Non-Linear Discriminant Analysis

Gaston BAUDAT, Fatiha ANOUAR

MEI, Mars Electronics International

1301 Wilson Drive, West Chester, PA 19380, USA

Email: gaston.baudat@eu.effem.com Phone: + 1 610 430 27 51

Email: fatiha.anouar@effem.com Phone: + 1 610 430 25 22

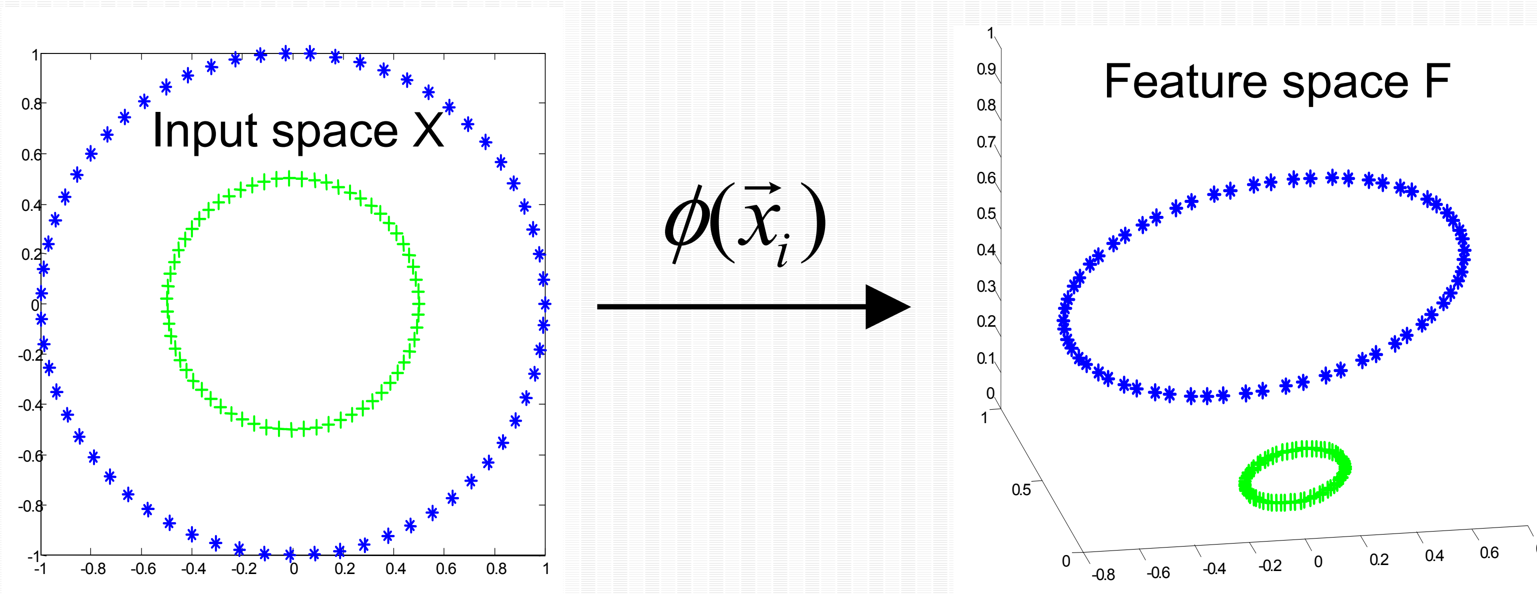
Fax: + 1 610 430 27 95

Presentation plan

- Kernel Trick
- Linear Discriminant Analysis (LDA).
- Generalized Discriminant Analysis (GDA).
- Feature Vector Selection (FVS).
- Sparse GDA using the FVS approach.
- Conclusions.
- Some references.

Kernel trick & dot product

- Let $\phi(\vec{x}_i)$ be an operator which maps data from an input space X into a feature space F:



Kernel trick & dot product
Ex: Polynomial mapping

- As an example assume the following mapping (2D -> 3D):

$$\phi(\vec{x}) = \begin{pmatrix} \varphi_{x,1} \\ \varphi_{x,2} \\ \varphi_{x,3} \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} \cdot x_1 \cdot x_2 \end{pmatrix} \qquad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Kernel trick & dot product
Ex: Dot product in F

- Explicit dot product in F:

$$\phi^T(\vec{x}) \cdot \phi(\vec{y}) = \varphi_{x,1}\varphi_{y,1} + \varphi_{x,2}\varphi_{y,2} + \varphi_{x,3}\varphi_{y,3}$$

- Implicit dot product in F:

$$\Rightarrow x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 x_2 y_2 = (x_1 y_1 + x_2 y_2)^2$$

$$\phi^T(\vec{x}) \cdot \phi(\vec{y}) = k(\vec{x}, \vec{y}) = (\vec{x}^T \cdot \vec{y})^2$$

Kernel trick & dot product
Kernel function

- The implicit form of the dot product in F uses a kernel function $k(\vec{x}, \vec{y})$.

- $k(\vec{x}, \vec{y})$ does not need the evaluation (or knowledge) of $\phi(\vec{x})$ nor $\phi(\vec{y})$.

- Any algorithm using only dot products can be expressed implicitly in F.

Kernel trick & dot product some classical kernels

- Gaussian: $(N_F = \infty)$

$$k(\vec{x},\vec{y}) = \exp\left(-\frac{\|\vec{x}-\vec{y}\|^2}{\sigma^2}\right)$$

- Sigmoid: $(N_F = \infty)$

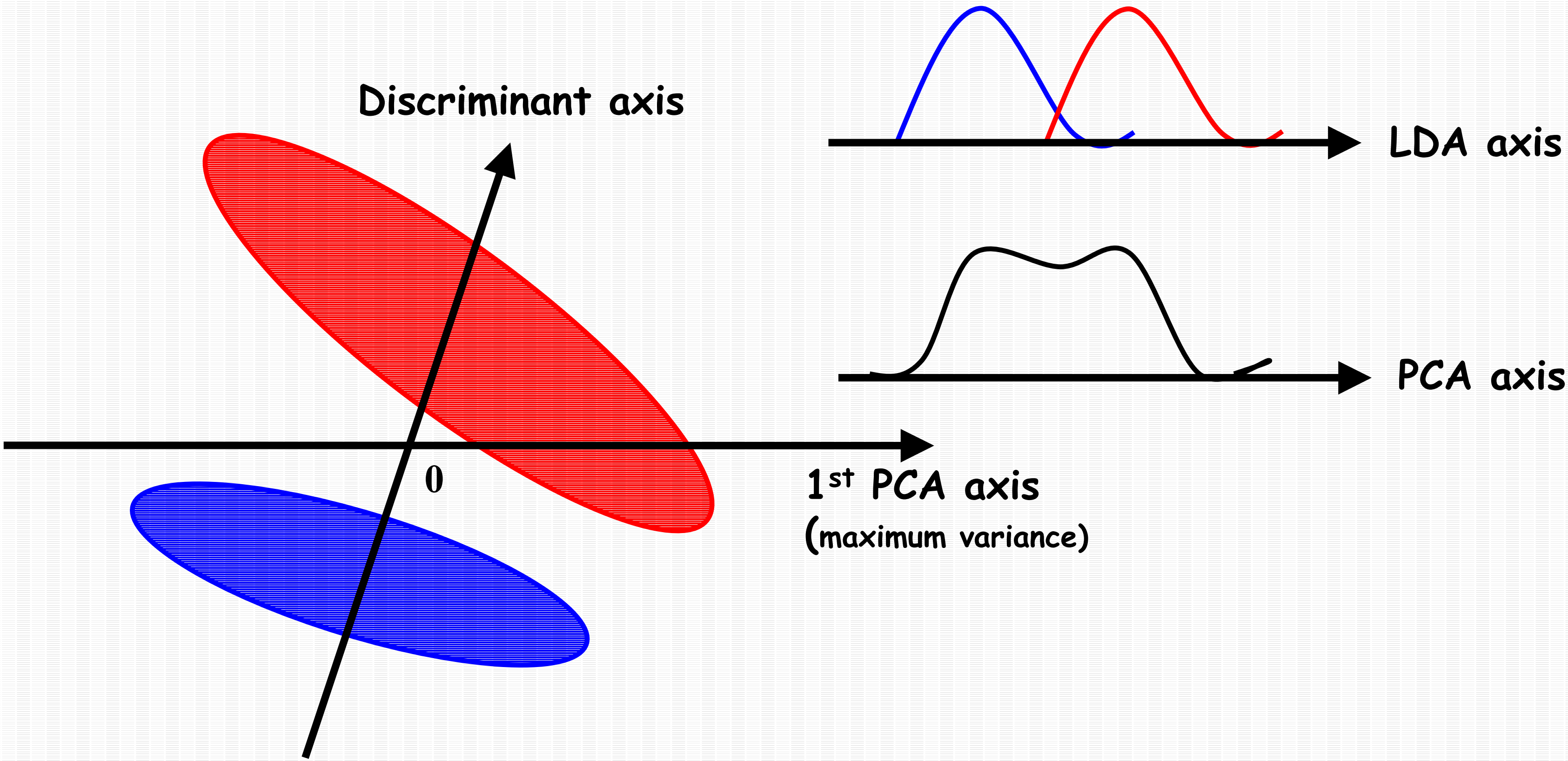
$$k(\vec{x},\vec{y}) = \tanh\left(a\,\vec{x}^T \cdot \vec{y} + b\right)$$

- Homogenous polynomial: $\left(N_F = \frac{(d+N_X-1)!}{d!\,(N_X-1)!}\right)$

$$k(\vec{x},\vec{y}) = \left(\vec{x}^T \cdot \vec{y}\right)^d \qquad \forall d \in \{1,2,3,\ldots\}$$

Linear Discriminant Analysis LDA

- LDA versus PCA projection:



LDA
Classical criterion

- Let's assume N clusters and M samples:
- C : *total covariance matrix*
- G : *covariance matrix of the centers*
- $\vec{v}_i$: *i*[th] *discriminant axis (i=1,…, N-1)*
- LDA maximizes the variance ratio:

$$\lambda_i = \frac{\text{Inter-class variance}}{\text{Total variance}} \longrightarrow \lambda_i = \frac{\vec{v}_i^{\,T} G \vec{v}_i}{\vec{v}_i^{\,T} C \vec{v}_i}$$

LDA
Resolution

- The solution is based on an eigen system:

$$\lambda_i \vec{v}_i = C^{-1} G \vec{v}_i$$

- The eigen vectors are linear combinations of the learning samples:

$$\vec{v}_i = \sum_{j=1}^{M} \alpha_{ij} \vec{x}_j$$

Generalized Discriminant Analysis (GDA)

- LDA in the feature space F.
- The eigen vectors are linear combinations of the learning samples:

$$\vec{v}_i = \sum_{j=1}^{M} \alpha_{ij} \phi(\vec{x}_j)$$

- Consequently any projection becomes:

$$\vec{v}_i^T \cdot \vec{z} = \sum_{j=1}^{M} \alpha_{ij} k(\vec{x}_j, \vec{z})$$

GDA
Covariance matrixes in F

- We assume the data are centered in F.
- Total covariance matrix:

$$\mathbf{V} = \frac{1}{M}\sum_{j=1}^{M}\phi(\vec{x}_j)\phi^T(\vec{x}_j)$$

- Covariance matrix of the N centers:

$$\mathbf{B} = \frac{1}{M}\sum_{l=1}^{N} n_l \overline{\phi}_l \overline{\phi}_l^{\,T} \qquad \overline{\phi}_l = \frac{1}{n_l}\sum_{k=1}^{n_l}\phi(\vec{x}_{lk})$$

GDA Resolution

- Let K be the kernel matrix (MxM):

$$K = (k(\vec{x}_i, \vec{x}_j))_{\substack{i=1,\ldots,M \\ j=1,\ldots,M}}$$

- The LDA criterion in F becomes:

$$\lambda_i = \frac{\vec{\alpha}_i^T KWK\vec{\alpha}_i}{\vec{\alpha}_i^T K^2 \vec{\alpha}_i}$$

where W is a (MxM) bloc diagonal matrix of weights $1/n_l$.

GDA
Resolution (cont.)

- Let's use an eigen decomposition of K:

$$K = U \Gamma U^T$$

- Then by substitution:

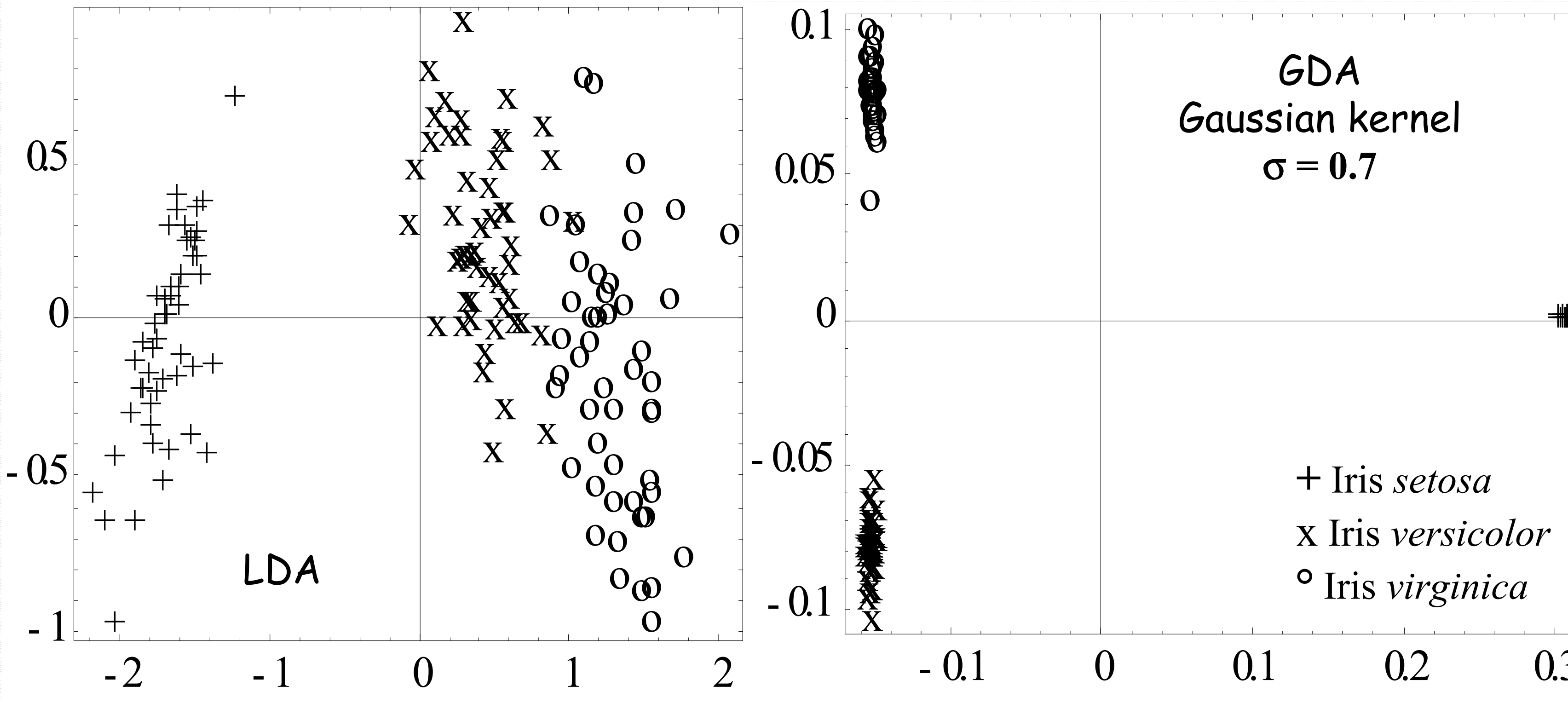
$$\lambda_i = \frac{\vec{\beta}_i^T U^T W U \vec{\beta}_i}{\vec{\beta}_i^T \vec{\beta}_i} \qquad where\ \vec{\beta}_i = \Gamma U^T \vec{\alpha}_i$$

- Finally it is just a classical eigen system:

$$\lambda_i \vec{\beta}_i = U^T W U \vec{\beta}_i$$

GDA
An example

- Fisher's iris data (3 clusters, 4D).



GDA
An application

- Seed classification (SNES-France)

  3 classes: *Medicago sativa* L., *Melilotus sp* & *Medicago lupulina* L

Medicago sativa L

Medicago lupulina L

Methods	*Learning error*	*Test error*
LDA	27.2%	32.7%
GDA*	0%	14.9%
Probabilistic NN	0%	14.4%

* Gaussian kernel (σ=0.5)

GDA
Some comments

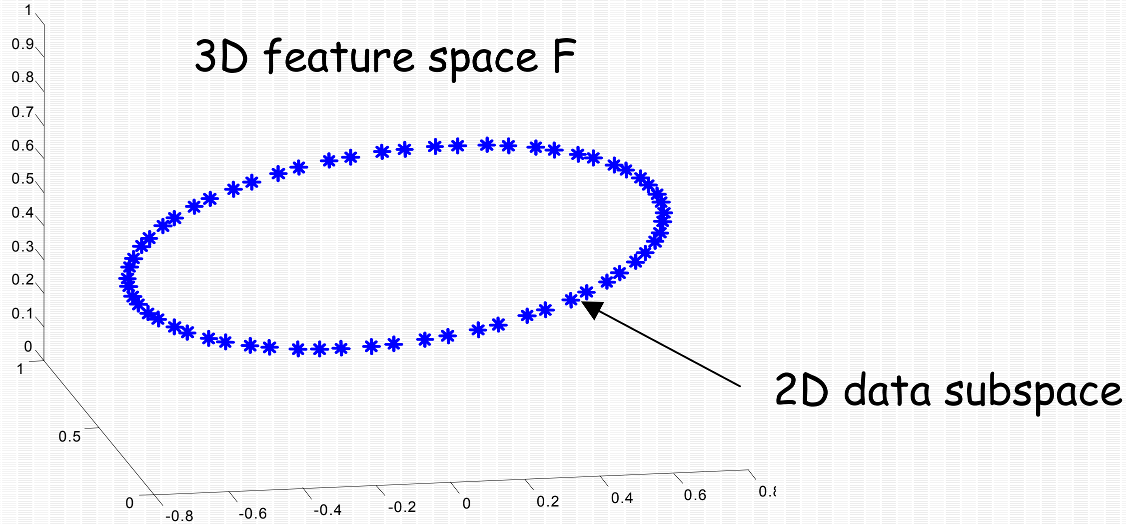
- $k(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T \vec{x}_j$ defines the LDA in X.
- The α_i coefficients are not unique. One possible solution is:

$$\vec{\alpha}_i = U\Gamma^{-1}\vec{\beta}_i$$

- Without special care this leads to a dense expansion for the discriminant axes. Meaning the all M samples are involved.

Feature Vector Selection (FVS)

- Often the data spans a subspace in F with a dimension lower than the size M of the learning data.



- **Idea**: Describe this subspace by L Feature Vectors (FV) taken among the samples.
They define a basis S in F, with $L \leq M$.

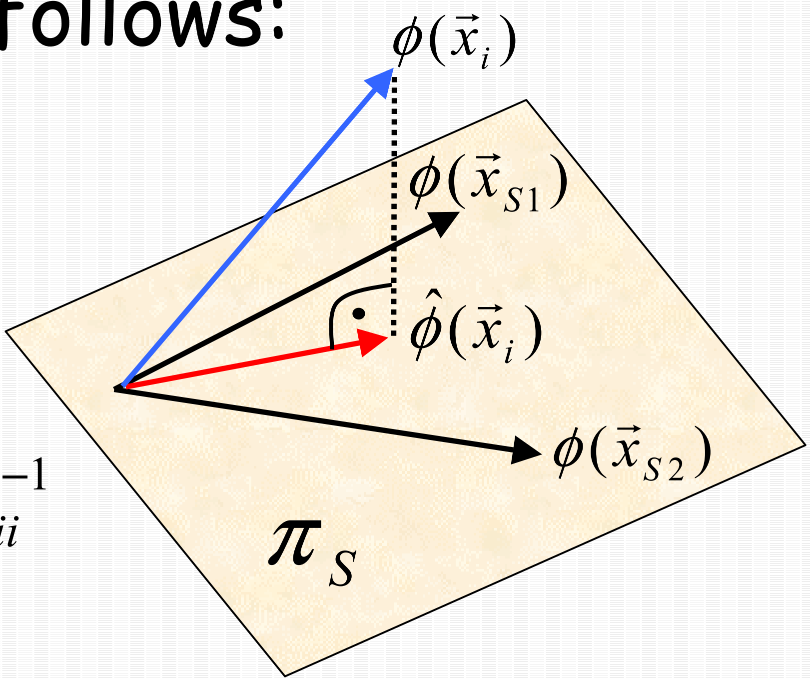
FVS Algorithm

- The FVS is based on a sequential forward selection maximizing a fitness function defined as follows:

$$J_S = \frac{1}{M}\sum_{i=1}^{M}\frac{\left\|\hat{\phi}(\vec{x}_i)\right\|^2}{\left\|\phi(\vec{x}_i)\right\|^2}$$

$$J_S = \frac{1}{M}\sum_{i=1}^{M}\vec{K}_{Si}^{T}K_{SS}^{-1}\vec{K}_{Si}\,k_{ii}^{-1}$$

$$0 \le J_S \le 1$$



FVS
Empirical kernel map

- After the FVS we can project any sample using the basis S. This provide new explicit vectors:

$$\vec{K}_{Si} = \left(k(\vec{x}_{S1}, \vec{x}_i), \ldots, k(\vec{x}_{SL}, \vec{x}_i)\right)^T$$

- This is known as an empirical kernel map.

FVS
Sparse GDA using FVS

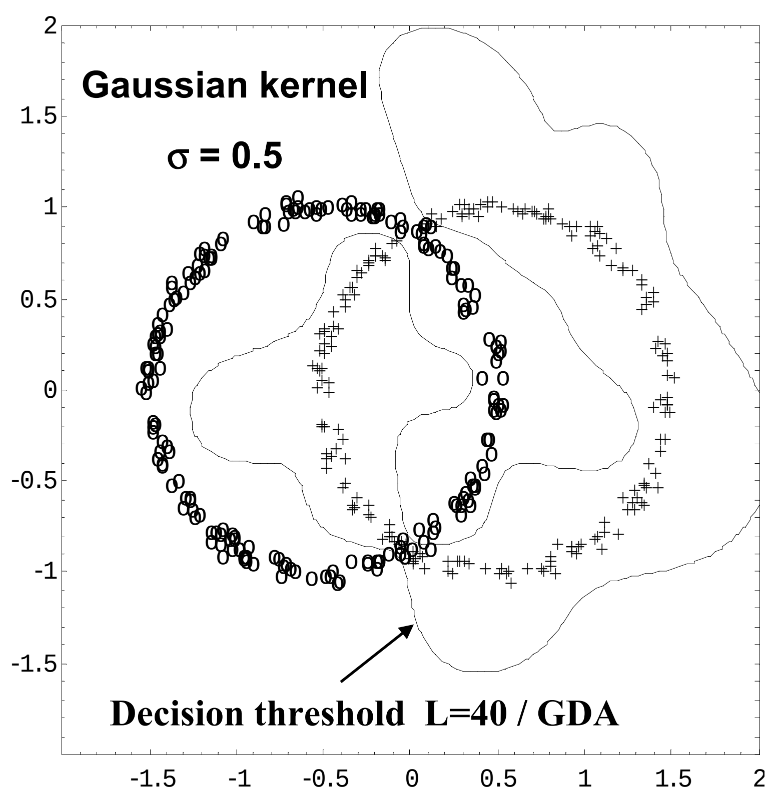
- After the FVS and projection we use the LDA to approximate the GDA.

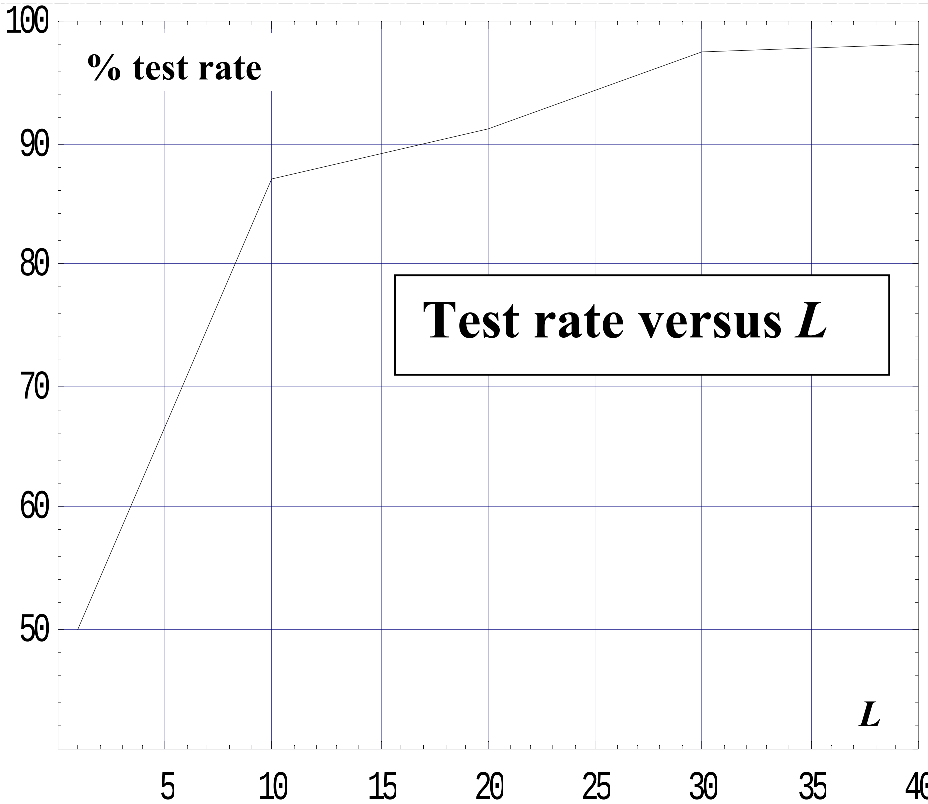
- Then any projection on a discriminant axis uses an expansion of only L terms:

$$\vec{v}_i^T \cdot \vec{z} = \sum_{j=1}^{L} \alpha_{ij} k(\vec{x}_{Sj}, \vec{z})$$

FVS-GDA
An example

- 2 clusters ('o' & '+'). 100 samples to learn and 100 others for testing.





Conclusions

- The GDA allows reusing the LDA approach for non-linear cases.

- The projection of samples using a non-linear discriminant scheme provides a convenient way to visualize, analyze, and perform other tasks, such as classification with linear methods.

- Sparse techniques such as FVS overcome the cost of a dense expansion for the discriminant axes.

Some references

- Mika S., Rätsch G., Weston J., Schölkopf B., Müller K. R., **"Fisher Discriminant Analysis with Kernels"**. Proc. *IEEE Neural Networks for Signal Processing Workshop*, NNSP, 1999.

- Mika S., Smola A., Schölkopf B., **"An Improved Training Algorithm for Kernel Fisher Discriminants"**. In *Artifical Intelligence and Statistics*, pages 98-104, San-Fransisco, CA USA, 2001. Morgan Kaufmann.

- Mika S., Rätsch G., Müller K. R., **"A Mathematical Programming Approach to the Kernel Fisher Discriminant"**. In *Advances in Neural Information Processing Systems 13*, 2001 (to appear).

- Schölkopf B., Smola A., "**Learning with Kernels**". *The MIT press, Cambridge, Massachusetts*, 2002.