主成分分析による Iris データセットの解析

2024 年度開講 数値解析学レポート

千葉工業大学 先進工学部 未来ロボティクス学科

22C1704 鷲尾 優作

2024 年 7 月 25 日

1　目的

本稿では、アヤメの花のデータセットである Iris データセットのうち、品種別データに対して主成分分析を行い、データの解析手法を学ぶことを目的とする．

2　理論

主成分分析（PCA）は、多次元データの次元削減を行うための統計手法である．PCA の目的は、データの分散を最大化する方向（主成分）を見つけ出し、元のデータを少数の主成分に変換することで、データの特徴を効果的に表現することである．PCA は以下のステップで実施される：

1. **データの標準化**：各変数を平均 0、分散 1 に標準化し、スケールの違いによる影響を排除する．
2. **共分散行列の計算**：標準化したデータから共分散行列を計算し、変数間の関係性を把握する．
3. **固有ベクトルと固有値の計算**：共分散行列の固有ベクトルと固有値を求め、主成分の方向とその重要性を評価する．
4. **主成分の選択**：固有値に基づき、主要な主成分を選択し、データをその主成分に投影して次元削減を行う．
5. **データの変換**：元のデータを選択した主成分に投影し、次元削減されたデータを得る．

2.1　問題設定

次の問題設定に基づいて、主成分分析を行う：

- **データセット**：Iris データセット（アヤメの 3 種類の品種に関する花弁とがく片の長さと幅のデータ）
- **目標**：データの次元を削減し、各品種が主成分空間でどのように分布しているかを視覚化する．
- **次元削減**：元の 4 次元の特徴量から、2 次元の主成分に次元削減を行う．

2.2　評価方法

主成分分析の結果を定量的に評価するために、以下の指標を用いる：

1. **分散説明率**：各主成分がデータの総分散に対してどの程度寄与しているかを示す指標であり、主成分の重要性を評価する．累積分散説明率も併せて確認し、次元削減の効果を評価する．
2. **主成分の可視化**：主成分分析によって得られた主成分空間におけるデータポイントの分布をプロットし、異なるクラス（品種）の分布の明確さを評価する．特に、品種ごとに色分けし、主成分空間での分離具合を確認する．
3. **再構成誤差**：データを主成分空間に投影した後、元の空間に再構成し、元データと再構成データとの差を計測する．これにより、次元削減による情報損失を評価する．

3　シミュレーション結果

シミュレーションの結果を以下に示す．図 1 に、主成分分析によって次元削減されたデータの状態遷移と観測の変化をグラフ化して示す．

図 1 は、主成分分析によって得られた 2 次元の主成分空間におけるデータポイントの分布を示している．元の 4 次元の特徴量が 2 次元の主成分に投影され、各アヤメの品種（Setosa、Versicolor、Virginica）が異なる色で表示されている．これにより、品種ごとのデータの分布状況と、主成分空間での分離度合いを視覚的に確認することができる．

具体的には、以下の点が図 1 から読み取れる．

- **品種間の分離**：主成分空間において、Setosa は他の 2 品種（Versicolor、Virginica）とは明確に分離している．主成分分析が品種間の違いを効果的に捉えていることが確認できる．Versicolor と Virginica のデータポイントは部分的に重なり合っているが、主成分空間内で一定のクラスタリングが観察できる．
- **データの分布**：主成分空間における各品種のデータポイントの密度や分布状況から、品種ごとのデータの集中度や分散状況が確認できる．これにより、次元削減による情報の保持とデータの可視化の有効性が評価できる．

分散説明率に関する具体的な結果は以下の通りである．

- **分散説明率（各主成分）**：主成分 1 が約 72.96%、主成分 2 が約 22.85% の分散を説明している．
- **累積分散説明率**：主成分 1 と主成分 2 を合わせると、データ全体の約 95.81% の分散を説明している．

再構成誤差は約 0.0418 であり、次元削減後のデータと再構成データとの差が比較的小さいことを示している．

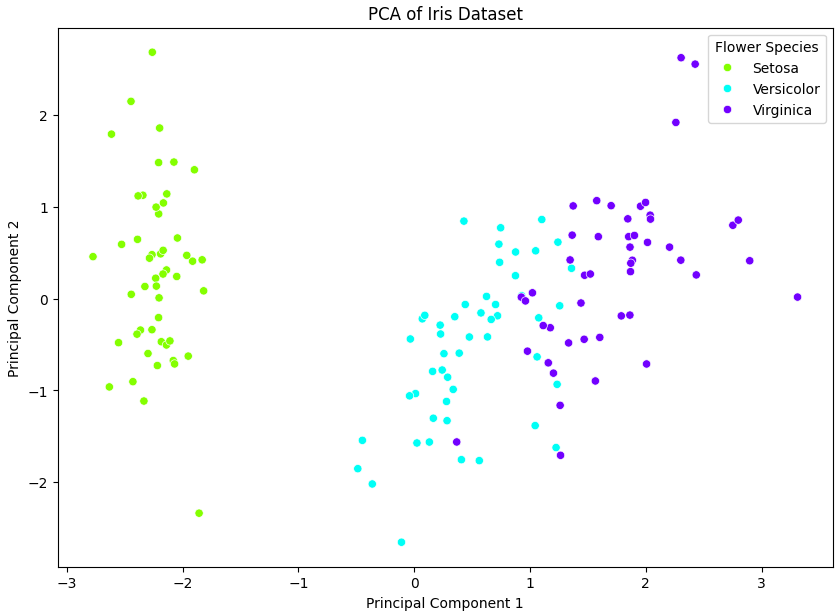

図 1: PCA による Iris データセットの主成分空間での可視化

4 考察

主成分分析の結果は、Iris データセットに対して次元削減が効果的であることを示している．特に、主成分分析を用いることで、元の 4 次元の特徴量から 2 次元の主成分空間に次元削減する際に、各品種のデータが明確に分離されることが確認できた．具体的には、図 1 に示されたように、Setosa は他の 2 品種（Versicolor、Virginica）とは明確に分離している．このことから、主成分分析が Setosa と他の品種との違いを効果的に捉えていることがわかる．主成分空間における Setosa のデータポイントは、他の品種と比較して分布が一貫しており、次元削減によって得られた主成分が品種の識別に有効であることが示されている．一方で、Versicolor と Virginica のデータポイントは部分的に重なり合っていることから、主成分分析が全ての品種の違いを完全に分離できるわけではないことがわかる．主成分分析はデータの分散を最大化する方向を見つけるが、すべての品種間の違いを完全に捉えることができるわけではない．したがって、さらなる解析や他の次元削減手法を併用することで、より精度の高い識別が可能となると考えられる．

分散説明率の結果は、主成分 1 と主成分 2 がデータの約 95.81% の分散を説明しており、主成分分析がデータの主要な情報を効果的に捉えていることを示している．再構成誤差が約 0.0418 であることから、次元削減後のデータと再構成データとの差が比較的小さいことが確認され、次元削減による情報の損失が限定的であることが示されている．

5 付録

5.1 シミュレーションプログラム

Listing 1: sim.png

```
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import mean_squared_error
import numpy as np

# データセットの読み込み Iris
iris = load_iris()
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
iris_df['species'] = iris.target

# データの表示
print(iris_df.head())

# データセットの読み込み Iris
iris = load_iris()
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
```

```
iris_df['species'] = iris.target

# 特徴量の標準化
scaler = StandardScaler()
scaled_data = scaler.fit_transform(iris_df.iloc[:, :-1])

# 主成分分析の実施
pca = PCA(n_components=2)
pca_result = pca.fit_transform(scaled_data)

# 主成分得点をデータフレームに追加
pca_df = pd.DataFrame(pca_result, columns=[
                      'Principal Component 1', 'Principal Component 2'])
pca_df['species'] = iris_df['species']

# 品種のラベルを文字列に変換
species_map = {0: 'Setosa', 1: 'Versicolor', 2: 'Virginica'}
pca_df['species'] = pca_df['species'].map(species_map)

# 主成分得点のプロット
plt.figure(figsize=(10, 7))
sns.scatterplot(
    x='Principal Component 1', y='Principal Component 2',
    hue='species',
    palette=sns.color_palette("hsv", 3),
    data=pca_df,
    legend='full'
)
plt.title('PCA of Iris Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title='Flower Species')
plt.show()

# 分散説明率の計算
explained_variance_ratio = pca.explained_variance_ratio_
cumulative_variance_ratio = np.cumsum(explained_variance_ratio)

# 再構成誤差の計算
pca_back = pca.inverse_transform(pca_result)
reconstruction_error = mean_squared_error(scaled_data, pca_back)

# 評価結果の表示
print(f'分散説明率（各主成分）: {explained_variance_ratio}')
print(f'累積分散説明率: {cumulative_variance_ratio}')
print(f'再構成誤差: {reconstruction_error}')
```

参考文献

[1] 主成分分析 - wikipedia. https://ja.wikipedia.org/wiki/%E4%B8%BB%E6%88%90%E5%88%86%E5%88%86%E6%9E%90. (Accessed on 07/25/2024).

[2] R. A. Fisher. Iris. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C56C76.