ML&MEA (2024)
Lecture 10 - Principal Component Analysis

Quanying Liu

BME, SUSTech

2024.4.2

Content

SVM Soft Margin + Kernel trick

Dual Problem with Soft Margin and Kernel Trick:

$$\text{Maximize: } Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \underbrace{\varphi^T(\mathbf{x}_i)\varphi(\mathbf{x}_j)}_{K(\mathbf{x}_i,\mathbf{x}_j)}$$

$$\text{s.t. } \sum_{i=1}^{N} \alpha_i d_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq \lambda$$

Mercer's condition:
Gram matrix K is *positive semi-definite* (i.e., its eigenvalues are nonnegative).

Redundancy in the data

- We have the measured data, $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N\times p}$, where N and p are the number of samples and features, respectively.

Redundancy in the data

- We have the measured data, $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times p}$, where N and p are the number of samples and features, respectively.
- Example: X is our ML course records. The p features includes age, sex, attendance, attention in the course, interactions, homework, course projects, and so on.

Redundancy in the data

- We have the measured data, $X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times p}$, where N and p are the number of samples and features, respectively.
- Example: X is our ML course records. The p features includes age, sex, attendance, attention in the course, interactions, homework, course projects, and so on.
- We notice that some features are correlated, such as the attention and interactions. $\longrightarrow$ The features have redundancy.

Redundancy in the data

- We have the measured data, $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times p}$, where N and p are the number of samples and features, respectively.
- Example: X is our ML course records. The p features includes age, sex, attendance, attention in the course, interactions, homework, course projects, and so on.
- We notice that some features are correlated, such as the attention and interactions. $\longrightarrow$ The features have redundancy.
- **Q: How can we remove the redundancy in the data?**

Redundancy in the data

- We have the measured data, $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times p}$, where N and p are the number of samples and features, respectively.
- Example: X is our ML course records. The p features includes age, sex, attendance, attention in the course, interactions, homework, course projects, and so on.
- We notice that some features are correlated, such as the attention and interactions. $\longrightarrow$ The features have redundancy.
- **Q: How can we remove the redundancy in the data?**
  1. Reduce the high-dimensional data space to a lower-dimensional latent space, while maximally maintain the information in the data. $\longrightarrow$ **Dimensionality Reduction**

Redundancy in the data

- We have the measured data, $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times p}$, where N and p are the number of samples and features, respectively.
- Example: X is our ML course records. The p features includes age, sex, attendance, attention in the course, interactions, homework, course projects, and so on.
- We notice that some features are correlated, such as the attention and interactions. $\longrightarrow$ The features have redundancy.
- **Q: How can we remove the redundancy in the data?**
  1. Reduce the high-dimensional data space to a lower-dimensional latent space, while maximally maintain the information in the data. $\longrightarrow$ **Dimensionality Reduction**
  2. Extract the most informative features from the original data. $\longrightarrow$ **Feature Extraction**

Feature extraction and *Dimensionality reduction*

- (Wikipedia) In machine learning, pattern recognition, and image processing, *feature extraction* starts from an initial set of measured data and builds derived values (features) intended to be **informative and non-redundant**, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations.

Feature extraction and *Dimensionality reduction*

- (Wikipedia) In machine learning, pattern recognition, and image processing, *feature extraction* starts from an initial set of measured data and builds derived values (features) intended to be **informative and non-redundant**, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations.
- (Wikipedia) *Dimensionality reduction*, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some **meaningful** properties of the original data, ideally close to its intrinsic dimension.

Feature extraction and *Dimensionality reduction*

- (Wikipedia) In machine learning, pattern recognition, and image processing, *feature extraction* starts from an initial set of measured data and builds derived values (features) intended to be **informative and non-redundant**, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations.
- (Wikipedia) *Dimensionality reduction*, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some **meaningful** properties of the original data, ideally close to its intrinsic dimension.
- Feature extraction is related to dimensionality reduction.

Feature extraction and *Dimensionality reduction*

- (Wikipedia) In machine learning, pattern recognition, and image processing, *feature extraction* starts from an initial set of measured data and builds derived values (features) intended to be **informative and non-redundant**, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations.
- (Wikipedia) *Dimensionality reduction*, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some **meaningful** properties of the original data, ideally close to its intrinsic dimension.
- Feature extraction is related to dimensionality reduction.
- **Q: What are the benefits to reduce dimension?**

Motivations

- Algorithms are not effective to deal with high-D data

Motivations

- Algorithms are not effective to deal with high-D data
  1. Performance decreases: Curse of dimensionality (维度灾难)

Motivations

- Algorithms are not effective to deal with high-D data
  1. Performance decreases: Curse of dimensionality (维度灾难)
  2. Computational cost increases (i.e., linear kernel in SVM, $O(d^2)$ with d dimensions)

Motivations

- Algorithms are not effective to deal with high-D data
  1. Performance decreases: Curse of dimensionality (维度灾难)
  2. Computational cost increases (i.e., linear kernel in SVM, $O(d^2)$ with d dimensions)
- Patterns in the data may have low intrinsic dimensions.

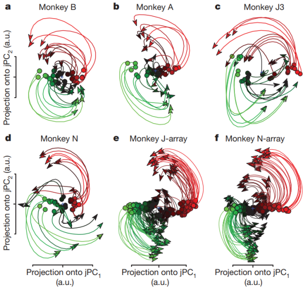


图 1: Low-dimensional neuronal representation in Macaques during reaching task. (Churchland et al. Nature, 2012)

Motivations

- Visualization: high-dimensional data $\longrightarrow$ 2D or 3D space.

Motivations

- Visualization: high-dimensional data $\longrightarrow$ 2D or 3D space.
- Data compression: efficient storage and retrieval

Motivations

- Visualization: high-dimensional data $\longrightarrow$ 2D or 3D space.
- Data compression: efficient storage and retrieval
- Noise reduction: to remove noise in high-D data

Motivations

- Visualization: high-dimensional data $\longrightarrow$ 2D or 3D space.
- Data compression: efficient storage and retrieval
- Noise reduction: to remove noise in high-D data
- Better interpretation: easy to understand the low-D features

Motivations

- Visualization: high-dimensional data $\longrightarrow$ 2D or 3D space.
- Data compression: efficient storage and retrieval
- Noise reduction: to remove noise in high-D data
- Better interpretation: easy to understand the low-D features
- Performance: better prediction, better generalization

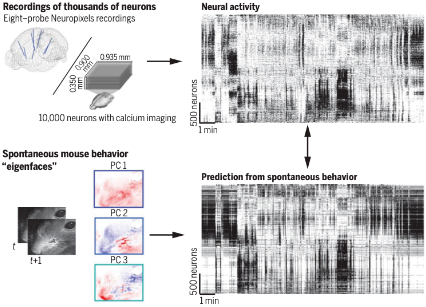


图 2: Large-scale neural population recordings can be predicted from behavior. (Stringer et al. Science, 2019)

Motivations

- Visualization: high-dimensional data ⟶ 2D or 3D space.
- Data compression: efficient storage and retrieval
- Noise reduction: to remove noise in high-D data
- Better interpretation: easy to understand the low-D features
- Performance: better prediction, better generalization

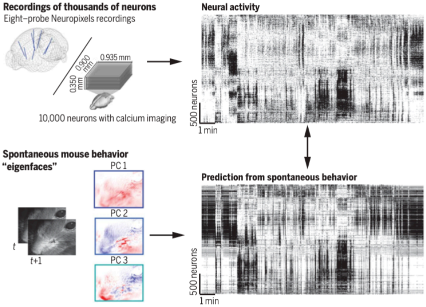


图 2: Large-scale neural population recordings can be predicted from behavior. (Stringer et al. Science, 2019)

An Observation on Generalization https://www.youtube.com/watch?v=AKMuA_TVz3A (Ilya Sutskever)

Dimensionality Reduction methods

1. **Unsupervised** (without class labels)

   Goal: *to minimize information loss*

   - Principal Component Analysis (PCA，主成分分析)
   - Nonnegative Matrix Factorization (NMF，非负矩阵分解)
   - Independent Component Analysis (ICA，独立成分分析)
   - T-distributed Stochastic Neighbor Embedding (t-SNE)
   - Multidimensional Scaling (MDS)
   - Uniform Manifold Approximation and Projection (UMAP)
   - Autoencoder (自编码器)

2. **Supervised** (with class labels)

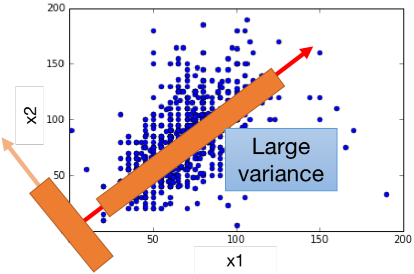
   Goal: *to maximize discrimination between classes*

   - Linear Discriminant Analysis (LDA，线性判别分析)
   - Canonical Correlation Analysis (CCA，典型相关分析)
   - Convolutional Neural Network (CNN，卷积神经网络)

1. Recap
2. Dimensionality Reduction
3. **PCA: methodology**
4. PCA: application
5. Summary

What is Principal Component Analysis?

- (Wikipedia) **PCA** is the process of computing the principal components (PCs) and using them to perform a change of **basis** on the data. (转换坐标系: orthonormal basis)
- **Variance of samples**. The principal components are ordered by the variance of PCs.
- **Goals**: 最大化投影方差 $\Longleftrightarrow$ 最小化重构距离



Mean and Covariance

- **Data**: $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times p}$
  N and p are the number of samples and features
  $\mathbf{x}_i \in \mathbb{R}^{p \times 1}$, with $i = 1,\ 2, \ldots,\ N$.
- **Mean** of $\mathbf{X}$: $(\bar{\mathbf{x}} \in \mathbb{R}^p)$

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i = \frac{1}{N} \mathbf{X}^T \mathbf{1}_N \tag{1}$$

where $\mathbf{1}_N = (1,\ 1, \ldots,\ 1)^T$. (N 个 1 的列向量)

- **Covariance** of $\mathbf{X}$: $(S \in \mathbb{R}^{p \times p})$

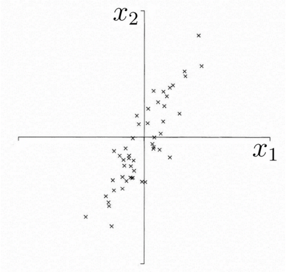
$$S = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{2}$$

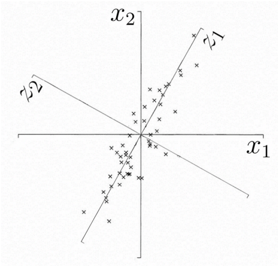
Covariance: in matrix calculus

- **Data**: $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times p}$
- **Covariance** of $\mathbf{X}$: $(S \in \mathbb{R}^{p \times p})$

$$
\begin{aligned}
S &= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \\
&= \frac{1}{N}\underbrace{(\mathbf{x}_1 - \bar{\mathbf{x}},\ \mathbf{x}_2 - \bar{\mathbf{x}}, \ldots,\ \mathbf{x}_N - \bar{\mathbf{x}})}_{(\mathbf{x}_1,\ \mathbf{x}_2, \ldots,\ \mathbf{x}_N) - (\bar{\mathbf{x}},\ \bar{\mathbf{x}}, \ldots,\ \bar{\mathbf{x}})}\begin{pmatrix}(\mathbf{x}_1 - \bar{\mathbf{x}})^T \\ (\mathbf{x}_2 - \bar{\mathbf{x}})^T \\ \cdots \\ (\mathbf{x}_N - \bar{\mathbf{x}})^T\end{pmatrix} \\
&= \frac{1}{N}(\mathbf{X}^T - \bar{\mathbf{x}}\mathbf{1}_N^T)(\ldots)^T = \frac{1}{N}(\mathbf{X}^T - \frac{1}{N}\mathbf{X}^T\mathbf{1}_N\mathbf{1}_N^T)(\ldots)^T \\
&= \frac{1}{N}(\mathbf{X}^T(\mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T))(\ldots)^T \\
&= \frac{1}{N}\mathbf{X}^T H H^T \mathbf{X} = \frac{1}{N}\mathbf{X}^T H \mathbf{X}
\end{aligned} \tag{3}
$$

Geometric view of Principal Components





The first PC z_1:

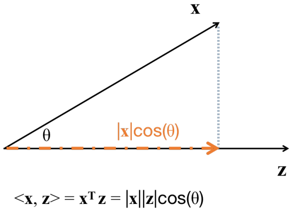
- 中心化: $\mathbf{x}_i - \bar{\mathbf{x}}$
- Projection: project the data to the vector z_1
- Maximize the variance of data in z_1 basis, with $|\mathbf{z}_1| = 1$

Project data to the vector z_1

Data: $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times p}$

- **Demean**: $\mathbf{x}_i - \bar{\mathbf{x}}$
- **Projection**: Project each data point to the vector $\mathbf{z}_1$
  $\Longrightarrow a_i = (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{z}_1$
- **Variance**:

$$\begin{aligned} \operatorname{var}[a_i] &= \mathbb{E}[(a_i - \bar{a})^2] \\ &= \frac{1}{N} \sum_{i=1}^{N} \left((\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{z}_1 \right)^2 \\ &= \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_1^T (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{z}_1 = \mathbf{z}_1^T S \mathbf{z}_1 \end{aligned}$$



Algorithm: PCA

- To find the optimal vector $\mathbf{z}_1^*$

Algorithm: PCA

- To find the optimal vector $\mathbf{z}_1^*$
- Objective: maximize the variance after projection $\mathbf{z}_1^T S \mathbf{z}_1$

Algorithm: PCA

- To find the optimal vector $\mathbf{z}_1^*$
- Objective: maximize the variance after projection $\mathbf{z}_1^T S \mathbf{z}_1$
- Constraints: $\mathbf{z}_1^T \mathbf{z}_1 = 1$

Algorithm: PCA

- To find the optimal vector $\mathbf{z}_1^*$
- Objective: maximize the variance after projection $\mathbf{z}_1^T S \mathbf{z}_1$
- Constraints: $\mathbf{z}_1^T \mathbf{z}_1 = 1$

  Let us denote λ as a Lagrange multiplier.
  The generalized Lagrangian function is:

$$\mathcal{L}(\mathbf{z}_1, \lambda) = -\mathbf{z}_1^T S \mathbf{z}_1 + \lambda(\mathbf{z}_1^T \mathbf{z}_1 - 1)$$

Algorithm: PCA

- To find the optimal vector $\mathbf{z}_1^*$
- Objective: maximize the variance after projection $\mathbf{z}_1^T S \mathbf{z}_1$
- Constraints: $\mathbf{z}_1^T \mathbf{z}_1 = 1$

  Let us denote λ as a Lagrange multiplier.
  The generalized Lagrangian function is:

$$\mathcal{L}(\mathbf{z}_1, \lambda) = -\mathbf{z}_1^T S \mathbf{z}_1 + \lambda(\mathbf{z}_1^T \mathbf{z}_1 - 1)$$

The derivative of $\mathcal{L}$ is:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{z}_1} = -2S\mathbf{z}_1 + 2\lambda \mathbf{z}_1 = S\mathbf{z}_1 - \lambda \mathbf{z}_1 = 0$$

$\mathbf{z}_1$ is an eigenvector of S, $\lambda = \lambda_1$ is the largest eigenvalue

Algorithm: PCA

Similarly, $\mathbf{z}_2$ is also an eigenvector of S whose eigenvalue $\lambda = \lambda_2$ is the second largest.
In general, we have the kth PCs:

$$\operatorname{var}[\mathbf{z}_k] = \mathbf{z}_k^T S \mathbf{z}_k = \lambda_k \tag{4}$$

The k^{th} largest eigenvalue of $S \longrightarrow k^{th}$ PC $\mathbf{z}_k$.

Dimensionality reduction:
$\longrightarrow$ From p-D to q-D with $q < p$
$\longrightarrow$ Find q largest eigenvalues of the covariance matrix S.

PCA for dimensionality reduction

- **1. Feature standardization**.
  To standardize the range of the raw data so that each feature contributes equally in the following analysis.
- **2. Obtain the covariance matrix** S.
  The covariance matrix $S \in \mathbb{R}^{p \times p}$ in Eq. (3),

$$S = \frac{1}{N}\mathbf{X}^T H \mathbf{X}, \quad H = \mathbf{I}_N - \frac{1}{N}\mathbf{1}_N^T \mathbf{1}_N$$

- **3. Eigendecomposition of the covariance matrix** S.
  Obtain all eigenvectors and eigenvalues
- **4. Sort the eigenvalues in a descending order**.
  The eigenvector with the highest eigenvalue is the first PC.
  The higher eigenvalues reflects the greater amounts of shared variance explained.

Main Steps for Principal Component Analysis

- **5. Select the number of principal components**.
  Select the top q eigenvectors (based on their eigenvalues) as the top q PCs.
- **6. Transformation matrix** G consists of the top q eigenvectors ($G \in \mathbb{R}^{p \times q}$).

$$G = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_q]$$

- **7. Project** p**-D data to** q**-D PC space**

$$\text{For each sample: } \mathbf{y}_i = G^T \mathbf{x}_i$$
$$\text{For entire dataset: } Y = XG$$

How many PCs are needed?

- Q: How many PCs are needed in order to not lose much information in the original data?
- Choose q based on how much variance to retain.
- Criterion (to find the smallest q):

$$\frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \geq 95\%$$

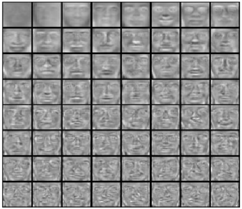
1. Recap
2. Dimensionality Reduction
3. PCA: methodology
4. PCA: application
5. Summary

A classical example: Eigenface

Dataset

Mean Face

Eigenfaces (PCs Visualization)



Face Reconstruction

PCA as feature extraction for classification

PCA as feature extraction for classification

- Project both training and testing data into the PCs space

$$Y = XG \quad \text{for the entire dataset}$$

- Run logistic regression or SVM on the q-D PCs space
- Problem 1: The classification accuracy may be sensitive to the choice of q
- Solution: $\longrightarrow$ Plot the accuracy curve with the varying q

PCA as feature extraction for classification

- Problem 2: PCA may not be suitable for classification.
- Q: Why?

  PCA is based on the sample covariance S which characterizes the scatter of the entire data set, **regardless of the class label**.

  The projection axes chosen by PCA might not provide good discrimination between classes.
- Solution: $\longrightarrow$ Supervised feature extraction

Algorithm: PCA

- To find the optimal vector $\mathbf{z}_1^*$
- Objective: maximize the variance after projection $\mathbf{z}_1^T S \mathbf{z}_1$
- Constraints: $\mathbf{z}_1^T \mathbf{z}_1 = 1$

Let us denote λ as a Lagrange multiplier.
The generalized Lagrangian function is:

$$\mathcal{L}(\mathbf{z}_1, \lambda) = -\mathbf{z}_1^T S \mathbf{z}_1 + \lambda(\mathbf{z}_1^T \mathbf{z}_1 - 1)$$

The derivative of $\mathcal{L}$ is:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{z}_1} = -2S\mathbf{z}_1 + 2\lambda \mathbf{z}_1 = S\mathbf{z}_1 - \lambda \mathbf{z}_1 = 0$$

$\mathbf{z}_1$ is an eigenvector of S, $\lambda = \lambda_1$ is the largest eigenvalue

Algorithm: PCA

Similarly, $\mathbf{z}_2$ is also an eigenvector of S whose eigenvalue $\lambda = \lambda_2$ is the second largest.

$$\operatorname{var}[\mathbf{z}_k] = \mathbf{z}_k^T S \mathbf{z}_k = \lambda_k \tag{5}$$

The k^{th} largest eigenvalue of $S \longrightarrow k^{th}$ PC $\mathbf{z}_k$.

Dimensionality reduction:
$\longrightarrow$ From p-D to q-D with $q < p$
$\longrightarrow$ Find p largest eigenvalues of the covariance matrix S.