CS 559: Machine Learning Fundamentals & Applications

Lecture 5: Linear Classification

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

Outline

5.0 Lecture 4 Review

Linear Regression

- One of simplest linear models in ML.
- Overcoming the conditions and assumptions may be challenge.

Model Selection

- Overfit vs. Underfit
- Regularization
- Bias-Variance Trade-off

5.1. Introduction

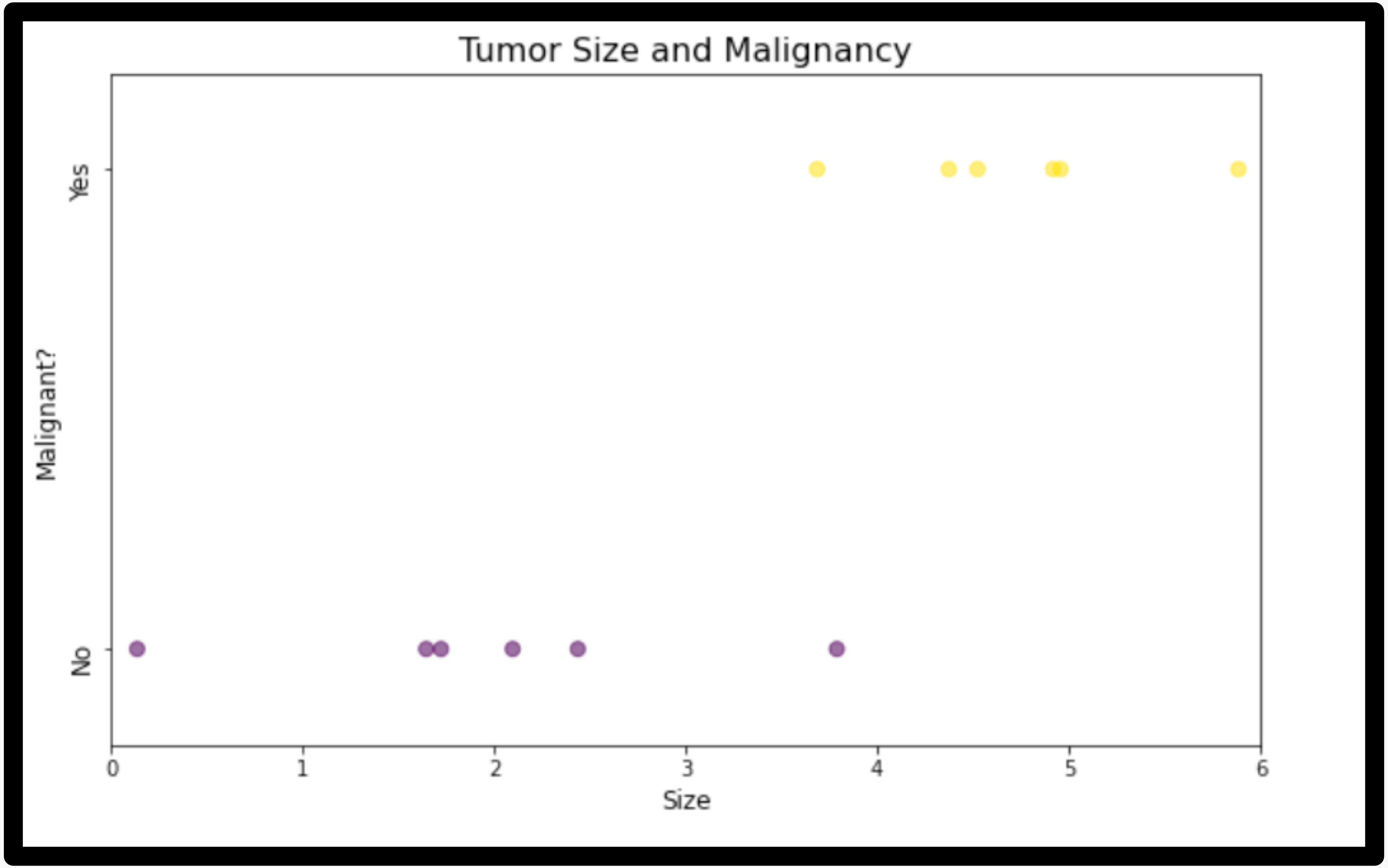
Goal: to take an input vector $\boldsymbol{x}$ and to assign it to one of K discrete classes C_k where $k = 1, \dots, K$.

- Data Condition: The classes are disjoint, and the target is discrete.
- The input space is divided into ***decision regions*** – linear decision surfaces called ***hyperplanes*** in D-1 dimensions if the input space is in D-dimensions.
- For binary problem, target $t \in \{0,1\}$ or $\{-1,1\}$.

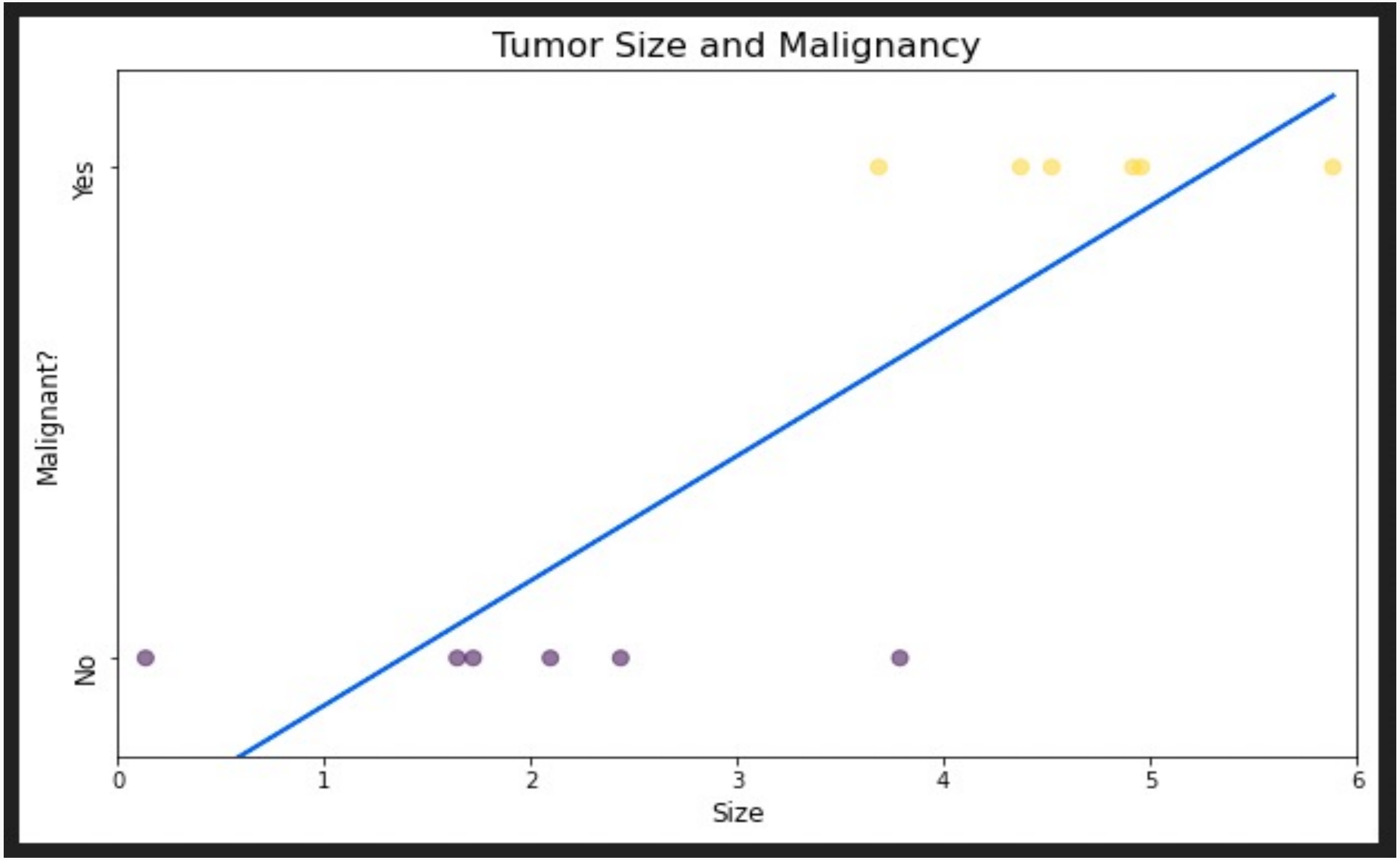
We approach in three different approaches:

1. A ***discriminant modeling***: directly assigns each vector **x** to a specific class (LDA, Perceptron).
2. A ***probabilistic modeling***: determines the conditional probability distribution $p(C_k|\boldsymbol{x})$ and represent them as a parametric model.
    1. ***Probabilistic generative modeling***: a classification modeling using Bayes' theorem (MLE).
    2. ***Probabilistic discriminative modeling****:* a classification modeling using MLE in a form of discriminative modeling (logistic regression).

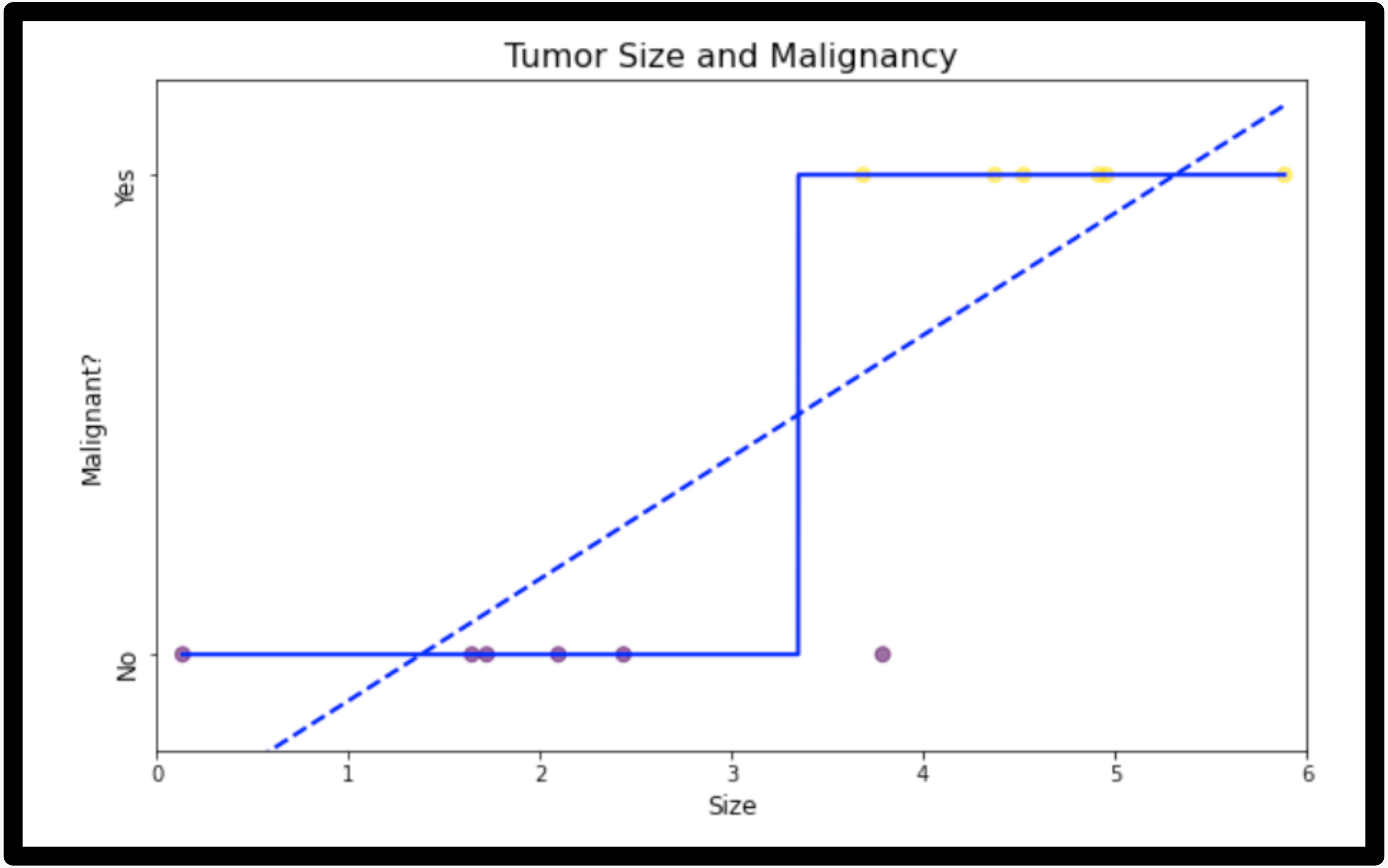
5.1. Introduction



5.1. Introduction



5.1. Introduction



5.2. Discriminant Functions

Consider a simple linear discriminant function of the input vector **x**,

$$y(\boldsymbol{x}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + w_0 \quad (5\text{-}1)$$

where **w** is called a *weight vector,* and w_0 is a bias.

- If $w_0 < 0$, sometimes we call it a *threshold*.
- If $y(\boldsymbol{x}) \geq 0$, then **x** is assigned to class C_1. Otherwise, C_2.
- The corresponding hyperplane is when $y(\boldsymbol{x}) = 0$.

5.2. Discriminant Functions

- If two points $\boldsymbol{x_A}$ and $\boldsymbol{x_B}$ both lying on the decision surface, $y(\boldsymbol{x_A}) = y(\boldsymbol{x_B}) = 0$.
- The normal distance from the origin to the decision surface is

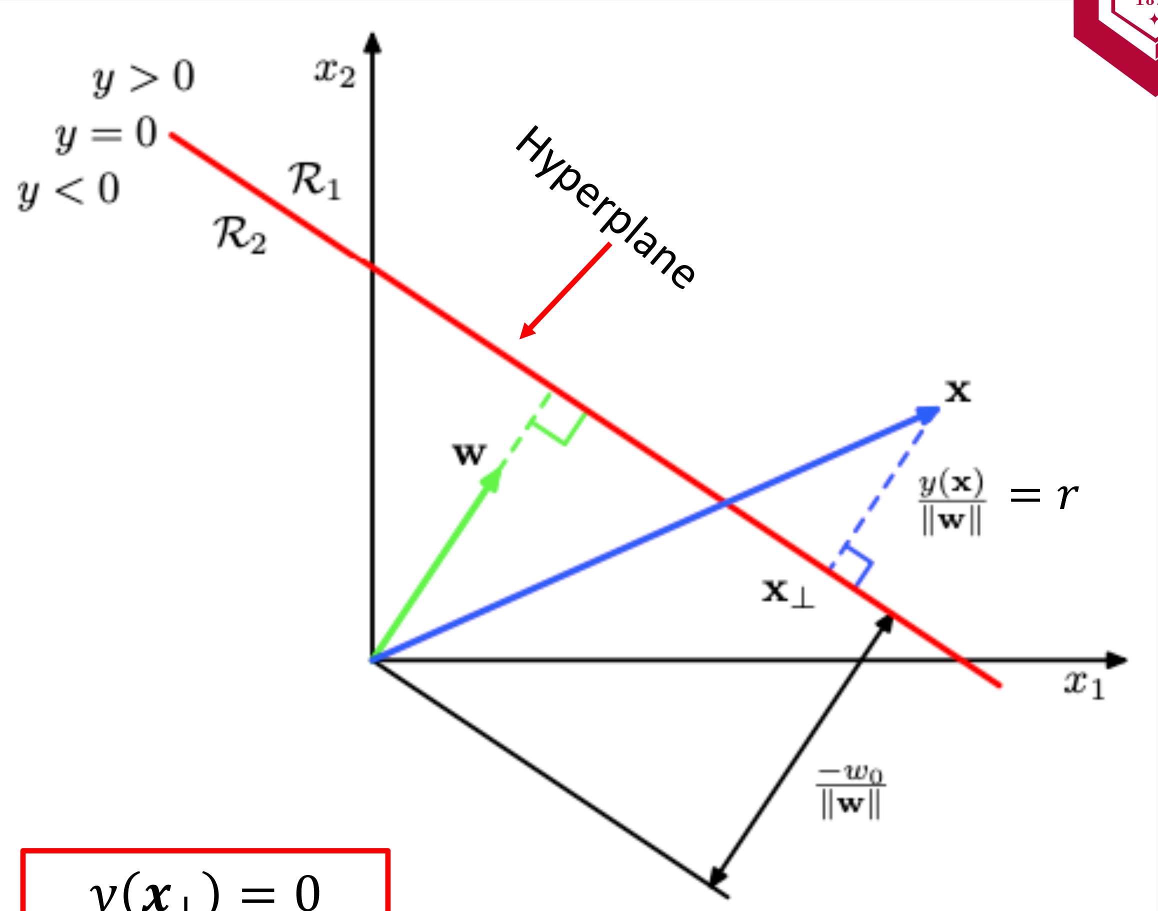
$$\frac{\boldsymbol{w}^T\boldsymbol{x}}{|\boldsymbol{w}|} = -\frac{w_0}{|\boldsymbol{w}|}.$$

- The perpendicular distance r of the point $\mathbf{x}$ from the hyperplane can be expressed as

$$\boldsymbol{x} = \boldsymbol{x}_\perp + r\left(\frac{\boldsymbol{w}}{|\boldsymbol{w}|}\right)$$

where $\boldsymbol{x}_\perp$ is the orthogonal projection onto the hyperplane. It also can be expressed in terms of y:

$$y(\boldsymbol{x}) = y(\boldsymbol{x}_\perp) + r\left(\frac{\boldsymbol{w}^T\boldsymbol{w}}{|\boldsymbol{w}|}\right) \rightarrow r = \frac{y(\boldsymbol{x})}{|\boldsymbol{w}|}.$$



$$\boxed{y(\boldsymbol{x}_\perp) = 0}$$

$$\boxed{|\boldsymbol{w}| = \sqrt{\boldsymbol{w}^T\boldsymbol{w}} \rightarrow \frac{|\boldsymbol{w}|}{\boldsymbol{w}^T\boldsymbol{w}} = \frac{1}{|\boldsymbol{w}|}}$$

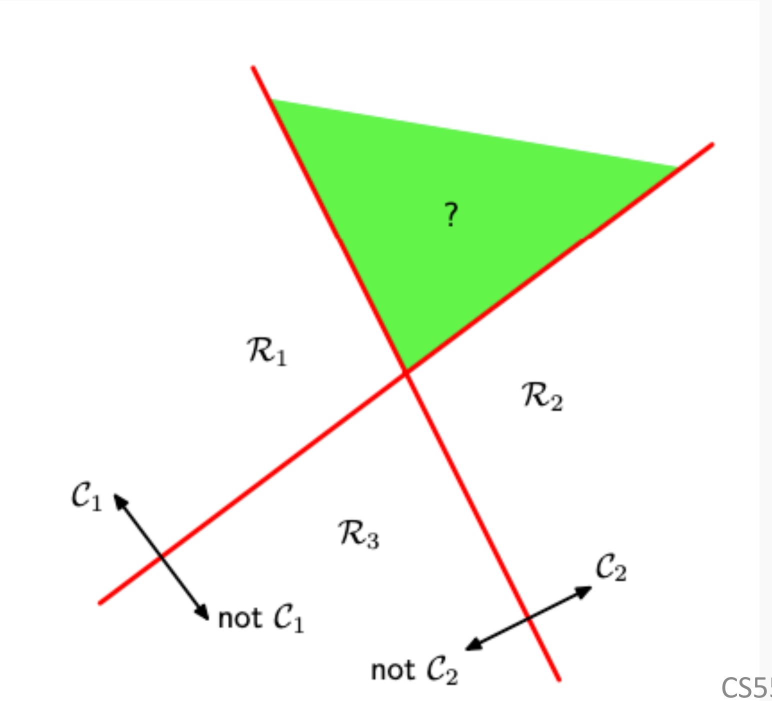
(5-2)

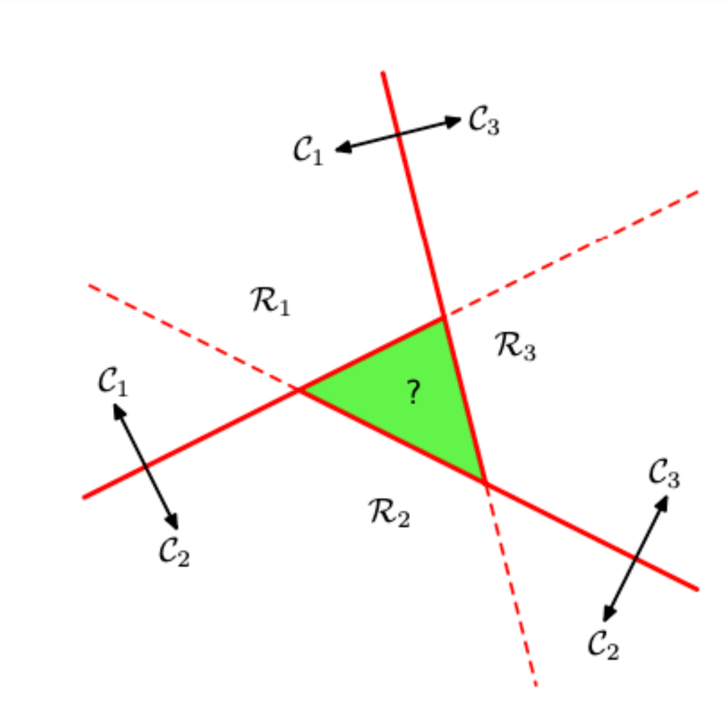
5.2. Discriminant Functions

What happens when the target has more than two classes?

If we approach as binary classification problem, there are two possible approaches.

- *One-verse-rest* classifier: the *K-1* classifier that solves as a two-class problem of separating points in a particular class C_k from points not in that class.
- *One-verse-one* classifier: each point is classified according to a majority vote amongst the discriminant functions.
- Either holds an *ambiguous region* where $y(\boldsymbol{x})$ is not possible for some $\mathbf{x}$.





5.2. Discriminant Functions

- Instead, we consider a single K-class linear discriminant function comprising K many functions in the form of

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}$$

  to assign a class C_k for a point $\mathbf{x}$ if $y_k(\boldsymbol{x}) > y_j(\boldsymbol{x})$ for all $j \neq k$.
- The class boundary between C_k and C_j is when $y_k = y_j$ defined by

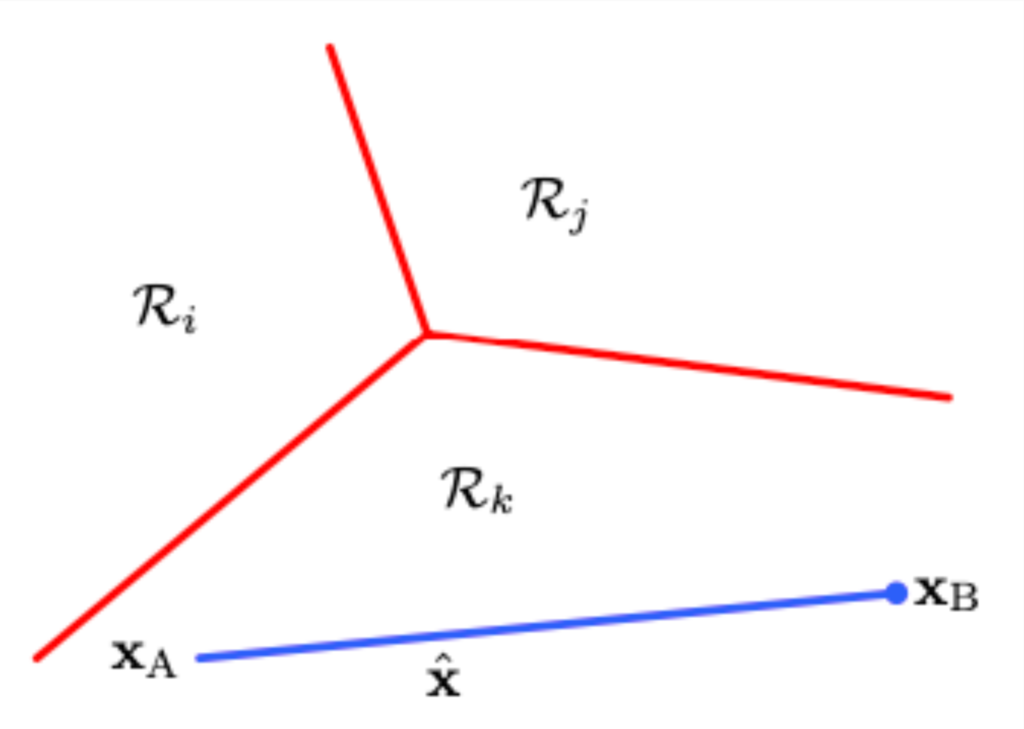
$$\left(\boldsymbol{w}_k - \boldsymbol{w}_j\right)^T \boldsymbol{x} + \left(w_{ko} - w_{j0}\right) = 0. \qquad \text{(5-3)}$$

- In this way, we can connect the decision regions into a single point and be convex.
- Consider any point $\widehat{\boldsymbol{x}}$ lies on the line between two points $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ in R_k,

$$\widehat{\boldsymbol{x}} = \lambda \boldsymbol{x}_A + (1-\lambda)\boldsymbol{x}_B$$

  where $0 \leq \lambda \leq 1$. From the linearity of the discriminant functions, it follows that

$$y_k(\widehat{\boldsymbol{x}}) = \lambda y_k(\boldsymbol{x}_A) + (1-\lambda) y_k(\boldsymbol{x}_B).$$



5.2.1. Fisher's Linear Discriminant Analysis (LDA)

- Consider a binary linear classifier in terms of *dimensionality reduction* for D-dimensional input vector $\mathbf{x}$ having an output $\{C_1, C_2\}$.

- Let y be C_1 if $y \geq -w_0$ and otherwise C_2.

- Suppose we projectile down to 1-dimension using
$$y = \boldsymbol{w}^T \boldsymbol{x}.$$

- We can control $\boldsymbol{w}$ to project by maximizing the class separation to avoid the considerable loss information that may occur in projection D-dimensional space to 1-dimentional space.

5.2.1. Fisher's Linear Discriminant Analysis (LDA)

If there are N_1 points in C_1 and N_2 in C_2, the mean vector of the two classes is

$$\boldsymbol{m_1} = \frac{1}{N_1}\sum_{n\in C_1} \boldsymbol{x_n}, \qquad \boldsymbol{m_2} = \frac{1}{N_2}\sum_{n\in C_2} \boldsymbol{x_n}.$$

The separation mean of the projected means can measure the separation of the classes to maximize

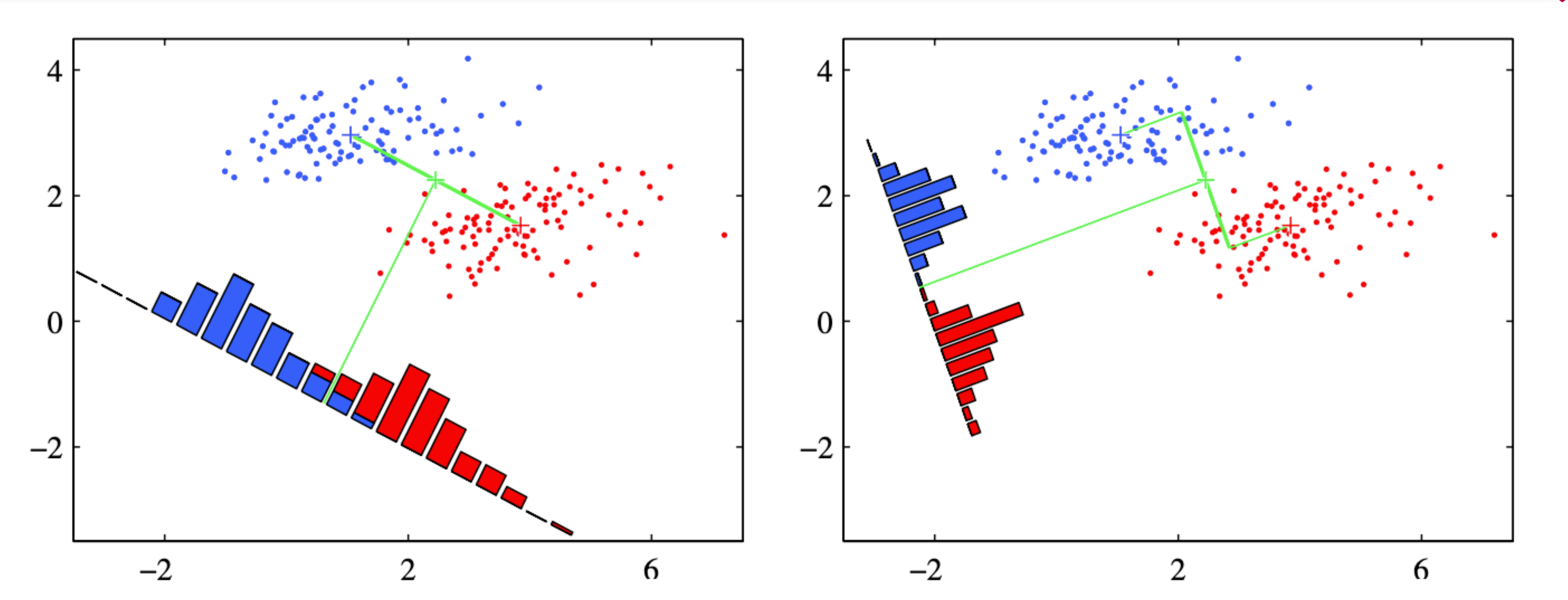
$$m_2 - m_1 = \boldsymbol{w}^T(\boldsymbol{m_2} - \boldsymbol{m_1}). \tag{5-13}$$

However, the separation becomes arbitrary large when the magnitude of **w** is large.

To avoid this, $\boldsymbol{w}$ will be forced to be a unit length $\sum_i w_i^2 = 1$.

- However, the strong nondiagonal **covariances** of the class distributions still may have **overlap** when projected onto the line joining their means.
- To minimize the overlap, the function must give the **largest separation** between the projected class means while the **smallest variance** within each class, also known as the within-class variance.

5.2.1. Fisher's Linear Discriminant Analysis (LDA)



5.2.1. Fisher's Linear Discriminant Analysis (LDA)

The **within-class variance** of the transformed data from class C_k is

$$s_k^2 = \sum_{n \in C_k} (y - m_k)^2 \qquad \text{where } y_n = \boldsymbol{w}^T \boldsymbol{x}_n \tag{5-4}$$

and the **total within-class variance** is

$$s^2 = \sum_{k=1}^{K} s_k^2 \,. \tag{5-5}$$

The **Fisher criterion**, $J(\boldsymbol{w})$, is the ratio of the between-class variance to the within-class variance (Eq. 5-5) and is defined as

$$J(\boldsymbol{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \,. \tag{5-6}$$

5.2.1. Fisher's Linear Discriminant Analysis (LDA)

It can be rewritten as the dependence on $\boldsymbol{w}$ as

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}}$$

where $\boldsymbol{S}_B$ is the ***between-class*** covariance matrix

$$\boldsymbol{S}_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T$$

and $\boldsymbol{S}_w$ is the **total *within-class*** covariance matrix is given by

$$\boldsymbol{S}_w = \sum_{n \in C_1} (\boldsymbol{x}_n - \boldsymbol{m}_1)(\boldsymbol{x}_n - \boldsymbol{m}_1)^T + (\boldsymbol{x}_n - \boldsymbol{m}_2)(\boldsymbol{x}_n - \boldsymbol{m}_2)^T.$$

$J(\boldsymbol{w})$ is maximized when $\nabla_{\mathbf{w}} J(\boldsymbol{w}) = 0$:

$$\left(\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}\right) \boldsymbol{S}_w \boldsymbol{w} = \left(\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}\right) \boldsymbol{S}_B \boldsymbol{w}. \quad (5\text{-}7)$$

5.2.1. Fisher's Linear Discriminant Analysis (LDA)

To obtain $\boldsymbol{w}$, we can consider the following two properties.

- The *between-class* shows that $\boldsymbol{S_B w}$ is always in the direction of $(\boldsymbol{m_2} - \boldsymbol{m_1})$.
- Considering the direction only (not the magnitude), the scalar factors $\left(\boldsymbol{w^T S_B w}\right)$ and $\left(\boldsymbol{w^T S_w w}\right)$ can be dropped in Eq. (5-7).

Then multiplying $\boldsymbol{S_w^{-1}}$ on both sides, then $\boldsymbol{w}$ can be obtained as

$$\boldsymbol{w} \propto \boldsymbol{S_w^{-1}}(\boldsymbol{m_2} - \boldsymbol{m_1}). \tag{5-8}$$

Alternatively, we can find $\boldsymbol{w}$ that maximizes the criterion using Linear Algebra via generalized eigenvalue problem:

- The eigenvalue, λ, can be computed as

$$\boldsymbol{S_B w} = \lambda \boldsymbol{S_w w} \rightarrow \left(\boldsymbol{S_w^{-1} S_B} - \lambda \boldsymbol{I}\right) = \boldsymbol{0} \tag{5-9}$$
$$\det\left(\boldsymbol{S_w^{-1} S_B} - \lambda \boldsymbol{I}\right) = \boldsymbol{0}$$

- The eigenvector for $\lambda_{\max}$ is then solution.

5.2.1. Fisher's Linear Discriminant Analysis (LDA)

We can relate to the least squares.

Suppose we can model the class-conditional densities $p(y|C_k)$ using Gaussian distribution.
The solution of $\boldsymbol{w}$ can be approached via MLE.

Consider a binary classification and let the target $t_1 = \frac{N}{N_1}$ for class C_1 where N is the total number and N_1 is the number in C_1. Similarly, let $t_2 = -\frac{N}{N_2}$.

Start from the sum-of-squares error function,

$$E = \frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{w}^T\boldsymbol{x}_n + w_0 - t_n)^2 .$$

5.2.1. Fisher's Linear Discriminant Analysis (LDA)

Set the derivative of the sum-of-squares error function w.r.t. **w** equals to 0,

$$\sum_{n=1}^{N} (\boldsymbol{w}^T \boldsymbol{x_n} + w_0 - t_n)\boldsymbol{x_n} = 0.$$

The use of

$$\sum_{n=1}^{N} t_n = \frac{N_1 N}{N_1} - \frac{N_2 N}{N_2} = 0$$

$$\boldsymbol{m} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x_n} = \frac{1}{N}(N_1 \boldsymbol{m_1} + N_2 \boldsymbol{m_2})$$

and

$$w_0 = -\boldsymbol{w}^T \boldsymbol{m},$$

the derivative of the sum-of-squares error function (using $\boldsymbol{S_w}$ and $\boldsymbol{S_B}$ from slide #19) becomes

$$\left(\boldsymbol{S_w} + \frac{N_1 N_2}{N}\boldsymbol{S_B}\right)\boldsymbol{w} = N(\boldsymbol{m_1} - \boldsymbol{m_2}). \quad (5\text{-}9)$$

5.2.1. Fisher's Linear Discriminant Analysis (LDA)

For $K > 2$ classes, assume $D > K$.

Suppose there are $D' > 1$ linear new features $y_k = \boldsymbol{w}_k^T \boldsymbol{x}$ for $k = 1, \dots, D'$ where $\mathbf{y}$ is the vector for y_k:

$$\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{x}$$

where the bias parameter w_0 is excluded.

The within-class covariance matrix is

$$\boldsymbol{S}_{\boldsymbol{w}} = \sum_{k=1}^{K} \boldsymbol{S}_k$$

where $\boldsymbol{S}_k$ is

$$\boldsymbol{S}_k = \sum_{n \in C_k} (\boldsymbol{x}_n - \boldsymbol{m}_k)(\boldsymbol{x}_n - \boldsymbol{m}_k)^T = \sum_{n \in C_k} \left(\boldsymbol{x}_n - \frac{1}{N_k} \sum_{n \in C_k} \boldsymbol{x}_n \right) \left(\boldsymbol{x}_n - \frac{1}{N_k} \sum_{n \in C_k} \boldsymbol{x}_n \right)^T .$$

5.2.1. Fisher's Linear Discriminant Analysis (LDA)

The total covariance matrix

$$\boldsymbol{S}_T = \sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{m})(\boldsymbol{x}_n - \boldsymbol{m})^T$$

where $\boldsymbol{m}$ is the mean of total set

$$\boldsymbol{m} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n = \frac{1}{N} \sum_{k=1}^{K} N_k \boldsymbol{m}_k \, .$$

The total covariance is the sum of within-class variance and the between-class covariance

$$\boldsymbol{S_B} = \sum_{k=1}^{K} N_k (\boldsymbol{m_k} - \boldsymbol{m})(\boldsymbol{m_k} - \boldsymbol{m})^T \, ,$$

$$\boldsymbol{S}_T = \boldsymbol{S}_W + \boldsymbol{S}_B .$$

5.2.1. Fisher's Linear Discriminant Analysis (LDA)

In the projected *D'*-dimensional **y**-space

$$\boldsymbol{s}_W = \sum_{k=1}^{K} \sum_{n \in C_k} (\boldsymbol{y_n} - \boldsymbol{\mu_k})(\boldsymbol{y_n} - \boldsymbol{\mu_k})^T$$

and

$$\boldsymbol{s}_B = \sum_{k=1}^{K} N_k (\boldsymbol{\mu_k} - \boldsymbol{\mu})(\boldsymbol{\mu_k} - \boldsymbol{\mu})^T$$

where

$$\boldsymbol{\mu_k} = \frac{1}{N_k} \sum_{n \in C_k} \boldsymbol{y}_n, \qquad \boldsymbol{\mu} = \frac{1}{N} \sum_{k=1}^{K} N_k \boldsymbol{\mu}_k .$$

Then, the criterion can be expressed as

$$J(\boldsymbol{W}) = Tr\{\boldsymbol{s}_{\boldsymbol{w}}^{-1} \boldsymbol{s}_{\boldsymbol{B}}\} \rightarrow J(\boldsymbol{w}) = Tr\left\{\left(\boldsymbol{W} \boldsymbol{S}_{\boldsymbol{W}} \boldsymbol{W}^{\boldsymbol{T}}\right)^{-\boldsymbol{1}} (\boldsymbol{W} \boldsymbol{S}_{\boldsymbol{B}} \boldsymbol{W})\right\}.$$

The weight values to maximize the criterion can be determined by the eigenvectors of $\boldsymbol{S}_{\boldsymbol{w}}^{-\boldsymbol{1}} \boldsymbol{S}_{\boldsymbol{B}}$ that correspond to the *D'* largest eigenvalues.

5.2.1. Fisher's Linear Discriminant Analysis (LDA)

Class	(x_1, x_2)	$\boldsymbol{m}$
C_1	(1,2), (2,3), (3,4.9)	(2, 3.3)
C_2	(2,1), (3,2), (4, 3.9)	(3,2.3)

The total within-class variance is

$$\boldsymbol{S_w} = \begin{bmatrix} 4 & 5.8 \\ 5.8 & 8.68 \end{bmatrix}$$

and its inverse is

$$\boldsymbol{S_w^{-1}} = \begin{bmatrix} 8.04 & -5.37 \\ -5.37 & 3.70 \end{bmatrix}.$$

Since $\boldsymbol{w}$ is proportional to $\boldsymbol{S_w^{-1}}(\boldsymbol{m_2} - \boldsymbol{m1})$,

$$\boldsymbol{w} \propto \boldsymbol{S_w^{-1}}(\boldsymbol{m_2} - \boldsymbol{m_1}) = \begin{bmatrix} -13.41 \\ 9.07 \end{bmatrix}$$

and its unit vector is

$$\hat{\boldsymbol{w}} = \frac{\boldsymbol{w}}{|\boldsymbol{w}|} = \begin{bmatrix} -0.83 \\ 0.56 \end{bmatrix}$$

and therefore, the model is

$$y = -0.83x_1 + 0.56x_2.$$

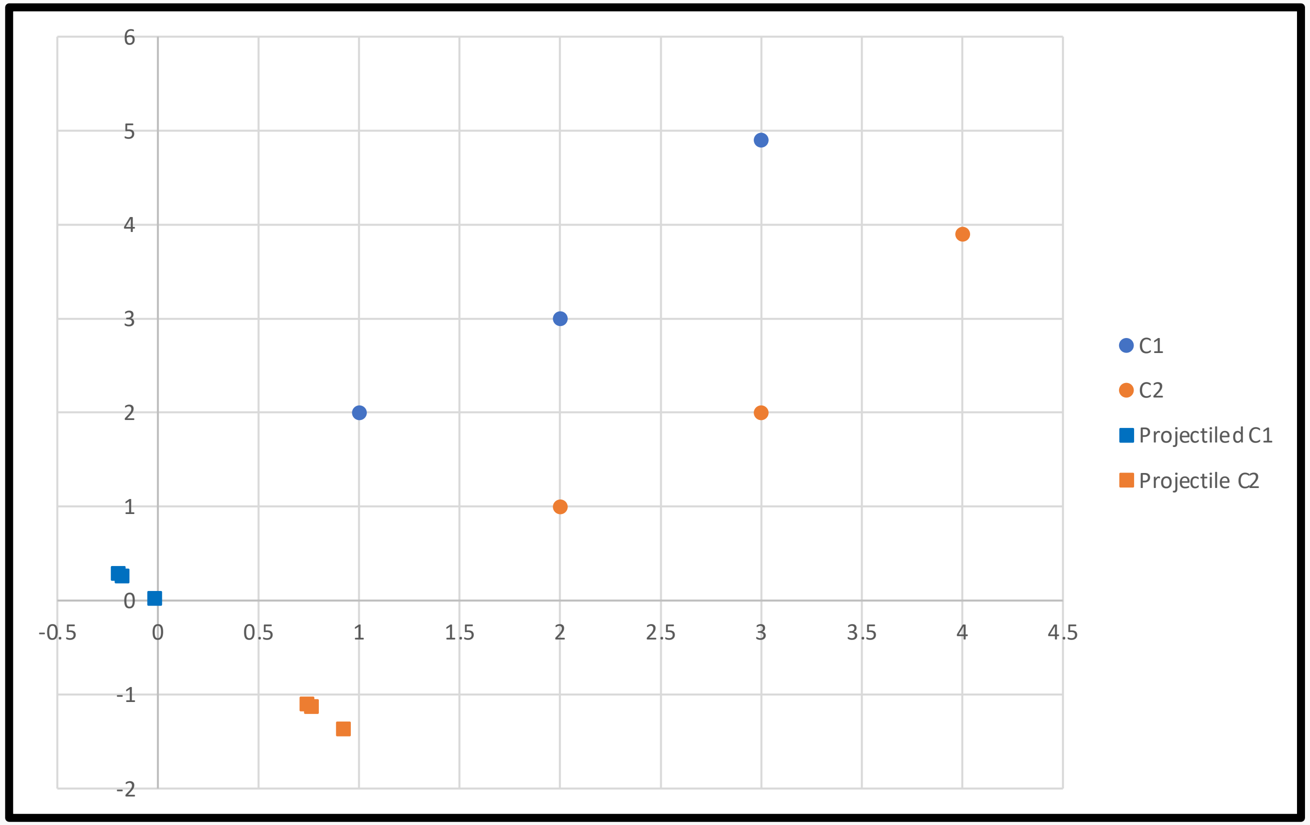
5.2.1. Fisher's Linear Discriminant Analysis (LDA)

Substituting data into the model reveals

$$y = [0.29 \quad 0.025 \quad 0.26 \quad -1.10 \quad -1.36 \quad -1.13].$$

For the complete orthogonal projection, set y=0 and find the model

$$x_2 = \frac{0.83}{0.56} x_1$$



5.2.2. Perceptron

Consider a binary class model of the form

$$y(\boldsymbol{x}) = f\left(\boldsymbol{w}^T \phi(\boldsymbol{x})\right)$$

where the nonlinear activation function $f(\cdot)$ is a step function of the form

$$f(a) = \begin{cases} 1, & a \geq 0 \\ -1, & a < 0 \end{cases}.$$

The target values are

$$t = \begin{cases} 1, & for\ C_1 \\ -1, & for\ C_2 \end{cases}.$$

5.2.2. Perceptron

The learning is motivated from the error function minimization.

- But the change of w can cause the change of decision boundary and lead to misclassification.
- Therefore, the approaching in gradient descent way is not possible. We need a different error function.
- Instead, we use the *perceptron criterion* to ensure the misclassification is minimized.

The perceptron criterion is given by

$$E_p(\boldsymbol{w}) = -\sum_{n \in \mathcal{M}} \boldsymbol{w}^T \phi_n t_n \tag{5-10}$$

where $\mathcal{M}$ is the set of all misclassified patterns.

5.2.2. Perceptron

- Suppose we seek for a linear pattern s.t. $\boldsymbol{w}^T\phi(\boldsymbol{x_n}) > 0$ for C_1.
- The patterns with $\boldsymbol{w}^T\phi(\boldsymbol{x_n})t_n > 0$ will also be followed. For $\boldsymbol{x_n}$ in C_2 will also follow the same pattern since $\boldsymbol{w}^T\phi(\boldsymbol{x_n}) < 0$ and $t_n = -1$ for all correct classification.
- In other words, misclassified $\boldsymbol{x_n}$, that is $n \in \mathcal{M}$, will have pattern of $\boldsymbol{w}^T\phi(\boldsymbol{x_n})t_n < 0$ for any classes.
- We need to therefore minimize the quantity $-\boldsymbol{w}^T\phi(\boldsymbol{x_n})t_n$ by applying the stochastic gradient descent algorithm,

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} - \eta\nabla E_p(\boldsymbol{x_n}).$$

- For each iteration, $y(\boldsymbol{x}, \boldsymbol{w})$ is evaluated.
- If the pattern is correctly classified, the weight does not change.

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} \tag{5-11a}$$

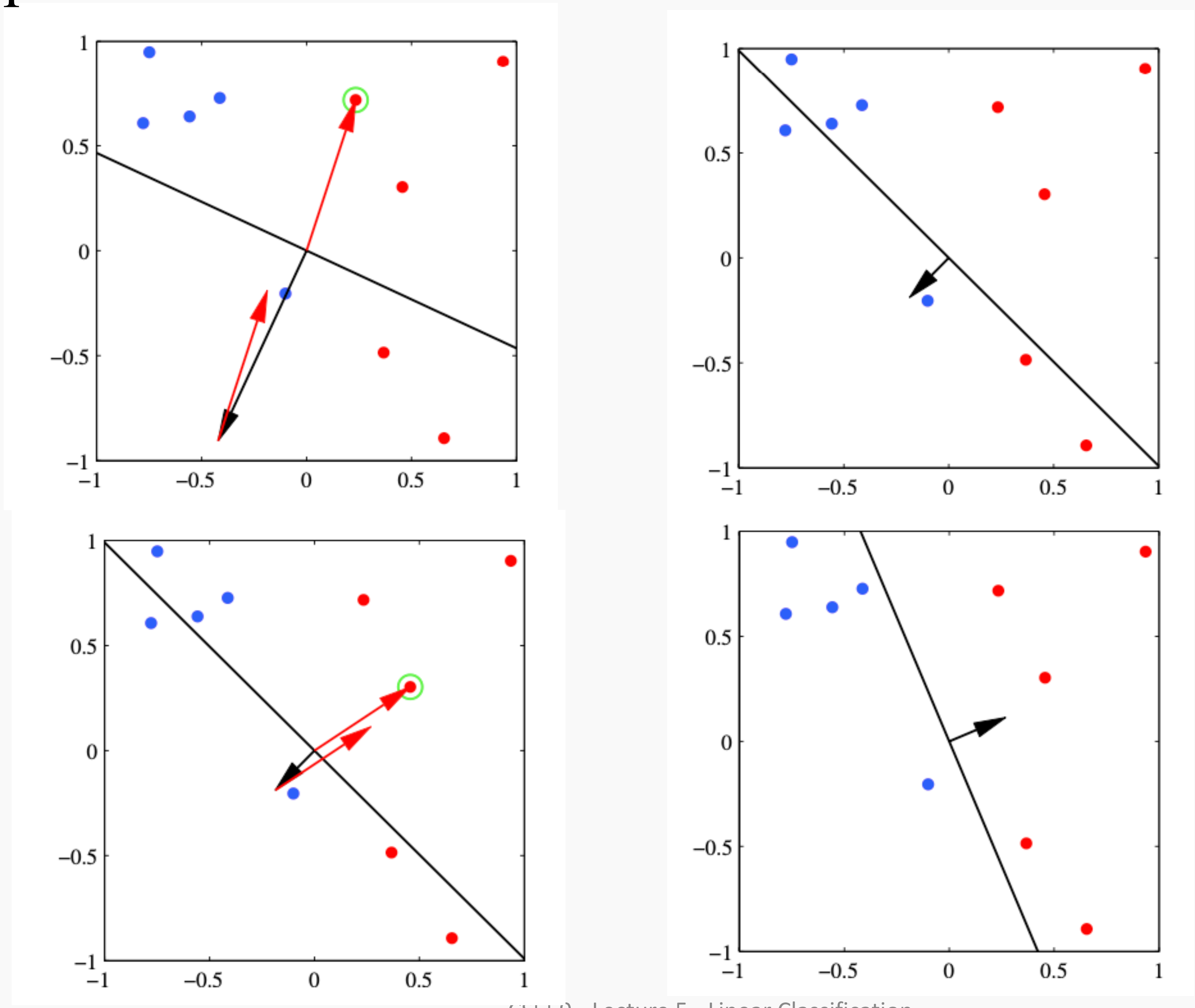
- If the classification is incorrect,
  - For C_1, we add the vector $\phi(\boldsymbol{x_n})$ to **w**.

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{\tau} + \boldsymbol{x}_n \tag{5-11b}$$

  - For C_2, we subtract vector $\phi(\boldsymbol{x_n})$ from **w**.

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{\tau} - \boldsymbol{x}_n \tag{5-11c}$$

5.2.2. Perceptron



5.2.2. Perceptron

Consider $\eta = 1$ and $||\phi_n t_n||^2 > 0$, the single update will reduce the error from the misclassification:

$$-\boldsymbol{w}^{(\tau+1)T}\phi_n t_n = -\boldsymbol{w}^{(\tau)T}\phi_n t_n - (\phi_n t_n)^T \phi_n t_n < -\boldsymbol{w}^{(\tau)T}\phi_n t_n.$$

- It will not impact the error from another misclassified pattern.
- It may change the result on previously corrected patterns.
- It does not guarantee the reduction of the total error function at each stage.
- However, if the training data is linearly separable, the perceptron guarantees the exact solution after the finite iteration – the *perceptron convergence theorem.*
- The perceptron may have many solutions depends on the parameter initialization.
- The learning algorithm will never converge if the training data is not linearly separable.

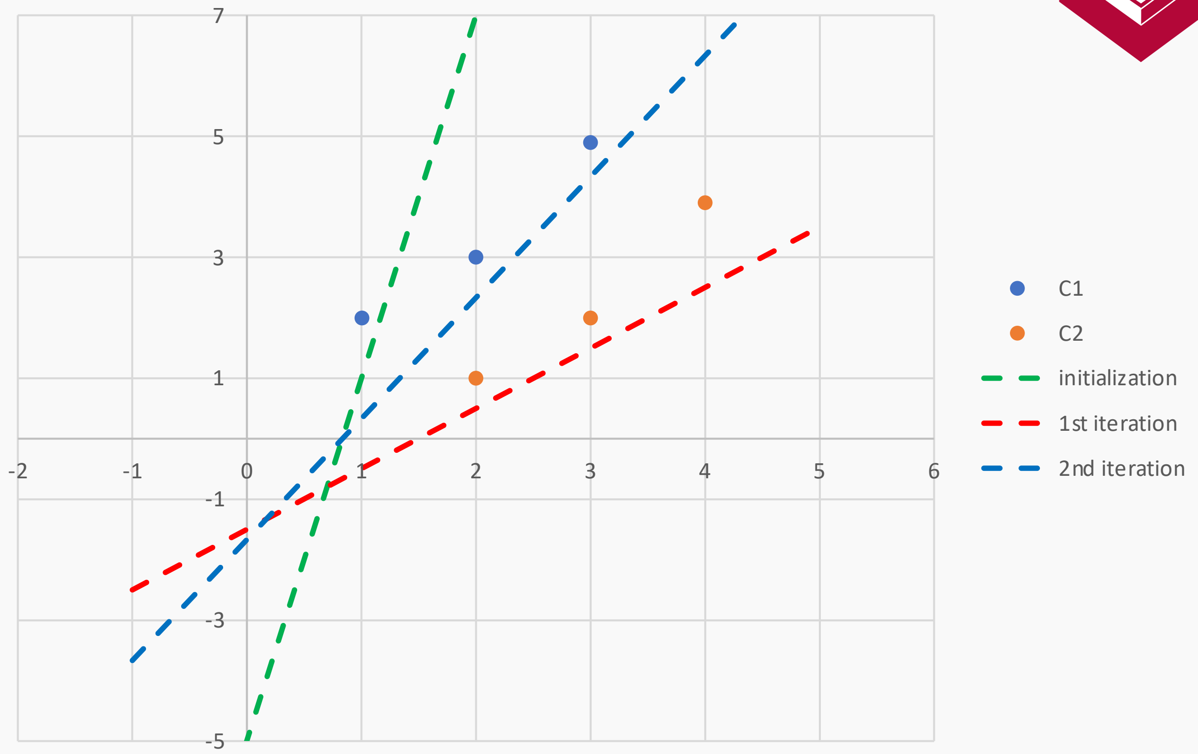
5.2.2. Perceptron

Class	(x_1, x_2)
1	(1,2), (2,3), (3,4.9)
-1	(2,1), (3,2), (4, 3.9)

Initial $\boldsymbol{w} = [\mathbf{5}, -\mathbf{6}, \mathbf{1}], y = w_0 + w_1 x_1 + w_2 x_2.$

	w_0	w_1	w_2
Initialization	5	-6	1
1st Iteration	6	-4	4
2nd Iteration	5	-6	3

$\hat{\boldsymbol{w}} = [\mathbf{0.598}, -\mathbf{0.717}, \mathbf{0.585}]$



	1	2	3	4	5	6
Initialization	1	-1	-1	-1	-1	-1
1st Iteration	1	1	1	1	1	1
2nd Iteration	1	1	1	-1	-1	-1

5.3. Probabilistic Generative Models

Consider a binary class classification, C_1 and C_2.
The posterior probability for C_1,

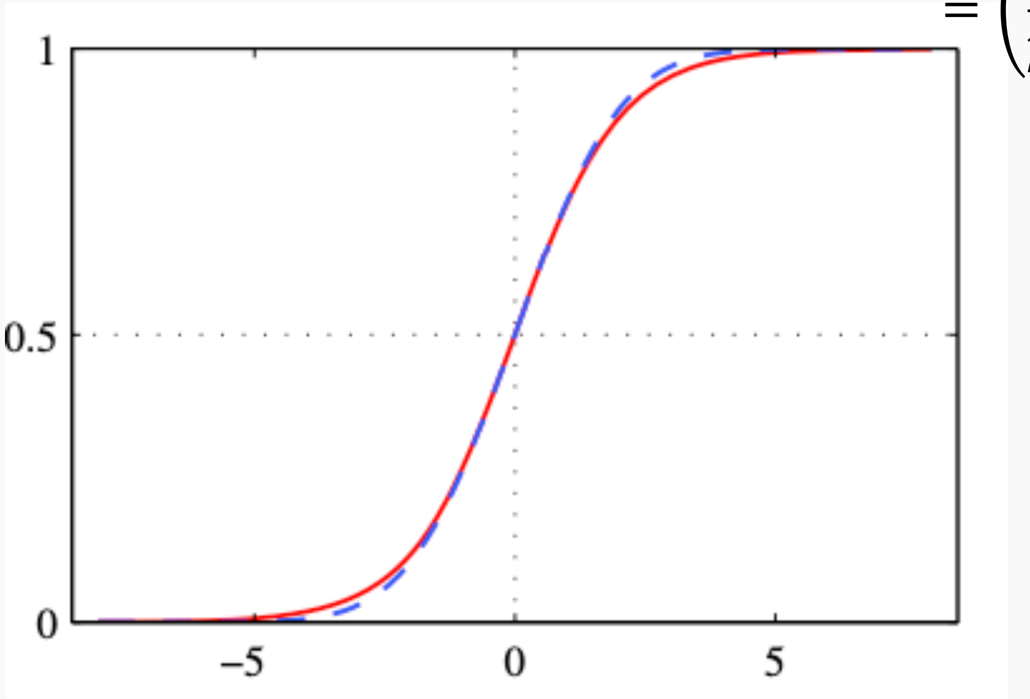
$$p(C_1|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_1)p(C_1)}{p(\boldsymbol{x}|C_1)p(C_1) + p(\boldsymbol{x}|C_2)p(C_2)} = \frac{1}{1+\exp(-a)} = \sigma(a) \qquad (5\text{-}12)$$

where

$$a = \ln \frac{p(\boldsymbol{x}|C_1)p(C_1)}{p(\boldsymbol{x}|C_2)p(C_2)}.$$

Note: $\exp(-a) = \exp\left(-\ln \frac{p(\boldsymbol{x}|C_1)p(C_1)}{p(\boldsymbol{x}|C_2)p(C_2)}\right) = \left(\frac{p(\boldsymbol{x}|C_1)p(C_1)}{p(\boldsymbol{x}|C_2)p(C_2)}\right)^{-1}$

$\sigma(a)$ is called the *logistic sigmoid.*



5.3. Probabilistic Generative Models

It follows the symmetry property

$$\sigma(-a) = 1 - \sigma(a).$$

The inverse of the logistic sigmoid is known as the *logit* function in the form of

$$a = \ln\left(\frac{\sigma}{1-\sigma}\right)$$

and represents to log of the ratio of $\ln[p(C_1|\boldsymbol{x})/p(C_2|\boldsymbol{x})]$ for the two classes, also known as the *log odds*.

5.3. Probabilistic Generative Models

For the case of $K > 2$ classes, the posterior probability of C_k is in the form

$$p(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)p(C_k)}{\sum_j p(\boldsymbol{x}|C_j)p(C_j)} = \frac{\exp a_k}{\sum_j \exp(a_j)}$$

which is known as the *normalized exponential* and can be regarded as a multiclass generalization of the logistic sigmoid and the quantities a_k are defined by

$$a_k = \ln p(\boldsymbol{x}|C_k)p(C_k)\,.$$

This is also known as the ***softmax function*** that smooths the max function:

$$p(C_k|\boldsymbol{x}) \approx 1$$
$$p(C_j|\boldsymbol{x}) \approx 0$$

for all $j \neq k$.

5.3. Probabilistic Generative Models

Assume the class-conditional densities are Gaussian and all classes share the same covariance matrix. The density for class C_k is

$$p(x|C_k) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_{\boldsymbol{k}})^T \boldsymbol{\Sigma}^{-\mathbf{1}} (\boldsymbol{x}-\boldsymbol{\mu}_{\boldsymbol{k}})\right\}.$$

Consider for binary classes,

$$p(C_1|\boldsymbol{x}) = \sigma\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + w_0\right)$$

Derivation:

Let the prior probability be $p(C_1) = \pi_1$ and $p(C_2) = \pi_2$.

5.3. Probabilistic Generative Models

$$p(C_1|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_1)p(C_1)}{p(\boldsymbol{x}|C_1)p(C_1) + p(\boldsymbol{x}|C_2)p(C_2)}$$

$$= \frac{\pi_1 \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^{\boldsymbol{T}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}(\boldsymbol{x}-\boldsymbol{\mu_1})\right\}}{\pi_1 \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^{\boldsymbol{T}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}(\boldsymbol{x}-\boldsymbol{\mu_1})\right\} + \pi_2 \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_2})^{\boldsymbol{T}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}(\boldsymbol{x}-\boldsymbol{\mu_2})\right\}}$$

The exponential term can be simplified as

$$-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^{\boldsymbol{T}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}(\boldsymbol{x}-\boldsymbol{\mu_1}) = -\frac{1}{2}\left(\boldsymbol{x}\boldsymbol{\Sigma}^{-\boldsymbol{1}}\boldsymbol{x} - 2\boldsymbol{\mu}_{\boldsymbol{1}}^{\boldsymbol{T}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}\boldsymbol{x} + \boldsymbol{\mu}_{\boldsymbol{1}}^{\boldsymbol{T}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}\boldsymbol{\mu_1}\right).$$

The first term in the right side is not associate with C_1 and it can be vanished. The equation above can be expressed as

$$\boldsymbol{\mu}_{\boldsymbol{1}}^{\boldsymbol{T}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_{\boldsymbol{1}}^{\boldsymbol{T}}\boldsymbol{\Sigma}^{-\boldsymbol{1}}\boldsymbol{\mu_1}.$$

5.3. Probabilistic Generative Models

The posterior probability then becomes

$$
\begin{aligned}
&= \frac{\pi_1 \exp\left\{\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1\right\}}{\pi_1 \exp\left\{\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1\right\} + \pi_2 \exp\left\{\boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2\right\}} \\
&= \frac{1}{1 + \frac{\pi_2}{\pi_1}\exp\left\{\boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \frac{1}{2}\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1\right\}} \\
&= \frac{1}{1 + \exp\left\{\boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \frac{1}{2}\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \ln\frac{\pi_2}{\pi_1}\right\}} \\
&= \frac{1}{1 + \exp\left\{-\left[(\boldsymbol{\mu}_1^T - \boldsymbol{\mu}_2^T)\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \frac{1}{2}\left(\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2\right) + \ln\frac{\pi_1}{\pi_2}\right]\right\}} \\
&= \frac{1}{1 + \exp(-a)}
\end{aligned}
$$

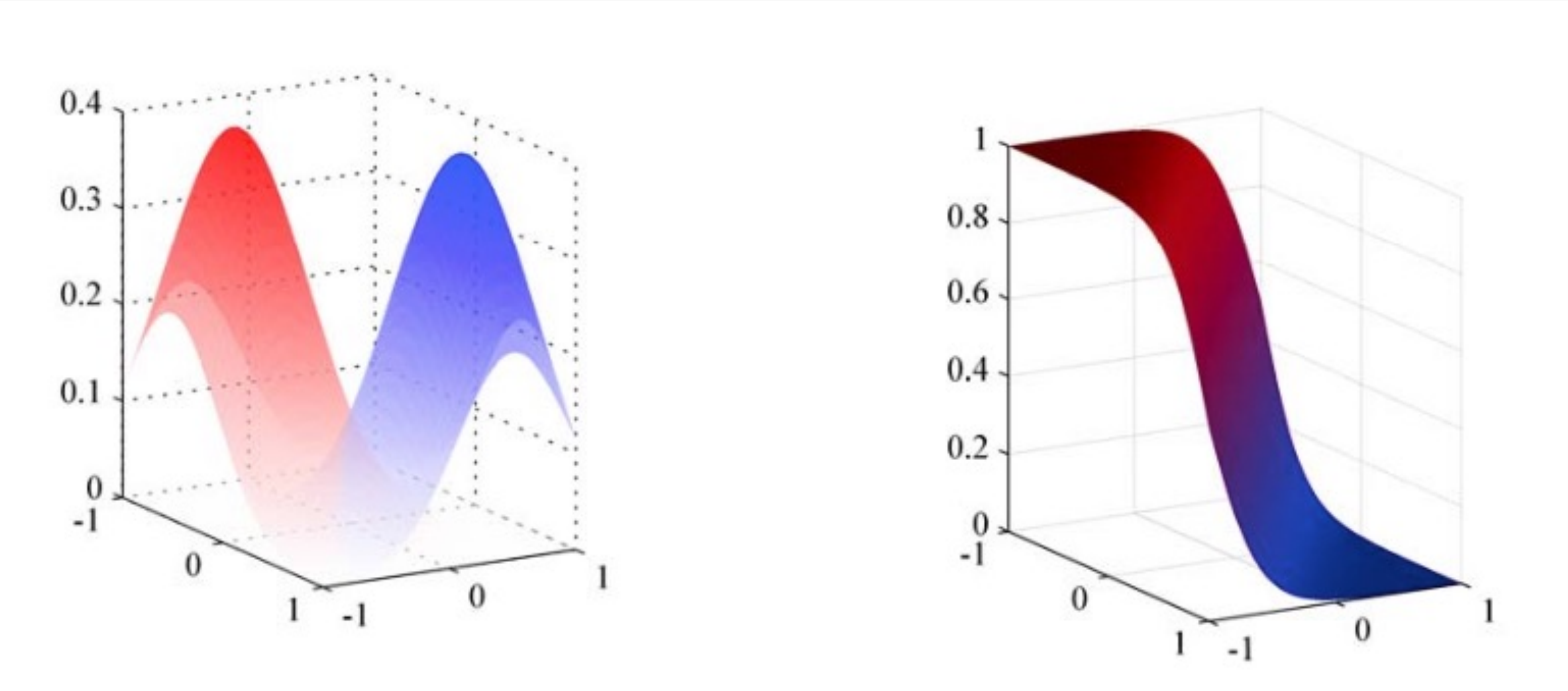
where

$$
\begin{aligned}
\boldsymbol{w}^T &= (\boldsymbol{\mu}_1^T - \boldsymbol{\mu}_2^T)\boldsymbol{\Sigma}^{-1}, \\
w_0 &= \frac{1}{2}\left(\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2\right) + \ln\frac{\pi_1}{\pi_2},
\end{aligned}
\qquad (5\text{-}13)
$$

and

$$a = \boldsymbol{w}^T\boldsymbol{x} + w_0.$$

5.3. Probabilistic Generative Models



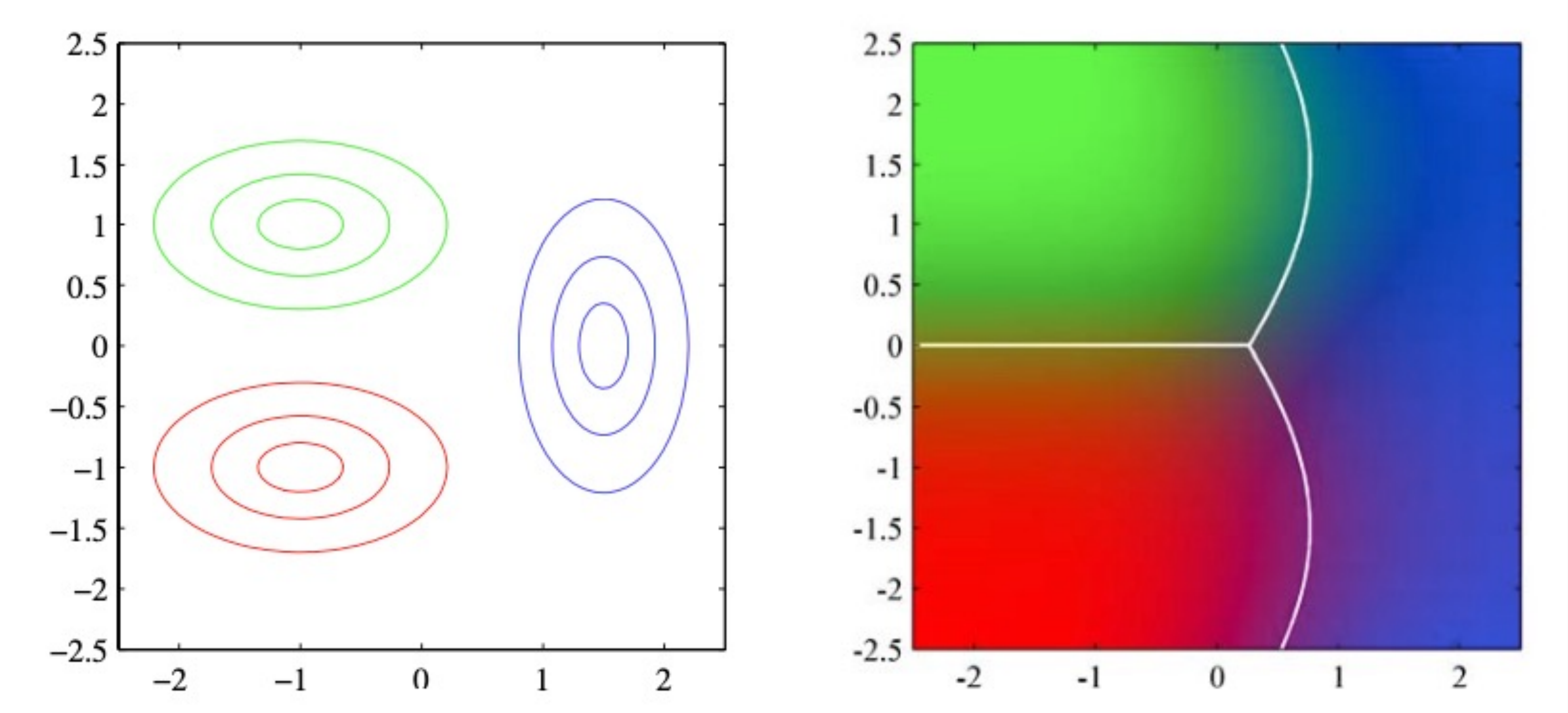
5.3. Probabilistic Generative Models

For the case of $K > 2$ classes,

$$a_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}$$

where $\boldsymbol{w}_{\boldsymbol{k}} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{\boldsymbol{k}}$ and $w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(C_k)$.

If each class has its own covariance matrix $\boldsymbol{\Sigma}_k$, then $\boldsymbol{x}\boldsymbol{\Sigma}_{\boldsymbol{k}}^{-1}\boldsymbol{x}$ term will not be vanished.



5.3. Probabilistic Generative Models

The parameters can be estimated using MLE if the data set comprises observes of $\mathbf{x}$ along with their corresponding class labels.

Consider a binary class dataset, $\{\boldsymbol{x}_n, t_n\}$ where $n = 1, \cdots, N$.

- Each class having a Gaussian class-conditional density with a shared Σ.
- $t_n = \{1,0\}$ for C_1 and C_2, respectively.
- Let the prior class probability be $p(C_1) = \pi$ and $p(C_2) = 1 - \pi$.

5.3. Probabilistic Generative Models

For a data point $\boldsymbol{x}_n$ from class C_1, $t_n = 1$ and hence

$$p(\boldsymbol{x}_n, C_1) = p(C_1)p(\boldsymbol{x}_n|C_1) = \pi\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}).$$

Similarly,

$$p(\boldsymbol{x}_n, C_2) = p(C_2)p(\boldsymbol{x}_n|C_2) = (1-\pi)\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}).$$

Then the likelihood function is given by

$$p(\boldsymbol{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^{N}[\pi\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n}[(1-\pi)\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$$

$$\propto \left(\pi exp\left[-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_1)\right]\right)^{t_n}\left((1-\pi)\ \exp\left[-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_2)\right]\right)^{1-t_n},$$

where log-likelihood terms that depend on π are

$$\sum_{n=1}^{N}\{t_n \ln \pi + (1-t_n)\ln(1-\pi)\}. \qquad (5\text{-}14)$$

5.3. Probabilistic Generative Models

Setting the derivative w.r.t. $\pi = 0$,

$$\pi = \frac{1}{N}\sum_{n=1}^{N} t_n = \frac{N_1}{N}$$

where N_1 is the number of data points in class C_1.

Consider the log-likelihood function terms that depend on $\boldsymbol{\mu}_1$,

$$\sum_{n=1}^{N} t_n \ln \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = -\frac{1}{2}\sum_{n=1}^{N} t_n(\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_1) + Const.$$

The derivative w.r.t. $\boldsymbol{\mu}_1$ will have

$$\boldsymbol{\mu}_1 = \frac{1}{N_1}\sum_{n=1}^{N} t_n \boldsymbol{x}_n ,$$

the mean of all the input vectors $\boldsymbol{x}_n$ assigned to C_1.

Similarly,

$$\boldsymbol{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N} (1 - t_n)\boldsymbol{x}_n .$$

5.3. Probabilistic Generative Models

The covariance then can be estimated from the derivative of log-likelihood w.r.t. $\boldsymbol{\Sigma}$,

$$\boldsymbol{\Sigma} = \frac{N_1}{N}\mathbf{S}_1 + \frac{N_2}{N}\boldsymbol{S}_2$$

where

$$\mathbf{S}_1 = \frac{1}{N_1}\sum_{n\in C_1}(\boldsymbol{x}_n - \boldsymbol{\mu}_1)(\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T$$

and

$$\boldsymbol{S}_2 = \frac{1}{N_2}\sum_{n\in C_2}(\boldsymbol{x}_n - \boldsymbol{\mu}_2)(\boldsymbol{x}_n - \boldsymbol{\mu}_2)^T .$$

5.3. Probabilistic Generative Models

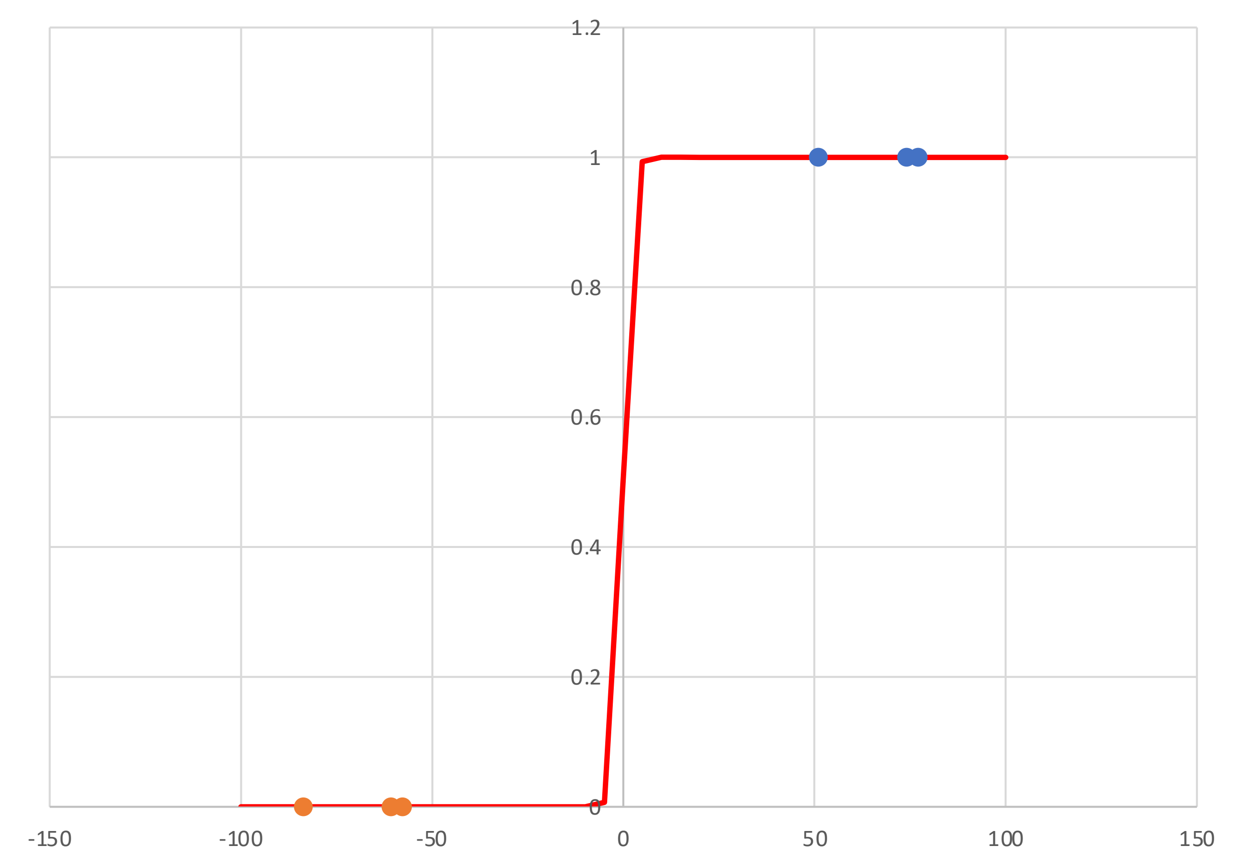
Class	(x_1, x_2)
1	(1,2), (2,3), (3,4.9)
-1	(2,1), (3,2), (4, 3.9)

$$\mathbf{\Sigma} = \begin{bmatrix} 0.667 & 0.967 \\ 0.967 & 1.467 \end{bmatrix}$$

$$\mathbf{\Sigma}^{-1} = \begin{bmatrix} 48.22 & -32.22 \\ -32.22 & 22.22 \end{bmatrix}$$

$$\boldsymbol{w} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\mathbf{\Sigma}^{-1} = [48.667, -80.444, 54.444]$$

$$\widehat{\boldsymbol{w}} = \frac{\boldsymbol{w}}{|\boldsymbol{w}|} = [\mathbf{0.448}, \mathbf{-0.828}, \mathbf{0.560}]$$



5.3. Probabilistic Generative Models

- Consider a case of discrete feature values x_i.
- For the simplicity, let $x_i \in \{0,1\}$ and there are D many inputs.
- Assume the *naïve Bayes*, **the feature values are independent.**
- The class-conditional distribution is

$$p(\boldsymbol{x}|C_k) = \prod_{i=1}^{D} \mu_{ki}^{x_i}(1-\mu_{ki})^{1-x_i}.$$

- Substituting into the quantities a_k,

$$a_k(\boldsymbol{x}) = \sum_{n=1}^{D} \{x_i \ln \mu_{ki} + (1-x_i)\ln(1-\mu_{ki})\} + \ln p(C_k)$$

- which is the linear function of the input values x_i.

5.3. Probabilistic Generative Models

Each class C_k is described by its own linear model

$$y_k(\boldsymbol{x}) = w_k^T \boldsymbol{x} + w_{k0} = \widetilde{\boldsymbol{w}}_k^T \widetilde{\boldsymbol{x}}$$

where $k = 1, \dots, K$, $\widetilde{\boldsymbol{w}}_k = \left(w_{k0}, \boldsymbol{w}_k^T\right)^T$, and $\widetilde{\boldsymbol{x}}$ is the corresponding augmented input vector $(1, \boldsymbol{x}^T)^T$. In the matrix formation, the expression becomes much simpler as below

$$\boldsymbol{y}(\boldsymbol{x}) = \widetilde{\boldsymbol{W}}^{\boldsymbol{T}} \widetilde{\boldsymbol{x}}.$$

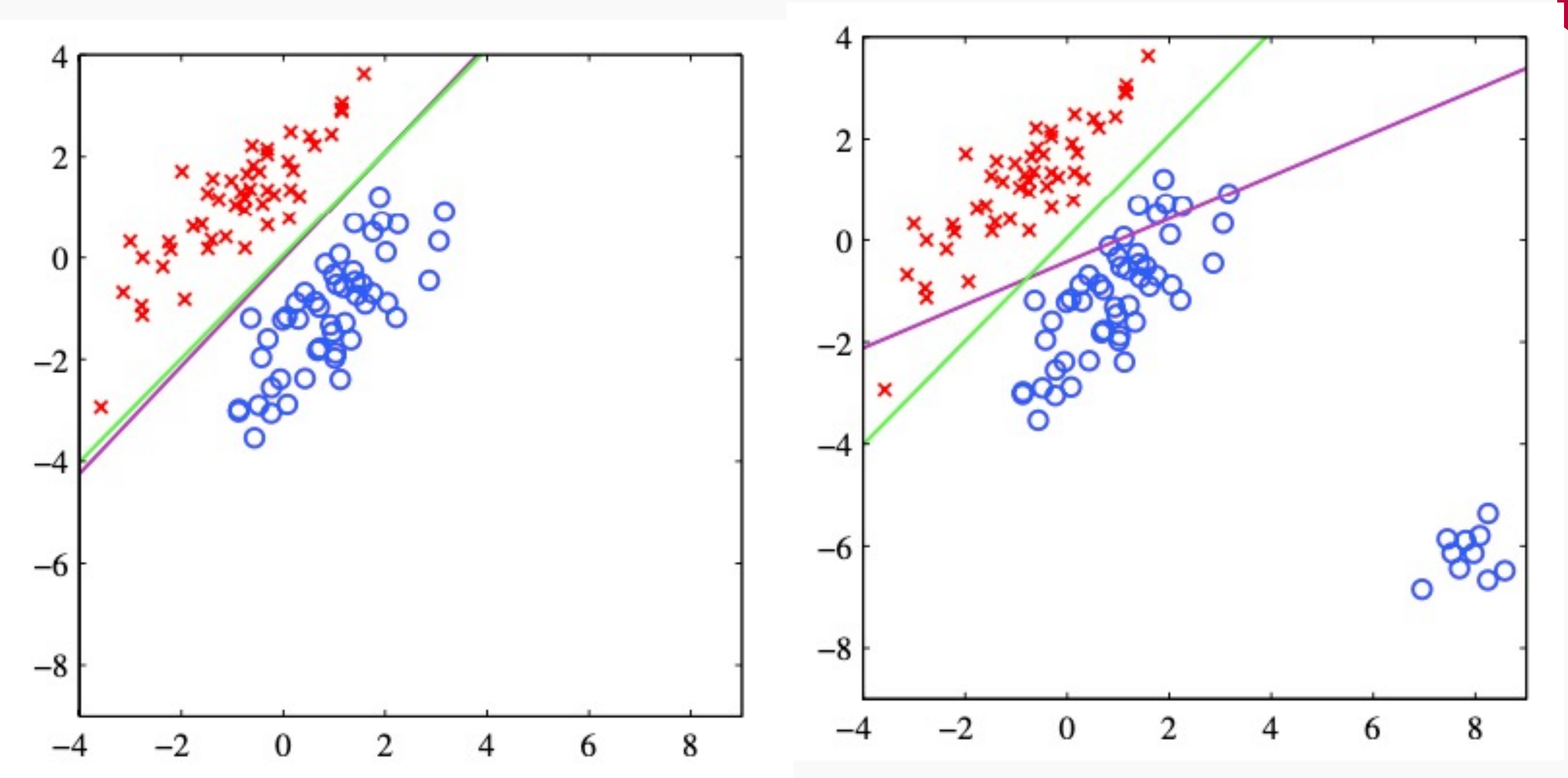
Consider a training data set $\{\boldsymbol{x_n}, \boldsymbol{t_n}\}$ where $n = 1, \dots, N$, the *sum-of-squares* error function

$$E_D\left(\widetilde{\boldsymbol{W}}\right) = \frac{1}{2} \mathrm{Tr}\left\{\left(\widetilde{\mathbf{X}}\widetilde{\boldsymbol{W}} - \boldsymbol{T}\right)^T \left(\widetilde{\mathbf{X}}\widetilde{\boldsymbol{W}} - \boldsymbol{T}\right)\right\}$$

can set the derivative w.r.t. $\widetilde{\boldsymbol{W}}$ to zero to have

$$\widetilde{\boldsymbol{W}} = \left(\widetilde{\boldsymbol{X}}^T \widetilde{\boldsymbol{X}}\right)^{-1} \widetilde{\boldsymbol{X}} \boldsymbol{T}.$$

5.3. Probabilistic Generative Models



5.4 Probabilistic Discriminative Models – Logistic Regression

Consider a binary class problem.
Assume the posterior probability of class C_1 can be written as a logistic sigmoid acting on a linear function of the feature vector ϕ,

$$p(C_1|\phi) = y(\phi) = \sigma(\boldsymbol{w}^T\phi)$$

with $p(C_2|\phi) = 1 - p(C_1|\phi)$. This model is known as *logistic regression*.

The parameters of the logistic regression model can be determined using maximum likelihood,

$$\frac{d\sigma}{da} = \sigma(1-\sigma).$$

5.4 Probabilistic Discriminative Models – Logistic Regression

For a data set $\{\phi_n, t_n\}$, where

- $t_n \in \{0,1\}$
- $\phi_n = \phi(\boldsymbol{x}_n)$ with $n = 1, \dots, N$.

The likelihood function can be written

$$p(\boldsymbol{t}|\boldsymbol{w}) = \prod_{n=1}^{N} y_n^{t_n}\{1 - y_n\}^{1-t_n}$$

where $\boldsymbol{t} = (t_1, \dots, t_N)^T$ and $y_n = p(C_1|\phi_n)$.
The *cross-entropy* error function is the negative log-likelihood in the form

$$E(\boldsymbol{w}) = -\ln p(\boldsymbol{t}|\boldsymbol{w}) = -\sum_{n=1}^{N}\{t_n \ln y_n + (1 - t_n)\ln(1 - y_n)\}$$

where $y_n = \sigma(\mathbf{w}^T\phi_n)$ and the gradient of the error function w.r.t. **w** is

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N}(y_n - t_n)\phi_n\,.$$

The maximum likelihood solution occurs when the hyperplane corresponding to $\sigma = 0.5$, equivalent to $\boldsymbol{w}^T\phi = 0$.

5.4 Probabilistic Discriminative Models – Logistic Regression

For the precise measurement, the error function can be minimized by an efficient iterative technique based on the *Newton-Raphson* iterative optimization, which uses a local quadratic approximation to the log-likelihood function in the form of

$$\boldsymbol{w}^{(new)} = \boldsymbol{w}^{(old)} - \boldsymbol{H}^{-1}\nabla E(\boldsymbol{w})$$

where **H** is the Hessian matrix whose elements are the second derivatives of $E(\boldsymbol{w})$ w.r.t. **w**:

$$\boldsymbol{H} = \nabla\nabla E(\boldsymbol{w}) = \nabla\left(\sum_{n=1}^{N}(\boldsymbol{w}^T\phi_n - t_n)\phi_n\right)$$

$$= \nabla\left(\boldsymbol{\phi}^T\boldsymbol{\phi}\boldsymbol{w} - \boldsymbol{\phi}^T\boldsymbol{t}\right) = \boldsymbol{\phi}^T\boldsymbol{\phi}.$$

5.4 Probabilistic Discriminative Models – Logistic Regression

Alternatively, we can update the cross-entropy error function as

$$\boldsymbol{H} = \nabla\left(\boldsymbol{\phi}^{\boldsymbol{T}}(\boldsymbol{y}-\boldsymbol{t})\right) = \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^T = \boldsymbol{\phi}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{\phi}$$

where **R** is the $N\times N$ weight matrix with elements $R_{nn} = y_n(1-y_n)$. The error function is not quadratic longer but is not constant.

The Newton-Raphson update formula for the logistic regression model then becomes

$$\boldsymbol{w}^{new} = \boldsymbol{w}^{old} - \left(\boldsymbol{\phi}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{\phi}\right)^{-1}\boldsymbol{\phi}^{\boldsymbol{T}}(\boldsymbol{y}-\boldsymbol{t})$$
$$= \left(\boldsymbol{\phi}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{\phi}\right)^{-1}\{\boldsymbol{\phi}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{\phi}\boldsymbol{w}^{(old)} - \boldsymbol{\phi}^{\boldsymbol{T}}(\boldsymbol{y}-\boldsymbol{t})\}$$
$$= \left(\boldsymbol{\phi}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{\phi}\right)^{-1}\boldsymbol{\phi}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{z}$$

where **z** is an N-dimensional vector with elements

$$\boldsymbol{z} = \boldsymbol{\phi}\boldsymbol{w}^{(\boldsymbol{old})} - \boldsymbol{R}^{-1}(\boldsymbol{y}-\boldsymbol{t}).$$

5.4 Probabilistic Discriminative Models – Logistic Regression

w0	1	-1	1	0.577350269	-0.57735027	0.577350269	
likelihood	-2.93486703						
z	2	2	2.9	0	0	0.9	
sigmoid	0.880797078	0.880797078	0.947846437	0.5	0.5	0.710949503	
y	1	1	1	0	0	1	
grad	-1.4203901	-4.82972856	-3.42113599				
H	0.964921047	2.535282692	2.318643613	H_inv	9.296241883	-2.81430131	-0.36016317
	2.535282692	7.507874965	6.772426941		-2.81430131	1.979349879	-0.9930948
	2.318643613	6.772426941	6.927476275		-0.36016317	-0.9930948	1.235767788
Delta	1.620188714	-2.16480455	1.080220871				
w_update	2.620188714	-3.16480455	2.080220871	0.568951952	-0.6872107	0.451702475	

$$\nabla E = \left(\boldsymbol{t} - \sigma(\boldsymbol{w}^T\boldsymbol{x})\right)\boldsymbol{x}$$

$$\boldsymbol{H} = \begin{bmatrix} \frac{\partial^2 \nabla E}{\partial x_0^2} & \frac{\partial^2 \nabla E}{\partial x_0 \partial x_1} & \frac{\partial^2 \nabla E}{\partial x_0 \partial x_2} \\ \frac{\partial^2 \nabla E}{\partial x_1 \partial x_0} & \frac{\partial^2 \nabla E}{\partial x_1^2} & \frac{\partial^2 \nabla E}{\partial x_1 \partial x_2} \\ \frac{\partial^2 \nabla E}{\partial x_2 \partial x_0} & \frac{\partial^2 \nabla E}{\partial x_2 \partial x_1} & \frac{\partial^2 \nabla E}{\partial x_2^2} \end{bmatrix}$$

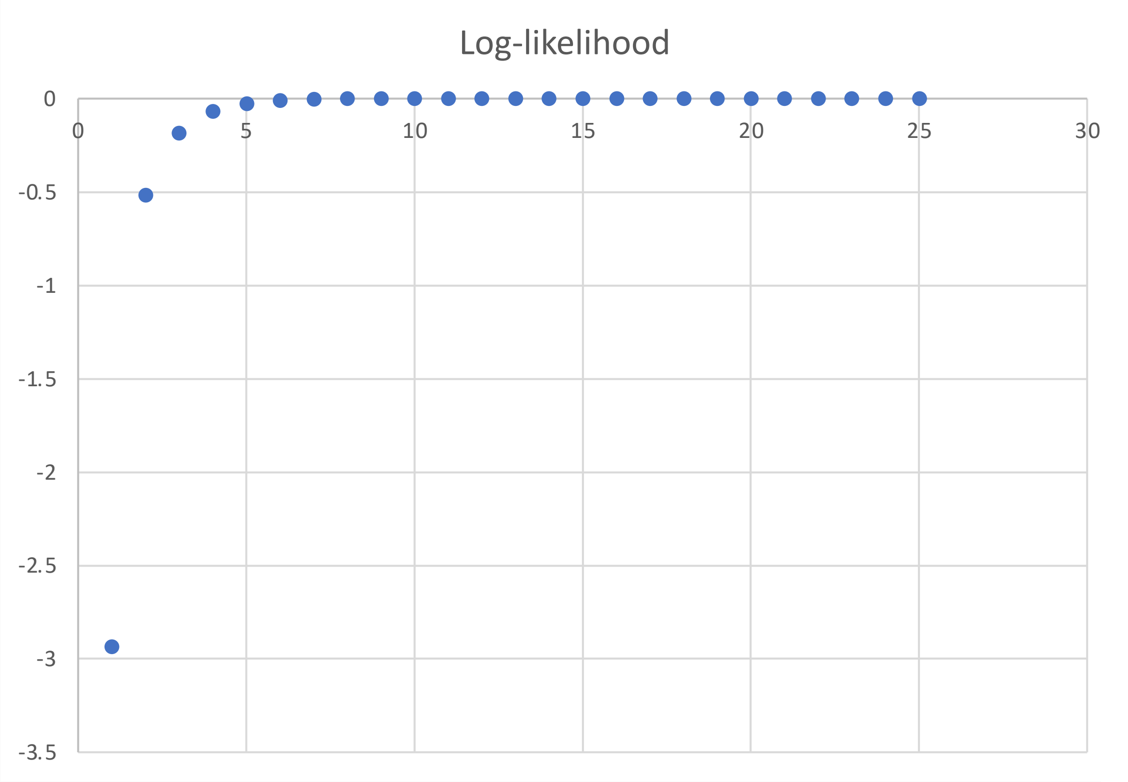
After 6 iterations:

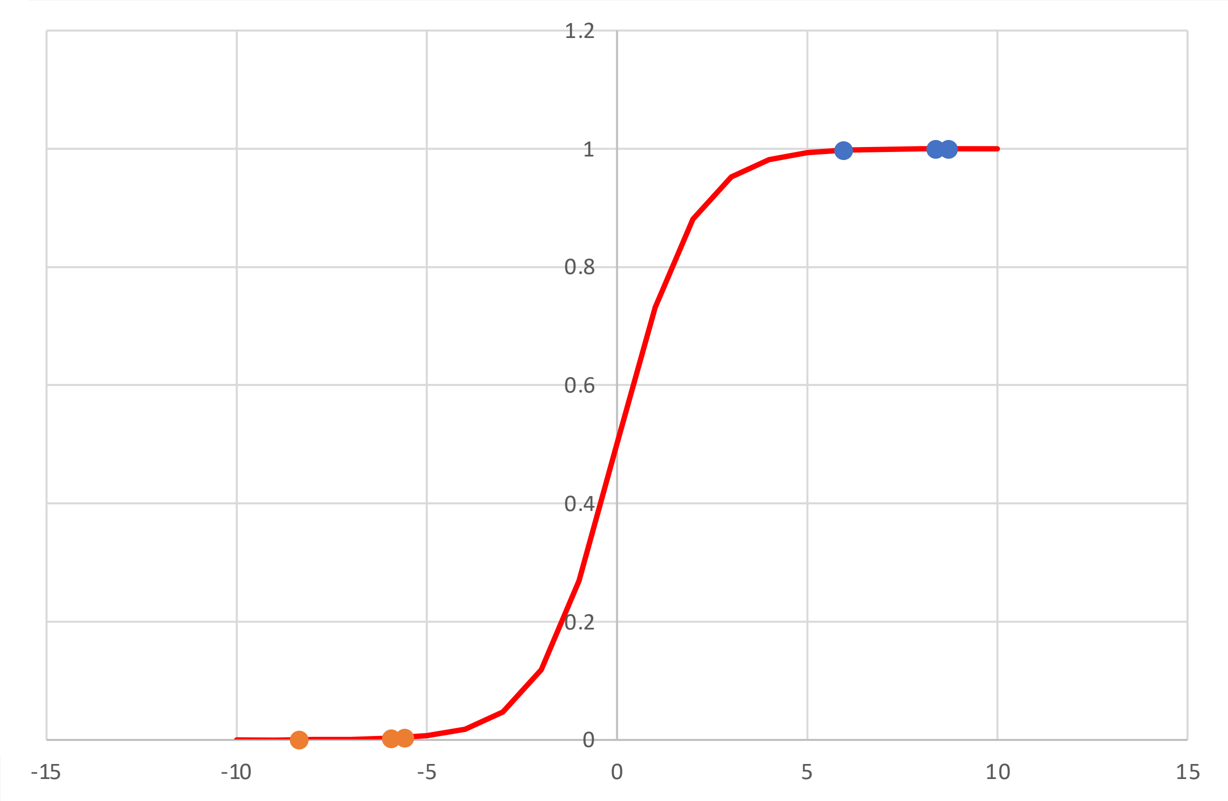
w_update	5.713665154	-8.53726067	5.768785273	0.484958884	-0.72461726	0.489637316	
likelihood	-0.00960079						
z	8.713975027	5.945499627	8.368930973	-5.59207092	-8.36054632	-5.93711497	
sigmoid	0.999835753	0.99738924	0.99976809	0.00371346	0.000233862	0.002632685	
y	1	1	1	0	0	0	
grad	-0.00357309	-0.01257775	-0.00515152				
H	0.009559251	0.024671452	0.02368409	H_inv	949.3811091	-344.421922	17.81857682
	0.024671452	0.071581703	0.069124327		-344.421922	263.5807941	-135.555362
	0.02368409	0.069124327	0.074231849		17.81857682	-135.555362	134.0146649
Delta	0.848035592	-1.38628593	0.950934258				

$$\boldsymbol{\Delta} = \boldsymbol{H}^{-1}\nabla E$$

$$\boldsymbol{w}^{(new)} = \boldsymbol{w}^{(old)} + \boldsymbol{\Delta}$$

5.4 Probabilistic Discriminative Models – Logistic Regression





5.4 Probabilistic Discriminative Models – Logistic Regression

The posterior probabilities are given by a softmax transformation of linear functions of the feature variables

$$p(C_k|\phi) = y_k(\phi) = \frac{\exp(\boldsymbol{w}_k^T \phi)}{\sum_j \exp(\boldsymbol{w}_k^T \phi)}.$$

The derivative of y_k w.r.t. the activation function $a_k = \boldsymbol{w}_k^T \phi$ is

$$\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j)$$

where I_{kj} are the elements of the identity matrix.

5.4 Probabilistic Discriminative Models – Logistic Regression

The target variables t_{nk} form a $N \times K$ matrix $\boldsymbol{T}$, can express the likelihood function as

$$p(\boldsymbol{T}|\boldsymbol{w}_1, \dots, \boldsymbol{w}_k) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(C_k|\phi_n)^{t_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}}$$

where $y_{nk} = y_k(\phi_n)$.

The negative logarithm gives the error function as

$$E(\boldsymbol{w}_1, \dots, \boldsymbol{w}_K) = -\ln p(\boldsymbol{T}|\boldsymbol{w}_1, \dots, \boldsymbol{w}_k) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_{nk}$$

which is known as the *cross-entropy* error function for the multiclass classification problem.

5.4 Probabilistic Discriminative Models – Logistic Regression

The gradient of the error function w.r.t. one of the parameter vectors $\boldsymbol{w}_j$ result the derivative of the softmax function,

$$\nabla_{\boldsymbol{w}_j} E(\boldsymbol{w}_1, \dots, \boldsymbol{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj}) \phi_n$$

and the parameter vectors can be found using via the iterative least square following Newton-Raphson method where the Hessian matrix is

$$\nabla_{\boldsymbol{w}_k} \nabla_{\boldsymbol{w}_j} = E(\boldsymbol{w}_1, \dots, \boldsymbol{w}_K) = -\sum_{n=1}^{N} y_{nk} (I_{kj} - y_{nj}) \phi_n \phi_n^T .$$