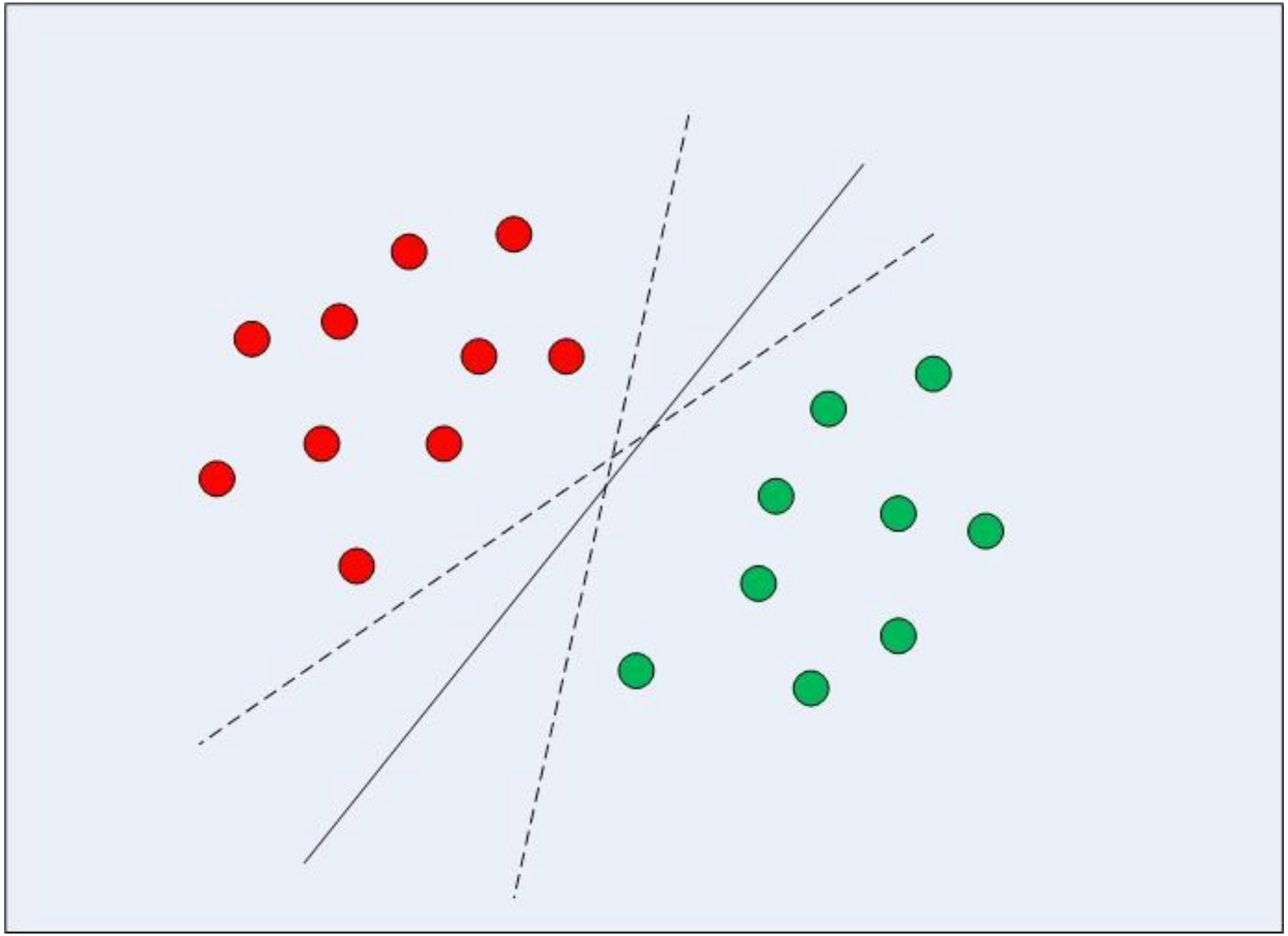
CS 559: Machine Learning Fundamentals & Applications

Lecture 8: Supportive Vector Machine

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®
1870

8.1. Motivation

8.1. Motivation



8.1. Motivation

Recall the perceptron algorithm.

1. It **guarantees** to find a solution in a finite number of steps.
2. The solution of parameters depends on the *initially chosen values*.

- There can be multiple solutions all of which classify the training data set perfectly.
- Then what is the best model among them?
- We want to select the one that gives the **smallest generalization error**.

8.2. Supportive Vector Machine

8.2. Supportive Vector Machine

- SVM was invented in 1963 by Vapnik and Chervonenkis to solve a classification problem.
- It was extended to solve nonlinear problems by applying *kernel methods* in 1992.
- Then, it became one of the popular classifiers (and regressions).

How does it work?

- It determines the **maximum margin** (the smallest perpendicular distance between the decision boundary and the closest data points) to a convex optimization problem.
- The solution is **globally** optimized.

8.2. Supportive Vector Machine

- Let the training data set compromises $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ with corresponding target values $\boldsymbol{t} = \{t_1, \dots, t_N\}$ where $t_n \in \{-1,1\}$.
- Consider a linear binary classifier

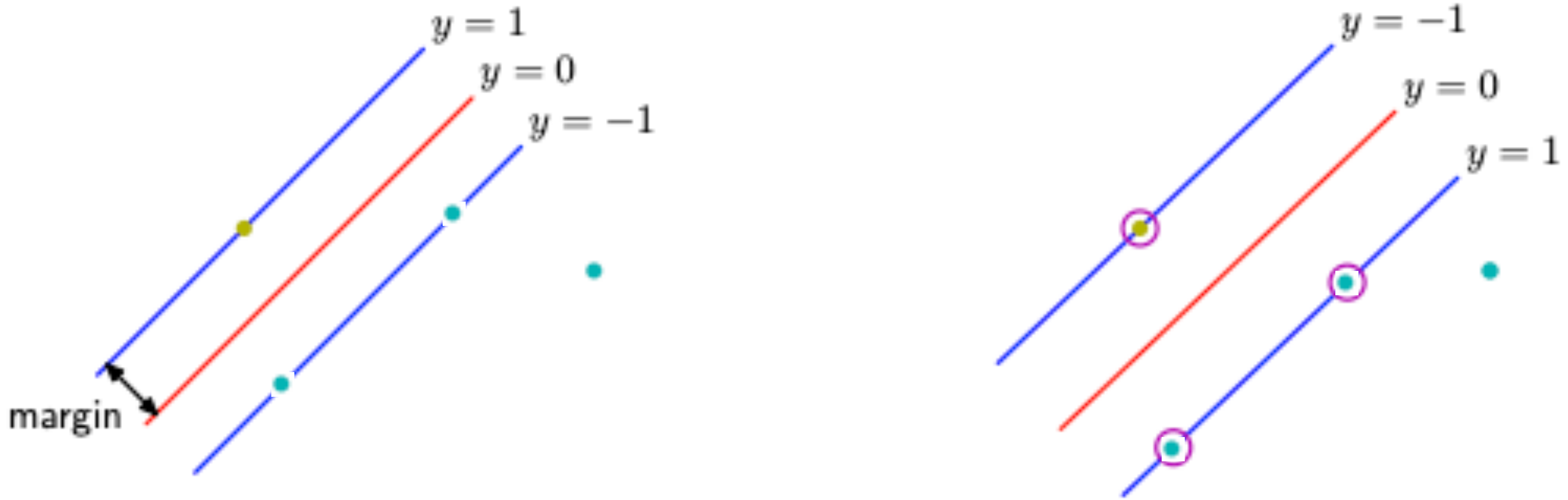
$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b.$$

- If the training data set is **linearly separable**, the solution for $\boldsymbol{w}$ and b is found that satisfies

$$t_n = \begin{cases} +1 & \text{if } y(\boldsymbol{x}) > 0 \\ -1 & \text{if } y(\boldsymbol{x}) < 0 \end{cases}.$$

8.2. Supportive Vector Machine

- In SVM, the decision boundary (hyperplane) is chosen for one with ***maximized margin***.
- The hyperplane is determined by the subset of data points that lies on the margin and they are called **support vectors** (SVs).



8.2. Supportive Vector Machine

- We are only interested only in SVs.
- For points that are correctly classified, $t_n y(\boldsymbol{x}_n) > 0$ for all n, the distance of a point $\boldsymbol{x}_n$ to the hyperplane is

$$\frac{t_n y(\boldsymbol{x}_n)}{||\boldsymbol{w}||} = \frac{t_n(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n) + b)}{||\boldsymbol{w}||}. \qquad \textit{Eq. 8 - 1}$$

- The ***maximized margin*** is found by solving

$$\underset{\boldsymbol{w},b}{\operatorname{argmax}} \left\{ \frac{1}{||\boldsymbol{w}||} \min_n [t_n(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n) + b)] \right\} \qquad \textit{Eq. 8 - 2}$$

where the factor $\frac{1}{||\boldsymbol{w}||}$ is taken outside of the optimization over n since $\boldsymbol{w}$ is independent of n.

8.2. Supportive Vector Machine

- Since for all correctly classified points will satisfy

$$t_n(w^T\phi(x_n)+b) \geq 1,$$

we can use all points that is **closest to the surface**

$$t_n(w^T\phi(x_n)+b) = 1.$$

- This simplifies the requirement that we maximize $||\boldsymbol{w}||^{-1}$ which is equivalent to minimizing $||\boldsymbol{w}||^{2}$ that is to solve the optimization

$$\underset{\boldsymbol{w},b}{\text{argmax}}\frac{1}{2}||\boldsymbol{w}||^2. \quad \textit{Eq. 8 - 3}$$

8.3. Lagrange Multipliers

8.3. Lagrange Multipliers

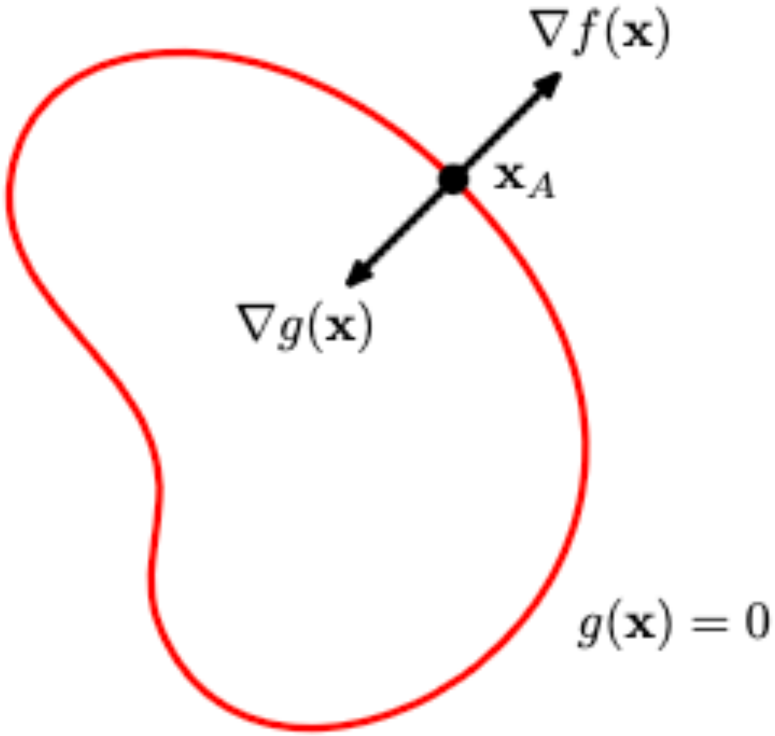
- In order to solve this, we need to introduce ***Lagrange multipliers***.
- Lagrange multipliers, also known as *undetermined multipliers*, are used to find the ***stationary points*** of a function of several variables subject to one or more constraints.
- Consider the problem of finding the maximum of a function $f(x_1, x_2)$ subject to a constraint relating x_1 and x_2, in form of

$$g(x_1, x_2) = 0.$$

- The analytical approach is not easy. Hence, **we can approach geometrically**.

8.3. Lagrange Multipliers

- Consider a D-dimensional variable $\boldsymbol{x}$ with components $x_1, \dots, x_D$.
- The constrain equation $\boldsymbol{g(x) = 0}$ represents a *(D-1)*-dimensional surface in $\boldsymbol{x}$-space.
- At any point on the constraint surface, $\nabla g(\boldsymbol{x})$ will be **orthogonal** to the surface. Consider a point $\boldsymbol{x}$ lies on the constraint surface and a nearby point $\boldsymbol{x} + \boldsymbol{\epsilon}$ lies on the surface. A Taylor expansion around $\boldsymbol{x}$ gives
$$g(\boldsymbol{x} + \boldsymbol{\epsilon}) \approx g(\boldsymbol{x}) + \boldsymbol{\epsilon}^T \nabla g(\boldsymbol{x}).$$
- In the limit $||\boldsymbol{\epsilon}|| \to 0$, we have $\boldsymbol{\epsilon}^T \nabla g(\boldsymbol{x}) = 0$ is parallel to the constrain surface where $g(\boldsymbol{x}) = 0$, ∇g is normal to the surface.
- If there is a point $\boldsymbol{x}^*$ maximizes $f(\boldsymbol{x})$, then the gradient of $f(\boldsymbol{x})$ also should be **orthogonal to the surface** in the opposite direction.



8.3. Lagrange Multipliers

- Since ∇f and ∇g are *anti-parallel*, there must be a non-zero Lagrange multiplier λ exist such that

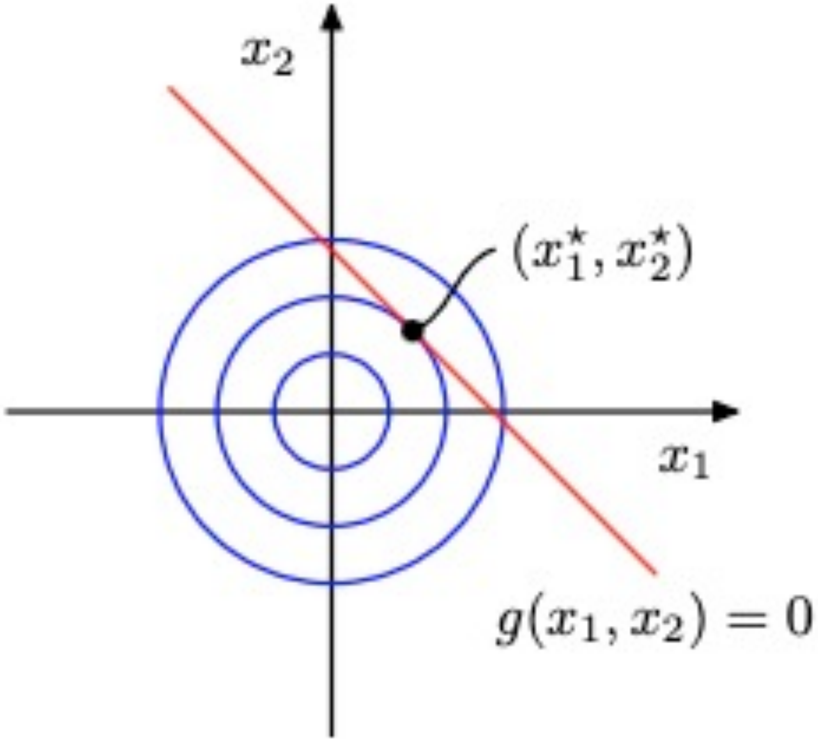
$$\nabla f + \lambda \nabla g = 0. \qquad \textit{Eq. 8 - 4}$$

- The *Lagrangian function*, $L(\boldsymbol{x}, \lambda)$, is then defined as

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x}). \qquad \textit{Eq. 8 - 5}$$

- By setting $\nabla_x L = 0$, the constrained condition Eq. 8-4 can be obtained.

8.3. Lagrange Multipliers



- Suppose we want to find the stationary point of
$$f(x_1, x_2) = 1 - x_1^2 - x_2^2$$
subject to the constraint $g(x_1, x_2) = x_1 + x_2 - 1 = 0$.
- The corresponding Lagrangian function is
$$L(x, \lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1).$$
- The conditions to be stationary w.r.t. x_1, x_2, and λ give the following equations:
$$-2x_1 + \lambda = 0$$
$$-2x_2 + \lambda = 0$$
$$x_1 + x_2 - 1 = 0.$$
- The solution of the stationary point $(x_1^*, x_2^*) = \left(\frac{1}{2}, \frac{1}{2}\right)$ for $\lambda = 1$.

8.3. Lagrange Multipliers

Two possible solutions can be obtained:

- ***Inactive*** **solution** – the constrained stationary point lies in the region $g(\boldsymbol{x}) > 0$.
  - The function $g(\boldsymbol{x})$ does not play a role, and the stationary condition is $\nabla f(\boldsymbol{x}) = 0$ and $\lambda = 0$.
- ***Active*** **solution** – the constrained stationary point lies on the boundary $g(\boldsymbol{x}) = 0$.
  - The sign $(+|-)$ of λ is crucial since $f(\boldsymbol{x})$ will be at a maximum if $\nabla f(\boldsymbol{x}) = -\lambda \nabla g(\boldsymbol{x})$ for some $\lambda > 0$.

8.3. Lagrange Multipliers

- Either case, $\lambda g(\boldsymbol{x}) = 0$.

- Thus, the solution to the problem of maximizing $f(\boldsymbol{x})$ subject to $g(\boldsymbol{x}) \geq 0$ is obtained by optimizing Eq. 8-5, $L = f + \nabla g$, w.r.t. $\boldsymbol{x}$ and λ:

$$g(\boldsymbol{x}) \geq 0$$

$$\lambda \geq 0$$

$$\lambda g(\boldsymbol{x}) = 0.$$

- This solution is also known as the ***Karush-Kuhn-Tucker*** *(KKT)* condition.

- If $f(x)$ is to be maximized $f(\boldsymbol{x})$ subject to $g_j(\boldsymbol{x}) = 0$ for $j = 1, \ldots, J$, and $h_k(\boldsymbol{x}) \geq 0$ for $k = 1, \ldots, K$, the Lagrangian function is

$$L\left(\boldsymbol{x}, \{\lambda_j\}, \{\mu_k\}\right) = f(\boldsymbol{x}) + \sum_{j=1}^{J} \lambda_j g_j(\boldsymbol{x}) + \sum_{k=1}^{K} \mu_k h_k(\boldsymbol{x}).$$

8.3. Lagrange Multipliers

- To optimize the solution for Eq. 8-3, the Lagrangian function can be form as

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{n=1}^{N} a_n\{t_n(\boldsymbol{w}^T\phi(\boldsymbol{x}_n) + b) - 1\} \qquad \textit{Eq. 8 - 6}$$

where the Lagrange multipliers $\boldsymbol{a} = (a_1, \dots, a_N)^T$ with $a_n \geq 0$.

- We are going to minimize Eq. 8-6 w.r.t. $\boldsymbol{w}$ and b, and maximize w.r.t. $\boldsymbol{a}$.

8.3. Lagrange Multipliers

- Setting $\nabla_{\boldsymbol{w},b} L = 0$ will have two conditions

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a})}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{n=1}^{N} a_n t_n \phi(\boldsymbol{x}_n) = 0 \rightarrow \boldsymbol{w} = \sum_{n=1}^{N} a_n t_n \phi(\boldsymbol{x}_n) \qquad \text{Eq. 8 - 7}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a})}{\partial b} = \sum_{n=1}^{N} a_n t_n = 0 \qquad \text{Eq. 8 - 8}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{a})}{\partial \boldsymbol{a}} = -\sum_{n=1}^{N} \{t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b) - 1\} = 0 \rightarrow b = t_n - w^T \phi(x_n)$$

8.4. Dual Representation in SVM

8.4. Dual Representation in SVM

Eliminating $\boldsymbol{w}$ and b from Eq. 8-6 using conditions, substituting Eq. 8-7 and -8 into Eq. 8-6, the dual representation of the maximum margin problem becomes

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{n=1}^{N} a_n\{t_n(\boldsymbol{w}^T\phi(\boldsymbol{x}_n) + b) - 1\}$$

$$\boxed{\boldsymbol{w} = \sum_{n=1}^{N} a_n t_n \phi(\boldsymbol{x}_n)}$$

$$\boxed{\sum_{n=1}^{N} a_n t_n = 0}$$

$$\tilde{L}(\boldsymbol{a}) = \frac{1}{2}\sum_{n=1}^{N} a_n t_n \phi_n \sum_{m=1}^{N} a_m t_m \phi_m - \sum_{n=1}\sum_{m=1} a_n t_n \phi_n a_m t_m \phi_m - \cancel{b\sum_{n} a_n \phi_n} + \sum_{n} a_n$$

8.4. Dual Representation in SVM

Then the **maximum margin dual representation** becomes

$$\tilde{L}(\boldsymbol{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) \qquad \text{Eq. 8 - 9}$$

w.r.t. ***a*** subject to the constraints

$$a_n \geq 0, \qquad \text{Eq. 8 - 10}$$

$$\sum_{n=1}^{N} a_n t_n = 0\,. \qquad \text{Eq. 8 - 11}$$

Eq. 8-9 becomes a ***non-parametric*** SVM using ***kernel*** method.

8.4. Dual Representation in SVM

To classify **new data points** using the training model, we evaluate the sign of $y(\boldsymbol{x})$ by expressing using Eq. 8-7 to give

$$y(\boldsymbol{x}) = \sum_{n=1}^{N} a_n t_n k(\boldsymbol{x}, \boldsymbol{x}_n) + b.$$

It satisfies the KKT conditions,

$$a_n \geq 0$$
$$t_n y(\boldsymbol{x}_n) - 1 \geq 0$$
$$a_n\{t_n y(\boldsymbol{x}_n) - 1\} = 0.$$

The data points satisfying $t_n y(\boldsymbol{x}_n) = 1$ are SVs.

8.4. Dual Representation in SVM

Once the value for $\boldsymbol{a}$ is found, the threshold parameter $\boldsymbol{b}$ can be determined,

$$t_n\left(\sum_{m\in\mathcal{S}} a_m t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) + b\right) = 1$$

where $\mathcal{S}$ is the set of indices of SVs.

The average over all SVs gives

$$b = \frac{1}{N_\mathcal{S}}\sum_{n\in\mathcal{S}}\left(t_n - \sum_{m\in\mathcal{S}} a_m t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m)\right)$$

where $N_\mathcal{S}$ is the total number of SVs.

8.4. Dual Representation in SVM

- The maximum-margin classifier in terms of the minimization of an error function with a **quadratic regularizer** is

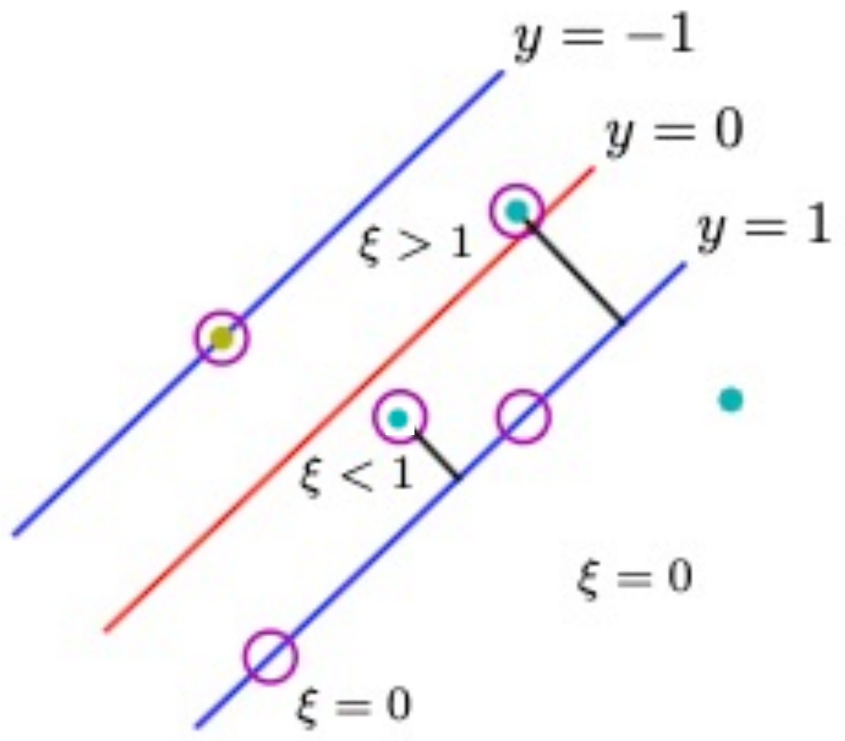
$$\sum_{n=1}^{N} E_\infty(y(\boldsymbol{x}_n)t_n - 1) + \lambda||\boldsymbol{w}||^2 \qquad \textit{Eq. 8 - 12}$$

where $E_\infty(z)$ is a function that is 0 if $z \geq 0$ and ∞ otherwise.

8.5. Overlapping Class Distributions

8.5. Overlapping Class Distributions

The class-conditional distributions may overlap in which the exact separation of the training data can lead to poor generalization.

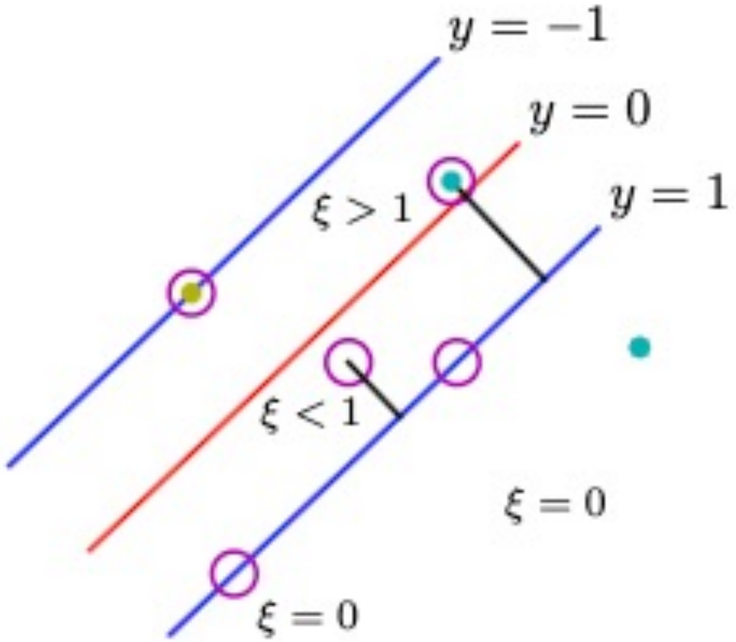


8.5. Overlapping Class Distributions

Eq. 8-12 needs to be modified by introducing **slack variables** $\xi_n \geq 0$ with one slack variable for each data point.

- For $\xi_n = 1$, a point on the boundary and **classified correctly**.
- For $0 < \xi \leq 1$, a point lies ***inside the margin*** and on the correct side of the decision boundary.
- For $\xi_n > 1$, a point is ***misclassified*** and the exact classification constrains then can be replaced with

$$t_n y(\boldsymbol{x}_n) \geq 1 - \xi_n. \qquad \textit{Eq. 8 - 13}$$



8.5. Overlapping Class Distributions

The goal is to maximize margin while *softly penalizing* points that **lie on the wrong side of the boundary**

$$C\sum_{n=1}^{N}\xi_n + \frac{1}{2}||\boldsymbol{w}||^2 \qquad \textit{Eq. 8 - 14}$$

where $C > 0$ controls the trade-off between the slack variable penalty and the margin, $C = 1/||\boldsymbol{w}||$.

8.5. Overlapping Class Distributions

We need to minimize Eq. 8-14 subject to Eq. 8-13 together with $\xi_n \geq 0$.
The corresponding Lagrangian is given by

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N} a_n\{t_n y(\boldsymbol{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N}\mu_n\xi_n \qquad \text{Eq. 8 - 15}$$

where $\{a_n \geq 0\}$ and $\{\mu_n \geq 0\}$ are Lagrange multipliers.

The corresponding KKT conditions are

$$a_n, \mu_n, \xi_n \geq 0$$
$$t_n y(x_n) - 1 + \xi \geq 0$$
$$a_n(t_n y(x_n) - 1 + \xi) = 0$$
$$\mu_n \xi_n = 0$$

where $n = 1, \dots, N$.

8.5. Overlapping Class Distributions

The optimization of $\boldsymbol{w}$, b, and $\{\xi_n\}$ gives

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \rightarrow \boldsymbol{w} = \sum_{n=1}^{N} a_n t_n \phi(\boldsymbol{x}_n)$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{n=1}^{N} a_n t_n = 0$$

$$\frac{\partial L}{\partial \xi_n} = 0 \rightarrow a_n = C - \mu_n.$$

Since $\mu_n \geq 0$, it is known as ***box constrains*** that $0 \leq a_n \leq C$.

8.6. Loss Function and Optimization

8.6. Loss Function and Optimization

- For $a_n > 0$, the data points must satisfy
$$t_n y(\boldsymbol{x}_n) = 1 - \xi_n.$$

- If $a_n < C$, then $a_n = C - \mu_n$ implies that $\mu_n > 0$ and $\xi_n = 0$.

- If $a_n = C$, points lie inside the margin and can either be correctly classified if $\xi \leq 1$ or misclassified if $\xi > 1$.

8.6. Loss Function and Optimization

- The support vectors for $0 < a_n < C$ having $\xi_n = 0$ and $t_n y(\boldsymbol{x}_n) = 1$ will satisfy

$$t_n\left(\sum_{m\in\mathcal{S}} a_m t_m k(x_n, x_m) + b\right) = 1.$$

- The parameter b is the averaging of those points

$$b = \frac{1}{N_{\mathcal{M}}}\sum_{n\in\mathcal{M}}\left(t_n - \sum_{m\in\mathcal{S}} a_m t_m k(x_n, x_m)\right).$$

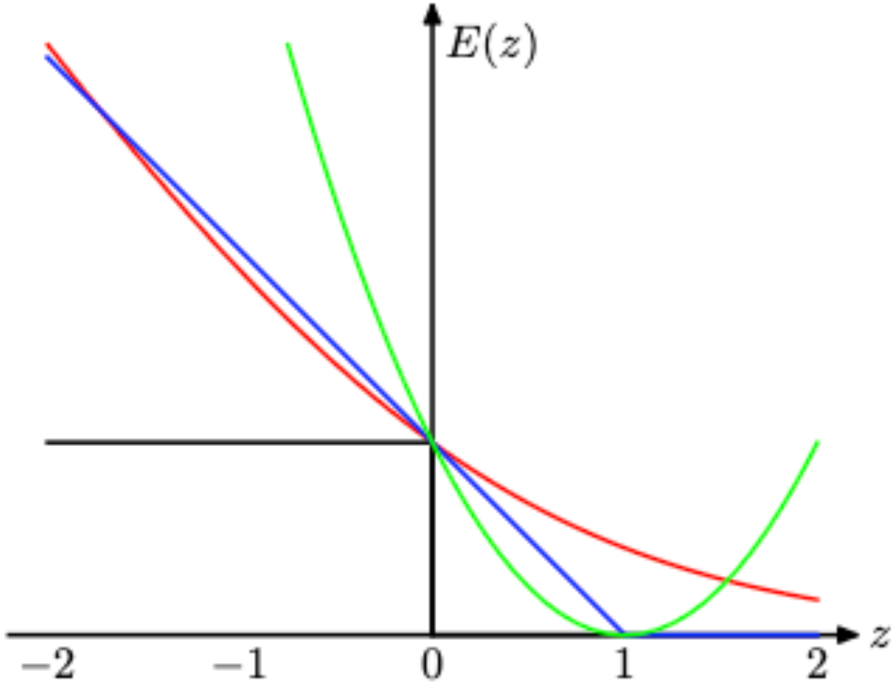
- The objective function can be written as

$$\sum_{n=1}^{N} E_{SV}(y_n, t_n) + \lambda||\mathbf{w}||^2$$

where $\lambda = (2C)^{-1}$ and $E_{SV}(\cdot)$ is the ***hinge*** error function defined as

$$E_{SV}(y_n t_n) = [1 - y_n t_n]_+$$

and $[\cdot]_+$ denotes the positive part.



8.6. Loss Function and Optimization

The constrain can be written more concisely as

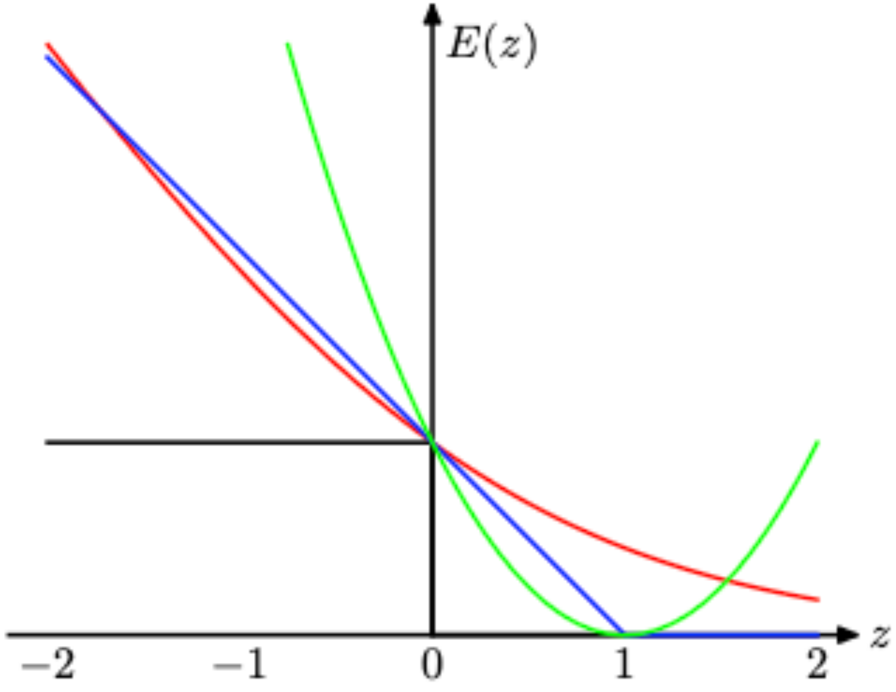
$$y_n t_n \geq 1 - \xi_n.$$

With $\xi_n > 0$,

$$\xi_n = \max(0, 1 - y_n t_n).$$

The learning problem is equivalent to the unconstrained optimization over $\boldsymbol{w}$:

$$\min_{w,b} \frac{||w||^2}{2} + \lambda \sum_{n=1}^{N} \max(0, 1 - y_n t_n)$$



8.7. SVMs for Regressions

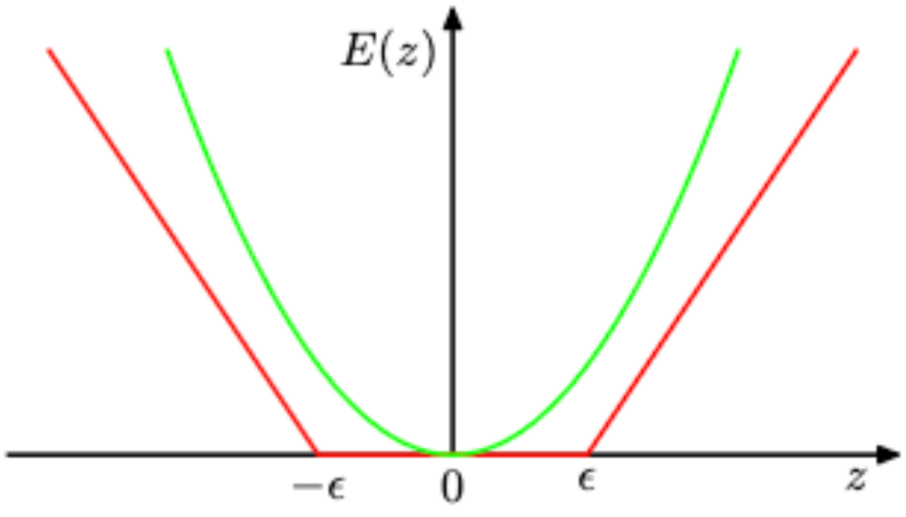
8.7. SVMs for Regressions

In simple linear regression, a regularized error function is

$$\frac{1}{2}\sum_{n=1}^{N}\{y_n - t_n\}^2 + \frac{\lambda}{2}||\boldsymbol{w}||^2.$$

To obtain sparse solutions, the error function is replaced by an **ϵ-insensitive *error function***:

$$E_\epsilon(y(x) - t) = \begin{cases} 0 & if\ |y(x) - t| < \epsilon \\ |y(x) - t| - \epsilon & otherwise \end{cases}.$$



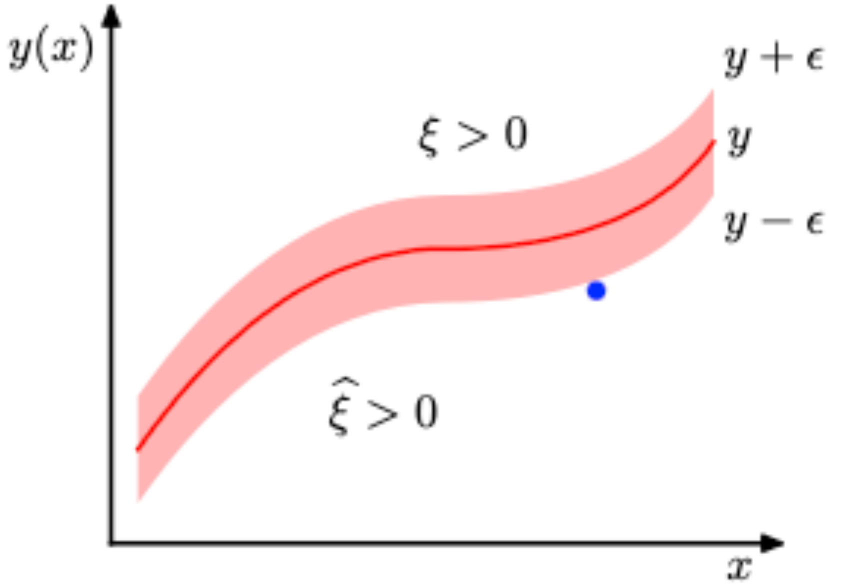
We minimize a regularized error function given by

$$C\sum_{n=1}^{N} E_\epsilon(y(\boldsymbol{x}_n) - t_n) + \frac{1}{2}||\boldsymbol{w}||^2.$$

8.7. SVMs for Regressions

For each data point $\boldsymbol{x}_n$, there are two slack variables $\xi_n \geq 0$ and $\widehat{\xi_n} \geq 0$.

- $\xi_n > 0$ corresponds to a point $t_n > y(\boldsymbol{x}_n) + \epsilon$.
- $\widehat{\xi_n} > 0$ corresponds to a point $t_n < y(x_n) - \epsilon$.



The condition for a target point to lie inside the ϵ-tube $y_n - \epsilon \leq t_n \leq y_n + \epsilon$ with the corresponding conditions

$$t_n \leq y(\boldsymbol{x}_n) + \epsilon + \xi_n$$
$$t_n \geq y(\boldsymbol{x}_n) - \epsilon - \widehat{\xi_n}.$$

8.7. SVMs for Regressions

The error function for support vector regression can be written as

$$C\sum_{n=1}^{N}(\xi_n + \widehat{\xi_n}) + \frac{1}{2}||\boldsymbol{w}||^2$$

which must be minimized subject to the constraints $\xi_n \geq 0$ and $\widehat{\xi_n} \geq 0$.

$$L = C\sum_{n=1}^{N}(\xi_n + \widehat{\xi_n}) + \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{n=1}^{N}(\mu_n\xi_n + \widehat{\mu_n}\,\widehat{\xi_n}) - \sum_{n=1}^{N}a_n(\epsilon + \xi_n + y_n - t_n) - \sum_{n=1}^{N}\widehat{a_n}(\epsilon + \widehat{\xi_n} - y_n + t_n).$$

Eq. 8 - 16

8.7. SVMs for Regressions

Setting the derivative of Eq. 8-16 w.r.t. $\boldsymbol{w}, b, \xi_n$, and $\widehat{\xi_n}$ to zero gives

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \rightarrow \boldsymbol{w} = \sum_{n=1}^{N} (a_n - \widehat{a_n}) \phi(\boldsymbol{x}_n) \qquad \textit{Eq. 8 - 17}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{n=1}^{N} (a_n - \widehat{a_n}) = 0$$

$$\frac{\partial L}{\partial \xi_n} = 0 \rightarrow a_n + \mu_n = C$$

$$\frac{\partial L}{\partial \widehat{\xi_n}} = 0 \rightarrow \widehat{a_n} + \widehat{\mu_n} = C.$$

8.7. SVMs for Regressions

Using the result to eliminate the corresponding variables from Eq.6-16, the dual maximization becomes

$$\tilde{L}(a,\hat{a}) = -\frac{1}{2}\sum_{n,m}(a_n - \widehat{a_n})(a_m - \widehat{a_m})k(x_n, x_m) - \epsilon\sum_n(a_n - \widehat{a_n}) + \sum_n(a_n - \widehat{a_n})t_n$$

with the box constraints

$$0 \le a_n \le C$$
$$0 \le \widehat{a_n} \le C.$$

Substituting Eq. 8-17 to the model, the prediction for new inputs can be made as

$$y(x) = \sum_n(a_n - \widehat{a_n})k(\boldsymbol{x}, x_n) + b\,.$$

Eq. 8 - 18

The bias parameter b can be found for a data point with $0 < a_n < C$ and $\xi_n = 0$ which must satisfy $\epsilon + y_n - t_n = 0$. Using Eq.6-18, b is

$$b = t_n - \epsilon - \sum_m(a_m - \widehat{a_m})k(\boldsymbol{x}_n, \boldsymbol{x}_m)\,.$$

8.8. Example

8.8. Example

Recall the perceptron example, we found the model $x_2 = 2x_1 - 1.667$.

```python
import pandas as pd
import matplotlib.pyplot as plt
X = pd.DataFrame({'X1':[1,2,3,2,3,4],'X2':[2,3,4.9,1,2,3.9],'Y':[1,1,1,-1,-1,-1]})

```

[1] ✓ 1.6s

```python
import numpy as np
Wp = 2
x1 = np.arange(1,4.5,0.5)
x2p = Wp*x1-1.667
x2p
```

[2] ✓ 0.3s

```
array([0.333, 1.333, 2.333, 3.333, 4.333, 5.333, 6.333])
```

8.8. Example

```python
Wsvm1 = np.dot(X['Y'],X['X1'])
Wsvm2 = np.dot(X['Y'],X['X2'])
Ysvm = Wsvm1*X['X1']+Wsvm2*X['X2']
bsvm = np.sum(X['Y'][0:3]-Ysvm[0:3])/6
print(f'y={Wsvm1}X1+{Wsvm2}X2{bsvm}')
Wsvm = -Wsvm1/Wsvm2
x2svm = Wsvm*x1-bsvm/Wsvm2
print(f'X2={Wsvm}X1+{-bsvm/Wsvm2}')
```

✓ 0.3s

```
y=-3X1+3.0000000000000004X2-1.4500000000000001
X2=0.9999999999999999X1+0.4833333333333336
```

In this example, assume $a_n = 1$.

The parameter $\boldsymbol{w}$ can be determined by

$$\boldsymbol{w} = \sum_i a_i t_i \boldsymbol{x}_i .$$

For x_1,

$$w_1 = \sum_i a_i t_i x_{1i}$$

$$= 1 \cdot 1(1+2+3) - 1 \cdot 1(2+3+4)$$

$$= -3$$

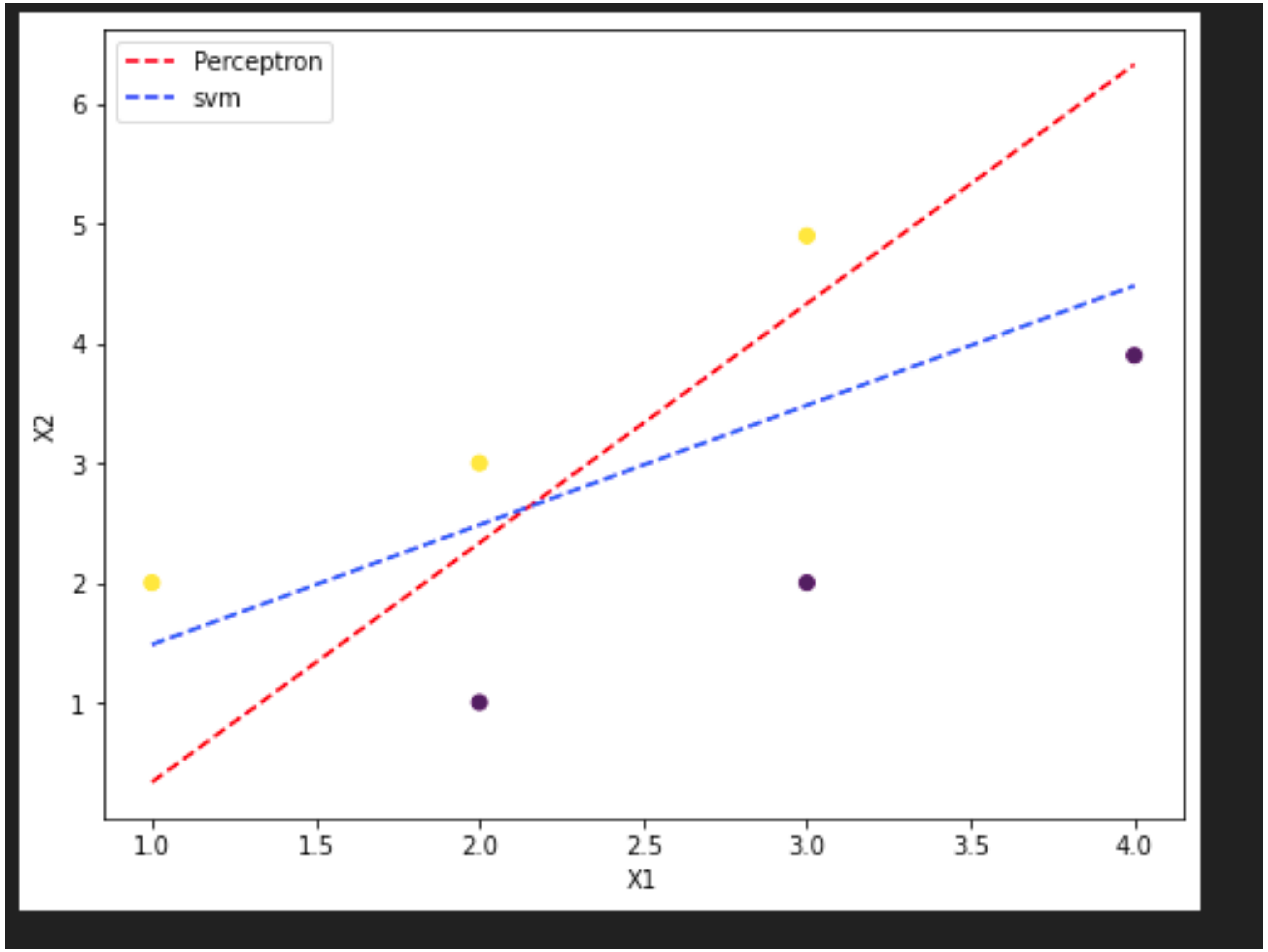
The bias parameter can be determined by

$$b = \frac{1}{N} \sum_{i,t \in 1} y_i - \boldsymbol{w}^T \boldsymbol{x}$$

$$= -1.45$$

The model is then $y = -3x_1 + 3x_2 - 1.45$ and it leads to $x_2 = x_1 + 0.483$.

Example



8.9. Example

8.9. Conclusion

- SVM was originally invented from linearly separable binary classification.
- SVM is the extension version of perceptron to find the best hyperplane.
- SVM is powerful.
  - SVM is a parametric model when a data is linearly separable.
  - SVM uses kernel method and becomes a non-parametric model for non-linear data.