Feature Selection

Scatter matrices, Fisher's discriminant, Principal Component Analysis.

Feature extraction – Structural

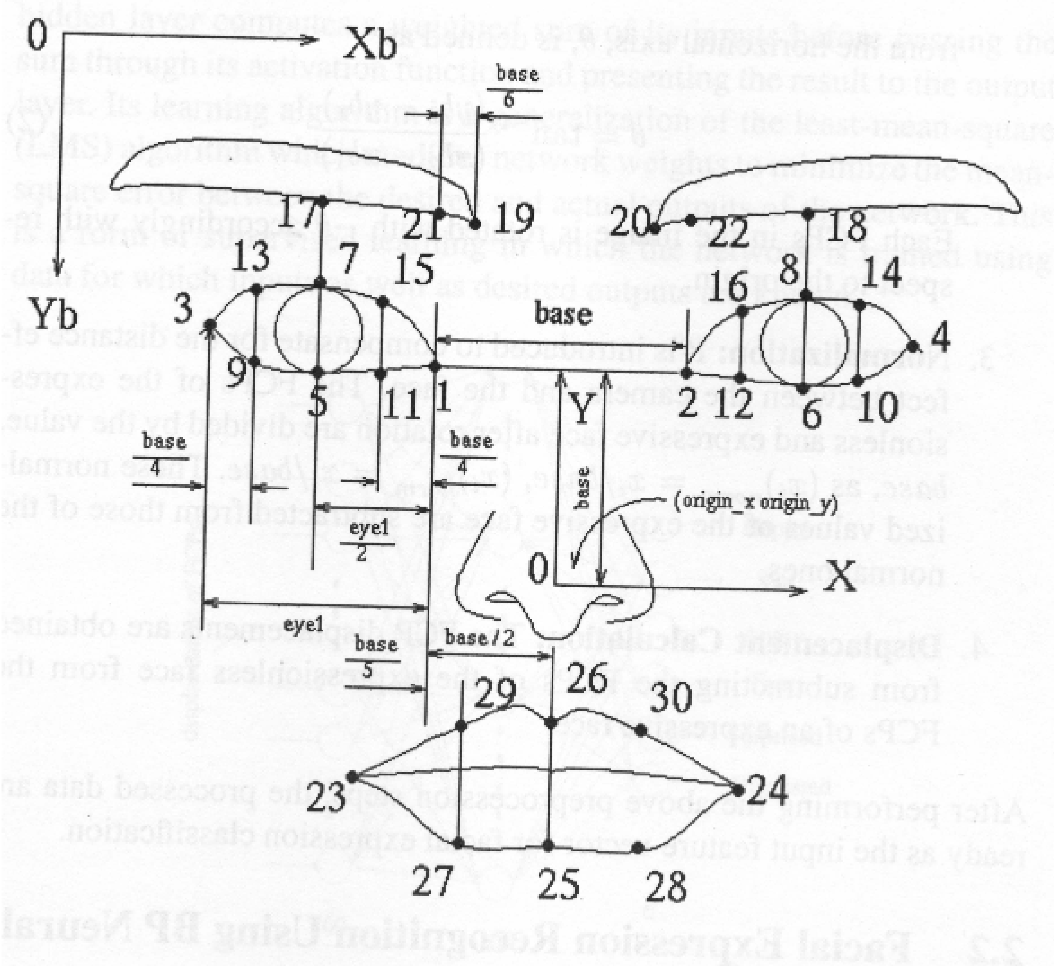


Figure 1. The 30 facial characteristic points.

Feature extraction – Structural

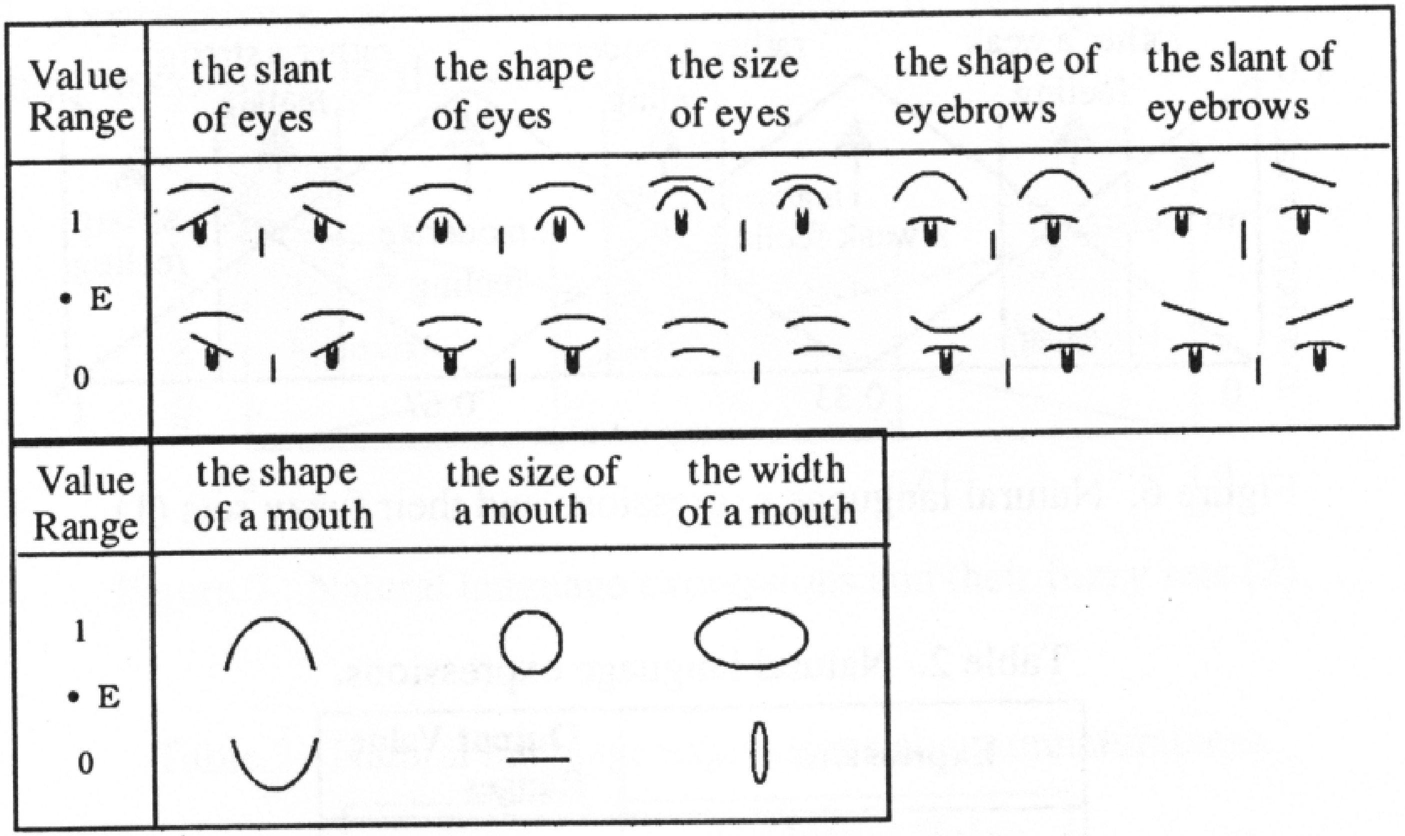


Figure 4. Parameter value range according to location of each feature in a face.

Feature extraction – Texture

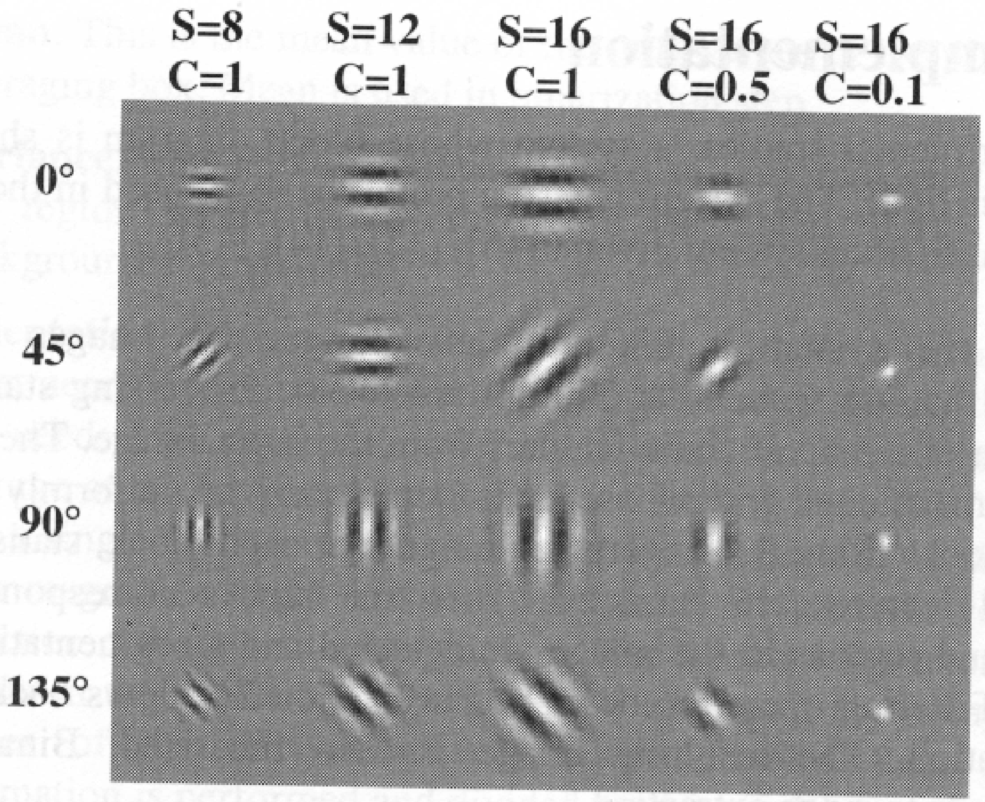


Figure 4. Ridge extraction filters at different orientations, ridge spacing and orientation certainty levels. Up to the middle column of the table the ridge spacing S is increased which means a decrease in frequency. After the middle column the frequency is kept constant but certainty C is decreased which means that bandwidth of the filter is increased.

Feature extraction – Texture

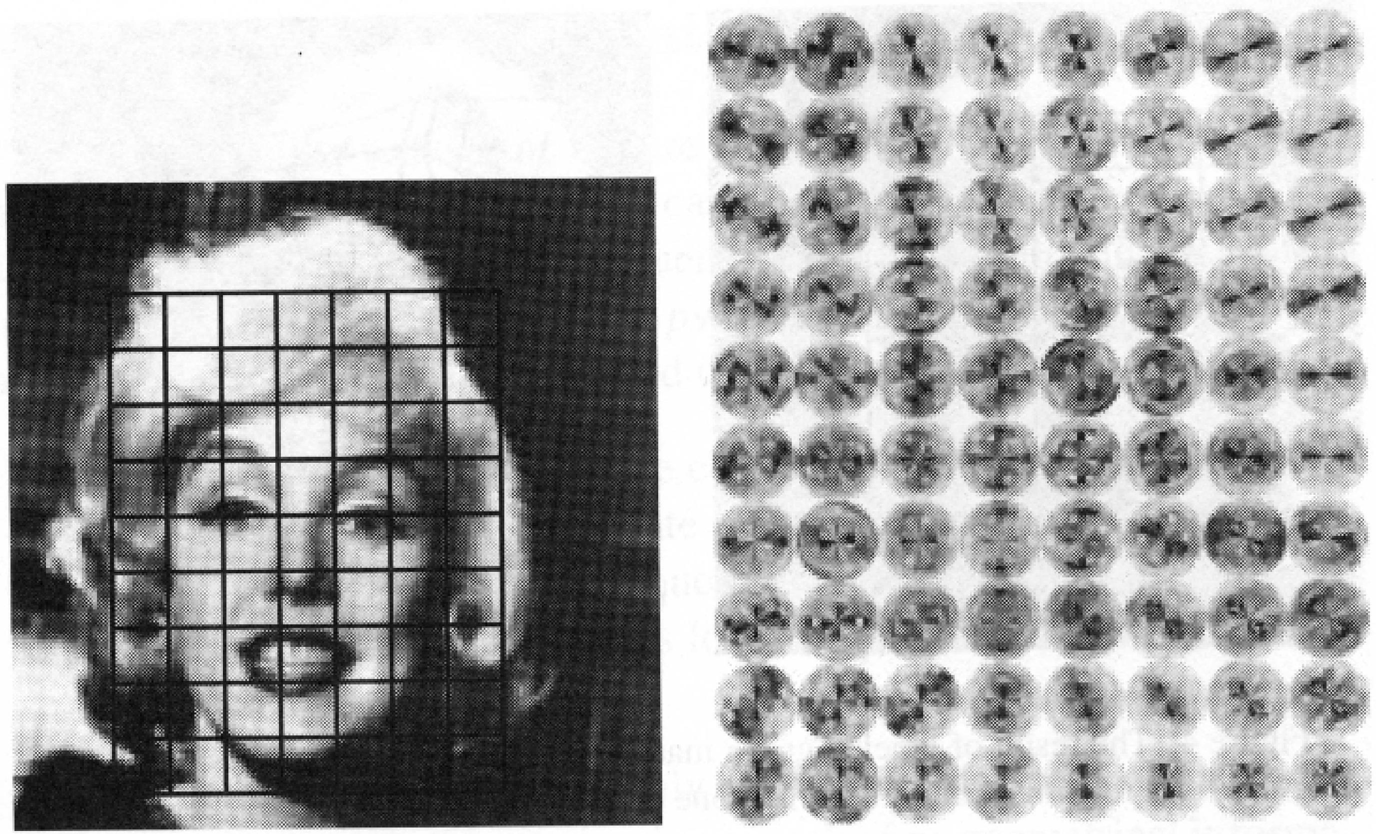


Figure 3. Visualization of Gabor features with frequency-independent sampling. Each point in the grid is assigned a little frequency space, which is visualized on the right-hand side. The gray values of the little segments show the moduli of the Gabor responses as a function of the (two-dimensional) spatial frequency.

Feature extraction – Texture

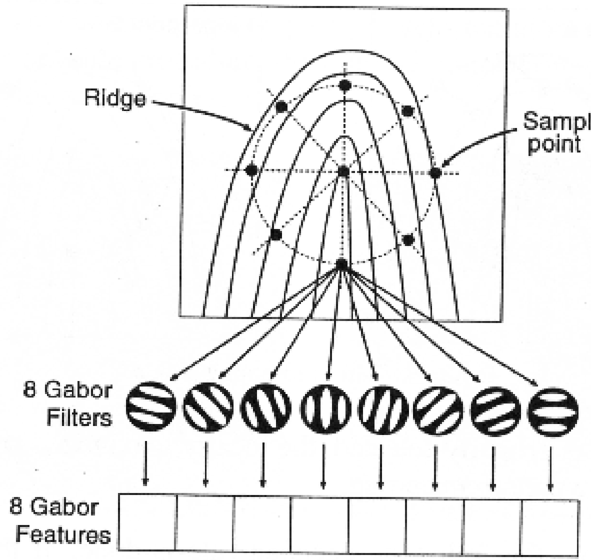


Figure 4. Extracted Gabor features.

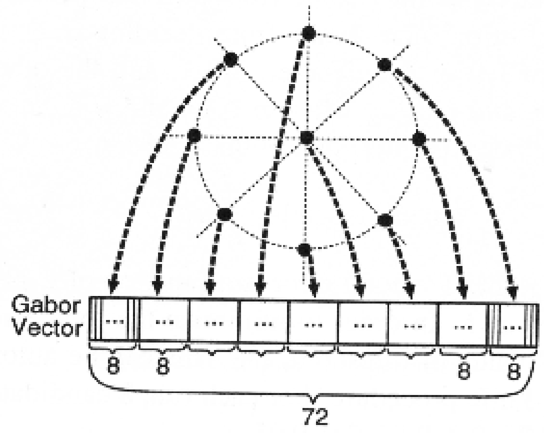
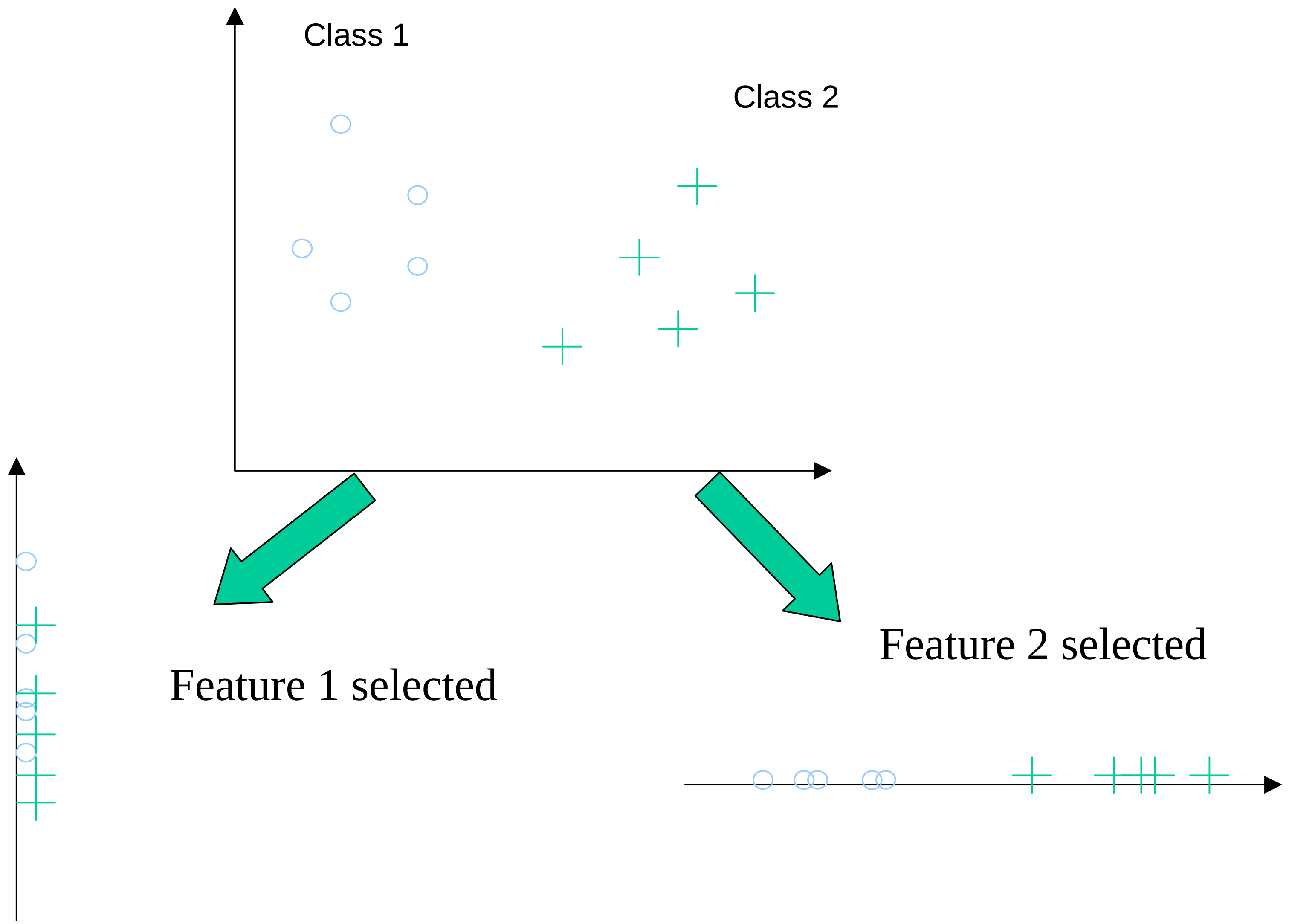


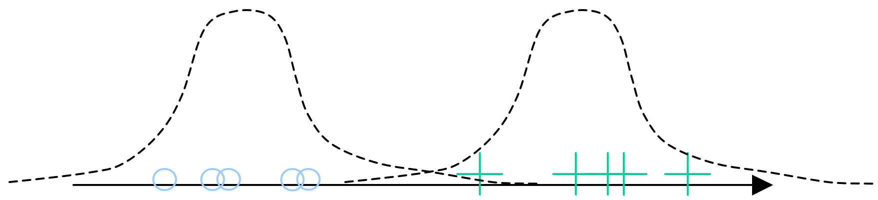
Figure 5. Gabor vector.

What features are good?



What features are good?

Good features have low intraclass variation and large interclass variation



Selection criteria:

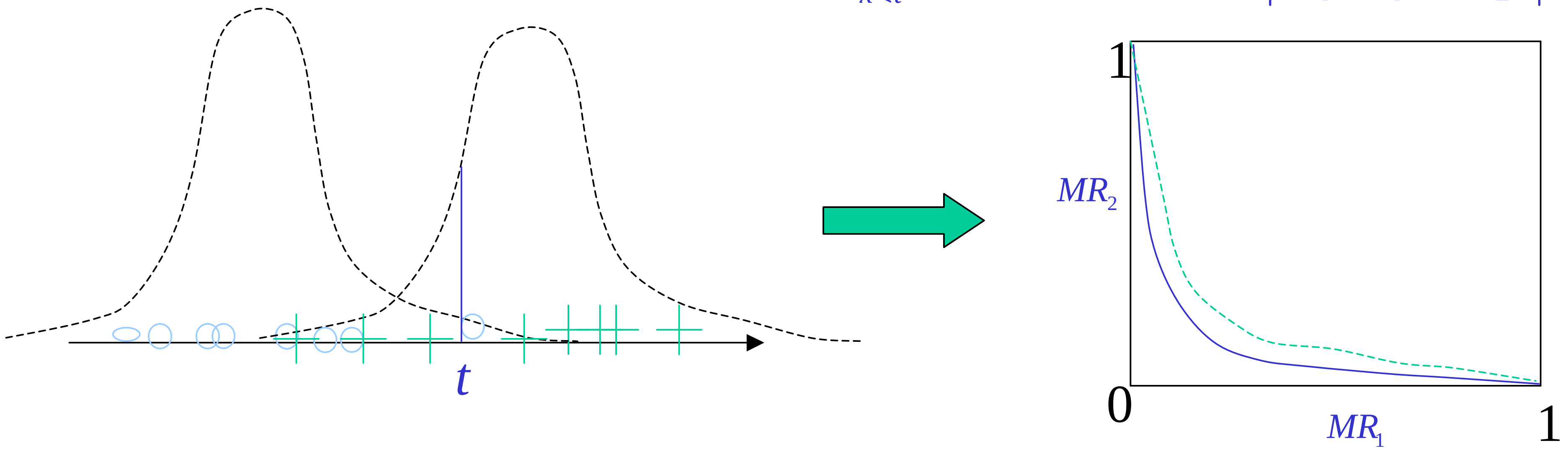
- Hypothesis testing, e.g. the hypothesis that distributions of each class have different means
- ROC curves
- Class separability measures

ROC curve for feature selection

We can construct ROC curves and somehow estimate the utility for every feature:

$$MR_1(t) = \int_{x>t} p(x \mid \omega_1)dx \approx \frac{\left|\{x_i \mid x_i > t, x_i \in \omega_1\}\right|}{\left|\{x_i \mid x_i \in \omega_1\}\right|}$$

$$MR_2(t) = \int_{x<t} p(x \mid \omega_2)dx \approx \frac{\left|\{x_i \mid x_i < t, x_i \in \omega_2\}\right|}{\left|\{x_i \mid x_i \in \omega_2\}\right|}$$



Scatter matrices

Covariance matrix for class i :

$$\mathbf{S}_i = E[(\mathbf{x}-\boldsymbol{\mu}_i)(\mathbf{x}-\boldsymbol{\mu}_i)^T]$$

Within-class scatter matrix: $$\mathbf{S}_w = \sum_{i=1}^{M} P_i \, \mathbf{S}_i$$

Between-class scatter matrix: $$\mathbf{S}_b = \sum_{i=1}^{M} P_i \, (\boldsymbol{\mu}_i-\boldsymbol{\mu}_0)(\boldsymbol{\mu}_i-\boldsymbol{\mu}_0)^T$$

where $$\boldsymbol{\mu}_0 = \sum_{i=1}^{M} P_i \, \boldsymbol{\mu}_i$$

Mixture scatter matrix: $$\mathbf{S}_m = E[(\mathbf{x}-\boldsymbol{\mu}_0)(\mathbf{x}-\boldsymbol{\mu}_0)^T]$$

$$\mathbf{S}_m = \mathbf{S}_b + \mathbf{S}_w$$

Criteria for feature selection using scatter matrices (1)

Small $trace(\mathbf{S}_w)$ or $\det(\mathbf{S}_w)$ $\Rightarrow$ small intraclass variation

Large $trace(\mathbf{S}_b)$ or $\det(\mathbf{S}_b)$ $\Rightarrow$ large interclass variation

Large $trace(\mathbf{S}_m)$ or $\det(\mathbf{S}_m)$ $\Rightarrow$ large interclass variation

Criteria for feature selection using scatter matrices (2)

- Maximize $$J_1 = \frac{trace(\mathbf{S}_m)}{trace(\mathbf{S}_w)}$$
- Maximize $$J_2 = \frac{\det(\mathbf{S}_m)}{\det(\mathbf{S}_w)} = \det(\mathbf{S}_w^{-1}\mathbf{S}_m)$$
- Maximize $$J_3 = trace(\mathbf{S}_w^{-1}\mathbf{S}_m)$$

Criteria for feature selection using scatter matrices (3)

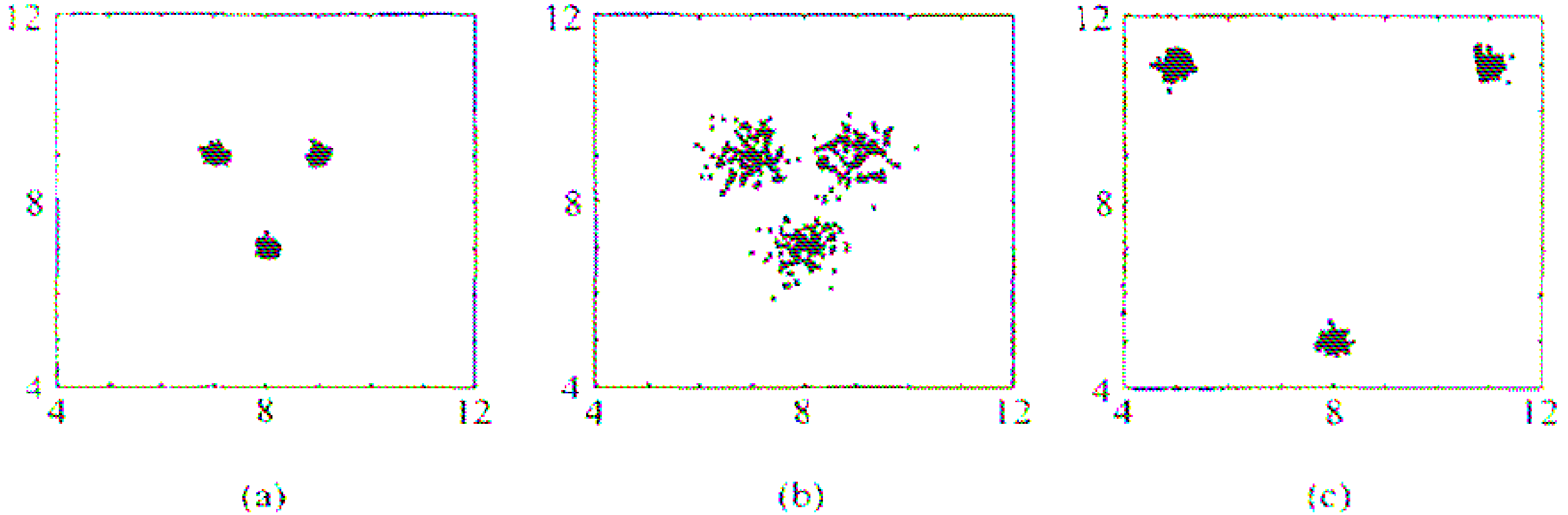


FIGURE 5.4: Classes with (a) small within-class variance and small between-class distances, (b) large within-class variance and small between-class distances and (c) small within-class variance and large between-class distances.

$J_3 =$ (a) 164.7 (b) 12.5 (c) 620.9

(Theodoridis, Koutroumbas – “Pattern Recognition”)

Fisher's discriminant criterion

In case of two classes and selection of one feature

$$\det(\mathbf{S}_w) \sim \sigma_1^2 + \sigma_2^2$$

$$\det(\mathbf{S}_b) \sim (\mu_1 - \mu_2)^2$$

Fisher's criterion is to maximize

$$J = \frac{\det(\mathbf{S}_b)}{\det(\mathbf{S}_w)} = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

Extension to multi-class problems:

$$J = \sum_{i=1}^{M} \sum_{j \neq i}^{M} \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2}$$

Feature Subset Selection

Using criteria for good features, we can explore intuitive algorithms for selecting best l features out of m original:

- Apply some criterion to each feature; select l features with best one-dimensional criterion's performance

  - *easy to implement*
  - *but features might be correlated: modify algorithm to only select features not strongly correlated with already selected*

- Try all different combinations of l features out of m; for each combination use l-dimensional criterion's performance

  - *computationally expensive*

Feature Selection by Linear Transform

Instead of selecting original features $x_1, \ldots, x_m$, we can select linear combinations of them:

$$y_j = a_j^1 x_1 + \ldots + a_j^m x_m, \quad j = 1, \ldots, l$$

In matrix form: $\mathbf{y} = \mathbf{A}^T \mathbf{x}$

$\mathbf{S}_{yw} = \mathbf{A}^T \mathbf{S}_{xw} \mathbf{A}$ - within-class scatter matrix for $\mathbf{y}$

$\mathbf{S}_{yb} = \mathbf{A}^T \mathbf{S}_{xb} \mathbf{A}$ - between-class scatter matrix for $\mathbf{y}$

Want to find $\mathbf{A}$ so that feature selection criterion function

$$J_3 = trace\,(\mathbf{S}_{yw}^{-1} \mathbf{S}_{yb}) = trace\,((\mathbf{A}^T \mathbf{S}_{xw} \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{S}_{xb} \mathbf{A}))$$

is maximized.

Feature Selection by Linear Transform

Solution: $\mathbf{A}$ is the projection onto subspace spanned by the l eigenvectors corresponding to l largest eigenvalues of $\mathbf{S}_{xw}^{-1}\mathbf{S}_{xb}$

$$J_3 = trace\,(\mathbf{S}_{yw}^{-1}\mathbf{S}_{yb}) = \lambda_1 + \ldots + \lambda_l$$

$\lambda_1 \geq \ldots \geq \lambda_m$ - eigenvalues of $\mathbf{S}_{xw}^{-1}\mathbf{S}_{xb}$

For two-class problems: $\mathbf{S}_{xb}$ has rank 1 and

$\mathbf{y} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \mathbf{S}_{xw}^{-1}\mathbf{x}$ is an optimal projection

- Fisher's linear discriminant

Principal Component Analysis (1)

Criteria for feature selection:

transformed features should be uncorrelated:

$$E[y_i y_j] = 0, \quad i \neq j$$

(assume $E[y_i] = 0$ for all i)

In matrix form: $\mathbf{y} = \mathbf{A}^T \mathbf{x}$

$$\mathbf{R}_y = E[\mathbf{y}\mathbf{y}^T] = E[\mathbf{A}^T \mathbf{x}\mathbf{x}^T \mathbf{A}] = \mathbf{A}^T \mathbf{R}_x \mathbf{A}$$

Need to find $\mathbf{A}$ so that $\mathbf{R}_y$ is diagonal.

Principal Component Analysis (2)

Solution:

$$\mathbf{R}_x = E[\mathbf{x}\mathbf{x}^T]$$ is symmetric

Hence it has a set of orthonormal eigenvectors $\mathbf{a}_i$:

$$\mathbf{R}_x \mathbf{a}_i = \lambda_i \mathbf{a}_i$$

If $\mathbf{A} = [\mathbf{a}_1 \ldots \mathbf{a}_m]$, then $\mathbf{R}_y = \mathbf{A}^T \mathbf{R}_x \mathbf{A} = \mathbf{\Lambda}$

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & & 0 \\ 0 & \lambda_2 & & \\ & & & \\ & & & \\ 0 & & & \lambda_m \end{pmatrix}, \quad \lambda_1 \geq \ldots \geq \lambda_m \geq 0$$

Principal Component Analysis (3)

$\mathbf{A}$ - Karhunen-Loeve transform

If $\mathbf{x} = \sum_{i=1}^{m} y_i \mathbf{a}_i$ then

$$\hat{\mathbf{x}} = \sum_{i=1}^{l} y_i \mathbf{a}_i$$

is the projection on subspace spanned by eigenvectors with largest eigenvalues

Can prove:

$$E[(\mathbf{x} - \hat{\mathbf{x}})^2] = \sum_{i=l+1}^{m} \lambda_i$$

So the approximation $\mathbf{x}$ by $\hat{\mathbf{x}}$ produces smallest mean square error.

Principal Component Analysis (4)

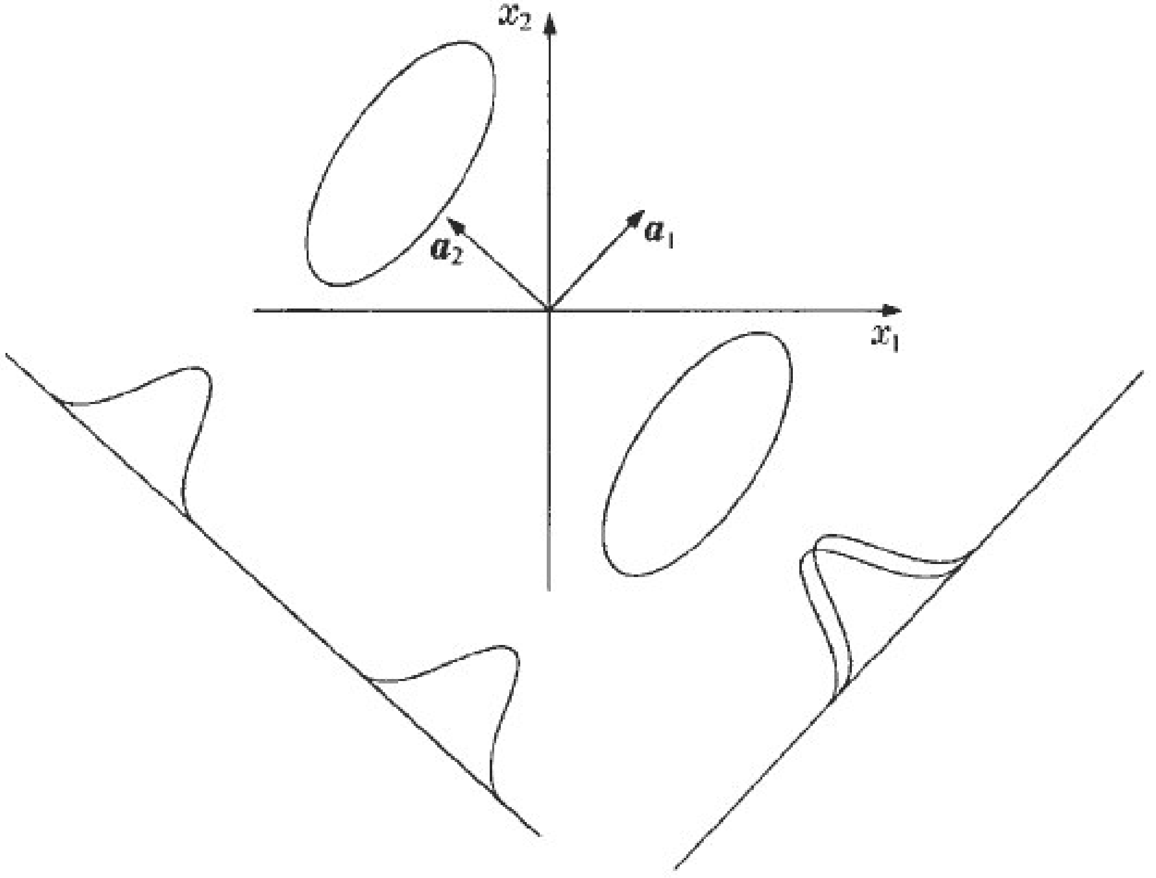


FIGURE 6.1: The KL transform is not always best for pattern recognition. In this example, projection on the eigenvector with the larger eigenvalue makes the two classes coincide. On the other hand, projection on the other eigenvector keeps the classes separated.

PCA for biometric data

• Biometric applications usually do not have enough data to learn intraclass variation.

• PCA can be well learned using interclass variations.

• Particularly, face recognition by PCA seems to deliver good results.

PCA for minutia extraction

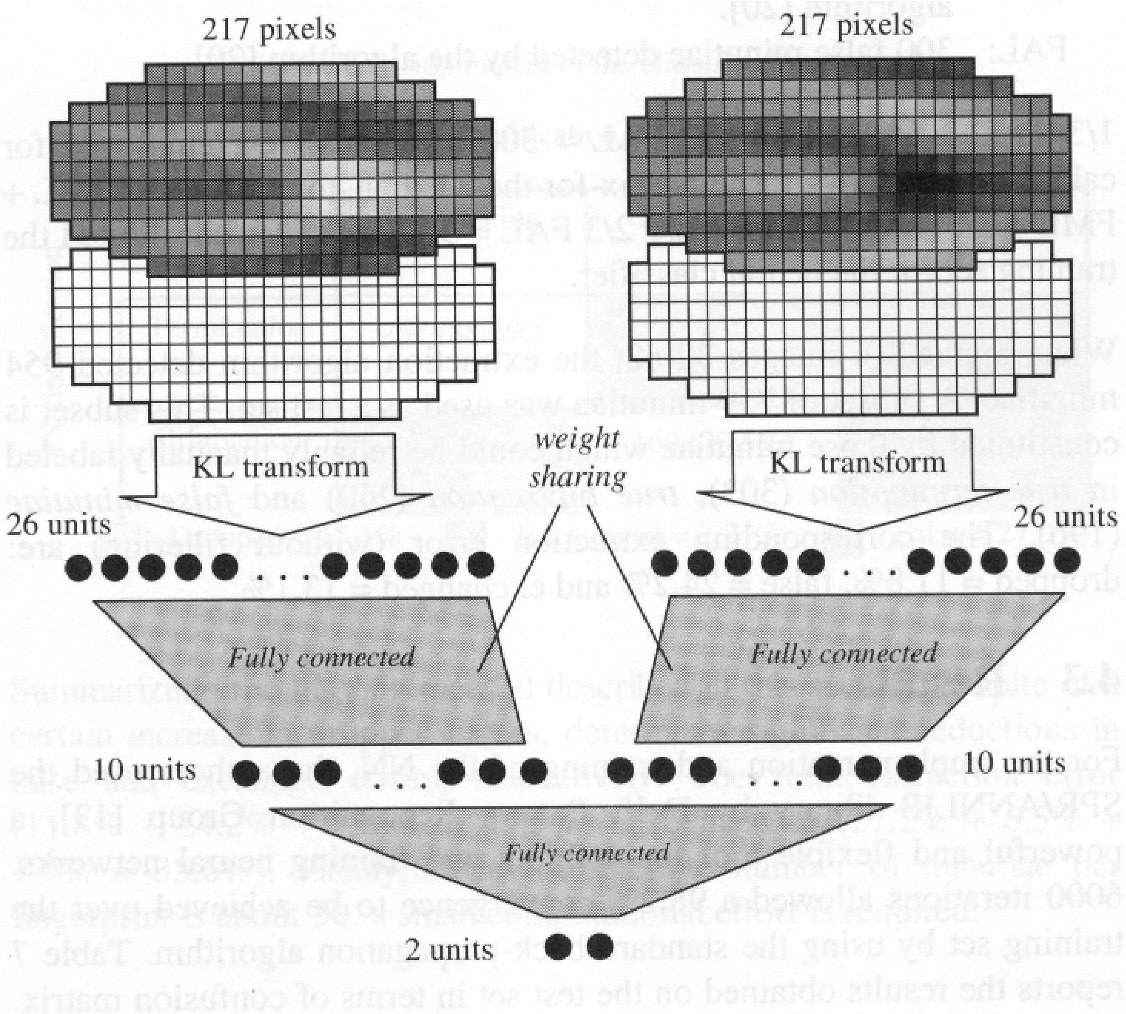
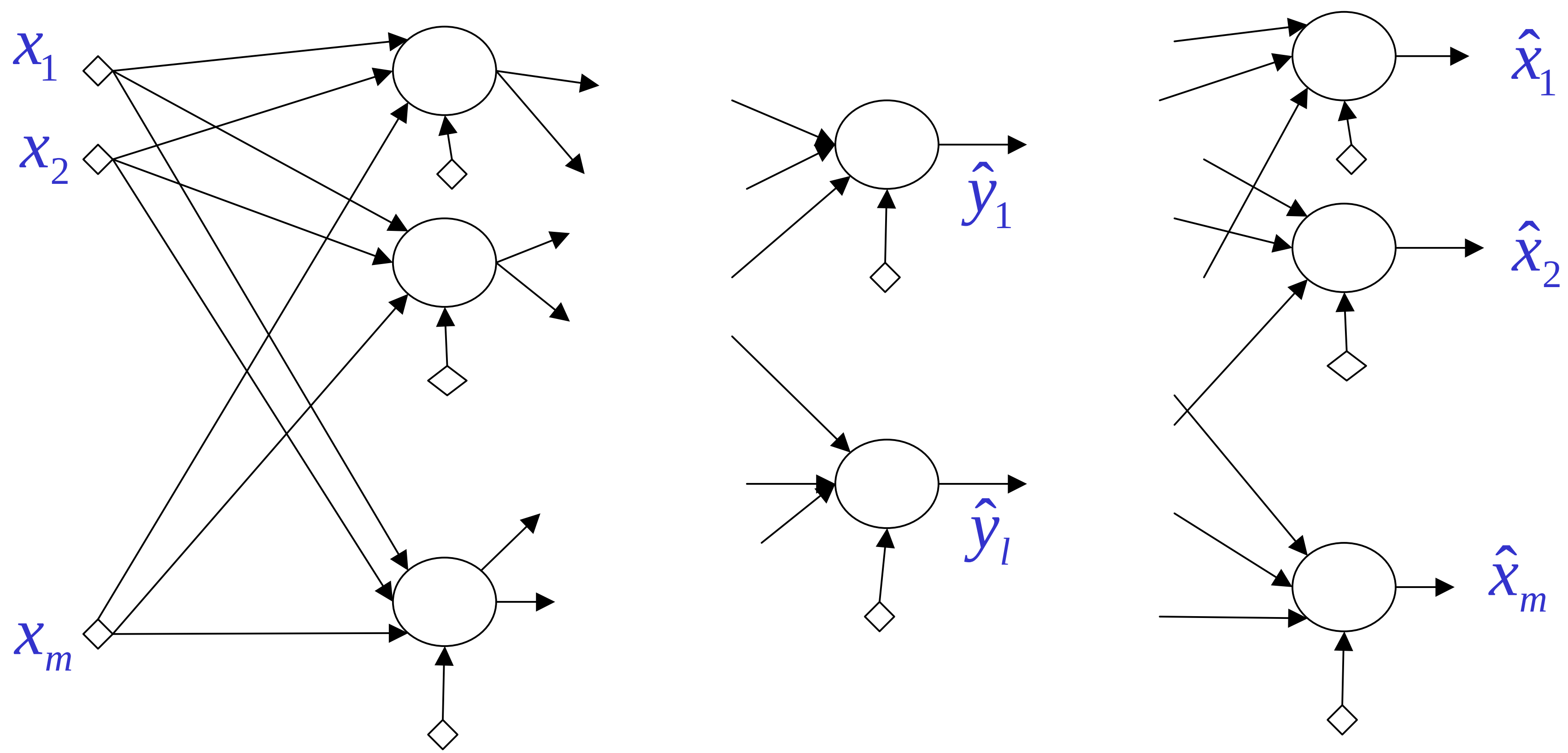


Figure 21. The Neural Network classifier architecture.

Maximizing information content of the features

- Fourier coefficients
- Moments
- DCT (discrete cosine transform)
- Quantization
- Neural networks

Neural Networks for Feature Selection

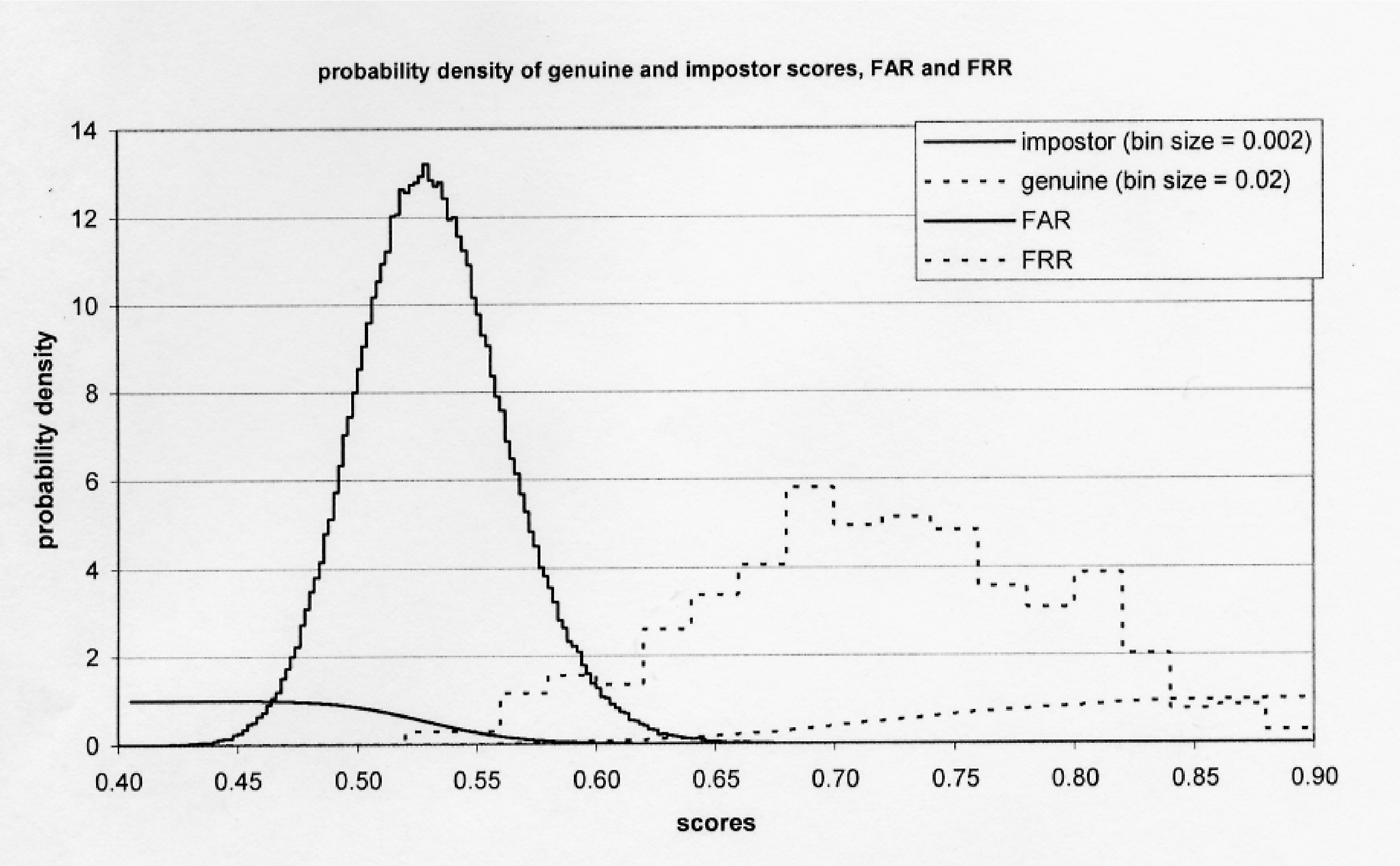


Some middle layer has $l < m$ neurons, whose values $y_1, \ldots, y_l$ are selected as features instead of $x_1, \ldots, x_m$

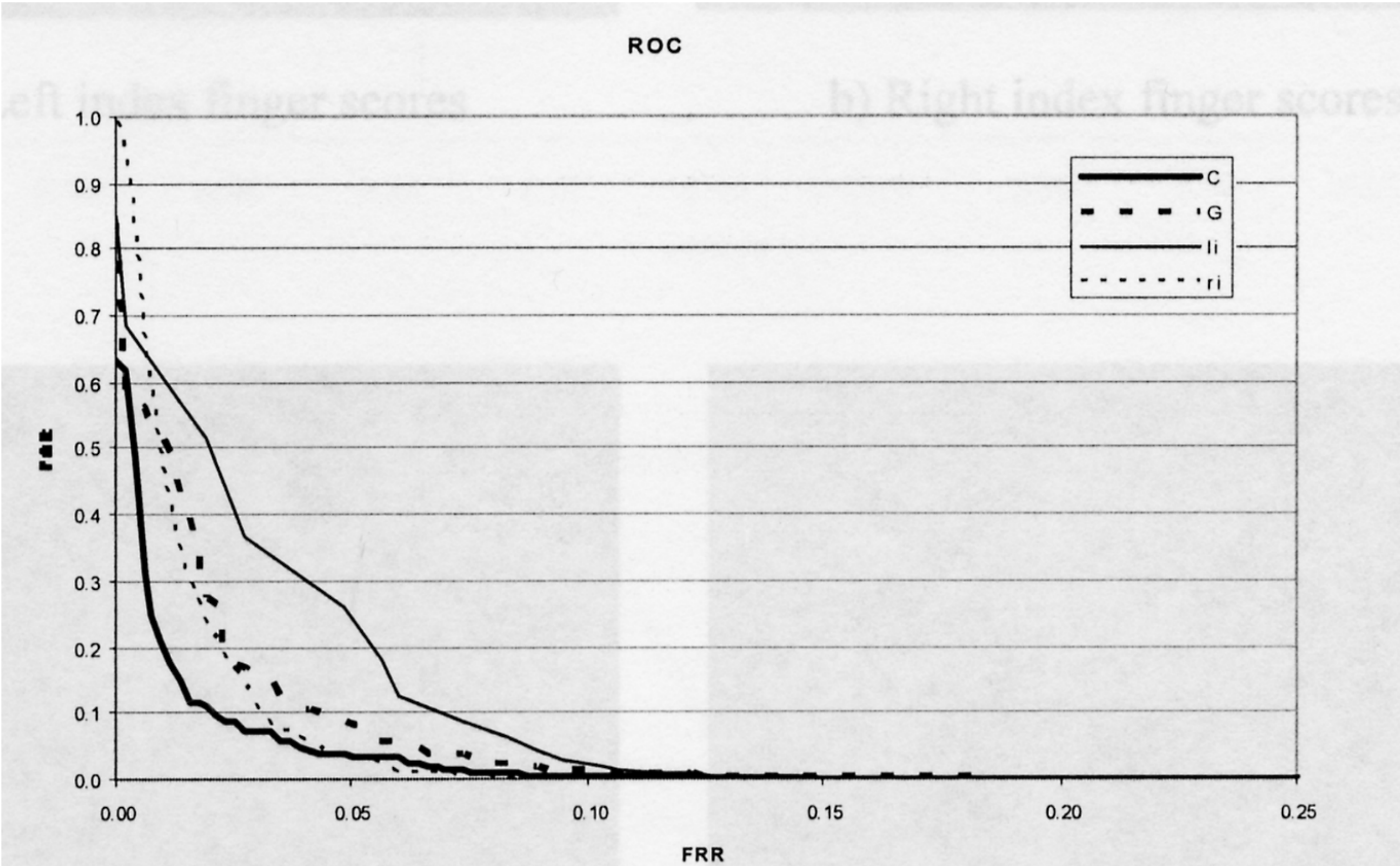
Project 1 discussion

- PDF estimation – need proper smoothing parameters (bin size in histogram method, kernel window in kernel methods)
- Draw ROC curves on the same graph in order to compare the performance of matchers
- Make sure same t is used for constructing ROC from FAR and FRR data

Sample pdf and FAR/FRR graph



Sample ROC graph



Face G matcher