HW

MNIST DATASET

H24101222_陳凱騫

2024-11-11

Table of contents

MDS

1. 載入必要的套件

首先，我們先載入一些必要的 R 套件：

```
# 載入套件
library(readr)   # 用於讀取CSV
library(dplyr)   # 用於資料處理
library(stats)   # MDS 方法來自 base R 的 stats package
library(ggplot2) # 用於視覺化
library(showtext) # Enable showtext to use system fonts
showtext_auto()
```

2. 讀取資料

接下來，我們讀取 MNIST 訓練數據，並查看其結構：

```
# 讀取資料 (將路徑替換為上傳的檔案路徑)
data <- read_csv("MNIST_train.csv")

# 查看前幾行以瞭解資料結構
head(data)
```

```
# A tibble: 6 x 785
  label pixel0 pixel1 pixel2 pixel3 pixel4 pixel5 pixel6 pixel7 pixel8 pixel9
  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1     1      0      0      0      0      0      0      0      0      0      0
2     0      0      0      0      0      0      0      0      0      0      0
```

```
3   1    0    0    0    0    0    0    0    0    0    0
4   4    0    0    0    0    0    0    0    0    0    0
5   0    0    0    0    0    0    0    0    0    0    0
6   0    0    0    0    0    0    0    0    0    0    0
# i 774 more variables: pixel10 <dbl>, pixel11 <dbl>, pixel12 <dbl>,
#   pixel13 <dbl>, pixel14 <dbl>, pixel15 <dbl>, pixel16 <dbl>, pixel17 <dbl>,
#   pixel18 <dbl>, pixel19 <dbl>, pixel20 <dbl>, pixel21 <dbl>, pixel22 <dbl>,
#   pixel23 <dbl>, pixel24 <dbl>, pixel25 <dbl>, pixel26 <dbl>, pixel27 <dbl>,
#   pixel28 <dbl>, pixel29 <dbl>, pixel30 <dbl>, pixel31 <dbl>, pixel32 <dbl>,
#   pixel33 <dbl>, pixel34 <dbl>, pixel35 <dbl>, pixel36 <dbl>, pixel37 <dbl>,
#   pixel38 <dbl>, pixel39 <dbl>, pixel40 <dbl>, pixel41 <dbl>, ...
```

3. 隨機抽樣並移除標籤

由於數據集較大，我們從中隨機抽取 1500 筆資料來進行分析，並移除標籤欄位：

```r
# 設定隨機種子以確保結果可重現
set.seed(123)

# 抽取隨機樣本
sampled_data <- data %>% sample_n(1500)

# 移除標籤欄位，並將其轉換為矩陣
data_matrix <- as.matrix(sampled_data[,-1])
```

4. 使用 MDS 進行降維

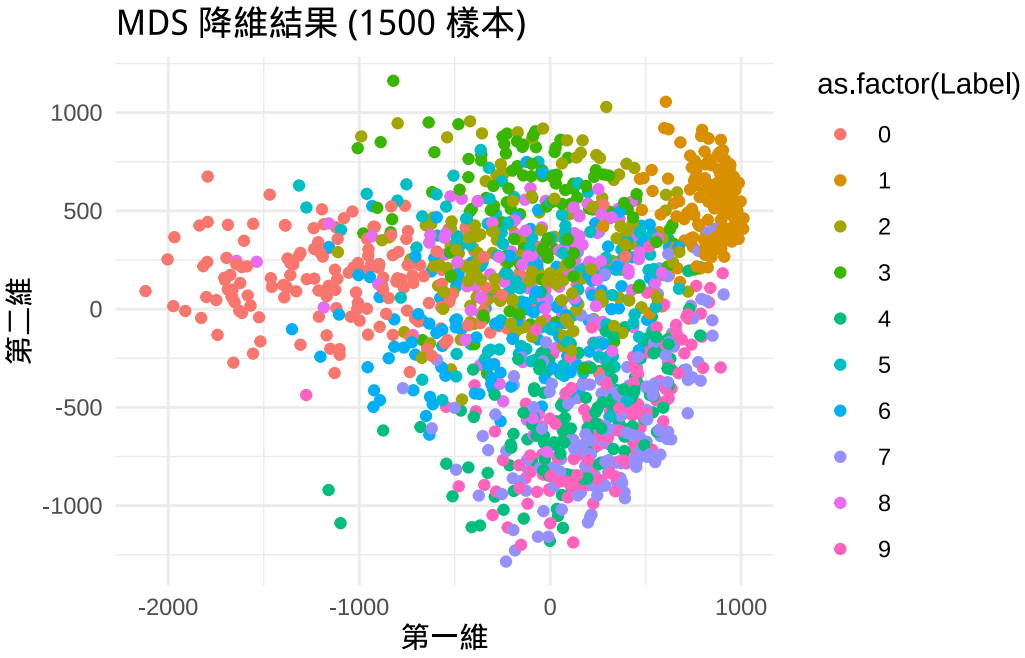
利用歐氏距離計算距離矩陣，然後使用 MDS 將數據降到 2 維空間中：

```r
# 計算距離矩陣，這裡使用歐氏距離
distance_matrix <- dist(data_matrix)

# 使用 MDS 進行降維 (設置降到2維)
mds_result <- cmdscale(distance_matrix, k = 2)

# 將結果轉換為 data frame 以便繪圖
mds_df <- as.data.frame(mds_result)
colnames(mds_df) <- c("Dim1", "Dim2")
```

5. 繪製 MDS 降維結果

我們將 MDS 結果繪製成 2 維散點圖，並根據數據的原始標籤進行上色：

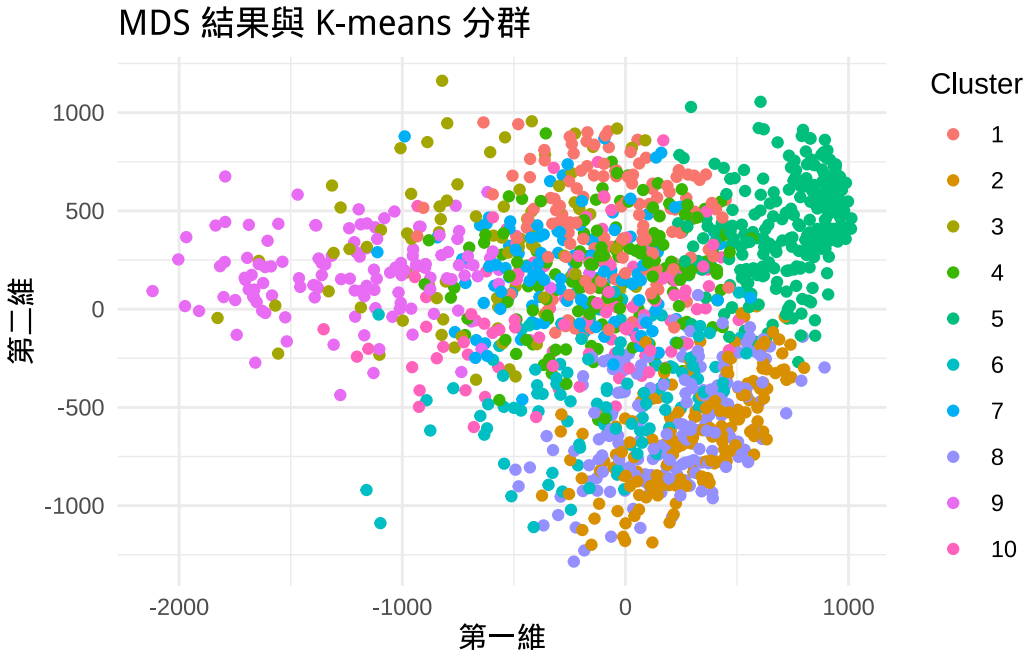
```r
# 將標籤加入到資料框
mds_df$Label <- sampled_data$label

# 繪製 MDS 結果
ggplot(mds_df, aes(x = Dim1, y = Dim2, color = as.factor(Label))) +
  geom_point() +
  labs(title = "MDS 降維結果 (1500 樣本)", x = "第一維", y = "第二維") +
  theme_minimal()
```



6. 使用 K-means 進行分群

我們使用 K-means 演算法將資料分成 10 群，並將分群結果繪製到 MDS 降維空間中：

```
# 執行 K-means 分群 (假設分成 10 群)
set.seed(123)
kmeans_result <- kmeans(data_matrix, centers = 10)

# 加入群組標籤到資料框
mds_df$Cluster <- as.factor(kmeans_result$cluster)

# 繪製 MDS 結果並根據群組上色
ggplot(mds_df, aes(x = Dim1, y = Dim2, color = Cluster)) +
  geom_point() +
  labs(title = "MDS 結果與 K-means 分群", x = "第一維", y = "第二維") +
  theme_minimal()
```



7. 結論

通過這次分析，我們成功將 MNIST 資料集中的部分樣本降維到 2 維空間，並應用了 K-means 演算法進行分群。而結果顯示，K-means 在降維空間中成功將數據分成了10個群組且分群效果呈現與訓練資料之label差不多。