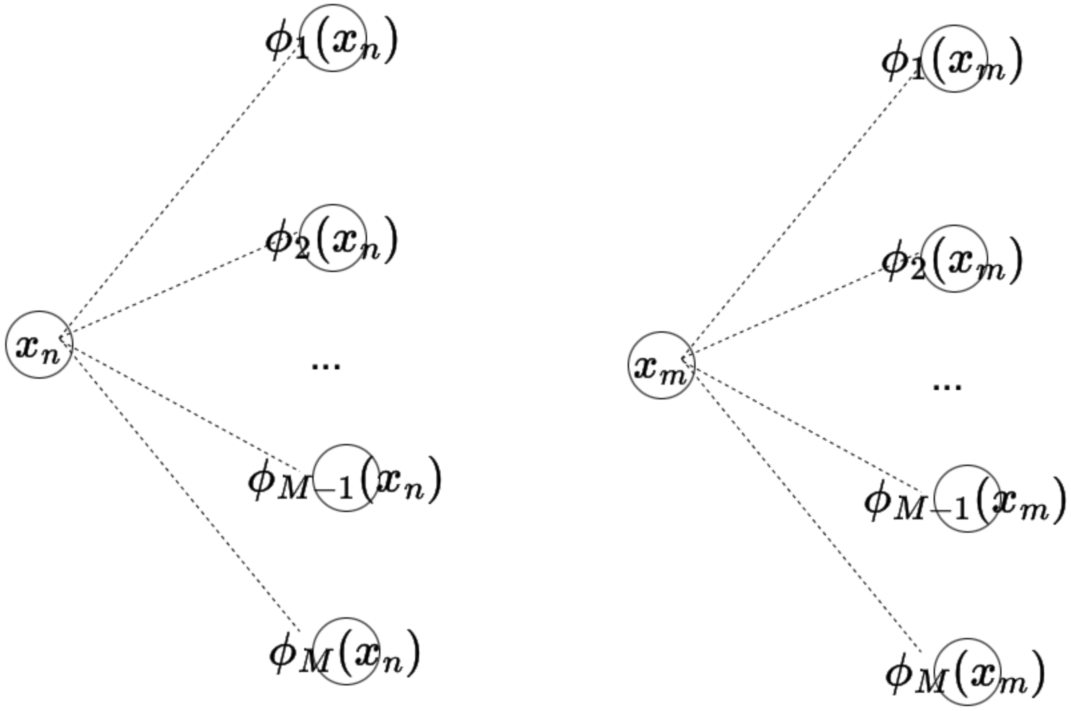
$$\kappa\ (x_n, x_m), \chi \ \times \ \chi \ \rightarrow \ \mathbb{R} = \phi(x_n)^t \phi(x_m)$$

[1] Kernel Function (definition)

- kernel function
  computes the inner product (similarity) between two data points in feature space.

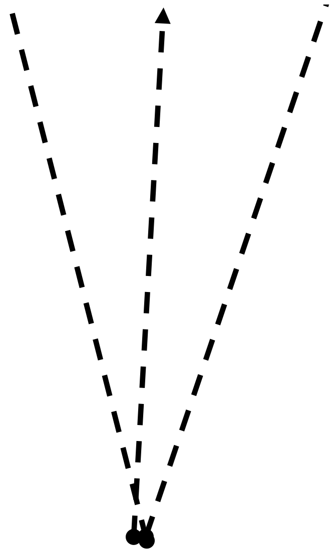
$$\kappa(x_n, x_m) = \phi(x_n)^t \phi(x_m)$$

$$\kappa(x_n, x) = \phi(x_n)^t \phi(x)$$



[2] Kernel Function (example1)

$1, x_1^2, x_2^2$



$(x1, x2)$

- feature map

Q: $\kappa(x, x')$?

$$\begin{aligned}\kappa(x,x') &= (1, x_1^2, x_2^2)^t \cdot (1, x_1'^{\,2}, x_2'^{\,2}) \\ &= 1 + x_1^2 x_1'^{\,2} + x_2^2 x_2'^{\,2}\end{aligned}$$

[3] Kernel Function (example2: polynomial kernel)

$$(1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

$$\begin{aligned}
\phi(x)^t \cdot \phi(x') &= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2)^t \cdot (1, \sqrt{2}x'_1, \sqrt{2}x'_2, {x'_1}^2, \sqrt{2}x'_1x'_2, {x'_2}^2) \\
&= 1 + 2x_1x'_1 + 2x_2x'_2 + x_1^2{x'_1}^2 + 2x_1x'_1x_2x'_2 + x_2^2{x'_2}^2 \\
&= (1 + x_1x'_1 + x_2x'_2)^2 \\
&= (1 + x^tx')^2
\end{aligned}$$

$(x1, x2)$

[feature map]

- linear kernel $\kappa(x, x') = x^tx$ [original space]
- polynomial kernel $\kappa(x, x') = (x^tx + 1)^p$

[polynomial space]

[4] Kernel Function (validity)

$\kappa(x_1, x_2)$ is a kernel function

iff Gram matrix K ($\Phi\Phi^{t}$) is a positive semidefinite.

[5] Kernel Function (Gram Matrix)

Given a kernel function and data samples, we can compute a gram matrix for data points $x_1, x_2, \ldots, x_n$:

$$K = \begin{bmatrix} \kappa(x_1, x_1) & \kappa(x_1, x_2) & \ldots & \kappa(x_1, x_n) \\ \kappa(x_2, x_1) & \kappa(x_2, x_2) & \ldots & \kappa(x_2, x_n) \\ \ldots & \ldots & \ldots & \ldots \\ \kappa(x_{n-1}, x_1) & \kappa(x_{n-1}, x_2) & \ldots & \kappa(x_{n-1}, x_n) \\ \kappa(x_n, x_1) & \kappa(x_n, x_2) & \ldots & \kappa(x_n, x_n) \end{bmatrix}$$

data matrix (# data × #features)

$$= \begin{bmatrix} \phi(x_1)^t\phi(x_1) & \phi(x_1)^t\phi(x_1) & \ldots & \phi(x_1)^t\phi(x_n) \\ \phi(x_2)^t\phi(x_1) & \phi(x_2)^t\phi(x_2) & \ldots & \phi(x_2)^t\phi(x_n) \\ \ldots & \ldots & \ldots & \ldots \\ \phi(x_{n-1})^t\phi(x_1) & \phi(x_{n-1})^t\phi(x_2) & \ldots & \phi(x_{n-1})^t\phi(x_n) \\ \phi(x_n)^t\phi(x_1) & \phi(x_n)^t\phi(x_2) & \ldots & \phi(x_n)^t\phi(x_n) \end{bmatrix} = \boxed{\Phi} \cdot \Phi^t$$

[6] Kernel Function (technique for constructing new kernels)

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = c\mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}'),\ \text{for any constant } c > 0$$

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = f(\boldsymbol{x})\mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}')f(\boldsymbol{x}'),\ \text{for any function } f$$

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = q(\mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}')) \text{ for any function polynomial } q \text{ with nonneg. coef.}$$

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp(\mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}'))$$

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^{\mathsf{T}}\mathbf{A}\boldsymbol{x}',\ \text{for any psd matrix } \mathbf{A}$$

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}') + \mathcal{K}_2(\boldsymbol{x}, \boldsymbol{x}')$$

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}') \times \mathcal{K}_2(\boldsymbol{x}, \boldsymbol{x}')$$

[7] Kernel Function (Gaussian Kernel)

$$\kappa(x, x') = \exp \frac{-||x - x'||^2}{2\sigma^2}$$

Q: how Gaussian kernel is valid?

[8] Kernel Function (Gaussian Kernel)

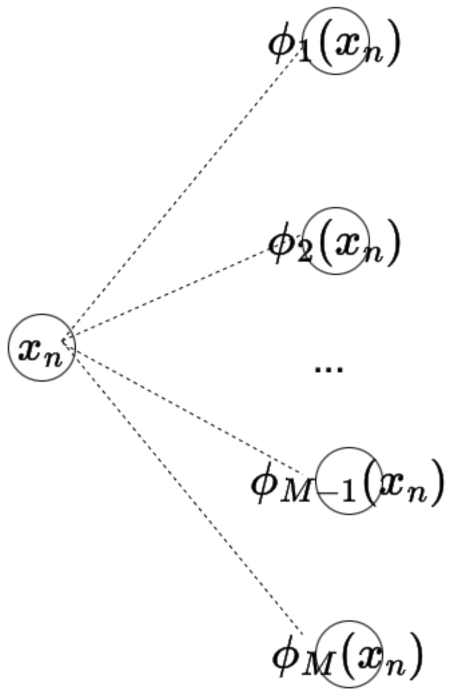
$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = c\mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}'), \text{ for any constant } c > 0$$
$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = f(\boldsymbol{x})\mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}')f(\boldsymbol{x}'), \text{ for any function } f$$
$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = q(\mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}')) \text{ for any function polynomial } q \text{ with nonneg. coef.}$$
$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp(\mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}'))$$
$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^\top \mathbf{A} \boldsymbol{x}', \text{ for any psd matrix } \mathbf{A}$$

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}') + \mathcal{K}_2(\boldsymbol{x}, \boldsymbol{x}')$$
$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \mathcal{K}_1(\boldsymbol{x}, \boldsymbol{x}') \times \mathcal{K}_2(\boldsymbol{x}, \boldsymbol{x}')$$

$$\begin{aligned}\kappa(x, x') &= \exp \frac{-\,||x - x'||^2}{2\sigma^2} \\ &= \exp \frac{-\,||x||^2 - ||x'||^2 + 2x^t x'}{2\sigma^2}\end{aligned}$$

[9] Kernel Function (Gaussian Kernel)

the feature vector of Gaussian kernel has infinite dimensionality



$$
\begin{aligned}
\kappa(x, x') &= \exp \frac{-||x - x'||^2}{2\sigma^2} \\
&= \exp \frac{-||x||^2 - ||x'||^2 + 2x^t x'}{2\sigma^2} \\
&= \exp -||x||^2/2\sigma^2 \cdot \exp x^t x' \cdot \exp -||x'||^2/2 \\
&= \exp -||x||^2/2\sigma^2 (\sum_{k=0}^{\infty} \frac{(x^t x')^k}{k!}) \cdot \exp -||x'||^2/2
\end{aligned}
$$

$$\exp^x = \sum_{k=1}^{\infty} \frac{x^k}{k!}$$ + Taylor series!

- Going back to Support Vector Machine

[1] Objective for Maximum Margin (kernel trick)

[dual problem]

$$\underset{\lambda_{n=1}^{N}}{\arg\max} -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\lambda_n \cdot \lambda_m \cdot t_n \cdot t_m \cdot \kappa(x_n, x_m) + \sum_{n=1}^{N}\lambda_n$$

$$\text{subject to} \quad \lambda_n \geq 0$$

$$\sum_{n=1}^{N}\lambda_n \cdot t_n = 0$$

[SVM classifier]

$$y(x) = \sum_{n=1}^{N}\lambda_n^* t_n \kappa(x_n, x) + b$$

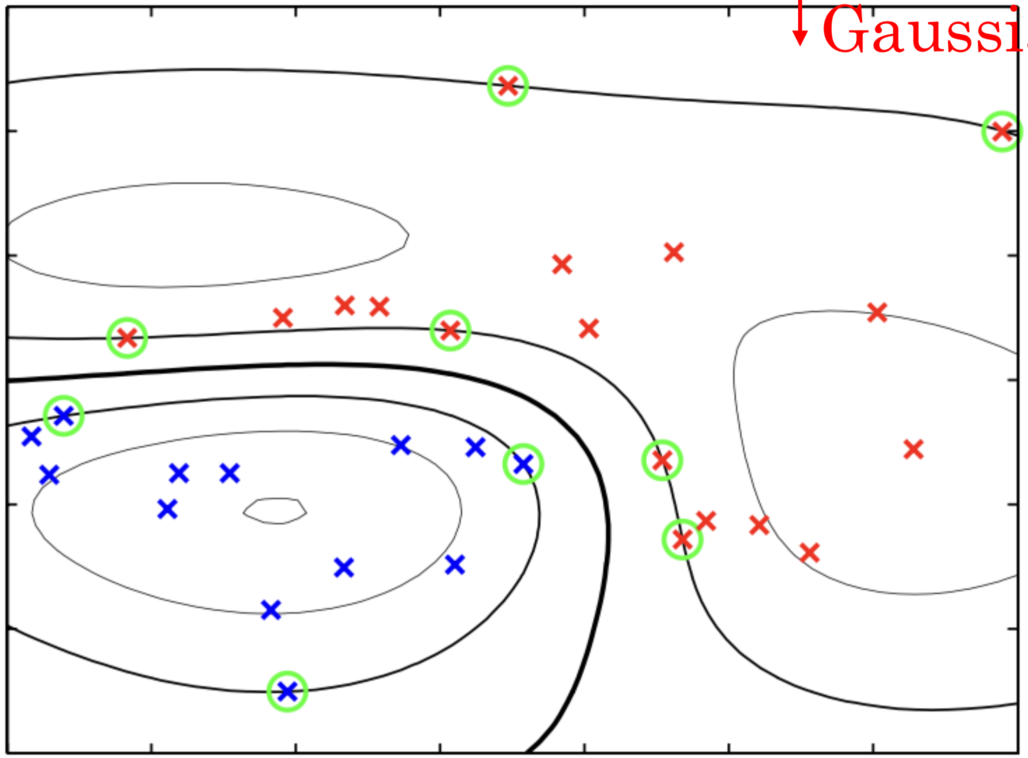
- by using a proper kernel function,
  we can compute a maximum margin classifier in a high dimensional feature space without designing a feature space directly.

[2] Objective for Maximum Margin (kernel trick)

- even when data is not linearly separable,
  we can make the data linearly separable in the high dimensional space.
- using Gaussian kernel, we can make the data separable always.
  (transform to infinite dimensions)

[3] Objective for Maximum Margin (Gaussian Kernel SVM)

[Bishop Fig 7.2] $$y(x) = \sum_{n=1}^{N} \lambda_n^* t_n \underbrace{\kappa(x_n, x)}_{\text{Gaussian kernel}} + b$$



[non-linear decision boundary & margins]

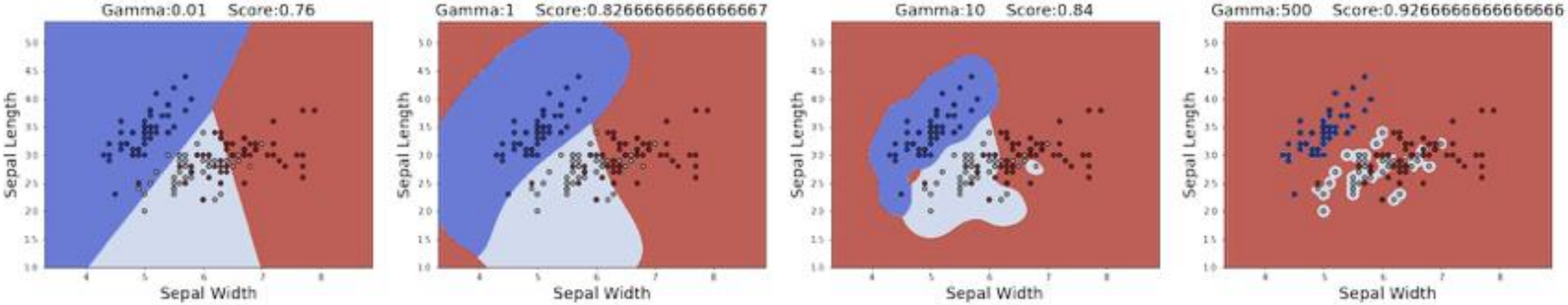
- In the original data space, the data points are not linearly separable, but the kernel tricks implicitly transform them to infinite dimensional feature space, so a maximum margin classifier can be found, making the feature space is linearly separable.

[4] Objective for Maximum Margin (Gaussian Kernel SVM)

$$\gamma = \frac{1}{\sigma^2}$$

From https://www.kaggle.com/code/gorkemgunay/understanding-parameters-of-svm

the effect of gamma on # of support vectors & decision Boundary



- small γ: some representative samples become support vectors.
- large γ : every samples become support vectors
- depending on γ, model complexity varies.

[5] Objective for Maximum Margin (Gaussian Kernel SVM)

Q: Gaussian kernel embeds infinite dimensional feature space. Is that high complexity okay with a finite number of data points?

[6] Objective for Maximum Margin (Gaussian Kernel SVM)

Q: Gaussian kernel embeds infinite dimensional feature space. Is that high complexity okay with a finite number of data points?

SVM is not based on MLE instead the maximum margin boundary can be defined by only two + / – data samples; SVM is less sensitive to # data points than other parametric/ MLE based models. (of course, more data is always helpful!)