Statistical Analysis of Shapes

by J. S. Marron

Department of Statistics
University of North Carolina

Statistics in Image Processing

Old Roles:

- Denoising
- Segmentation

New Roles:

- Understanding *populations* of images / objects
- Discrimination (i.e. classification)

Statistics in Image Processing (cont.)

Personal Interest: development of new statistical methods

Main Challenge: High Dimension, Low Sample Size

- Endemic to Image Analysis
- Classical Statistical Methods useless
- Huge Need for Invention of New Methods!

Relevant New Statistical Area

Functional Data Analysis

A personal view: what is the “atom” of the statistical analysis?

1st course in statistics: “atoms” are numbers

Statistical multivariate analysis: “atoms” are vectors

Functional Data: “atoms” are *more complex objects*

Functional Data Analysis (cont.)

FDA: "atoms" are more complex objects, e. g.

- curves [toy example]

- images, e.g. Cornea data (Cohen, Tripoli) [example]

- shapes, e.g. Corpus Callosum Data (Ho, Gerig) [example]

  M-rep version (Yushkevich) [example]

Functional Data Analysis (cont.)

Recommended Source:

Ramsay, J. O. & Silverman, B. W. (1997) *Functional Data Analysis*, Springer, N.Y.

(there is 2nd book that I have not seen yet, "more applied and example oriented")

Drawback: Only curves, no more complex data objects

Strength: Excellent source for many deep analytic ideas

Data Representation

Object Space $\leftrightarrow$ Feature space

Curves

Images

Shapes

Vectors

$$\begin{pmatrix} x_{1,1} \\ \vdots \\ x_{d,1} \end{pmatrix}, \cdots, \begin{pmatrix} x_{1,n} \\ \vdots \\ x_{d,n} \end{pmatrix}$$

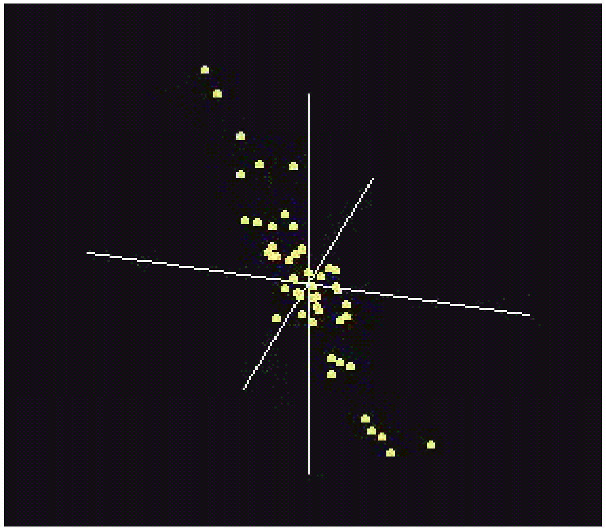
One to one mapping couples visualization in Object Space, with statistical analysis in Feature Space

High Dim'al Data Conceptualization

Feature space $\leftrightarrow$ Point Clouds

Vectors

$$\begin{pmatrix} x_{1,1} \\ \vdots \\ x_{d,1} \end{pmatrix}, \cdots, \begin{pmatrix} x_{1,n} \\ \vdots \\ x_{d,n} \end{pmatrix}$$



[Spinning Point Cloud Graphic] .

E.g. 1: Curves [example]

Data Objects: $f_1(x),\ldots,f_n(x)$ (conceptual model)

Digital version: $\begin{pmatrix} f_1(x_1) \\ \vdots \\ f_1(x_d) \end{pmatrix},\ldots,\begin{pmatrix} f_n(x_1) \\ \vdots \\ f_n(x_d) \end{pmatrix}$, for a "grid" $x_1,\ldots,x_d$

Object Space View: Overlay plots of curves

Feature space: $\left\{\begin{pmatrix} f_i(x_1) \\ \vdots \\ f_i(x_d) \end{pmatrix} : i = 1,\ldots,n\right\}$, e.g. dimension $d = 10$

E.g. 2: Images, Corneas [example]

Special thanks to K. L. Cohen and N. Tripoli,
UNC Ophthalmology

Reference:

Locantore, N., Marron, J. S., Simpson, D. G., Tripoli, N., Zhang, J. T. and Cohen, K. L. (1999) Robust Principal Component Analysis for Functional Data, *Test*, 8, 1-73.

Data Objects: color map of "temperature scale radial curvature"

- "hot" = more curvature
- "cool" = less curvature

E.g. 2: Images, Corneas [example] (cont.)

Feature vectors: Digitized version is “large and wasteful”

Instead use coefficients of Zernike Basis repres’n, $d = 66$

Schwiegerling, J., Greivenkamp, J. E., and Miller, J. M. (1995) Representation of videokeratoscopic height data with Zernike polynomials, *Journal of the Optical Society of America, Series A*, 12, 2105-2113.

Born, M. and Wolf, E. (1980) *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Pergamon Press, New York.

E.g. 2: Images, Corneas [example] (cont.)

Object Space view: can't overlay images

Instead show images sequentially

Hard to see "population structure"

E.g. 3: shapes, Corpora Callosa [example]

Data Objects: boundaries of "segmented" corpora callosa

Feature vectors: use coefficients of Fourier boundary representation, $d = 80$

Object Space view: can either overlay, or show sequentially

In either case: hard to see "population structure"

M-rep version: same issues

Finding and visualizing structure in populations

Powerful method: Principal Component Analysis

Presentation here:

- Focus on visualization

Underlying mathematics:

- Eigen-analysis of covariance matrix
- Singular Value Decomposition of Data Matrix

Principal Component Analysis (PCA)

There are many names (lots of reinvention?):

Statistics: Principal Component Analysis (PCA)

Social Sciences: Factor Analysis (PCA is a subset)

Probability / Electrical Eng: Karhunen – Loeve expansion

Applied Mathematics: Proper Orthog'l Decomposition (POD)

Geo-Sciences: Empirical Orthogonal Functions (EOF)

PCA, Optimization View

Goal: find "direction of greatest variability"

[Spinning point Cloud] - [Axis of greatest variability]

Question: "direction" from where?

PCA, Optimization View (cont.)

Step 1: Start with Center Point:

$$\text{Sample Mean: } \ \underset{\rightarrow}{\bar{x}} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_d \end{pmatrix} = \begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n} x_{i1} \\ \vdots \\ \frac{1}{n}\sum_{i=1}^{n} x_{id} \end{pmatrix},$$

Aside: "mean vector" = "vector of means" is not obvious

Notation: "under-arrow" used for vectors

PCA, Optimization View (cont.)

Step 2: Work with re-centered data:

$\underset{\rightarrow}{x}_i - \overline{\underset{\rightarrow}{x}}, \quad i = 1, \ldots, n,$ the "mean residuals"

Step 3: Consider all possible "directions"

Step 4: Project (find closest point) data onto direction vector

Step 5: Maximize "spread" (sample variance), over direction

Step 6: Project data onto orthogonal subspace, and repeat.

Curves, Toy Example I

Features of Graphic:

- Data
- Mean
- Residuals
- PC Decomps – projections
- PC Decomps – mean +- extremes
- PC Residuals
- % Sums of Square – summarizing “signal power”
- Summarization of % SS
- 1-d Projections

More Curve Toy Examples

Toy Example 2:

Shows 1-d projections useful to highlight “clusters”

Toy Example 3:

Shows 1-d projections useful to identify “outliers”

Toy Example 4:

Show 2-d Projections useful   [2-d Projections]

PCA Analysis of Cornea Data

Recall [Raw Data]

PCA: same as before (do analysis in feature space)

Visual problem: can't overlay projections

Solution:

March along eigenvector in feature space

Study corresponding image in object space

PCA Analysis of Cornea Data (cont.)

PC1: Overall curvature & “with the rule” astigmatism

PC2: Big problem caused by an outlier?!? [toy 2-d graphic]

Approach 1: Outlier deletion?

- Problem: too many outliers (recheck raw data)

Approach 2: Robust PCA (i.e. “reduced influence” methods)

PCA Analysis of Cornea Data (cont.)

Robust PCA I: “Projection Pursuit”

- Works well in 4-5 dimensions, 20 max (<< 66 here)

Robust PCA II: Eigenanalysis of Robust covariance est.

- “affine invariance” fails for HDLSS

Robust PCA III: Spherical and Elliptical PCA

Elliptical PC1

Elliptical PC2

PCA analysis of M-rep Corpora Caloosa

Recall Raw Data: Hard to see "structure of population"

PC1: Overall Curvature

PC2: Rotation of ends

PC3: Variation of curvature

Note: "Directions are orthogonal"

Discrimination

Main idea:

Have several “classes” of data. E.g “Healthy” and “Diseased”

Find a “rule” for assigning new cases to each class.

Standard statistical method:

Fisher Linear Discrimination (FLD)

i. e. Linear Discriminant Analysis (LDA)

Big Problem: Fails in HDLSS contexts

Discrimination (cont.)

Serious competitor (in HDLSS situations):

"Mean Difference", i.e. "centroid" method

Idea: choose class with closest mean

Weakness: ignores covariance information

Strength: ignores covariance information

Personal observation: simplicity is often a strength…

Discrimination (cont.)

Comparison between FLD and Mean Difference [toy data]:

I. Large Sample "Two Meatballs" [PCA works]:

FLD ≈ Mean Difference

II. Large Sample "Parallel Squished" Gaussian [PCA fails]:

FLD >> Mean Difference

III. HDLSS Gaussian:

FLD << Mean Difference

Discrimination (cont.)

Interesting "High Tech" method from "Machine Learning":

Support Vector Machine

Main Idea: Find "separating hyperplane"

To maximize the "margin" i.e. minimum distance to plane

[Graphical Illustration]

- Add "penalty" when have "violations"
- Find solution by quadratic programming

Support Vector Machine

Classical References:

Vapnik (1982) *Estimation of dependences based on empirical data*, Springer (Russian version, 1979)

Vapnik (1995) *The nature of statistical learning theory*, Springer.

Recommended tutorial:

Burges (1998) A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, **2**, 955-974, see also web site:
http://citeseer.nj.nec.com/burges98tutorial.html

Support Vector Machine (cont.)

HDLSS performance of SVM:

- Somewhat Shaky
- Reason is too strongly feels "support vectors"
- And too many of them

But FLD is much worse

Support Vector Machine (cont.)

Coming Improvement (for HDLSS contexts)

Distance Weighted Discrimination

- Idea: feels all of the data, not just “support vectors”
- Needs more sophisticated optimization

  (2^{nd} Order Cone Programming)

- Nearly ready for prime time

Discrimination (cont.)

Apparent Weakness of Above Methods:

Only allow "separating planes"

Solution 1: Gaussian Likelihood methods

Weakness: fails in HDLSS situations

Solution 2: Nonlinear surfaces:

- Nearest Neighbors
- Parzen Windows
- Neural Nets

Drawback: No insights about data

Discrimination (cont.)

Solution 3: "Kernel Embedding" methods

Fundamental Reference:

Aizerman, Braverman and Rozoner (1964) *Automation and Remote Control*, **15**, 821-837.

{Historical Note: way before SVM and "machine learning"}

Polynomial Embedding

Motivating idea: extend “scope” of linear discrimination,

by adding “nonlinear components” to data

(better use of name “nonlinear discrimination”????)

E.g. In 1d, “linear separation” splits the domain

$$\{x : x \in \Re\}$$

into only 2 parts [toy graphic]

Polynomial Embedding (cont.)

But in the "quadratic embedded domain"

$$\left\{\left(x, x^2\right) : x \in \Re\right\} \subset \Re^2$$

linear separation can give 3 parts   [toy graphic]

- original data space lies in 1d manifold
- very sparse region of $\Re^2$
- curvature of manifold gives better linear separation
- can have *any* 2 break points (2 points $\Rightarrow$ line)

Polynomial Embedding (cont.)

Stronger effects for higher order polynomial embedding:

E.g. for cubic, $\left\{\left(x, x^2, x^3\right): x \in \Re\right\} \subset \Re^3$

linear separation can give 4 parts (or fewer) [toy graphic]

- original space lies in 1d manifold, even sparser in $\Re^3$
- higher d curvature gives improved linear separation
- can have *any* 3 break points (3 points $\Rightarrow$ plane)?
- relatively few "interesting separating planes"

Polynomial Embedding (cont.)

General View: for original data matrix:

$$\begin{pmatrix} x_{11} & & x_{1n} \\ \vdots & \cdots & \vdots \\ x_{d1} & & x_{dn} \end{pmatrix}$$

"add rows":

$$\begin{pmatrix} x_{11} & & x_{1n} \\ \vdots & & \vdots \\ x_{d1} & & x_{dn} \\ x_{11}^2 & \cdots & x_{1n}^2 \\ \vdots & & \vdots \\ x_{d1}^2 & & x_{dn}^2 \\ x_{11}x_{21} & & x_{1n}x_{2n} \\ \vdots & & \vdots \end{pmatrix}$$

Polynomial Embedding (cont.)

Now apply linear methods (FLD, SVM, …) in *embedded* space.

- image of class boundaries in original space is *nonlinear*

- allows much more *complicated* class regions

Polynomial Embedding Toy Examples

E.g. 1: Donut

- FLD: poor for low degree, then good

- SVM: similar excellent perfromance

Polynomial Embedding Toy Examples (cont.)

E.g. 2: Parallel Clouds

- FLD good for all embeddings

- SVM OK, but begin to see overfitting problems

Polynomial Embedding (cont.)

Drawback to polynomial embedding:

- extra terms may create spurious structure

- i.e. potential for "overfitting"

- High Dimension Low Sample Size problems worse

Kernel Machines

Idea: replace polynomials by other "nonlinear functions"

e.g. 1: "sigmoid functions" from neural nets

e.g. 2: "radial basis functions" – Gaussian kernels

Related to "kernel density estimation" (smoothed histogram)

Kernel Machines (cont.)

Radial basis functions: at some "grid points" $\underset{\rightarrow}{g}_1, \ldots, \underset{\rightarrow}{g}_k$,

For a "bandwidth" (i.e. standard deviation) σ,

Consider (d dim'al) functions: $\varphi_\sigma\left(\underset{\rightarrow}{x} - \underset{\rightarrow}{g}_1\right), \ldots, \varphi_\sigma\left(\underset{\rightarrow}{x} - \underset{\rightarrow}{g}_k\right)$

Replace data matrix with:

$$\begin{pmatrix} \varphi_\sigma\left(\underset{\rightarrow}{X}_1 - \underset{\rightarrow}{g}_1\right) & & \varphi_\sigma\left(\underset{\rightarrow}{X}_n - \underset{\rightarrow}{g}_1\right) \\ \vdots & \cdots & \vdots \\ \varphi_\sigma\left(\underset{\rightarrow}{X}_1 - \underset{\rightarrow}{g}_k\right) & & \varphi_\sigma\left(\underset{\rightarrow}{X}_n - \underset{\rightarrow}{g}_k\right) \end{pmatrix}$$

Kernel Machines (cont.)

For discrimination: work in radial basis function domain,

With new data vector $\underset{\rightarrow}{X}_0$ represented by: $\begin{pmatrix} \varphi_\sigma\left(\underset{\rightarrow}{X}_0 - \underset{\rightarrow}{g}_1\right) \\ \vdots \\ \varphi_\sigma\left(\underset{\rightarrow}{X}_0 - \underset{\rightarrow}{g}_1\right) \end{pmatrix}$

Kernel Machines (cont.)

Toy Examples:

E.g. 1: Donut – mostly good (slight mistake for one kernel)

E.g. 2: Parallel Clouds – good at data, poor outside

Main lesson: generally good in regions with data,
unpredictable results where data are sparse

Kernel Machines (cont.)

E.g. 7: Checkerboard

- Kernel embedding (FLD or SVM) is excellent
- While polynomials (FLD – SVM) lack flexibility
- Lower degree is worse

Kernel Machines (cont.)

$\exists$ generalizations of this idea to other types of analysis,

and some clever computational ideas.

E.g. "Kernel based, nonlinear Principal Components Analysis"

Schölkopf, Smola and Müller (1998) "Nonlinear component analysis as a kernel eigenvalue problem", *Neural Computation*, **10**, 1299-1319.

Discrimination (cont.)

M-rep Corpora Callosa Data:

Try to find differences between Schizophrenics and Controls

Most interesting views: Projections onto normal vector

Mean Difference FLD SVM

Verification: None signif'ly better than "random permutations"

Discrimination (cont.)

Paul Yushkevich Toy Data: (simulate from PCA, and a bump)

Raw Data   Raw Data with bumps

Discrimination Performance (again check projections):

Mean Difference: OK

FLD: Better, no overlap

SVM: Best? Or too much “piling at the margin”?

Also check direction: SVM

Independent Component Analysis

Our application:

Find directions of "least Gaussian" projections

Origins: "blind source extraction"

Motivating Example: Cocktail Party Problem

- Start with signals
- Do linear mixing
- Recover signals (without knowledge of mixing coeff's)

Independent Component Analysis (cont.)

How it works: Scatterplot views:

- Original Data
- Mixed Data & result of sphering

  (now rotate to "least Gaussian" directions)

- PCA gets it wrong, signals & scatterplot

Independent Component Analysis (cont.)

Recommended References:

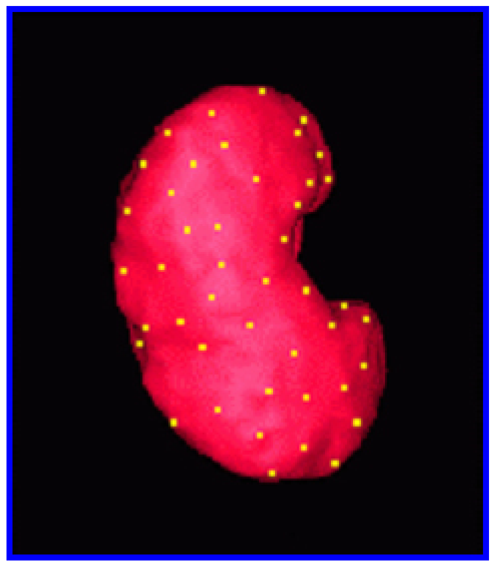
Hyvärinen and Oja (1999) *Independent Component Analysis: A Tutorial*, http://www.cis.hut.fi/projects/ica

Hyvärinen, A., Karhunen, J. and Oja, E. (2001) *Independent Component Analysis*, John Wiley & Sons.

Independent Component Analysis (cont.)

Goal (from James Chen):

Simulate kidney images to test segmentation



Simple Approach: Gaussian simulation from PCA

Independent Component Analysis (cont.)

Ticklish Question: Are data Gaussian?

Approach (joint work with Inge Koch and James Chen):

Look for non-Gaussianity, using ICA:

- Sphere Data
- Look for single “Least Gaussian Direction”
- Repeat, since algorithm depends on random start
- Results: find outliers in “several directions”
- Question: are these “spurious”?

Gaussianity Check

Approach:

- Simulate from the Gaussian
- Recompute ICA
- Compare data abs skewness with simulated values
- Result: shows clearly non-Gaussian

Distributional Fix

Idea: transform to fix above problem

(BIG) Assumption: Distribution is “radially symmetric”

Transformation: Power Transformation

Choose power to make “radii looks as expected for Gaussian”

Result: Raise radii to power $1/0.55 \approx 1.8$

Final Check: Apply above tests to simulated data

Exciting new area

FDA of populations of tree structured objects

Motivation:

- Current FDA methods are powerful
- but limited to populations of *fixed length* feature vectors
- can’t handle “variable topology shapes”
- severe limitation for “multifigural objects”

Exciting new area (cont.)

Challenging problem:

Statistical Analysis of Populations of Trees

A first approach to this slippery area: Haonan Wang

Careful axiomatic mathematics *required*!

(because "our intuition is too Euclidean")

FDA on Trees

30,000 foot view:

1. Start with a "metric" (distance measure)

2. Define "centerpoint" as "point closest to all data points"

3. Define PC1 as "simplest 1-d representations"

For details: Talk by Haonan