Linear Models for Classification

Three Approaches to Classification Problems

1. Deterministic model
2. Probabilistic generative model
3. Probabilistic deterministic model

Linear Models for Classification

Goal: Take an input vector $\boldsymbol{x}$ and assign it to one of K classes $C_1, \ldots, C_K$.

The input space is divided into regions whose boundaries are called **decision boundaries** or **decision surfaces**.

Linear classification models yield linear decision boundaries, which are $(D-1)$-dimensional hyperplanes within the D-dimensional input space.

4.1 Discriminant Functions

A **discriminant** is a function that takes an input vector $\boldsymbol{x}$ and assigns it to one of K classes $C_1, \dots, C_K$. A **linear discriminant** is a function whose decision surfaces are hyperplanes.

The simplest representation of a linear discriminant function is

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

where $\boldsymbol{w}$ is a weight vector and w_0 is a bias.

Binary Classification ($K = 2$)

If $y(\boldsymbol{x}) \geq 0$, assign C_1. Otherwise, assign C_2.

The decision surface is defined by $y(\boldsymbol{x}) = 0$, which is a $(D-1)$-dimensional hyperplane in the D-dimensional input space.

If $\boldsymbol{v}$ is any vector that lies within the decision surface, then $\boldsymbol{w}^T\boldsymbol{v} = 0$, so $\boldsymbol{w}$ is orthogonal to every vector in the decision surface. $\boldsymbol{w}$ determines the orientation of the decision surface.

The bias term w_0 determines the location of the decision boundary.

Classification Involving Multiple Classes ($K > 2$)

Use a single K-class discriminant comprising K linear functions of the form

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}, \text{ for } k = 1, \dots, K.$$

Assign $\boldsymbol{x}$ to class C_k if $y_k(\boldsymbol{x}) > y_j(\boldsymbol{x})$ for all $j \neq k$. The decision boundary between class C_k and class C_j is given by $y_k(\boldsymbol{x}) = y_j(\boldsymbol{x})$, which is a $(D-1)$-dimensional hyperplane defined by

$$(\boldsymbol{w}_k - \boldsymbol{w}_j)^T \boldsymbol{x} + (\boldsymbol{w}_{k0} - \boldsymbol{w}_{j0}) = 0.$$

Classification Involving Multiple Classes ($K > 2$)

The decision regions of such a discriminant are **convex.** To see this, if $\boldsymbol{x}_A, \boldsymbol{x}_B$ are two points lying in the same decision region $\mathcal{R}_k$, the line $\hat{\boldsymbol{x}}$ connecting $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ can be expressed as

$$\hat{\boldsymbol{x}} = \lambda \boldsymbol{x}_A + (1 - \lambda)\boldsymbol{x}_B, \text{ where } 0 \leq \lambda \leq 1.$$

By linearity of the discriminant function,

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

Since $y_k(\boldsymbol{x}_A) > y_j(\boldsymbol{x}_A)$ and $y_k(\boldsymbol{x}_B) > y_j(\boldsymbol{x}_B)$ for all $j \neq k$, $y_k(\hat{\boldsymbol{x}}) > y_j(\hat{\boldsymbol{x}})$, so $\hat{\boldsymbol{x}}$ lies inside $\mathcal{R}_k$.

Learning the Parameters of Linear Discriminant Functions

Three approaches:

1. Least squares

2. Fisher's linear discriminant

3. Perceptron algorithm

Least Squares

For ease of notation, let

$$\tilde{\boldsymbol{w}}_k = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix}, \ \tilde{\boldsymbol{x}} = \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix}, \ \tilde{\boldsymbol{W}} = \begin{bmatrix} \tilde{\boldsymbol{w}}_1 & \tilde{\boldsymbol{w}}_2 & \cdots & \tilde{\boldsymbol{w}}_K \end{bmatrix}.$$

Then we can combine the K equations

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}, \ \text{for } k = 1, \dots, K$$

into

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

Least Squares

Note: $D =$ input space dimension, $N =$ number of samples in the training data

We determine the parameter matrix $\tilde{\boldsymbol{W}}$ by minimizing a sum-of-squares error function.

Let

$$\boldsymbol{T} = \begin{bmatrix} \tilde{\boldsymbol{t}}_1^T \\ \tilde{\boldsymbol{t}}_2^T \\ \vdots \\ \tilde{\boldsymbol{t}}_N^T \end{bmatrix}$$

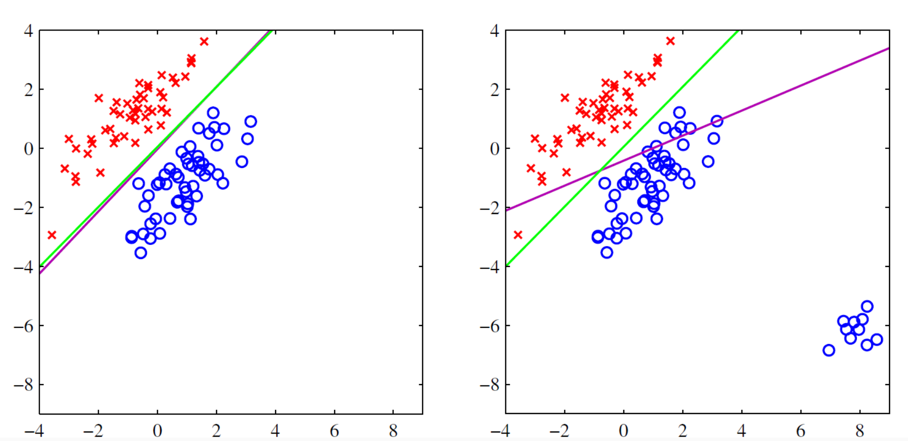
where the training data set is $\{\boldsymbol{x}_n, \boldsymbol{t}_n\}$ for $n = 1, \ldots, N$. Then the sum-of-sqaure error function is

$$E(\tilde{\boldsymbol{W}}) = \frac{1}{2}\mathrm{Tr}\{(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{W}} - \boldsymbol{T})^T(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{W}} - \boldsymbol{T})\}.$$

The solution for $\tilde{\boldsymbol{W}}$ is

$$\tilde{\boldsymbol{W}} = (\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}^T\boldsymbol{T} = \tilde{\boldsymbol{X}}^\dagger\boldsymbol{T}.$$

Drawbacks of Least Squares



Least squares is highly sensitive to outliers.

Fisher's Linear Discriminant

We can view linear classification as dimensionality reduction.

First look at binary classification ($K = 2$).

Consider taking a D-dimensional input vector $\boldsymbol{x}$ and projecting it down to one dimension using

$$y = \boldsymbol{w}^T \boldsymbol{x}.$$

Projection onto one dimension leads to a loss of information, and classes that are well separated in the original D-dimensional space may become overlapping in one dimension.

By adjusting the components of the weight vector $\boldsymbol{w}$, we can select a projection that maximizes the class separation.

Fisher's Linear Discriminant

Suppose there are N_1 points of class C_1 and N_2 points of class C_2. The mean vectors of the two classes are

$$\boldsymbol{m}_1 = \frac{1}{N_1} \sum_{n \in C_1} \boldsymbol{x}_n, \qquad \boldsymbol{m}_2 = \frac{1}{N_1} \sum_{n \in C_2} \boldsymbol{x}_n.$$

The simplest measure of the separation of the classes, when projected onto $\boldsymbol{w}$, is the separation of the projected class means. So we can choose $\boldsymbol{w}$ to maximize

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

where

$$m_k = \boldsymbol{w}^T \boldsymbol{m}_k$$

is the mean of the projected data from class C_k.

Fisher's Linear Discriminant

Constrain $\boldsymbol{w}$ to have unit length, so that $\sum_i w_i^2 = 1$. Perform constrained maximization using Lagrange multiplier. We get $\boldsymbol{w} \propto (\boldsymbol{m}_2 - \boldsymbol{m}_1)$. This can still have problems, such as having considerable class overlap in the projected space.

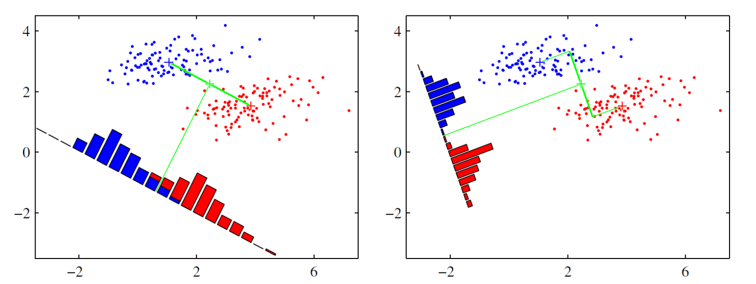


Figure 1: Left: projection with class overlap; Right: projection based on the Fischer linear discriminant

Fisher's Linear Discriminant

Maximize a function that will give a large separation between the projected class means while also giving a small variance within each class.

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

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2$$

where $y_n = \boldsymbol{w}^T \boldsymbol{x}$.

Define the total within-class variance for the whole data set to be $s_1^2 + s_2^2$.

The Fisher criterion is defined as the ratio of the between-class variance to the within-class variance:

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

Fisher's Linear Discriminant

We can rewrite $J(w)$ as

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}} \tag{4.26}$$

where $\mathbf{S}_{\mathrm{B}}$ is the *between-class* covariance matrix and is given by

$$\mathbf{S}_{\mathrm{B}} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}} \tag{4.27}$$

and $\mathbf{S}_{\mathrm{W}}$ is the total *within-class* covariance matrix, given by

$$\mathbf{S}_{\mathrm{W}} = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^{\mathrm{T}} + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^{\mathrm{T}}. \tag{4.28}$$

Differentiating (4.26) with respect to $\mathbf{w}$, we find that $J(\mathbf{w})$ is maximized when

$$(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w})\mathbf{S}_{\mathrm{W}}\mathbf{w} = (\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w})\mathbf{S}_{\mathrm{B}}\mathbf{w}. \tag{4.29}$$

From (4.27), we see that $\mathbf{S}_{\mathrm{B}}\mathbf{w}$ is always in the direction of $(\mathbf{m}_2 - \mathbf{m}_1)$. Furthermore, we do not care about the magnitude of $\mathbf{w}$, only its direction, and so we can drop the scalar factors $(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w})$ and $(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w})$. Multiplying both sides of (4.29) by $\mathbf{S}_{\mathrm{W}}^{-1}$ we then obtain

$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1). \tag{4.30}$$

Fisher's Linear Discriminant

Fisher's linear discriminant gives a specific choice of direction for projection of the data down to one dimension.

Projected data can subsequently be used to construct a discriminant, by choosing a threshold y_0 so that we classify a new point as belonging to C_1 if $y(x) \geq y_0$ and classify it as belonging to C_2 otherwise.

Perceptron Algorithm

For binary classification where the input vector $\boldsymbol{x}$ is transformed using a nonlinear transformation to give a feature vector $\phi(\boldsymbol{x})$. We construct a **generalized linear model** is of the form

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

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

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

Perceptron Algorithm

Find a weight vector $\boldsymbol{w}$ such that for $\boldsymbol{x}_n$ in class C_1 will have $\boldsymbol{w}^T\phi(\boldsymbol{x}_n) > 0$, whereas $\boldsymbol{x}_n$ in class C_2 have $\boldsymbol{w}^T\phi(\boldsymbol{x}_n) < 0$.

The perceptron criterion associates zero error with any pattern that is correctly classified, whereas for a misclassified $\boldsymbol{x}_n$ it tries to minimize the quantity $-\boldsymbol{w}^T\phi(\boldsymbol{x}_n)t_n$.

The perceptron criterion is given by

$$E(\boldsymbol{w}) = -\sum_{n \in M} \boldsymbol{w}^T\phi_n t_n$$

where M is the set of misclassified data.

Bayesian Probability Recap

Bayes' Theorem:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

Bayes' theorem can be used to convert a **prior probability** into a **posterior probability** by incorporating the evidence from the observed data.

Bayesian Probability Recap

Bayes' Theorem:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

Bayes' theorem can be used to convert a **prior probability** into a **posterior probability** by incorporating the evidence from the observed data.

Bayesian view of probability: probabilities provide a quantification of uncertainty.

We can use probability theory to describe the uncertainty in model parameters such as $\boldsymbol{w}$, or in the choice of model itself.

Bayesian Probability Recap

We capture our assumptions about the model parameter $\boldsymbol{w}$, before observing the data, in the form of a **prior probability distribution** $p(\boldsymbol{w})$.

The effect of the observed data $D = \{t_1, \dots, t_N\}$ is expressed through the conditional probability $p(D|\boldsymbol{w})$.

Bayes's theorem gives us

$$p(\boldsymbol{w}|D) = \frac{p(D|\boldsymbol{w})p(\boldsymbol{w})}{p(D)} \qquad \text{(posterior} \propto \text{ likelihood} \times \text{prior)}$$

which allows us to evaluate the uncertainty in $\boldsymbol{w}$ after we have observed D in the form of the **posterior probability** $p(\boldsymbol{w}|D)$.

$p(D|\boldsymbol{w})$ on the right hand side can be viewed as a function of $\boldsymbol{w}$, which is called the **likelihood function.** It expresses how probable the observed data set is for different settings of the parameter vector $\boldsymbol{w}$.