Principal Component Analysis

an introduction

alexander lerch

github.com/alexanderlerch/2025-PCA-Lectu

about
about me

- **education**
  - Electrical Engineering (Technical University, Berlin)
  - Tonmeister (music production, University of Arts, Berlin)

- **professional**
  - Professor, School of Music, Georgia Tech
  - Associate Dean for Research & Creative Practice, College of Design, Georgia Tech
  - 2000-2013: CEO at zplane.development

- **experience**
  - audio algorithm design (20+ years)
  - machine learning for music (15+ years)
  - professional music software engineering & development (10+ years)
  - entrepreneurship (10+ years)
  - research administration (2+ years)

www.linkedin.com/in/lerch

introduction

principal component analysis

goal: data reduction while maximum information content

introduction

pca use cases

- dimensionality reduction
  - overfitting, curse of dimensionality
  - visualization of high-dimensional spaces
  - performance/runtime
- remove feature correlation, redundancy
- feature analysis

principal component analysis
introduction

- **objective**
  - map features to new coordinate system

$$\boldsymbol{u}(n) = \boldsymbol{T}^{\mathrm{T}} \cdot \boldsymbol{v}(n)$$

    - $\boldsymbol{u}(n)$: transformed features (same dimension as input $\boldsymbol{v}(n)$)
    - $\boldsymbol{T}$: transformation matrix $(\mathcal{F}_v \times \mathcal{F}_v)$

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{c}_0 & \boldsymbol{c}_1 & \dots & \boldsymbol{c}_{\mathcal{F}-1} \end{bmatrix}$$

- **properties**
  - $\boldsymbol{c}_0$ points in the direction of highest *variance* (sorted by eigenvalue)
  - variance concentrated in as few output components as possible
  - $\boldsymbol{c}_i$ orthogonal

$$\boldsymbol{c}_i^{\mathrm{T}} \cdot \boldsymbol{c}_j = 0 \quad \forall\ i \neq j$$

  - transformation is invertible

$$\boldsymbol{v}(n) = \boldsymbol{T} \cdot \boldsymbol{u}(n)$$

principal component analysis

introduction

- **objective**
  - map features to new coordinate system

$$\boldsymbol{u}(n) = \boldsymbol{T}^{\mathrm{T}} \cdot \boldsymbol{v}(n)$$

    - $\boldsymbol{u}(n)$: transformed features (same dimension as input $\boldsymbol{v}(n)$)
    - $\boldsymbol{T}$: transformation matrix ($\mathcal{F}_v \times \mathcal{F}_v$)

$$\boldsymbol{T} = \left[\begin{array}{cccc} \boldsymbol{c}_0 & \boldsymbol{c}_1 & \ldots & \boldsymbol{c}_{\mathcal{F}-1} \end{array}\right]$$

- **properties**
  - $\boldsymbol{c}_0$ points in the direction of highest *variance* (sorted by eigenvalue)
  - variance concentrated in as few output components as possible
  - $\boldsymbol{c}_i$ orthogonal

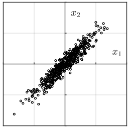
$$\boldsymbol{c}_i^{\mathrm{T}} \cdot \boldsymbol{c}_j = 0 \quad \forall \; i \neq j$$

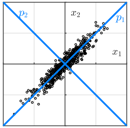
  - transformation is invertible

$$\boldsymbol{v}(n) = \boldsymbol{T} \cdot \boldsymbol{u}(n)$$

feature space transformation

PCA visualization





calculation of the transformation matrix

1. normalize input data
2. compute covariance matrix $\boldsymbol{R}$

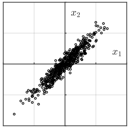
$$\boldsymbol{R} = \mathcal{E}\{(V - \mathcal{E}\{V\})(V - \mathcal{E}\{V\})\}$$

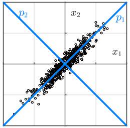
3. compute eigenvectors and eigenvalues
4. sort eigenvectors by eigenvalue and use as axes for the new coordinate system
5. apply transformation matrix

matlab source: plotPca.m

feature space transformation

PCA visualization





calculation of the transformation matrix

1. normalize input data
2. compute covariance matrix $\boldsymbol{R}$

$$\boldsymbol{R} = \mathcal{E}\{(V - \mathcal{E}\{V\})(V - \mathcal{E}\{V\})\}$$

3. compute eigenvectors and eigenvalues
4. sort eigenvectors by eigenvalue and use as axes for the new coordinate system
5. apply transformation matrix

matlab source: plotPca.m

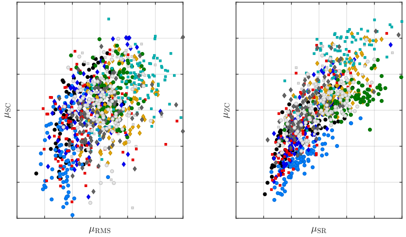
feature space transformation

PCA visualization

PCA examples

example 1

- 4-dimensional feature space: Spectral Centroid, RMS, Zero Crossing, Spectral Rolloff
- 10 classes of music signals (classical, jazz, blues, metal, ...)

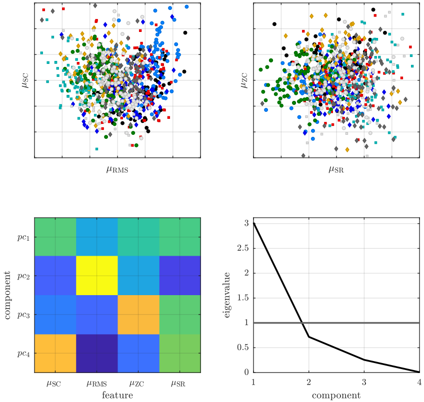


matlab source: plotFeatureScatter.m

PCA examples

example 1

- 4-dimensional feature space: Spectral Centroid, RMS, Zero Crossing, Spectral Rolloff
- 10 classes of music signals (classical, jazz, blues, metal, ...)

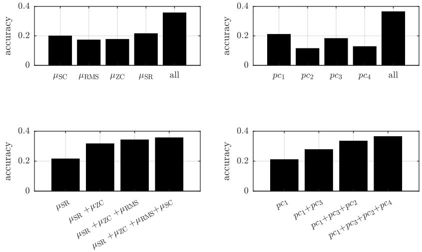


matlab source: plotFeatureScatterPca.m

PCA examples

example 1

- **observations**
  - combined feature performance is not sum of individual performance
  - variance/eigenvalue ranking does not necessarily correlate to task performance
  - simple feature selection is not necessarily inferior to PCA



matlab source: plotPcaClassification.m

PCA examples

example 2

PCA transformation matrix $\mathbf{T}^{\mathrm{T}}$

$$\begin{bmatrix} -0.5638 & -0.3596 & -0.5024 & -0.5481 \\ 0.1738 & -0.9139 & 0.3539 & 0.0965 \\ 0.2408 & -0.1882 & -0.7606 & 0.5728 \\ -0.7707 & -0.0018 & 0.2096 & 0.6017 \end{bmatrix}$$

PCA examples

example 2

PCA transformation matrix $\mathbf{T}^{\mathrm{T}}$

$$\begin{bmatrix} -0.5638 & -0.3596 & -0.5024 & -0.5481 \\ 0.1738 & -0.9139 & 0.3539 & 0.0965 \\ 0.2408 & -0.1882 & -0.7606 & 0.5728 \\ -0.7707 & -0.0018 & 0.2096 & 0.6017 \end{bmatrix}$$

conclusion

conclusion

thank you!

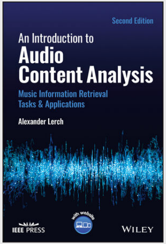
links

alexander lerch: www.linkedin.com/in/lerch

mail: alexander.lerch@gatech.edu

book: www.AudioContentAnalysis.org

music informatics group: musicinformatics.gatech.edu



github.com/alexanderlerch

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page will go away, becau
LaTeX now knows how many pages to expect for this document.