Supervised Dimension Reduction

Prof. Dr. H. H. Takada

Quantitative Research – Itaú Asset Management
Institute of Mathematics and Statistics – University of São Paulo

Supervised Linear Projections

- In cases where class information is available, and our ultimate interest is to reduce dimensionality for improved classification, it makes sense to use the available class information in forming the projection.
- We consider data from two different classes. For class 1, we have a set of N_1 datapoints

  $$\mathcal{X}_1 = \left\{\mathbf{x}_1^1, \ldots, \mathbf{x}_1^{N_1}\right\}$$

  and similarly for class 2, we have a set of N_2 datapoints

  $$\mathcal{X}_2 = \left\{\mathbf{x}_2^1, \ldots, \mathbf{x}_2^{N_2}\right\}$$

  Our interest is then to find a linear projection,

  $$\mathbf{y} = \mathbf{W}^{\mathsf{T}}\mathbf{x}$$

  where $\dim \mathbf{W} = D \times L$, $L < D$, such that for two datapoints $\mathbf{x}^i$ and $\mathbf{x}^j$ in the same class, the distance between their projections $\mathbf{y}^i$ and $\mathbf{y}^j$ should be small.

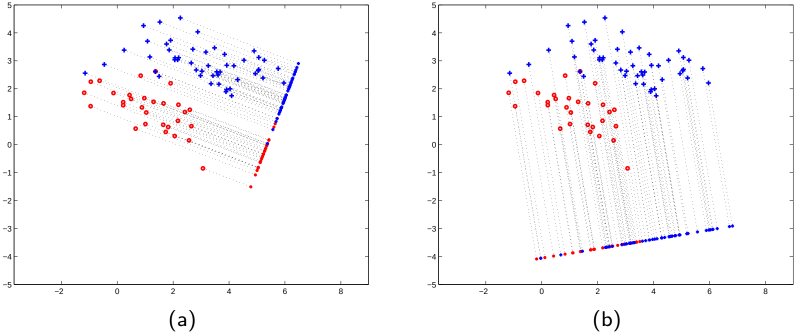


Figure: The large crosses represent data from class 1, and the large circles from class 2. Their projections onto 1 dimension are represented by their small counterparts. **(a)**: Fisher's Linear Discriminant Analysis. Here there is little class overlap in the projections. **(b)**: Unsupervised dimension reduction using Principal Components Analysis for comparison. There is considerable class overlap in the projection. In both (a) and (b) the one dimensional projection is the distance along the line, measured from an arbitrary chosen fixed point on the line.

Fisher's Linear Discriminant

We model the data from each class with a Gaussian. That is

$$p(\mathbf{x}_1) = \mathcal{N}\left(\mathbf{x}_1 | \mathbf{m}_1, \mathbf{S}_1\right), \qquad p(\mathbf{x}_2) = \mathcal{N}\left(\mathbf{x}_2 | \mathbf{m}_2, \mathbf{S}_2\right)$$

where $\mathbf{m}_1$ is the sample mean of class 1 data, and $\mathbf{S}_1$ the sample covariance; similarly for class 2. The projections of the points from the two classes are then given by

$$y_1^n = \mathbf{w}^\mathsf{T}\mathbf{x}_1^n, \qquad y_2^n = \mathbf{w}^\mathsf{T}\mathbf{x}_2^n$$

Because the projections are linear, the projected distributions are also Gaussian,

$$\begin{aligned} p(y_1) &= \mathcal{N}\left(y_1 | \mu_1, \sigma_1^2\right), & \mu_1 &= \mathbf{w}^\mathsf{T}\mathbf{m}_1, & \sigma_1^2 &= \mathbf{w}^\mathsf{T}\mathbf{S}_1\mathbf{w} \\ p(y_2) &= \mathcal{N}\left(y_2 | \mu_2, \sigma_2^2\right), & \mu_2 &= \mathbf{w}^\mathsf{T}\mathbf{m}_2, & \sigma_2^2 &= \mathbf{w}^\mathsf{T}\mathbf{S}_2\mathbf{w} \end{aligned}$$

We search for a projection $\mathbf{w}$ such that the projected distributions have minimal overlap. A useful objective function therefore is

$$\frac{(\mu_1 - \mu_2)^2}{\pi_1\sigma_1^2 + \pi_2\sigma_2^2}$$

where π_i represents the fraction of the dataset in class i.

In terms of the projection $\mathbf{w}$, the objective is

$$F(\mathbf{w}) = \frac{\mathbf{w}^\mathsf{T}(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\mathsf{T}\mathbf{w}}{\mathbf{w}^\mathsf{T}(\pi_1\mathbf{S}_1 + \pi_2\mathbf{S}_2)\mathbf{w}} = \frac{\mathbf{w}^\mathsf{T}\mathbf{A}\mathbf{w}}{\mathbf{w}^\mathsf{T}\mathbf{B}\mathbf{w}}$$

where

$$\mathbf{A} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\mathsf{T}, \qquad \mathbf{B} = \pi_1\mathbf{S}_1 + \pi_2\mathbf{S}_2$$

The optimal $\mathbf{w}$ can be found by differentiating:

$$\frac{\partial}{\partial\mathbf{w}}\frac{\mathbf{w}^\mathsf{T}\mathbf{A}\mathbf{w}}{\mathbf{w}^\mathsf{T}\mathbf{B}\mathbf{w}} = \frac{2}{(\mathbf{w}^\mathsf{T}\mathbf{B}\mathbf{w})^2}\left[\left(\mathbf{w}^\mathsf{T}\mathbf{B}\mathbf{w}\right)\mathbf{A}\mathbf{w} - \left(\mathbf{w}^\mathsf{T}\mathbf{A}\mathbf{w}\right)\mathbf{B}\mathbf{w}\right]$$

and therefore the zero derivative requirement is

$$\left(\mathbf{w}^\mathsf{T}\mathbf{B}\mathbf{w}\right)\mathbf{A}\mathbf{w} = \left(\mathbf{w}^\mathsf{T}\mathbf{A}\mathbf{w}\right)\mathbf{B}\mathbf{w}$$

Multiplying by the inverse of $\mathbf{B}$ we have

$$\mathbf{B}^{-1}(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\mathsf{T}\mathbf{w} = \frac{\mathbf{w}^\mathsf{T}\mathbf{A}\mathbf{w}}{\mathbf{w}^\mathsf{T}\mathbf{B}\mathbf{w}}\mathbf{w}$$

Since $(\mathbf{m}_1 - \mathbf{m}_2)^\mathsf{T}\mathbf{w}$ is a scalar, the optimal projection is explicitly given by

$$\mathbf{w} \propto \mathbf{B}^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

When the naive method breaks down

- The above derivation relied on the existence of the inverse of $\mathbf{B}$.
- A case where $\mathbf{B}$ is not invertible is when there are fewer datapoints $N_1 + N_2$ than dimensions D.
- A related problematic case is when there are elements of the input vectors that never vary. For example, in the hand-written digits case, the pixels at the corner edges are actually always zero. Let's call such a pixel z. The matrix $\mathbf{B}$ will then have a zero entry for $[B]_{z,z}$ (indeed the whole z^{th} row and column of $\mathbf{B}$ will be zero) so that for any vector of the form

$$\mathbf{w}^{\mathsf{T}} = (0, 0, \ldots, w_z, 0, 0, \ldots, 0) \Rightarrow \mathbf{w}^{\mathsf{T}}\mathbf{B}\mathbf{w} = 0$$

  This shows that the denominator of Fisher's objective can become zero, and the objective ill defined.

Canonical Variates

Canonical Variates generalizes Fisher's method to projections of more than one dimension and more than two classes. The projection of any point is given by

$$\mathbf{y} = \mathbf{W}^{\mathsf{T}}\mathbf{x}$$

where $\mathbf{W}$ is a $D \times L$ matrix. Assuming that the data $\mathbf{x}$ from class c is Gaussian distributed,

$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\mathbf{m}_c, \mathbf{S}_c\right)$$

the projections $\mathbf{y}$ are also Gaussian

$$p(\mathbf{y}) = \mathcal{N}\left(\mathbf{y}\middle|\mathbf{W}^{\mathsf{T}}\mathbf{m}_c, \mathbf{W}^{\mathsf{T}}\mathbf{S}_c\mathbf{W}\right)$$

Find the mean $\mathbf{m}$ of the whole dataset and $\mathbf{m}_c$, the mean of the each class c. Form

$$\mathbf{A} \equiv \sum_{c=1}^{C} N_c \left(\mathbf{m}_c - \mathbf{m}\right)\left(\mathbf{m}_c - \mathbf{m}\right)^{\mathsf{T}}$$

where N_c is the number of datapoints in class c.
Compute the covariance matrix $\mathbf{S}_c$ of the data for each class c. Define

$$\mathbf{B} \equiv \sum_{c=1}^{C} N_c \mathbf{S}_c$$

Assuming $\mathbf{B}$ is invertible we can define the Cholesky factor $\tilde{\mathbf{B}}$, with

$$\tilde{\mathbf{B}}^{\mathsf{T}}\tilde{\mathbf{B}} = \mathbf{B}$$

A natural objective is then to maximize

$$F(\mathbf{W}) \equiv \frac{\mathrm{trace}\left(\mathbf{W}^{\mathsf{T}}\tilde{\mathbf{B}}^{-\mathsf{T}}\mathbf{A}\tilde{\mathbf{B}}^{-1}\mathbf{W}\right)}{\mathrm{trace}\left(\mathbf{W}^{\mathsf{T}}\mathbf{W}\right)}$$

If we assume an orthonormality constraint on $\mathbf{W}$, then we equivalently require the maximization of

$$F(\mathbf{W}) \equiv \mathrm{trace}\left(\mathbf{W}^{\mathsf{T}}\mathbf{C}\mathbf{W}\right), \text{ subject to } \mathbf{W}^{\mathsf{T}}\mathbf{W} = \mathbf{I}$$

where

$$\mathbf{C} \equiv \frac{1}{D}\tilde{\mathbf{B}}^{-\mathsf{T}}\mathbf{A}\tilde{\mathbf{B}}^{-1}$$

Since $\mathbf{C}$ is symmetric and positive semidefinite, it has a real eigen-decomposition

$$\mathbf{C} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^\mathsf{T}$$

where $\mathbf{\Lambda} = \mathrm{diag}\left(\lambda_1, \lambda_2, \ldots, \lambda_D\right)$ is diagonal with non-negative entries containing the eigenvalues, sorted by decreasing order, $\lambda_1 \geq \lambda_2 \geq \ldots$ and $\mathbf{E}^\mathsf{T}\mathbf{E} = \mathbf{I}$. Hence

$$F(\mathbf{W}) = \mathrm{trace}\left(\mathbf{W}^\mathsf{T}\mathbf{E}\mathbf{\Lambda}\mathbf{E}^\mathsf{T}\mathbf{W}\right)$$

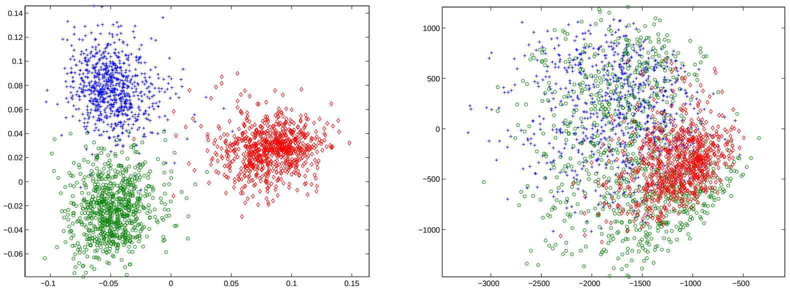
- By setting $\mathbf{W} = [\mathbf{e}_1, \ldots, \mathbf{e}_L]$, where $\mathbf{e}_l$ is the l^{th} eigenvector, the objective $F(\mathbf{W})$ becomes the sum of the first L eigenvalues.
- This setting maximizes the objective function since forming $\mathbf{W}$ from any other columns of $\mathbf{E}$ would give a lower sum.
- Note that since $\mathbf{A}$ has rank C, there can be no more than $C - 1$ non-zero eigenvalues and corresponding directions.

Canonical Variates Algorithm

Algorithm 1 Canonical Variates

1: Compute the between and within class scatter matrices $\mathbf{A}$, and $\mathbf{B}$.
2: Compute the Cholesky factor $\tilde{\mathbf{B}}$ of $\mathbf{B}$.
3: Compute the L principal eigenvectors $[\mathbf{e}_1, \ldots, \mathbf{e}_L]$ of $\tilde{\mathbf{B}}^{-\mathsf{T}}\mathbf{A}\tilde{\mathbf{B}}^{-1}$.
4: Return $\mathbf{W} = [\mathbf{e}_1, \ldots, \mathbf{e}_L]$ as the projection matrix.

Using canonical variates on the digit data



(a) (b)

Figure: **(a)**: Canonical Variates projection of examples of handwritten digits 3 ('+'), 5 ('o') and 7 (diamond). There are 800 examples from each digit class. Plotted are the projections down to 2 dimensions. **(b)**: PCA projections for comparison.

Matlab

```
>> setup;
>> demoCanonVarDigits;
```