深 圳 大 学 实 验 报 告

课程编号：**2801000049**

课程名称：机器学习

实验项目名称：**Machine Learning Task 4**

学院：电子与信息工程学院

专业：电子信息工程

指导教师：麦晓春

报告人：廖祖颐 学号：**2022110131** 班级：文华班

实验时间：**2024.6.13**

实验报告提交时间：**2024.6.17**

教务部制

1. Experimental Purposes and Requirements

(1). Load the dataset seeds.csv.

(2). Extract the grain variety and the feature data, and transform the feature data into a numpy array.

(3). Run the non-negative matrix decomposition on the data multiple times (rank=3, repeat_times=20).

(4). Calculate the Silhouette coefficients for each clustering result, and plot a line graph, labeling the best scores.

(5). Transform the clustering results corresponding to the best score into a Dataframe, and make a crosstab of the varieties and the best clustering results pd.crosstab(df1, df2)

2. Experiment Contents and Process

1) For Library-based NMF with Sklearn:

This experiment involves applying the Non-negative Matrix Factorization (NMF) technique using the sklearn library to a dataset containing grain varieties. The primary goal is to reduce the dimensionality of the dataset, identify latent features, and cluster data points into groups that correspond to these features. Here comes the mainly processing

- Grain varieties are extracted and set aside as labels, while the remaining columns are treated as features.
- Features are scaled using MinMaxScaler to ensure all values are within the range [0, 1], which is suitable for NMF operations.
- The sklearn NMF implementation is employed with the initialization method set to 'nndsvd'
- The NMF is run multiple times (20 iterations) with different random states to ensure robustness and consistency in the results.
- A label is assigned to each data point based on the dominant feature in the transformed data
- Silhouette scores are calculated for each iteration
- The best iteration is highlighted using a vertical dashed line on the plot
- Finally, the best clustering results are transformed into a DataFrame to create a crosstab comparing the original grain varieties with the predicted clusters, providing insights into the clustering efficacy.

```python
import numpy as np
import pandas as pd
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import NMF

# Load the dataset
data = pd.read_csv('seeds.csv')

# Extract grain variety and feature data
varieties = data['grain_variety']
features = data.drop('grain_variety', axis=1)

# Use MinMaxScaler to scale the data to [0, 1]
scaler = MinMaxScaler()
feature_array = scaler.fit_transform(features)

# Proceed with NMF
rank = 3
repeat_times = 20
silhouette_scores = []
```

```python
for i in range(repeat_times):
    model = NMF(n_components=rank, init='nndsvd', random_state=i, max_iter=500)
    W = model.fit_transform(feature_array)
    H = model.components_
    labels = np.argmax(W, axis=1)

    # Compute silhouette scores
    score = silhouette_score(feature_array, labels)
    silhouette_scores.append(score)

# Plotting silhouette scores
plt.figure(figsize=(10, 5))
plt.plot(range(len(silhouette_scores)), silhouette_scores, marker='o')
plt.title('Silhouette Scores for NMF Clustering')
plt.xlabel('Valid Iteration')
plt.ylabel('Silhouette Score')
best_score_index = np.argmax(silhouette_scores)
plt.axvline(x=best_score_index, color='r', linestyle='--', label=f'Best Score at Iteration {best_score_index}')
plt.legend()
plt.grid(True)
plt.show()
```

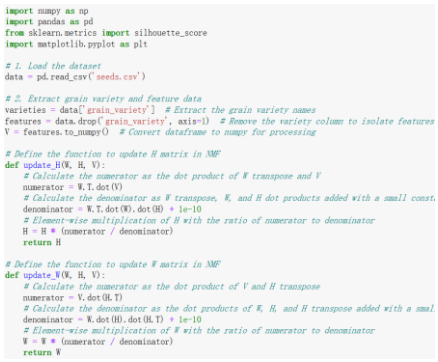
Fig 1.1 Code of Library-based NMF

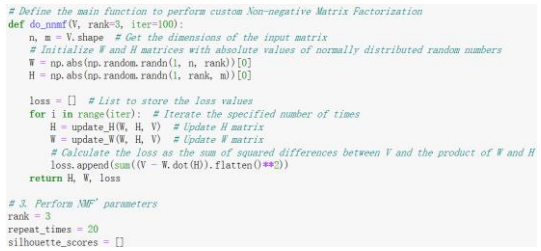
2) Manual NMF

This experiment involves a custom implementation of the Non-negative Matrix

Factorization (NMF) on the same 'seeds.csv' dataset. This manual approach allows for detailed manipulation and understanding of the NMF process by explicitly defining the update rules for matrices W and H.

- Similar to the library-based approach, the dataset is loaded and preprocessed. Varieties are extracted as labels, and the remaining features are converted to a numpy array for matrix operations.
- Custom functions update_H and update_W are defined to manually perform the matrix factorization. These functions are designed to update the matrices using the multiplication update rules, which are fundamental to NMF.
- The NMF is executed with initialized random values (absolutely valued to ensure non-negativity) for matrices W and H and is iterated over 100 times for convergence within each of the 20 different trials.
- Data points are clustered based on the highest contribution factor in matrix W
- Silhouette scores are calculated for each iteration
- The best iteration is highlighted using a vertical dashed line on the plot
- Finally, the best clustering results are transformed into a DataFrame to create a crosstab comparing the original grain varieties with the predicted clusters, providing insights into the clustering efficacy.





```
for _ in range(repeat_times):
    H, W, loss = do_nnmf(V, rank=rank, iter=100)  # Perform NMF
    labels = np.argmax(W, axis=1)  # Assign labels based on the highest contribution in W
    score = silhouette_score(V, labels)  # Calculate the silhouette score
    silhouette_scores.append(score)  # Store the scores

# 4. Plot the silhouette scores to evaluate clustering quality
plt.figure(figsize=(10, 5))
plt.plot(range(repeat_times), silhouette_scores, marker='o')
plt.title('Silhouette Scores for Custom NMF Clustering')
plt.xlabel('Iteration')
plt.ylabel('Silhouette Score')
best_score_index = np.argmax(silhouette_scores)  # Find the iteration with the best silhouette score
plt.axvline(x=best_score_index, color='r', linestyle='--', label=f'Best Score at Iteration {best_score_index}')
plt.legend()
plt.grid(True)
plt.show()

# 5. Transform the clustering results corresponding to the best score into a DataFrame and create a crosstab
H, W, loss = do_nnmf(V, rank=rank, iter=100)  # Perform NMF with the best settings
best_labels = np.argmax(W, axis=1)  # Assign labels based on the highest contribution in W
df1 = pd.DataFrame({'Variety': varieties})  # Dataframe of varieties
df2 = pd.DataFrame({'Cluster': best_labels})  # Dataframe of clusters
cross_tab = pd.crosstab(df1['Variety'], df2['Cluster'])  # Create a crosstab of varieties and clusters
print(cross_tab)  # Print the crosstab
```

Fig 1.2 Code of Manual NMF

3 Experimental Results and Analysis

1) For Library-based NMF with Sklearn:

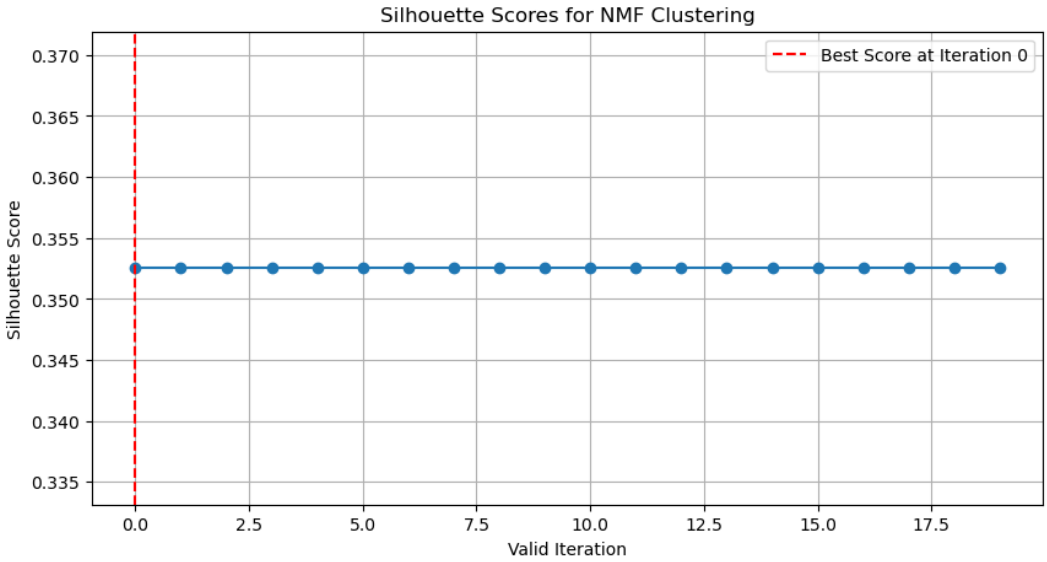


Fig 1.3 Result of Library-based NMF

Table 1 Clustering Results 1

Variety	Cluster 0	Cluster 1	Cluster 2
Canadian wheat	1	62	7
Kama wheat	17	2	51
Rosa wheat	67	0	3

Silhouette Score Analysis:

The silhouette scores for the library-based NMF approach are consistent and relatively high across all iterations (ranging approximately from 0.34 to 0.37), as seen in the fig 1.3. This indicates that the clustering has relatively good structure and separation between clusters.

The best silhouette score occurs at the very first iteration, which might suggest that the initialization and optimization by the NMF model are highly effective in finding a good solution early on.

Crosstab Results:

The library-based NMF crosstab indicates a strong clustering of Canadian and Kama wheat, with the majority of Canadian wheat being correctly clustered into one group (62 out of 70) and the majority of Kama wheat into another (51 out of 70).

Rosa wheat predominantly falls into a single cluster (67 out of 70), showing a high degree of homogeneity within this variety.

2) Manual NMF

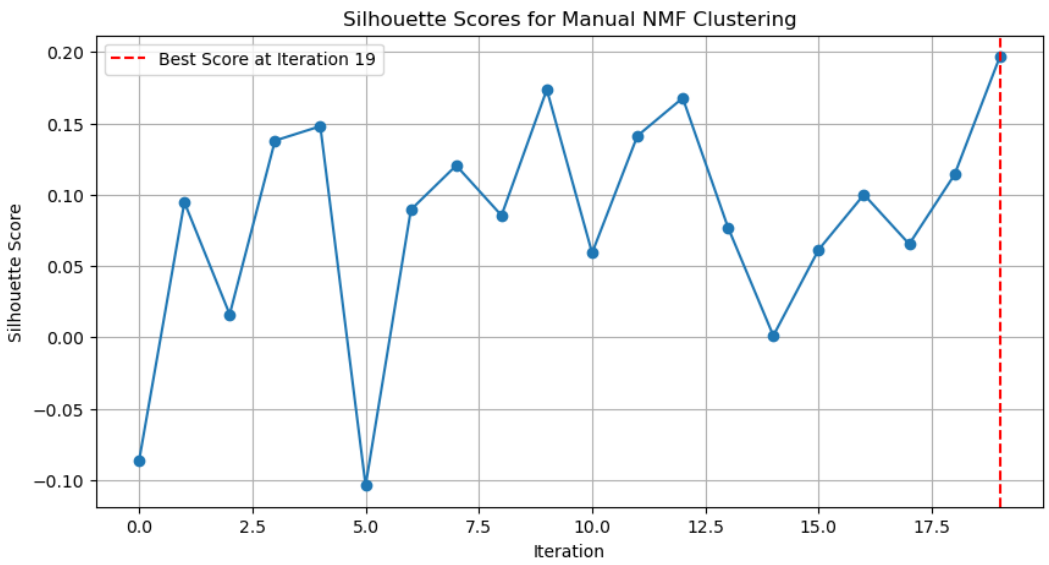


Fig 1.4 Result of Manual NMF

Table 2 Clustering Results 2

Variety	Cluster 0	Cluster 1	Cluster 2
Canadian wheat	10	59	1
Kama wheat	32	19	19
Rosa wheat	29	15	26

Silhouette Score Analysis:

In contrast, the silhouette scores for the manual NMF vary significantly across iterations, including negative scores, which imply some overlapping clusters and poor separation. Scores fluctuate from below zero up to approximately 0.20.

The highest score appears at the last iteration (19th), indicating a less stable or consistent clustering outcome compared to the manual approach. This variation could be due to the random initialization and less optimized update mechanisms inherent in the custom implementation.

Crosstab Results:

The manual NMF crosstab results are less definitive. Canadian wheat is almost evenly split between two clusters, Kama wheat is dispersed across all three clusters, and Rosa wheat, while primarily in one cluster, is also somewhat dispersed.

This distribution suggests less effective clustering, where the algorithm struggles to distinctly separate the varieties into unique clusters.

4. Discussion and Conclusions

Discussion:

Theoretical Insights from the Paper

Lee and Seung's paper discusses two types of cost functions for NMF:

1. **Euclidean distance** - which measures the direct distances between the original data points and their approximations.
2. **Generalized Kullback-Leibler divergence** - which measures the difference in information gain between the original and reconstructed data, appropriate for probability distributions.

These cost functions reflect different underlying assumptions about the data and affect the convergence behavior of the NMF algorithms. Specifically, the paper highlights the use of multiplicative update rules, which have been shown to improve the quality of approximations monotonically under certain conditions.

Specifically, the choice of cost function and update rules critically impacts the performance and application outcomes of the algorithms.

For other questions, this **manual way of NMF** uses the initial matrices(W and H)with random numbers drawn from the standard normal distribution, constrained to non-negative values by taking their absolute values or clipping negative values to zero. Its pros maybe simplicity and quick setup. But it can lead to slower convergence or convergence to suboptimal local minima, each run can yield different results due to the randomness, affecting the reproducibility, the quality of the starting point heavily depends on the random numbers generated. **The Library-based NMF** uses more sophisticated initialization 'nndsvd' This method is based on obtaining two smaller matrices whose product approximates the non-negative matrix. It uses the SVD (Singular Value Decomposition) of the data and then approximates all negative elements with zeros. Its pros maybe 'Better start' and 'Reduces randomness'. But it results in high complexity and computational cost, making it slow to process

Conclusions:

The **library-based NMF** shows a **high level of effectiveness in clustering** wheat varieties, indicated by higher and more stable silhouette scores and clear grouping in the crosstab results. This outcome underscores the advantages of using robust, well-optimized library methods for tasks like NMF, especially with good initialization methods such as 'nndsvd'.

The **manual NMF**, while offering valuable insights into the algorithmic underpinnings and flexibility for customization, exhibits **greater variability in performance and less effective clustering**. This variability is likely due to the manual handling of matrix updates and random initializations, which may not consistently approach optimal solutions.

指导教师批阅意见： 成绩评定： 指导教师签字： 年　　月　　日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。