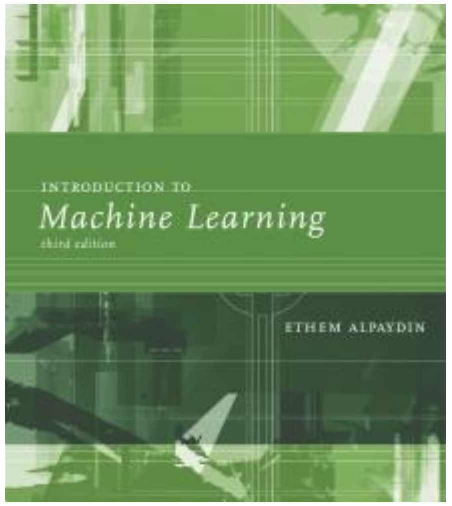


Dimensionality Reduction

Lecture notes by Ethem Alpaydın
Introduction to Machine Learning (Boğaziçi Üniversitesi)

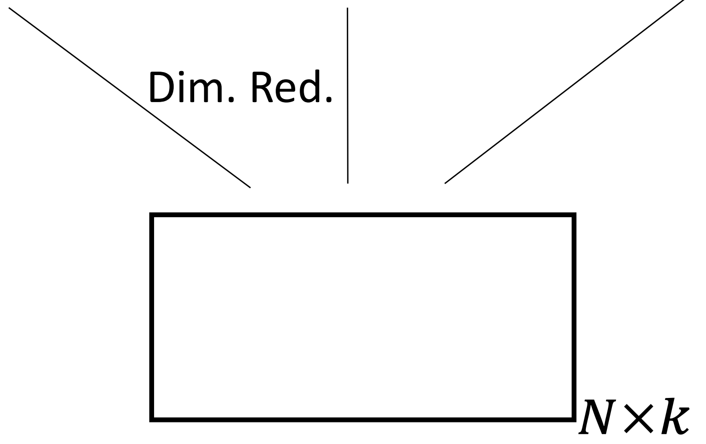
Lecture notes by Kevyn Collins-Thompson
Applied Machine Learning (Coursera)

Dimensionality Reduction

- Reduce the dimensionality of inputs
- Useful if there are correlation between input dimensionalities
- Find correlation between features and group features
- Reduce time and space complexity
- Easier to explain model
- Data visualization
- Speed up learning algorithm

Dimensionality Reduction

$$\begin{bmatrix} \left(x^{(1)}\right)^T \\ \cdots \\ \cdots \\ \left(x^{(N)}\right)^T \end{bmatrix}_{N\times d}$$



$$k < d$$

Dimensionality Reduction

Feature Selection

- Choosing $k<d$ important features, ignoring the remaining $d - k$

  Subset selection algorithms

Feature Extraction

- Project the

  original x_i , i =1,...,d dimensions

  to

  new $k<d$ dimensions, z_j , j =1,...,k

  - *Principal Components Analysis* (PCA)
  - *Linear Discriminant Analysis* (LDA)
  - …

Subset Selection

- We are interested in finding the best subset of the set of features.
- The best subset contains the least number of dimensions that most contribute to accuracy.
- Using a suitable error function, this can be used in both regression and classification problems.

Subset Selection

- There are 2^d possible subsets of d variables,
  - but we cannot test for all of them unless d is small and we employ heuristics to get a reasonable (but not optimal) solution in reasonable (polynomial) time.

Subset Selection

Forward Selection

Start with no variables and add them one by one, at each step adding the one that decreases the error the most, until any further addition does not decrease the error

Backward Selection

- Start with all variables and remove them one by one, at each step removing the one that decreases the error the most (or increases it only slightly), until any further removal increases the error significantly.

Subset Selection

- In either case, checking the error should be done on a validation set distinct from the training set because we want to test the generalization accuracy.

Subset Selection

- Let us denote by *F*, a feature set of input dimensions, x_i, $i = 1, . . . , d$.
- *E(F)* denotes the error incurred on the validation sample when only the inputs in *F* are used.
- Depending on the application, the error is either the mean square error or misclassification error.

Sequential Forward Selection

- Start with no features: $F = \emptyset$
- At each step, find the best new feature

$$j = \underset{i}{\operatorname{argmin}} E(F \cup x_i)$$

for all possible x_i, train our model on the training set and calculate the $\mathrm{E}(F \cup x_i)$ on the validation set, then choose that input x_j that causes the least error

- Add x_j to F if $E\left(F \cup x_j\right) < E(F)$
- Stop if adding any feature does not decrease *E*.

Sequential Forward Selection

- This process may be costly because to decrease the dimensions from d to k, we need to train and test the system

$$d+(d-1)+(d-2)+\cdots+(d-k) \text{ times,}$$

which is $O(d^2)$.

Sequential Forward Selection

- This is a local search procedure and does not guarantee finding the optimal subset, namely, the minimal subset causing the smallest error.
- For example, x_i and x_j by themselves may not be good but together may decrease the error a lot, but because this algorithm is greedy and adds attributes one by one, it may not be able to detect this.

Sequential Backward Selection

- Start with all features: $F = \{x_1, x_2, \dots, x_d\}$
- At each step, find the one that causes the least error

$$j = \underset{i}{\operatorname{argmin}}\, E(F - x_i)$$

Remove x_j to F if $E\left(F - x_j\right) < E(F)$

- Stop if removing any feature does not decrease *E*.

Feature Selection vs. Feature Extraction

- In an application like face recognition, feature selection is not a good method for dimensionality reduction because individual pixels by themselves do not carry much discriminative information; it is the combination of values of several pixels together that carry information about the face identity.

- This is done by feature extraction methods

Principal Components Analysis (PCA)

- In projection methods, we are interested in finding a mapping from the inputs in the original *d*-dimensional space to a new *(k < d)*-dimensional space, with minimum loss of information.
- The projection of ***x*** on the direction of ***w*** is

$$z = w^T x$$

Principal Components Analysis

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components.

Principal Components Analysis

Principal components analysis (PCA) is an unsupervised method in that it does not use the output information; the criterion to be maximized is the variance.

Principal Components Analysis

Find w such that Var(z) is maximized

$$
\begin{aligned}
\mathrm{Var}(z) = \mathrm{Var}(\boldsymbol{w}^T\boldsymbol{x}) &= \mathrm{E}[(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})^2] \\
&= \mathrm{E}[(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})] \\
&= \mathrm{E}[\boldsymbol{w}^T(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{w}] \\
&= \boldsymbol{w}^{\mathrm{T}}\,\mathrm{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T]\boldsymbol{w} = \boldsymbol{w}^T \boldsymbol{\sum} \boldsymbol{w}
\end{aligned}
$$

where $\mathrm{Var}(\boldsymbol{x}) = \mathrm{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T] = \boldsymbol{\sum}$

Principal Components Analysis

- The principal component w_1 is such that the sample, after projection on to w_1, is most spread out so that the difference between the sample points becomes most apparent.
- For a unique solution and to make the direction the important factor, we require $||w_1|| = 1$

$$\max w_1^T \Sigma w_1 \quad subject\ to\ \|w_1\| = 1$$

Principal Components Analysis

$$\max w_1^T \Sigma w_1 \quad subject\ to\ \|w_1\| = 1$$

Writing this as a Lagrange problem, we have

$$\max\ w_1^T \Sigma w_1 + \alpha(1 - w_1^T w_1)$$

Taking the derivative wrt w_1 and setting it equal to 0, we have

$$2\Sigma w_1 - 2\alpha w_1 = 0 \quad \rightarrow \quad \Sigma w_1 = \alpha w_1$$

which holds if w_1 is an eigenvector of Σ and α the corresponding eigenvalue.

Principal Components Analysis

Because we want to maximize variance

$$w_1^T \Sigma w_1 = w_1^T \alpha w_1 = \alpha w_1^T w_1 = \alpha . 1 = \alpha$$

we choose the eigenvector with the largest eigenvalue for the variance to be maximum.

Therefore the principal component is the eigenvector of the covariance matrix of the input sample with the largest eigenvalue, $\lambda_1 = \alpha$.

Principal Components Analysis

The second principal component, w_2, should also maximize variance, be of unit length, and be orthogonal to $w_1 (w_1^T w_2 = 0)$

$$\mathrm{z}_2 = w_2^T x$$

$$\max w_2^T \Sigma w_2 \ - \alpha(w_2^T w_2 - 1) - \beta(w_2^T w_1 - 0)$$

Taking the derivative wrt w_2 and setting it equal to 0, we have

$$2\Sigma w_2 - 2\alpha w_2 - \beta w_1 = 0$$

Principal Components Analysis

$$2\Sigma w_2 - 2\alpha w_2 - \beta w_1 = 0$$

Premultiply with w_1^T

$$2\mathrm{w}_1^{\mathrm{T}}\Sigma w_2 - 2\alpha w_1^T w_2 - \beta w_1^T w_1 = 0$$

Principal Components Analysis

Note that:

$w_1^T w_2 = 0$

$\mathrm{w}_1^{\mathrm{T}} \Sigma w_2$ is scalar, equal to its transponse $w_2^T \Sigma w_1$, where w_1 is the leading eigenvector of Σ, $\Sigma w_1 = \lambda_1 w_1$

$$\mathrm{w}_1^{\mathrm{T}} \Sigma w_2 = w_2^T \Sigma w_1 = \lambda_1 w_2^T w_1 = 0$$

Principal Components Analysis

$$2\mathrm{w}_1^{\mathrm{T}}\Sigma w_2 - 2\alpha w_1^T w_2 - \beta\, w_1^T w_1 = 2.0 - 2\alpha.0 - \beta.1 = 0$$
$$\beta = 0$$

Then, $\beta = 0$

$$2\Sigma w_2 - 2\alpha w_2 - \beta w_1 = 0 \quad \rightarrow \; \Sigma w_2 = \alpha w_2$$

which implies that w_2 should be the eigenvector of the covariance matrix of the input sample with the second largest eigenvalue, $\lambda_2 = \alpha$.

Principal Components Analysis

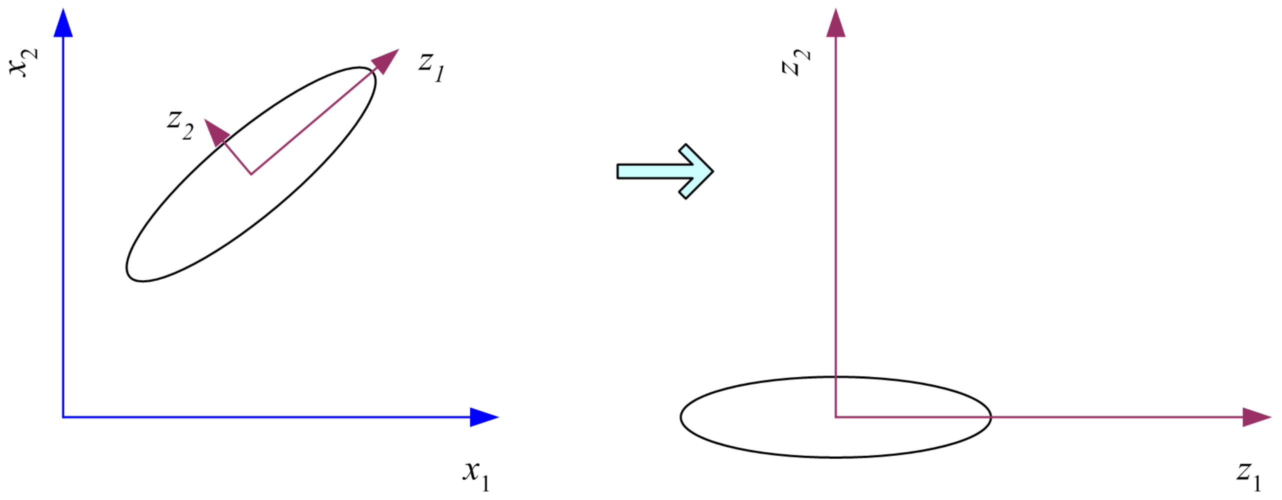
The first eigenvector w_1 (the principle component) explains the largest part of the variance; the second explains the second largest and so on.

What PCA does

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

where the columns of W are the eigenvectors of ∑ and m is sample mean

Centers the data at the origin and rotates the axes



Principal components analysis centers the sample and then rotates the axes to line up with the directions of highest variance. If the variance on z_2 is too small, it can be ignored and we have dimensionality reduction from two to one.

Step by Step PCA

(1) Data Preprocessing (Feature Scaling/Mean Normalization)

- More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables.
- That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges (For example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results.
- So, transforming the data to comparable scales can prevent this problem

$$new\_value \leftarrow \frac{value\ - mean}{standard\ deviation} \qquad x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{s_j}$$

Step by Step PCA

(2) Compute covariance matrix Σ

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Covariance matrix for 3-dimensional data

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

Step by Step PCA

(3) Compute the eigenvectors of matrix Σ

Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the ***principal components*** of the data

Step by Step PCA

(4) Feature vector

In this step, what we do is, to choose whether to keep all these components or discard those of lesser significance (of low eigenvalues), and form with the remaining ones a matrix of vectors that we call *Feature vector*.

So, the feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep.

This makes it the first step towards dimensionality reduction, because if we choose to keep only ***p*** eigenvectors (components) out of ***n***, the final data set will have only ***p*** dimensions

PCA

u^i :$eigenvectors$

$$U = [\, u^1 \quad u^2 \quad \dots \quad u^d]_{d \times d}$$

Take k vectors

$$U_{reduced} = [\, u^1 \quad u^2 \quad \dots \quad u^k]_{d \times k}$$

$$z = U_{reduced}^T . x$$

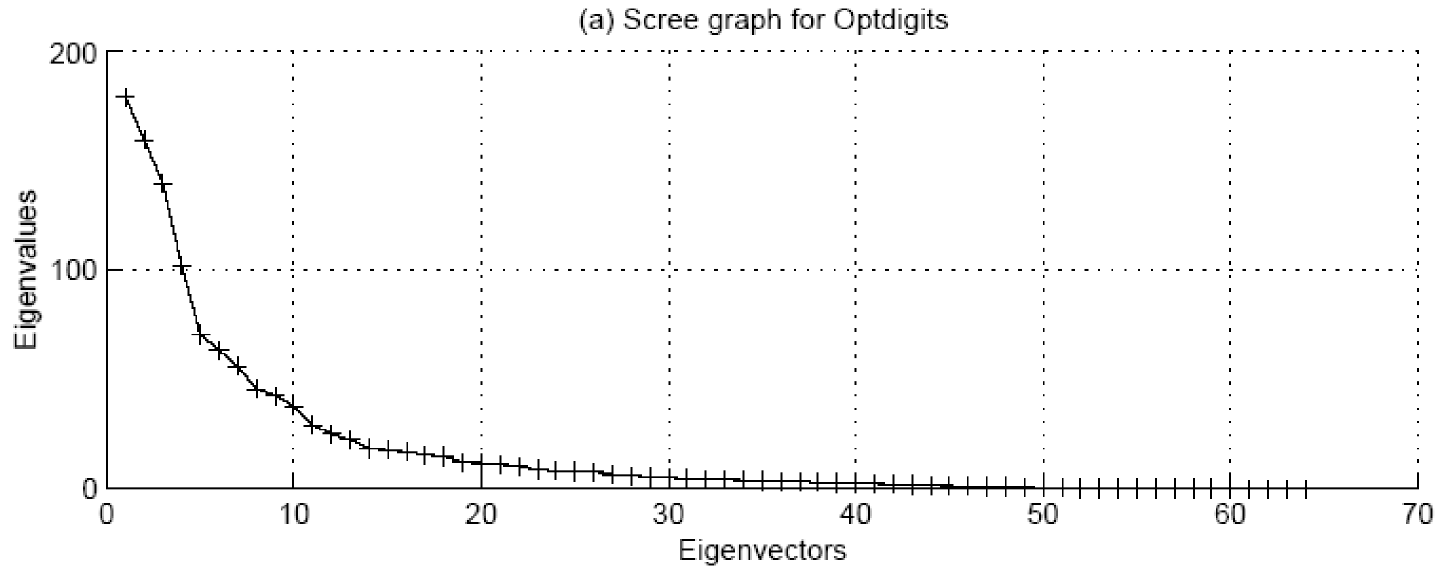
$$U_{reduced}^T = [\,]_{k \times d}, \qquad x = [\quad]_{d \times 1} \rightarrow z = [\quad]_{k \times 1}$$

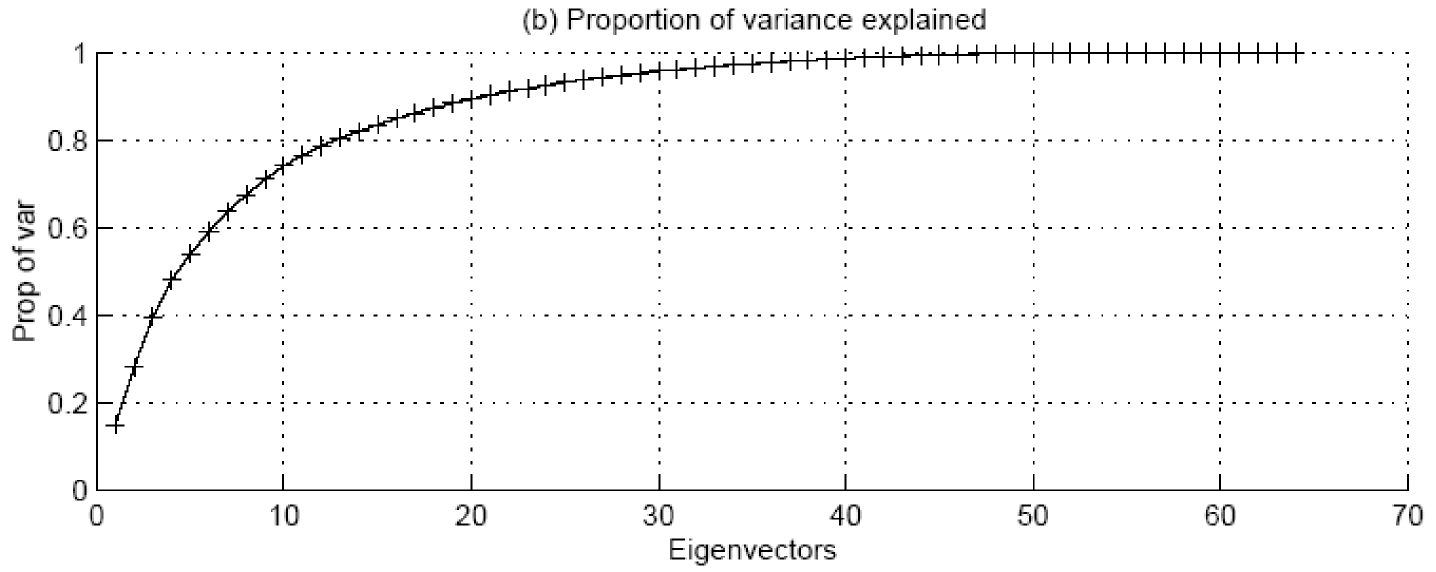
How to choose k ?

- Proportion of Variance (PoV) explained

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

  when λ_i are sorted in descending order
- Typically, stop at PoV>0.9
- Scree graph plots of PoV vs *k*, stop at "elbow"





This is a handwritten digit dataset with ten classes and sixty-four dimensional inputs.

The first twenty eigenvectors explain 90 percent of the variance

Linear Discriminant Analysis

- *Linear discriminant analysis* (LDA) is a supervised method for dimensionality reduction for classification problems.
- Given samples from two classes C1 and C2, we want to find the direction, as defined by a vector ***w***, such that when the data are projected onto ***w***, the examples from the two classes are as well separated as possible.

$$z = w^T x$$

Linear Discriminant Analysis

- $\mathbf{m_1}$ and m_1 are the means of sample from C_1 before and after projection, respectively.
- $\mathbf{m_1} \in R^d$ and $m_1 \in R$
- We are given a sample $X = \{x^t, r^t\}$ such that $r^t = 1$ if $x^t \in C_1$ and $r^t = 0$ if $x^t \in C_2$

$$m_1 = \frac{\sum_t w^T x^t r^t}{\sum_t r^t} = w^T \mathbf{m_1}$$

$$m_2 = \frac{\sum_t w^T x^t (1 - r^t)}{\sum_t (1 - r^t)} = w^T \mathbf{m_2}$$

Linear Discriminant Analysis

- The scatter of samples from C_1 and C_2 after projection

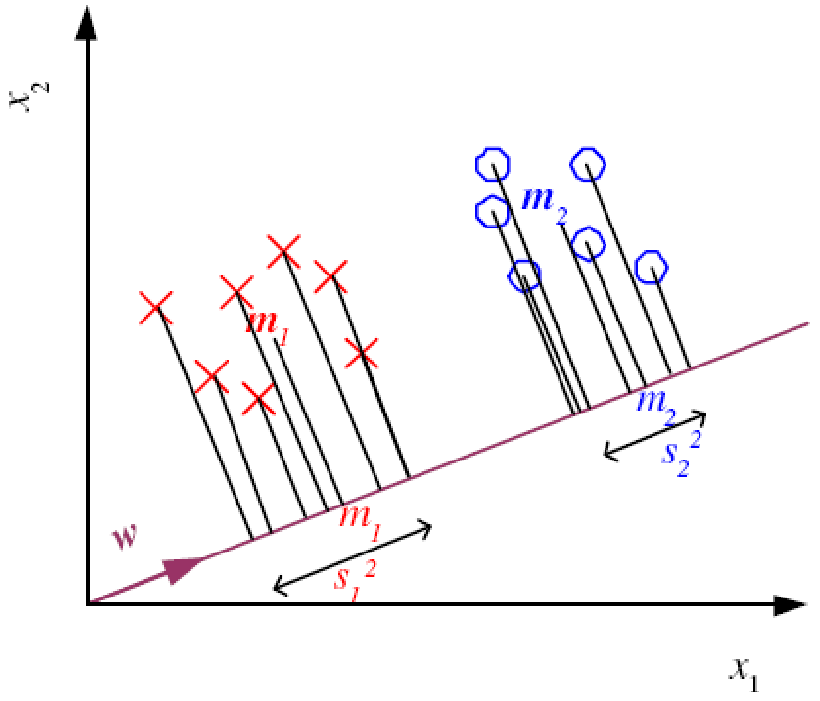
$$s_1^2 = \sum_t (w^T x^t - m_1)^2 \, r^t$$

$$s_2^2 = \sum_t (w^T x^t - m_2)^2 (1 - r^t)$$

Linear Discriminant Analysis

After projection, for the two classes to be well separated, we would like the means to be as far apart as possible and the examples of classes be scattered in as small a region as possible.

So we want $|m_1 - m_2|$ to be large and $s_1^2 + s_2^2$ to be small

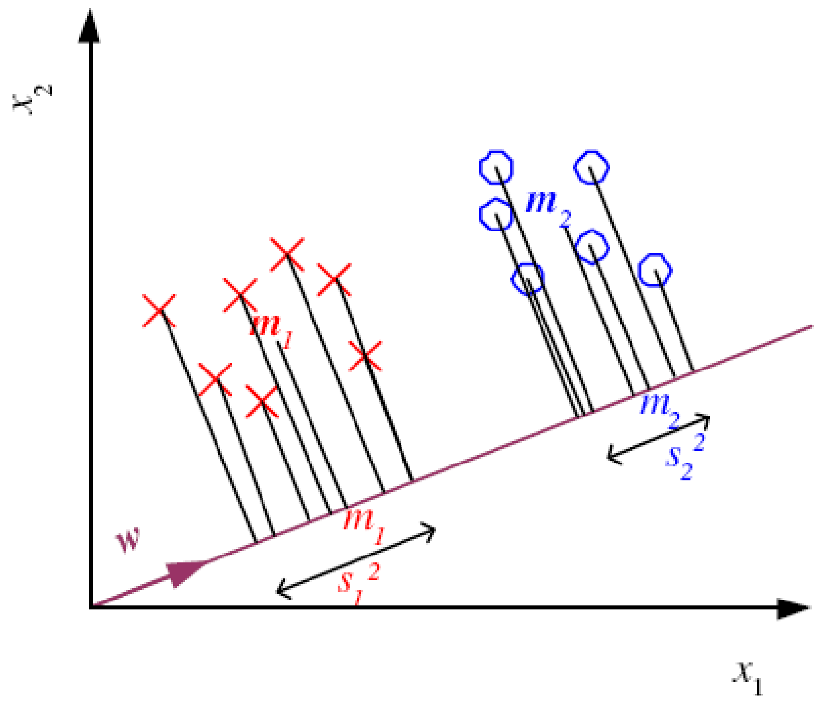


Linear Discriminant Analysis

Fisher's linear discriminant is ***w*** that maximizes

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

$$m_1 = \frac{\sum_t \mathbf{w}^T \mathbf{x}^t r^t}{\sum_t r^t} \qquad s_1^2 = \sum_t \left(\mathbf{w}^T \mathbf{x}^t - m_1\right)^2 r^t$$



Linear Discriminant Analysis

- Between-class scatter:

$$\begin{aligned}(m_1 - m_2)^2 &= \left(\mathbf{w}^T\mathbf{m}_1 - \mathbf{w}^T\mathbf{m}_2\right)^2 \\ &= \mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T\mathbf{w} \\ &= \mathbf{w}^T\mathbf{S}_B\mathbf{w} \text{ where } \mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T\end{aligned}$$

- Within-class scatter:

$$\begin{aligned}s_1^2 &= \sum_t \left(\mathbf{w}^T\mathbf{x}^t - m_1\right)^2 r^t \\ &= \sum_t \mathbf{w}^T\left(\mathbf{x}^t - \mathbf{m}_1\right)\left(\mathbf{x}^t - \mathbf{m}_1\right)^T\mathbf{w}r^t = \mathbf{w}^T\mathbf{S}_1\mathbf{w}\end{aligned}$$

$$\text{where } \mathbf{S}_1 = \sum_t \left(\mathbf{x}^t - \mathbf{m}_1\right)\left(\mathbf{x}^t - \mathbf{m}_1\right)^T r^t$$

$$s_1^2 + s_1^2 = \mathbf{w}^T\mathbf{S}_W\mathbf{w} \text{ where } \mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$$

Fisher's Linear Discriminant

- Find ***w*** that max

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} = \frac{\left|\mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)\right|^2}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

- LDA soln:

$$\mathbf{w} = c \cdot \mathbf{S}_W^{-1} (\mathbf{m}_1 - \mathbf{m}_2)$$

- Parametric soln:

$$\mathbf{w} = \Sigma^{-1} (\mu_1 - \mu_2)$$

$$\text{when } p(\mathbf{x} | C_i) \sim \mathcal{N}(\mu_i, \Sigma)$$

K>2 Classes

- Within-class scatter:

$$\mathbf{S}_W = \sum_{i=1}^{K} \mathbf{S}_i \qquad \mathbf{S}_i = \sum_t r_i^t \left(\mathbf{x}^t - \mathbf{m}_i\right)\left(\mathbf{x}^t - \mathbf{m}_i\right)^T$$

- Between-class scatter:

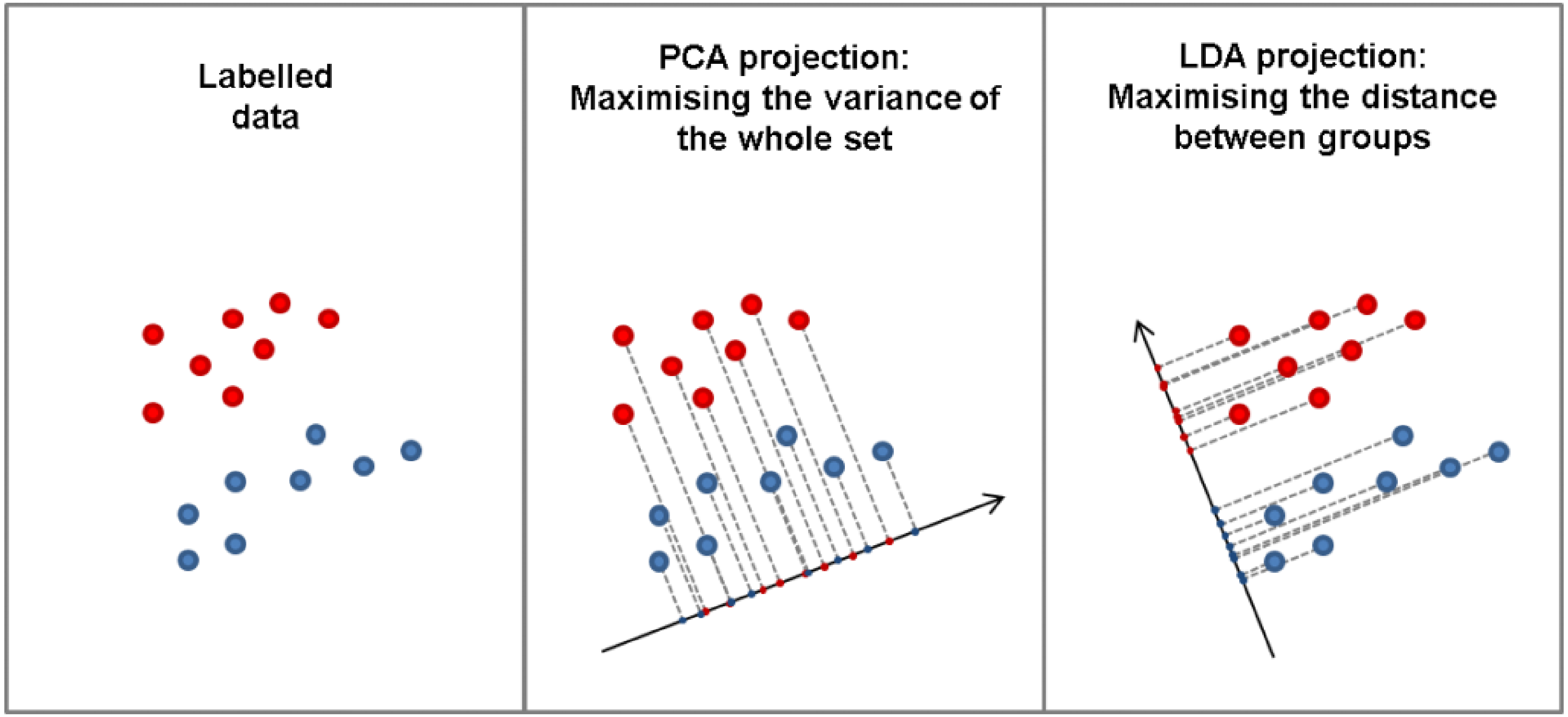
$$\mathbf{S}_B = \sum_{i=1}^{K} N_i \left(\mathbf{m}_i - \mathbf{m}\right)\left(\mathbf{m}_i - \mathbf{m}\right)^T \qquad \mathbf{m} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{m}_i$$

- Find **W** that max

$$J(\mathbf{W}) = \frac{\left|\mathbf{W}^T \mathbf{S}_B \mathbf{W}\right|}{\left|\mathbf{W}^T \mathbf{S}_W \mathbf{W}\right|}$$

The largest eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_{B;}$ maximum rank of K-1

PCA vs LDA


Labelled data
PCA projection: Maximising the variance of the whole set
LDA projection: Maximising the distance between groups