《机器学习》课件

主成分分析（PCA）

北京航空航天大學

BEIHANG UNIVERSITY

PCA降维：Theory

- 在变换后的特征空间中，每个特征向量w_i对应的特征值λ_i的大小代表该特征向量所描述的方向上的方差的大小

所以…

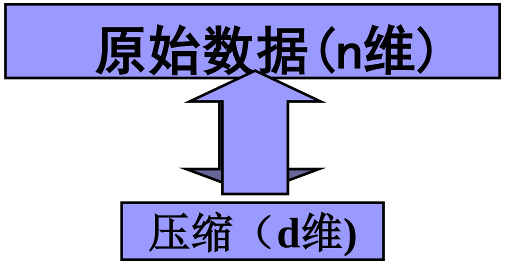
- 从W中去掉那些对应较小特征值的特征向量，意味着在信息丢失最小的意义上降维！

PCA降维：Practice

- 按照其所相应的特征值的大小对特征向量排序
- 选择头d个对应最大特征值的特征向量构成变换矩阵W_{nxd}

从n维空间到d维空间的投影

(d< n)!



$$y = W^T(x-\bar{x}) \qquad x = \bar{x} + Wy$$

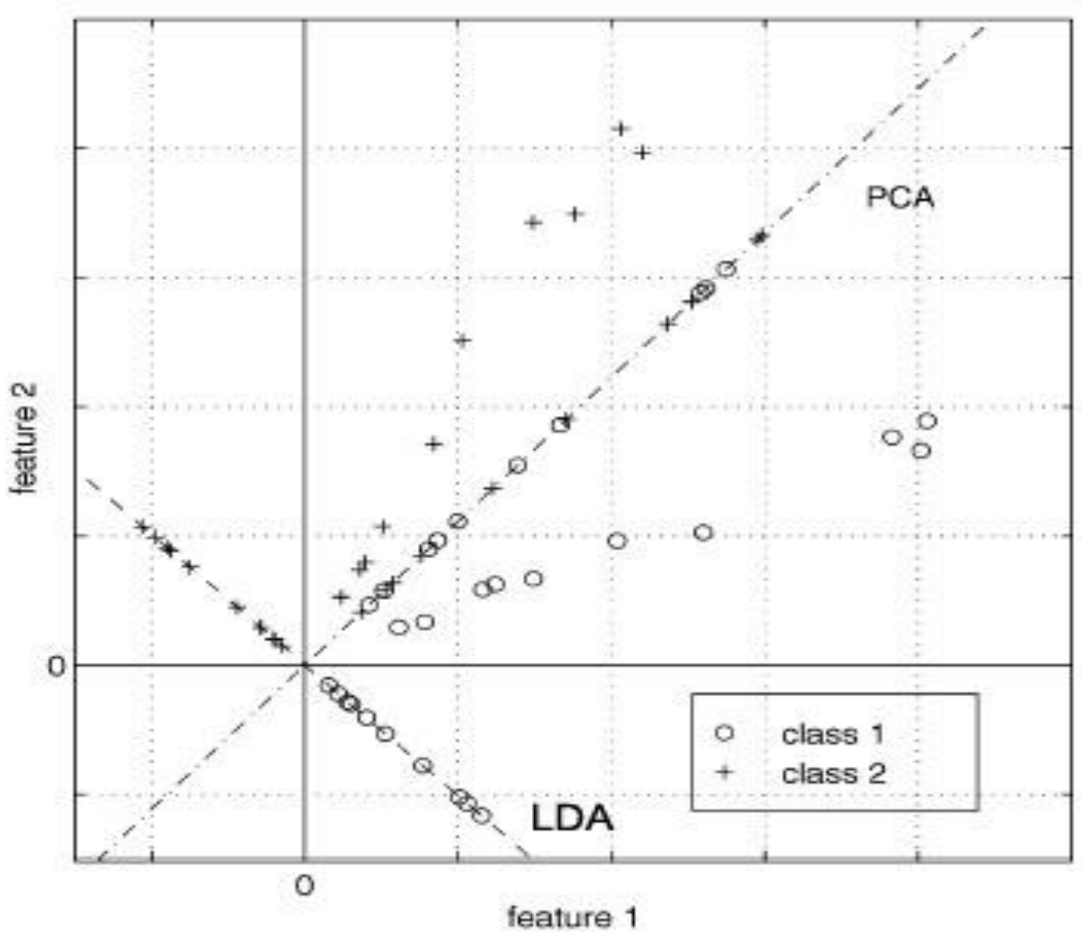
人脸识别

特征人脸

PCA方法的优缺点

- 从压缩能量的角度看，PCA方法是最优的。从高维空间降到低维空间后，它不仅使得和原样本的均方误差最小，而且变换后的低维空间有很好的人脸表达能力
- 但是没有考虑到人脸的类别信息
- PCA用于人脸识别并不是一个很好的方法，它只是起了信息压缩减少特征的降维作用，提高了以后的识别效率。

PCA和LDA产生的两个不同的线性投影方向



Fisher线性投影方向

Linear Discriminant Analysis(LDA)

Fisher方法推导

最大化Fisher的判别准则：

$$J(W)=\frac{tr(W^TS_bW)}{tr(W^TS_wW)}$$

为什么这么定义？

线性变换之后使得不同类的样本（平均类间距离）尽可能远，同类样本（平均类内距离）尽可能近。

Fisher方法推导

线性变换或者投影之后使得不同类的样本（平均类间距离）尽可能远

$$W^T(m_1 - m_2)(m_1 - m_2)^T W$$

同类样本（平均类内距离）尽可能近

$$\sum_i W^T(x^i - m_i)(x^i - m_i)^T W$$

类间散度矩阵

- Ω_i类与Ω_j类之间的类间散度矩阵：

$$S_B^{(ij)} = \left(\mathbf{m}^{(i)} - \mathbf{m}^{(j)}\right)\left(\mathbf{m}^{(i)} - \mathbf{m}^{(j)}\right)^T$$

- 总的类间散度矩阵：

$$S_B = \frac{1}{2}\sum_{i=1}^{M} P(\Omega_i)\sum_{j=1}^{M} P(\Omega_j) S_B^{(ij)} = \frac{1}{2}\sum_{i=1}^{M} P(\Omega_i)\sum_{j=1}^{M} P(\Omega_i)\left(\mathbf{m}^{(i)} - \mathbf{m}^{(j)}\right)\left(\mathbf{m}^{(i)} - \mathbf{m}^{(j)}\right)$$

类内散度矩阵

- Ω_i类的类内散度矩阵：

$$S_w^{(i)} = \frac{1}{N_i}\sum_{k=1}^{N_i}\left(\mathbf{X}_k^{(i)} - \mathbf{m}^{(i)}\right)\left(\mathbf{X}_k^{(i)} - \mathbf{m}^{(i)}\right)^T$$

- 总的类内散度矩阵：

$$S_w = \sum_{i=1}^{M} P(\Omega_i) S_w^{(i)} = \sum_{i=1}^{M} P(\Omega_i)\frac{1}{N_i}\sum_{k=1}^{N_i}\left(\mathbf{X}_k^{(i)} - \mathbf{m}^{(i)}\right)\left(\mathbf{X}_k^{(i)} - \mathbf{m}^{(i)}\right)^T$$

Fisher方法推导

最大化如下表达式，可以满足我们的要求：

$$J(W) = \frac{W^T S_b W}{W^T S_w W}$$

如何计算？

$$W^T S_w W = c \neq 0$$

定义Lagrange函数为：

$$L(w, \lambda) = W^T S_b W - \lambda (W^T S_w W - c)$$

Fisher方法推导

如何计算？

$$\frac{\partial L(w,\lambda)}{\partial w} = S_b w - \lambda S_w w = 0$$

$$S_b w = \lambda S_w w$$

$$S_w^{-1} S_b w = \lambda w$$

则可以利用线性代数中的方法求解

Fisher基本过程

- Fisher的投影方向是下面方程的解：

$$S_w^{-1} S_b w = \lambda w$$

- 可以证明 S_w的秩最大为N-C，所以当N-C<d时,S_w一定是奇异的。N训练样本的个数

Fisherface方法 = PCA+Fisher

1. 用PCA降维。运用PCA方法将S_w降至p=N-C维。

$$S_w = W_{pca}^T S_w W_{pca} \qquad S_b = W_{pca}^T S_b W_{pca}$$

$$W_{pca} = (u_{pca_1}, u_{pca_2}, \quad , u_{pca_p})$$

为S_t最大的前N-C个特征值对应的特征向量。

2. 运用上述Fisher方法求 $W_{lda} = \arg\max_{W} \dfrac{tr(W^T S_b W)}{tr(W^T S_w W)}$

最后求出理想的投影矩阵为: $W_{opt}^T = W_{lda}^T W_{pca}^T$

参数选取

$$p = N - C - 10$$

- 这是因为在训练的人脸图像中可能有些比较相像，S_w的秩不一定能达到最大（N−C），或者降到N−C维时仍然很接近奇异，所以在PCA方法中采取多降几维。

$$L = C - 1$$

- 这是因为S_b的秩最大为C−1，前C−1个特征向量已经代表了全部的类间散度的信息，L取太大并不能保留更多的类间散度，反而会保留更多的类内散度，对分类无益。

经典文章结论

- Belhumeur对用特征脸方法和Fisher脸方法分别求出来的一些特征脸进行比较后得出结论，认为特征脸方法很大程度上反映了光照等差异，而Fisher脸方法则能去掉图像之间的与识别信息无关的差异。
- Belhumeur的实验是通过对160幅人脸图像（一共16人，每个人10幅不同条件下的图像）进行测试的，采用特征脸方法的识别率为81%，而采用Fisher脸方法的识别率为99.4%。显然，Fisher脸方法有了很大的改进。

作业

- 简要描述特征脸方法和Fisher脸方法的异同点。

PCA

- **Regular PCA**:
  Find the direction u s.t. projecting n points in d dimensions onto u gives the largest variance.

- u is the eigenvector of covariance matrix $Cu=\lambda u$.

Kernel PCA

- Kernel PCA is used for:
  - De-noising
  - Compression
  - Interpretation (Visualization)
  - Extract features for classifiers

Why Use Kernel

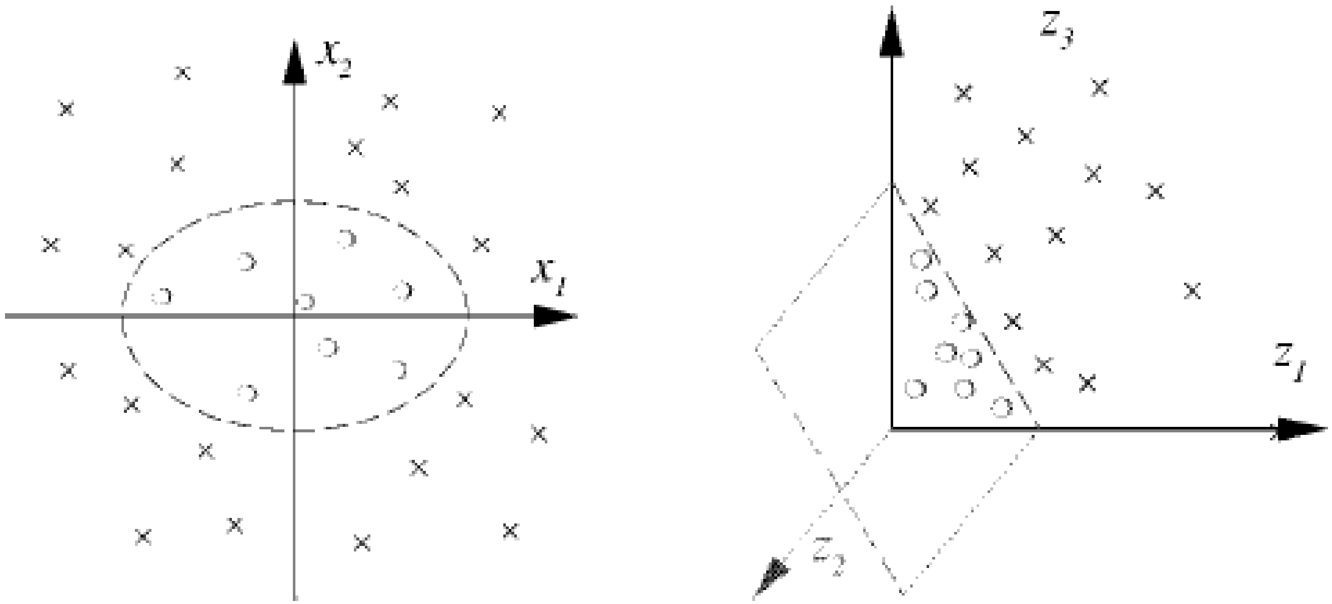


Fig. 4. Two-dimensional classification example. (a) Using the second-order monomials x_1^2, $\sqrt{2}\,x_1x_2$ and x_2^2 as features a separation in feature space can be found using a *linear* hyperplane. (b) In input space this construction corresponds to a *nonlinear* ellipsoidal decision boundary (figure from [48]).

Why Use Kernel

- 涉及 … 中只 需进 … 。根 据H … r条 件，…

一些典型的核函数

- SVM中不同的内积核函数将形成不同的算法，常用的核函数有三类：
- 多项式核函数 $K(x,x_i)=[(x.x_i)+1]^q$
- 径向基函数 $K(x,x_i)=\exp(-\frac{|x-x_i|^2}{\sigma^2})$
- S形函数 $K(x,x_i)=\tanh(v(x.x_i)+c)$

Hibert

- Hibert-Schmidt
- 希尔伯特，D.(Hilbert,David，1862～1943)德国数学家
- 1880年,他不顾父亲让他学法律的意愿,进入哥尼斯堡大学攻读数学。
- 1893年被任命为正教授
- 1942年成为柏林科学院荣誉院士。
- 希尔伯特是一位正直的科学家,他敢于公开发表文章悼念“敌人的数学家”达布

Kernel PCA

- Extension to feature space:
  - compute covariance matrix based 0-mean data

$$C = \frac{1}{l}\sum_{j=1}^{l} \Phi(\mathbf{x}_i)\Phi(\mathbf{x}_j)^T$$

  - solve ei CV=λV

$$V = \sum_{i=1}^{l} \alpha_i \Phi(\mathbf{x}_i)$$

Kernel PCA

- Define in terms of dot products:

$$\lambda(\Phi(\mathbf{x}_k)\cdot\mathbf{V}) = (\Phi(\mathbf{x}_k)\cdot C\mathbf{V})$$

- Then the problem becomes:

$$\lambda\alpha = K\alpha$$

l xd rather than d x d

where

$$K_{ij} = (\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)) = k(\mathbf{x}_i,\mathbf{x}_j)$$

Kernel PCA

- (1,1) (2,2)(3,3)
- For example, $k(x_1,x_2)=(1*2+1*2+1)^q$
- $k(x_1,x_3)=(1*3+1*3+1)^q$
- $k(x_2,x_3)=(2*3+2*3+1)^q$

$$K_{ij} = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) = k(\mathbf{x}_i, \mathbf{x}_j)$$

$$V = \sum_{i=1}^{l} \alpha_i \Phi(\mathbf{x}_i)$$

Applications – Kernel PCA

Kernel PCA Pattern Reconstruction via Approximate Pre-Images

B. Schölkopf, S. Mika, A. Smola, G. Rätsch, and K.-R. Müller.

In L. Niklasson, M. Bodén, and T. Ziemke, editors, Proceedings of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing, pages 147-152, Berlin, 1998. Springer Verlag.

Applications – Kernel PCA

- Input toy data:

3 point sources

(100 points each)

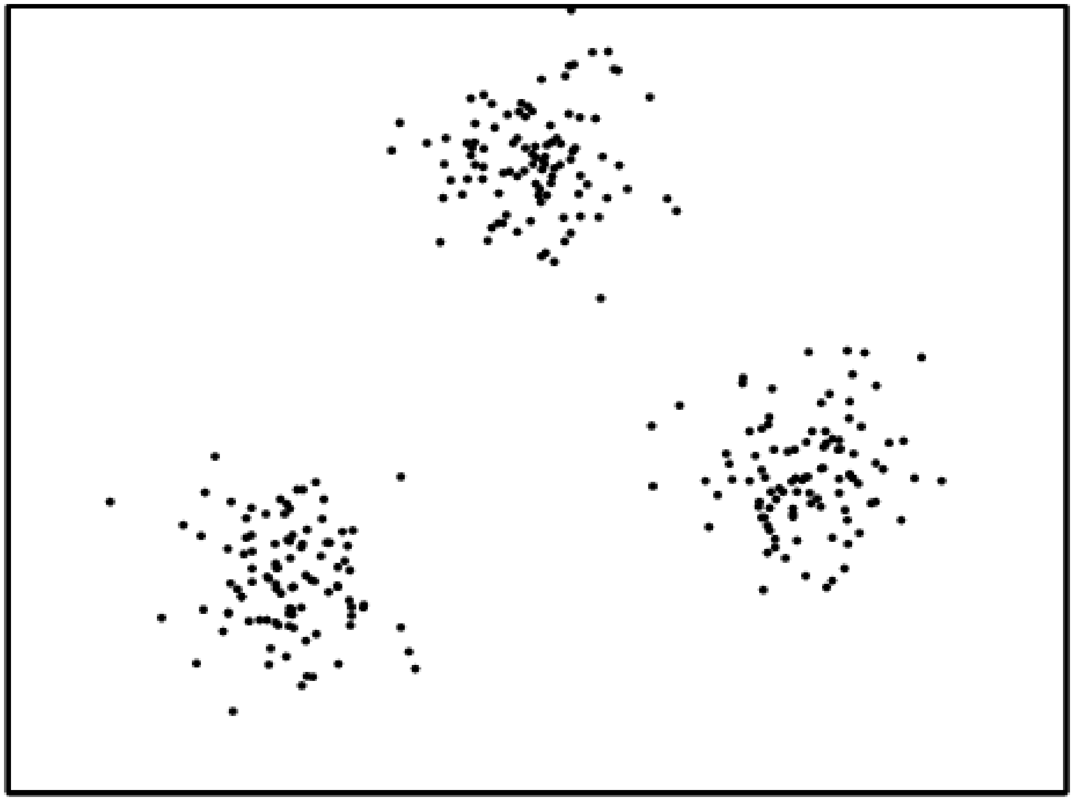
with Gaussian noise

σ=0.1

(-0.5,-0.1)

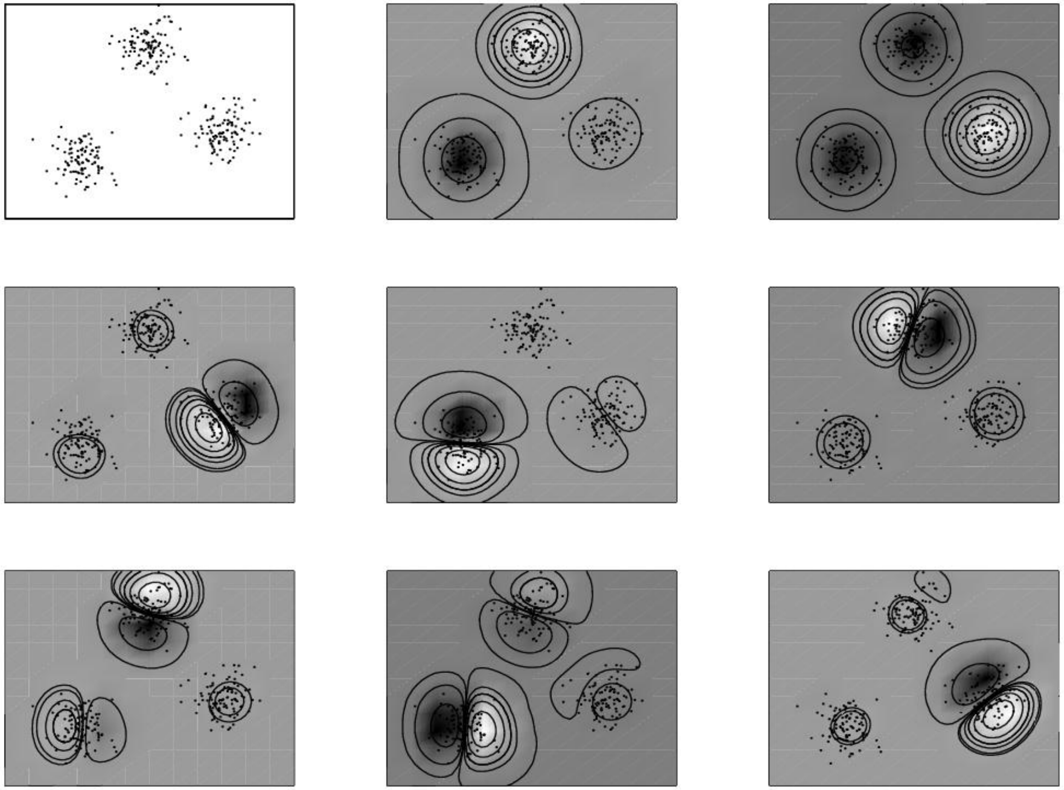
(0,0.7)

(0.5,0.1)

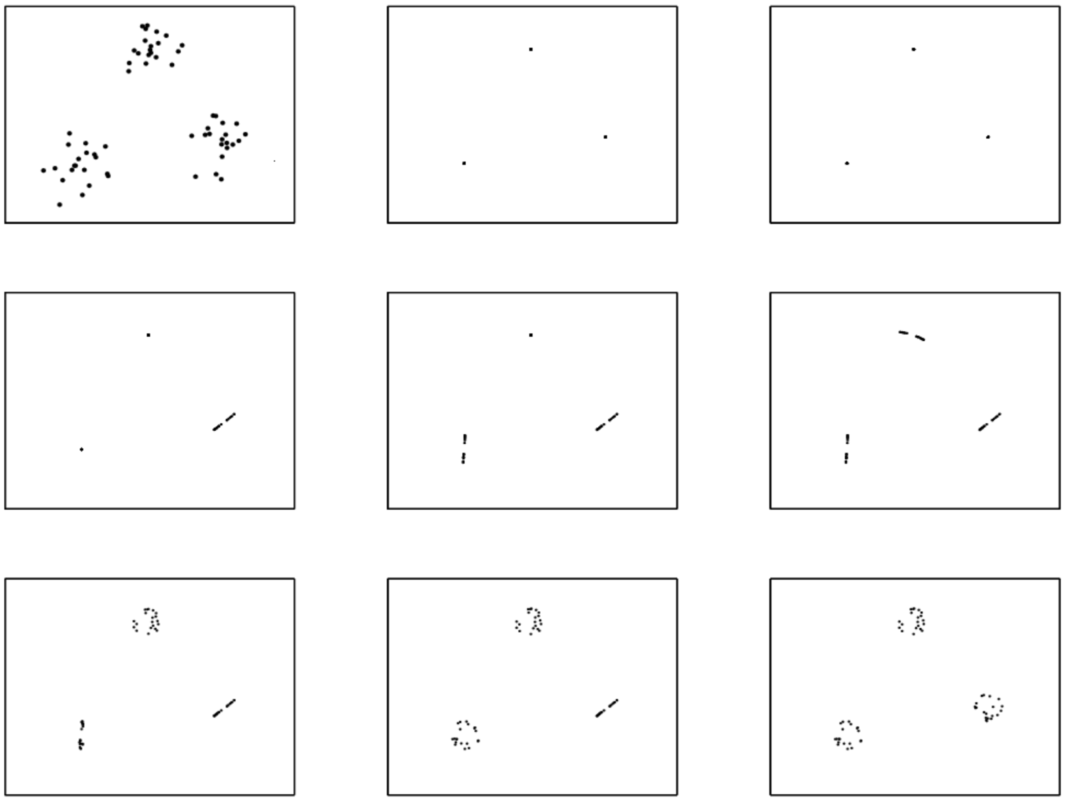


- Using RBF

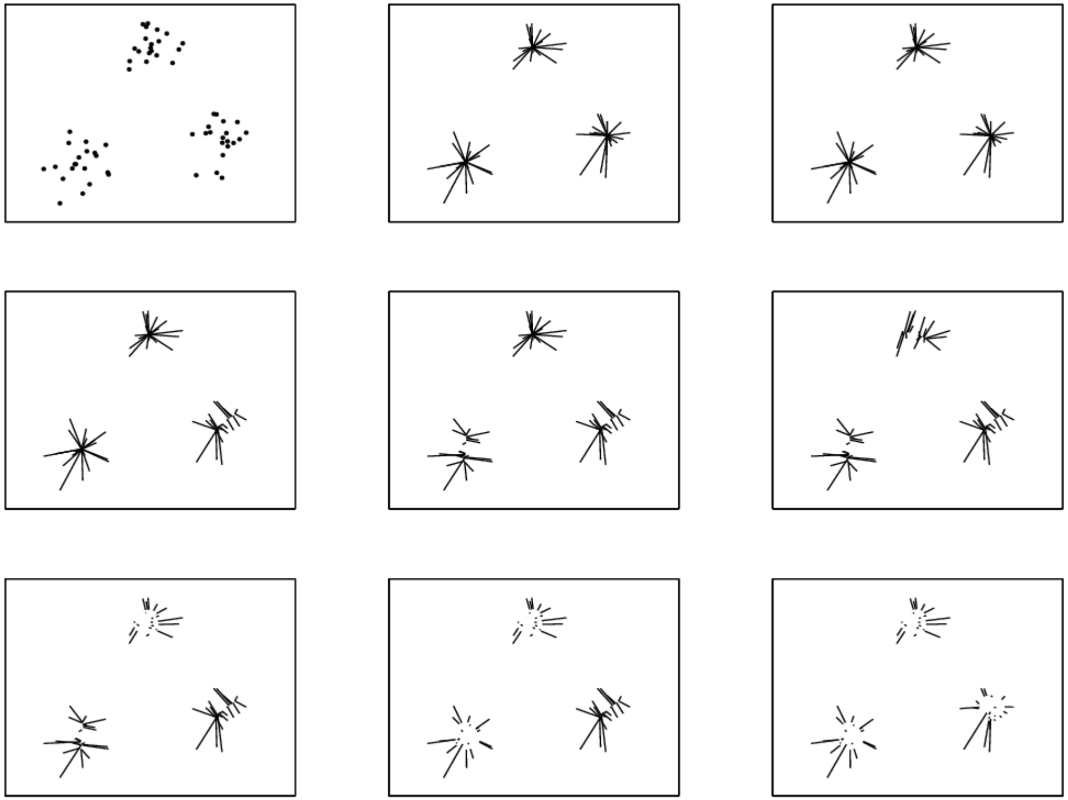
Applications – Kernel PCA



Applications – Kernel PCA

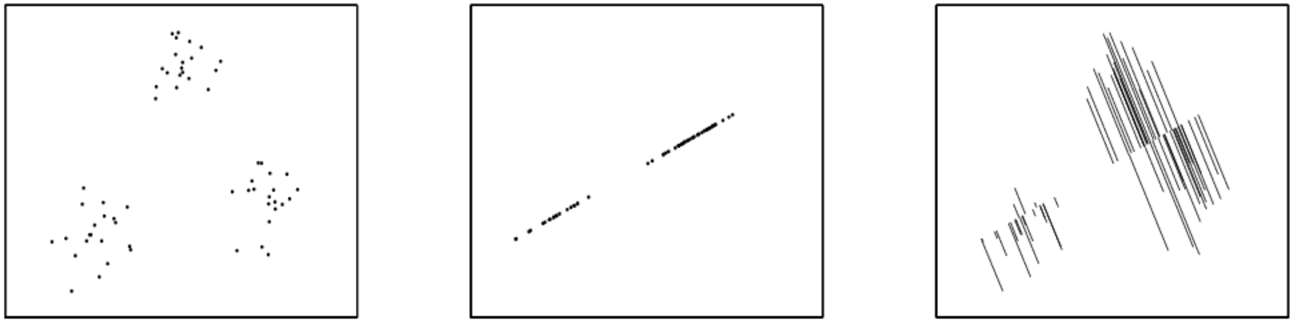


Applications - Kernel PCA



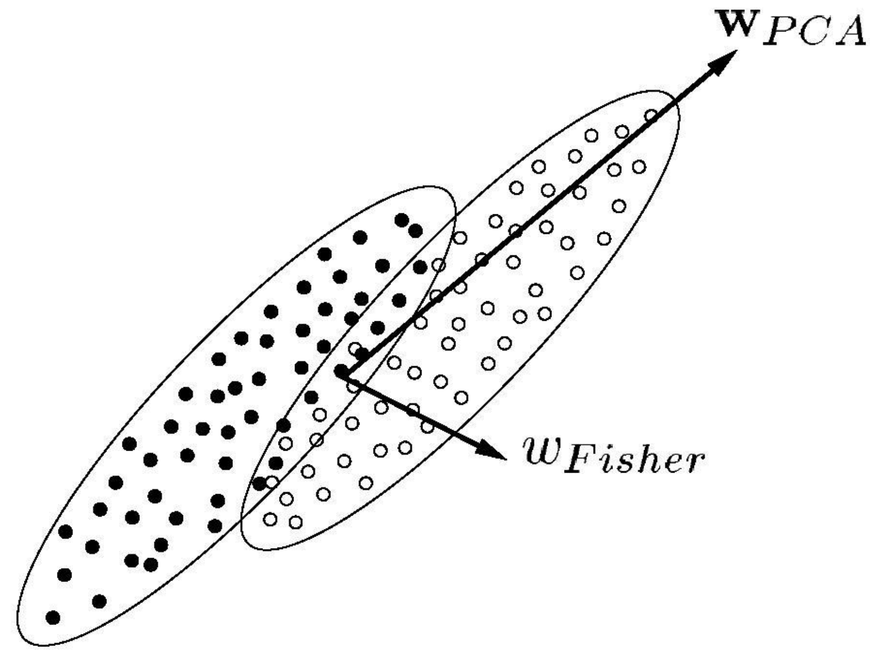
Applications – Kernel PCA

Compare to the linear PCA



Fisher Linear Discriminant

Finds a direction w, projected on which the classes are "best" separated



Fisher Linear Discriminant

- Equivalent to finding w which maximizes:

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

where

$$S_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T$$
$$S_W = \sum_{i=1,2} \sum_{\mathbf{x} \in \mathcal{X}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T$$

Kernel Fisher Discriminant

- Kernel forr

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

where

$$S_B^{\Phi} = (\mathbf{m}_1^{\Phi} - \mathbf{m}_2^{\Phi})(\mathbf{m}_1^{\Phi} - \mathbf{m}_2^{\Phi})^T$$

$$S_W^{\Phi} = \sum_{i=1,2} \sum_{\mathbf{x} \in \mathcal{X}_i} (\Phi(\mathbf{x}) - \mathbf{m}_i^{\Phi})(\Phi(\mathbf{x}) - \mathbf{m}_i^{\Phi})^T$$

$$\mathbf{m}_i^{\Phi} = \frac{1}{l_i} \sum_{j=1}^{l_i} \Phi(\mathbf{x}_i^j)$$

Kernel Fisher Discriminant

- From the theory of reproducing kernels:

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i \Phi(\mathbf{x}_i)$$

- Substituting it into the J(w) reduces the problem to maximizing:

$$J(W) = \frac{W^T K_b W}{W^T K_w W}$$

Kernel Fisher Discriminant

$$\mathbf{K}_w = \sum_{i=1}^{C} p(\varpi_i) E(\eta_j - m_i)(\eta_j - m_i)^T ,$$

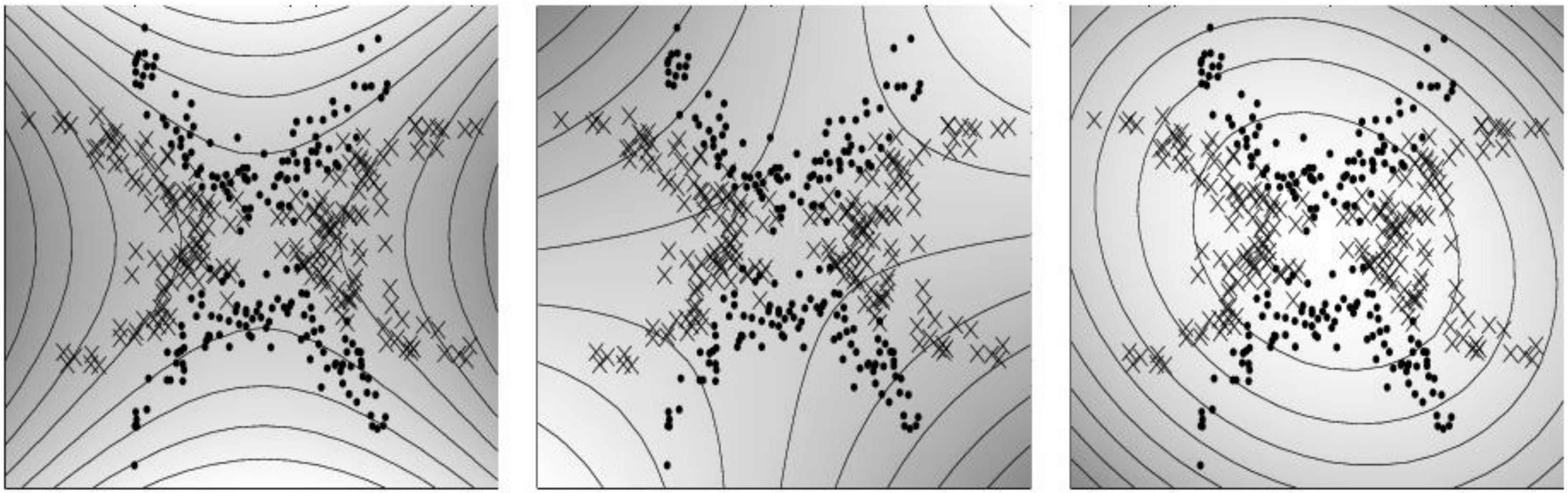
$$\mathbf{K}_b = \sum_{i=1}^{C} p(\varpi_i)(m_i - \bar{m})(m_i - \bar{m})^T ,$$

$$\eta_j = (k(x_1, x_j), k(x_2, x_j), \ldots, k(x_n, x_j))^T$$

$$m_i = (\frac{1}{n_i}\sum_{j=1}^{n_i} k(x_1, x_j), \frac{1}{n_i}\sum_{j=1}^{n_i} k(x_2, x_j), \ldots, \frac{1}{n_i}\sum_{j=1}^{n_i} k(x_n, x_j))^T$$

Details refer to b. zhang's cvpr 2005 paper

Kernel Fisher Discriminant – Toy Example



KFDA | **KPCA – 1^{st} eigenvector** | **KPCA – 2^{nd} eigenvector**

the feature value (indicated by grey level) and contour lines of identical feature value. Each class consists of two noisy parabolic shapes mirrored at the x and 9 axis respectively. We see, that the KFD feature discriminates the two classes in a nearly optimal way, whereas the Kernel PCA features, albeit describing interesting properties of the data set, do not separate the two classes well

Applications – Fisher Discriminant Analysis

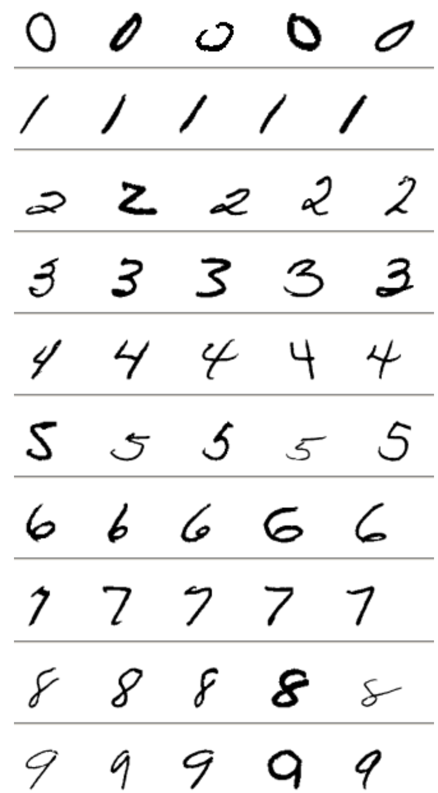
Fisher Discriminant Analysis with Kernels

S.Mika et. al.

In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, Neural Networks for Signal Processing IX, pages 41-48. IEEE, 1999.

Applications – Fisher Discriminant Analysis

- Input (USPS handwritten digits):
  - Training set: 3000



- Constructed:
  - 10 class/non-class KFD classifiers
  - Take the class with maximal output

Applications – Fisher Discriminant Analysis

- Results:
  3.7% error on a ten-class classifier
  Using RBF with $\sigma = 0.3*256$
- Compare to 4.2% using SVM
- KFDA vs. SVM

Summary…

- PCA
- Fisher Discriminant Analysis or LDA
- Kernel PCA
- Kernel Fisher Discriminate Analysis (KFDA)