1103 - Clustering - Design

Input Dataset

Mall Customer Dataset is the dataset used in original CAMIS for analyzing R result of running K Means clustering. It consists of 200 customer subjects and four column features (gender, age, annual income, and spending score). Accessible from Kaggle (https://www.kaggle.com/datasets/shwetabh123/mall-customers).

Numeric Agreement Criteria

Within 0.0001 to be considered equivalent (i.e. difference <= 0.0001).

Expectations for Supported Statistics

SAS (PROC FASTCLUS)

- Cluster Summary: frequency, RMS Std Deviation, Maximum Distance from Seed to Observation, Nearest Cluster, Distance Between Cluster Centroids
- Table of statistics for each variable: Total STD, Within STD, R-Square, etc.
- Pseudo F Statistic, Approximate Expected Over-All R-Squared, Cubic Clustering Criterion
- Cluster Means for each variable, Cluster Standard Deviations for each variable

R (stats.kmeans)

- All components:
  `cluster`: Cluster assignment for each observation
  `centers`: Cluster centers (means for each cluster)
  `totss`: Total sum of squares (overall variance)
  `size`: Number of points in each cluster
  `withins`: Vector of within-cluster SS (per cluster)
  `tot.withinss`: Total within-cluster SS
  `betweenss`: Between-cluster SS (explained variance)
  `size`: Number of points in each cluster
  `iter`: Number of iterations until convergence
- Other: Within cluster sum of squares by cluster

Python (sklearn)

- Cluster assignment for each customer subject
- Within-cluster sum of squares
- Cluster Sizes
- Cluster Centers
- Total within-cluster sum of squares
- Number of clusters
- Algorithm used for initialization
- Number of iterations

Function Usage & Critical Arguments

SAS (PROC FASTCLUS)

- Data needs to be standardized first using `proc stdize`

- Arguments include `maxlusters`, `replace`, `var`

```
proc fastclus data=analysis_data_scaled
    maxclusters=5
    out=clustered_data
    outstat=cluster_stats
    outseed=cluster_seeds
    replace = FULL;
    var 'Annual Income (k$)'n 'Spending Score (1-100)'n;
run;
```

R (stats.kmeans)

- Standardization to have a mean of 0 and sd of 1:

```
df1 <- df %>% mutate(across(where(is.numeric), scale))
```

- Arguments: data, number of clusters, number of initial random centroids to try (later finds the best)

```
#make this example reproducible
set.seed(1)
#perform k-means clustering with k = 5 clusters
fit <- kmeans(df1, 5, nstart=25)
#view results
fit
```

Python (sklearn)

- Standardization

```
df1[numeric_columns] = df1[numeric_columns].apply(lambda x: (x - x.mean())
df1_selected = df1.iloc[:, 3:5]
```

- Define KMeans class & call function: `n_clusters` (number of clusters), `n_init` (initialization algorithm), `random_state`

```
kmeans = KMeans(n_clusters=5, n_init=25, random_state=1)
fit = kmeans.fit(df1_selected)
```

- Obtain output statistics

```
fit.n_clusters
fit.n_iter_
fit.cluster_centers_
fit.inertia_
kmeans.predict(df1_selected)
kmeans.get_params()
```

Known Incompatibilities

- Python sklearn `StandardScaler()` and R's default standardization produce different result. Usually R's standardization approach is preferred.
- Python's cluster id is 0-indexed while R's is 1-index.
- SAS default `OUTSTAT` doesn't support the display of cluster assignment for each subject. Individual clustering assignment is not stored in the ADaM BDS dataset.
- SAS supports metrics used for estimating the numebr of clusters –– Cubic Clustering Criterion, Pseudo F Statistic, Overall R-Squared.
- SAS supports RMS Standard Deviation for each cluster.

Comparison Protocol & Metrics

Metrics

To test equality of results from the three packages (SAS, R, Python), we expect

- Since the number of cluster (k) is prespecified, we expect the cluster size for each cluster to agree across packages.
- We expect the cluster center (with two variables, annual income and spending score) for each cluster to agree (within 0.0001) across packages.
- If available, we expect the cluster assignment of each subject (observation) to match.
- If available, we expect the WCSS (within-cluster sum of square) –– for each cluster and in total –– to match.

Notes

- The method setup such as number of clusters (k) and initialization algorithm can be influential in result comparison. For reference, we used a prespecified number of clusters (k) of 5 and the initialization methods of 'Lloyd' for Sklearn, 'Hartigan and Wong' for R (with `n_start = 25`), and SAS FASTCLUS uses its unique initialization method of selecting a set of observations as initial cluster seeds and iteratively refining them.

- **We identified an existing package difference in our analysis of the Mall Customer Dataset**. While R kmeans & Python Sklearn produced the same result (matching cluster size, cluster assignments, cluster center, & WCSS), SAS FASTCLUS yields a slightly varied result: while R & Python obtained 5 clusters of sizes 35, 39, 22, 23, 81, SAS obtained 5 clusters of sizes 21, 38, 21, 39, 81 (disregard the ordering here). The difference can also be viewed in the cluster center statistics (disregard cluster ordering):
  - SAS:

USUBJID	PARAMCD	PARAM	AVAL
CLUST1	CENINCOME	Cluster Center: Annual Income	1.0523622
CLUST1	CENSPEND	Cluster Center: Spending Score	-1.28122394
CLUST2	CENINCOME	Cluster Center: Annual Income	0.989101
CLUST2	CENSPEND	Cluster Center: Spending Score	1.23640011
CLUST3	CENINCOME	Cluster Center: Annual Income	-1.3262173
CLUST3	CENSPEND	Cluster Center: Spending Score	1.12934389
CLUST4	CENINCOME	Cluster Center: Annual Income	-1.3042458
CLUST4	CENSPEND	Cluster Center: Spending Score	-1.13411939
CLUST5	CENINCOME	Cluster Center: Annual Income	-0.2004097
CLUST5	CENSPEND	Cluster Center: Spending Score	-0.02638995

  - Python/R:

USUBJID	PARAMCD	PARAM	AVAL
CLUST1	MEANINC	Cluster 1 Mean Annual Income	-1.350281302
CLUST1	MEANSCOR	Cluster 1 Mean Spending Score	1.155830697
CLUST2	MEANINC	Cluster 2 Mean Annual Income	1.006673546
CLUST2	MEANSCOR	Cluster 2 Mean Spending Score	-1.222467697
CLUST3	MEANINC	Cluster 3 Mean Annual Income	-1.34846826
CLUST3	MEANSCOR	Cluster 3 Mean Spending Score	-1.187916616
CLUST4	MEANINC	Cluster 4 Mean Annual Income	0.989100984
CLUST4	MEANSCOR	Cluster 4 Mean Spending Score	1.236400114
CLUST5	MEANINC	Cluster 5 Mean Annual Income	-0.248824596
CLUST5	MEANSCOR	Cluster 5 Mean Spending Score	-0.013481823

Reference

- Python Sklearn: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans
- SAS: https://documentation.sas.com/doc/en/statug/15.2/statug_fastclus_toc.htm
- R KMeans: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/kmeans.html
- Dataset: https://www.kaggle.com/datasets/shwetabh123/mall-customers
- CAMIS clustering result using R: https://psiaims.github.io/CAMIS/Clustering_Knowhow.html
- FASTCLUS initialization: https://www.math.wpi.edu/saspdf/stat/chap27.pdf