Principal Component Analysis

an introduction

alexander lerch

github.com/alexanderlerch/2025-PCA-Lectu

about

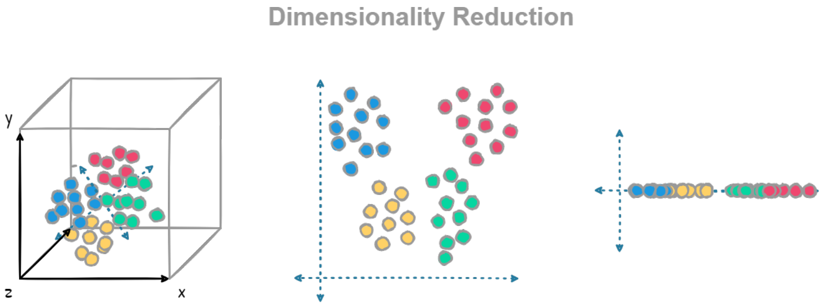
about me

- **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

dimensionality reduction



https://blog.roboflow.com/what-is-dimensionality-reduction

typical goals:

- reduce overfitting, curse of dimensionality
- visualize high-dimensional spaces
- increase performance/runtime

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

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

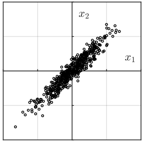
principal component analysis
calculation

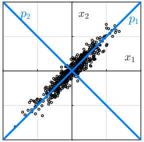
calculation of the transformation matrix

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. (remove irrelevant components and) apply transformation matrix





matlab source: plotPca.m

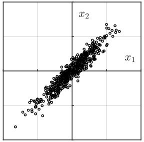
principal component analysis
calculation

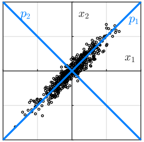
calculation of the transformation matrix

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. (remove irrelevant components and) apply transformation matrix





properties:

- linear transformation
- resulting principal components are
  - uncorrelated
  - sorted (by variance)

matlab source: plotPca.m

principal component analysis

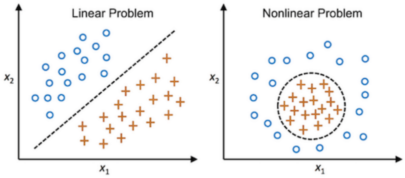
drawbacks

- no **component interpretability**: principal components are 'unintuitive' combinations of input features
- **linear data** only: nonlinear relationships between inputs are ignored
- **sorting criteria** not necessarily task-relevant
- can be **affected by outliers**
- unclear **optimum number** of resulting components
  - eigenvalue > 1
  - cumulative variance $> 95\%$
  - elbow in scree plot

principal component analysis

drawbacks

- no **component interpretability**: principal components are 'unintuitive' combinations of input features
- **linear data** only: nonlinear relationships between inputs are ignored
- **sorting criteria** not necessarily task-relevant
- can be **affected by outliers**
- unclear **optimum number** of resulting components
  - eigenvalue > 1
  - cumulative variance > 95%
  - elbow in scree plot

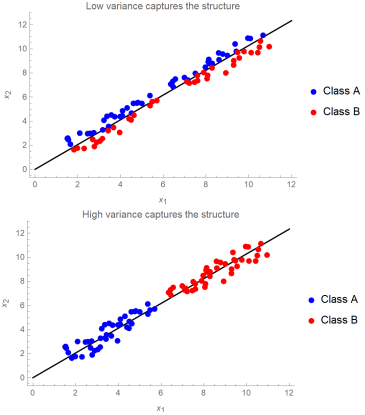


medium.com/@sakethyalamanchili/pca-vs-kpca-which-technique-will-work-best-for-your-dat

principal component analysis

drawbacks

- no **component interpretability**: principal components are 'unintuitive' combinations of input features
- **linear data** only: nonlinear relationships between inputs are ignored
- **sorting criteria** not necessarily task-relevant
- can be **affected by outliers**
- unclear **optimum number** of resulting components
  - eigenvalue > 1
  - cumulative variance $> 95\%$
  - elbow in scree plot

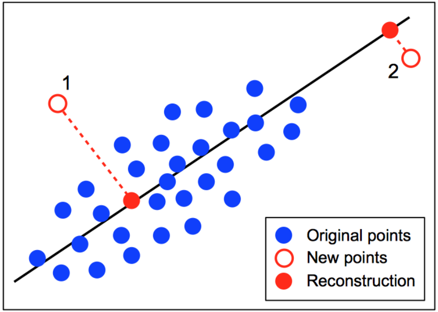


ekamperi.github.io/mathematics/2021/02/23/pca-limitations.html

principal component analysis

drawbacks

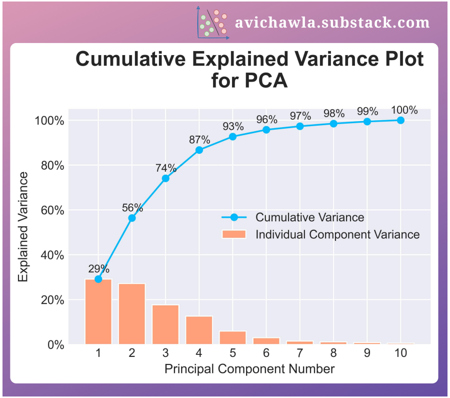
- no **component interpretability**: principal components are 'unintuitive' combinations of input features
- **linear data** only: nonlinear relationships between inputs are ignored
- **sorting criteria** not necessarily task-relevant
- can be **affected by outliers**
- unclear **optimum number** of resulting components
  - eigenvalue > 1
  - cumulative variance $> 95\%$
  - elbow in scree plot



stats.stackexchange.com/questions/259806/anomaly-detection-using-pca-reconstruction-er

principal component analysis
drawbacks

- no **component interpretability**: principal components are 'unintuitive' combinations of input features
- **linear data** only: nonlinear relationships between inputs are ignored
- **sorting criteria** not necessarily task-relevant
- can be **affected by outliers**
- unclear **optimum number** of resulting components
  - eigenvalue > 1
  - cumulative variance $> 95\%$
  - elbow in scree plot

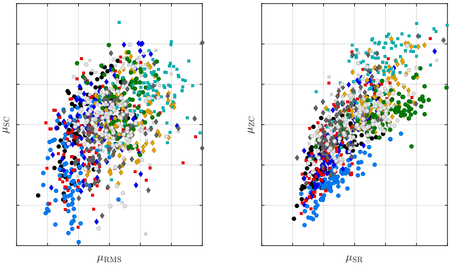


blog.dailydoseofds.com/p/how-many-dimensions-should-you-reduce

PCA examples

example 1: scatter and classification

- 4-dimensional feature space:
  - Spectral Centroid μ_{SC}
  - RMS μ_{RMS}
  - Zero Crossing μ_{ZC}
  - Spectral Rolloff μ_{SR}
- 10 classes of music signals
  - blues
  - classical
  - country
  - disco
  - hiphop
  - jazz
  - metal
  - pop
  - reggae
  - rock

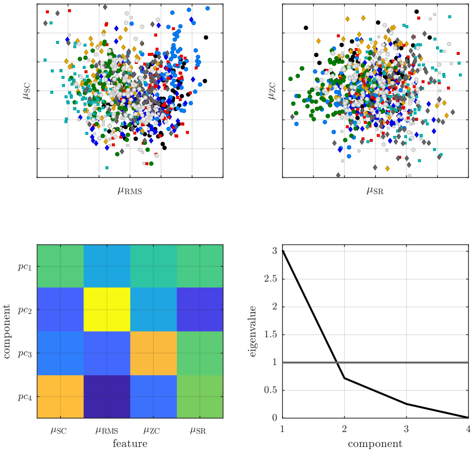


matlab source: plotFeatureScatter.m

PCA examples

example 1: scatter and classification

- 4-dimensional feature space:
  - Spectral Centroid μ_{SC}
  - RMS μ_{RMS}
  - Zero Crossing μ_{ZC}
  - Spectral Rolloff μ_{SR}
- 10 classes of music signals
  - blues
  - classical
  - country
  - disco
  - hiphop
  - jazz
  - metal
  - pop
  - reggae
  - rock



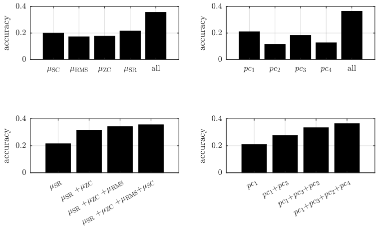
matlab source: plotFeatureScatterPca.m

PCA examples

example 1: scatter and classification

- **experiment: feature/pc classification performance**
  - individual
  - cumulative

- **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



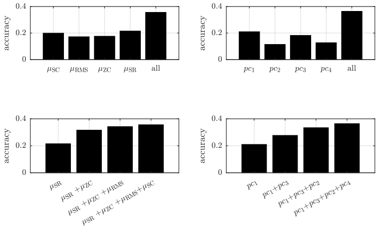
matlab source: plotPcaClassification.m

PCA examples

example 1: scatter and classification

- **experiment: feature/pc classification performance**
    - individual
    - cumulative

- **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



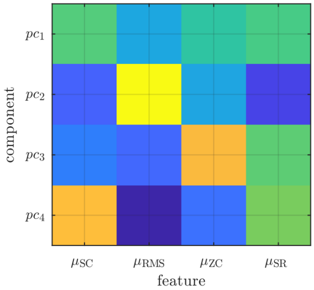
matlab source: plotPcaClassification.m

PCA examples

example 2: feature analysis

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

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

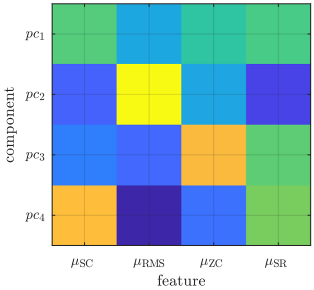


PCA examples

example 2: feature analysis

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

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



principal component analysis

conclusion

- **well-known** tool for dimensionality reduction
- widely **implemented**
- **convenient** characteristics (sorting of components, uncorrelated output)
- common **user errors**
  - missing or incorrect normalization
  - final number of components improperly set
  - classification tasks: computing the transformation matrix from full data and not training data
- **potential problems**
  - nonlinear relationships between inputs
  - interpretability of components
  - impacted by outliers

principal component analysis

conclusion

- **well-known** tool for dimensionality reduction
- widely **implemented**
- **convenient** characteristics (sorting of components, uncorrelated output)
- common **user errors**
  - missing or incorrect normalization
  - final number of components improperly set
  - classification tasks: computing the transformation matrix from full data and not training data
- **potential problems**
  - nonlinear relationships between inputs
  - interpretability of components
  - impacted by outliers

principal component analysis

conclusion

- **well-known** tool for dimensionality reduction
- widely **implemented**
- **convenient** characteristics (sorting of components, uncorrelated output)
- common **user errors**
  - missing or incorrect normalization
  - final number of components improperly set
  - classification tasks: computing the transformation matrix from full data and not training data
- **potential problems**
  - nonlinear relationships between inputs
  - interpretability of components
  - impacted by outliers

thank you!

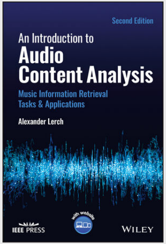
links

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