Fisher Discriminant Analysis with Kernels

- by Mika, Rätsch, Weston, Schölkopf, Müller
- presented by Boley.
Discriminate between two classes

- Need to identify good set of features
- PCA: unsupervised algorithm to reduce reconstruction error
- Better to take advantage of label info
- Classical approaches: bayes classifier - requires assumptions on data distribution within each class
- Often: assume Gaussian distribution within each class
  \( \rightarrow \) leads to quadratic or linear discriminants, like Fisher
This work

- Authors propose kernel idea used in SVMs, K-PCA.
- Use in supervised Fisher’s Discriminant
- Result often competitive with K SVMs.
- Dot-product in kernel space $\rightarrow$ closed form solution
Classical Fisher Linear Discriminant

- samples from two classes: $X_1 = [x_1, \ldots, x_{\ell_1}]$, $X_2 = [x_{\ell_1+1}, \ldots, x_{\ell_1+\ell_2}]$, with $\ell = \ell_1 + \ell_2$.

- Fisher’s discriminant projects all the data onto a direction $w$ maximizing the separation of the means along the projection while minimizing the scatter with each class

$$\max J(w) = \frac{w^T S_B w}{w^T S_W w}$$

where

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

between cluster scatter

$$S_W = \sum_{i=1,2} \sum_{x \in X_i} (x - m_i)(x - m_i)^T$$

within class scatter

$$m_i = \frac{1}{\ell_i} \sum_{x \in X_i} x$$

class mean

$$m = \frac{1}{\ell} \sum x = \frac{\ell_1}{\ell} m_1 + \frac{\ell_2}{\ell} m_2$$

global mean
Statistical Motivation - Bayes

- Optimal Bayes assigns class based on maximum a-posteriori probability
- Simplifying assumption: each class has a normal distribution
- Measures Mahalanobis distance of a sample to class center
- Result is a quadratic separator
- With a single common Covariance matrix $\rightarrow$ linear separator
- Linear separator advantage: robust against noise
- Direction of separator aligned with direction of maximal variance within each class
- Linear separator $\leftrightarrow$ Fisher’s $w$.
- Crucial: have enough samples to get good estimate of Covariance.
Fisher’s discriminant in feature space

• Linear discriminant is not rich enough
• Want to keep robustness and statistical foundation while allowing richer separators
• Answer: use high-dimensional feature space $\mathcal{F}$
• Map $x \mapsto \hat{x} = \phi(x) \in \mathcal{F}$.
• Fisher’s Disc. is now:

$$\max J(w) = \frac{w^T \hat{S}_B w}{w^T \hat{S}_W w}$$

where

$$\hat{S}_B = (\hat{m}_1 - \hat{m}_2)(\hat{m}_1 - \hat{m}_2)^T$$
between cluster scatter

$$\hat{S}_W = \sum_{i=1,2} \sum_{\hat{x} \in \hat{X}_i} (\hat{x} - \hat{m}_i)(\hat{x} - \hat{m}_i)^T$$
within class scatter

$$\hat{m}_i = \frac{1}{\ell_i} \sum_{\hat{x} \in \hat{X}_i} \hat{x}$$
class mean

$$\hat{m} = \frac{1}{\ell} \sum_{\hat{x}} x = \frac{\ell_1}{\ell} \hat{m}_1 + \frac{\ell_2}{\ell} \hat{m}_2$$
global mean
Kernel Function

• Need to formulate problem in terms of dot-products of input patterns
• Any solution \( w \) must lie in span of training samples \( \hat{x}_1, \ldots, \hat{x}_\ell \) in \( F \).
• \( w = \sum_1^\ell \alpha_j \hat{x}_j = \sum_1^\ell \alpha_j \phi(x_j) \).
• Inner Product with mean: \( w^T \hat{m}_i = \sum_{j=1}^\ell \alpha_j \frac{1}{\ell_i} \sum_{x \in X_i} k(x_j, x) \).
• Wish to optimize \( \max J(w) = w^T \hat{S}_B w / w^T \hat{S}_W w \)
• Numerator: \( w^T \hat{S}_B w = \alpha^T (M_1 - M_2)(M_1 - M_2)^T \alpha \)
• Here \( M_i \) is the \( \ell \)-vector of weighted row sums of the kernel matrix \( K = \{K_{ij}\} = \{k(x_i, x_j)\}_{i,j=1,\ldots,\ell} \).
Kernel Function 2

- Wish to optimize $\max J(w) = w^T \hat{S}_B w / w^T \hat{S}_W w$

- Denominator: $w^T \hat{S}_W w = \alpha^T \left( \frac{K_1 (I - 1_{\ell_1}) K_1^T}{N} + \frac{K_2 (I - 1_{\ell_2}) K_2^T}{N} \right) \alpha$

where $K_1 = \{(K_1)_{ij}\} = \{k(x_i, x_j)\}_{i=1,\ldots,\ell}^{j=1,\ldots,\ell_1} \ (\ell \times \ell_1 \text{ matrix})$

$K_2 = \{(K_2)_{ij}\} = \{k(x_i, x_j)\}_{i=1,\ldots,\ell}^{j=1,\ldots,\ell_2} \ (\ell \times \ell_2 \text{ matrix})$

$K = (K_1, K_2)$. 
Kernel Fisher Discriminant

• KFD is now solved by optimizing

\[
\max J(w) = \frac{w^T N w}{w^T M w}.
\]

• Solve by finding leading eigenvector of \( N^{-1} M \) [or better, solve generalized eigenproblem \( M w = \lambda N w \)].

• Project new pattern \( \hat{x} = \phi(x) \) onto \( w \) by

\[
\langle w, \phi(x) \rangle = \sum_{i=1}^{\ell} \alpha_i k(x_i, x)
\]
Numerical Issues

• Estimating $\ell$ covariance structures from $\ell$ samples $\rightarrow$ ill-posed.

• $N$ could be singular or badly conditioned

• Need capacity control in $\mathcal{F}$

Solution

• Replace $N$ with $N_\mu = N + \mu I$.

• Effect: Makes $N$ better conditioned

• Decreases bias in sample-based eigenvalue estimates

• Imposes regularization on $\|\alpha\|^2$, favoring solutions with small expansion coefficients.

• Regularization effect not fully understood.

• Other forms of regularization possible.
Figure 1: Comparison of feature found by KFD (left) and those found by Kernel PCA: first (middle) and second (right); details see text.

- KFD: polynomial kernel degree two, regularized with $\mu = 10^{-3}$.
- Two classes (×’s & ’s), parabolic mirrored around axes.
- Contour lines = level sets
- KFD level sets discriminate classes well
- KPCA less so.
Experiments

- Compare to other state-of-the-art classifiers
- KFD: Kernel Fisher Discriminant with Gaussian kernel
  - Once $w$ obtained, used 1-d linear SVM to classify
- Adaboost
- Regularized Adaboost
- SVM: Support Vector Machine with Gaussian kernel
Data Sets

- Sources: ICI DELVE STATLOG Benchmark data sets
- Treated all as two-class problems
- 100 partitions into training/test sets (about 60%:40%)
- Hyperparameters estimated using 5-fold cross-validation over first 5 realizations
- Table shows average test error & standard deviation over 100 runs
Results

Preliminary Experiment with USPS Digit Data

• Used 3000 training samples
• Compared KFD with KSVM, both with Gaussian kernels
• 10 class error: KFD: 3.7%, KSVM: 4.2%

In General

• Noticed: both KFD & SVM yield optimal hyperplane in $\mathcal{F}$: often former is better.
• Complexity of SVM classifier is $O(\text{supportvectors})$.
• Complexity of KFD classifier is $O(\text{alltrainingvectors})$.
• Dependence on all training vectors $\rightarrow$ maybe more robust.
• KFD: closed form solution.
  Other methods involve a search or an optimization problem.
• Table on next page: 1st place in bold, 2nd place in italic (lower is better)
# Experiments

Table 1: Comparison between KFD, a single RBF classifier, AdaBoost (AB), regularized AdaBoost (AB<sub>R</sub>) and Support Vector Machine (SVM) (see text). Best method in bold face, second best emphasized.

<table>
<thead>
<tr>
<th></th>
<th>RBF</th>
<th>AB</th>
<th>AB&lt;sub&gt;R&lt;/sub&gt;</th>
<th>SVM</th>
<th>KFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>10.8±0.6</td>
<td>12.3±0.7</td>
<td>10.9±0.4</td>
<td>11.5±0.7</td>
<td>10.8±0.5</td>
</tr>
<tr>
<td>B.Cancer</td>
<td>27.6±4.7</td>
<td>30.4±4.7</td>
<td>26.5±4.5</td>
<td>26.0±4.7</td>
<td>25.8±4.6</td>
</tr>
<tr>
<td>Diabetes</td>
<td>24.3±1.9</td>
<td>26.5±2.3</td>
<td>23.8±1.8</td>
<td>23.5±1.7</td>
<td>23.2±1.6</td>
</tr>
<tr>
<td>German</td>
<td>24.7±2.4</td>
<td>27.5±2.5</td>
<td>24.3±2.1</td>
<td>23.6±2.1</td>
<td>23.7±2.2</td>
</tr>
<tr>
<td>Heart</td>
<td>17.6±3.3</td>
<td>20.3±3.4</td>
<td>16.5±3.5</td>
<td>16.0±3.3</td>
<td>16.1±3.4</td>
</tr>
<tr>
<td>Image</td>
<td>3.3±0.6</td>
<td>2.7±0.7</td>
<td>2.7±0.6</td>
<td>3.0±0.6</td>
<td>4.8±0.6</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>1.7±0.2</td>
<td>1.9±0.3</td>
<td>1.6±0.1</td>
<td>1.7±0.1</td>
<td>1.5±0.1</td>
</tr>
<tr>
<td>F.Sonar</td>
<td>34.4±2.0</td>
<td>35.7±1.8</td>
<td>34.2±2.2</td>
<td>32.4±1.8</td>
<td>33.2±1.7</td>
</tr>
<tr>
<td>Splice</td>
<td>10.0±1.0</td>
<td>10.1±0.5</td>
<td>9.5±0.7</td>
<td>10.9±0.7</td>
<td>10.5±0.6</td>
</tr>
<tr>
<td>Thyroid</td>
<td>4.5±2.1</td>
<td>4.4±2.2</td>
<td>4.6±2.2</td>
<td>4.8±2.2</td>
<td>4.2±2.1</td>
</tr>
<tr>
<td>Titanic</td>
<td>23.3±1.3</td>
<td>22.6±1.2</td>
<td>22.6±1.2</td>
<td>22.4±1.0</td>
<td>23.2±2.0</td>
</tr>
<tr>
<td>Twonorm</td>
<td>2.9±0.3</td>
<td>3.0±0.3</td>
<td>2.7±0.2</td>
<td>3.0±0.2</td>
<td>2.6±0.2</td>
</tr>
<tr>
<td>Waveform</td>
<td>10.7±1.1</td>
<td>10.8±0.6</td>
<td>9.8±0.8</td>
<td>9.9±0.4</td>
<td>9.9±0.4</td>
</tr>
</tbody>
</table>
Conclusions and Discussion

• Fisher’s discriminant: standard linear statistical technique, but too limited.
• This is one of many approaches to obtain more general class separability.
• Advantage: closed form solution.
• Flexibility: wide choice of kernels.
• Experimental results: competitive with many other methods.
• Complexity scales with all training samples (not just the difficult ones)

Future Work

• Suitable approximation schemes
• Numerical methods to find a few leading eigenvectors
• Multi-class discriminants
• Generalization bounds.
Novelty Detection

Kernel PCA for Novelty Detection by Heiko Hoffman

- Novelty Detection is a one-class classification problem.
- Use training data to see typical acceptable data.
- Called One-Class because training data contains only acceptable data.
- Test data may be similar to training data or not: objective is to distinguish those that are different.
- Abnormal examples are generally rare.
- Alternate algorithm: One-class SVM: find tightest separator from origin in $F$.
- Alternate algorithm: SVDD: Find smallest enclosing sphere in kernel space $F$. RBF kernel leads to same as one-class SVM.
- Here we try to generate a simplified model.
- Alternate approaches: Gaussian Mixture models, auto-associative multilayer perceptrons, principal curves and surfaces, All these lead to non-linear (often non-convex) optimization problems.
- Here we use PCA in kernel space to reduce dimensionality.
Method

• Training data are mapped into an infinite-dimensional feature space.

• In this space, kernel PCA extracts the principal components of the data distribution. Eigenvectors \( \{v_\ell\}_{\ell=1}^q \) of \( \bar{K} \) with \( \bar{K}_{ij} = K_{ij} - \frac{1}{n} \sum_r K_{ir} - \frac{1}{n} \sum_r K_{rj} + \frac{1}{n^2} \sum_r \sum_s K_{rs} \) where \( K_{ij} = k(x_i, x_j) \).

• Potential: \( p_S(z) = \|\phi(z) - \bar{\phi}\|_2^2 = k(z, z) - \frac{2}{n} \sum_{i=1}^n k(z, x_i) + \frac{1}{n^2} \sum_{i,j} k(x_i, x_j) \)

• Projection: \( f_\ell(z) = \left\langle \left[ \phi(z) - \frac{1}{n} \sum_{r=1}^n \phi(x_r) \right], [v_\ell - \bar{\phi}(x)] \right\rangle \)

where \( v_\ell = \ell\)-th eigenvector & \( \bar{\phi}(x) \) is center in \( \mathcal{F} \) (both linear comb’s of \( \phi(x_i)'s \)).

• The squared distance to the corresponding principal subspace is the measure for novelty:

\[ p(z) = p_S(z) - \sum_{i=1}^q f_\ell(z)^2 \]
Fig. 12. The difference between the distance to be optimized in denoising and the reconstruction error $p$. 

principal component

$\Phi (z)$

$\Phi (x)$

denoising distance

$\sqrt{p}$

← center?
Decision Boundary Sketch

Fig. 1. Decision boundaries in the feature space of an RBF kernel, comparing one-class SVM, SVDD, and the reconstruction error: (A) The boundaries are illustrated in a three-dimensional feature space. All data points $\Phi(x_i)$ lie on a sphere. (B) Cross-section through the center of the SVDD sphere and orthogonal to the principal component for the situation in A.
Illustration

(A) Reconstruction-error boundary

(B) One-class SVM boundary

SVDD boundary

Principal component

Reconstruction-error boundary

One-class SVM boundary

Origin

Reconstruction-error boundary
Example - classical methods

PCA

Parzen
Example - kernel methods

Kernel PCA, polynomial

Kernel PCA, RBF
Fig. 3. Decision boundary for the ring-line-square distribution using the reconstruction error in $\mathcal{F}$ with $\sigma = 0.4$ and $q = 40$. 
Fig. 4. Decision boundary for the spiral distribution using the reconstruction error in $\mathcal{F}$ with $\sigma = 0.25$ and $q = 40$. 
Noisy Data - One-class SVM

One-class SVM

K-LDA-slides.17.10.4.135
Noisy Data - K-PCA

Kernel PCA

![Diagram of Noisy Data and Kernel PCA](image)
Samples from sine curve plus uniform noise

Fig. 6. Decision boundaries for the sine-noise distribution comparing kernel PCA ($\sigma = 0.4$, $q = 40$) with the one-class SVM ($\sigma = 0.489$, $v = \frac{2}{\pi}$).
Vary Parameters: $\sigma = .05$

$q = 0$

$q = 40$
Vary Parameters: $\sigma = 0.10$

$q = 0$

$q = 40$
Vary Parameters: $\sigma = .40$
Real Data ROC curves: Classifier

Digit 0

Cancer

KPCA

1-class SVM, $\sigma = 1.2$

1-class SVM, $\sigma = 2$

1-class SVM, $\sigma = 10$

true positives

false positives

true positives

false positives
Real Data: vary kernel width

Digit 0

Cancer

area under ROC curve

area under ROC curve

kernel width $\sigma$

kernel width $\sigma$

$q = 200$
$q = 10$
Parzen
1-class SVM

$0.9958$
$0.996$
$0.9962$
$0.9964$
$0.9966$
$0.9968$
$0.997$
$0.9972$
Real Data: vary $\#$ eigenvectors

Digit 0

Cancer

$\sigma = 4$
$\sigma = 50$
$\sigma = 100$
PCA

$\sigma = 1$
$\sigma = 5$
$\sigma = 15$
PCA

area under ROC curve

num. of eigenvectors $q$
Most Unusual Zero Digits

Fig. 11. The 10 most unusual ‘0’ digits from the MNIST test set. The digits are arranged in descending order of their reconstruction error $p$ ($\sigma = 4$, $q = 100$). The figure shows the unprocessed digits of size $28 \times 28$ pixels; for novelty detection, however, the processed digits ($8 \times 8$ pixels) were used.
Fig. 12. The difference between the distance to be optimized in denoising and the reconstruction error $p$. 