Dimensionality Reduction

Jayanta Mukhopadhyay
Dept. of Computer Science and Engg.

Books

- Chapters 6 of “Introduction to Machine Learning” by Ethem Alpaydin.

Why to reduce dimension?

- For reducing complexity of inference, memory and computation.
  - In most learning algorithms, the complexity depends on
    - the number of input dimensions, d
    - the size of the data sample, N,
- Saving cost of extraction of features.
- Simpler models more robust in small datasets.
- Explanation with fewer features convenient for knowledge extraction.
- Convenient to plot, visualize, etc.

Two major approaches

- Feature selection
  - To find k of the d dimensions that give us the most information discarding the other $(d - k)$ dimensions.
  - Subset selection method
- Feature extraction
  - A new set of k dimensions that are combination of original d dimensions.
  - Supervised and unsupervised techniques
    - PCA, LDA.
      - Projection of feature vectors to a lower dimensional space.

Subset Selection

- F: A feature set of input dimensions x_i, $i=1,2,..d$.
- $E(F)$: Error in validation set if F is used as input.
- Supervised method (requires training and testing).
  - Any method would do.
- Two methods (Greedy methods)
  - Sequential forward selection.
  - Sequential backward selection.

Sequential forward selection

- F=NULL.
- Select x_i which provides least $E(F \text{ U } x_i)$.
- Add x_i if $E(F \text{ U } x_i) < E(F)$.
- Repeat above two steps till no more addition possible.

- Local search method.
- Does not guarantee optimal feature combination.
- The cost of training and testing is O(d^2).

Sequential backward selection

- F= Set of all the features.
- Select x_i which provides least $E(F - x_i)$.
- Remove x_i if $E(F - x_i) < E(\mathrm{F})$.
- Repeat above two steps till no more removal possible.

- Local search method.
- Does not guarantee optimal feature combination.
- The cost of training and testing is O(d^2),
  - training with more features more costly.

Principal component analysis (PCA)

- To find a mapping from the inputs in the original d-dimensional space to a new ($k < d$)-dimensional space, with minimum loss of information.
- $\boldsymbol{x}$: Input feature vector of dimension d
- w: A direction (unit vector) of dimension d.
- Projection of $\boldsymbol{x}$ along w: $w^T\boldsymbol{x}$
  - *Make data centered around origin of the space.*
- Principal component: component along the direction w_1 such that its variance is maximum among all possible projections.

Principal components

- Principal component: component along the direction w_1 such that its variance is maximum among all possible projections.
  - 1st principal component.
- 2nd component:
  - component along a direction w_2 orthogonal to w_1 having the maximum variance.
- Similarly other principal components are defined.
  - For a d-dimensional space there are maximum d principal components.

Computation of 1st component

- $z_1 = w_1^T \boldsymbol{x}$
  - Corresponding random variable be denoted as Z_1
  - X is the random variable whose instance is x with mean $\boldsymbol{m}$ and covariance matrix Σ.
  - Mean of Z_1: $w_1^T \boldsymbol{m}$
  - Variance of Z_1 : $w_1^T \Sigma w_1$
- Optimization problem:
  - To maximize variance keeping w_1 as a unit vector.
  - $w_1 = \mathrm{argmax}_w \{w^T \Sigma w - l.\ (w^T w - 1)\}$
  - l is the Lagrange coefficient.

Computation of 1st component

- $z_1 = w_1^T \boldsymbol{x}$
- $w_1 = \text{argmax}_w \{ w^T \Sigma w - l.\ (w^T w - 1) \}$
  - l is the Lagrange coefficient.
- Taking the derivative of the argument w.r.t. w and setting it to 0:
  - 2 $\Sigma w_1 - 2 l w_1 = 0 \rightarrow \Sigma w_1 = l w_1$
  - $\rightarrow w_1^T \Sigma w_1 = w_1^T l w_1 = l w_1^T w_1 = l$ (variance)
  - w_1 is the eigen vector of Σ corresponding to the maximum eigen value.

Computation of 2nd component

- $z_2 = w_2^T \boldsymbol{x}$
- Optimization problem
  - $w_2 = \mathrm{argmax}_w \{ w^T \Sigma\, w - l_1 (w^T w - 1) - l_2 (w_1^T w - 0) \}$
  - l_1 and l_2 are Lagrange coefficients.
  - w_2 orthogonal to w_1.
- Taking the derivative of the argument w.r.t. w and setting it to 0:
  - $2\, \Sigma\, w_2 - 2 l_1 w_2 - l_2 w_1 = 0$

Computation of 2nd component

- 2 $\Sigma w_2\text{-}2l_1w_2\text{-}l_2w_1=0$
- Pre-multiplying with w_1 we get
- 2 $w_1^T\Sigma w_2\text{-}2l_1w_1^Tw2\text{-}l_2w_1^Tw_1=0$
  - →2 $w_1^T\Sigma w2\text{-}l_2=0$
  - As $w_1^T\Sigma w_2$ is scalar, $w_2^T\Sigma w_1$ is also scalar.
  - Replacing Σw_1 by $l\, w_1$
    - $w_2^T\Sigma w_1 =0$. Hence $l_2=0$
- 2 Σw_2 - $2l_1w_2=0$ → $\Sigma w_2 = l_1w_2$
  - → w_2 is eigen vector and l_1 is the variance.
- w_2 : eigen vector of Σ to the 2nd maximum eigen value.

and so on …

$z=W^T(x-m)$

PCA-Algorithm

- Input: A set of data points: S={$\boldsymbol{x}_j=(x_{1j},x_{2j},\ldots x_{dj})$| $\boldsymbol{x}_j$ in R^d}.
- Output: A set of k eigen vectors providing tx. matrix: $W=[w_1,w_2,\ldots,w_k]$

1. Compute mean of data points.
2. Translate all data points to their mean.
3. Compute covariance matrix of the set.
4. Compute eigen vetcors and eigen values (in increasing order).
5. Choose k such that the fraction of variance accounted for is more than a threshold.
6. Use those k-components for representing any data point.

Example

- Data : {(5, 3, 2), (4, 6, 0), (3, -7, 14), (2, 5, 3), (3, 13, -6)}
- Perform PCA and if applicable, reduce the dimension of data.

Example (contd.)

$$\mathrm{X} = \begin{bmatrix} 5 & 4 & 3 & 2 & 3 \\ 3 & 6 & -7 & 5 & 13 \\ 2 & 0 & 14 & 3 & -6 \end{bmatrix} \qquad \bar{\mathrm{S}} = \begin{bmatrix} 3.4 & 3.4 & 3.4 \\ 4 & 4 & 4 \\ 2.6 & 2.6 & 2.6 \end{bmatrix}$$

$$\tilde{\mathrm{X}} = \mathrm{X} - \bar{\mathrm{S}} = \begin{bmatrix} 1.6 & .6 & -.4 & -1.4 & -.4 \\ -1 & 2 & -11 & 1 & 9 \\ -.6 & -2.6 & 11.4 & .4 & -8.6 \end{bmatrix}$$

$$\mathrm{C} = \frac{1}{5}\tilde{\mathrm{X}}\tilde{\mathrm{X}}^{\mathrm{T}} \qquad \mathrm{C} = \begin{bmatrix} 1.04 & -.2 & -.84 \\ -.2 & 41.6 & -41.4 \\ -.84 & -41.4 & 42.24 \end{bmatrix}$$

Example (contd.)

$$C = \begin{bmatrix} \mathbf{1.04} & -.2 & -.84 \\ -.2 & \mathbf{41.6} & -41.4 \\ -.84 & -41.4 & \mathbf{42.24} \end{bmatrix}$$

Total variance: Trace(C)=1.04+41.6+42.24=84.88

Eigen values of C: (83.3238, 1.5562, 0)

Sum of eigen values

Respective eigen vectors:

$$\mathbf{e_1} = \begin{bmatrix} -.0055 \\ -.7043 \\ .7099 \end{bmatrix} \quad \mathbf{e_2} = \begin{bmatrix} -.8165 \\ .413 \\ .4034 \end{bmatrix} \quad \mathbf{e_3} = \begin{bmatrix} -.5774 \\ -.5774 \\ -.5774 \end{bmatrix}$$

Example (contd.)

Respective eigen vectors:

$$\mathbf{e_1} = \begin{bmatrix} -.0055 \\ -.7043 \\ .7099 \end{bmatrix} \quad \mathbf{e_2} = \begin{bmatrix} -.8165 \\ .413 \\ .4034 \end{bmatrix} \quad \mathbf{e_3} = \begin{bmatrix} -.5774 \\ -.5774 \\ -.5774 \end{bmatrix}$$

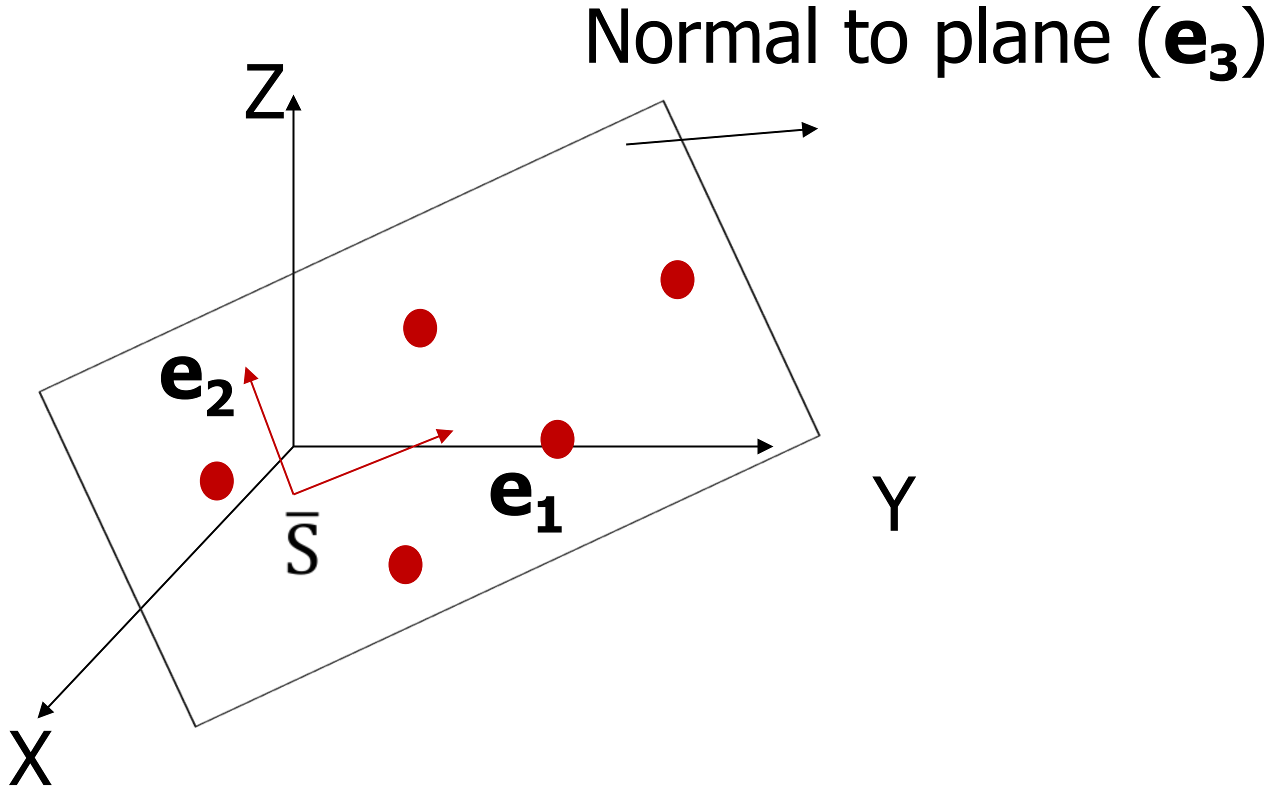
$$\mathrm{B} = [\mathbf{e_1} \quad \mathbf{e_2} \quad \mathbf{e_3}] = \begin{bmatrix} -.0055 & -.8165 & -.5774 \\ -.7043 & .413 & -.5774 \\ .7099 & .4034 & -.5774 \end{bmatrix}$$

$$\tilde{X}^T.\mathrm{B} = \begin{bmatrix} .2696 & -1.9615 & 0 \\ -3.2576 & -0.7128 & 0 \\ 15.8421 & 0.3825 & 0 \\ -.4126 & 1.7175 & 0 \\ -12.4415 & 0.5742 & 0 \end{bmatrix}$$

Points lying in the plane:
X+Y+Z=10

Redundant dimension

Coordinate transformation



PCA properties

- PCA diagonalizes the data covariance matrix Σ.
- $\Sigma = CDC^T$,
  - D: Diagonal matrix;
  - C: Columns are unit eigen vectors of Σ. → $CC^T=C^TC=I$
- Components are uncorrelated
  - As covariance among components is zero.
- By normalizing components with their variances (eigen values), Euclidean distance could be used for classification.
- Reconstr. error from lower dimensional space minimum among all linear transforms of the data.

Application of PCA

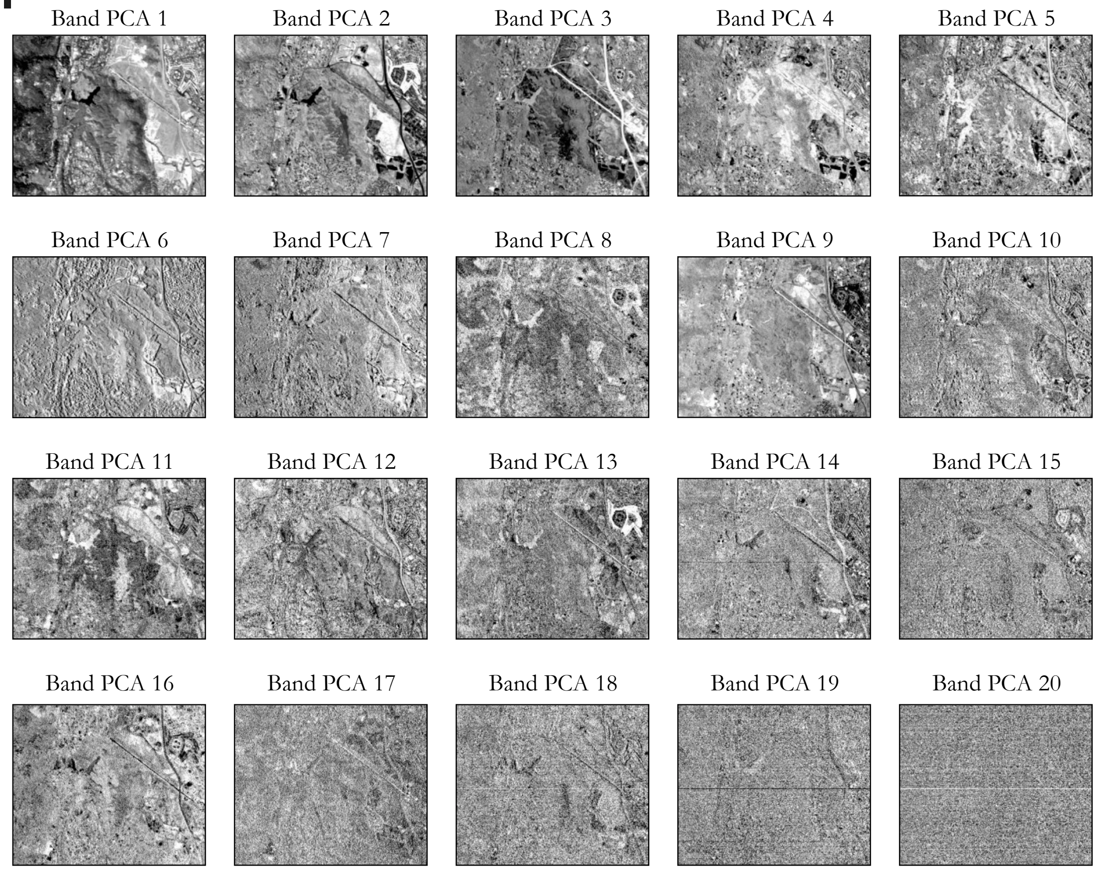
- Data compression
  - Provides optimum set of orthonormal basis vectors for a set of data points.
    - Data dependent.
    - Basis vectors also called as ‘Karhunen-Loeve’ basis, and the transform called ‘Karhunen-Loeve Transform’ (KLT).
    - Type-2 DCT basis vectors are approximately the eigen vectors of a 2-D matrix with (j,k) the entries as $r^{|j-k|}$.
      - Covariance matrix for a useful class of signals, where r is the measure of correlation between adjacent samples and a value near to 1.

Application of PCA

- Decorrelating components
  - Color images in RGB space highly correlated.
    - By performing PCA with different blocks of color images a color transformation matrix obtained, useful for segmentation.
      - (R+G+B)/3, R-B, (2G-R-B)/2
  - Multispectral, hyperspectral and ultraspectral remote sensing images.
    - Multispectral – 10's of bands
    - Hyperspectral – 100's of bands
    - Ultraspectral - 1000's of bands
    - PCA required to highlight decorrelated information.

Y.I. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation", Computer Graphics and Image Processing, 13, 222-241,

PCA components of a hyperspectral image



After component 20, not much details are available.

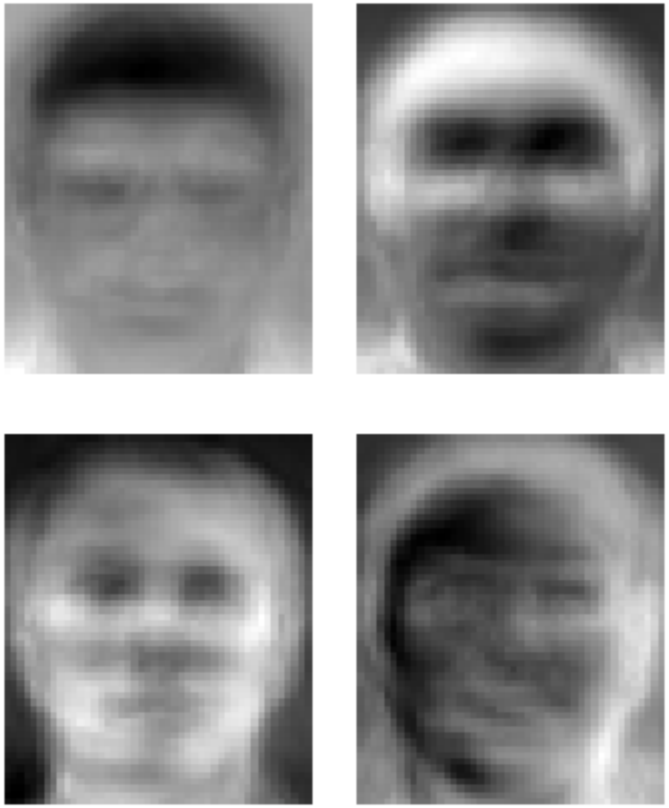
Removal of data redundancy.

Courtesy: Li et al, "A New Subspace Approach for Supervised Hyperspectral Image Classification", 2011 IEEE International Geoscience and Remote Sensing Symposium.

Application of PCA

- Factor analysis.
  - Highlights decorrelated factors.
    - Useful for classification.
  - For example, eigen faces for representing human faces.
    - Performs PCA on a large set of images of human faces cropped to the same size.
    - Any arbitrary face expressed as linear combination of them.
    - Coefficients of linear combination represent an arbitrary face.

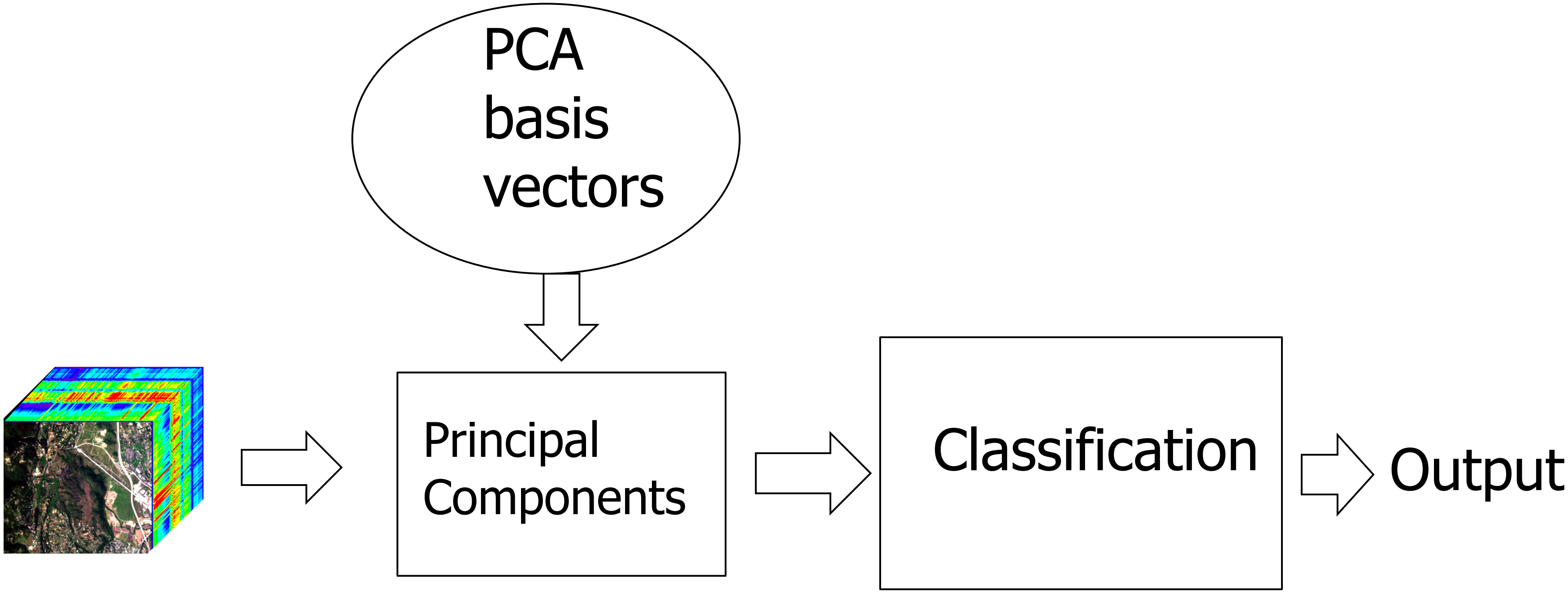
PCA: Eigen faces



http://en.wikipedia.org/wiki/Image:Eigenfaces.png

Application of PCA

- Classification / High level processing
  - Using the representation derived by principal component analysis.



PCA basis vectors

Principal Components

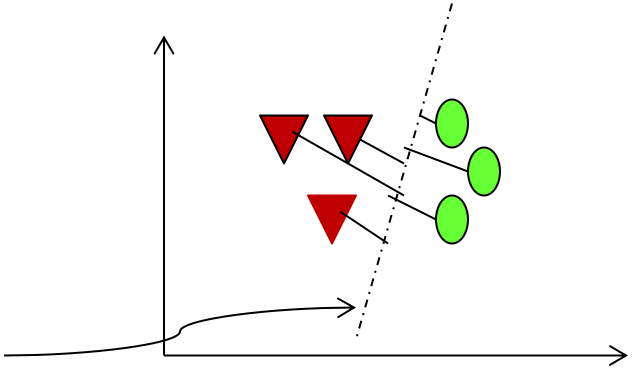
Classification

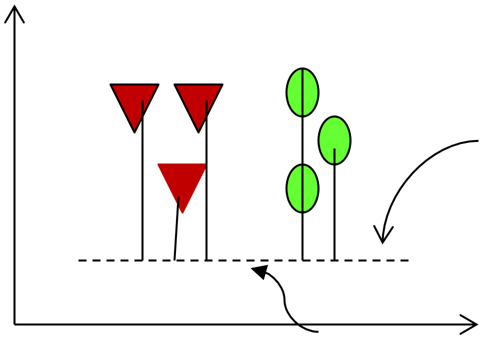
Output

Linear discriminant analysis

- For the purpose of classification, dimensional reduction using PCA may not work.
  - It captures the direction of maximum variance for a data set.
  - For labelled data sets, it does not capture the direction of maximum separation between the groups of data points of differing labels.

Direction for principal component.





Well separated but not along the direction of principal component.

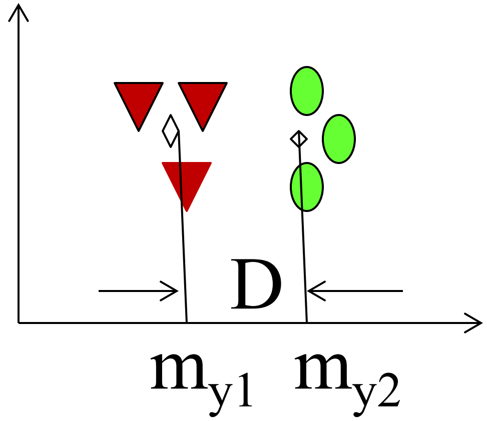
Fisher linear discriminant (FLD)

Fisher linear discriminant

- Consider a set of data points $S=\{x_i | x_i \text{ in } R^d\}$.
  - N_1 points in class w_1.
  - N_2 points in class w_2.
  - Say, $N_1+ N_2=N$ (total data points).
- Consider a line with direction u.
- Projection of data x_i on u: $y_i= x_i^T u$
  - One dimensional subspace representing data.

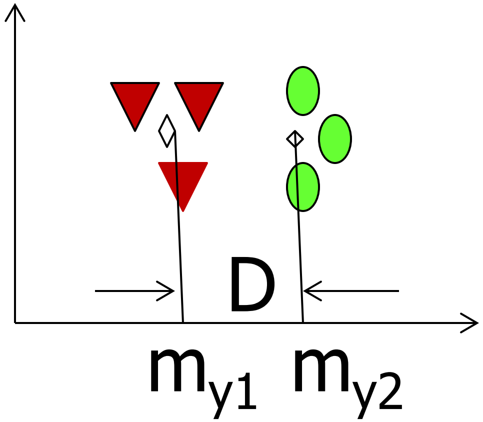
Separation between projected data of different classes

- m_1= mean of data points in w_1.
- m_2= mean of data points in w_2.
- Projection of means:
  - $m_{y1}=m_1^T u$
  - $m_{y2}=m_2^T u$
- A measure of separation:
  - $D=|m_{y1} - m_{y2}|$
  - Does not consider variance of data.



A better measure of separation

- Normalized by a factor proportional to class variances.
- Scatter of data belonging to class C:



$$s^2 = \sum_{y \epsilon C} (y - m_c)^2$$

Class Variance x Number of samples

Mean

- Measure of separation:

$$J(u) = \frac{D^2}{(s_1^2 + s_2^2)}$$

Scatter of class w1

Scatter of class w2

- To obtain u maximizing $\mathrm{J(u)}$.

Scatter of projected samples should be small.

Scatter matrix

- Scatter matrix for samples of class C in original space :

$$\mathrm{S_C} = \sum_{\mathrm{x} \in \mathrm{C}} (\mathrm{x} - \mathrm{m_C})\,(\mathrm{x} - \mathrm{m_C})^{\mathrm{T}}$$

Within the class Scatter matrix

Scatter matrixes for class w1 and w2.

Within the class scatter matrix: $S_w = S_1 + S_2$

$$s_1^2 = \sum_{y \epsilon W1} (y - m_{y1})^2 \Rightarrow \sum_{x \,\epsilon\, W_1} (u^T x - u^T m_1)(u^T x - u^T m_1)^T$$

$$\Downarrow$$

$$\sum_{x \,\epsilon\, W_1} u^T (x - m_1)(x - m_1)^T u$$

$$\swarrow$$

$$u^T \Big(\sum_{x \,\epsilon\, W_1} (x - m_1)(x - m_1)^T \Big) u$$

$$\Uparrow$$

$$u^T S_1 u \longleftrightarrow s_1^2$$

$$\Longrightarrow \quad s_1^2 + s_2^2 = u^T S_w u$$

Between the class scatter matrix

Between the class scatter matrix:

Means of w1 and w2

$$S_B = (m_1 - m_2)(m_1 - m_2)^T$$

$$\begin{tikzcd}
D^2 = (m_{y1} - m_{y2})^2 \arrow[r, Rightarrow] & (u^T m_1 - u^T m_2)(u^T m_1 - u^T m_2)^T \arrow[d, Rightarrow] \\
u^T S_B u \arrow[u, leftrightarrow] & u^T (m_1 - m_2)(m_1 - m_2)^T u \arrow[l, Rightarrow]
\end{tikzcd}$$

Rewriting optimization function

To maximize $J(u) = \frac{D^2}{(s_1^2 + s_2^2)}$ $\Rightarrow$ $J(u) = \frac{u^T S_B u}{u^T S_W u}$

Solution

To maximize $J(u) = \dfrac{D^2}{(s_1^2 + s_2^2)}$ ⇒ $J(u) = \dfrac{u^T S_B u}{u^T S_W u}$

u should be such that $S_W^{-1} S_B u = \lambda u$

Should be invertible Eigen value problem.

For any vector z, $S_B z = k.(m_1 - m_2)$

$\Uparrow$

$$(m_1-m_2)\underbrace{(m_1-m_2)^T z}_{k} = k\,(m_1-m_2)$$

⇒ $u = S_W^{-1}\,(m_1-m_2)$ Only direction matters.

Example

- Data points:
  - X_1={(5, 3, 2), (4, 6, 0), (3, -7, 14)}
  - X2={(-2 -5 17), (3 -13 10), (-4 -2 16)}
- Perform LDA and get the optimum direction. Check separability in the line of projection.
- Perform PCA on the whole data set ignoring class information and get the dominant principal direction. Check the separability of projected points on it.

Example (contd.)

- LDA: $X1 = \begin{bmatrix} 5 & 4 & 3 \\ 3 & 6 & -7 \\ 2 & 0 & 14 \end{bmatrix}$ $X2 = \begin{bmatrix} -2 & 3 & -4 \\ -5 & -13 & -2 \\ 17 & 10 & 16 \end{bmatrix}$

$$\mu_1 = \begin{bmatrix} 4 & 4 & 4 \\ .67 & .67 & .67 \\ 5.33 & 5.33 & 5.33 \end{bmatrix}$$

$$S_1 = (X1 - \mu_1)(X1 - \mu_1)^T$$

$$S1 = \begin{bmatrix} 2 & 10 & -12 \\ 10 & 92.66 & -102.67 \\ -12 & -102.67 & 114.67 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} -1 & -1 & -1 \\ .-6.67 & -6.67 & -6.67 \\ 14.33 & 14.33 & 14.33 \end{bmatrix}$$

$$S2 = \begin{bmatrix} 26 & -41 & -25 \\ -41 & 64.67 & 39.67 \\ -25 & 39.67 & 28.66 \end{bmatrix}$$

SW=S1+S2

$$Sw = \begin{bmatrix} 28 & -31 & -37 \\ -31 & 157.33 & -63 \\ -37 & -63 & 143.33 \end{bmatrix}$$

$$u = SW^{-1}(\text{mean1} - \text{mean2})$$

$$u = \begin{bmatrix} 3.2070 \\ -1.1952 \\ 1.2904 \end{bmatrix}$$

Example (contd.)

- LDA: Separability

$$u = \begin{bmatrix} 3.2070 \\ -1.1952 \\ 1.2904 \end{bmatrix}$$

$$Y1 = X1^T u$$

$$Y1 = \begin{bmatrix} 22.2 \\ 19.99 \\ 19.31 \end{bmatrix}$$

$\Updownarrow$ Well separated.

$$Y2 = X2^T u$$

$$Y2 = \begin{bmatrix} 9.55 \\ 6.99 \\ 5.43 \end{bmatrix}$$

- PCA: $X = \begin{bmatrix} 5 & 4 & 3 & -2 & 3 & -4 \\ 3 & 6 & -7 & -5 & -13 & -2 \\ 2 & 0 & 14 & 17 & 10 & 16 \end{bmatrix}$

$$\bar{S} = \begin{bmatrix} 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ -3 & -3 & -3 & -3 & -3 & -3 \\ 9.83 & 9.83 & 9.83 & 9.83 & 9.83 & 9.83 \end{bmatrix} \quad C = \begin{bmatrix} 10.92 & 4 & -17.42 \\ 4 & 39.67 & -27 \\ -17.42 & -27 & 44.14 \end{bmatrix}$$

Eigen values: 72.96, 20.29, 1.47

Eigen vectors:

$$\mathbf{e_1} = \begin{bmatrix} -0.25 \\ -0.63 \\ 0.74 \end{bmatrix} \qquad \mathbf{e_2} = \begin{bmatrix} -.52 \\ .73 \\ .44 \end{bmatrix} \qquad \mathbf{e_3} = \begin{bmatrix} -0.82 \\ -0.27 \\ -.51 \end{bmatrix}$$

Example (contd.)

- PCA: Separability

$$\mathbf{e_1} = \begin{bmatrix} -0.25 \\ -0.63 \\ 0.74 \end{bmatrix}$$

$$\mathrm{Z1} = \mathrm{X1}^{\mathrm{T}}\mathbf{e_1} \qquad \mathrm{Z1} = \begin{bmatrix} -1.65 \\ -4.76 \\ 13.98 \end{bmatrix}$$

$$\mathrm{Z2} = \mathrm{X2}^{\mathrm{T}}\mathbf{e_1} \qquad \mathrm{Z2} = \begin{bmatrix} 16.18 \\ 14.8 \\ 14.05 \end{bmatrix}$$

Reduced margin of separation.

Summary

- Feature selection (Subset Selection)
  - Forward and backward sequential selection methods.
- Unsupervised dimension reduction method.
  - PCA
- Supervised dimension reduction method
  - LDA

Thank you!