

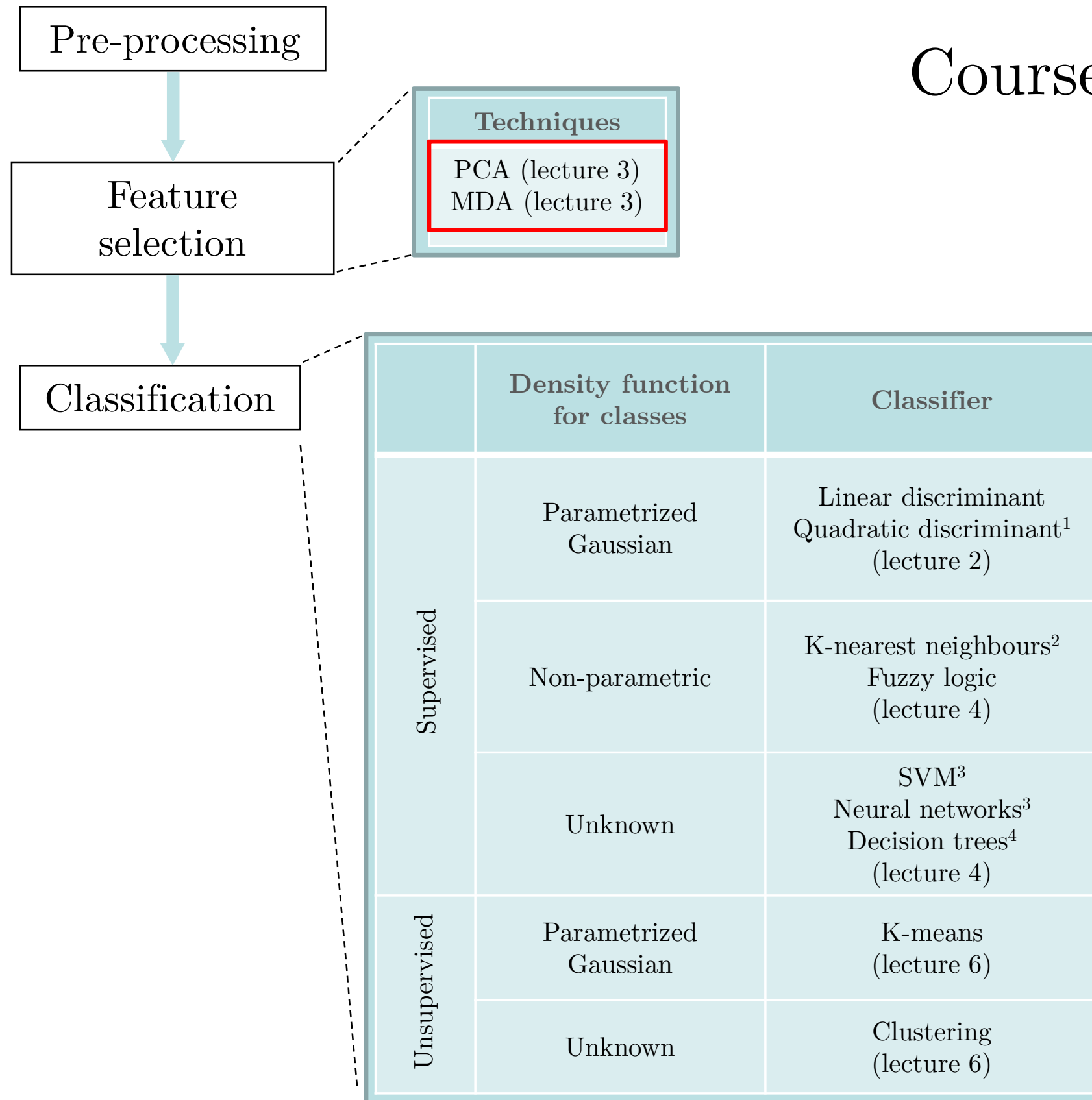
Chapter 3

Features selection

Recommended bibliography: *Pattern Classification (2nd ed)* by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000, Chapters 3.7 & 3.8

Credits: Some figures are taken from *Pattern Classification (2nd ed)* by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000 with the permission of the authors

Course overview



1. Useful only if covariance matrices are not rank deficient.
2. Useful with the number of features is very large, even larger than the number of training vectors.
3. Imposes a structure to the classifier irrespective of the training data base.
4. Useful when non-numeric features are present.

CONTENTS

- 3.1. Dimensionality problems**
- 3.2. Principal components analysis (PCA)**
- 3.3. Multiple discriminants analysis (MDA)**
- 3.4. Non-linear approaches**
- 3.5. Conclusions**

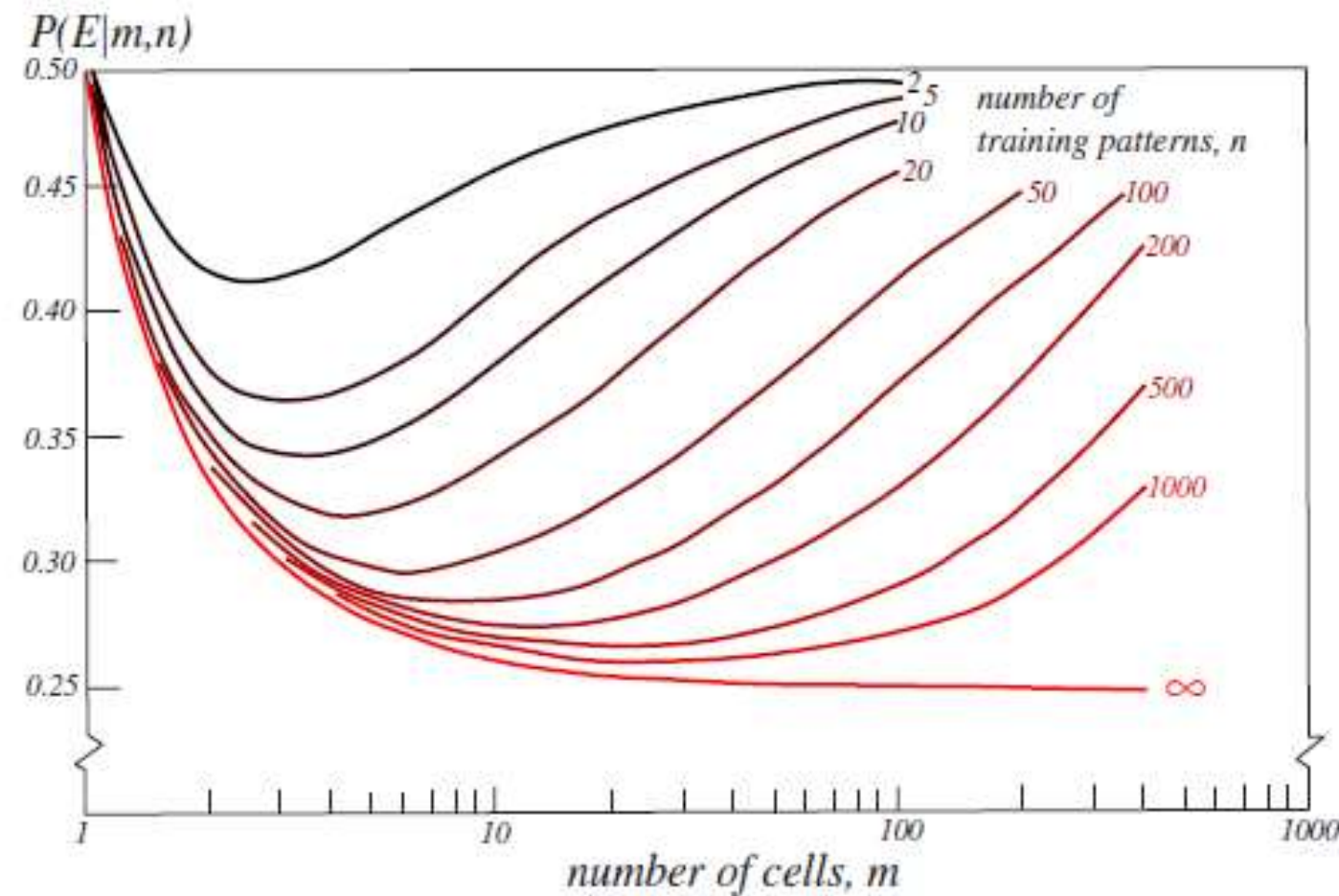
3.1 DIMENSIONALITY PROBLEMS

Deciding the number of relevant features is needed because...

- ⇒ ... it is possible that some features:
 - do not convey relevant information for the ML problema at hand,
 - are statistically dependent of others, or
 - do not match the assumed pdf.

- ⇒ ... considering too many features increase the complexity of classifier and makes training more difficult.

⇒ ... considering too many features may impair the probability of error if the size of the training data base is small.



In the case of classifiers requiring covariance matrix \mathbf{C} , if the number of vectors N is small compared to the number of features d , matrix \mathbf{C} can be rank deficient.

Reducing the number of relevant features is also useful for...

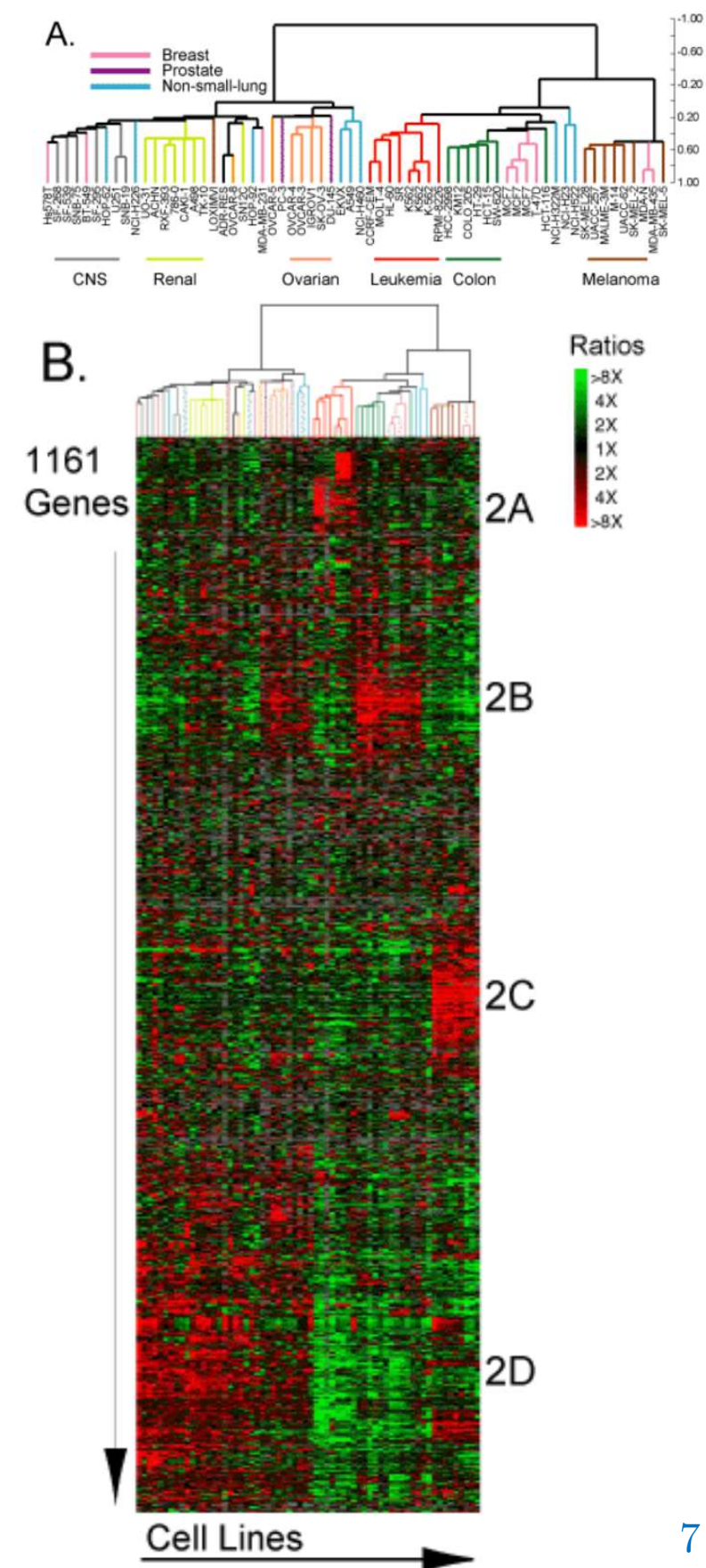
⇒ Data visualization

⇒ Data storage

Example 1. Microarrays data

```
NCI microarray data
Source and reference:
http://genome-www.stanford.edu/nci60/index.shtml
NCI microarray data
6830 genes
missing values have been imputed via SVD
60 cell lines, labels are below

1: CNS          5 samples
2: RENAL        7 samples
3: BREAST       9 samples
4: NSCLC        9 samples
5: UNKNOWN      1 samples
6: OVARIAN      6 samples
7: MELANOMA     8 samples
8: PROSTATE     2 samples
9: LEUKEMIA     6 samples
10: K562B-repro 1 samples
11: K562A-repro 1 samples
12: COLON       7 samples
13: MCF7A-repro 1 samples
14: MCF7D-repro 1 samples
```



WARNING! Reducing the number of features can increase the misclassification error rate. In fact, the error can be reduced to zero but only if the number of independent features is large.

Example 2. Classification task with two equiprobable Gaussian classes:

$$f_{\mathbf{x}}(\mathbf{x} \mid \omega_i) \sim N(\boldsymbol{\mu}_i; \mathbf{C}) \quad i = 1, 2 \quad \mathbf{x} \in \mathbb{R}^d$$
$$\Pr(\omega_1) = \Pr(\omega_2)$$

The probability of error is defined as:

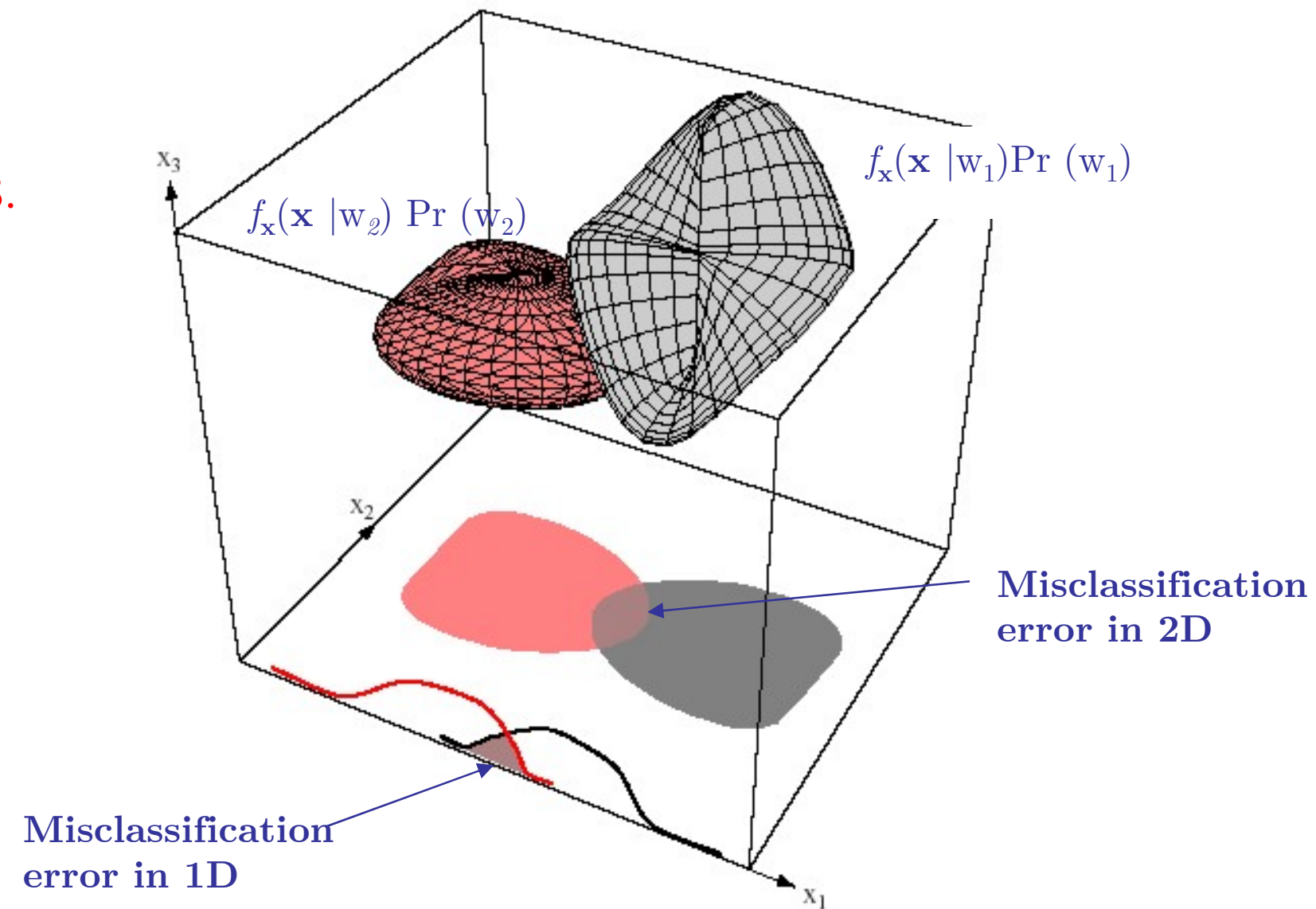
$$\Pr(e) = \frac{1}{\sqrt{2\pi}} \int_{r/2}^{\infty} \exp(-u^2 / 2) du$$
$$r^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \mathbf{C}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

If features are independent, the covariance matrix is diagonal:

$$\mathbf{C} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_d^2 \end{bmatrix} \quad r^2 = \sum_{i=1}^d \left(\frac{\mu_{i,1} - \mu_{i,2}}{\sigma_i} \right)^2$$

If features are relevant to the problem, if $d \rightarrow \infty$, $r \rightarrow \infty$, $\Pr(e) \rightarrow 0$

Example 3.



In this case, the pdf of the classes overlap when reduced to one feature (x_1) or two features (x_1, x_2). When a third feature is added (x_3), the two pdf are completely separated and misclassification error is zero.

Solutions:

Reducing the number of features: reducing the size of vector \mathbf{x} according to some criterion (in this lecture!).

Random forest: generate K classifiers by randomly choosing on each $d' < d$ features, and combine the K decisions (in chapter 5).

We will study the reduction of the number of features through a linear transform:

$$\mathbf{z}_k = \mathbf{W}^T (\mathbf{x}_k - \mathbf{a}) \quad \mathbf{x} \in \mathbb{R}^d, \mathbf{z} \in \mathbb{R}^{d'}, \mathbf{W}^T \in \mathbb{R}^{d' \times d} \quad d' < d$$

Possible solutions for \mathbf{W} :

1. Project vectors \mathbf{x}_k on the subspace that minimizes the reconstruction mean squared error (MSE)
 \Rightarrow principal components analysis (PCA)
2. Project vectors \mathbf{x}_k on the subspace that maximizes the separation between the resulting classes
 \Rightarrow multiple discriminant analysis (MDA)

3.2. PRINCIPAL COMPONENTS ANALYSIS (PCA)

We have N feature vectors associated to all classes:

$$D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \quad \mathbf{x} \in \mathbb{R}^d$$

Let us look for the matrix \mathbf{W} that best approximates the original vectors

$$\mathbf{x}_k \cong \mathbf{W}\mathbf{z}_k + \mathbf{a}$$

in the mean squared error sense:

$$J = \sum_{k=1}^N \|\mathbf{W}\mathbf{z}_k + \mathbf{a} - \mathbf{x}_k\|_2^2$$

Result: the reduced dimension vectors are given by:

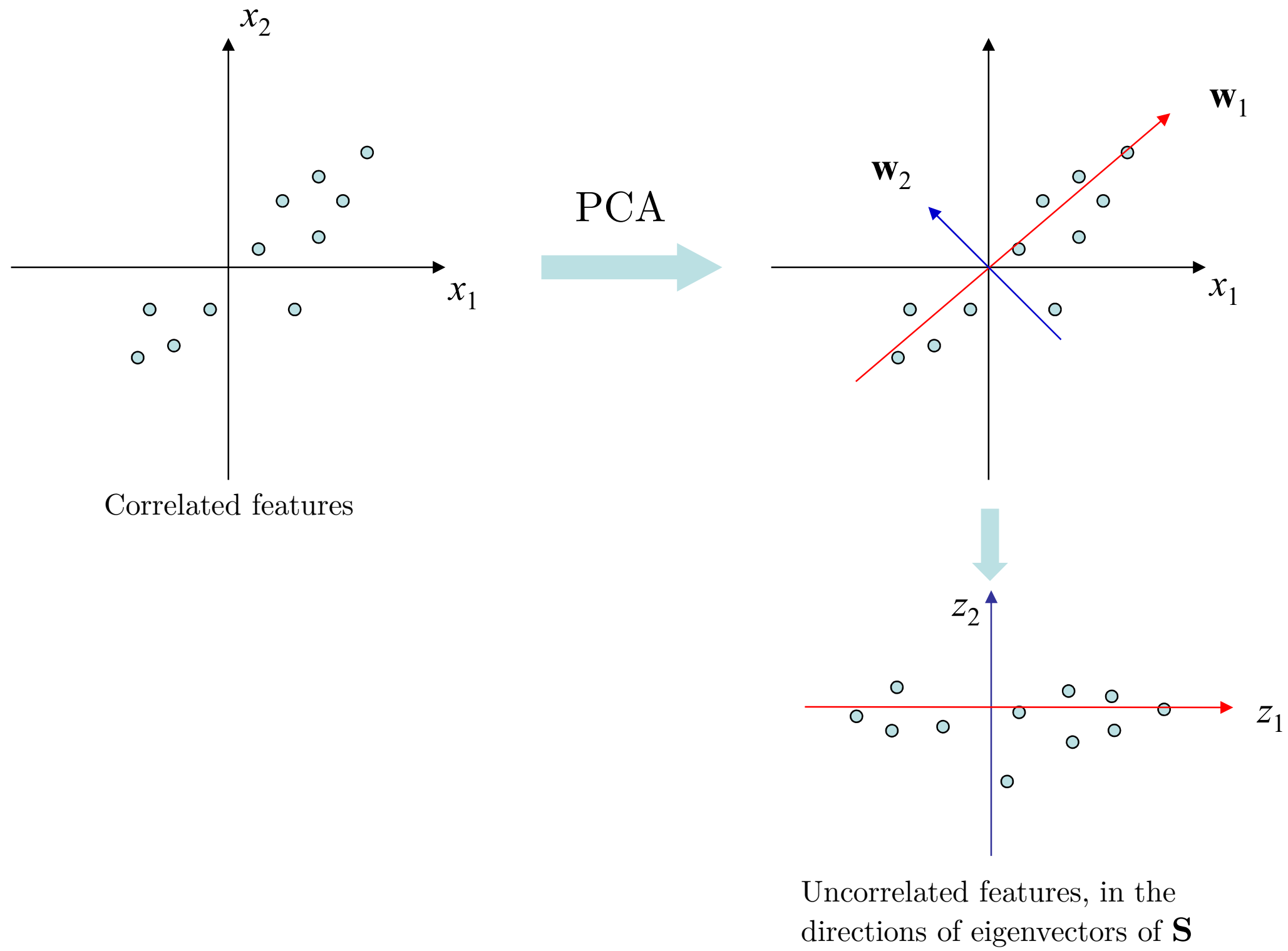
$$\mathbf{z}_k = \mathbf{W}^T (\mathbf{x}_k - \mathbf{m}) \quad \mathbf{m} = \mathbf{a} = \frac{1}{N} \sum_{k=1}^N \mathbf{x}_k$$

and the transformation matrix is $\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_{d'}]$

where vectors \mathbf{w}_j are:

$$\mathbf{S} \mathbf{w}_j = \lambda_j \mathbf{w}_j$$
$$\mathbf{S} = \sum_{k=1}^N (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T$$

The minimization of the MSE implies using in \mathbf{W} the eigenvectors associated to the largest eigenvalues.



Proof:



$$J = \sum_{k=1}^N \|\mathbf{W}\mathbf{z}_k + \mathbf{m} - \mathbf{x}_k\|_2^2 =$$

$$= \sum_{k=1}^N \left(\mathbf{z}_k^T \mathbf{W}^T \mathbf{W} \mathbf{z}_k + \mathbf{z}_k^T \mathbf{W}^T \mathbf{m} - \mathbf{z}_k^T \mathbf{W}^T \mathbf{x}_k + \mathbf{m}^T \mathbf{W} \mathbf{z}_k + \mathbf{m}^T \mathbf{m} - \mathbf{m}^T \mathbf{x}_k - \mathbf{x}_k^T \mathbf{W} \mathbf{m} + \mathbf{x}_k^T \mathbf{x}_k \right)$$

$$\nabla_{\mathbf{y}_k} J = 2\mathbf{W}^T \mathbf{W} \mathbf{z}_k + 2\mathbf{W}^T (\mathbf{m} - \mathbf{x}_k) = \mathbf{0}$$

$$\mathbf{z}_k = (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T (\mathbf{x}_k - \mathbf{m})$$

Inject the result in J to obtain the optimum column vectors in \mathbf{W} . After some algebra and removing the terms that do not contain \mathbf{W} ...

$$\hat{J} = -\sum_{k=1}^N (\mathbf{x}_k - \mathbf{m})^T \mathbf{W} (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T (\mathbf{x}_k - \mathbf{m})$$

Let us inspect the structure of matrix $\mathbf{W} (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T$ using the *singular value decomposition* (SVD) of \mathbf{W} ...

$$\mathbf{W} = \mathbf{U} \tilde{\Gamma} \mathbf{V}^T$$

Orthogonal matrices Diagonal matrix of non-negative elements

$$\tilde{\Gamma} = \begin{bmatrix} \Gamma \\ \mathbf{0} \end{bmatrix}$$

$$\begin{aligned}
\mathbf{W}(\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T &= \mathbf{U} \tilde{\mathbf{\Gamma}} \mathbf{V}^T \left(\mathbf{V} \tilde{\mathbf{\Gamma}}^T \mathbf{U}^T \mathbf{U} \tilde{\mathbf{\Gamma}} \mathbf{V}^T \right)^{-1} \mathbf{V} \tilde{\mathbf{\Gamma}}^T \mathbf{U}^T = \mathbf{U} \tilde{\mathbf{\Gamma}} \mathbf{V}^T \left(\mathbf{V} \tilde{\mathbf{\Gamma}}^T \tilde{\mathbf{\Gamma}} \mathbf{V}^T \right)^{-1} \mathbf{V} \tilde{\mathbf{\Gamma}}^T \mathbf{U}^T = \text{💡} \\
&= \mathbf{U} \tilde{\mathbf{\Gamma}}^T \mathbf{V}^T \mathbf{V}^{-T} \left(\tilde{\mathbf{\Gamma}}^T \tilde{\mathbf{\Gamma}} \right)^{-1} \mathbf{V}^{-1} \mathbf{V} \tilde{\mathbf{\Gamma}}^T \mathbf{U}^T = \mathbf{U} \tilde{\mathbf{\Gamma}}^T \mathbf{V}^T \mathbf{V} \left(\tilde{\mathbf{\Gamma}}^T \tilde{\mathbf{\Gamma}} \right)^{-1} \mathbf{V}^T \mathbf{V} \tilde{\mathbf{\Gamma}}^T \mathbf{U}^T = \mathbf{U} \tilde{\mathbf{\Gamma}} \left(\tilde{\mathbf{\Gamma}}^T \tilde{\mathbf{\Gamma}} \right)^{-1} \tilde{\mathbf{\Gamma}}^T \mathbf{U}^T = \\
&= \mathbf{U} \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{0} \end{bmatrix} \left(\begin{bmatrix} \mathbf{\Gamma} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{0} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{\Gamma} & \mathbf{0} \end{bmatrix} \mathbf{U}^T = \mathbf{U} \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{0} \end{bmatrix} \mathbf{\Gamma}^{-2} \begin{bmatrix} \mathbf{\Gamma} & \mathbf{0} \end{bmatrix} \mathbf{U}^T = \begin{bmatrix} \mathbf{U}_L & \mathbf{U}_R \end{bmatrix} \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{0} \end{bmatrix} \mathbf{\Gamma}^{-2} \begin{bmatrix} \mathbf{\Gamma} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}_L^T \\ \mathbf{U}_R^T \end{bmatrix} = \\
&= \begin{bmatrix} \mathbf{U}_L & \mathbf{U}_R \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}_L^T \\ \mathbf{U}_R^T \end{bmatrix} = \begin{bmatrix} \mathbf{U}_L & \mathbf{U}_R \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}_L^T \\ \mathbf{U}_R^T \end{bmatrix} = \mathbf{U}_L \mathbf{U}_L^T
\end{aligned}$$

Inject this result in $J...$

$$\hat{J} = - \sum_{k=1}^N (\mathbf{x}_k - \mathbf{m})^T \mathbf{U}_L \mathbf{U}_L^T (\mathbf{x}_k - \mathbf{m})$$

Note that the singular values in $\mathbf{\Gamma}$ and matrix \mathbf{V} are irrelevant to our problem. Therefore, \mathbf{W} can be an orthogonal matrix $\mathbf{W} = \mathbf{U}_L$, and hence complexity is reduced...

$$\mathbf{W} = \arg \max_{\mathbf{W}} \sum_{k=1}^N (\mathbf{x}_k - \mathbf{m})^T \mathbf{W} \mathbf{W}^T (\mathbf{x}_k - \mathbf{m})$$



$$\begin{aligned}
\mathbf{W} &= \arg \max_{\mathbf{W}} \sum_{k=1}^N (\mathbf{x}_k - \mathbf{m})^T \mathbf{W} \mathbf{W}^T (\mathbf{x}_k - \mathbf{m}) = \\
&= \arg \max_{\mathbf{W}} \text{tr} \left(\sum_{k=1}^N (\mathbf{x}_k - \mathbf{m})^T \mathbf{W} \mathbf{W}^T (\mathbf{x}_k - \mathbf{m}) \right) = \\
&= \arg \max_{\mathbf{W}} \text{tr} \left(\sum_{k=1}^N \mathbf{W}^T (\mathbf{x}_k - \mathbf{m}) (\mathbf{x}_k - \mathbf{m})^T \mathbf{W} \right) = \\
&= \arg \max_{\mathbf{W}} \text{tr} \left(\mathbf{W}^T \sum_{k=1}^N (\mathbf{x}_k - \mathbf{m}) (\mathbf{x}_k - \mathbf{m})^T \mathbf{W} \right) = \\
&= \arg \max_{\mathbf{W}} \text{tr} (\mathbf{W}^T \mathbf{S} \mathbf{W}) = \arg \max_{\mathbf{W}} \sum_{i=1}^{d'} \mathbf{w}_i^T \mathbf{S} \mathbf{w}_i
\end{aligned}$$

We need a restriction on the vectors, since the function is quadratic and increasing. As an example, the norm of vectors can be limited...

$$\mathbf{w}_i^T \mathbf{w}_i = E$$

The optimization with restrictions entails building the Lagrangian...

$$\mathbf{W} = \arg \max_{\mathbf{W}} \sum_{i=1}^{d'} \mathbf{w}_i^T \mathbf{S} \mathbf{w}_i + \sum_{i=1}^{d'} \lambda_i (E - \mathbf{w}_i^T \mathbf{w}_i)$$



As a result, the optimum vectors are the eigenvectors of \mathbf{S} :

$$\mathbf{S}\mathbf{w}_i = \lambda_i \mathbf{w}_i$$

Among all the d eigenvectors, which ones do we have to choose? Let us replace the result in the objective function...

$$\sum_{i=1}^{d'} \mathbf{w}_i^T \mathbf{S} \mathbf{w}_i = \sum_{i=1}^{d'} \lambda_i \mathbf{w}_i^T \mathbf{w}_i = E \sum_{i=1}^{d'} \lambda_i$$

Since the eigenvalues of \mathbf{S} are positive, minimizing the function entails choosing the d' eigenvectors associated to the largest eigenvalues. The new feature vectors are

$$\mathbf{z}_k = \mathbf{W}^T (\mathbf{x}_k - \mathbf{m})$$

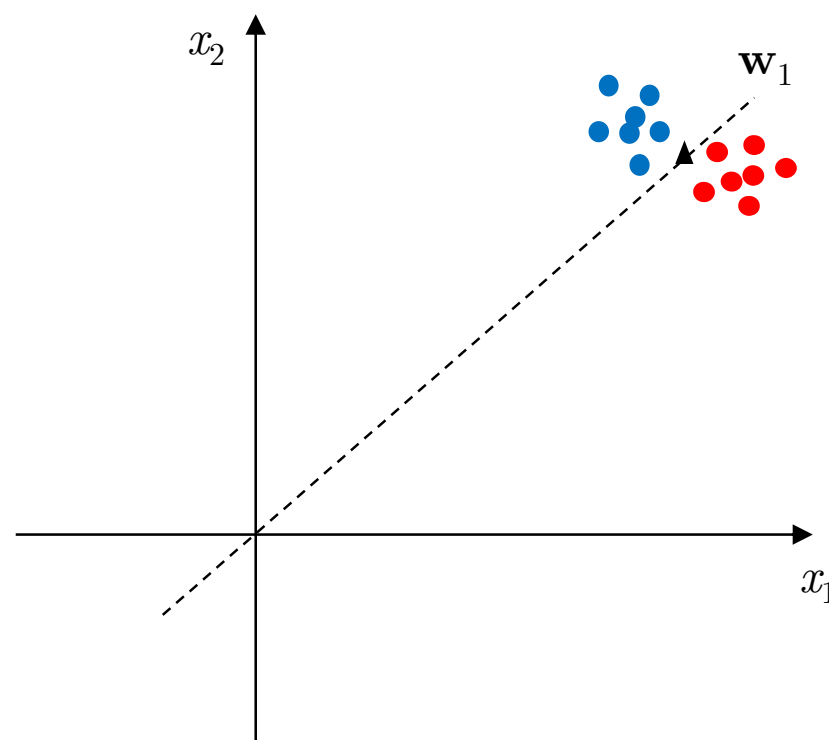
The minimum mean squared error can be computed as...

$$\begin{aligned} J_{\min} &= \sum_{k=1}^N \|\mathbf{W}\mathbf{z}_k + \mathbf{m} - \mathbf{x}_k\|_2^2 = \sum_{k=1}^N \|\mathbf{W}\mathbf{W}^T (\mathbf{x}_k - \mathbf{m}) + \mathbf{m} - \mathbf{x}_k\|_2^2 = \\ &= \sum_{k=1}^N \|(\mathbf{I} - \mathbf{W}\mathbf{W}^T)(\mathbf{x}_k - \mathbf{m})\|_2^2 = \dots = \sum_{i=d'+1}^d \lambda_i \end{aligned}$$

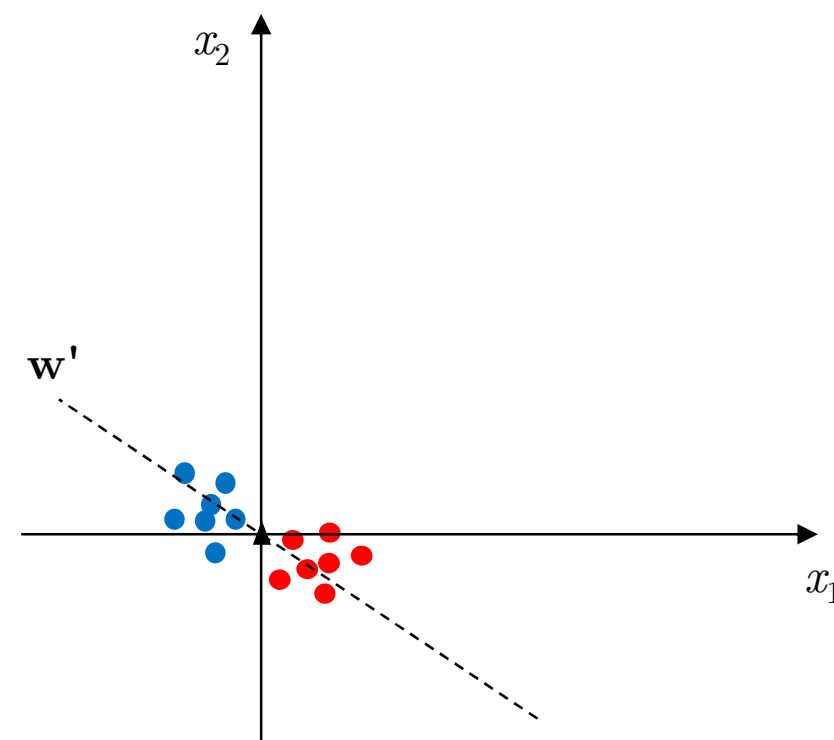
which is the sum of the discarded eigenvalues. ■

Why do we subtract the average of data? Note that this entails using the covariance instead of correlation matrix. In many situations it improves separability of classes.

Example 4. Reduction from 2 to 1 features for a two-class problem...



Classes are not separable after projection



Classes are separable after projection

Note that projection on w_1 generates a classification error that is larger than the error when projecting on w' !

Example 5. Face recognition

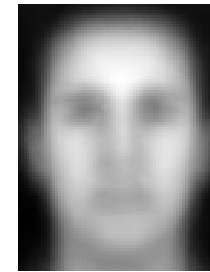
M. Turk and A. Pentland,
“Eigenfaces for recognition,”
J. Cognitive Neuroscience, 1991



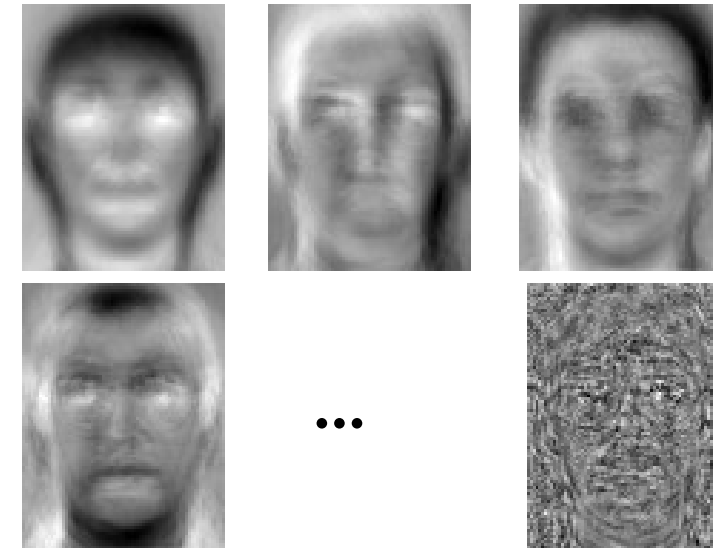
Image database (XM2VTS)



Mean

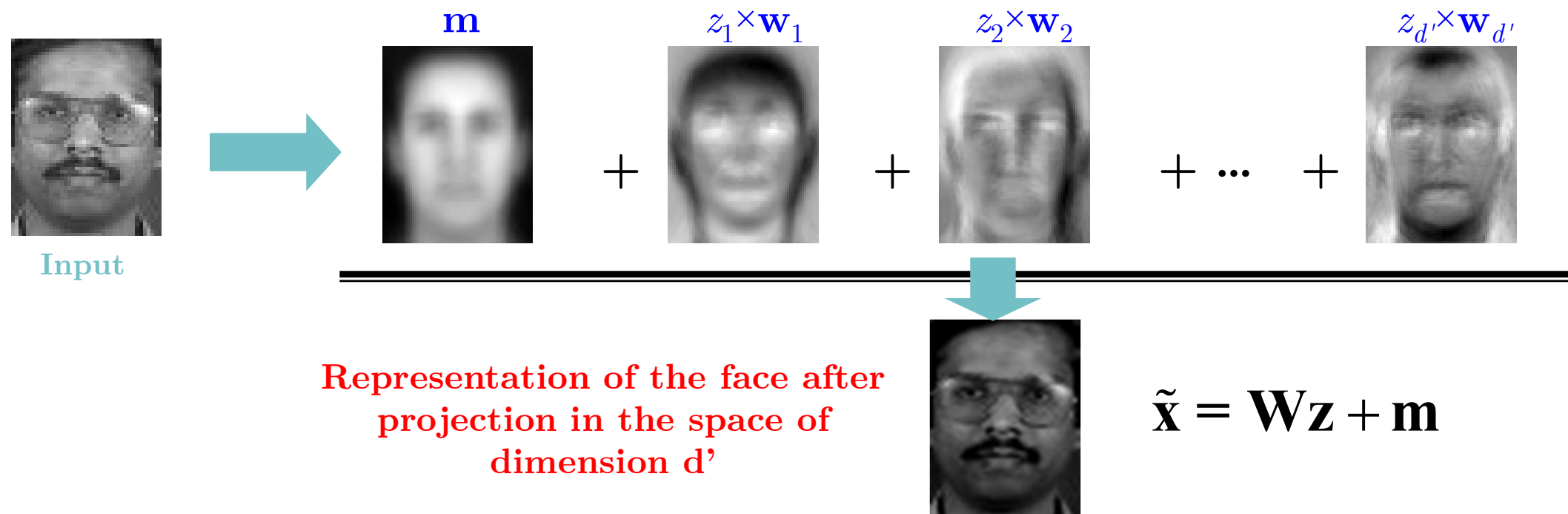


Largest eigenvectors



Smallest eigenvectors

The set of eigenvectors form a basis of the space of faces:



Property: The eigenvectors define the principal directions of a hiper-ellipsoid. They are ortogonal and point in the direction of the maximum/minimum dispersion.

Proof: Assume that the set of all classes were Gaussian, and let us take a level curve...

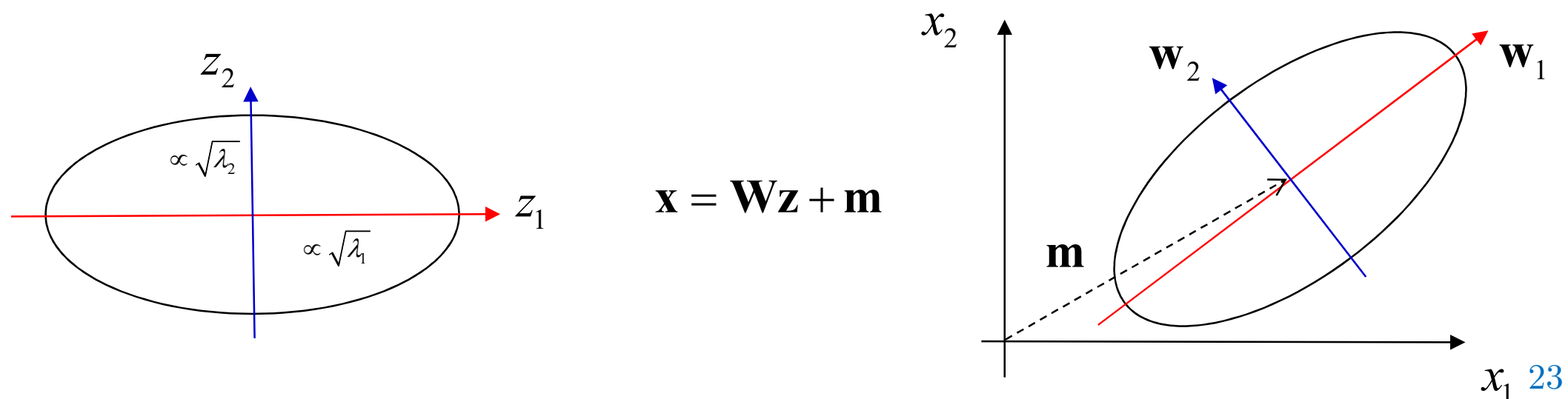
$$(\mathbf{x} - \mathbf{m})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{m}) = K$$

$$(\mathbf{x} - \mathbf{m})^T \mathbf{W} \underbrace{\Lambda^{-1} \mathbf{W}^T}_{\mathbf{y}} (\mathbf{x} - \mathbf{m}) = K$$

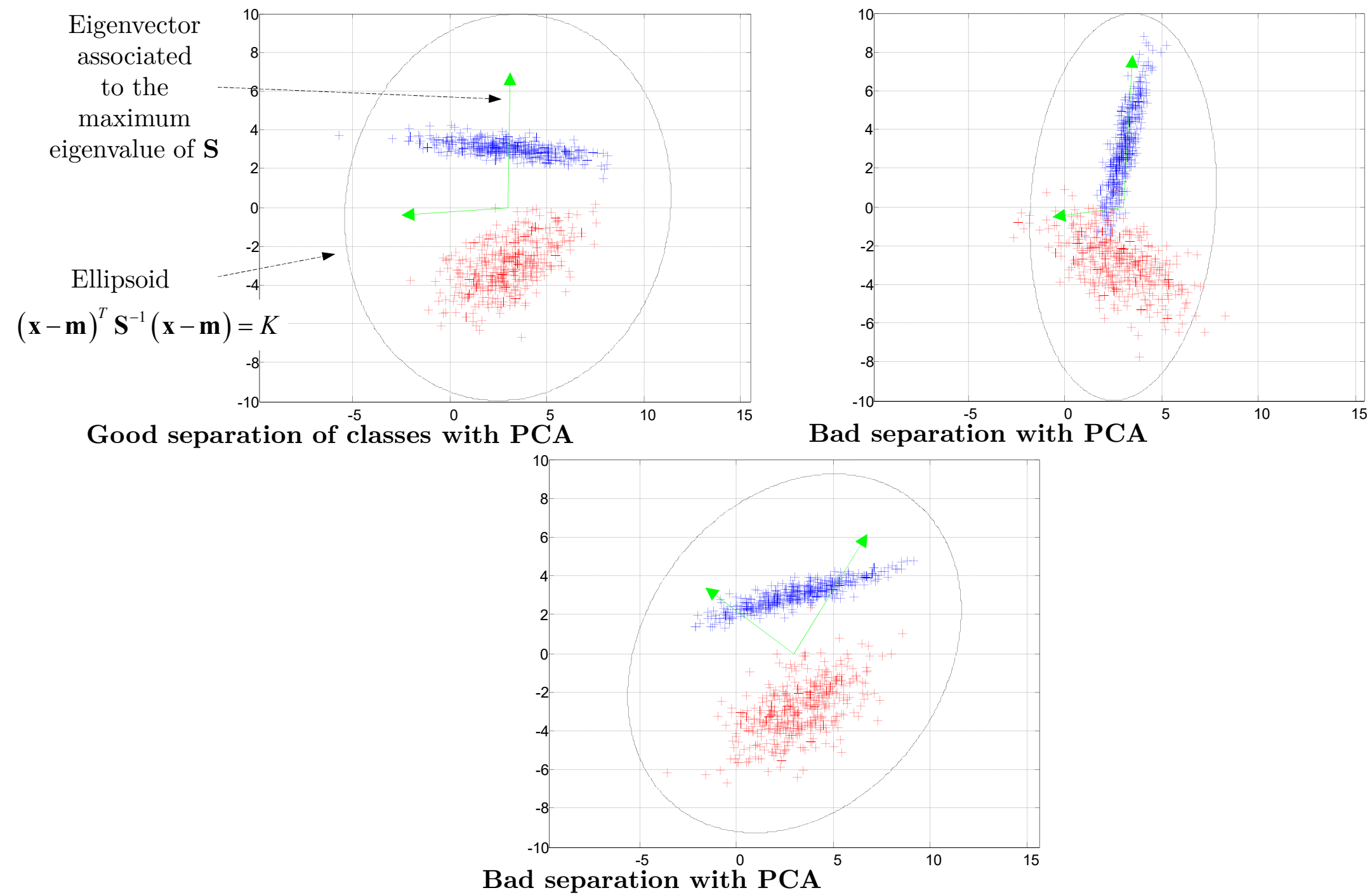
$$\mathbf{z}^T \Lambda^{-1} \mathbf{z} = K$$

$$\frac{z_1^2}{K\lambda_1} + \frac{z_2^2}{K\lambda_2} + \dots + \frac{z_d^2}{K\lambda_d} = 1 \quad \text{Ellipse whose semiaxis length is } \sqrt{K\lambda_i}$$

Keeping the largest eigenvectors, we keep the features of máximo variability of data. These directions are the eigenvectors of **S**:



PCA does not guarantee in every case a good separation of classes...

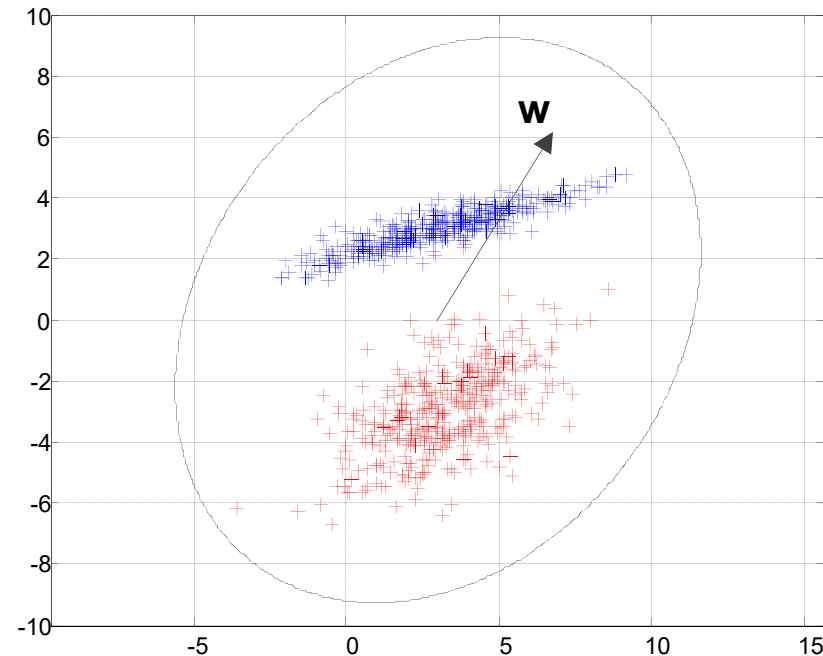


3.3. MULTIPLE DISCRIMINANTS ANALYSIS (MDA)

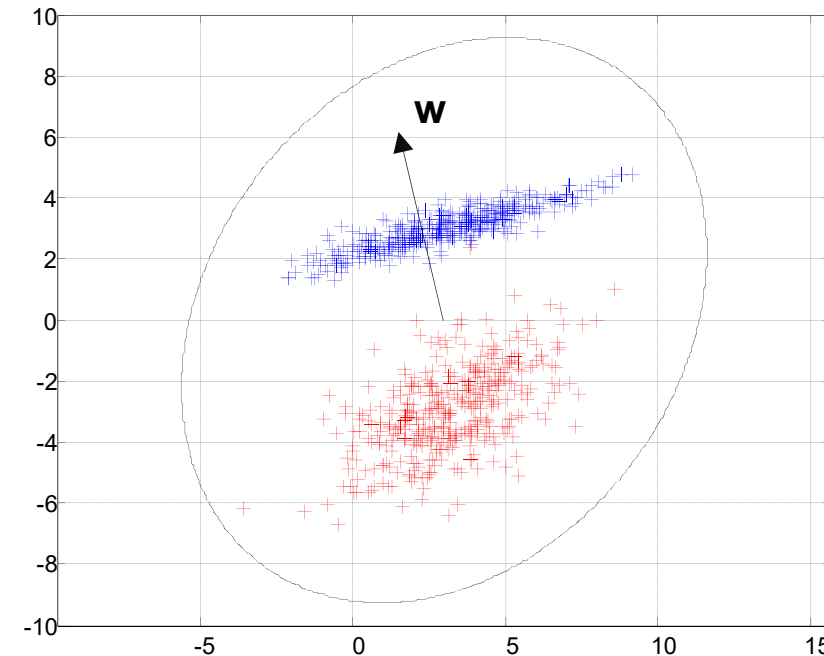
Transforming with the principal components is not always useful to discriminate among classes. It is desirable to define a transform that

- Increases the inter-class distance and
- Reduces the intra-class dispersion.

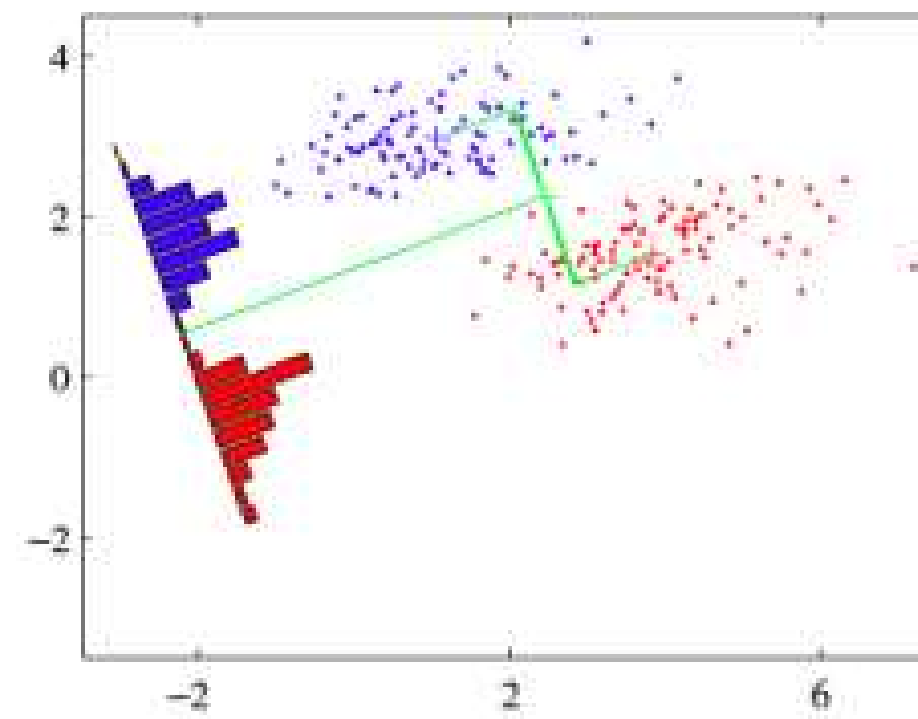
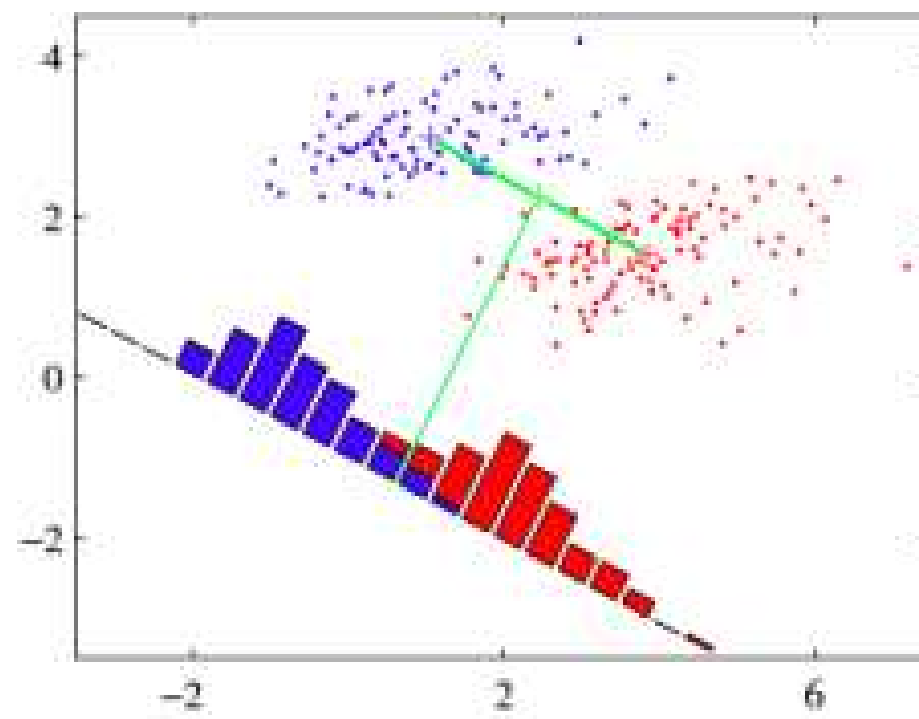
Example 6.



Bad separation with PCA when projecting onto w



Good separation when projecting onto w



Let us build a transform \mathbf{W}^T from a space of dimension d (size of observed vectors \mathbf{x}) to a space of lower dimension d' .

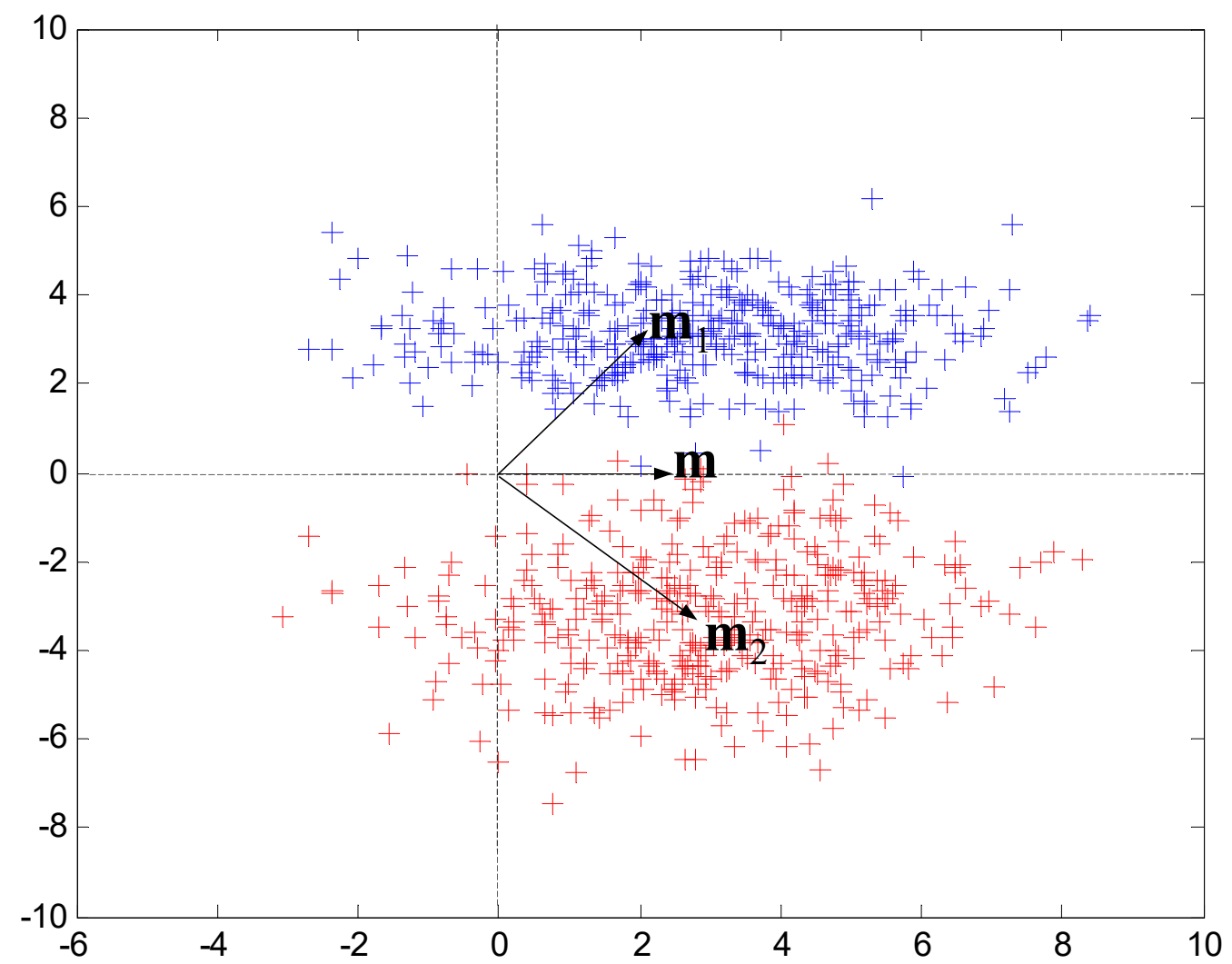
We need a measure of the **distance inter-classes** and a measure of the **intra-class dispersion**, to that end we define:

$$\mathbf{m}_i = \frac{1}{N_i} \sum_{\mathbf{x} \in D_i} \mathbf{x} \quad \text{Average of vectors of the class } i$$


$$\mathbf{m} = \frac{1}{N} \sum_{\mathbf{x} \in \{D_1, \dots, D_c\}} \mathbf{x} = \frac{1}{N} \sum_{i=1}^c N_i \mathbf{m}_i \quad \text{Average of all data}$$

$$\mathbf{S}_T = \sum_{\mathbf{x} \in \{D_1, \dots, D_c\}} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T \quad \text{Dispersion matrix of all data}$$

Unlike PCA, MDA is a supervised method: labels per feature vector are needed.



$$\begin{aligned}
\mathbf{S}_T &= \sum_{\mathbf{x} \in \{D_1, \dots, D_c\}} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T = \\
&= \sum_{i=1}^c \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i + \mathbf{m}_i - \mathbf{m})(\mathbf{x} - \mathbf{m}_i + \mathbf{m}_i - \mathbf{m})^T = \\
&= \sum_{i=1}^c \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T + \sum_{i=1}^c \sum_{\mathbf{x} \in D_i} (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T = \\
&= \sum_{i=1}^c \mathbf{S}_{C,i} + \sum_{i=1}^c N_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T = \mathbf{S}_C + \mathbf{S}_B
\end{aligned}$$



Sum of intra-class dispersion matrices
Inter-class dispersion matrix

The total covariance matrix (used in PCA) contains a term related to the magnitude to be maximized, and a term related to the magnitude to be minimized. This is why PCA does not always work!

The transform to be applied is:

$$\mathbf{z}_k = \mathbf{W}^T \mathbf{x}_k \quad \mathbf{x} \in \mathbb{R}^d, \mathbf{z} \in \mathbb{R}^{d'}, \mathbf{W}^T \in \mathbb{R}^{d' \times d} \quad d' < d$$

Intra-class and inter-class dispersion matrices for vectors \mathbf{z}_k are related to intra-class and inter-class dispersion matrices in \mathbf{x} as:

$$\mathbf{S}_C \rightarrow \mathbf{W}^T \mathbf{S}_C \mathbf{W}$$

$$\mathbf{S}_B \rightarrow \mathbf{W}^T \mathbf{S}_B \mathbf{W}$$

A scalar measure of the dispersion can be the volumen of ellipsoids. For a general matrix \mathbf{S} , the equation of the hiper-ellipsoid and its volume are given by:

$$(\mathbf{x} - \mathbf{m})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{m}) = K$$

$$V = V_d |\mathbf{S}|^{1/2} K^{d/2}$$

$$V_d = \begin{cases} \frac{1}{(d/2)!} \pi^{d/2} & d \text{ even} \\ \frac{2^d \left(\frac{d-1}{2}\right)!}{d!} \pi^{(d-1)/2} & d \text{ odd} \end{cases}$$

Therefore, assuming \mathbf{z} is Gaussian, the volumen associated to all classes is proportional to

$$|\mathbf{W}^T \mathbf{S}_C \mathbf{W}|^{1/2}$$

and the distance between classes is given by

$$|\mathbf{W}^T \mathbf{S}_B \mathbf{W}|^{1/2}$$

We can choose the definition of \mathbf{S}_B matrix in slide 29 or any other matrix that measures the inter-class distance

Discriminant criterion:

$$\mathbf{W} = \arg \max_{\mathbf{W}} \frac{|\mathbf{W}^T \mathbf{S}_B \mathbf{W}|}{|\mathbf{W}^T \mathbf{S}_C \mathbf{W}|}$$

The solution for $\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_{d'}]$ is the set of generalized eigenvectors:

$$\mathbf{S}_B \mathbf{w}_j = \sigma_j \mathbf{S}_C \mathbf{w}_j$$

associated to the d' largest eigenvalues. Gathering all equations into a single one:

$$\mathbf{S}_B \mathbf{W} = \mathbf{S}_C \mathbf{W} \Sigma \quad \Rightarrow \quad \mathbf{S}_C^{-1} \mathbf{S}_B \mathbf{W} = \mathbf{W} \Sigma$$

where Σ is a diagonal matrix that contains all eigenvalues.

Among all eigenvalues, which ones do we have to consider? If \mathbf{W} contains the eigenvectors, the discriminant is :

$$\frac{|\mathbf{W}^T \mathbf{S}_B \mathbf{W}|}{|\mathbf{W}^T \mathbf{S}_C \mathbf{W}|} = |\Sigma| = \prod_{i=1}^{d'} \sigma_i$$

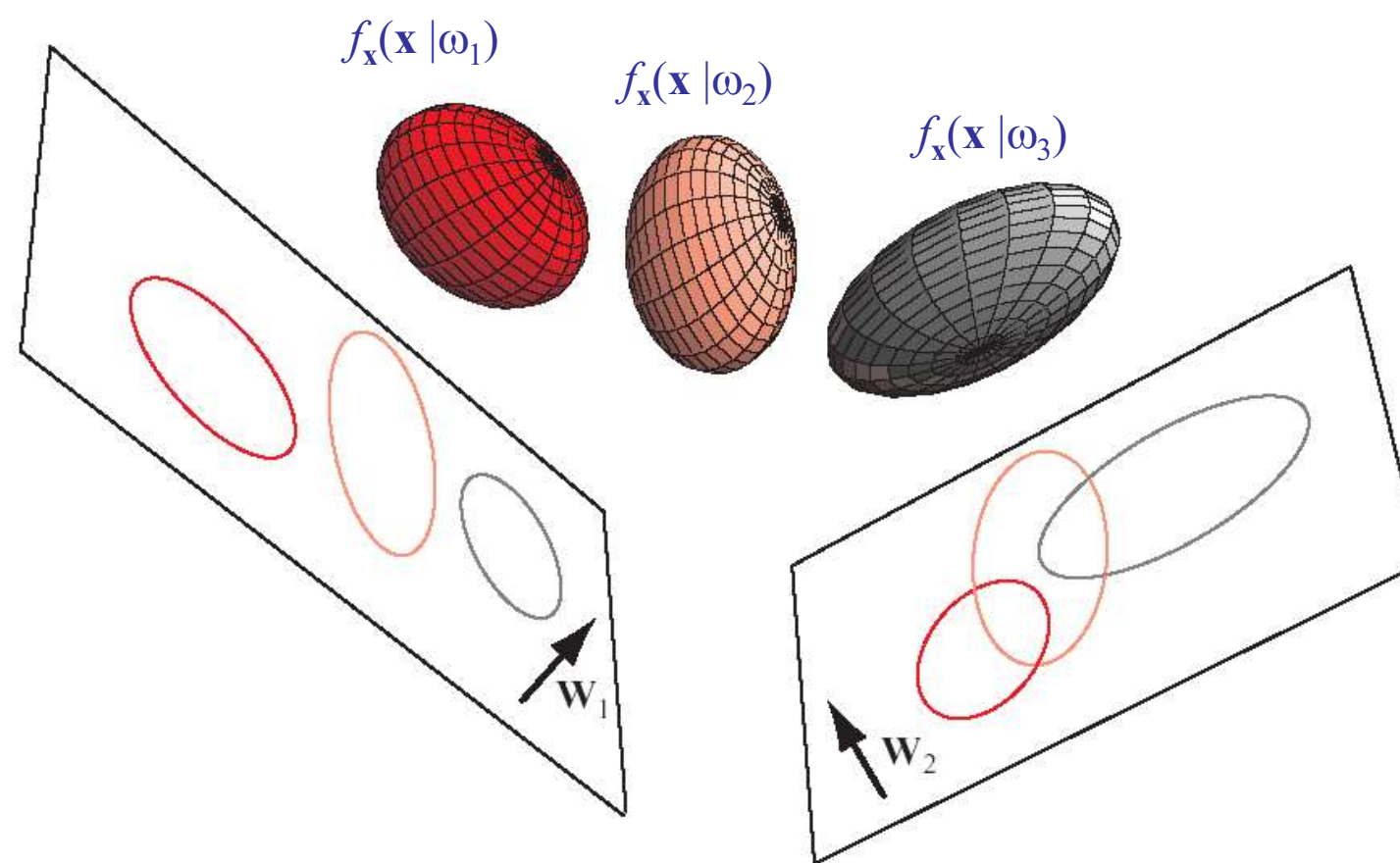
The maximum dimension of vectors \mathbf{z} is $d' \leq \min(d, c-1)$ because:

1. The discriminant above implies that only non-zero eigenvalues have to be considered (otherwise we have no discrimination at all).
2. The number of non-zero eigenvalues is upper bounded by

$$\text{rank}(\mathbf{S}_C^{-1} \mathbf{S}_B) \leq \min(\text{rank}(\mathbf{S}_C^{-1}), \text{rank}(\mathbf{S}_B))$$
3. $\text{rank}(\mathbf{S}_B) \leq \min(d, c-1)$ and $\text{rank}(\mathbf{S}_C) = d$, so $\min(d, c-1)$ eigenvalues are non-zero (to be checked in the assignment).

Should we be interested in keeping more than $\min(d, c - 1)$ features in \mathbf{z} , we could define groups of features in \mathbf{x} and compute one MDA matrix on each one: e.g. with the data base PHONEME in the lab.

Example 7.



Three-dimensional distributions associated to 3 classes are projected onto planes. The projection on the plane defined by \mathbf{W}_1 provides more separation between classes than the plane defined by \mathbf{W}_2 .

Decision boundaries on the transformed vectors

The computation of the boundaries can be done following the guidelines of a Bayesian detector for the Gaussian case (see chapter #2) on vectors \mathbf{z} , where means and covariance matrices are related to means and covariance matrices in \mathbf{x} :

$$\boldsymbol{\mu}_i = \mathbf{W}^T \mathbf{m}_i \quad i = 1, \dots, c$$

$$\mathbf{C}_i = \frac{1}{N_i} \mathbf{W}^T \mathbf{S}_{C,i} \mathbf{W}$$

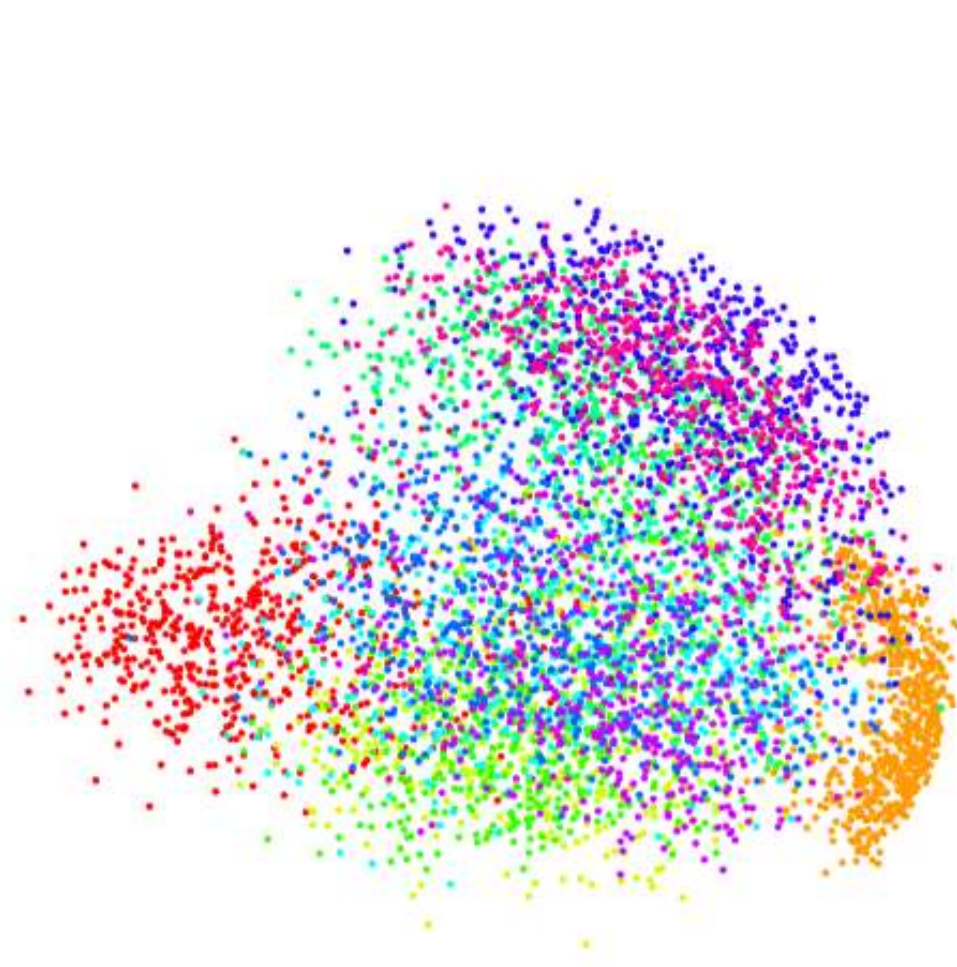
3.4 NON-LINEAR APPROACHES

Many other non-linear feature reduction methods have been proposed in the literature, like:

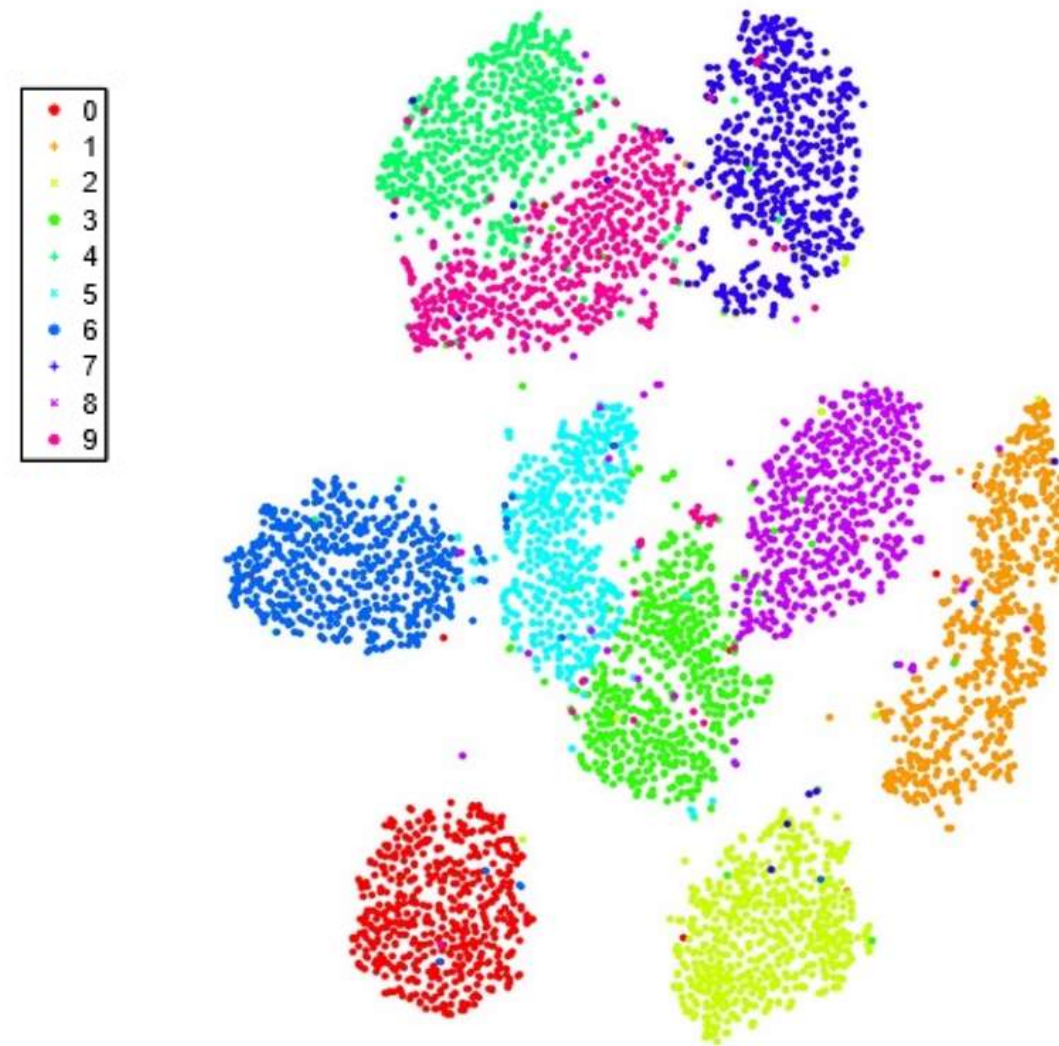
- t-SNE
- autoencoders

- t-SNE: reduction of dimensionality by preserving distance among vectors

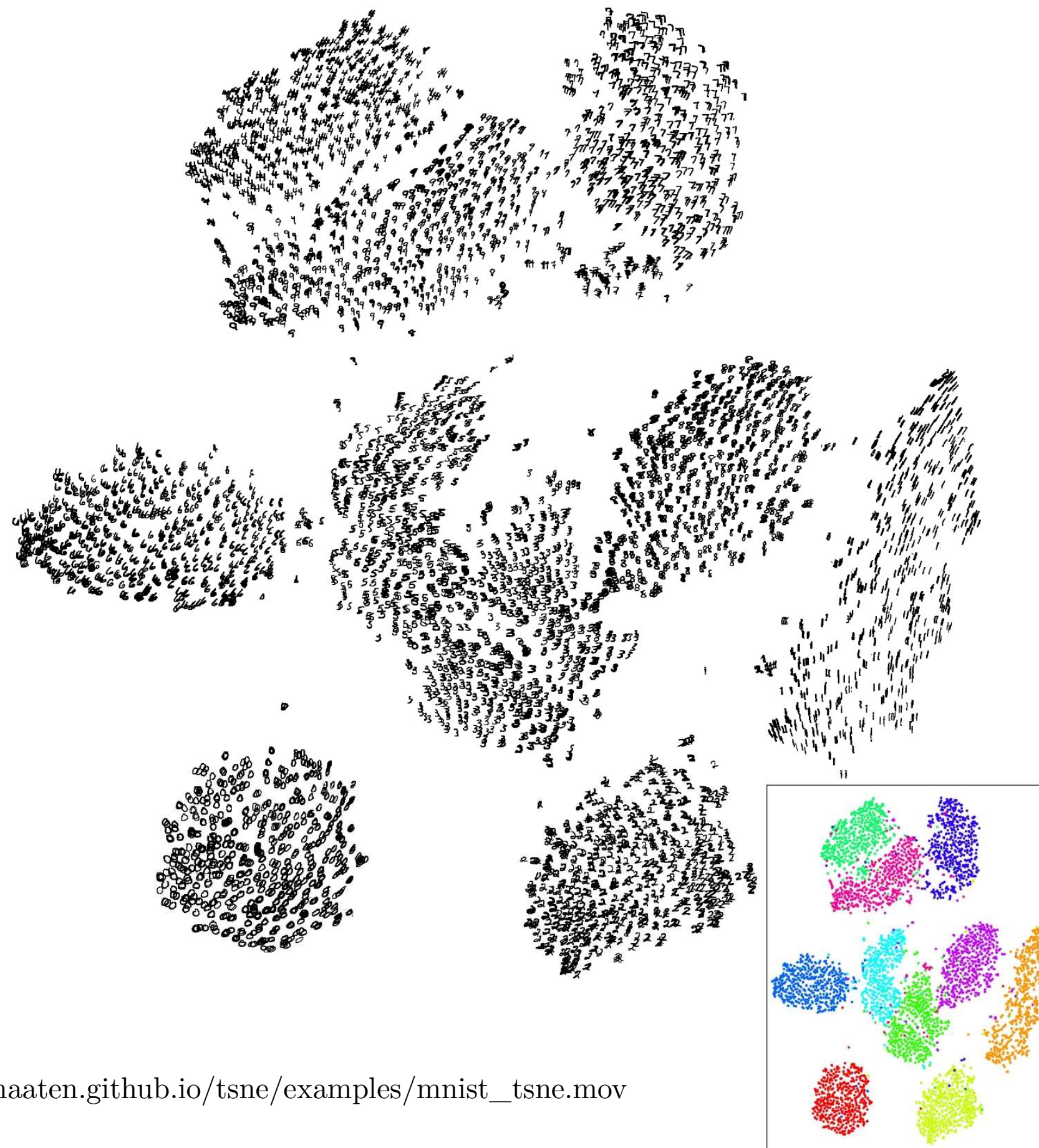
Example 8. Data base NIST of 6.000 handwritten digits.



Visualization with PCA

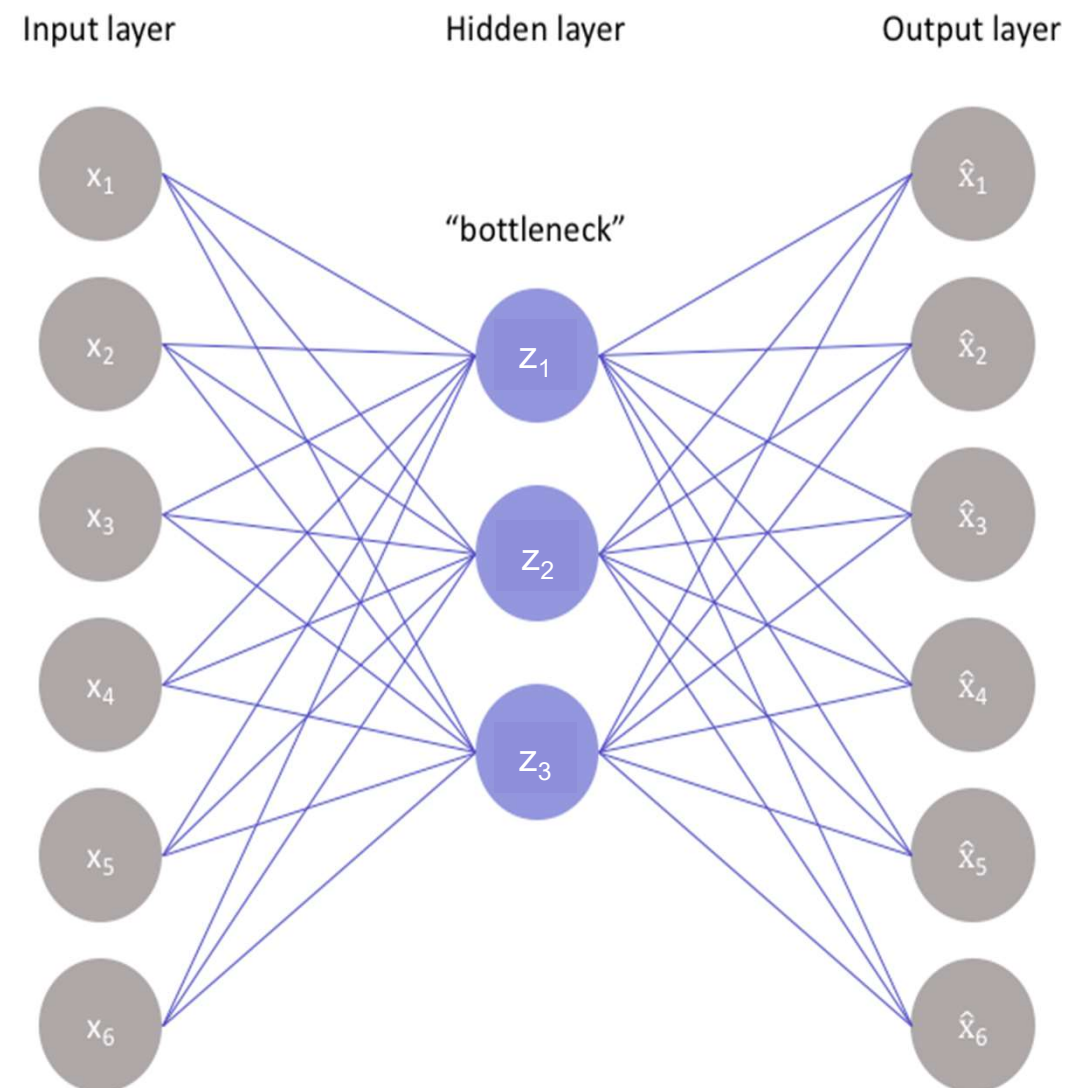


Visualization with t-SNE

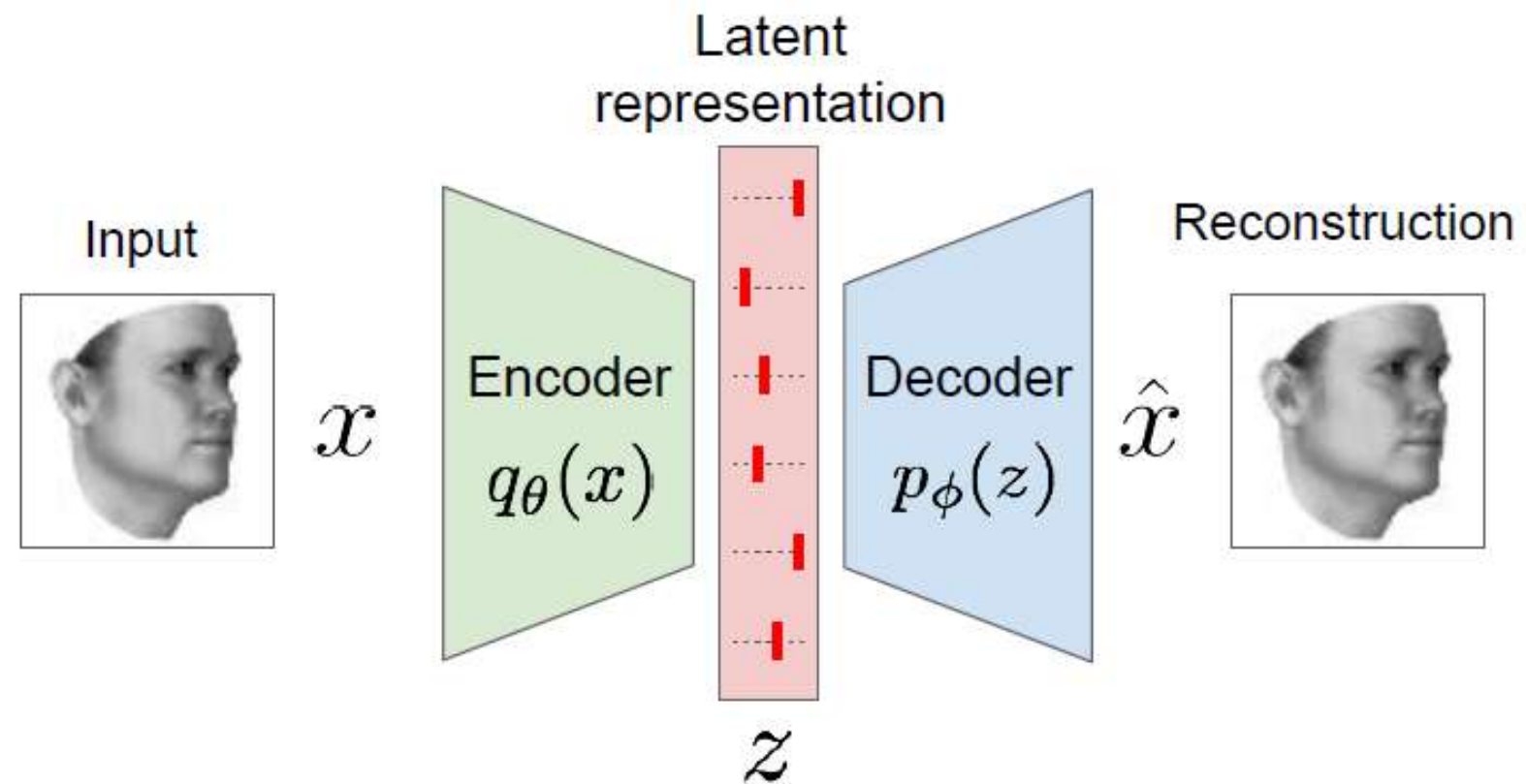


https://lvdmaaten.github.io/tsne/examples/mnist_tsne.mov

- Autoencoder based on a NN



- Autoencoder based on a NN



3.5 CONCLUSIONS

- PCA is an unsupervised method: it does not requires labels y_k to reduce the number of features.
- PCA does not guarantee class separation, even though it is widely ised in many applications.
- MDA is a supervised method, and can separate classes with some limitations on the number of features. This limitation can be avoided by applying MDA on subsets of features.
- Many other non-linear feature reduction methods have been proposed in the literature:
 - t-SNE
 - autoencoders