

Component Analysis for PR & HS

- Computer Vision & Image Processing
 - Structure from motion.
 - Spectral graph methods for segmentation.
 - Appearance and shape models.
 - Fundamental matrix estimation and calibration.
 - Compression.
 - Classification.
 - Dimensionality reduction and visualization.
- Signal Processing
 - Spectral estimation, system identification (e.g. Kalman filter), sensor array processing (e.g. cocktail problem, echo cancellation), blind source separation, ...
- Computer Graphics
 - Compression (BRDF), synthesis, ...
- Speech, bioinformatics, combinatorial problems.

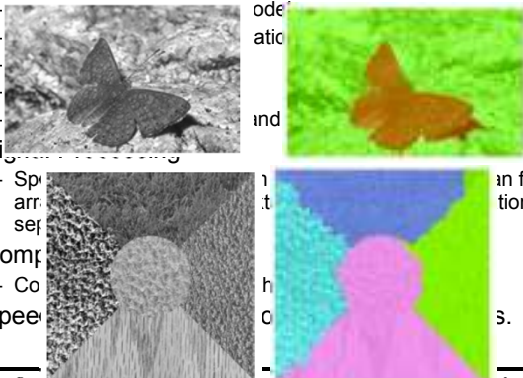
Component Analysis for PR & HS

- Computer Vision & Image Processing
 - **Structure from motion.**
 - Spectral graph methods for segmentation.
 - Appearance and shape models.
 - Fundamental matrix estimation and calibration.
 - Compression.
 - Classification.
 - Dimensionality reduction and visualization.
- Signal Processing
 - Spectral estimation, system identification (e.g. Kalman filter), sensor array processing (e.g. cocktail problem, echo cancellation), blind source separation, ...
- Computer Graphics
 - Compression (BRDF), synthesis, ...
- Speech, bioinformatics, combinatorial problems.



Component Analysis for PR & HS

- Computer Vision & Image Processing
 - Structure from motion.
 - **Spectral graph methods for segmentation.**
- Signal Processing
 - Spectral estimation, system identification (e.g. Kalman filter), sensor array processing (e.g. cocktail problem, echo cancellation), blind source separation, ...
- Computer Graphics
 - Compression (BRDF), synthesis, ...
- Speech, bioinformatics, combinatorial problems.



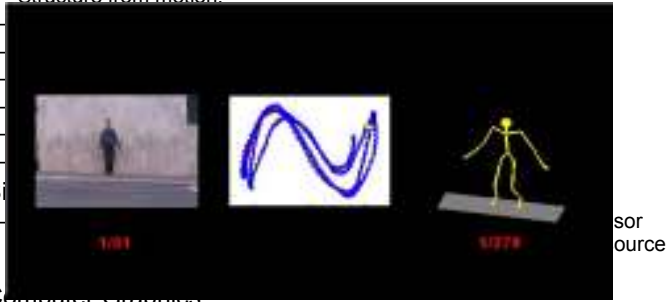
Component Analysis for PR & HS

- Computer Vision & Image Processing
 - Structure from motion.
 - Spectral graph methods for segmentation.
 - **Appearance and shape models.**
 - Fundamental matrix estimation and calibration.
 - Compression.
 - Classification.
 - Dimensionality reduction and visualization.
- Signal Processing
 - Spectral estimation, system identification (e.g. Kalman filter), sensor array processing (e.g. cocktail problem, echo cancellation), blind source separation, ...
- Computer Graphics
 - Compression (BRDF), synthesis, ...
- Speech, bioinformatics, combinatorial problems.



Component Analysis for PR & HS

- Computer Vision & Image Processing
 - Structure from motion.

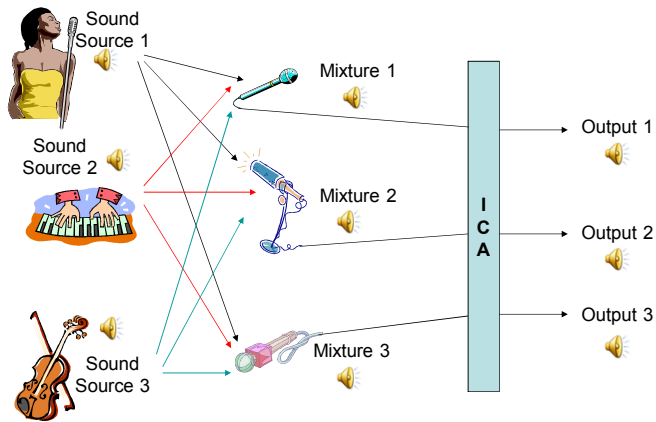


- Signal Processing
 - Spectral estimation, system identification (e.g. Kalman filter), sensor array processing (e.g. cocktail problem, echo cancellation), blind source separation, ...
- Computer Graphics
 - Compression (BRDF), synthesis, ...
- Speech, bioinformatics, combinatorial problems.

Component Analysis for PR & HS

- Computer Vision & Image Processing
 - Structure from motion.
 - Spectral graph methods for segmentation.
 - Appearance and shape models.
 - Fundamental matrix estimation and calibration.
 - Compression.
 - Classification.
 - Dimensionality reduction and visualization.
- Signal Processing
 - Spectral estimation, system identification (e.g. Kalman filter), sensor array processing (e.g. cocktail problem, echo cancellation), blind source separation, ...
- Computer Graphics
 - Compression (BRDF), synthesis, ...
- Speech, bioinformatics, combinatorial problems.

Independent Component Analysis (ICA)



Why CA for PR & HS?

- Learn from high dimensional data and few samples.
 - Useful for dimensionality reduction specially when functions are smooth.
- Natural geometric interpretation
- Easy to formulate, to solve and to extend
 - Non-linearities (Kernel methods) (Scholkopf & Smola,2002; Shawe-Taylor & Cristianini,2004)
 - Probabilistic (latent variable models) (Everitt,1984)
 - Multi-factorial (tensors) (Paatero & Tapper, 1994 ;O'Leary & Peleg,1983; Vasilescu & Terzopoulos,2002; Vasilescu & Terzopoulos,2003)
 - Exponential family PCA (Gordon,2002; Collins et al. 01)
- Efficient methods $O(d \times n)$
 - features d
 - samples n

Are CA methods popular/useful/used?

- About 28% of CVPR-07 papers use CA.
- Google:
 - Results 1 - 10 of about 1,870,000 for "[principal component analysis](#)".
 - Results 1 - 10 of about 506,000 for "[independent component analysis](#)".
 - Results 1 - 10 of about 273,000 for "[linear discriminant analysis](#)".
 - Results 1 - 10 of about 46,100 for "[negative matrix factorization](#)".
 - Results 1 - 10 of about 491,000 for "[kernel methods](#)".
- Still work to do
 - Results 1 - 10 of about 83,000 for "Spanish crisis"
 - Results 1 - 10 of about 287,000,000 for "[Britney Spears](#)"

Outline

- Introduction (15 min)
- **Generative models (40 min)**
 - (PCA, k-means, spectral clustering, NMF, ICA, MDS)
- Discriminative models (40 min)
 - (LDA, SVM, OCA, CCA)
- Standard extensions of linear models (30 min)
 - (Kernel methods, Latent variable models, Tensor factorization)
- Unified view (20 min)

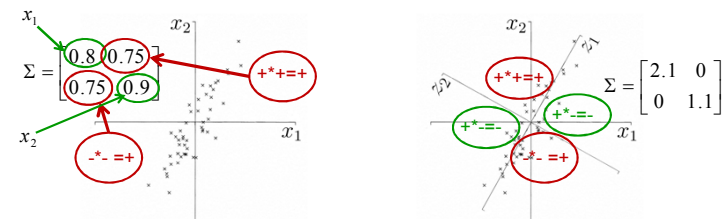
Generative models

$$D \approx BC$$

- Principal Component Analysis/Singular Value Decomposition
- Non-Negative Matrix Factorization
- Independent Component Analysis
- K-means and spectral clustering
- Multi-dimensional Scaling


Principal Component Analysis (PCA)

(Pearson, 1901; Hotelling, 1933; Mardia et al., 1979; Jolliffe, 1986; Diamantaras, 1996)



- PCA finds the directions of maximum variation of the data
- PCA decorrelates the original variables

PCA







$\text{pixels} \equiv p$

$\mathbf{n} = \text{images}$

$$\mathbf{D} = [\mathbf{d}_1 \ \mathbf{d}_2 \ \dots \ \mathbf{d}_n] \approx \mathbf{B}\mathbf{C} + \mu \mathbf{1}_n^T$$

$\mathbf{D} \in \mathfrak{R}^{d \times n}$ $\mathbf{B} \in \mathfrak{R}^{d \times k}$

$\mathbf{C} \in \mathfrak{R}^{k \times n}$ $\mu \in \mathfrak{R}^{d \times 1}$


 $\approx \mu$

 $+ c_1$

 $+ c_2$
 \dots
 $+ c_k$


• Assuming zero mean data, the basis \mathbf{B} that preserve the maximum variation of the signal is given by the eigenvectors of $\mathbf{D}\mathbf{D}^T$.

$$d \mid \mathbf{D}\mathbf{D}^T \mathbf{B} = \mathbf{B}\mathbf{\Lambda}$$

d=pixels

Snap-shot method & SVD

- If $d \gg n$ (e.g., images 100×100 vs. 300 samples) no $\mathbf{D}\mathbf{D}^T$.
- $\mathbf{D}\mathbf{D}^T$ and $\mathbf{D}^T\mathbf{D}$ have the same eigenvalues (energy) and related eigenvectors (by \mathbf{D}).
- \mathbf{B} is a linear combination of the data! $\mathbf{B} = \mathbf{D}\boldsymbol{\alpha}$ (Sirovich, 1987)

$$\mathbf{D}\mathbf{D}^T \mathbf{B} = \mathbf{B}\mathbf{\Lambda} \quad \cancel{\mathbf{D}^T \mathbf{D} \mathbf{D} \boldsymbol{\alpha}} = \cancel{\mathbf{D}^T} \mathbf{D} \boldsymbol{\alpha} \mathbf{\Lambda}$$

- $[\boldsymbol{\alpha}, \mathbf{L}] = \text{eig}(\mathbf{D}^T\mathbf{D})$ $\mathbf{B} = \mathbf{D} \boldsymbol{\alpha} (\text{diag}(\text{diag}(\mathbf{L})))^{-0.5}$
- SVD factorizes the data matrix \mathbf{D} as: $\mathbf{D}\mathbf{D}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$

$$\mathbf{D} = \mathbf{B}\mathbf{C}$$

$$\mathbf{B} \in \mathfrak{R}^{d \times k} \quad \mathbf{C} \in \mathfrak{R}^{k \times n}$$

$$\mathbf{B}^T \mathbf{B} = \mathbf{I} \quad \mathbf{C}\mathbf{C}^T = \mathbf{\Lambda}$$

PCA

$$\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

$$\mathbf{D}^T \mathbf{D} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

$$\mathbf{U} \in \mathfrak{R}^{d \times k} \quad \mathbf{\Sigma} \in \mathfrak{R}^{k \times n} \quad \mathbf{V} \in \mathfrak{R}^{n \times n}$$

$$\mathbf{U}^T \mathbf{U} = \mathbf{I} \quad \mathbf{V}^T \mathbf{V} = \mathbf{I} \quad \mathbf{\Sigma} \text{ diagonal}$$

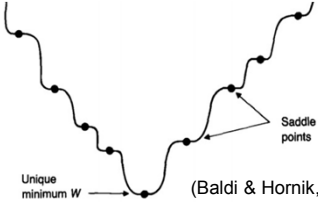
SVD

Error function for PCA

- PCA minimizes the following function:
(Eckardt & Young, 1936; Gabriel & Zamir, 1979; Baldi & Hornik, 1989; Shum et al., 1995; De la Torre & Black, 2003a)

$$E_1(\mathbf{B}, \mathbf{C}) = \sum_{i=1}^n \|\mathbf{d}_i - \mathbf{B}\mathbf{c}_i\|_2^2 = \|\mathbf{D} - \mathbf{B}\mathbf{C}\|_F^2$$

- Not unique solution: $\mathbf{B}\mathbf{R}\mathbf{R}^{-1}\mathbf{C} = \mathbf{B}\mathbf{C}$ $\mathbf{R} \in \mathfrak{R}^{k \times k}$ (De la Torre 2012)



Unique minimum \mathbf{W} Saddle points

(Baldi & Hornik, 1989)

PCA/SVD in Computer Vision

- PCA/SVD has been applied to:
 - Recognition (eigenfaces: Turk & Pentland, 1991; Sirovich & Kirby, 1987; Leonardis & Bischof, 2000; Gong et al., 2000; McKenna et al., 1997a)
 - Parameterized motion models (Yacoob & Black, 1999; Black et al., 2000; Black, 1999; Black & Jepson, 1998)
 - Appearance/shape models (Cootes & Taylor, 2001; Cootes et al., 1998; Pentland et al., 1994; Jones & Poggio, 1998; Casia & Sclaroff, 1999; Black & Jepson, 1998; Blanz & Vetter, 1999; Cootes et al., 1995; McKenna et al., 1997; de la Torre et al., 1998b; de la Torre et al., 1998b)
 - Dynamic appearance models (Soatto et al., 2001; Rao, 1997; Orriols & Binfa, 2001; Gong et al., 2000)
 - Structure from Motion (Tomasi & Kanade, 1992; Bregler et al., 2000; Sturm & Triggs, 1996; Brand, 2001)
 - Illumination based reconstruction (Hayakawa, 1994)
 - Visual servoing (Murase & Nayar, 1995; Murase & Nayar, 1994)
 - Visual correspondence (Zhang et al., 1995; Jones & Malik, 1992)
 - Camera motion estimation (Hartley, 1992; Hartley & Zisserman, 2000)
 - Image watermarking (Liu & Tan, 2000)
 - Signal processing (Moonen & de Moor, 1995)
 - Neural approaches (Oja, 1982; Sanger, 1989; Xu, 1993)
 - Bilinear models (Tenenbaum & Freeman, 2000; Marimont & Wandell, 1992)
 - Direct extensions (Welling et al., 2003; Penev & Atick, 1996)

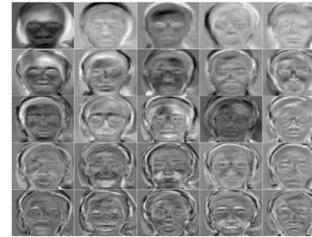
Generative models

$$\mathbf{D} \approx \mathbf{BC}$$

- Principal Component Analysis/Singular Value Decomposition
- Non-Negative Matrix Factorization
- Independent Component Analysis
- K-means and spectral clustering
- Multi-dimensional Scaling

“Intercorrelations among variables are the bane of the multivariate researcher’s struggle for meaning”

Cooley and Lohnes, 1971



Part-based representation

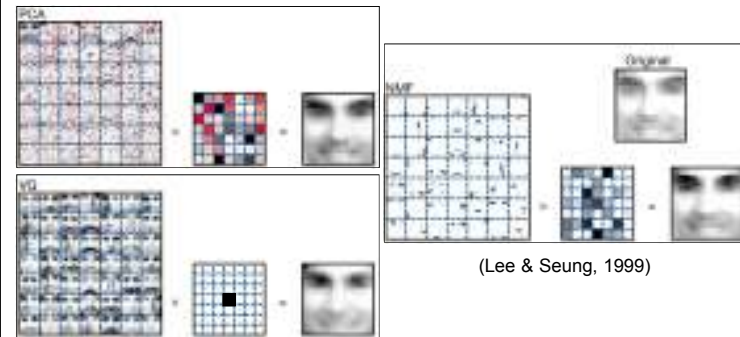


- The firing rates of neurons are never negative
- Independent representations

NMF & ICA

Non-negative Matrix Factorization (NMF)

- Positive factorization.
$$E(\mathbf{B}, \mathbf{C}) = \|\mathbf{D} - \mathbf{BC}\|_F \quad \mathbf{B}, \mathbf{C} \geq 0$$
- Leads to part-based representation.



NMF

(Lee & Seung, 1999; Lee & Seung, 2000)

$$\min_{\mathbf{B} \geq 0, \mathbf{C} \geq 0} F = \sum_{ij} |d_{ij} - (\mathbf{BC})_{ij}|^2$$

Derivatives:

$$\frac{\partial F}{\partial \mathbf{C}_{ij}} = (\mathbf{B}^T \mathbf{BC})_{ij} - (\mathbf{B}^T \mathbf{C})_{ij}$$

$$\frac{\partial F}{\partial \mathbf{B}_{ij}} = (\mathbf{BCC}^T)_{ij} - (\mathbf{DC}^T)_{ij}$$

Inference:

$$\mathbf{C}_{ij} \leftarrow \mathbf{C}_{ij} \frac{(\mathbf{B}^T \mathbf{D})_{ij}}{(\mathbf{B}^T \mathbf{B} \mathbf{C})_{ij}}$$

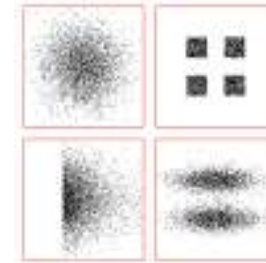
Learning:

$$\mathbf{B}_{ij} \leftarrow \mathbf{B}_{ij} \frac{(\mathbf{DC}^T)_{ij}}{(\mathbf{BCC}^T)_{ij}}$$

- Multiplicative algorithm can be interpreted as diagonally rescaled gradient descent

Independent Component Analysis (ICA)

- We need more than second order statistics to represent the signal

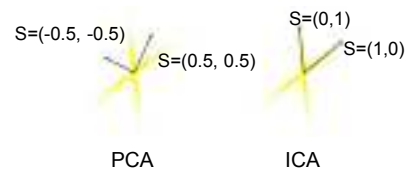


ICA

(Hyvriinen et al., 2001)

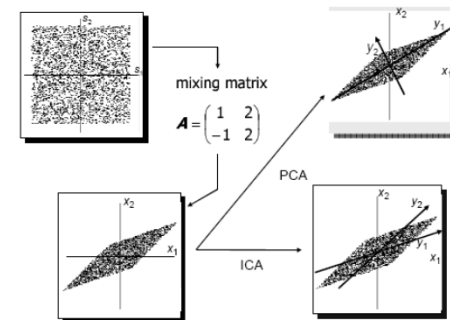
$$\mathbf{D} = \mathbf{BC} \quad \mathbf{C} \approx \mathbf{S} = \mathbf{WD} \quad \mathbf{W} \approx \mathbf{B}^{-1}$$

- Look for s_i that are independent
- PCA finds uncorrelated variables
- For Gaussian distributions independence and uncorrelated is the same
- Uncorrelated $E(s_i s_j) = E(s_i)E(s_j)$
- Independent $E(g(s_i)f(s_j)) = E(g(s_i))E(f(s_j))$ for any non-linear f, g



ICA vs PCA

(Hyvriinen et al., 2001)



Many optimization criteria

$$\mathbf{S} = \mathbf{W}\mathbf{D}$$

- Minimize high order moments: e.g. kurtosis

$$\text{kurt}(\mathbf{W}) = E\{s^4\} - 3(E\{s^2\})^2$$

- Many other information criteria.

- Also an error function: (Olhausen & Field, 1996)

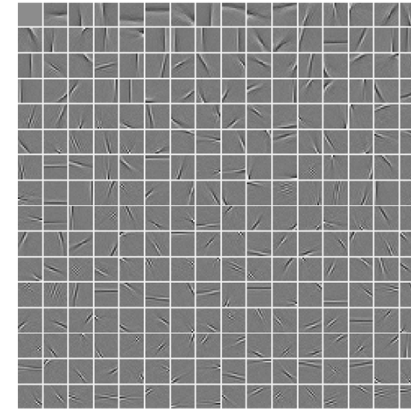
$$\sum_{i=1}^n \|\mathbf{d}_i - \mathbf{B}\mathbf{c}_i\| + \sum_{i=1}^n S(\mathbf{c}_i)$$

Sparseness (e.g. $S=|\cdot|$)

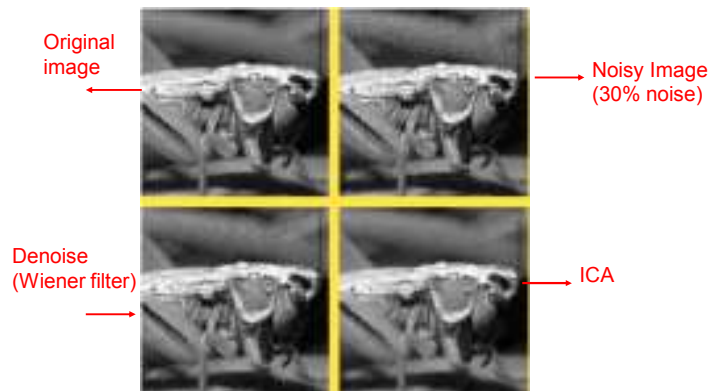
- Other sparse PCA.

(Chennubhotla & Jepson, 2001b; Zou et al., 2005; dAspremont et al., 2004;)

Basis of natural images



Denoising



Generative models

$$\mathbf{D} \approx \mathbf{B}\mathbf{C}$$

- Principal Component Analysis/Singular Value Decomposition
- Non-Negative Matrix Factorization
- Independent Component Analysis
- K-means and spectral clustering
- Multi-dimensional Scaling

The clustering problem

- Partition the data set in c -disjoint "clusters" of data points.



- Number of possible partitions

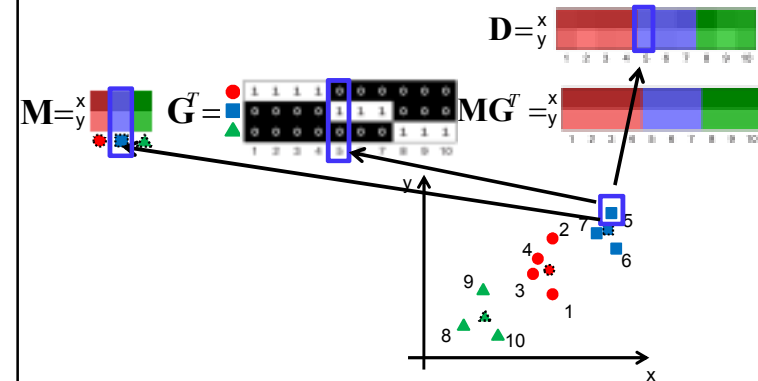
$$S(n, c) = \frac{1}{c} \sum_{i=1}^c (-1)^i \binom{c}{i} i^n \quad \begin{cases} n = 21 \\ c = 4 \end{cases} \approx 10^{12}$$

- NP-hard and approximate algorithms (k-means, hierarchical clustering, mog, ...)

K-means

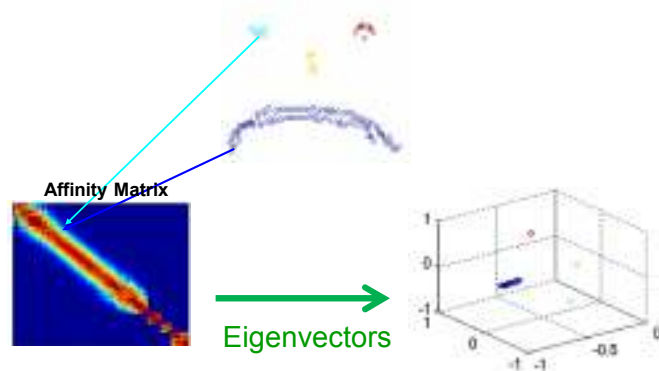
(Ding et al., '02, Torre et al '06)

$$E_0(\mathbf{M}, \mathbf{G}) = \|\mathbf{D} - \mathbf{M}\mathbf{G}^T\|_F$$



Spectral Clustering

(Dhillon et al., '04, Zass & Shashua, 2005; Ding et al., 2005, De la Torre et al '06)



Generative models

$$\mathbf{D} \approx \mathbf{B}\mathbf{C}$$

- Principal Component Analysis/Singular Value Decomposition
- Non-Negative Matrix Factorization
- Independent Component Analysis
- K-means and spectral clustering
- Multi-dimensional Scaling

Multi-Dimensional Scaling (MDS)

- MDS takes a matrix of pair-wise distances and finds an embedding that preserves the inter-point distances.

Pair-wise distances of US cities

CHICAGO	0				
MIAMI	144	0			
SEATTLE	161	168	0		
DENVER	172	230	148	0	
PHOENIX	183	238	208	144	0
LOS ANGELES	194	251	218	208	144

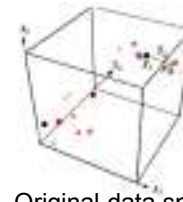
Spatial layout of cities in an embedded space



MDS (II)

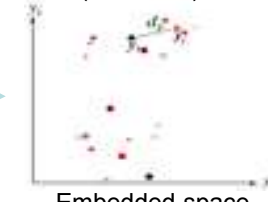
$$\min_{\{y_1, \dots, y_n\}} \sum_i \sum_j [\delta_{ij} - f(d_{ij})]^2$$

Pair-wise distances in original data space (Given)



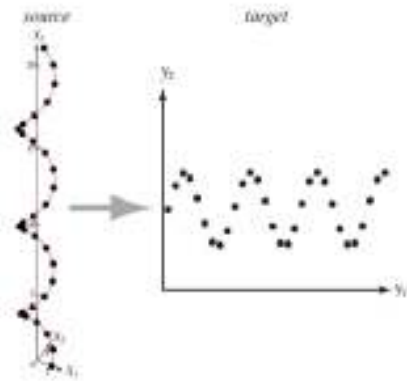
Original data space

Pair-wise distances in the embedded space (Unknown)



Embedded space

MDS (III)



Outline

- Introduction (15 min)
- Generative models (40 min)
 - (PCA, k-means, spectral clustering, NMF, ICA, MDS)
- Discriminative models (40 min)
 - (LDA, SVM, OCA, CCA)
- Standard extensions of linear models (30 min)
 - (Kernel methods, Latent variable models, Tensor factorization)
- Unified view (20 min)

Discriminative models

- Linear Discriminant Analysis (LDA)
- Support Vector Machines (SVM)
- Oriented Component Analysis (OCA)
- Canonical Correlation Analysis (CCA)

Linear Discriminant Analysis (LDA)

(Fisher, 1938; Mardia et al., 1979; Bishop, 1995)

$$S_b = \sum_{i=1}^C \sum_{j=1}^C (\mu_i - \mu_j)(\mu_i - \mu_j)^T$$

$$S_i = DD^T = \sum_{i=1}^n d_i d_i^T$$

$$J(B) = \frac{|B^T S_b B|}{|B^T S_i B|}$$

$$S_b B = S_i B A$$

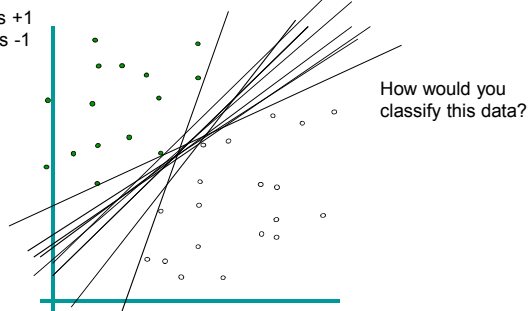
$$S_w = \sum_{j=1}^C \sum_{i=1}^{C_j} (d_i - \mu_j)(d_i - \mu_j)^T$$

- Optimal linear dimensionality reduction if classes are Gaussian with equal covariance matrix.

Support Vector Machines (SVM)

- Linear classifier $f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$

- ● denotes +1
- ○ denotes -1



- Infinite amount lines classify the data well, but which is the best?

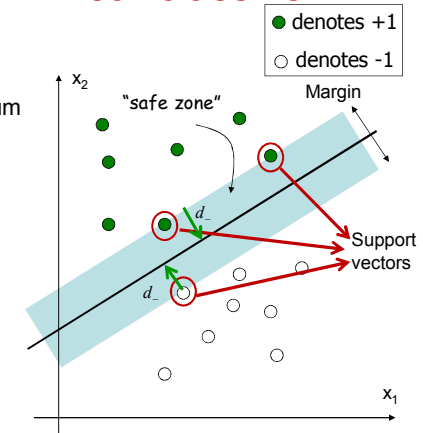
Large margin linear classifier

- The linear discriminant function with the maximum margin is the best

- Why is the best?
 - Robust to outliers and shown to improve generalization

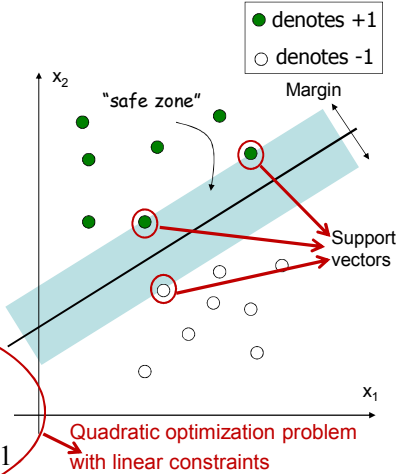
- The margin with is:

$$M = d_- + d_+ = \frac{2}{\|\mathbf{w}\|_2}$$



Large margin linear classifier

- The linear discriminant function with the maximum margin is the best
- Why is the best?
 - Robust to outliers and shown to improve generalization
- The margin with is:



$$\min \|\mathbf{w}\|_2^2 \text{ s.t.}$$

$$\text{for } y_i = +1 \quad \mathbf{w}^T \mathbf{x}_i + b \geq 1$$

$$\text{for } y_i = -1 \quad \mathbf{w}^T \mathbf{x}_i + b \leq -1$$

Quadratic optimization problem with linear constraints

SVM formulation

- But what if we have error or non-linear decision boundaries.
- Slack variables ξ_i can be added to allow misclassification of difficult or noisy data points

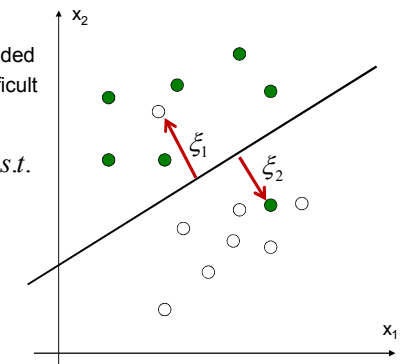
$$\min \|\mathbf{w}\|_2^2 + C \sum_{i=1}^n \xi_i \text{ s.t.}$$

$$y_i = +1 \quad \mathbf{w}^T \mathbf{x}_i + b \geq 1 - \xi_i$$

$$y_i = -1 \quad \mathbf{w}^T \mathbf{x}_i + b \leq -1 + \xi_i$$

$$\xi_i \geq 0$$

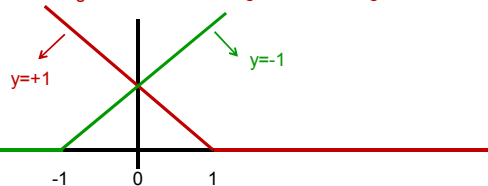
Balance trade-off between margin and classification error. Controls over-fitting



SVM is a regularized network

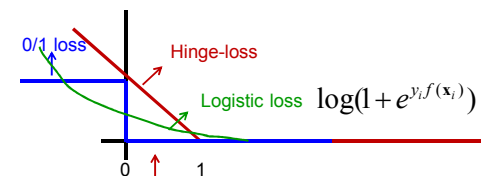
- In the noiseless case, LDA on the support vectors is equivalent to SVM (Shashua 99)
- SVM classifier can be also optimized in the primal without constraints:

$$\min \underbrace{\|\mathbf{w}\|_2^2}_{\text{Regularization}} + C \sum_{i=1}^N \underbrace{\max(0, 1 - y_i(\mathbf{x}_0 + \mathbf{x}_i^T \mathbf{w}))}_{\text{Training error with hinge-loss function}}$$



Other classifiers

- Several other loss-functions for other classifiers (e.g., logistic regression, Adaboost)

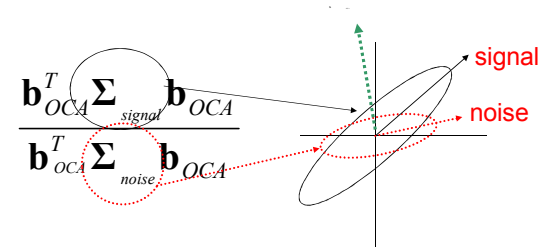


Penalize examples that are well classified but close to the margin

Discriminative Models

- Linear Discriminant Analysis (LDA)
- Support Vector Machines (SVM)
- **Oriented Component Analysis (OCA)**
- Canonical Correlation Analysis (CCA)

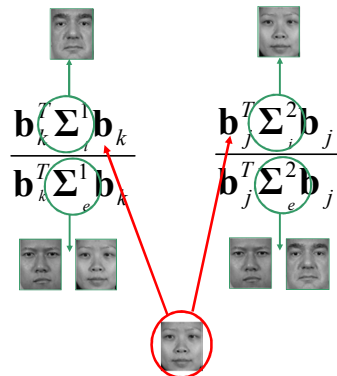
Oriented Component Analysis (OCA)



- Generalized eigenvalue problem: $\Sigma_i \mathbf{b}_k = \Sigma_e \mathbf{b}_k \lambda$
- \mathbf{b}_{oca} is steered by the distribution of noise.

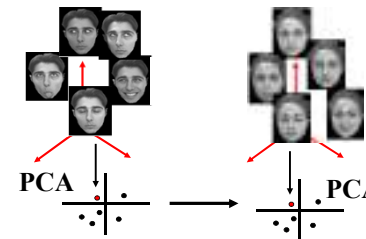
OCA for face recognition

(De la Torre et al. 2005)



Canonical Correlation Analysis (CCA)

- Perform PCA independently and learn a mapping



- Independent dimensionality reduction between set can loose signals with small energy but highly correlated

Canonical Correlation Analysis (CCA)

(Mardia et al., 1979; Borga 98)

- Learn relations between multiple data sets? (e.g. find features in one set related to another data set)
- Given two sets $X \in \mathbb{R}^{d_1 \times n}$ and $Y \in \mathbb{R}^{d_2 \times n}$, CCA finds the pair of directions w_x and w_y that maximize the correlation between the projections (assume zero mean data)

$$\rho = \frac{w_x^T X^T Y w_y}{\sqrt{w_x^T X^T X w_x w_y^T Y^T Y w_y}}$$

- Several ways of optimizing it:

$$A = \begin{bmatrix} 0 & X^T Y \\ X^T Y & 0 \end{bmatrix} \in \mathbb{R}^{(d_1+d_2) \times (d_1+d_2)}, \quad B = \begin{bmatrix} X^T X & 0 \\ 0 & Y^T Y \end{bmatrix} \in \mathbb{R}^{(d_1+d_2) \times (d_1+d_2)} \quad w = \begin{bmatrix} w_x \\ w_y \end{bmatrix}$$

- An stationary point of r is the solution to CCA.

$$Aw = \lambda Bw$$

Virtual avatars with CCA

(De la Torre & Black 2001)

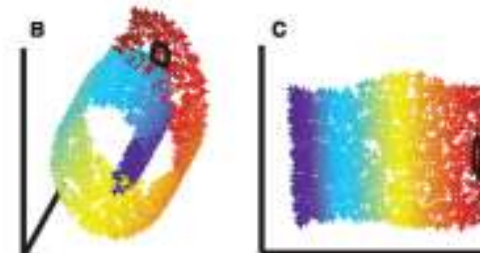


Outline

- Introduction (15 min)
- Generative models (40 min)
 - (PCA, k-means, spectral clustering, NMF, ICA, MDS)
- Discriminative models (40 min)
 - (LDA, SVM, OCA, CCA)
- **Standard extensions of linear models (30 min)**
 - (Kernel methods, Latent variable models, Tensor factorization)
- Unified view (20 min)

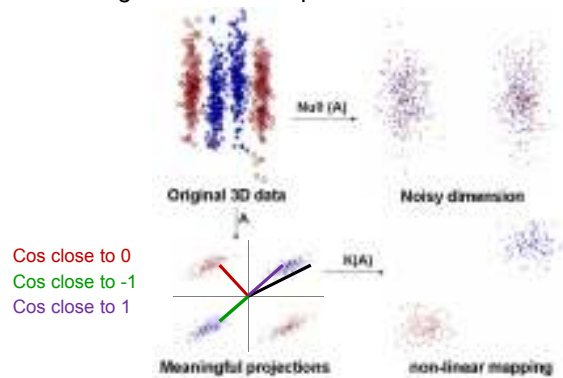
Linear methods fail

- Data points in a non-linear manifold
- There is no good linear mapping to map to a plane
- Linear methods only rotate/translate/scale the data



Linear methods fail

- Learning a non-linear representation for classification

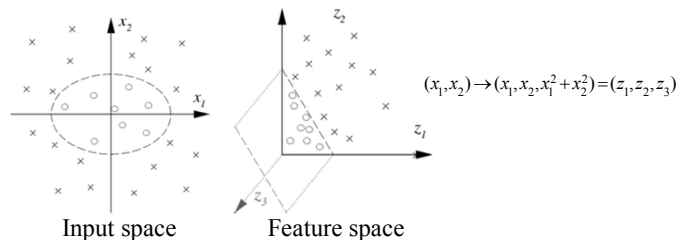


Linear methods not enough

- Linear methods:
 - Unique optimal solutions
 - Fast learning algorithms
 - Better statistical analysis
- Problem:
 - Insufficient capacity. Minsky and Pappert pointed out in their books Perceptrons
 - Neural networks adding non-linear layers (e.g., MLP). Solve the capacity problem but **hard to train and local minima**.
- Kernel methods:
 - Use linear techniques but work in a high-dimensional space.

$$\mathbf{x} \rightarrow \Phi(\mathbf{x})$$

Kernel methods



- The kernel defines an implicit mapping (usually high dimensional) from the input to feature space, so the data becomes linearly separable.
- Computation in the feature space can be costly because it is (usually) high dimensional
 - The feature space can be infinite-dimensional!

Kernel methods (II)

- Suppose $\phi(\cdot)$ is given as follows

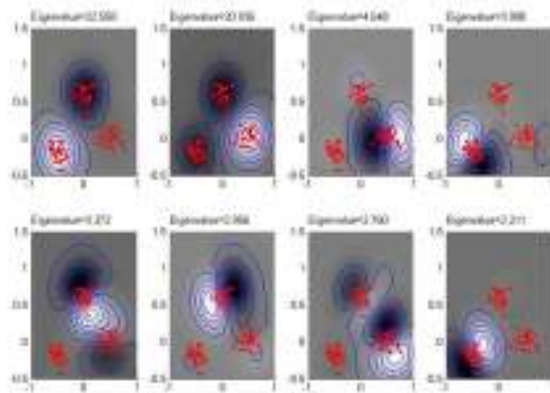
$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$
- An inner product in the feature space is

$$\langle \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right), \phi\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) \rangle = (1 + x_1y_1 + x_2y_2)^2$$
- So, if we define the kernel function as follows, there is no need to carry out $\phi(\cdot)$ explicitly

$$K(\mathbf{x}, \mathbf{y}) = (1 + x_1y_1 + x_2y_2)^2$$
- This use of kernel function to avoid carrying out $\phi(\cdot)$ explicitly is known as the **kernel trick**. In any linear algorithm that can be expressed by inner products can be made nonlinear by going to the feature space

Kernel PCA

(Scholkopf et al., 1998)



Kernel PCA (II)

(Scholkopf et al., 1998)

- Eigenvectors of the cov. Matrix in feature space.

$$\bar{\mathbf{C}} = \frac{1}{n} \sum_{i=1}^n \Phi(\mathbf{d}_i) \Phi(\mathbf{d}_i)^T \quad \bar{\mathbf{C}} \mathbf{b}_1 = \mathbf{b}_1 \lambda$$

- Eigenvectors lie in the span of data in feature space.

$$\mathbf{b}_1 = \sum_{i=1}^n \alpha_i \Phi(\mathbf{d}_i)$$

$$\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \Phi(\mathbf{d}_i) K(\mathbf{d}_i, \mathbf{d}_j) = \left[\sum_{i=1}^n \alpha_i \Phi(\mathbf{d}_i) \right] \lambda$$

$$\mathbf{K} \boldsymbol{\alpha} = \boldsymbol{\alpha} \lambda$$

Outline

- Introduction (5 min)
- Generative models (20 min)
 - (PCA, k-means, spectral clustering, NMF, ICA, MDS)
- Discriminative models (20 min)
 - (LDA, SVM, OCA, CCA)
- Standard extensions of linear models (15 min)
 - (Kernel methods, Latent variable models, Tensor factorization)
- Unified view (15 min)
- Extended generative models (50 min)
 - RPCA, PaCA, ACA
- Extended discriminative models (1 hour)
 - MODA, Parda, CTW, seg-SVM

Factor Analysis

- A Gaussian distribution on the coefficients and noise is added to PCA \rightarrow Factor Analysis. (Mardia et al., 1979)

$$\mathbf{d} = \boldsymbol{\mu} + \mathbf{B}\mathbf{c} + \boldsymbol{\eta}$$

$$p(\mathbf{c}) = N(\mathbf{c} | \mathbf{0}, \mathbf{I}_k) \quad p(\mathbf{d} | \mathbf{c}, \mathbf{B}) = N(\mathbf{d} | \boldsymbol{\mu} + \mathbf{B}\mathbf{c}, \Psi)$$

$$p(\boldsymbol{\eta}) = N(\boldsymbol{\eta} | \mathbf{0}, \Psi) \quad \Psi = \text{diag}(\eta_1, \eta_2, \dots, \eta_d)$$

$$\text{cov}(\mathbf{d}) = E((\mathbf{d} - \boldsymbol{\mu})(\mathbf{d} - \boldsymbol{\mu})^T) = \mathbf{B}\mathbf{B}^T + \Psi$$

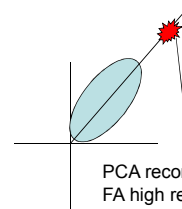
- Inference (Roweis & Ghahramani, 1999; Tipping & Bishop, 1999a)

$$p(\mathbf{c}, \mathbf{d}) \text{ Jointly Gaussian}$$

$$p(\mathbf{c} | \mathbf{d}) = N(\mathbf{c} | \mathbf{m}, \mathbf{V})$$

$$\mathbf{m} = \mathbf{B}^T (\mathbf{B}\mathbf{B}^T + \Psi)^{-1} (\mathbf{d} - \boldsymbol{\mu})$$

$$\mathbf{V} = (\mathbf{I} + \mathbf{B}^T \Psi^{-1} \mathbf{B})^{-1}$$



PCA reconstruction low error.

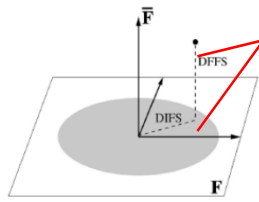
FA high reconstruction error (low likelihood).

Ppca

- If $\Psi = E(\eta\eta^T) = \delta I_d$ PPCA.
- If $\varepsilon \rightarrow 0$ is equivalent to PCA. $\varepsilon \rightarrow 0 \quad \mathbf{B}^T(\mathbf{B}\mathbf{B}^T + \Psi)^{-1} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T$
- Probabilistic visual learning (Moghaddam & Pentland, 1997;)

$$p(\mathbf{d}) = \int p(\mathbf{d} | \mathbf{c}) p(\mathbf{c}) d\mathbf{c} = \frac{e^{-\frac{1}{2}(\mathbf{d}-\mu)^T \Sigma^{-1}(\mathbf{d}-\mu)}}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} = \frac{e^{-\frac{1}{2}(\mathbf{d}-\mu)^T (\mathbf{B}\mathbf{B}^T + \delta \mathbf{I})^{-1}(\mathbf{d}-\mu)}}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} = \frac{e^{-\frac{1}{2} \sum_{i=1}^k \frac{c_i^2}{\lambda_i}}}{(2\pi)^{\frac{d}{2}} \prod_{i=1}^k \lambda_i^{1/2}} \left[\frac{e^{-\frac{\varepsilon^2(\mathbf{d})}{2\rho}}}{(2\pi\rho)^{\frac{(d-k)}{2}}} \right]$$

$$\mathbf{c}_i = \mathbf{B}^T \mathbf{d}_i$$



More on PPCA

- Extension to mixtures of Ppca (mixture of subspaces). (Tipping & Bishop, 1999b; Black et al., 1998; Jebara et al., 1998)
- Tracking (Yang et al., 1999; Yang et al., 2000a; Lee et al., 2005; de la Torre et al., 2000b)
- Recognition/Detection (Moghaddam et al., 2000; Shakhnarovich & Moghaddam, 2004; Everingham & Zisserman, 2006)
- PCA for the exponential family (collins et al., 2001)

Outline

- Introduction (5 min)
- Generative models (20 min)
 - (PCA, k-means, spectral clustering, NMF, ICA, MDS)
- Discriminative models (20 min)
 - (LDA, SVM, OCA, CCA)
- **Standard extensions of linear models (15 min)**
 - (Kernel methods, Latent variable models, **Tensor factorization**)
- Unified view (15 min)
- Extended generative models (50 min)
 - RPCA, PaCA, ACA
- Extended discriminative models (1 hour)
 - MODA, Parda, CTW, seg-SVM

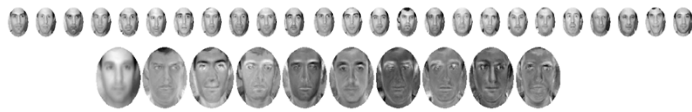
Tensor faces

(Vasilescu & Terzopoulos, 2002; Vasilescu & Terzopoulos, 2003)



Eigenfaces

- Facial images (identity change)



- Eigenfaces bases vectors capture the variability in facial appearance (do not decouple pose, illumination, ...)



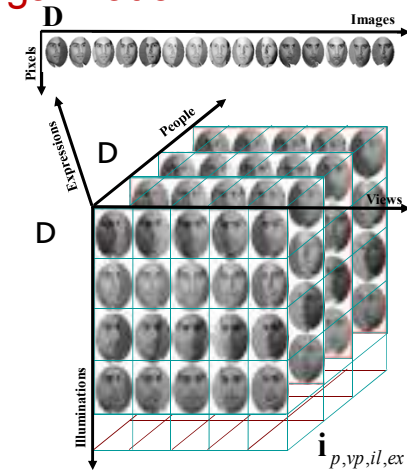
Data Organization

Linear/PCA: Data Matrix

- $\mathbb{R}^{\text{pixels} \times \text{images}}$
- a matrix of image vectors

Multilinear: Data Tensor

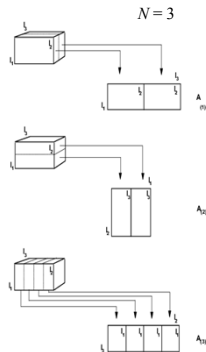
- $\mathbb{R}^{\text{people} \times \text{views} \times \text{illums} \times \text{express} \times \text{pixels}}$
- N-dimensional matrix
- 28 people, 45 images/person
- 5 views, 3 illuminations, 3 expressions per person



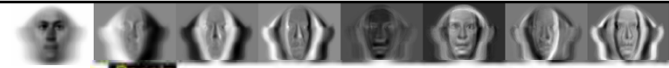
N-Mode SVD Algorithm

$$d_{ijk} = \sum_l \sum_p \sum_s z_{lps} u_s^l u_p^i u_l^k$$

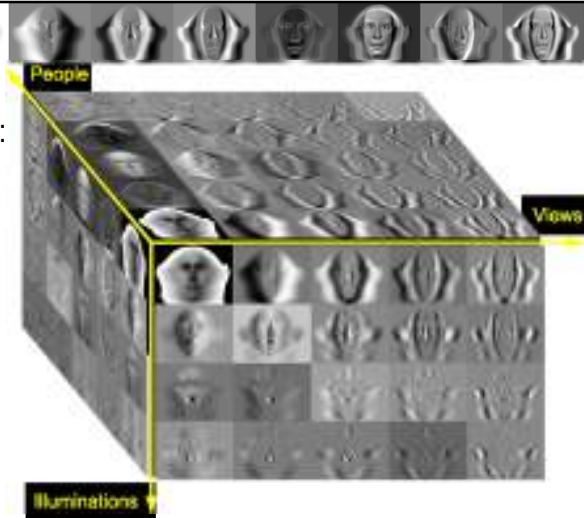
$$\mathcal{D} = \mathbf{Z} \times_1 \mathbf{U}_{\text{people}} \times_2 \mathbf{U}_{\text{views}} \times_3 \mathbf{U}_{\text{illums}} \times_4 \mathbf{U}_{\text{express}} \times_5 \mathbf{U}_{\text{pixels}}$$



PCA:

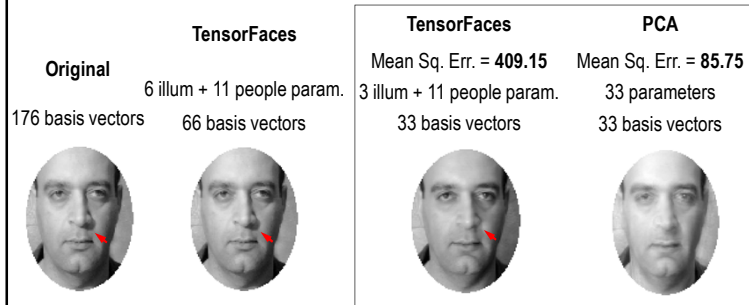


TensorFaces:



Strategic Data Compression = Perceptual Quality

- TensorFaces data reduction in illumination space primarily degrades illumination effects (cast shadows, highlights)
- PCA has *lower mean square error* but *higher perceptual error*



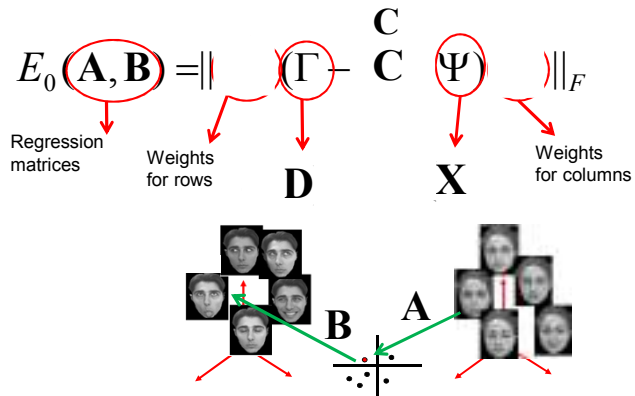
Outline

- Introduction (15 min)
- Generative models (40 min)
 - (PCA, k-means, spectral clustering, NMF, ICA, MDS)
- Discriminative models (40 min)
 - (LDA, SVM, OCA, CCA)
- Standard extensions of linear models (30 min)
 - (Kernel methods, Latent variable models, Tensor factorization)
- **Unified view (20 min)**

The fundamental equation of CA

(De la Torre & Kanade 06, De la Torre 2012)

Given two datasets $\mathbf{D} \in \mathbb{R}^{d \times n}$ and $\mathbf{X} \in \mathbb{R}^{x \times n}$



Properties of the cost function

- $E_0(\mathbf{A}, \mathbf{B})$ has a unique global minimum (Baldi and Hornik-89).
- Closed form solutions for \mathbf{A} and \mathbf{B} are:

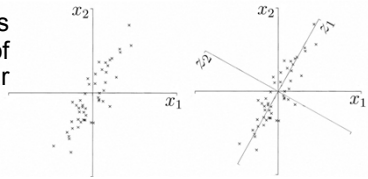
$$E_0(\mathbf{A}) = \text{tr} \left(\left(\mathbf{A}^T \mathbf{A} \right)^{-1} \mathbf{A}^T \mathbf{A} \right)$$

$$E_0(\mathbf{B}) = \text{tr} \left(\left(\mathbf{B}^T \mathbf{B} \right)^{-1} \mathbf{B}^T \mathbf{B} \right)$$

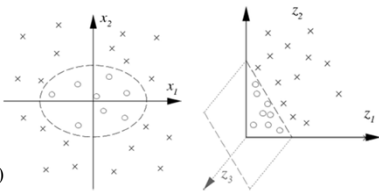
Principal Component Analysis (PCA)

(Pearson, 1901; Hotelling, 1933; Mardia et al., 1979; Jolliffe, 1986; Diamantaras, 1996)

- PCA finds the directions of maximum variation of the data based on linear correlation.



- Kernel PCA finds the directions of maximum variation of the data in the feature space.



$$(x_1, x_2) \rightarrow (x_1, x_2, x_1^2 + x_2^2) = (z_1, z_2, z_3)$$

PCA-Kernel PCA

- Error function for KPCA: (Eckardt & Young, 1936; Gabriel & Zamir, 1979; Baldi & Hornik, 1989; Shum et al., 1995; de la Torre & Black, 2003a)

$$E_0(\mathbf{A}, \mathbf{B}) = \|\mathbf{r}(\Gamma - \mathbf{B}\mathbf{A}^T)\mathbf{c}\|_F$$

$$E_{kPCA}(\mathbf{A}, \mathbf{B}) = \|\mathbf{D} - \mathbf{B}\mathbf{A}^T\|_F$$

$\varphi(\mathbf{D})$

- The primal problem:

$$E_{kPCA}(\mathbf{B}) = \text{tr}((\mathbf{B}^T \mathbf{B})^{-1} (\mathbf{B}^T \mathbf{D} \mathbf{D}^T \mathbf{B}))$$

- The dual problem:

$$E_{kPCA}(\mathbf{A}) = \text{tr}((\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{A}^T \varphi(\mathbf{D})^T \varphi(\mathbf{D}) \mathbf{A}))$$

Error function for LDA

(de la Torre & Kanade, 2006)

$$E_{LDA}(\mathbf{A}, \mathbf{B}) = \|\mathbf{G}^T \mathbf{G}^{-\frac{1}{2}} (\mathbf{G}^T - \mathbf{B}\mathbf{A}^T \mathbf{D})\|_F$$

$\mathbf{g}_{ij} \in \{0, 1\}$
 $\mathbf{G} \mathbf{1}_k = \mathbf{1}_n$
 $\mathbf{G} \in \mathbb{R}^{n \times k}$

$\mathbf{G}^T = \begin{bmatrix} 1 & \dots & 0 \\ 0 & \dots & 1 \\ \dots & \dots & \dots \\ 0 & \dots & 0 \end{bmatrix}$

\mathbf{A} (K=dim subspace) \rightarrow [d₁ d₂ ... d_n]
 d=pixels

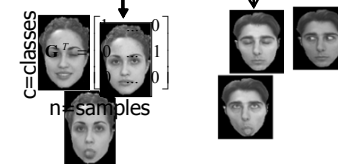
Equations $n \times c$ Unknowns $d \times c$

- $d \gg n$ an UNDETERMINED system of equations! (over-fitting)

Canonical Correlation Analysis (CCA)

$$E_0(\mathbf{A}, \mathbf{B}) = \|\mathbf{W}_r(\Gamma - \mathbf{B}\mathbf{A}^T \Psi)\mathbf{c}\|_F$$

$$E_{CCA}(\mathbf{A}, \mathbf{B}) = \|\mathbf{D}^T \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{B}\mathbf{A}^T \mathbf{X})\|_F$$



- CCA is the same as LDA changing the label matrix by a new set \mathbf{X}

K-means

(Ding et al., '02, Torre et al '06)

$$E_0(\mathbf{A}, \mathbf{B}) = \|\mathbf{D} - \mathbf{B}\mathbf{A}^T\|_F$$

Machine Perception of Human Behavior with CA F. De la Torre/J. Cohn PAVIS school on CV and PR 83

Normalized cuts

(Dhillon et al., '04, Zass & Shashua, 2005; Ding et al., 2005, De la Torre et al '06)

$$E_0(\mathbf{A}, \mathbf{B}) = \|\mathbf{\Gamma} - \mathbf{B}\mathbf{A}^T\mathbf{W}\|_F$$

$\mathbf{\Gamma} = [\varphi(\mathbf{d}_1) \ \varphi(\mathbf{d}_2) \ \dots \ \varphi(\mathbf{d}_n)]$ $\mathbf{\Gamma} = \varphi(\mathbf{D})$

Normalized Cuts (Shi & Malik '00)
Ratio-cuts (Hagen & Kahng '02)

Machine Perception of Human Behavior with CA F. De la Torre/J. Cohn PAVIS school on CV and PR 84

Other Connections

- The LS-KRRR (E_0) is also the generative model for:
 - Laplacian Eigenmaps, Locality Preserving projections, MDS, Partial least-squares,
- Benefits of LS framework:
 - Common framework to understand difference and communalities between different CA methods (e.g. KPCA, KLDA, KCCA, Ncuts)
 - Better understanding of normalization factors and generalizations
 - Efficient numerical optimization less than $\theta(n^3)$ or $\theta(d^3)$, where n is number of samples and d dimensions

Machine Perception of Human Behavior with CA F. De la Torre/J. Cohn PAVIS school on CV and PR 85