Statistical Multilinear Models: TensorFaces

Last Time: PCA

- Identifies an m dimensional explanation of n dimensional data where m < n.
- Originated as a statistical analysis technique.
- Attempts to minimize the reconstruction error under the following restrictions
  - Linear Reconstruction
  - Orthogonal Factors
- Equivalently, PCA attempts to maximize variance.

PART I: 2D Vision

- Appearance-Based Methods
  - Statistical Linear Models:
    - PCA
    -ICA, FLD
  - Non-negative Matrix Factorization, Sparse Matrix Factorization

Today ➔ Statistical Tensor Models:
- Multilinear PCA
- Multilinear ICA
- Person and Activity Recognition

The Problem with Linear (PCA) Appearance Based Recognition Methods

- Eigenimages work best for recognition when only a single factor – e.g., object identity – is allowed to vary
- Natural images are the composite consequences of multiple factors (or modes) related to scene structure, illumination and imaging

Unsupervised vs Supervised

- Unsupervised: - there are no labels for the data
  - PCA
  - ICA
- Supervised: - each image has label information that describes the factors that constructed the observed image
  - FLD
  - Modular PCA (View Based PCA)
Modular PCA: View-based PCA

Linear Representations

- PCA and ICA:
  - One linear model
  - Each image has a unique representation
  - Face (object) recognition
  - Each image has a unique representation
  - Every person has multiple representations

- FLD:
  - One linear model
  - Goal: one representation per person
  - Practice:
    - One representation per image
    - Every person (object) has multiple representations

- Modular PCA:
  - Assumption the data lies on a curved space that can be modeled locally by a linear model
  - A linear model for every cluster

Images are multivariate functions that respond to changes in viewpoint, illumination and geometry

Goal:

TensorFaces:

Multilinear Representation of Image Ensembles for Recognition and Compression

Linear vs Multilinear Manifolds

Previous Efforts to Improve Linear Models

- Improved PCA Methods:
    - "View-Based and Modular Eigenspaces for Face Recognition"
  - Nayar, Murase & Nene 1996
    - "Parametric Appearance Representation"

- Fisher Linear Discriminant:
  - Belhumeur & Hespanha & Kriegman
    - "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection"

- Bilinear Models:
  - Marimont & Wandell 1992
    - "Linear models of surface and illuminant spectra"
  - Freeman & Tenenbaum 1997
    - "Learning Bilinear Models for Two-Factor Problems in Vision"
Eigenfaces Experience Difficulties When More Than a Single Factor Varies

- Eigenfaces capture the variability across images without disentangling identity variability from variability in other factors such as lighting, viewpoint or expressions.

TensorFaces vs Eigenfaces (PCA)

<table>
<thead>
<tr>
<th>PIE Recognition Experiment</th>
<th>PCA</th>
<th>TensorFaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training: 23 people, 3 viewpoints (0, 34, -34), 4 illuminations</td>
<td>61%</td>
<td>80%</td>
</tr>
<tr>
<td>Testing: 23 people, 2 viewpoints (+17, -17), 4 illuminations (center, left, right, left-right)</td>
<td>27%</td>
<td>88%</td>
</tr>
</tbody>
</table>

Eigenfaces Experience Difficulties When More Than a Single Factor Varies

- Eigenfaces capture the variability across images without disentangling identity variability from variability in other factors such as lighting, viewpoint or expressions.

Multilinear Model Approach

- Non-linear appearance based technique
- Appearance based model that explicitly accounts for each of the multiple factors inherent in image formation
- Multilinear algebra, the algebra of higher order tensors
- Applied to facial images, we call our tensor technique “TensorFaces” [Vasilescu & Terzopoulos, ECCV'02, ICPR'02]

Data Organization

- Linear/PCA: Data Matrix
  - A matrix of image vectors
- Multilinear: Data Tensor
  - N-dimensional array
  - 28 people, 45 images/person
  - 5 views, 3 illuminations, 3 expressions per person
Data Organization

- Linear / PCA: Data Matrix
  - $R_{\text{pixels} \times \text{images}}$
  - A matrix of image vectors

- Multilinear: Data Tensor
  - $R_{\text{people} \times \text{views} \times \text{illums} \times \text{express} \times \text{pixels}}$
  - A 5-dimensional matrix
  - 28 people, 45 images/person
  - 5 views, 3 illums., 3 expressions per person

Tensor Decomposition

- $D$ is a $n$-dimensional matrix, comprising $N$-spaces
- $N$-mode SVD is the natural generalization of SVD
- $N$-mode SVD orthogonalizes these spaces & decomposes $D$ as the mode-$n$ product of $N$-orthogonal spaces

$$D = Z \times_1 U_1 \times_2 U_2 \ldots \times_n U_n$$

- $Z$ core tensor; governs interaction between mode matrices
- $U_n$, mode-$n$ matrix, is the column space of $D_{(n)}$

Learning Stage

Matrix Decomposition - SVD

- A matrix $D \in IR^{\times_i}$ has a column and row space
- SVD orthogonalizes these spaces and decomposes $D$

$$D = U_1 \sum U_2^T \quad (U_1 \text{ contains the eigenfaces})$$

- Rewrite in terms of mode-$n$ products

$$D = \sum x \times_1 U_1 \times_2 U_2$$

Tensor Decomposition

- $D = Z \times_1 U_1 \times_2 U_2 \times_3 U_3$
Multilinear (Tensor) Decomposition

\[ \mathcal{D} = Z \times_1 U_1 \times_2 U_2 \times_3 U_3 \]

\[ = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sigma_{ijk} u_{1i} \otimes u_{2j} \otimes u_{3k} \]

\[ \text{vec}(\mathcal{D}) = (U_3 \otimes U_2 \otimes U_1) \text{vec}(Z) \]

Facial Data Tensor Decomposition

\[ \mathcal{D} = Z \times_1 U_{\text{people}} \times_2 U_{\text{views}} \times_3 U_{\text{illums}} \times_4 U_{\text{express}} \times_5 U_{\text{pixels}} \]

N-Mode SVD Algorithm

1. For \( n = 1, \ldots, N \), compute matrix \( B_n \) by computing the SVD of the flattened matrix \( U_n \), and setting \( U_n \) to be the left matrix of the SVD.

2. Solve for the core tensor as follows

\[ Z = \mathcal{D} \times_1 U_{\text{1}}^T \times_2 U_{\text{2}}^T \times_3 U_{\text{3}}^T \times_4 U_{\text{4}}^T \times_5 U_{\text{5}}^T \]

Multilinear (Tensor) Algebra

\[ A \in \mathbb{R}^{l_1 \times l_2 \times \ldots \times l_N} \]

\[ \text{flattening} \]

\[ A_{(n)} \in \mathbb{R}^{l_1 \times (l_2 \times l_3 \times \ldots \times l_{n-1} \times l_{n+1} \times \ldots \times l_N)} \]
\textbf{Computing } \mathbf{U}_{\text{illums}} \textbf{ }

\begin{itemize}
  \item \( \mathbf{D}_{(\text{illums})} \) – flatten \( \mathbf{D} \) along the illumination dimension
  \item \( \mathbf{U}_{\text{illums}} \) – orthogonalizes the column space of \( \mathbf{D}_{(\text{illums})} \)
\end{itemize}

\textbf{Computing } \mathbf{U}_{\text{views}} \textbf{ }

\begin{itemize}
  \item \( \mathbf{D}_{(\text{views})} \) – flatten \( \mathbf{D} \) along the view point dimension
  \item \( \mathbf{U}_{\text{views}} \) – orthogonalize the column space of \( \mathbf{D}_{(\text{views})} \)
\end{itemize}

\textbf{Computing } \mathbf{U}_{\text{illums}} \textbf{ }

\begin{itemize}
  \item \( \mathbf{D}_{(\text{illums})} \) – flatten \( \mathbf{D} \) along the illumination dimension
  \item \( \mathbf{U}_{\text{illums}} \) – orthogonalize \( \mathbf{D}_{(\text{illums})} \)
\end{itemize}

\textbf{Computing } \mathbf{U}_{\text{views}} \textbf{ }

\begin{itemize}
  \item \( \mathbf{D}_{(\text{views})} \) – flatten \( \mathbf{D} \) along the view point dimension
  \item \( \mathbf{U}_{\text{views}} \) – orthogonalize \( \mathbf{D}_{(\text{views})} \)
\end{itemize}

\textbf{Computing } \mathbf{U}_{\text{pixels}} \textbf{ }

\begin{itemize}
  \item \( \mathbf{D}_{(\text{pixels})} \) – flatten \( \mathbf{D} \) along the pixel dimension
  \item \( \mathbf{U}_{\text{pixels}} \) – orthogonalize \( \mathbf{D}_{(\text{pixels})} \)
\end{itemize}

\textbf{N-Mode SVD Algorithm}

1. For \( n=1,\ldots,N \), compute matrix \( \mathbf{U}_n \) by computing the SVD of the flattened matrix \( \mathbf{D}_{(n)} \) and setting \( \mathbf{U}_n \) to be the left matrix of the SVD.

2. Solve for the core tensor as follows

\[
\mathbf{Z} = \mathbf{D} \mathbf{x}_1 \mathbf{U}_1^T \mathbf{x}_2 \mathbf{U}_2^T \mathbf{x}_3 \mathbf{U}_3^T \mathbf{x}_4 \mathbf{U}_4^T \mathbf{x}_5 \mathbf{U}_5^T
\]
Mode-\(n\) Product

- Mode-\(n\) product is a generalization of the product of two matrices.
- It is the product of a tensor with a matrix.
- Mode-\(n\) product of \(A\in\mathbb{R}^{I_1\times I_2\times...\times I_{n}}\) and \(M\in\mathbb{R}^{J_1\times J_2\times...\times J_{n}}\):
\[
\begin{aligned}
B &= A \times_n M \\
&= \sum_{l_n} a_{I_1I_2...I_{n-1}l_nJ_1J_2...J_{n-1}}^{} m_{J_nl_n}
\end{aligned}
\]

Mode-N Product

- Mode-N product is a generalization of the product of two matrices.
- It is the product of a tensor with a matrix.
- Mode-N product of \(A\in\mathbb{R}^{I_1\times I_2\times...\times I_{N}}\) and \(M\in\mathbb{R}^{J_1\times J_2\times...\times J_{N}}\):
\[
\begin{aligned}
\langle A \times_n M \rangle_{I_1J_1,...,I_nJ_n} &= \sum_{l_n} a_{I_1I_2...I_{n-1}l_nJ_1J_2...J_{n-1}}^{} m_{J_nl_n}
\end{aligned}
\]

Multilinear (Tensor) Algebra

- \(N\)-th order tensor \(A\in\mathbb{R}^{I_1\times I_2\times...\times I_N}\)
- matrix (2nd order tensor) \(M\in\mathbb{R}^{J_1\times J_2\times...\times J_N}\)
- mode-\(n\) product:
\[
B = A \times_n M \quad \text{where} \quad B_{(n)} = MA_{(n)}
\]

Eigenfaces vs TensorFaces

- Multilinear Analysis / TensorFaces:
\[
\Omega = Z_\text{views} U_\text{people} \otimes Z_\text{illums} U_\text{express} \otimes Z_\text{pixels} U_\text{pixels}
\]
- Linear Analysis / Eigenfaces:
\[
D_{\text{data}} = U_\text{basis}^T Z_\text{views} (U_\text{people} \otimes U_\text{illums} \otimes U_\text{express} \otimes U_\text{pixels})
\]
- TensorFaces subsumes Eigenfaces

TensorFaces:

- \(B = Z \times_5 U_\text{pixels}\)
- explicitly represent covariance across factors

PCA:

- People
- Views
- Illums.
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

<table>
<thead>
<tr>
<th></th>
<th>TensorFaces</th>
<th>PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>176 basis vectors</td>
<td>3 illum + 11 people param.</td>
</tr>
<tr>
<td>TensorFaces</td>
<td>66 basis vectors</td>
<td>3 illum + 11 people param.</td>
</tr>
<tr>
<td>PCA</td>
<td>33 basis vectors</td>
<td></td>
</tr>
</tbody>
</table>

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