Module 1.3: Classification

Linear (Fisher's) Discriminant Analysis

Suppose we have vectors $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N$ and we know that N_1 of these vectors belong to class $\mathcal{D}_1$, N_2 of them belong to class $\mathcal{D}_2$, and so forth for c classes. Assume $\vec{x} \in \mathbb{R}^n$

We want to find a vector in $\mathbb{R}^n$ (and in fact set of vectors) such that when we project all $\vec{x} \in \mathcal{D}_i$, they will be distant from the projection of other $\vec{x} \in \mathcal{D}_{j \neq i}$ onto the same vector.

A good way to do this as by having the mean values of the projections of all $\vec{x} \in \mathcal{D}_i$ distant from the projected mean value of the entire data set.

Consider a candidate vector ω, the projection of $\vec{x} \in \mathcal{D}_i$ onto ω is

$$\underset{\text{row vector}}{\rightarrow} y_i = \vec{\omega}^T X_i \quad \text{where } X_i \text{ is a matrix whose columns are } \vec{x} \in \mathcal{D}_i$$

The mean of the projections of $\vec{x} \in \mathcal{D}_i$ onto $\vec{\omega}$ is

$$\tilde{\mu}_i = \frac{1}{N_i} y_i \vec{1}_i = \vec{\omega}^T \left(\frac{1}{N_i} X_i \vec{1}_i \right) \quad \text{where } 1_i = \underbrace{[1, 1, \ldots, 1]}_{N_i}^T$$

$$= \omega^T \vec{\mu}_i$$

where $\vec{\mu}_i$ is the mean of the vectors of $\vec{x} \in \mathcal{D}_i$

$$\vec{\mu}_i = \frac{1}{N_i} \sum_{\vec{x} \in \mathcal{D}_i} x$$

Now, the mean of all the samples is $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$, and its projection to ω is

$$\tilde{\mu} = \vec{\omega}^T \vec{\mu}$$

We want

$$\sum_{i=1}^{c} N_i (\tilde{\mu}_i - \tilde{\mu})^2 = \sum_{i=1}^{c} N_i \vec{\omega}^T (\vec{\mu}_i - \vec{\mu})(\vec{\mu}_i - \vec{\mu})^T \vec{\omega}$$

$$= \vec{\omega}^T \underbrace{\left(\sum_{i=1}^{c} N_i (\vec{\mu}_i - \vec{\mu})(\vec{\mu}_i - \mu)^T \right)}_{S_B} \vec{\omega}$$

S_B ← Between class scatter matrix

to be as large as possible.

Meanwhile we want to variance of the projections, within a class,

$$\sum_{y \in Y_i} (y - \tilde{\mu}_i)^2 = \sum_{x \in D_i} (\vec{\omega}^T \vec{x} - \vec{\omega}^T \vec{\mu}_i)^2$$

$$= \sum_{x \in D_i} \vec{\omega}^T (\vec{x} - \vec{\mu}_i)(\vec{x} - \vec{\mu}_i)^T \omega$$

$$= \vec{\omega}^T \underbrace{\left(\sum_{x \in D_i} (\vec{x} - \vec{\mu}_i)(x - \vec{\mu}_i)^T \right)}_{S_i} \vec{\omega}$$

S_i ← Scatter matrix for class i

to be small as possible.

Let $S_W = \sum_{i=1}^{c} S_i$ be the total within-class scatter matrix.

We want

$$J(\omega) = \frac{\omega^T S_B \omega}{\omega^T \left(\sum_{i=1}^{c} S_i \right) \omega} = \frac{\omega^T S_B \omega}{\omega^T S_W \omega}$$

to be maximized (with $\|\omega\| = 1$). We need ω^* that accomplishes this

$$\omega^* = \arg\max_{\omega} \frac{\omega^T S_B \omega}{\omega^T S_W \omega}$$

or $\omega^* = \arg\max_{\omega} \omega^T S_B \omega$ subject to $\omega^T S_W \omega = K$

The Lagrangian is given as

$$\mathcal{L} = \omega^T S_B \omega - \lambda\left(\omega^T S_W \omega - K\right) = \omega^T (S_B - \lambda S_W)\omega + \lambda K$$

$$\nabla_\omega \mathcal{L} = 2(S_B - \lambda S_W)\omega = 0 \Rightarrow S_B \omega^* = \lambda S_W \omega^*$$

which is a generalized eigenvalue problem.
If S_W is full-rank (or invertible), we can write

$$S_W^{-1} S_B \omega^* = \lambda \omega^*$$

From our definition, $S_B = \sum_{i=1}^{c} N_i (\vec{\mu}_i - \vec{\mu})(\mu_i - \mu)^T \in \mathbb{R}^{n \times n}$ has, at most, rank $c-1$. Meanwhile, $S_W = \sum_{i=1}^{c} \sum_{x_i \in D_i} (x - \mu_i)(x - \mu_i)^T \in \mathbb{R}^{n \times n}$ has, at most rank $c(N_i - 1) = N - c$.

If the data size $N > c + n$, and $c > n$, S_B and S_W are generally likely to be full rank

For projection onto more vectors, m, in general, the maximization problem is

$$W^* = \underset{W}{\operatorname{argmax}} \frac{|W^T S_B W|}{|W^T S_W W|} = \underset{W}{\operatorname{argmax}} \frac{|\tilde{S}_B|}{|S_W|}$$

Which yields an extension of the previous solution

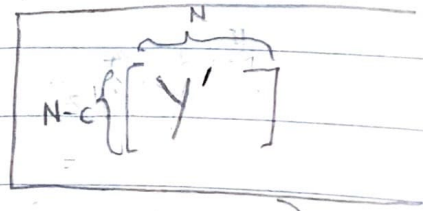
$$S_B \omega_k^* = \lambda S_W \omega_k^* \qquad k = 1, \ldots m$$

But the issue of S_W being singular still persists. We can address this the following ways:

(1) S_W is very likely to have a rank of $N-c$, we can reduce the dimension of our data from n to $N-c$. And what better way can we preserve the information in our data while reducing its dimension than by using PCA? So we pick out the first $N-c$ (or less) components of our PCA transformation

$$y_i' = y_i(1:N-c) = \left(T(:, 1:N-c)\right)^T x_i \qquad i = 1, \ldots, N$$

We now carry out finding S_B' & S_W' using y_i' & find $W' \in \mathbb{R}^{N-c \times m}$



(2) We can find the the Moore-Penrose Inverse (Pseudoinverse)

$$S_W^{\dagger} S_B \omega_k^* = \lambda \omega_k^*$$

NOTE: In (1)

We have taken two steps which can be expressed as

$$y_{FLD} = \underbrace{W'^{T}_{FLD} \; T^{T}_{I:N-C}}_{W_{opt}^{T}} \; x$$

$m \times N-C \times N-C \times n$

$= m \times n$

Note that because S_B is rank $C-1$, m should be, at most, $C-1$

The columns of W_{opt} are called Fisherfaces.

---

(3) Regularize S_W:

$S_w + \varepsilon I_d$ where ε is a small constant

You can show that for a 2-class case:

$$w = \frac{1}{\lambda} S_w^{-1} S_B w = S^{-1}(\mu_1 - \mu_2)$$