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 $\mathcal{C}_k$, $k = 1, \dots, K$

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 $\mathbf{x}$
- $(D-1)$-dimensional hyperplanes, in the D-dimensional input space

Classes that can be separated well by linear surfaces are **linearly separable**

Linear models for classification (cont.)

For regression problems, the target variable $\mathbf{t}$ was a vector of real numbers

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

Two-class problems: Binary representation

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

- $t = 1$ represents class $\mathcal{C}_1$
- $t = 0$ represents class $\mathcal{C}_2$

It is the probability of class $\mathcal{C}_1$, with the probability only taking values of 0 and 1

Multi-class problems: 1-of-K coding scheme

There is a K-long target vector $\mathbf{t}$, such that

- If the class is $\mathcal{C}_j$, all elements t_k of $\mathbf{t}$ are zero for $k \neq j$ and one for $k = j$
- t_k is the probability that the class is $\mathcal{C}_k$

$K = 6$ and $\mathcal{C}_k = 4$, then $\mathbf{t} = (0, 0, 0, 1, 0, 0)^T$

Linear models for classification (cont.)

The simplest approach to classification problems is through construction of a **discriminant function** that directly assigns each vector $\mathbf{x}$ to a specific class

More powerful is to **model the conditional probability distribution** $p(\mathcal{C}_k|\mathbf{x})$ in an inference stage, and use this distribution to make optimal decisions

- **Discriminative modelling**: $p(\mathcal{C}_k|\mathbf{x})$ can be modelled directly, using a parametric model and optimising the parameters using a training set
- **Generative modelling**: We model class-conditional densities $p(\mathbf{x}|\mathcal{C}_k)$ and the prior probabilities $p(\mathcal{C}_k)$ for the classes, and we compute the posterior probabilities using Bayes' theorem

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} \tag{1}$$

Discriminant functions

Linear models for classification

Discriminant functions

We start with the construction of classifiers based on discriminant functions

In linear regression models

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

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

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

$$y(\mathbf{x}) = f(\mathbf{w}^T \mathbf{x} + w_0) \tag{2}$$

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

Discriminant functions (cont.)

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

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

Decision surfaces correspond to $y(\mathbf{x}) = \text{constant}$ so $\mathbf{w}^T \mathbf{x} + w_0 = \text{constant}$

- Decision surfaces are linear functions of $\mathbf{x}$, 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

Two classes

Discriminant functions

Two classes

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

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 \tag{3}$$

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

An input vector $\mathbf{x}$ is assigned to class $\mathcal{C}_1$ if $y(\mathbf{x}) \geq 0$ and to class $\mathcal{C}_2$ otherwise

The corresponding decision boundary is defined by the relationship $y(\mathbf{x}) = 0$

- $(D-1)$-dimensional hyperplane within the D-dimensional input space

Two classes (cont.)

Consider two points $\mathbf{x}_A$ and $\mathbf{x}_B$ on the decision boundary

$$\begin{cases} y(\mathbf{x}_A) = \mathbf{w}^T\mathbf{x}_A = 0 \\ y(\mathbf{x}_B) = \mathbf{w}^T\mathbf{x}_B = 0 \end{cases} \longrightarrow \quad \mathbf{w}^T(\mathbf{x}_A - \mathbf{x}_B) = 0 \quad \longrightarrow \quad \mathbf{w} \perp (\mathbf{x}_A - \mathbf{x}_B)$$

Vector $\mathbf{w}$ is orthogonal to every vector in the boundary

- $\mathbf{w}$ sets the orientation of the boundary

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

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

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 $y(\mathbf{x})$ gives a signed measure of perpendicular distance too

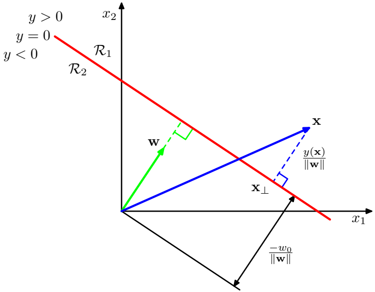
- The distance from the point $\mathbf{x}$ to the decision surface

Let $\mathbf{x}$ be any point and $\mathbf{x}_\perp$ its orthogonal projection onto the boundary

$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{||\mathbf{w}||} \qquad (5)$$

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

$$r = \frac{y(\mathbf{x})}{||\mathbf{w}||} \qquad (6)$$



Two classes (cont.)

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

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

Remark

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

Multiple classes

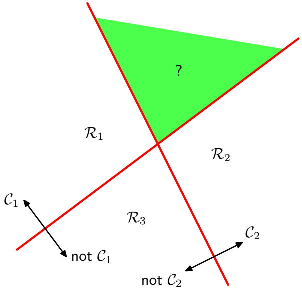
Discriminant functions

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 $\mathcal{C}_k$ from points not in $\mathcal{C}_k$
- It is a **one-versus-the-rest** classifier



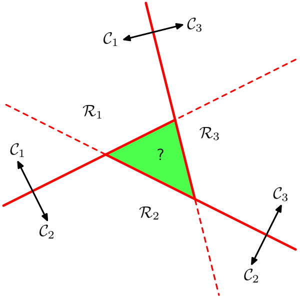
This approach leads to regions of the input space that are ambiguously classified

- By definition, the green area cannot be classified as both $\mathcal{C}_1$ and $\mathcal{C}_2$

Multiple classes (cont.)

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

- Separate points in class $\mathcal{C}_k$ from points in $\mathcal{C}_{j \neq k}$, with $j = 1, \dots, K$
- 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

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0} \tag{8}$$

A point $\mathbf{x}$ is then assigned to class $\mathcal{C}_k$, if $y_k(\mathbf{x}) > y_j(\mathbf{x})$, for all $j \neq k$

- The boundary between class $\mathcal{C}_k$ and class $\mathcal{C}_j$ is $y_k(\mathbf{x}) = y_j(\mathbf{x})$ or

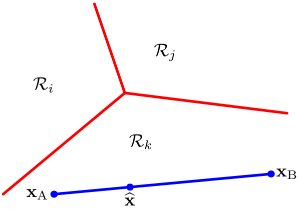
$$(\mathbf{w}_k - \mathbf{w}_j)^T \mathbf{x} + (w_{k0} - w_{j0}) = 0 \tag{9}$$

- 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 $\mathbf{x}_A$ and $\mathbf{x}_B$ in region $\mathcal{R}_k$

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

$$\hat{\mathbf{x}} = \lambda \mathbf{x}_A + (1-\lambda)\mathbf{x}_B, \ \lambda \in [0,1] \qquad (10)$$

Because of the linearity of the discriminant function

$$y_k(\hat{\mathbf{x}}) = \lambda y_k(\mathbf{x}_A) + (1-\lambda) y_k(\mathbf{x}_B) \qquad (11)$$

Because $\mathbf{x}_A$ and $\mathbf{x}_B$ are in $\mathcal{R}_k$, we have $y_k(\mathbf{x}_A) > y_j(\mathbf{x}_A)$ and $y_k(\mathbf{x}_B) > y_j(\mathbf{x}_B)$, for all $j \neq k$, and hence $y_k(\hat{\mathbf{x}}) > y_j(\hat{\mathbf{x}})$ so $\tilde{\mathbf{x}}$ also lies within the region $\mathcal{R}_k$

Least squares for classification

Discriminant functions

Least squares for classification

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 $\mathbf{t}$

Least squares for classification (cpnt.)

Remark

One justification is 'least squares approximates the conditional expectation $\mathbb{E}[\mathbf{t}|\mathbf{x}]$ 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 $\mathcal{C}_k$ is described by its own linear model in the form

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0}, \quad k = 1, \dots, K \tag{12}$$

The K models can be grouped using vector notation to obtain

$$\mathbf{y}(\mathbf{x}) = \tilde{\mathbf{W}}^T \tilde{\mathbf{x}} \tag{13}$$

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

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

By minimising the sum-of-squares error function, get the parameter matrix $\tilde{\mathbf{W}}$

Least squares for classification (cont.)

Consider a training data set $\{\mathbf{x}_n, \mathbf{t}_n\}_{n=1}^{N}$ and define matrix $\mathbf{T}$ and matrix $\tilde{\mathbf{X}}$

- The n-th column of $\mathbf{T}$ is vector $\mathbf{t}_n^T$
- The n-th row of $\tilde{\mathbf{X}}$ is vector $\tilde{\mathbf{x}}_n^T$

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

$$E_D(\tilde{\mathbf{X}}) = \frac{1}{2}\mathrm{Tr}\Big((\tilde{\mathbf{X}}\tilde{\mathbf{W}} - \mathbf{T})^T(\tilde{\mathbf{X}}\tilde{\mathbf{W}} - \mathbf{T})\Big) \quad (14)$$

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

$$\tilde{\mathbf{W}} = (\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^T\mathbf{T} = \tilde{\mathbf{X}}^{\dagger}\mathbf{T} \quad (15)$$

where $\tilde{\mathbf{X}}^{\dagger}$ is the Moore-Penrose pseudo-inverse of the matrix $\tilde{\mathbf{X}}$

Least squares for classification (cont.)

The discriminant function is

$$\mathbf{y}(\mathbf{x}) = \tilde{\mathbf{W}}^T \tilde{\mathbf{x}} = \mathbf{T}^T (\tilde{\mathbf{X}}^{\dagger})^T \tilde{\mathbf{x}} \tag{16}$$

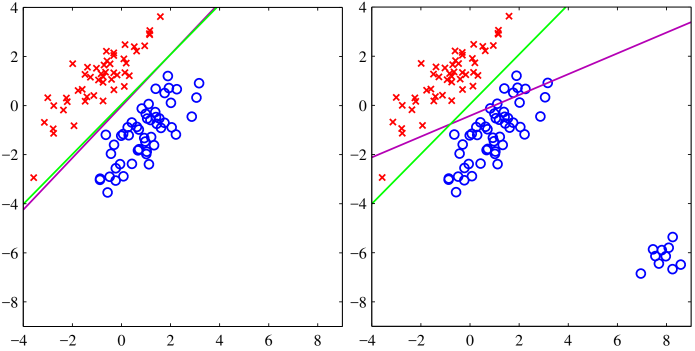
Using 1-of-K coding for K classes, the elements of the predictions $y(\mathbf{x})$ will sum to one for any value of $\mathbf{x}$, though cannot be interpreted as probabilities

- the elements of $y(\mathbf{x})$ 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



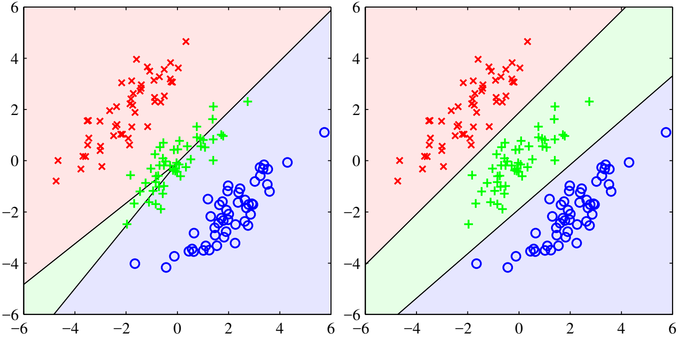
A synthetic set from two classes in a two-dimensional space (x_1, x_2)

- The magenta line is the decision boundary from least squares

Least squares for classification (cont.)

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

Consider the two-classes case

We project the D-dimensional input vector $\mathbf{x}$ down onto $1D$

$$y = \mathbf{w}^T \mathbf{x} \quad (17)$$

For classification, we place a threshold on y

- $y \geq -w_0, \quad \longrightarrow \mathcal{C}_1$
- otherwise, $\quad \longrightarrow \mathcal{C}_2$

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 $\mathbf{w}$

Fisher's linear discriminant (cont.)

The basic idea: Set $\mathbf{w}$ so that the projection maximises class separation

Consider a two-class problem

- N_1 points of class $\mathcal{C}_1$
- N_2 points of class $\mathcal{C}_2$

The mean vectors of the two classes

$$\begin{aligned} \mathbf{m}_1 &= \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n \\ \mathbf{m}_2 &= \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n \end{aligned} \tag{18}$$

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

$$m_2 - m_1 = \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1) \quad (19)$$

$$m_k = \mathbf{w}^T \mathbf{m}_k \quad (20)$$

m_k, the mean of projected $\mathcal{C}_k$ data

Fisher's linear discriminant (cont.)

$$m_2 - m_1 = \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1)$$

Pseudocode

This expression can be made arbitrarily large by increasing the magnitude of $\mathbf{w}$

1. Constrain $\mathbf{w}$ to unit-length, $\sum_i w_i^2 = 1$
2. Use Lagrange multipliers for the constrained maximisation
3. Find the solution, $\mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$ ($\star$[1])

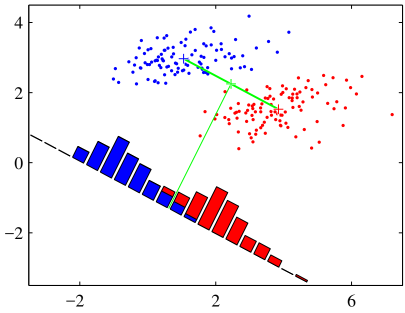
The optimal projection is along the line joining the original class means

[1] $L = \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1) + \lambda(\mathbf{w}^T\mathbf{w} - 1)$, then $\nabla L = \mathbf{m}_2 - \mathbf{m}_1 + 2\lambda\mathbf{w} = 0$ to get $\mathbf{w} = 1/(2\lambda)(\mathbf{m}_2 - \mathbf{m}_1)$

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 linear discriminant (cont.)

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 $y = \mathbf{w}^T \mathbf{x}$ transforms labelled points in $\mathbf{x}$ into a labelled set in y

The **within-class variance** of the projected data

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2 \quad (21)$$

The **total within-class variance** for the whole data (two-classes)

$$s_1^2 + s_2^2 \quad (22)$$

The **between-class variance**

$$(m_2 - m_1)^2 \quad (23)$$

Fisher's linear discriminant (cont.)

Definition

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

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \tag{24}$$

To make the dependence on $\mathbf{w}$ explicit[2], we can write the Fisher's criterion as

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \tag{25}$$

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

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T \tag{26}$$

- $\mathbf{S}_W$ is the **total within-class covariance matrix**

$$\mathbf{S}_W = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T \tag{27}$$

[2] $J(\mathbf{w}) = \dfrac{\mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{w}}{\mathbf{w}^T (\mathbf{S}_{W_1} + \mathbf{S}_{W_2}) \mathbf{w}}$

Fisher's linear discriminant (cont.)

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

$$(\mathbf{w}^T \mathbf{S}_B \mathbf{w}) \mathbf{S}_W \mathbf{w} = (\mathbf{w}^T \mathbf{S}_W \mathbf{w}) \mathbf{S}_B \mathbf{w} \tag{28}$$

The between-class covariance matrix shows that $\mathbf{S}_B \mathbf{w}$ is always in the direction of $(\mathbf{m}_2 - \mathbf{m}_1)$[4] and we can drop the scalar factors $(\mathbf{w}^T \mathbf{S}_B \mathbf{w})$ and $(\mathbf{w}^T \mathbf{S}_W \mathbf{w})$[5]

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

$$\mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1) \tag{29}$$

[3] $\dfrac{\partial}{\partial \mathbf{w}} \Big(\dfrac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \Big) = \dfrac{2}{(\mathbf{w}^T \mathbf{S}_W \mathbf{w})^2} \Big[(\mathbf{w}^T \mathbf{S}_W \mathbf{w}) \mathbf{S}_B \mathbf{w} - (\mathbf{w}^T \mathbf{S}_B \mathbf{w}) \mathbf{S}_W \mathbf{w} \Big]$

[4] $(\mathbf{m}_1 - \mathbf{m}_2)\mathbf{w}$ is a scalar.

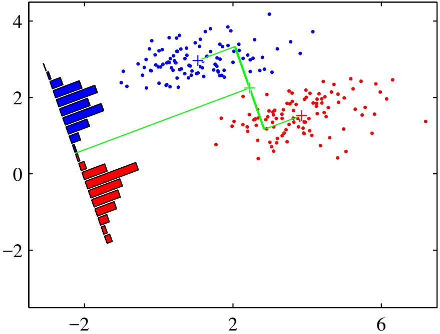
[5] We are not interested in the magnitude of $\mathbf{w}$, only its direction.

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 $\mathbf{x}$, we must define a threshold y_0

- If $y(\mathbf{x}) \geq y_0 \quad \rightarrow \mathcal{C}_1$
- If $y(\mathbf{x}) < y_0 \quad \rightarrow \mathcal{C}_2$



Projection based on the Fisher linear discriminant

Relation to least squares

Discriminant functions

Relation to least squares

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

Relation to least squares (cont.)

Consider a total number of patterns N

Let N_1 be the number of patterns in class $\mathcal{C}_1$

- We take the target for class $\mathcal{C}_1$ to be N/N_1

Let N_2 be the number of patterns in class $\mathcal{C}_2$

- We take the target for class $\mathcal{C}_2$ to be $-N/N_2$

The target value for class $\mathcal{C}_1$ approximates the reciprocal of the prior probability for the class

Relation to least squares (cont.)

We write the sum-of-squares error function

$$E(\mathbf{w}, w_0) = \frac{1}{2}\sum_{n=1}^{N}\Big[\underbrace{(\mathbf{w}^T\mathbf{x}_n + w_0)}_{y_n} - t_n\Big]^2 \tag{30}$$

We set derivatives wrt w_0 and $\mathbf{w}$ to zero

$$\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n + w_0 - t_n) = 0 \tag{31}$$

$$\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n + w_0 - t_n)\mathbf{x}_n = 0 \tag{32}$$

Relation to least squares (cont.)

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

- The bias is given by

$$w_0 = -\mathbf{w}^T\mathbf{m} \tag{33}$$

- where we have used

$$\sum_{n=1}^{N} t_n = N_1\frac{N}{N_1} - N_2\frac{N}{N_2} = 0 \tag{34}$$

$$\mathbf{m} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n = \frac{1}{N}(N_1\mathbf{m}_1 + N_2\mathbf{m}_2) \tag{35}$$

m is the **mean of the total data set**

Relation to least squares (cont.)

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

$$\Big(\mathbf{S}_W + \frac{N_1 N_2}{N}\mathbf{S}_B\Big)\mathbf{w} = N(\mathbf{m}_1 - \mathbf{m}_2) \tag{36}$$

- with $\mathbf{S}_W = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$
- with $\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$
- with $w_0 = -\mathbf{w}^T \mathbf{m}$

$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$ shows that $\mathbf{S}_B\mathbf{w}$ is in the direction of $\mathbf{m}_2 - \mathbf{m}_1$

$$\mathbf{w} \propto \mathbf{S_W^{-1}}(\mathbf{m_2} - \mathbf{m_1}) \tag{37}$$

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 $\mathcal{C}_1$
- Vector $\mathbf{x}$ with $y(\mathbf{x}) = \mathbf{w}^T(\mathbf{x} - \mathbf{m}) \leq 0$ is classified as belonging to class $\mathcal{C}_2$

Fisher's discriminant for multiple classes

Discriminant functions

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** $y_k = \mathbf{w}_k^T \mathbf{x}$ with $k = 1, \dots, D'$

$$\mathbf{y} = \mathbf{W}^T \mathbf{x} \tag{38}$$

- with $\mathbf{y}$ grouping $\{y_k\}$
- with $\mathbf{W}$ grouping $\{\mathbf{w}_k\}$

We are not including any bias parameter term in the definition of $\mathbf{y}$

Fisher's discriminant for multiple classes (cont.)

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

$$\mathbf{S}_W = \sum_{k=1}^{K} \mathbf{S}_k \tag{39}$$

- $$\mathbf{S}_k = \sum_{n \in \mathcal{C}_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T \tag{40}$$
- $$\mathbf{m}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{x}_n \tag{41}$$

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**

$$\mathbf{S}_T = \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^T \tag{42}$$

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

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n \tag{43}$$

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 $\mathbf{S}_W$ plus an additional matrix $\mathbf{S}_B$

$$\mathbf{S}_T = \mathbf{S}_W + \mathbf{S}_B \tag{44}$$

We identify $\mathbf{S}_B$ as a measure of between-class covariance

$$\mathbf{S}_B = \sum_{k=1}^{K} N_k(\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T \tag{45}$$

Covariance matrices $\mathbf{S}_W$ and $\mathbf{S}_B$ are defined in the original $\mathbf{x}$-space

Fisher's discriminant for multiple classes

We define similar matrices in the projected D'-dimensional $\mathbf{y}$-space

$$\mathbf{S}_W = \sum_{k=1}^{K} \sum_{n \in \mathbf{C}_k} (\mathbf{y}_n - \boldsymbol{\mu}_k)(\mathbf{y}_n - \boldsymbol{\mu}_k)^T \tag{46}$$

$$\mathbf{S}_B = \sum_{k=1}^{K} N_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T \tag{47}$$

Where the mean vectors $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}$ have been defined as always

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{y}_n \qquad \boldsymbol{\mu} = \frac{1}{N} \sum_{k=1}^{N} N_k \boldsymbol{\mu}_k \tag{48}$$

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

$$J(\mathbf{W}) = \text{Tr}(\mathbf{S}_{\mathbf{W}}^{-1}\mathbf{S}_{\mathbf{B}}) \tag{49}$$

This criterion can be written as an explicit function of the projection matrix $\mathbf{W}$

$$J(\mathbf{W}) = \text{Tr}\Big((\mathbf{W}\mathbf{S}_W\mathbf{W}^T)^{-1}(\mathbf{W}\mathbf{S}_B\mathbf{W}^T)\Big) \tag{50}$$

The maximisation is given in the literature and involved, it leads to weights given by the eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$ associated to its D' largest eigenvalues

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 $\mathbf{x}$ is transformed first by using a fixed nonlinear transformation, to give a feature vector $\phi(\mathbf{x})$

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

$$y(\mathbf{x}) = f(\mathbf{w}^T \phi(\mathbf{x})) \tag{51}$$

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

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

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

Convenient to use target values $t = +1$ for class $\mathcal{C}_1$ and $t = -1$ for class $\mathcal{C}_2$

- 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
  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.)

$$y(\mathbf{w}^T\phi(\mathbf{x}_n)) = \begin{cases} +1, & \mathbf{w}^T\phi(\mathbf{x}_n) \geq 0 \\ -1, & \mathbf{w}^T\phi(\mathbf{x}_n) < 0 \end{cases}$$

We are seeking a weight vector $\mathbf{w}$ such that

- patterns $\mathbf{x}_n$ in class $\mathcal{C}_1$ $(t = +1)$ will have $\mathbf{w}^T\phi(\mathbf{x}_n) > 0$
- patterns $\mathbf{x}_n$ in class $\mathcal{C}_2$ $(t = -1)$ will have $\mathbf{w}^T\phi(\mathbf{x}_n) < 0$

We want all patterns satisfy $\mathbf{w}^T\phi(\mathbf{x}_n)t_n > 0$

The perceptron criterion associates zero error with correctly classified patterns, whereas for a misclassified patter $\mathbf{x}_n$ it tries to minimise quantity $-\mathbf{w}^T\phi(\mathbf{x}_n)t_n$

$$E_P(\mathbf{w}) = -\sum_{n \in \mathcal{M}} \mathbf{w}^T\phi_n t_n \tag{53}$$

$\phi_n = \phi(\mathbf{x}_n)$ and $\mathcal{M}$ denotes the set of misclassified patters

The perceptron (cont.)

$$E_P(\mathbf{w}) = -\sum_{n \in \mathcal{M}} \mathbf{w}^T \phi_n t_n$$

Misclassified patterns contribute to the error with a linear function of **w**

We can apply a stochastic gradient algorithm to this error function

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_P(\mathbf{w}) = \mathbf{w}^{(\tau)} + \eta \phi_n t_n \tag{54}$$

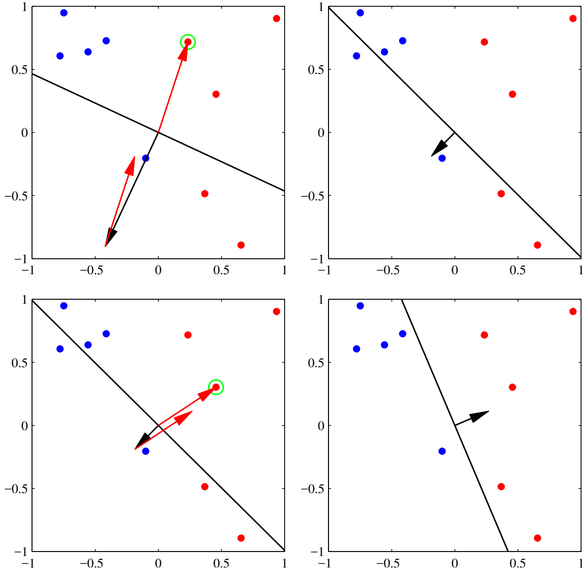
It changes the weight vector using a learning rate η at each step τ

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 $\mathcal{C}_1$, we add vector $\phi(\mathbf{x})$ to current $\mathbf{w}$
   - For class $\mathcal{C}_2$, we subtract vector $\phi(\mathbf{x})$ from current $\mathbf{w}$

The perceptron (cont.)



The perceptron (cont.)

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