MULTIVARIATE (& MULTIMODALITY) IMAGE ANALYSIS

Duygu Tosun-Turgut, Ph.D.
Center for Imaging of Neurodegenerative Diseases
Department of Radiology and Biomedical Imaging

duygu.tosun@ucsf.edu
Curse of dimensionality

- A major problem in imaging statistics is the curse of dimensionality.

- If the data $x$ lies in high dimensional space, then an enormous amount of data is required to learn distributions or decision rules.
Curse of dimensionality

• One way to deal with dimensionality is to assume that we know the form of the probability distribution.

• For example, a Gaussian model in N dimensions has $N + N(N-1)/2$ parameters to estimate.

• Requires $O(N^2)$ data to learn reliably. This may be practical.
Typical solution: divide into subproblems

- Each subproblem relates single voxel to clinical variable
- Known as voxel-wise, univariate, or pointwise regression
- Approach popularized by SPM (Friston, 1995)

- Dependencies between spatial variables neglected!!
Dimension reduction

• One way to avoid the curse of dimensionality is by projecting the data onto a lower-dimensional space.

• Techniques for dimension reduction:
  • Principal Component Analysis (PCA)
  • Fisher’s Linear Discriminant
  • Multi-dimensional Scaling.
  • Independent Component Analysis (ICA)
A dual goal

• Find a good representation
  • The features part

• Reduce redundancy in the data
  • A side effect of “proper” features
A “good feature”

• “Simplify” the explanation of the input
  • Represent repeating patterns
  • When defined makes the input simpler

• How do we define these abstract qualities?
  • On to the math …
Linear features

\[ Z = WX \]
A 2D example

Matrix representation of data

$Z = WX = \begin{bmatrix} Z_1^T \\ Z_2^T \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \end{bmatrix}$
Defining a goal

- Desirable feature features
  - Give “simple” weights
  - Avoid feature similarity

- How do we define these?
One way to proceed

• “Simple weights”
  • Minimize relation of the two dimensions
  • Decorrelate:
    \[ z_1^T z_2 = 0 \]

• “Feature similarity”
  • Same thing!
  • Decorrelate:
    \[ w_1^T w_2 = 0 \]
Diagonalizing the covariance

- Covariance matrix

\[ \text{Cov}(z_1, z_2) = \begin{bmatrix} z_1^T z_1 & z_1^T z_2 \\ z_2^T z_1 & z_2^T z_2 \end{bmatrix} / N \]

- Diagonalizing the covariance suppresses cross-dimensional co-activity

- if \( z_1 \) is high, \( z_2 \) won’t be…

\[ \text{Cov}(z_1, z_2) = \begin{bmatrix} z_1^T z_1 & z_1^T z_2 \\ z_2^T z_1 & z_2^T z_2 \end{bmatrix} / N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \]
Problem definition

• For a given input $\mathbf{X}$

• Find a feature matrix $\mathbf{W}$

• So that the weights decorrelate

$$ (\mathbf{WX})(\mathbf{WX})^T = NI \Rightarrow ZZ^T = NI $$

$$ \Rightarrow \mathbf{WCov(X)W}^T = \mathbf{I} $$

• Solving for diagonalization!
Solving for diagonalization

Covariance matrices are positive definite
• Therefore symmetric
  • have orthogonal eigenvectors and real eigenvalues
• and are factorizable by:

\[ U^T A U = \Lambda \]

where \( U \) has eigenvectors of \( A \) in its columns

\[ \Lambda = \text{diag}(\lambda_i), \text{ where } \lambda_i \text{ are the eigenvalues of } A \]
Decorrelation

- An implicit Gaussian assumption
  - $N$-D data has $N$ directions of variance
Undoing the variance

- The decorrelating matrix $W$ contains two vectors that normalize the input’s variance.
Resulting transform

- Input gets scaled to a well behaved Gaussian with unit variance in all dimensions
A more complex case

- Having correlation between two dimensions
  - We still find the directions of maximal variance
  - But we also rotate in addition to scaling
One more detail

- So far we considered zero-mean inputs
  - The transforming operation was a rotation
- If the input mean is not zero bad things happen!
  - Make sure that your data is zero-mean!
Principal component analysis

- This transform is known as PCA
  - The features are the principal components
    - They are orthogonal to each other
    - And produce orthogonal (white) weights
  - Major tool in statistics
    - Removes dependencies from multivariate data

- Also known as the KLT
  - Karhunen-Loeve transform
A better way to compute PCA

• The Singular Value Decomposition way

\[
[U, S, V] = \text{SVD}(A) \Rightarrow A = USV^T
\]

• Relationship to eigendecomposition
  • In our case (covariance input \(A\)), \(U\) and \(S\) will hold the eigenvectors/values of \(A\)

• Why the SVD?
  • More stable, more robust, fancy extensions
Dimensionality reduction

• PCA is great for high dimensional data

• Allows us to perform dimensionality reduction
  • Helps us find relevant structure in data
  • Helps us throw away things that won’t matter
What is the number of dimensions?

- If the input was $M$ dimensional, how many dimensions do we keep?
  - No solid answer (estimators exist)

- Look at the singular/eigen-values
  - They will show the variance of each component, at some point it will be small
Example

- Eigenvalues of 1200 dimensional video data
  - Little variance after component 30
  - We don’t need to keep the rest of the data

Dimension reduction occurs by ignoring the directions in which the covariance is small.
Recap…

PCA decorrelates multivariate data, finds useful components, reduces dimensionality.

• PCA is only powerful if the biological question is related to the highest variance in the dataset
• If not other techniques are more useful: Independent Component Analysis
Limitations of PCA

- PCA may not find the best directions for discriminating between two classes.
- $1^{\text{st}}$ eigenvector is best for representing the probabilities.
- $2^{\text{nd}}$ eigenvector is best for discrimination.
Fisher’s linear discriminant analysis (LDA)

- Performs dimensionality reduction “while preserving as much of the class discriminatory information as possible”.

- Seeks to find directions along which the classes are best separated.

- Takes into consideration the scatter within-classes but also the scatter between-classes.
LDA via PCA

• PCA is first applied to the data set to reduce its dimensionality.

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N
\end{bmatrix}
\xrightarrow{\text{PCA}}
\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_K
\end{bmatrix}
\]

• LDA is then applied to find the most discriminative directions.

\[
\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_K
\end{bmatrix}
\xrightarrow{\text{LDA}}
\begin{bmatrix}
  z_1 \\
  z_2 \\
  \vdots \\
  z_{C-1}
\end{bmatrix}
\]
PCA fails…

- Focus on **uncorrelated** and **Gaussian** components
- Second-order statistics
- Orthogonal transformation

- Focus on **independent** and **non-Gaussian** components
- Higher-order statistics
- Non-orthogonal transformation
Statistical independence

- If we know something about $x$, that should tell us *nothing* about $y$. 

![Graph showing dependent and independent variables]
Concept of ICA

- A given signal ($x$) is generated by linear mixing ($A$) of independent components ($s$)
- ICA is a statistical analysis method to estimate those independent components ($z$) and mixing rule ($W$)

$$z = Wx = WAs$$
Example: PCA and ICA

Model:

\[
\begin{bmatrix}
    x_1 \\
    x_2
\end{bmatrix} =
\begin{bmatrix}
    a_{11} & a_{12} \\
    a_{21} & a_{22}
\end{bmatrix}
\begin{bmatrix}
    s_1 \\
    s_2
\end{bmatrix}
\]
How it works?

If independent as well, the pdf is separable:

$$p(s_1, s_2) = p_1(s_1) p_2(s_2)$$

which implies

$$E\{f_1(s_1) f_2(s_2)\} - E\{f_1(s_1)\} E\{f_2(s_2)\} = 0$$

for any functions $f_1$ and $f_2$ ⇒ useful for solving.
Rationale of ICA

• Find the components $S_i$ that are as independent as possible in the sense of maximizing some function $F(s_1, s_2, \ldots, s_k)$ that measures independence.

• All ICs (except 1) should be non-Gaussian.

• The variance of all ICs is 1.

• There is no hierarchy between ICs.
How to find ICs?

- Many choices of objective function $F$
- Mutual information

$$MI = \int f(s_1, s_2, ..., s_k) \log \frac{f(s_1, s_2, ..., s_k)}{f_1(s_1)f_2(s_2)...f_k(s_k)}$$

- Use the kurtosis of the variables to approximate the distribution function

- The number of ICs is chosen by the user
Difference with PCA

- It is not a dimensionality reduction technique.
- There is no single (exact) solution for components (in R: FastICA, PearsonICA, MLICA)
- ICs are of course uncorrelated but also as independent as possible
- Uninteresting for Gaussian distributed variables
Feature Extraction in ECG data (Raw Data)
Feature Extraction in ECG data (PCA)
Feature Extraction in ECG data (Extended ICA)
Feature Extraction in ECG data (flexible ICA)
Application domains of ICA

• Image denoising
• Medical signal processing – fMRI, ECG, EEG
• Feature extraction, face recognition
• Compression, redundancy reduction
• Watermarking
• Clustering
• Time series analysis (stock market, microarray data)
• Topic extraction
• Econometrics: Finding hidden factors in financial data
Gene clusters [Hsiao et al. 2002]

- Each gene is mapped to a point based on the value assigned to the gene in the 14th (x-axis), 15th (y-axis) and 55th (z-axis) independent components:
  - Red - enriched with liver-specific genes
  - Orange – enriched with muscle-specific genes
  - Green – enriched with vulva-specific genes
  - Yellow – genes not annotated as liver-, muscle-, or vulva-specific
Segmentation by ICA

Image, $I$ $\xrightarrow{\text{ICA Filter Bank}}$ With $n$ filters $\xrightarrow{I_1, I_2, \ldots, I_n}$ Clustering

Segmented image, $C$

Above is an unsupervised setting.

Segmentation (i.e., classification in this context) can also be performed by a supervised method on the output feature images $I_1, I_2, \ldots, I_n$.

Recap: PCA vs LDA vs ICA

- **PCA**: Proper for **dimension reduction**

- **LDA**: Proper for **pattern classification** if the number of training samples of each class are large

- **ICA**: Proper for **blind source separation** or classification using ICs when class id of training data is not available
Multimodal Data Collection

Brain Function (spatiotemporal, task-related)

Brain Structure (spatial)

Genetic Data (spatial/chromosomal)

Covariates (Age, etc.)

EEG

tfMRI

T1/T2

SNP

Gene Expression

Task 1-N

Other*
Joint ICA

- Variation on ICA to look for components that appear jointly across features or modalities

\[
\begin{bmatrix}
X_{\text{Modality 1}} & X_{\text{Modality 2}}
\end{bmatrix} = A \times \begin{bmatrix}
S_{\text{Modality 1}} & S_{\text{Modality 2}}
\end{bmatrix}
\]
jICA Results: Multitask fMRI

One component
Significant at p<0.01

jICA Results: Joint Histograms

Parallel ICA: Two Goals


MAX : \{H(Y1) + H(Y2) \}, \text{<Infomax>}
Subject to: arg max g\{W1,W2 \; \hat{s}_1, \hat{s}_2 \}, \quad g(\cdot) = \text{Correlation}(A_1, A_2)^2 = \frac{\text{Cov}(a_{1i}, a_{2j})^2}{\text{Var}(a_{1i}) \times \text{Var}(a_{2j})}
Example 2: Default Mode Network & Multiple Genetic Factors

fMRI Component t-map

Mixing Coefficients ($\rho=0.31$)

SNPs
Component
rs1011313
rs10503929
rs10936143
rs2150157
rs2236797
rs4784320
rs6136
rs6578993
rs7702598
rs7961819
rs3757934
rs3772917
rs8027035

Partial least squares (PLS)

- Understanding relationships between process & response variables
PLS fundamentals
PLS is the multivariate version of regression

• PLS uses two different **PCA models**, one for the X’s and one for the Y’s, and finds the links between the two.

• Mathematically, the difference is as follows:
  • In PCA, we are maximizing the *variance* that is explained by the model.
  • In PLS, we are maximizing the *covariance*. 
Conceptually how PLS works?

A step-wise process:

- PLS finds a set of orthogonal components that:
  - maximize the level of explanation of both X and Y
  - provide a predictive equation for Y in terms of the X’s

- This is done by:
  - fitting a set of components to X (as in PCA)
  - similarly fitting a set of components to Y
  - reconciling the two sets of components so as to maximize explanation of X and Y

- **PLS finds components of X that are also relevant to Y**
  - *Latent vectors are components that simultaneously decompose X and Y*
  - *Latent vectors explain the covariance between X and Y*
PLS – the “Inner Relation”

- The way PLS works visually is by “tweeking” the two PCA models (X and Y) until their covariance is optimised. It is this “tweeking” that led to the name partial least-squares.
Canonical correlation analysis (CCA)

- CCA was developed first by H. Hotelling.

- CCA measures the linear relationship between two multidimensional variables.
- CCA finds two bases, one for each variable, that are optimal with respect to correlations.
- Applications in economics, medical studies, bioinformatics and other areas.
Canonical correlation analysis (CCA)

- Two multidimensional variables
  - Two different measurements on the same set of objects
    - Web images and associated text
    - Protein (or gene) sequences and related literature (text)
    - Protein sequence and corresponding gene expression
    - In classification: feature vector and class label
  - Two measurements on the same object are likely to be correlated.
    - May not be obvious on the original measurements.
    - Find the maximum correlation on transformed space.
Canonical correlation analysis (CCA)

$X^T$

$Y^T$

measurement

$W_X$

transformation

$W_Y$

Transformed data

Correlation
Problem definition

- Find two sets of basis vectors, one for \( x \) and the other for \( y \), such that the correlations between the projections of the variables onto these basis vectors are maximized.

Given \( S = ((x_1, y_1), \ldots, (x_n, y_n)) \) of \( (x, y) \)

Compute two basis vectors \( w_x \) and \( w_y \):

\[
x \rightarrow \langle w_x, x \rangle \quad S_{x, w_x} = (\langle w_x, x_1 \rangle, \ldots, \langle w_x, x_n \rangle)
\]

\[
y \rightarrow < w_y, y > \quad S_{y, w_y} = (\langle w_y, y_1 \rangle, \ldots, \langle w_y, y_n \rangle)
\]
Problem definition

- Compute the two basis vectors so that the correlations of the projections onto these vectors are maximized.

\[ \rho = \max_{w_x, w_y} \text{corr}(S_x w_x, S_y w_y) \]

\[ = \max_{w_x, w_y} \frac{\langle S_x w_x, S_y w_y \rangle}{\| S_x w_x \| \| S_y w_y \|}. \]
Least squares regression

- Estimate the parameters $\beta$ based on a set of training data: $(x_1, y_1) \ldots (x_N, y_N)$

- Minimize residual sum of squares

\[
\text{RSS}(\beta) = \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2
\]

\[
\hat{\beta} = \left( X^T X \right)^{-1} X^T y \quad \text{poor numeric properties!}
\]
Subset selection

• We want to eliminate unnecessary features
• Use additional penalties to reduce coefficients: coefficient shrinkage

Ridge Regression
• Minimize least squares s.t. \[ \sum_{j=1}^{p} \beta_j^2 \leq s \]

The Lasso
• Minimize least squares s.t. \[ \sum_{j=1}^{p} | \beta_j | \leq s \]

Principal Components Regression
• Regress on \( M < p \) principal components of \( X \)

Partial Least Squares
• Regress on \( M < p \) directions of \( X \) weighted by \( y \)
TABLE 3.3. Estimated coefficients and test error results, for different subset and shrinkage methods applied to the prostate data. The blank entries correspond to variables omitted.

<table>
<thead>
<tr>
<th>Term</th>
<th>LS</th>
<th>Best Subset</th>
<th>Ridge</th>
<th>Lasso</th>
<th>PCR</th>
<th>PLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>2.480</td>
<td>2.495</td>
<td>2.467</td>
<td>2.477</td>
<td>2.513</td>
<td>2.452</td>
</tr>
<tr>
<td>lcavol</td>
<td>0.680</td>
<td>0.740</td>
<td>0.389</td>
<td>0.545</td>
<td>0.544</td>
<td>0.440</td>
</tr>
<tr>
<td>lweight</td>
<td>0.305</td>
<td>0.367</td>
<td>0.238</td>
<td>0.237</td>
<td>0.337</td>
<td>0.351</td>
</tr>
<tr>
<td>age</td>
<td>-0.141</td>
<td>-0.029</td>
<td></td>
<td>-0.152</td>
<td>-0.017</td>
<td></td>
</tr>
<tr>
<td>lbph</td>
<td>0.210</td>
<td>0.159</td>
<td>0.098</td>
<td>0.213</td>
<td>0.248</td>
<td></td>
</tr>
<tr>
<td>svi</td>
<td>0.305</td>
<td>0.217</td>
<td>0.165</td>
<td>0.315</td>
<td>0.252</td>
<td></td>
</tr>
<tr>
<td>lcpx</td>
<td>-0.288</td>
<td>0.026</td>
<td></td>
<td>-0.053</td>
<td>0.078</td>
<td></td>
</tr>
<tr>
<td>gleason</td>
<td>-0.021</td>
<td>0.042</td>
<td></td>
<td>0.230</td>
<td>0.003</td>
<td></td>
</tr>
<tr>
<td>pgg45</td>
<td>0.267</td>
<td>0.123</td>
<td>0.059</td>
<td>-0.053</td>
<td>0.080</td>
<td></td>
</tr>
<tr>
<td>Test Error</td>
<td>0.586</td>
<td>0.574</td>
<td>0.540</td>
<td>0.491</td>
<td>0.527</td>
<td>0.636</td>
</tr>
<tr>
<td>Std. Error</td>
<td>0.184</td>
<td>0.156</td>
<td>0.168</td>
<td>0.152</td>
<td>0.122</td>
<td>0.172</td>
</tr>
</tbody>
</table>
Error comparison
Shrinkage methods: Ridge regression

FIGURE 3.6. Profiles of ridge coefficients for the prostate cancer example, as tuning parameter $\lambda$ is varied. Coefficients are plotted versus $\text{df}$, the effective degrees of freedom. A vertical line is drawn at $\text{df} = 4.16$, the value chosen by cross-validation.
Shrinkage methods: Lasso regression

FIGURE 3.8. Profiles of lasso coefficients, as tuning parameter $t$ is varied. Coefficients are plotted versus $s = t / \sum_{j} |\hat{\beta}_j|$. A vertical line is drawn at $s = 0.5$, the value chosen by cross-validation. Compare Figure 3.4.3; the lasso profiles hit zero, while those for ridge do not.
A unifying view

We can view all the linear regression techniques under a common framework:

\[
\hat{\beta} = \arg\min_\beta \left\{ \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} | \beta_j |^q \right\}
\]

\(\lambda\) includes bias, \(q\) indicates a prior distribution on \(\beta\)
- \(\lambda = 0\): least squares
- \(\lambda > 0, q = 0\): subset selection (counts number of nonzero parameters)
- \(\lambda > 0, q = 1\): the lasso
- \(\lambda > 0, q = 2\): ridge regression
Family of shrinkage regression

\[ \hat{\beta} = \arg \min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\} \]

FIGURE 3.12. Contours of constant value of \( \sum_{j} |\beta_j|^q \) for given values of \( q \).
Atrophy associated with delayed memory

Spatial correlation

Ridge regression