Machine Learning

Amin Golzari Oskouei

a.golzari@azaruniv.ac.ir

a.golzari@tabrizu.ac.ir

https://github.com/Amin-Golzari-Oskouei

Azarbaijan Shahid Madani University
2023

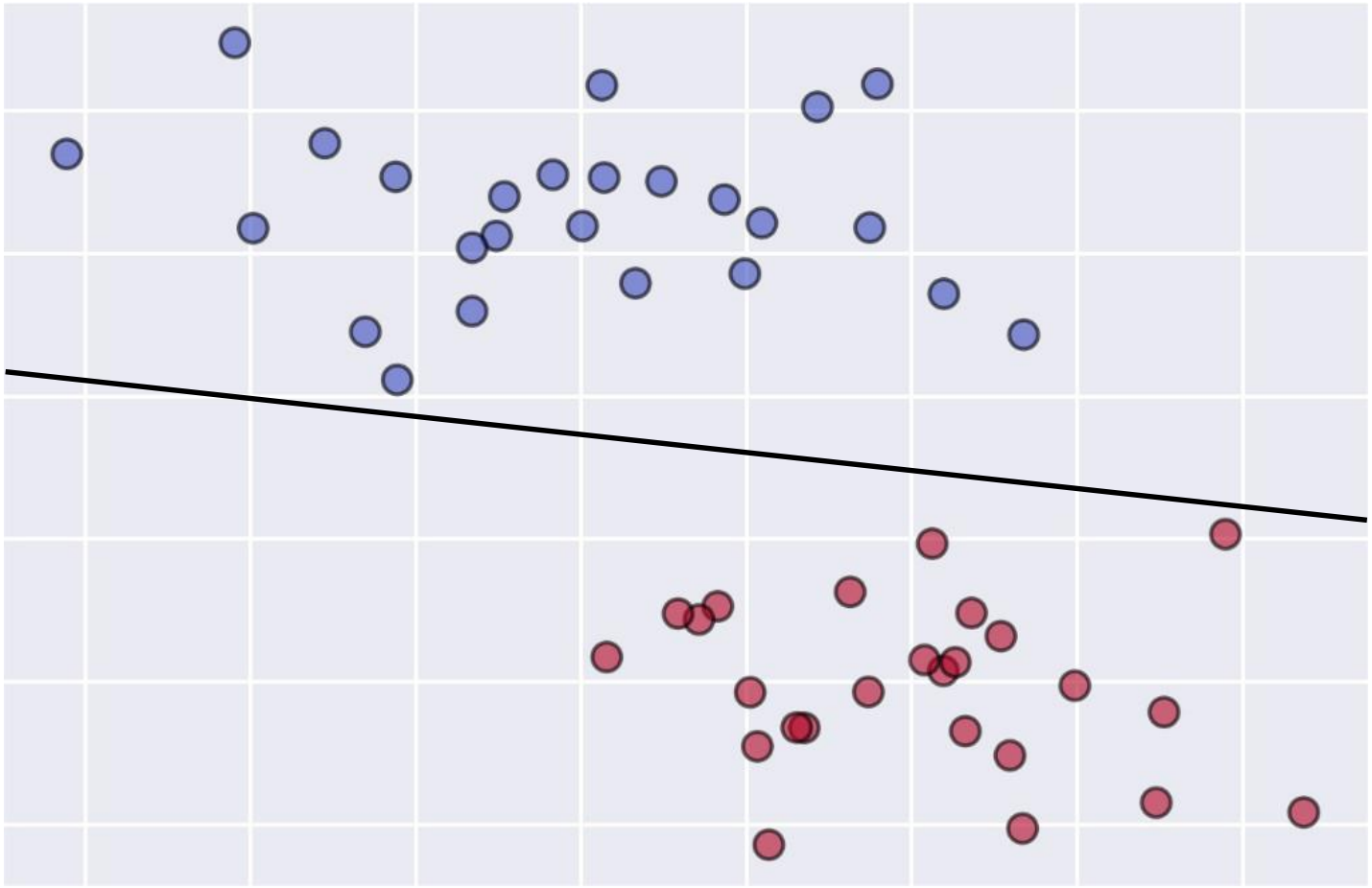
Support Vector Machines : SVM

Table of Contents

Introduction

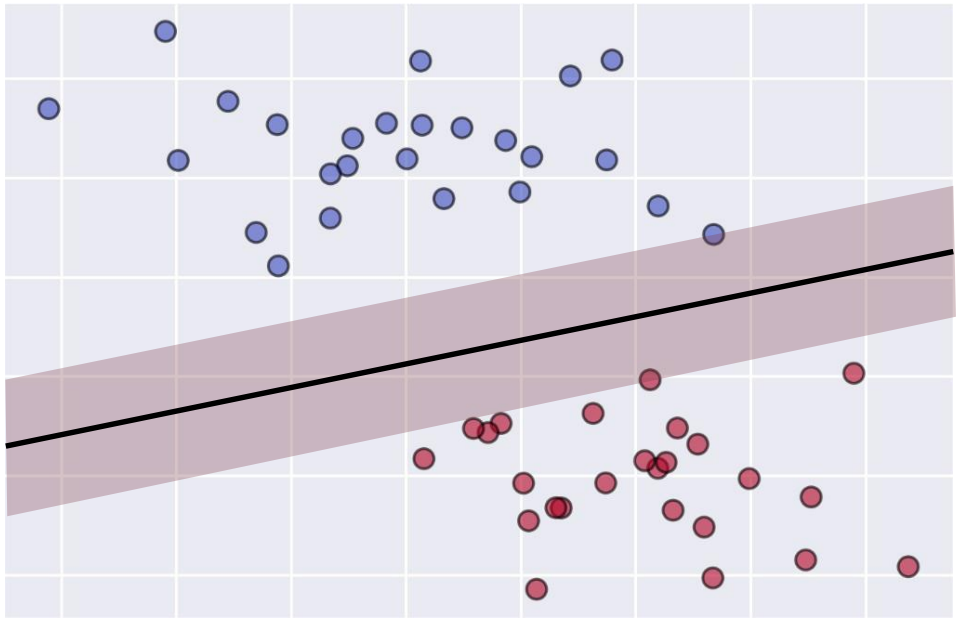
- Support Vector Machines [Vanpik, 1990's]
  - One of the most popular machine learning algorithms!
  - Better data separation compared to other machine learning methods (classification problems)!
  - Relatively easy to use!
  - Using kernel tricks :
    - classification, regression, distribution estimation, one-class classification, and more.

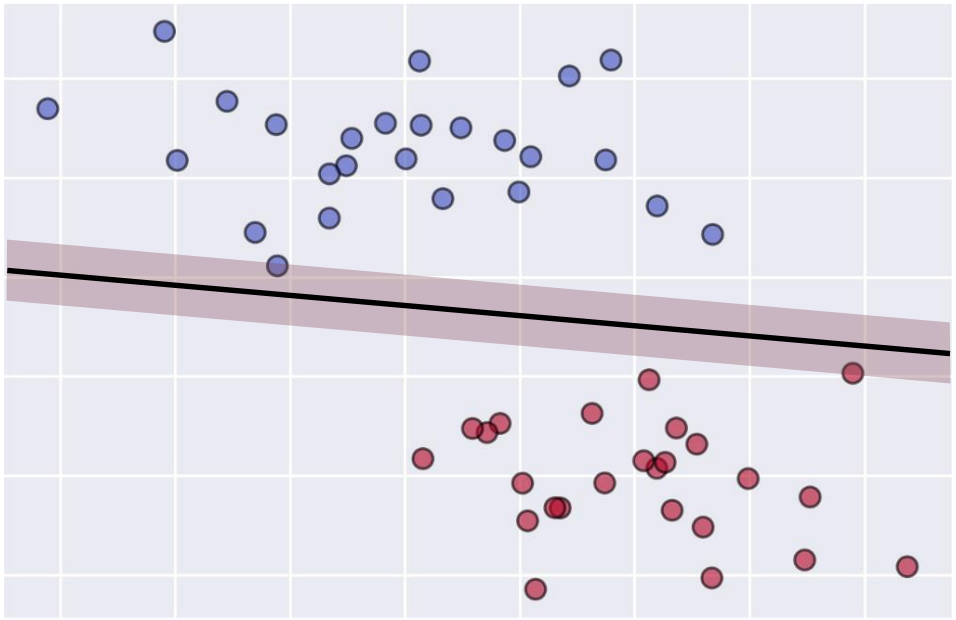
Motive: Linearly separable data



Motive: Optimal decision boundary

- Question : Which decision boundary is better?

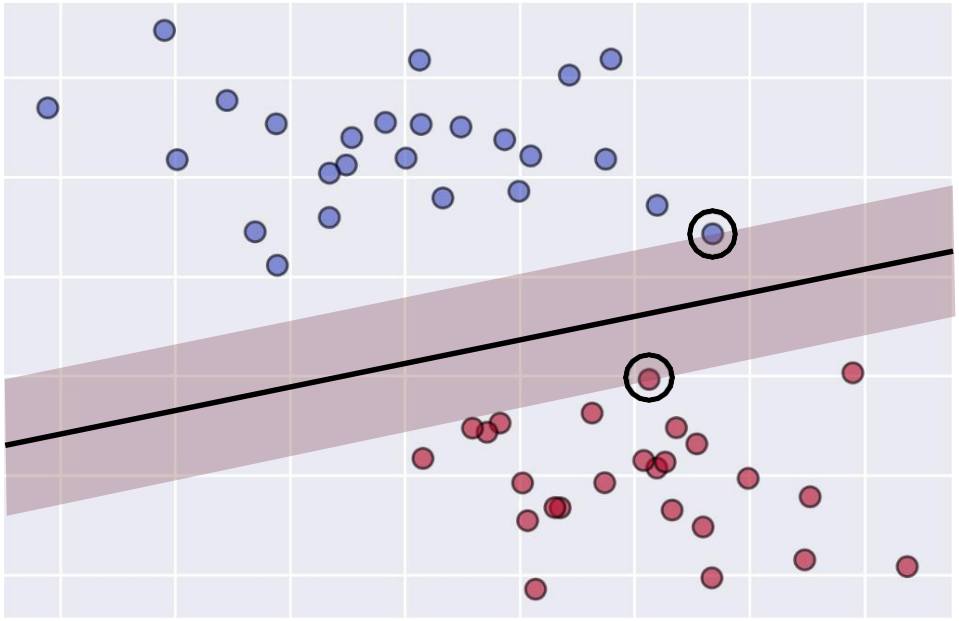




- Solution for maximum margin : Maximum robustness against data corruption. [Increasing generalization capability]

Motive: Support Vectors

- Support Vector : Nearest data points to the decision boundary.

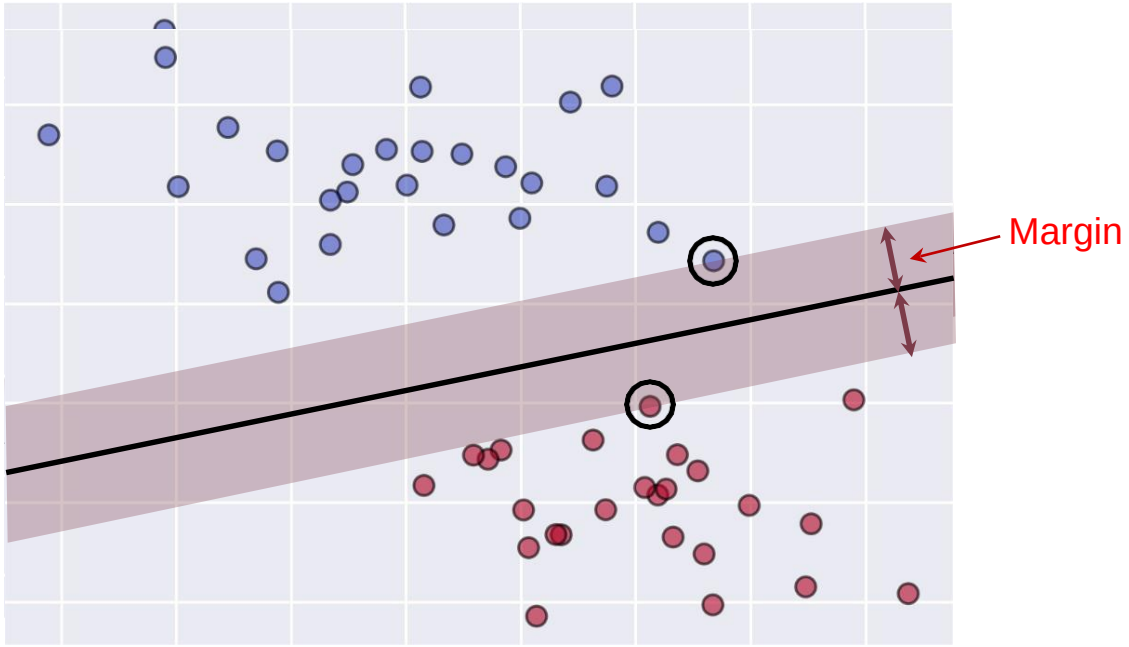


- Objective: Maximizing the distance between support vectors and the decision boundary.

Motive: Support Vectors

- Support Vector : Distance of support vectors from decision boundary.

$$w^T x + b = 0$$



- Objective: Maximizing the distance between support vectors and the decision boundary.

Optimal Decision Boundary: Symbols

- Training examples

$$X = (x^t, y^t), \qquad y^t = \begin{cases} +1 & if \;\; x^t \epsilon \, C_1 \\ -1 & if \;\; x^t \, \epsilon \, C_2 \end{cases}$$

- Objective: Finding the vector w and the value b in a way that :

$$\boldsymbol{w}^T \boldsymbol{x}^t + b \geq +1 \qquad \text{for} \quad y^t = +1$$

$$\boldsymbol{w}^T \boldsymbol{x}^t + b \leq -1 \qquad \text{for} \quad y^t = -1$$

$$\Rightarrow \quad y^t(\boldsymbol{w}^T \boldsymbol{x}^t + b) \geq +1$$

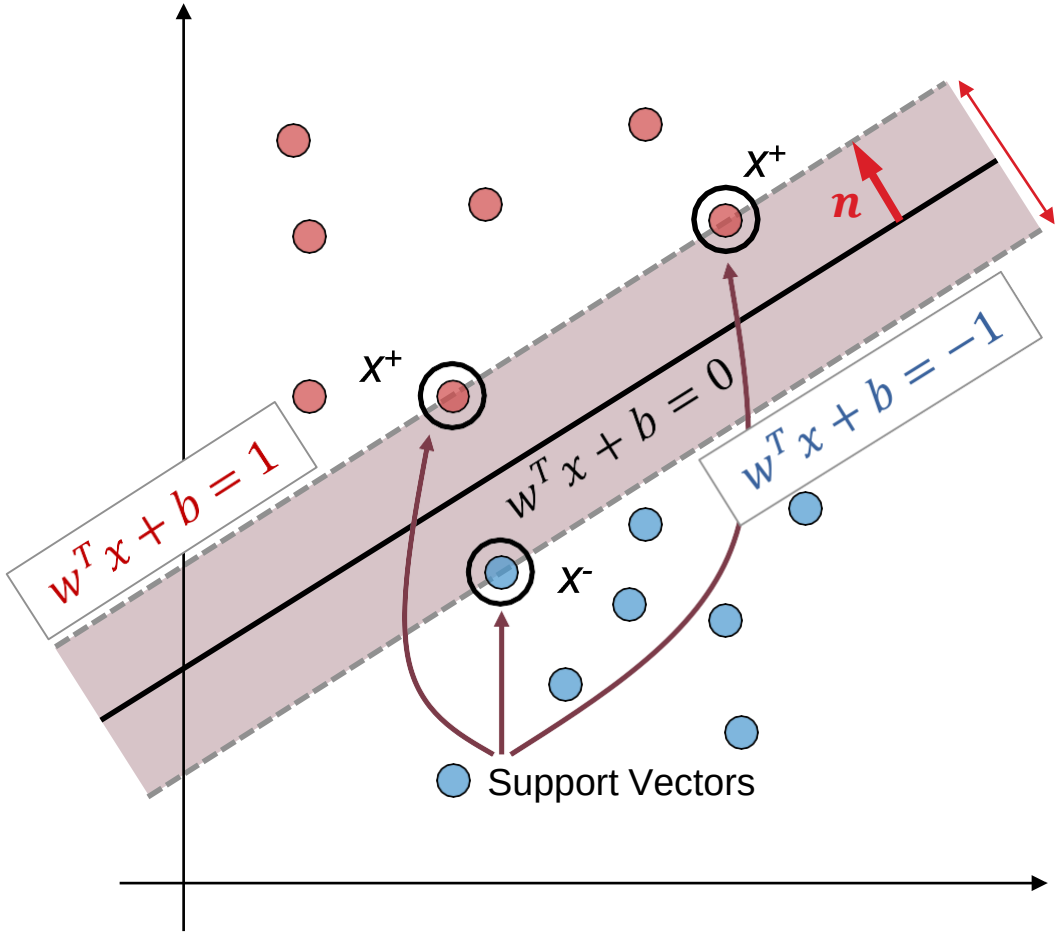
$$\max(0, 1 - \; y^t(\boldsymbol{w}^T \boldsymbol{x}^t + b) \;)$$

Objective function : Calculate margin

- We already know :

$$w^T x^+ + b = +1$$

$$w^T x^- + b = -1$$



Objective function : Calculate margin

- We already know :

$$w^T x^+ + b = +1$$

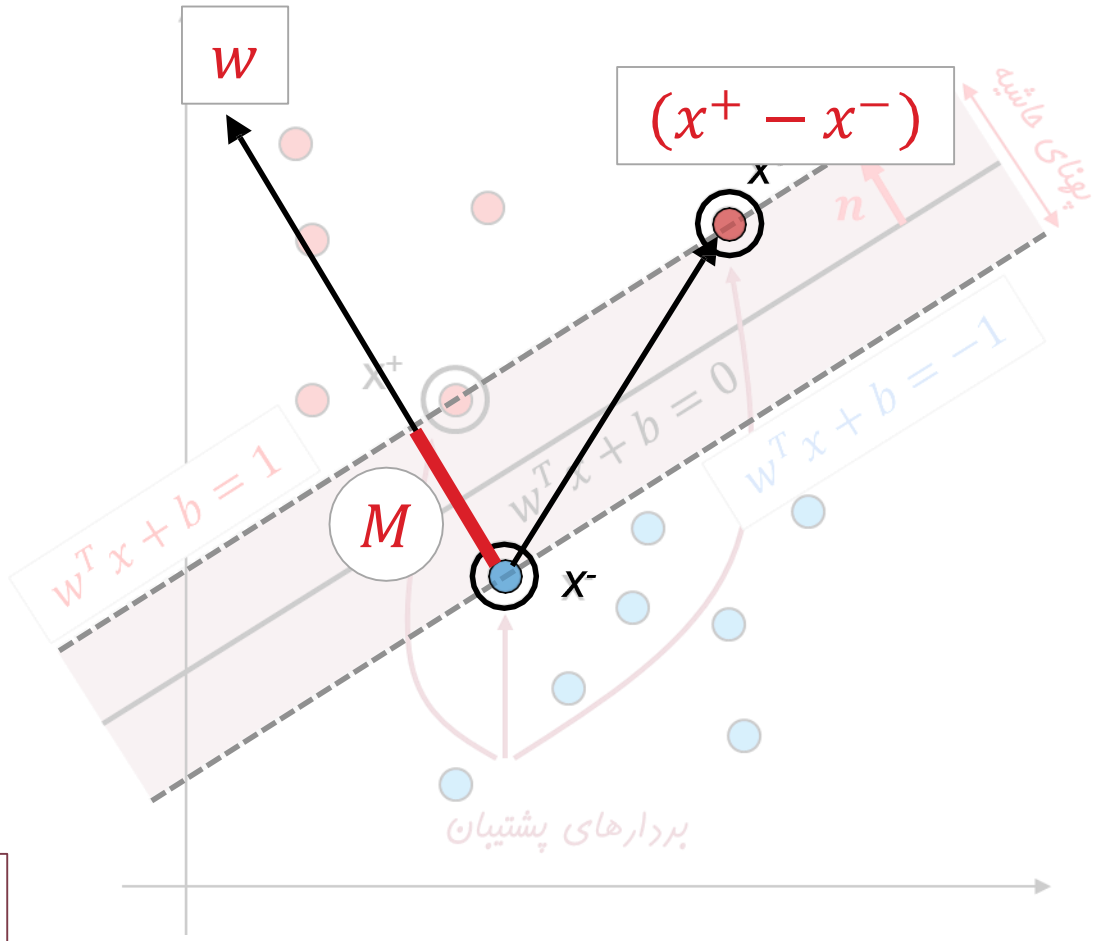
$$w^T x^- + b = -1$$

- Then :

$$w^T (x^+ - x^-) = 2$$

$$\Rightarrow \|w\| \cdot \|x^+ - x^-\| \cos\alpha = 2$$

$$\Rightarrow \|w\| \cdot M = 2 \quad \Rightarrow \quad \boxed{M = 2\mathrm{T}\|w\|}$$



Objective function

- Objective : Maximizing the margin size [the distance between support vectors and the decision boundary].

$$M = \frac{2}{\|w\|}$$

- Attention : To maximize the margin, one can minimize the size of the vector w.
- Constraints: The decision boundary must correctly separate the data points of both classes.

Objective function : Formal expression

- Objective function :

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$\text{s.t.} \quad (\boldsymbol{w}^T\boldsymbol{x}^t + b) \geq +1 \quad if\ y^t = +1$$

$$(\boldsymbol{w}^T\boldsymbol{x}^t + b) \leq -1 \quad if\ y^t = -1$$

- Simplifying :

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$\text{s.t.} \quad y^t(\boldsymbol{w}^T\boldsymbol{x}^t + b) \geq +1$$

Objective function : Formal expression

- Objective function :

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$\text{s.t.} \quad y^t(\boldsymbol{w}^T\boldsymbol{x}^t + b) \geq +1$$

- Solving the problem using Lagrange Multipliers :

$$L_p = \frac{1}{2}\|w\|^2 - \sum_{t=1}^{m} \alpha^t [y^t(w^T x^t + b) - 1]$$

$$= \frac{1}{2}\|w\|^2 - \sum_{t=1}^{m} \alpha^t y^t (w^T x^t + b) + \sum_{t=1}^{m} \alpha^t$$

Objective function : Formal expression

- Objective function :

$$L_p = \frac{1}{2}\|w\|^2 - \sum_{t=1}^{m} \alpha^t [y^t (w^T x^t + b) - 1]$$

$$= \frac{1}{2}\|w\|^2 - \sum_{t=1}^{m} \alpha^t y^t (w^T x^t + b) + \sum_{t=1}^{m} \alpha^t$$

The decision boundary is a linear combination of the training data.

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{t=1}^{m} \alpha^t y^t x^t$$

$$\frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{t=1}^{m} \alpha^t y^t = 0$$

Objective function : Dual form

- Objective function :

$$L_d = \frac{1}{2}(\mathrm{w}^{\mathrm{T}}\mathrm{w}) - \mathrm{w}^{\mathrm{T}}\sum_{t=1}^{m}\alpha^t y^t x^t - b\sum_{z=1}^{m}\alpha^t y^t + \sum_{t=1}^{m}\alpha^t$$

$$= \frac{1}{2}(w^T w) + \sum_{t=1}^{m}\alpha^t$$

$$= \frac{1}{2}\sum_{t=1}^{m}\sum_{s=1}^{m}\alpha^t\alpha^s y^t y^s (x^t)^T x^s + \sum_{t=1}^{m}\alpha^t$$

subject to $\sum_{t=1}^{m} a^t y^t = 0 \; and \; a^t \geqq 0 \; \forall t$

- Many of the alpha coefficients are equal to zero, and only a few have values greater than zero.
- The data points for which the alpha values are greater than zero are the support vectors.

Objective function : Vector form

- Objective function :

$$L_d = \frac{1}{2}\sum_{t=1}^{m}\sum_{s=1}^{m}\alpha^t\alpha^s y^t y^s (x^t)^T x^s + \sum_{t=1}^{m}\alpha^t$$

$$= -\frac{1}{2}\alpha^T Q\alpha + e^T\alpha$$

$Q_{ts} = y^t y^s (x^t)^T x^s,$ $e = [1 \quad 1 \quad \dots \quad 1]^T \in \mathbb{R}^m$

subject to $\sum_{t=1}^{m} a^t y^t = 0$ *and* $a^t \geqq 0\ \forall t$

- Many of the alpha coefficients are equal to zero, and only a few have values greater than zero.
- The data points for which the alpha values are greater than zero are the support vectors.

Non-linearly Separable Data: Soft Margin

- If the data is not linearly separable, what happens?
- Soft Margin: Allowing for a small margin of error in separation
- Soft Error
- New objective function

$$y^t(\boldsymbol{w}^T\boldsymbol{x}^t + b) \geq 1 - \varepsilon^t$$

$$soft\ error : \sum_{t=1}^{m} \varepsilon^t$$

Penalty coefficient

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{t=1}^{m}\varepsilon^t$$

$$\text{s.t.} \quad y^t(\boldsymbol{w}^T\boldsymbol{x}^t + b) \geq 1 - \varepsilon^t$$

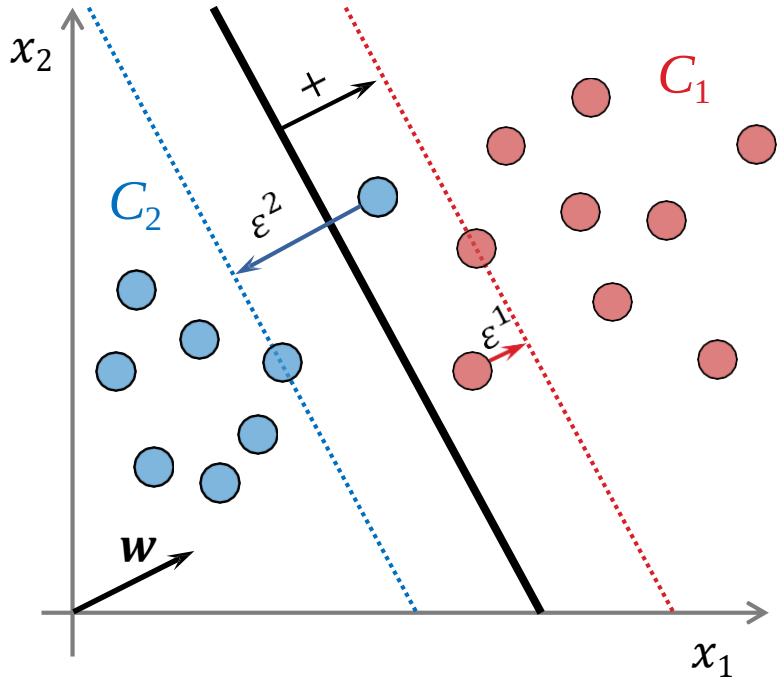
$$\varepsilon^t \geq 0$$

Non-linearly Separable Data: Soft Margin

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{t=1}^{m}\varepsilon^t$$

$$\text{s.t.} \quad y^t(\boldsymbol{w}^T\boldsymbol{x}^t + b) \geq 1 - \varepsilon^t$$

$$\varepsilon^t \geq 0$$



$$L_p = \frac{1}{2}\|w\|^2 + c\sum_{t=1}^{m}\varepsilon^t - \sum_{t=1}^{m}\alpha^t[y^t(w^T x^t + b) - 1 + \varepsilon^t]$$

$$- \sum_{t=1}^{m}\mu^t\varepsilon^t$$

Non-linearly Separable Data: Soft Margin

$$L_p = \frac{1}{2}\|w\|^2 + c\sum_{t=1}^{m}\varepsilon^t - \sum_{t=1}^{m}\alpha^t[y^t(w^T x^t + b) - 1 + \varepsilon^t] - \sum_{t=1}^{m}\mu^t\varepsilon^t$$

$$\frac{\partial L_p}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{t=1}^{m}\alpha^t y^t \boldsymbol{x}^t$$

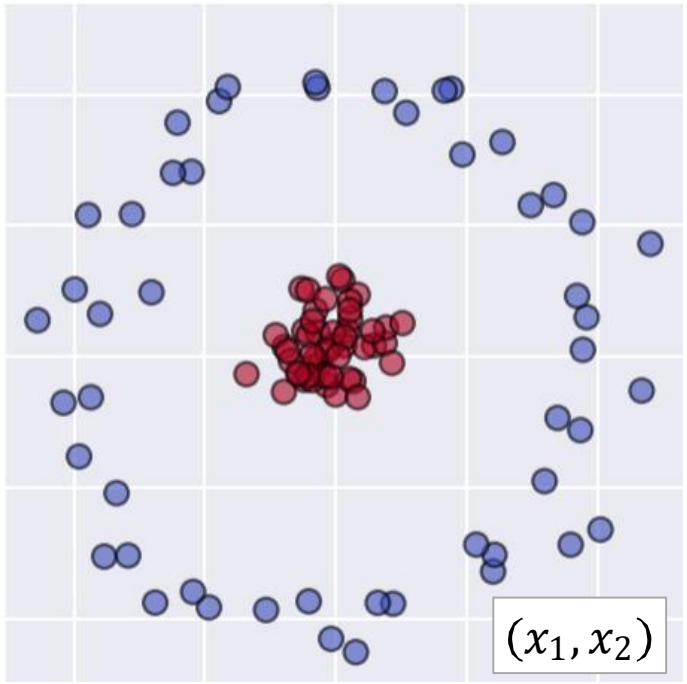
$$\frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{t=1}^{m}\alpha^t y^t = 0$$

$$\frac{\partial L_p}{\partial \varepsilon^t} = 0 \Rightarrow C - \alpha^t - \mu^t = 0 \Rightarrow 0 \le \alpha^t \le C$$

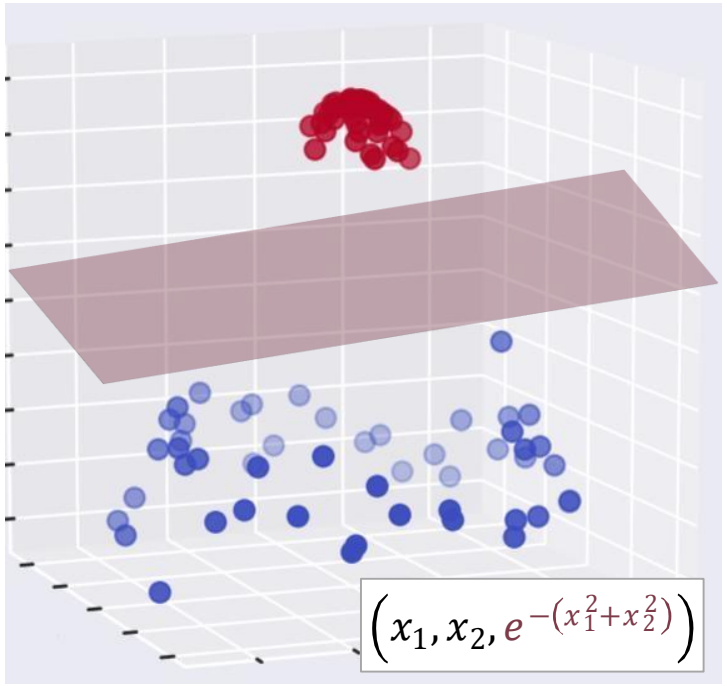
Kernel Trick and Non-Linear Classification

Kernel Functions

- Idea : Mapping the problem to a new feature space using non-linear transformations
  - Using a linear model in the new feature space for data classification.

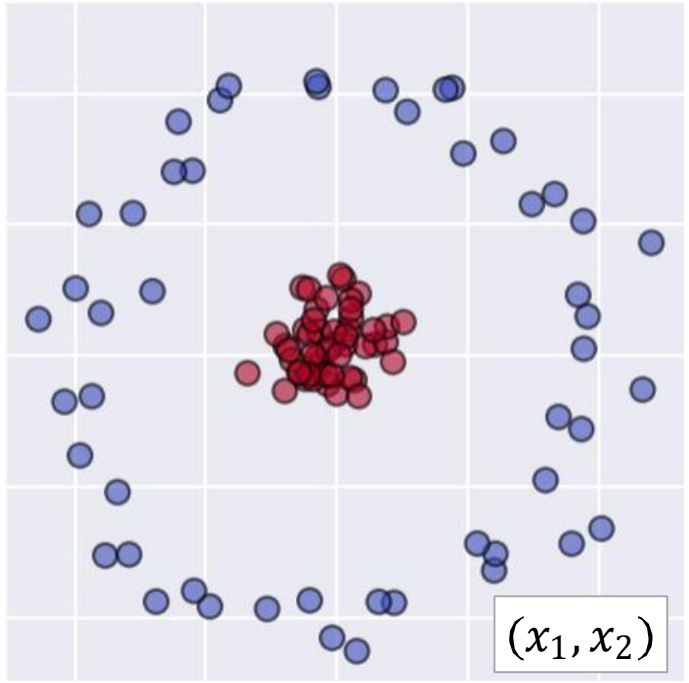


$\Phi: \boldsymbol{x} \rightarrow \varphi(\boldsymbol{x})$

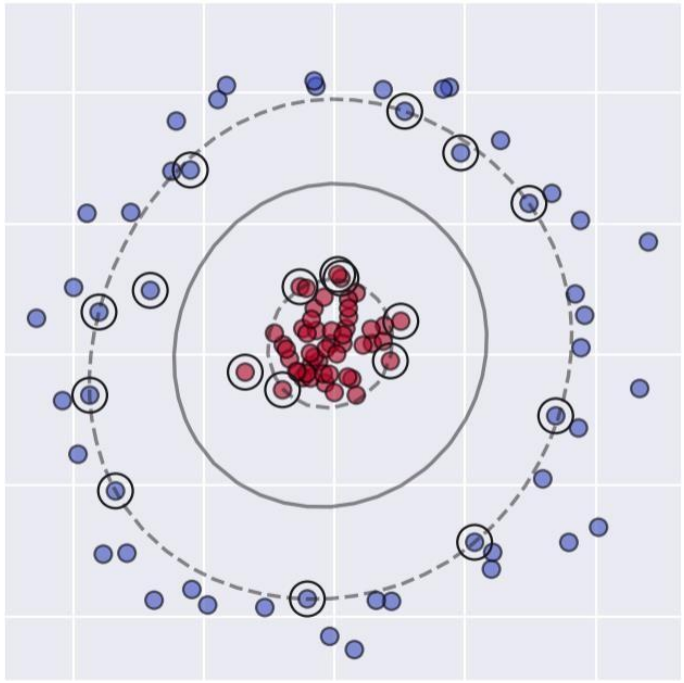


Kernel Functions

- Idea : Mapping the problem to a new feature space using non-linear transformations
  - Using a linear model in the new feature space for data classification.
  - A linear model in the new feature space corresponds to a non-linear model in the original space.

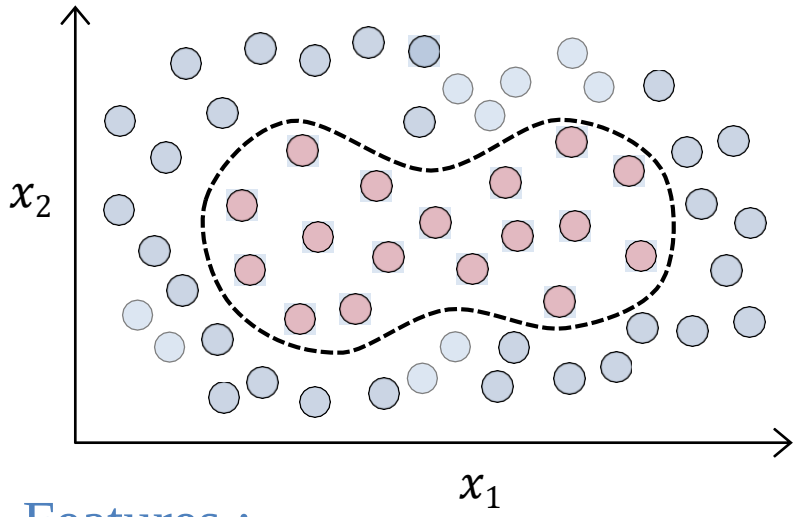


$\Phi: \boldsymbol{x} \rightarrow \varphi(\boldsymbol{x})$



Non-Linear Decision Boundary

- Prediction : $y = 1\ if$:



$$h(x) = b + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + \cdots \geq 0$$

- Features :

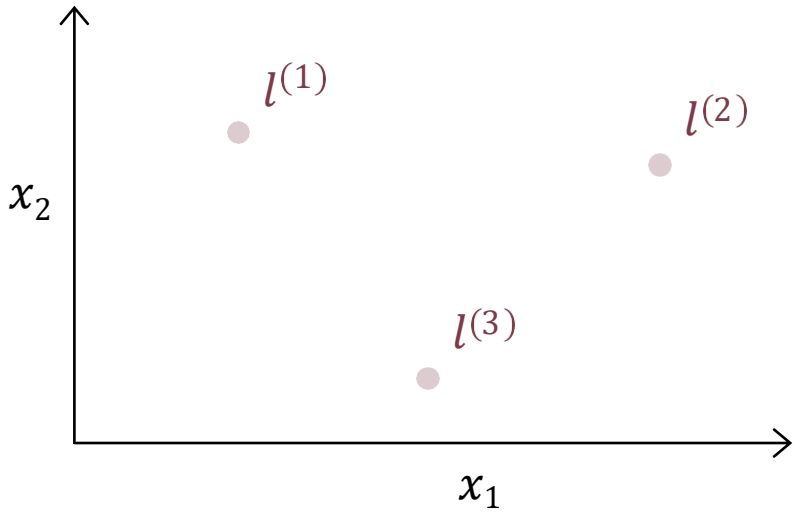
$$f_1 = x_1, \quad f_2 = x_2, \quad f_3 = x_1^2, \quad f_4 = x_2^2, \quad f_5 = x_1x_2, \quad \cdots$$

$$h(f) = b + w_1f_1 + w_2f_2 + w_3f_3 + w_4f_4 + w_5f_5 + \cdots$$ ⟵ Linear decision boundary

- Question : Is there a better method for feature selection?

Kernels as Similarity Measures

- Idea : Given x, select a new set of features based on its similarity to reference points using l^2, l^1, and l^3 norms.



$$f_1 = sim(x, l^{(1)}) = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$$

$$f_2 = sim(x, l^{(2)}) = \exp\left(-\frac{\|x - l^{(2)}\|^2}{2\sigma^2}\right)$$

$$f_3 = sim(x, l^{(3)}) = \exp\left(-\frac{\|x - l^{(3)}\|^2}{2\sigma^2}\right)$$

Kernel (Gaussian Kernel)

- Kernel Function: A measure for calculating the similarity between data x and y.

Kernels as Similarity Measures

- Kernel function :

$$f_i = sim(x, l^{(i)}) = \exp\left(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2}\right)$$

- First case : $x \approx l^{(i)}$

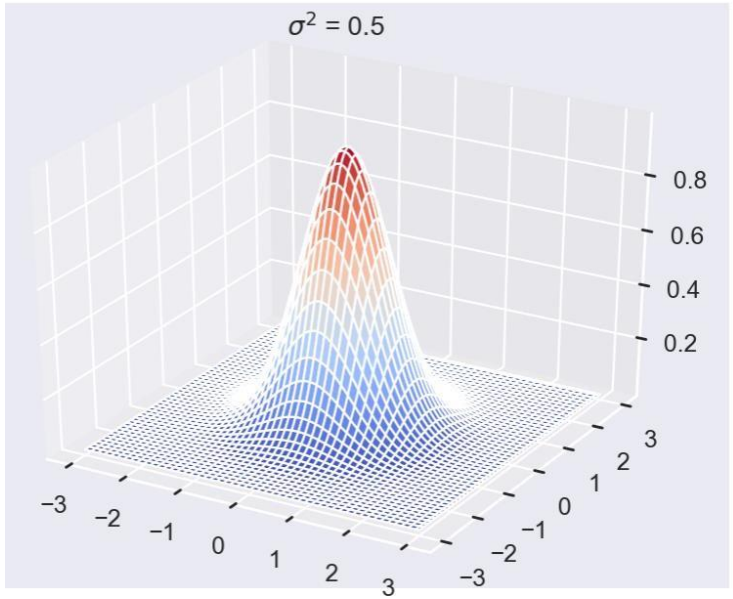
$$f_i \approx \exp\left(-\frac{0}{2\sigma^2}\right) = \exp(0) = 1$$

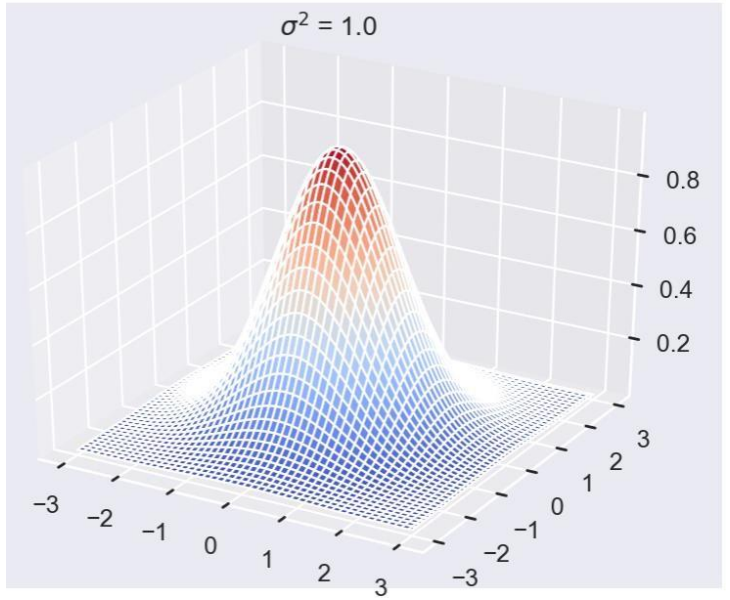
- Second case : x is very far from $l^{(i)}$.

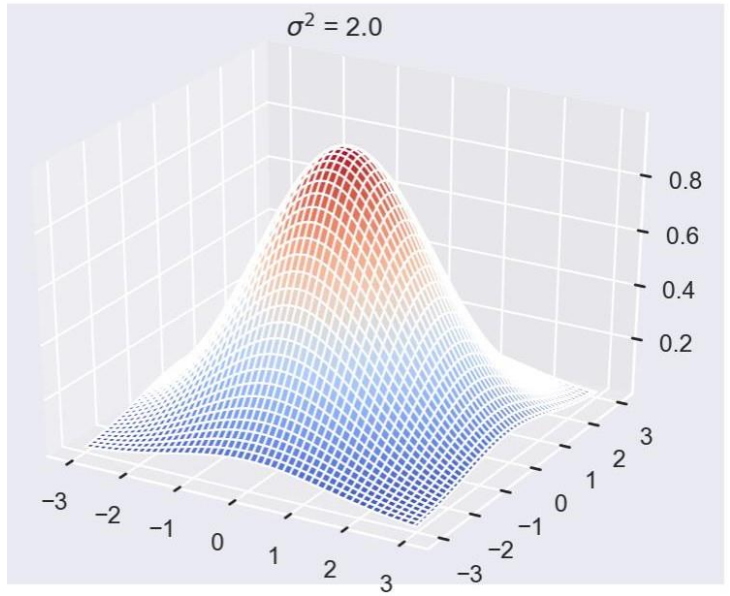
$$f_i \approx \exp\left(-\frac{\infty}{2\sigma^2}\right) = \exp(-\infty) = 0$$

Kernels as Similarity Measures

$$f_i = sim\left(x, l^{(i)}\right) = \exp\left(-\frac{\left\|x - l^{(i)}\right\|^2}{2\sigma^2}\right)$$

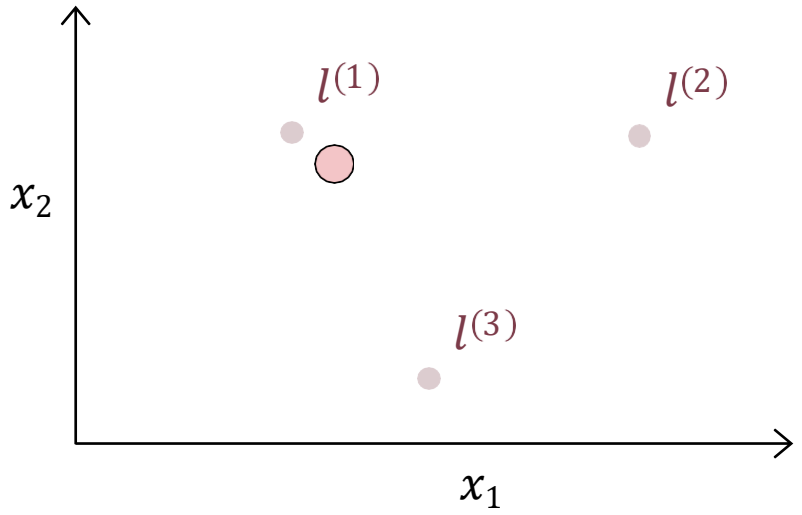






Kernels as Similarity Measures

- Prediction : $y = 1\ if$:



$$b + w_1 f_1 + w_2 f_2 + w_3 f_3 \geq 0$$

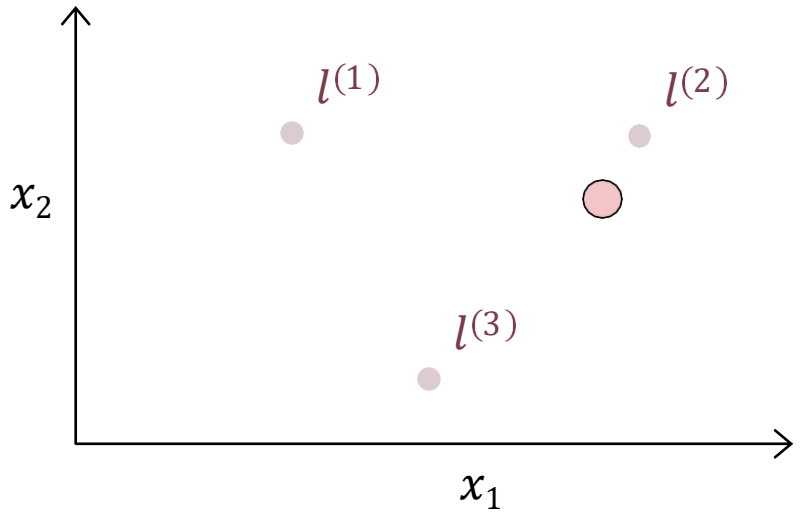
$-0.5 \quad 1.0 \quad 1.0 \quad 0.0$

$f_1 \approx 1, f_2 \approx f_3 \approx 0$

$$h(f) \approx -0.5 + (1.0)(1.0) + (1.0)(0.0) + (0.0)(0.0) = 0.5 \geq 0 \Rightarrow \boxed{y = 1}$$

Kernels as Similarity Measures

- Prediction : $y = 1\ if$:



$$b + w_1 f_1 + w_2 f_2 + w_3 f_3 \geq 0$$

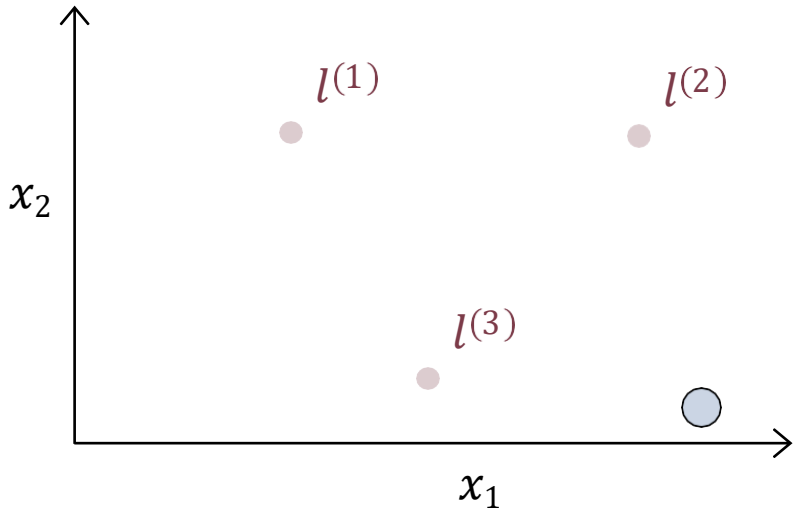
$-0.5 \quad 1.0 \quad 1.0 \quad 0.0$

$f_1 \approx f_3 \approx 0, f_2 \approx 1$

$$h(f) \approx -0.5 + (1.0)(0.0) + (1.0)(1.0) + (0.0)(0.0) = 0.5 \geq 0 \Rightarrow \boxed{y = 1}$$

Kernels as Similarity Measures

- Prediction : $y = 1\ if$:



$$b + w_1 f_1 + w_2 f_2 + w_3 f_3 \geq 0$$

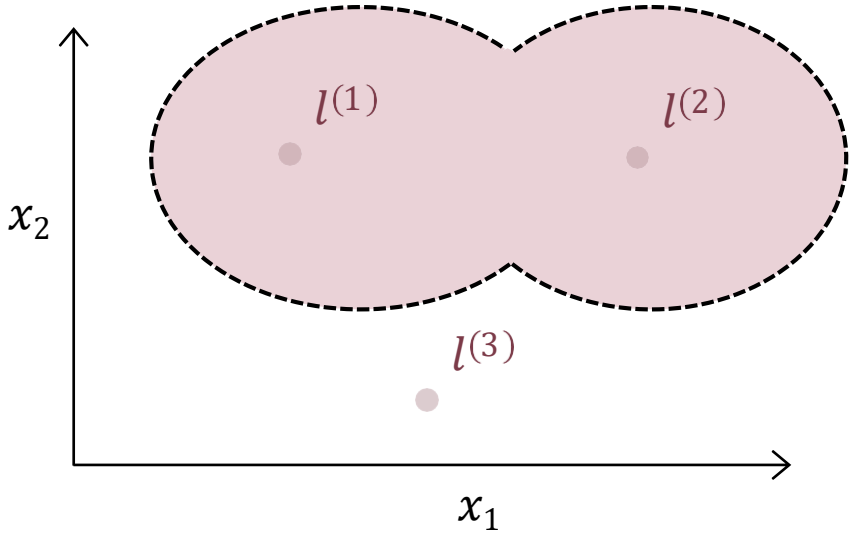
$-0.5 \quad 1.0 \quad 1.0 \quad 0.0$

$$f_1 \approx f_2 \approx f_3 \approx 0$$

$$h(f) \approx -0.5 + (1.0)(0.0) + (1.0)(0.0) + (0.0)(0.0) = -0.5 < 0 \quad \Rightarrow \quad \boxed{y = 0}$$

Kernels as Similarity Measures

- Prediction : $y = 1\ if$:



$$b + w_1 f_1 + w_2 f_2 + w_3 f_3 \geq 0$$

-0.5 1.0 1.0 0.0

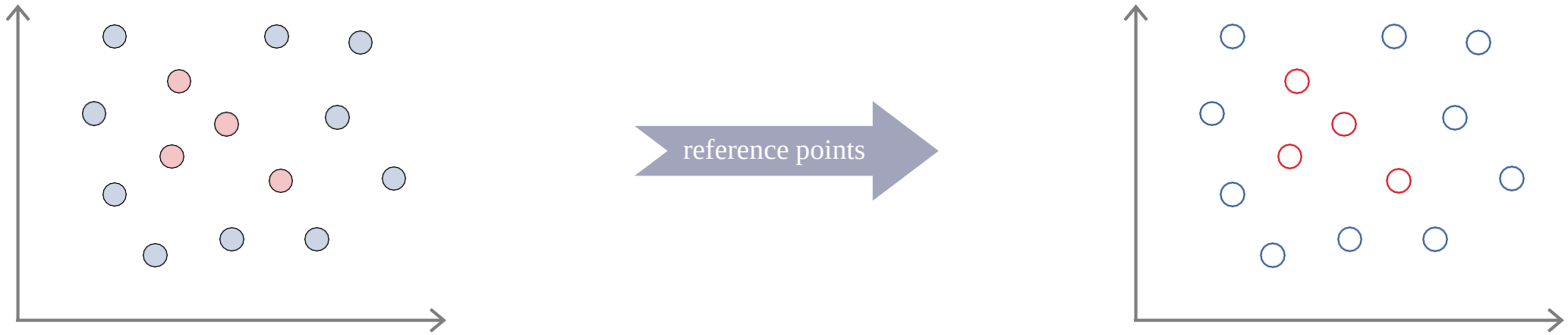
- Decision Boundary : Classifies points near $l^{(1)}$ and $l^{(2)}$ as Class 1, and other points as Class 0.

Several Questions

- How does the learning algorithm automatically select reference points?
- How are appropriate values for kernel function parameters determined?
- Are there other types of kernels?

Select Reference Points

- How does the learning algorithm automatically select reference points?
  - For each sample in the training set, a reference point is chosen equal to that sample.



Feature Mapping

- Training set :

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$$

- Reference points :

$$l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \cdots, l^{(m)} = x^{(m)}$$

- Mapping feature space :

$$x = \begin{bmatrix} x_0 = 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$f = \begin{bmatrix} f_0 = 1 \\ f_1 = K(x, l^{(1)}) \\ f_2 = K(x, l^{(2)}) \\ \vdots \\ f_m = K(x, l^{(m)}) \end{bmatrix}$$

Kernel Trick

- Kernel Function : Preprocessing data x using kernel functions.

$$z = \varphi(x) = (\varphi_1(x), \varphi_2(x), \dots, \varphi_k(x))$$

It may be infinite!

$$g(z) = w^T z + b$$

$$g(x) = w^T \varphi(x) + b$$

- Decision boundary :

$$w = \sum_{t=1}^{m} a^t y^t z^t = \sum_{t=1}^{m} a^t y^t \varphi(x^t)$$

- Classification of new data :

$$g(x) = w^T \varphi(x) + b = \left(\sum_{t=1}^{m} a^t y^t \varphi(x^t)^T \right) = \left(\sum_{t=1}^{m} a^t y^t \varphi(x^t)^T \varphi(x) \right) + b = \left(\sum_{t=1}^{m} a^t y^t k(x^t, x) \right) + b$$

Kernel Function

$$L_p = \frac{1}{2}\|W\|^2 + C\sum_{t=1}^{m}\varepsilon^t$$

s.t. $\quad y^t W^t \varphi(x^t) \geq 1 - \varepsilon^t$

$$\varepsilon^t \geq 0$$

$$L_p = \frac{1}{2}\|W\|^2 + C\sum_{t=1}^{m}\varepsilon^t - \sum_{t=1}^{m}\alpha^t y^t [W^T \varphi(x^t) - 1 + \varepsilon^t] - \sum_{t=1}^{m}\mu^t \varepsilon^t$$

Lagrange Coefficients

Lagrange Coefficients

Kernel Functions: The Primary Problem

$$L_p = \frac{1}{2}\|W\|^2 + C\sum_{t=1}^{m}\varepsilon^t - \sum_{t=1}^{m}\alpha^t y^t [W^T \varphi(x^t) - 1 + \varepsilon^t] - \sum_{t=1}^{m}\mu^t \varepsilon^t$$

$$\frac{\partial L_p}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{t=1}^{m} \alpha^t y^t \varphi(\boldsymbol{x}^t)$$

$$\frac{\partial L_p}{\partial \varepsilon^t} = 0 \Rightarrow C - \alpha^t - \mu^t = 0 \Rightarrow 0 \leq \alpha^t \leq C$$

Kernel Functions: The Dual Problem

$$L_d = -\frac{1}{2}\sum_{t=1}^{m}\sum_{s=1}^{m} a^t a^s y^t y^s \varphi(x^t)^T \varphi(x^s) + \sum_{t=1}^{m} a^t$$

subject to $\sum_{t=1}^{m} \alpha^t y^t = 0$ and $0 \le \alpha^t \le C \ \forall t$

- Idea of Kernel Machines : [Kernel Trick]
  - Replacing the dot product of basis functions with a kernel function as $K(x^t, x^s)$

$$L_d = -\frac{1}{2}\sum_{t=1}^{m}\sum_{s=1}^{m} a^t a^s y^t y^s K(x^t, x^s) + \sum_{t=1}^{m} a^t$$

Gram Matrix: A symmetric positive definite matrix (for linear separability).

Kernel Functions: Polynomial Kernel

- Polynomial Kernel : A polynomial of degree q

$$K(x^t, x) = (x^T x^t + 1)^q$$

- Example : $[q = 2, d = 2]$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$K(x, y) = (x^T y + 1)^2$$

$$= (x_1 y_1 + x_2 y_2 + 1)^2$$

$$= 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 + x_1^2 y_1^2 + x_2^2 y_2^2$$

3 multiplications,

2 additions

$$\varphi(x) = \left[1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2\right]^T$$

$$\varphi(y) = \left[1, \sqrt{2}y_1, \sqrt{2}y_2, \sqrt{2}y_1 y_2, y_1^2, y_2^2\right]^T$$

6 multiplications,

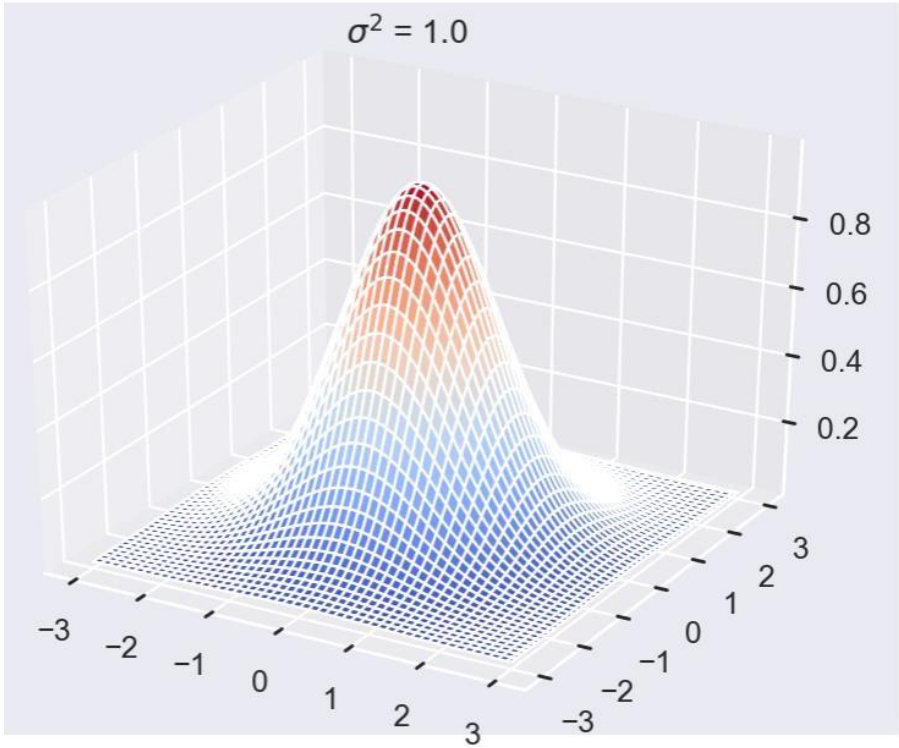
5 additions

Kernel Functions: Gaussian Kernel

- Gaussian kernel :

$$K(\boldsymbol{x}^t, \boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x}^t - \boldsymbol{x}\|^2}{2\sigma^2}\right)$$

- Finding an appropriate value for σ
  - Using a validation set [Model selection]
  - Larger values : Smoother decision boundary

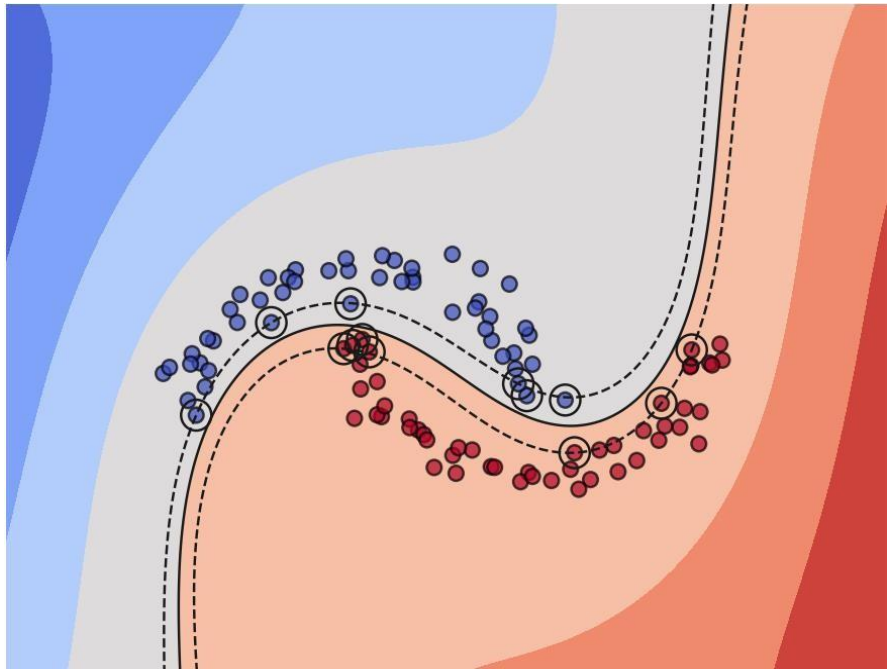


Kernel Functions: Gaussian Kernel

- Gaussian kernel :

$$K(\boldsymbol{x}^t, \boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x}^t - \boldsymbol{x}\|^2}{2\sigma^2}\right)$$

- Finding an appropriate value for σ
  - Using a validation set [Model selection]
  - Larger values : Smoother decision boundary



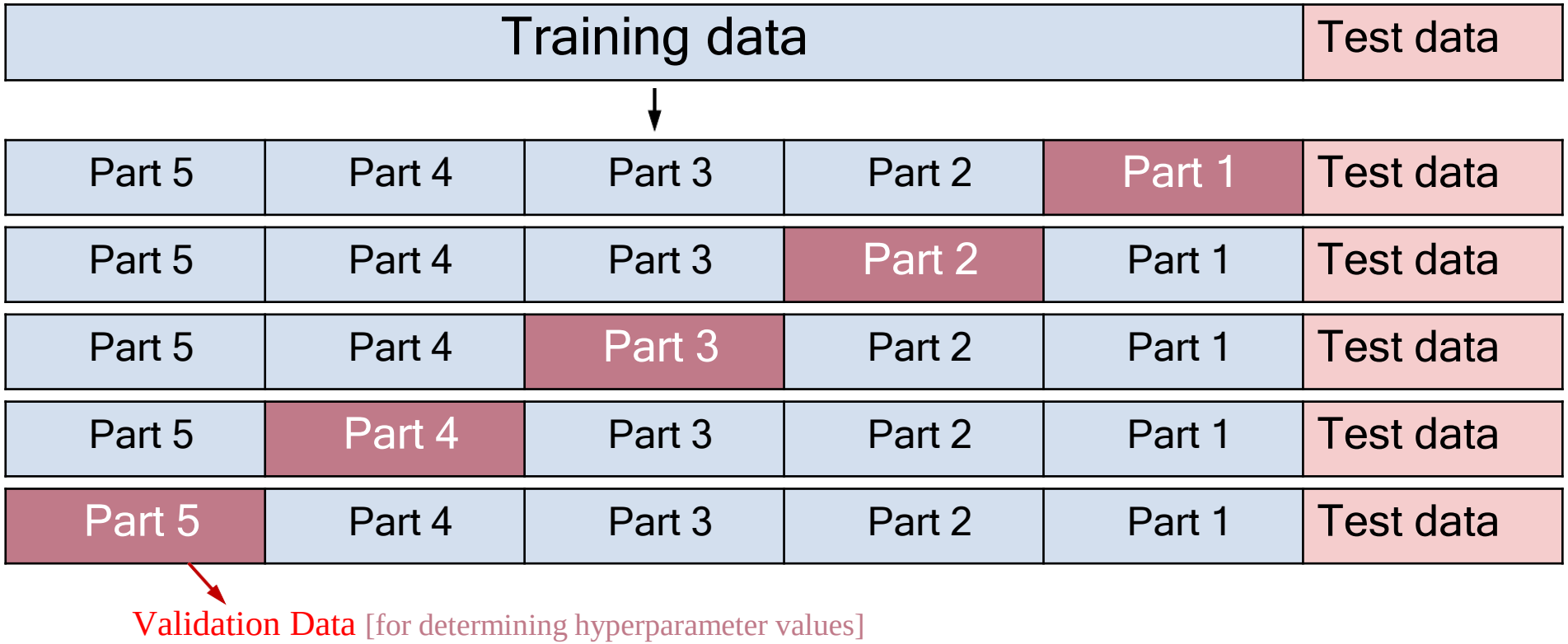
Hyperparameters: Model Selection

- First Solution: [A very bad idea]
  - Selecting a value that results in the highest classification accuracy on the test dataset

- Attention : [Very important]
  - Use the test set at the end of the process and only for estimating generalization capability of the classifier

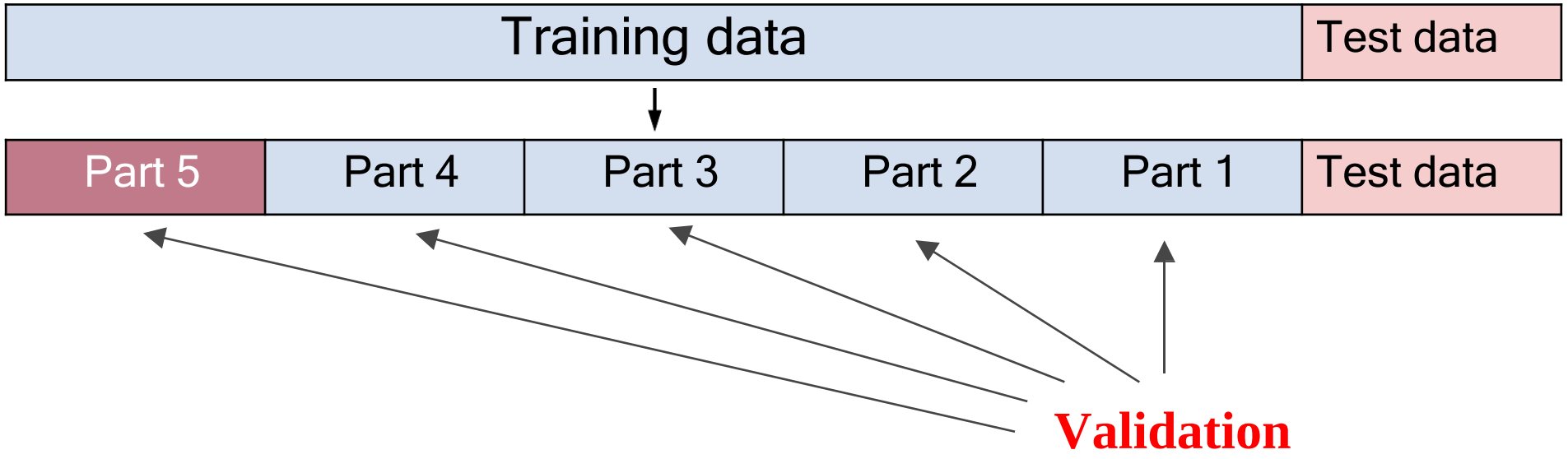
Hyperparameters: Model Selection

- Second Solution: [Cross-Validation]



Hyperparameters: Model Selection

- Second Solution: [Cross-Validation]



- Select one portion as validation data each time and then take the average of the obtained results.

Hyperparameters: Model Selection

Splitting training data into training and validation data.

```
cv = StratifiedKFold(n_splits=5, shuffle=True)

C_range = np.logspace(-3, 5, 9)
gamma_range = np.logspace(-3, 5, 9)

pgrid = dict(gamma=gamma_range, C=C_range)
grid = GridSearchCV(SVC(), param_grid=pgrid, cv=cv)

grid.fit(X, y)
```

Hyperparameters: Model Selection

Determining the search range for hyperparameter tuning.

```
cv = StratifiedKFold(n_splits=5, shuffle=True)

C_range = np.logspace(-3, 5, 9)
gamma_range = np.logspace(-3, 5, 9)

pgrid = dict(gamma=gamma_range, C=C_range)
grid = GridSearchCV(SVC(), param_grid=pgrid, cv=cv)

grid.fit(X, y)
```

$\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}\}$

Hyperparameters: Model Selection

Creating a classifier

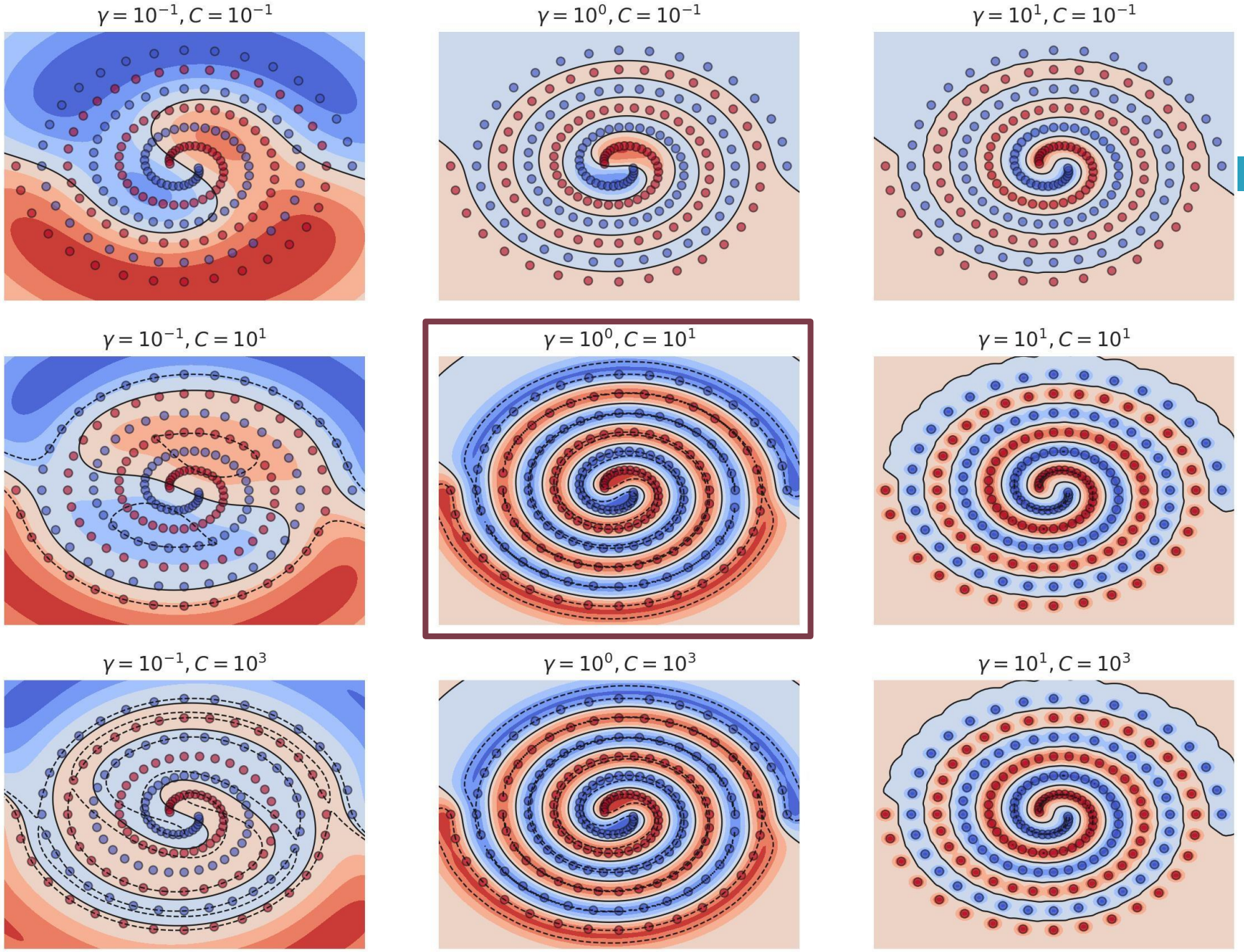
```
cv = StratifiedKFold(n_splits=5, shuffle=True)

C_range = np.logspace(-3, 5, 9)
gamma_range = np.logspace(-3, 5, 9)

pgrid = dict(gamma=gamma_range, C=C_range)
grid = GridSearchCV(SVC(), param_grid=pgrid, cv=cv)

grid.fit(X, y)
```

Hyperparameters: Model Selection


```
cv = StratifiedKFold(n_splits=5, shuffle=True)

C_range = np.logspace(-3, 5, 9)
gamma_range = np.logspace(-3, 5, 9)

pgrid = dict(gamma=gamma_range, C=C_range)
grid = GridSearchCV(SVC(), param_grid=pgrid, cv=cv)

grid.fit(X, y)
```

Training the classifier

Grid Search

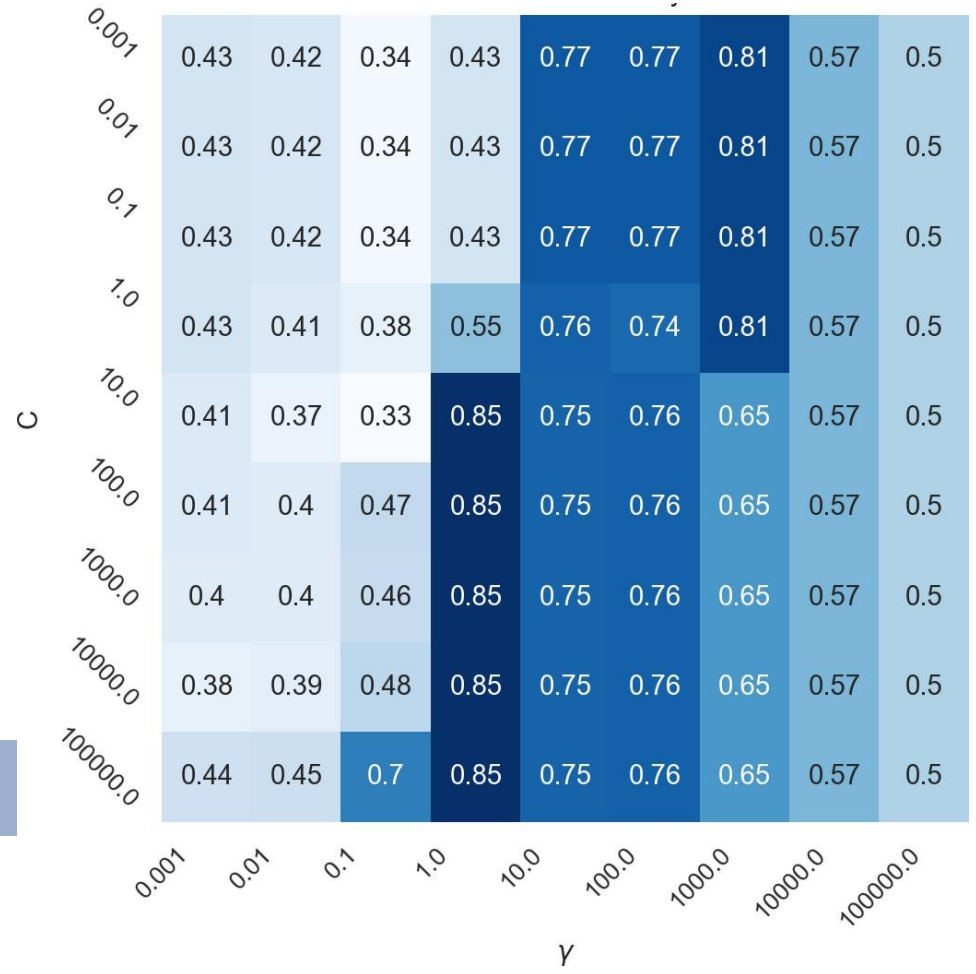

$\gamma = 10^{-1}, C = 10^{-1}$
$\gamma = 10^{0}, C = 10^{-1}$
$\gamma = 10^{1}, C = 10^{-1}$
$\gamma = 10^{-1}, C = 10^{1}$
$\gamma = 10^{0}, C = 10^{1}$
$\gamma = 10^{1}, C = 10^{1}$
$\gamma = 10^{-1}, C = 10^{3}$
$\gamma = 10^{0}, C = 10^{3}$
$\gamma = 10^{1}, C = 10^{3}$
$C = 10^{-1}$
$C = 10^{1}$
$C = 10^{3}$
$\gamma = 10^{-1}$
$\gamma = 10^{0}$
$\gamma = 10^{1}$

Support Vector Machine Parameters

- Question : How are the appropriate values for the kernel function parameters determined?

- Parameter C :
  - Smaller values: Higher bias, lower variance.
  - Larger values: Lower bias, higher variance.

- Parameter σ :
  - Smaller values: Lower bias, higher variance.
  - Larger values: Higher bias, lower variance.



Classification accuracy on validation data

Guide to Using Support Vector Machines

- Implementation: Using existing software packages like SVM^{light} and LIBSVM

- Determining the kernel function :
  - Linear kernel (no kernel): When $m \gg n$.
  - Gaussian, polynomial, string, and ...

- Determining parameter values : Grid search
  - Selecting a value for parameter C.
  - Selecting values for kernel function parameters (like σ).

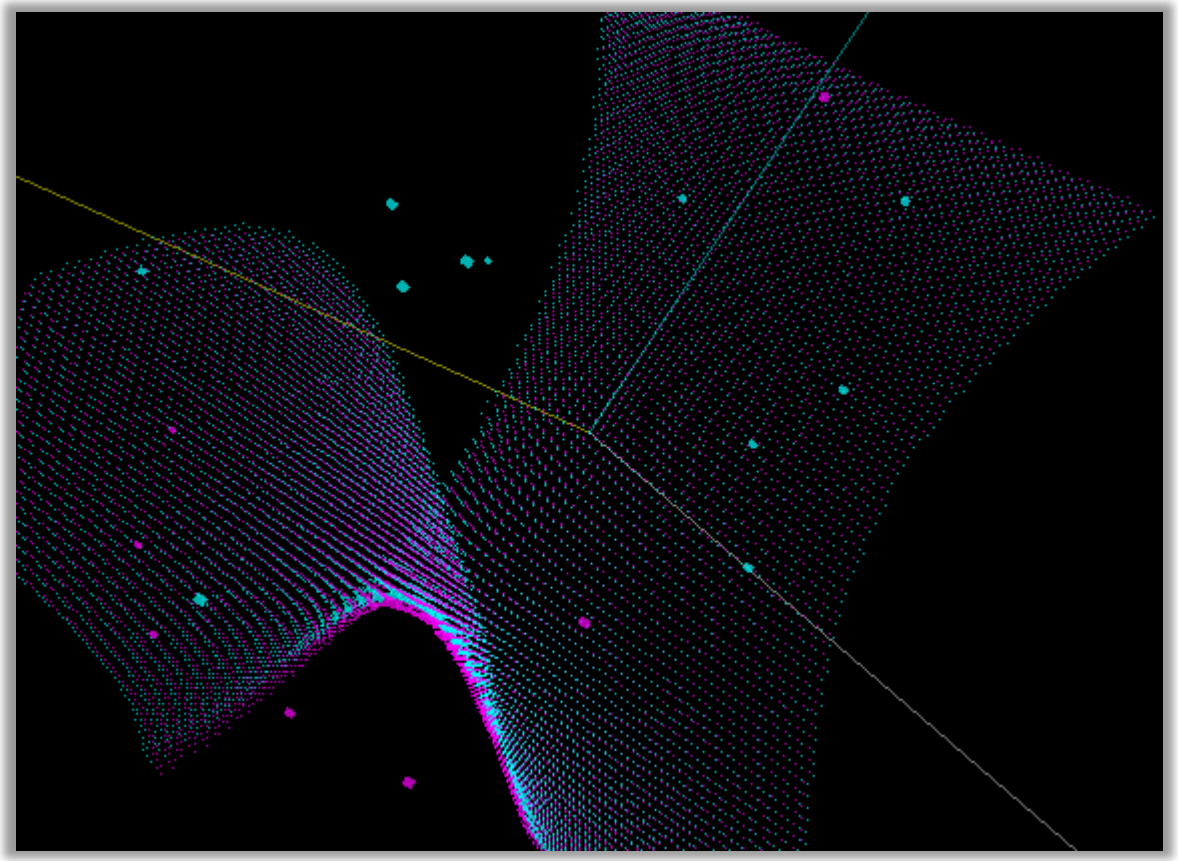
Support Vector Machine, Neural Network, or Logistic Regression

- First Scenario : $[n \gg m]$
  - Example: Spam detection (1000 training samples, 50,000 features).
  - Logistic Regression or Linear Support Vector Machine

- Second Scenario : [Few features, a large number of training samples]
  - Support Vector Machine with Gaussian Kernel.

Attention: Neural networks can be used in all the above scenarios, but they may require more training time.

Demo Execution



Appendix: More on Kernels

- Question: How do we know that using kernels helps us in data separation?
  - In an n-dimensional space, any set of n independent vectors is linearly separable.
  - If the matrix K is positive definite, then the data is linearly separable.
- Theorem: The matrix K is positive definite because $L^{\mathrm{T}}L = K$.
  - The i-th column in matrix L is equal to the vector $\phi(\mathrm{x}^{(\mathrm{i})})$.
- Proof : Consider a non-zero vector v. In this case:

$$v^T K v = v^T L^T L v = (Lv)^T (Lv) = w^T w = \|w\|^2 \geq 0$$

And since both L and v are non-zero, the vector w is also non-zero. That is :

$$\|w\|^2 > 0 \Rightarrow v^T K v > 0 \;\Rightarrow K \; is \; positive \; definite$$