MSPR 4: PCA II

Dr. Hendrik Purwins

AAU CPH

September 28, 2015

This lecture is based on presentations made in my course 'Advanced Topics in in Music Technology' at Music Technology Group, Universitat Pompeu Fabra between 2007-2011, Barcelona, especially the ones by Stefan Kersten and Srikanth Cherla, closely following

- *Andrew Moore's machine learning tutorial lectures: Gaussians, Gaussian Mixture Models,* *http://www.autonlab.org/tutorials/*
- *Christopher Bishop: Pattern Recognition and Machine Learning: Chapter 1 (Introduction) 1.2.3 (The Gaussian Distribution) p. 24 - 27, 2.3 (The Gaussian Distribution) p. 78, 84 bottom - 85 top.*

Outline

Eigenvalues and Eigenvectors

Eigenvalues and Eigenvectors

Strang: Chapter 6.1: Introduction to Eigenvalues pp. 283-297

Eigenvalues and Eigenvectors

Generally the vector $\mathbf{x}$ *changes direction* when multiplied by $\mathbf{A}$...

However there are some special vectors $\mathbf{x}$ that are in the same direction as $\mathbf{A}\mathbf{x}$:

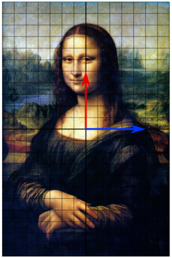
$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

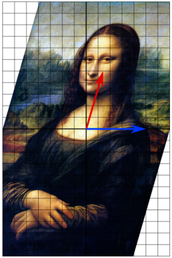
- such a vector $\mathbf{x}$ is an *eigenvector*
- λ is the *eigenvalue* corresponding to $\mathbf{x}$
- λ can is a number, it scales $\mathbf{x}$
- if $\mathbf{A} = \mathbf{I}$, $\mathbf{A}\mathbf{x} = \mathbf{x}$, i.e. all vectors are eigenvectors of $\mathbf{I}$

Example I

- Shearing: $\mathbf{A} = \begin{bmatrix} 1 & 0.25 \\ 0 & 1 \end{bmatrix}$
- $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ is an eigenvector with eigenvalue 1
- $\begin{bmatrix} 0 \\ -1 \end{bmatrix}$ is another eigenvector with eigenvalue 1

http://en.wikipedia.org/wiki/File:Mona_Lisa_eigenvector_grid.png





Example II

Video: `http://en.wikipedia.org/wiki/File:Eigenvectors.gif`

- Transformation matrix
$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$
- Example of eigenvectors: $\begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ with corresponding eigenvalues $1, 3$.

Eigenvalue Decomposition for Symmetric Matrices

Eigenvalue Decomposition for Symmetric Matrices

Strang: Chapter 6.2: Diagonalizing a Matrix pp. 298-311, Chapter 6.4: Symmetric Matrices

Eigenvalue Decomposition for Symmetric Matrices

Definition

An *orthonormal* matrix V is a matrix in which

- all column vectors (row vectors) are orthogonal to each other, i.e. $\mathbf{v} \cdot \mathbf{w} = 0$ for $\mathbf{v} \neq \mathbf{w}$
- all column vectors (row vectors have length 1, i.e. $\|\mathbf{v}\| = 1$
- $\Rightarrow \mathbf{V}^T\mathbf{V} = \mathbf{I} = \mathbf{V}^{-1}\mathbf{V} \Rightarrow \mathbf{V}^{-1} = \mathbf{V}^T$

Theorem

(*Eigenvalue decomposition for symmetric matrices:*) Every symmetric matrix has the matrix decomposition $\mathbf{A} = \mathbf{V\Lambda V}^T$ with real eigenvalues in $\mathbf{\Lambda}$ and orthogonal eigenvectors in $\mathbf{V}$:

$$\mathbf{A} = \mathbf{V\Lambda V}^{-1} = \mathbf{V\Lambda V}^T \qquad \text{with} \qquad \mathbf{V}^{-1} = \mathbf{V}^T (Matlab: [\mathtt{V}\ \Lambda] = \mathtt{eig(A)})$$

Eigenvalue Decomposition for Symmetric Matrices

The *eigenvector matrix* $\mathbf{V}$ has the eigenvectors $\mathbf{v}_1 \ldots \mathbf{v}_n$ as columns.

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \ldots & \mathbf{v}_n \end{bmatrix}$$

The *eigenvalue matrix* $\mathbf{\Lambda}$ has the eigenvalues as diagonal elements in corresponding order:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

Eigenvalue Decomposition for Symmetric Matrices

Remark on Diagonalization

- The order of eigenvectors in $\mathbf{V}$ corresponds to the order of eigenvalues in $\mathbf{\Lambda}$ (see proof on diagonalization)

Class Assignment

Find the eigenvectors $\mathbf{v}_1, \mathbf{v}_2$ *and corresponding eigenvalue(s)* λ_1, λ_2 *for*

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

and sort them according to a descending order of the eigenvalues. Use the following Matlab functions (and others):

- `a=diag(A)` *returns the diagonal of matrix* `A` *as a vector* `a`
- `[x idx]=sort(x,'descend'); A=A(:,idx)` *sorts the elements of* x *in descending order. It also sorts the columns of* A *in descending order of the corresponding elements of* x*.*

Class Assignment

Perform an eigenvalue decomposition of the sample covariance matrix **C***of covariances between the features (sepal length / width, petal length / width). Sort the eigenvalues in descending order, rearrange the eigenvalues and the eigenvectors in that order and plot the eigenvalues. Use the following Matlab functions (and others):*

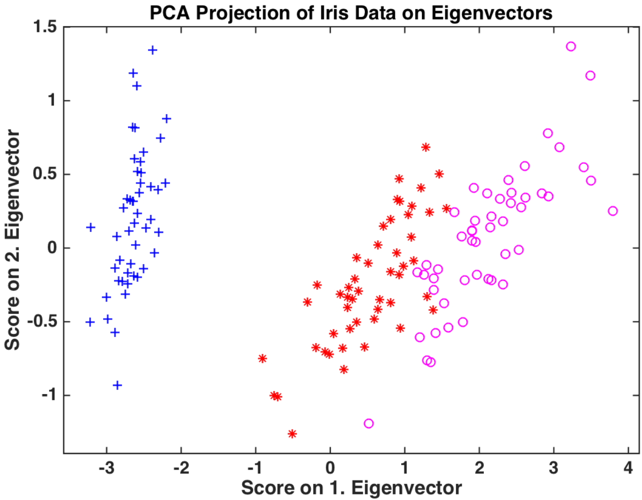
- `a=diag(A)` *returns the diagonal of matrix* `A` *as a vector* `a`
- `[x idx]=sort(x,'descend'); A=A(:,idx)` *sorts the elements of x in descending order. It also sorts the columns of A in descending order of the corresponding elements of x.*

Class Assignment

Display the Eigenvectors over the four features (sepal length / width, petal length / width) using `plot` *and* `hold on` *and explain how they relate to the features.*

Class Assignment

Center the data via: $X_c = X - \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \bar{\mathbf{x}}$ *where the vertical vector consisting of* I *ones (number of observations/rows in observation matrix* $\mathbf{X}$*) and* $\bar{\mathbf{x}} = (\bar{\mathbf{X}}_{[:,1]}, \bar{\mathbf{X}}_{[:,2]}, \ldots, \bar{\mathbf{X}}_{[:,J]})$ *is the vector of column means of features* $1, 2, \ldots, J$. *Calculate the scores, i.e. the marks on the axes of the new coordinate system (= the eigenvectors) via:* $\mathbf{S} = \mathbf{X}_c \mathbf{V}_{[:,1:2]}$, *where* $\mathbf{V}_{[:,1:2]}$ *is the matrix that consists of just the first two columns (eigenvectors) of eigenvector matrix* $\mathbf{V}$ *of sample covariance matrix* $\mathbf{C}$ *with corresponding 2 highest eigenvectors. Then display the scores of all iris instances. Mark each iris species in a different color. Use the prtools function* `scatterd` *for plotting.*


PCA Projection of Iris Data on Eigenvectors
Score on 2. Eigenvector
Score on 1. Eigenvector
1.5
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3
4

The *Inertia* (average square norm of all points) for set of column-centered points $\mathbf{X}_{c[1,:]}, \mathbf{X}_{c[2,:]}, \ldots, \mathbf{X}_{c[I,:]}$ is a measure for the variance in the data:

$$\frac{1}{I} \sum_{i=1}^{I} \left\| \mathbf{X}_{c[i,:]} \right\|^2 .$$

Consider the scores $\mathbf{S} = \mathbf{X}_c \mathbf{V}_{[:,1:j]}$, where $\mathbf{V}_{[:,1:j]}$ is the matrix that consists of just the j columns (eigenvectors) of eigenvector matrix $\mathbf{V}$ of sample covariance matrix $\mathbf{C}$ that correspond to its j highest eigenvalues. We can measure how much variance of the original data $\mathbf{X}_{[i,:]}$ is kept in the scores $\mathbf{S}_{[i,:]}$ by calculating the *inertia (data variance) quotient*:

$$\tau := \frac{\sum_{i=1}^{I} \|\mathbf{S}_{[i,:]}\|^2}{\sum_{i=1}^{I} \|\mathbf{X}_{[i,:]}\|^2}$$

Principal component analysis (eigenvalue decomposition on the sample covariance matrix of the data) provides the projection with best variance preservation.

Class Assignment

1. *Calculate the inertia quotient τ between the scores and the column centered iris data.*
2. *Calculate the cumulative sum of eigenvalue percentage for the two ($k = 2$) highest eigenvalues λ_1, λ_2 according to: $c_k = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{J} \lambda_j}$ and compare it with the inertia quotient calculated before.*
3. *Calculate and plot the cumulative sum of eigenvalue percentage c_k for the $k = 1, 2, 3, 4$ highest eigenvalues using Matlab function* `cumsum` *for calculating cumulative sums.*

Class Assignment

(Principal Component Analysis for Compression and Reconstruction) *Reconstruct the original Iris data, using the following reconstruction equation for scores* $\mathbf{S}$ *and* $\mathbf{V}'_{[:,1:j]}$ *whos columns are are the* j *eigenvectors corresponding to the highest* j *eigenvalues of the sample covariance matrix of the data:*

$$\mathbf{X}_r = \mathbf{S}\mathbf{V}'_{[:,1:j]} + \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \bar{\mathbf{x}}$$

where the vertical vector consisting of I *ones (number of observations/rows in observation matrix* $\mathbf{X}$*) and* $\bar{\mathbf{x}} = (\bar{\mathbf{X}}_{[:,1]}, \bar{\mathbf{X}}_{[:,2]}, \ldots, \bar{\mathbf{X}}_{[:,J]})$ *is the vector of column means of features* $1, 2, \ldots, J$*.*

Principal Component Analysis Cookbook I

1. *Data matrix*: I observations of J features:

$$\mathbf{X} := \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1J} \\ x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2J} \\ x_{31} & x_{32} & \cdots & x_{3j} & \cdots & x_{3J} \\ \vdots & \vdots & \ldots & \vdots & \ldots & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{iJ} \\ & & \cdots & & \cdots & \\ x_{I1} & x_{I2} & \cdots & x_{Ij} & \cdots & x_{IJ} \end{pmatrix}$$

2. Calculate: *sample mean vector* $\bar{\mathbf{x}}$ (Matlab: `mean(X)`):

$$\bar{\mathbf{x}} = \frac{1}{I} \underbrace{(1, 1 \ldots, 1)}_{I\times} \mathbf{X}$$

3. *Centerize data* column/featurewise via: $X_c = X - \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \bar{\mathbf{x}}$

Principal Component Analysis Cookbook II

1. Calculate *sample covariance*:

$$\mathbf{C} = \frac{1}{l-1}\mathbf{X}_c^T\mathbf{X}_c$$

   Matlab: `C=cov(X)`

2. *Eigenvalue decomposition for symmetric matrices*: $\mathbf{C}$ is symmetric: $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ with eigenvectors as the columns of $\mathbf{V}$ and the corresponding eigenvectors in the same order on the diagonal of $\mathbf{\Lambda}$
   Matlab: `[V   D]=eig(C)`

3. Pick eigenvalues from diagonal of $\mathbf{\Lambda}$ and form a vector λ:
   Matlab: `eigval=diag(D);`

Principal Component Analysis Cookbook III

1. *Sort eigenvectors* and eigenvalues in descending order: Matlab: `[eigval idx]=sort(eigval,'descend'); V=V(:,idx);`
2. *[Plot eigenvectors]* and interpret them in terms of the original features, they are a linear combination of. Matlab: `plot (V(: , j ) )`
3. Calculate and plot the *cumulative sum of eigenvalue percentage* :
   $c_k = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{J} \lambda_j}$

```
cs=cumsum(eigval)/sum(eigval); plot(cs)
```

The eigenvector corresponding to the highest eigenvalue points in the direction of maximal variance. Check the graph for a 'knee' to decide how many dimensions k to keep.

Principal Component Analysis Cookbook IV

1. *Reduction to k dimensions:* Calculate the scores via: $\mathbf{S} = \mathbf{X}_c \mathbf{V}_{[:,1:k]}$, where $\mathbf{V}_{[:,1:k]}$ is the matrix that consists of just the first k columns (eigenvectors) of eigenvector matrix $\mathbf{V}$ of sample covariance matrix C with corresponding k highest eigenvalues. Matlab:

```
scores=Xc*V(: ,1:k)
```

2. *Visualize scores* (2d for the 2 highest eigenvalues) Matlab (if the data have labels):

```
z=prdataset(scores , labels); scatterd(z)
```

Principal Component Analysis Cookbook V

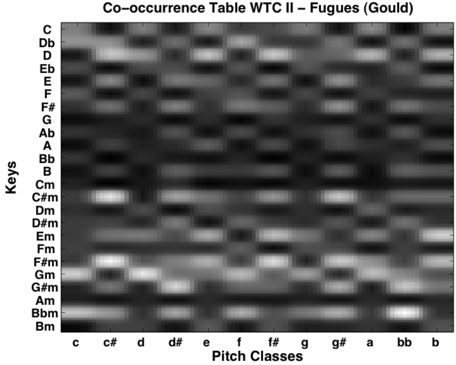
1. *[Reconstruct data]* from the compressed representation:

$$\mathbf{X}_r = \mathbf{S}\mathbf{V}'_{[:,1:k]} + \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \bar{\mathbf{x}}$$

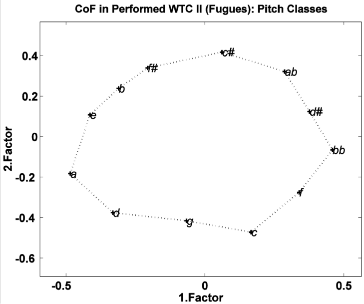
```
Xr=scores*V(:,1:k) +ones(I,1)*mean(X);
```

Relations between Pitch Classes in Bachs Wohltemperiertes Klavier II

- Bach: Wohltemperiertes Klavier, Part II (Fugen), Glenn Gould (Audio) Greyscale matrix: estimated strength of 12 pitch clases in all 24 keys (CQ-Profile, Purwins et al. 2000) Pitch classes: 24-dimensional frequency vector in each key
- Dimension reduction (correspondence analysis: Greenacre, 1984) maps 24 dimensions to 2 dimensions, to preserve structural relationships between keys.



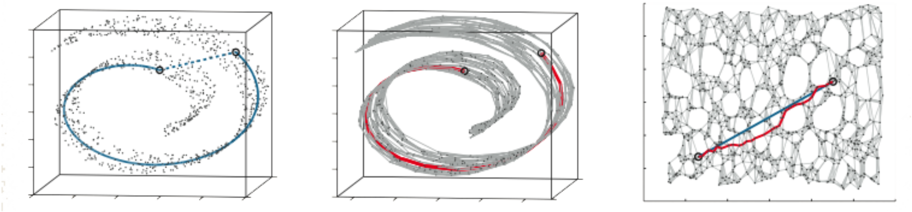
Visualisation of Pitch Classes in Bach's WT II, Fugues, recording Glenn Gould



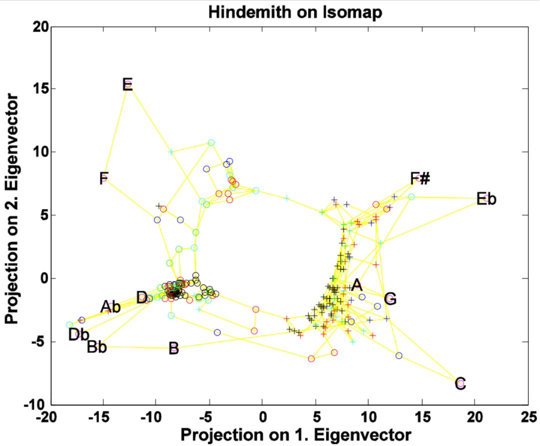
Circle of fifths emerges! Check out Matlab dimension reduction toolbox:
`http://lvdmaaten.github.io/drtoolbox/`

Isomap

Non-linear Mapping: Geodesic distance + PCA



Musical Style Visualization



'Landscapes' of all pieces: Bach (black), Chopin (red), Alkan (cyan), Scriabin (green), Shostakovich (blue), Hindemith (magenta, outlier) marked, 'o'/'+:Major/minor (Purwins et al. 2004)