Discriminant functions

Two classes
Multiple classes
Least squares for classification
Fisher's linear discriminant
Relation to least squares
Fisher's discriminant for multiple classes
The perceptron

Discriminant functions

Linear models for classification

Francesco Corona
Linear models for classification


## Linear models for classification

A class of regression models with simple analytical/computational properties

- The analogous class of models for solving classification problems

### Remark

The goal in classification

- Take a $D$-dimensional input vector $\mathbf{x}$
- Assign it to one of $K$ discrete classes $C_k$, $k = 1, \ldots, K$

In the most common scenario, the classes are taken to be disjoint

- each input is assigned to one and only one class
Linear models for classification (cont.)

Remark

The input space is divided into decision regions

The boundaries of the decision regions
- decision boundaries
- decision surfaces

With linear models for classification, the decision surfaces are linear functions

- These decision surfaces are linear functions of the input vector $x$
- $(D - 1)$-dimensional hyperplanes, in the $D$-dimensional input space

Classes that can be separated well by linear surfaces are linearly separable
For regression problems, the target variable $t$ was a vector of real numbers

- In classification, there are various ways of representing class labels

There is a single target variable $t \in \{0, 1\}$

- $t = 1$ represents class $C_1$
- $t = 0$ represents class $C_2$

It is the probability of class $C_1$, with the probability only taking values of 0 and 1

There is a $K$-long target vector $t$, such that

- If the class is $C_j$, all elements $t_k$ of $t$ are zero for $k \neq j$ and one for $k = j$
- $t_k$ is the probability that the class is $C_k$

$K = 6$ and $C_k = 4$, then $t = (0, 0, 0, 1, 0, 0)^T$
The simplest approach to classification problems is through construction of a **discriminant function** that directly assigns each vector $x$ to a specific class.

More powerful is to **model the conditional probability distribution** $p(C_k|x)$ in an inference stage, and use this distribution to make optimal decisions.

- **Discriminative modelling**: $p(C_k|x)$ can be modelled directly, using a parametric model and optimising the parameters using a training set.
- **Generative modelling**: We model class-conditional densities $p(x|C_k)$ and the prior probabilities $p(C_k)$ for the classes, and we compute the posterior probabilities using Bayes’ theorem

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} \quad (1)$$
Discriminant functions

Two classes
Multiple classes
Least squares for classification
Fisher’s linear discriminant
Relation to least squares
Fisher’s discriminant for multiple classes
The perceptron

Discriminant functions
Linear models for classification
We start with the construction of classifiers based on discriminant functions.

In linear regression models
- The model prediction $y(x, \mathbf{w})$ is a linear function of parameters $\mathbf{w}$
- In the simplest case, the model is also linear in the inputs

$$y(x) = \mathbf{w}^T x + w_0, \quad \text{with } y \text{ a real number}$$

In classification problems, we would want to predict discrete class labels
- More generally, posterior probabilities that are in $(0, 1)$

We can achieve this with a generalisation of the linear regression model

$$y(x) = f(\mathbf{w}^T x + w_0)$$

(2)

We transform the linear function of $\mathbf{w}$ using some nonlinear function $f(\cdot)$
Discriminant functions (cont.)

$$y(x) = f(w^T x + w_0)$$

Function $f(\cdot)$ is the **activation function** and its inverse is the **link function**

Decision surfaces correspond to $y(x) = \text{constant}$ so $w^T x + w_0 = \text{constant}$
- Decision surfaces are linear functions of $x$, even if $f(\cdot)$ is nonlinear
Discriminant functions (cont.)

Remark

This is the class of models known as **generalised linear models**

- They are not linear in the parameters, because of $f(\cdot)$
- More complex analytical and computational properties
Outline

1 Discriminant functions

Two classes
Multiple classes
Least squares for classification
Fisher’s linear discriminant
Relation to least squares
Fisher’s discriminant for multiple classes
The perceptron
Two classes

Discriminant functions
Two classes

A simple linear discriminant function is a linear function of the input vector $\mathbf{x}$

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

- $\mathbf{w}$ is the weight vector
- $w_0$ is a bias term
- $-w_0$ is a threshold

An input vector $\mathbf{x}$ is assigned to class $C_1$ if $y(\mathbf{x}) \geq 0$ and to class $C_2$ otherwise.

The corresponding decision boundary is defined by the relationship $y(\mathbf{x}) = 0$
- $(D - 1)$-dimensional hyperplane within the $D$-dimensional input space
Consider two points $x_A$ and $x_B$ on the decision boundary

$$\begin{align*}
  y(x_A) &= w^T x_A = 0 \\
  y(x_B) &= w^T x_B = 0
\end{align*}$$

$\rightarrow \quad w^T (x_A - x_B) = 0 \quad \rightarrow \quad w \perp (x_A - x_B)$

Vector $w$ is orthogonal to every vector in the boundary

- $w$ sets the orientation of the boundary

If $x$ is a point of the decision surface, $y(x) = 0$ and $w^T x = -w_0$ and

$$\frac{w^T x}{||w||} = -\frac{w_0}{||w||} \quad (4)$$

This is the normal distance from the origin to the decision surface

- $w_0$ sets the location of the decision boundary
Two classes (cont.)

The value of $y(x)$ gives a signed measure of perpendicular distance too

- The distance from the point $x$ to the decision surface

Let $x$ be any point and $x_\perp$ its orthogonal projection onto the boundary

$$x = x_\perp + r \frac{w}{||w||} \quad (5)$$

Multiply both sides by $w^T$ and adding $w_0$ with $y(x) = w^T x + w_0$ and $y(x_\perp) = w^T x_\perp + w_0 = 0$

$$r = \frac{y(x)}{||w||} \quad (6)$$
As with linear models for regression, it is sometimes convenient to use a more compact notation and introduce an additional dummy input value $x_0 = 1$

- We define $\tilde{w} = (w_0, w)$ and $\tilde{x} = (x_0, x)$, so that

$$y(x) = \tilde{w}^T \tilde{x} \quad (7)$$

**Remark**

The decision surface is now a $D$-dimensional hyperplane passing through the origin of the $(D + 1)$-dimensional expanded input space.
Multiple classes

Discriminant functions
Discriminant functions

UFC/DC
ATAI-I (CK0146)
2017.1

Multiple classes

Now consider the extension of linear discriminants to the case of $K > 2$ classes

Consider the use of $K - 1$ classifiers, each of which solves a two-class problem

- Separate points in class $C_k$ from points not in $C_k$
- It is a **one-versus-the-rest** classifier

This approach leads to regions of the input space that are ambiguously classified

- By definition, the green area cannot be classified as both $C_1$ and $C_2$
Multiple classes (cont.)

Consider $K(K - 1)/2$ classifiers, one for every possible pair of classes

- Separate points in class $C_k$ from points in $C_{j \neq k}$, with $j = 1, \ldots, K$
- It is a **one-versus-one** classifier
- Majority voting classifies them

Also this approach leads to regions of the input space that are ambiguously classified
Multiple classes (cont.)

We can avoid these difficulties by considering a single $K$-class discriminant

- with $K$ linear functions of the form

$$y_k(x) = w_k^T x + w_{k0} \quad (8)$$

A point $x$ is then assigned to class $C_k$, if $y_k(x) > y_j(x)$, for all $j \neq k$

- The boundary between class $C_k$ and class $C_j$ is $y_k(x) = y_j(x)$ or

$$\begin{align*}
(w_k - w_j)^T x + (w_{k0} - w_{j0}) &= 0 \\
&= 0 \quad (9)
\end{align*}$$

- A $(D - 1)$-dimensional hyperplane

It has the same form of the decision boundary for the two-classes case
Multiple classes (cont.)

The decision regions from such a discriminant are singly connected and convex.

Consider two point \( x_A \) and \( x_B \) in region \( \mathcal{R}_k \)

- Any point \( \hat{x} \) on the segment between them can be expressed as a their convex combination

\[
\hat{x} = \lambda x_A + (1 - \lambda) x_B, \quad \lambda \in [0, 1] \quad (10)
\]

Because of the linearity of the discriminant function

\[
y_k(\hat{x}) = \lambda y_k(x_A) + (1 - \lambda) y_k(x_B) \quad (11)
\]

Because \( x_A \) and \( x_B \) are in \( \mathcal{R}_k \), we have \( y_k(x_A) > y_j(x_A) \) and \( y_k(x_B) > y_j(x_B) \), for all \( j \neq k \), and hence \( y_k(\hat{x}) > y_j(\hat{x}) \) so \( \hat{x} \) also lies within the region \( \mathcal{R}_k \).
Least squares for classification

Discriminant functions
In regression, models that are linear functions of the parameters could be solved for the parameters using a simple closed-form

- Minimisation of the sum-of-squares error function

Question is, would this work also for classification problems?

- We consider a classification problem with $K$ classes, using a 1-of-$K$ binary encoding for the target vector $t$
Least squares for classification (cpnt.)

Remark

One justification is ‘least squares approximates the conditional expectation $\mathbb{E}[t|x]$ on the target values given the input vector’

- Here, a vector of posterior class probabilities

These probabilities happen to be very poorly approximated

- They can take values outside $(0, 1)$
Least squares for classification (cont.)

Each class \( C_k \) is described by its own linear model in the form
\[
y_k(x) = \mathbf{w}_k^T x + w_{k0}, \quad k = 1, \ldots, K
\]  

(12)

The \( K \) models can be grouped using vector notation to obtain
\[
y(x) = \mathbf{	ilde{w}}^T \mathbf{	ilde{x}}
\]

(13)

- \( \mathbf{	ilde{W}} \) is a matrix whose \( k \)-th column comprises the \((D + 1)\)-dimensional vector \( \mathbf{	ilde{w}}_k = (w_{k0}, \mathbf{w}_k^T)^T \)
- \( \mathbf{	ilde{x}} \) is the corresponding augmented input vector \( (1, x^T)^T \) with the dummy input \( x_0 = 1 \)

A new input \( x \) is assigned to the class for which \( y_k = \mathbf{	ilde{w}}_k^T \mathbf{	ilde{x}} \) is largest

By minimising the sum-of-squares error function, get the parameter matrix \( \mathbf{	ilde{W}} \)
Least squares for classification (cont.)

Consider a training data set \( \{x_n, t_n\}_{n=1}^N \) and define matrix \( T \) and matrix \( \tilde{X} \)

- The \( n \)-th column of \( T \) is vector \( t_n^T \)
- The \( n \)-th row of \( \tilde{X} \) is vector \( \tilde{x}_n^T \)

The sum-of-squares error function can be then written as

\[
E_D(\tilde{X}) = \frac{1}{2} \text{Tr} \left( (\tilde{X}\tilde{W} - T)^T (\tilde{X}\tilde{W} - T) \right) \tag{14}
\]

By setting to zero the derivative of \( E_D(\tilde{W}) \) wrt \( \tilde{W} \) and rearranging

\[
\tilde{W} = (\tilde{X}^T\tilde{X})^{-1} \tilde{X}^T T = \tilde{X}^\dagger T \tag{15}
\]

where \( \tilde{X}^\dagger \) is the Moore-Penrose pseudo-inverse of the matrix \( \tilde{X} \)
Least squares for classification (cont.)

The discriminant function is

\[ y(x) = \tilde{W}^T \tilde{x} = T^T (\tilde{X}^\dagger)^T \tilde{x} \quad (16) \]

Using 1-of-\( K \) coding for \( K \) classes, the elements of the predictions \( y(x) \) will sum to one for any value of \( x \), though cannot be interpreted as probabilities

- the elements of \( y(x) \) are not constrained to be in \((0, 1)\)

It gives an exact closed-form solution for the discriminant function parameters
Least squares for classification (cont.)

- More worrying is that least-squares solutions lack of robustness to outliers
- Outliers lead to large changes in the location of the decision boundary

A synthetic set from two classes in a two-dimensional space \((x_1, x_2)\)
- The magenta line is the decision boundary from least squares
A synthetic set from three classes in a two-dimensional input space \((x_1, x_2)\)

- Linear decision boundaries could separate classes well

Least squares corresponds to maximum likelihood under the assumption of a Gaussian conditional distribution, but binary target vectors are not Gaussian.
Fisher’s linear discriminant

Discriminant functions
**Fisher’s linear discriminant**

We view linear classification from the viewpoint of dimensionality reduction.

We project the $D$-dimensional input vector $x$ down onto 1D

$$y = w^T x$$  \hspace{1cm} (17)

Consider the two-classes case

For classification, we place a threshold on $y$

- $y \geq -w_0$, $\rightarrow C_1$
- otherwise, $\rightarrow C_2$

Projection onto 1D leads to a considerable loss of information, in general

- Classes that are well separated in the original space may become strongly overlapping in one dimension

Nevertheless, we can always adjust the components of the weight vector $w$
Fisher’s linear discriminant (cont.)

The basic idea: Set $w$ so that the projection maximises class separation

The mean vectors of the two classes

Consider a two-class problem

- $N_1$ points of class $C_1$
- $N_2$ points of class $C_2$

The mean vectors of the two classes

$$m_1 = \frac{1}{N_1} \sum_{n \in C_1} x_n$$

$$m_2 = \frac{1}{N_2} \sum_{n \in C_2} x_n \quad \text{(18)}$$
Fisher’s linear discriminant (cont.)

We need a measure of the separation of the classes, after projection onto \( \mathbf{w} \)

An intuitive measure is separation of projected class means

\[
m_2 - m_1 = \mathbf{w}^T (m_2 - m_1) \tag{19}
\]

\[
m_k = \mathbf{w}^T \mathbf{m}_k \tag{20}
\]

\( m_k \), the mean of projected \( C_k \) data
Discriminant functions

Fisher’s linear discriminant (cont.)

\[ m_2 - m_1 = w^T(m_2 - m_1) \]

Pseudocode

This expression can be made arbitrarily large by increasing the magnitude of \( w \)

1. Constrain \( w \) to unit-length, \( \sum_i w_i^2 = 1 \)
2. Use Lagrange multipliers for the constrained maximisation
3. Find the solution, \( w \propto (m_2 - m_1) (\star^1) \)

The optimal projection is along the line joining the original class means

\[ L = w^T(m_2 - m_1) + \lambda(w^T w - 1), \text{ then } \nabla L = m_2 - m_1 + 2\lambda w = 0 \text{ to get } w = 1/(2\lambda)(m_2 - m_1) \]
Fisher’s linear discriminant (cont.)

Projection onto the line joining the class means

- Good separation in the original 2D space
- Considerable class overlap in the projection 1D space
Fisher’s idea is to maximise a function that gives
- Large separation between projected class means
- Small variance within each projected class
Or, find a direction that minimises class overlap

The projection $y = w^T x$ transforms labelled points in $x$ into a labelled set in $y$

The **within-class variance** of the projected data
$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2$$ (21)

The **total within-class variance** for the whole data (two-classes)
$$s_1^2 + s_2^2$$ (22)

The **between-class variance**
$$(m_2 - m_1)^2$$ (23)
Fisher’s linear discriminant (cont.)

**Definition**

**Fisher’s criterion:** The ratio of between-class and within-class variance

\[
J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}
\]

(24)

To make the dependence on \( w \) explicit\(^2 \), we can write the Fisher’s criterion as

\[
J(w) = \frac{w^T S_B w}{w^T S_W w}
\]

(25)

- \( S_B \) is the **between-class covariance matrix**

\[
S_B = (m_2 - m_1)(m_2 - m_1)^T
\]

(26)

- \( S_W \) is the **total within-class covariance matrix**

\[
S_W = \sum_{n \in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2} (x_n - m_2)(x_n - m_2)^T
\]

(27)

\[^2 J(w) = \frac{w^T(m_1 - m_2)(m_1 - m_2)^Tw}{w^T(S_{W_1} + S_{W_2})w} \]
Fisher’s linear discriminant (cont.)

After differentiating with respect to $\mathbf{w}^3$, we get that $J(\mathbf{w})$ is maximised when

$$
(w^T S_B w) S_W w = (w^T S_W w) S_B w
$$

(28)

The between-class covariance matrix shows that $S_B \mathbf{w}$ is always in the direction of $(m_2 - m_1)^4$ and we can drop the scalar factors $(\mathbf{w}^T S_B \mathbf{w})$ and $(\mathbf{w}^T S_W \mathbf{w})^5$

Multiplying both sides of $(\mathbf{w}^T S_B \mathbf{w}) S_W \mathbf{w} = (\mathbf{w}^T S_W \mathbf{w}) S_B \mathbf{w}$ by $S_W^{-1}$, we obtain

$$
\mathbf{w} \propto S_W^{-1} (m_2 - m_1)
$$

(29)

---

$$
3 \frac{\partial}{\partial \mathbf{w}} \left( \frac{\mathbf{w}^T S_B \mathbf{w}}{w^T S_W \mathbf{w}} \right) = \frac{2}{(w^T S_W w)^2} \left[ (\mathbf{w}^T S_W w) S_B w - (\mathbf{w}^T S_B w) S_W w \right]
$$

$^4(m_1 - m_2)w$ is a scalar.

$^5$We are not interested in the magnitude of $\mathbf{w}$, only its direction.
Fisher’s linear discriminant (cont.)

This is the Fisher’s linear discriminant, although it is not a discriminant.

To construct a discriminant and classify point $x$, we must define a threshold $y_0$:

- If $y(x) \geq y_0 \rightarrow C_1$
- If $y(x) < y_0 \rightarrow C_2$

Projection based on the Fisher linear discriminant.
Relation to least squares

Discriminant functions
The least-squares approach to determining a linear discriminant is motivated by making model predictions as close as possible to a set of target values.

Fisher criterion pursues maximum class separation in the output space.

Is there a relation between these two approaches?

- For the two-class problem, Fisher’s criterion is a special case of least squares.
- Fisher’s solution can be equivalent to the least square solution for the weight.
- We need to adopt a slightly different coding scheme for the target variables.
Consider a total number of patterns $N$

Let $N_1$ be the number of patterns in class $C_1$
- We take the target for class $C_1$ to be $N/N_1$

Let $N_2$ be the number of patterns in class $C_2$
- We take the target for class $C_2$ to be $-N/N_2$

The target value for class $C_1$ approximates the reciprocal of the prior probability for the class
Relation to least squares (cont.)

We write the sum-of-squares error function

$$E(w, w_0) = \frac{1}{2} \sum_{n=1}^{N} \left[ (w^T x_n + w_0) - t_n \right]^2$$

(30)

We set derivatives wrt $w_0$ and $w$ to zero

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

(31)

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

(32)
Relation to least squares (cont.)

From $\sum_{n=1}^{N} [(w^T x_n + w_0) - t_n] = 0$ and using the target scheme encoding

- The bias is given by $w_0 = -w^T m$ \hspace{0.5cm} (33)

- where we have used

$$\sum_{n=1}^{N} t_n = N_1 \frac{N}{N_1} - N_2 \frac{N}{N_2} = 0$$ \hspace{0.5cm} (34)

$$m = \frac{1}{N} \sum_{n=1}^{N} x_n = \frac{1}{N}(N_1 m_1 + N_2 m_2)$$ \hspace{0.5cm} (35)

$m$ is the mean of the total data set
Relation to least squares (cont.)

Using the target encoding, from \(\sum_{n=1}^{N} [(w^T x_n + w_0) - t_n] x_n = 0\) we get

\[
\left( S_W + \frac{N_1 N_2}{N} S_B \right) w = N(m_1 - m_2) \tag{36}
\]

- with \(S_W = \sum_{n \in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2} (x_n - m_2)(x_n - m_2)^T\)
- with \(S_B = (m_2 - m_1)(m_2 - m_1)^T\)
- with \(w_0 = -w^T m\)

\(S_B = (m_2 - m_1)(m_2 - m_1)^T\) shows that \(S_B w\) is in the direction of \(m_2 - m_1\)

\[
w \propto S_W^{-1}(m_2 - m_1) \tag{37}
\]
Relation to least squares (cont.)

The weight vector $\mathbf{w}$ coincides with what found from the Fisher’s criterion

- Vector $\mathbf{x}$ with $y(\mathbf{x}) = \mathbf{w}^T(\mathbf{x} - \mathbf{m}) > 0$ is classified as belonging to class $C_1$
- Vector $\mathbf{x}$ with $y(\mathbf{x}) = \mathbf{w}^T(\mathbf{x} - \mathbf{m}) \leq 0$ is classified as belonging to class $C_2$
Fisher’s discriminant for multiple classes

Discriminant functions

Two classes
Multiple classes
Least squares for classification
Fisher’s linear discriminant
Relation to least squares
Fisher’s discriminant for multiple classes
The perceptron
Fisher’s discriminant for multiple classes

We consider a generalisation of the Fisher discriminant to $K > 2$ classes

- Assumption: Input dimensionality $D$ is greater than class number $K$

We firstly introduce $D' > 1$ linear features $y_k = w_k^T x$ with $k = 1, \ldots, D'$

$$y = W^T x \quad (38)$$

- with $y$ grouping $\{y_k\}$
- with $W$ grouping $\{w_k\}$

We are not including any bias parameter term in the definition of $y$
Fisher’s discriminant for multiple classes (cont.)

Generalise the **within-class covariance matrix** to $K$ classes, $N_k$ cases per class

$$S_W = \sum_{k=1}^{K} S_k$$  \hspace{1cm} (39)

- $$S_k = \sum_{n \in C_k} (x_n - m_k)(x_n - m_k)^T$$  \hspace{1cm} (40)

- $$m_k = \frac{1}{N_k} \sum_{n \in C_k} x_n$$  \hspace{1cm} (41)
Fisher’s discriminant for multiple classes (cont.)

Define the generalisation of the between-class covariance matrix to $K$ classes

Consider first the total covariance matrix

$$S_T = \sum_{n=1}^{N} (x_n - m)(x_n - m)^T \quad (42)$$

$N = \sum_k N_k$ is the total number of points

$$m = \frac{1}{N} \sum_{n=1}^{N} x_n \quad (43)$$

$m$ above is the mean of the total data set
Fisher’s discriminant for multiple classes (cont.)

Total covariance matrix can be decomposed into the sum of within-class covariance matrix $S_W$ plus an additional matrix $S_B$

$$S_T = S_W + S_B$$ (44)

We identify $S_B$ as a measure of between-class covariance

$$S_B = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T$$ (45)

Covariance matrices $S_W$ and $S_B$ are defined in the original $x$-space
Fisher’s discriminant for multiple classes

We define similar matrices in the projected $D'$-dimensional $y$-space

$$S_W = \sum_{k=1}^{K} \sum_{n \in C_k} (y_n - \mu_k)(y_n - \mu_k)^T \quad (46)$$

$$S_B = \sum_{k=1}^{K} N_k (\mu_k - \mu)(\mu_k - \mu)^T \quad (47)$$

Where the mean vectors $\mu_k$ and $\mu$ have been defined as always

$$\mu_k = \frac{1}{N_k} \sum_{n \in C_k} y_n \quad \mu = \frac{1}{N} \sum_{k=1}^{N} N_k \mu_k \quad (48)$$
Fisher’s discriminant for multiple classes (cont.)

Construct a scalar that is large when the between-class covariance is large and also when the within-class covariance is small, there are many possible choices

$$ J(W) = \text{Tr}(S_W^{-1}S_B) $$

(49)

This criterion can be written as an explicit function of the projection matrix \( W \)

$$ J(W) = \text{Tr}\left( (WS_WW^T)^{-1}(WS_BW^T) \right) $$

(50)

The maximisation is given in the literature and involved, it leads to weights given by the eigenvectors of \( S_W^{-1}S_B \) associated to its \( D' \) largest eigenvalues
Discriminant functions

Two classes
Multiple classes
Least squares for classification
Fisher's linear discriminant
Relation to least squares
Fisher's discriminant for multiple classes

The perceptron
Discriminant functions
The perceptron

Another example of a linear discriminant model is Rosenblatt’s perceptron

- It occupies an important place in the history of pattern recognition

It corresponds to a two-class model in which the input vector $x$ is transformed first by using a fixed nonlinear transformation, to give a feature vector $\phi(x)$

The feature vector is used to construct a generalised linear model of the form

$$y(x) = f(w^T \phi(x)) \quad (51)$$

The nonlinear activation function $f(\cdot)$ is given by a step function

$$f(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0 \end{cases} \quad (52)$$

The feature vector $\phi(x)$ includes a bias component $\phi_0(x) = 1$

Convenient to use target values $t = +1$ for class $C_1$ and $t = -1$ for class $C_2$

- To match the behaviour of the activation function
The perceptron (cont.)

The determination of $\mathbf{w}$ can be motivated by error function minimisation

- A natural choice of error function is the total number of misclassified patterns

Not a simple algo because the error is a piecewise constant function of $\mathbf{w}$

- Discontinuities wherever a change in $\mathbf{w}$ causes the decision boundary to move across one of the points
- Methods based on changing $\mathbf{w}$ using the gradient of the error function cannot then be applied, because the gradient is zero almost everywhere

We consider an alternative error function, known as the perceptron criterion
The perceptron (cont.)

\[ y(w^T \phi(x_n)) = \begin{cases} 
+1, & w^T \phi(x_n) \geq 0 \\
-1, & w^T \phi(x_n) < 0 
\end{cases} \]

We are seeking a weight vector \( w \) such that

- patterns \( x_n \) in class \( C_1 \) \( (t = +1) \) will have \( w^T \phi(x_n) > 0 \)
- patterns \( x_n \) in class \( C_2 \) \( (t = -1) \) will have \( w^T \phi(x_n) < 0 \)

We want all patterns satisfy \( w^T \phi(x_n)t_n > 0 \)

The perceptron criterion associates zero error with correctly classified patterns, whereas for a misclassified pattern \( x_n \) it tries to minimise quantity \( -w^T \phi(x_n)t_n \)

\[ E_P(w) = -\sum_{n \in M} w^T \phi_n t_n \] (53)

\( \phi_n = \phi(x_n) \) and \( M \) denotes the set of misclassified patterns
The perceptron (cont.)

\[ E_P(w) = - \sum_{n \in M} w^T \phi_n t_n \]

Misclassified patterns contribute to the error with a linear function of \( w \).

We can apply a stochastic gradient algorithm to this error function:

\[ w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E_P(w) = w^{(\tau)} + \eta \phi_n t_n \]  \( (54) \)

It changes the weight vector using a learning rate \( \eta \) at each step \( \tau \).
The perceptron (cont.)

Pseudocode

1. We cycle through the training patterns
2. We evaluate the perceptron function
3. If the pattern is correctly classified, the weights remain unchanged
4. If the pattern is wrongly classified, then
   - For class $C_1$, we add vector $\phi(x)$ to current $w$
   - For class $C_2$, we subtract vector $\phi(x)$ from current $w$
Discriminant functions

Two classes
Multiple classes
Least squares for classification
Fisher's linear discriminant
Relation to least squares
Fisher’s discriminant for multiple classes
The perceptron

The perceptron (cont.)

-1 -0.5 0 0.5 1
-1 -0.5 0 0.5 1
-1 -0.5 0 0.5 1
-1 -0.5 0 0.5 1

Discriminant functions

UFC/DC
ATAI-I (CK0146)
2017.1
Remark

Issues with convergence, as a substantial number of iterations is required and more worryingly guaranteed only for linearly separable classes.

Issue with generalisation to more than two classes problems.