shrinkageClust: an R package for Shrinkage Clustering

 by Chenyue W. Hu, Hanyang Li and Amina A. Qutub

**Abstract**

…

**Introduction**

…

**Shrinkage Clustering**

…

**The shrinkageClust package**

The **shrinkageClust** package was developed with the aim of enabling and promoting the usage of the Shrinkage Clustering algorithm in the R community. The package includes a main function SuperCluster(), a function that converts a data set into a similarity matrix which can be used as the input of SuperCluster(), a plot method to visualize the cluster assignment, and a function that quantitatively evaluates the accuracy of a clustering result. To perform the Shrinkage Clustering algorithm using the shrinkageClust package, users can first use function simiMatrix() to convert a feature-based data matrix to a similarity matrix, and then run the main function SuperCluster() to obtain the clustering result. If the data are already in the form of similarity matrices, users can directly run SuperCluster() with the data. After clustering the data, users can run scplot() to plot the cluster assignment result.

**The SuperCluster() function**

The SuperCluster function takes a data matrix as input, performs Shrinkage Clustering, and returns three objects, including a predicted label for all examples, a cluster assignment matrix, and an indicator of convergence. Because in some cases, users have some rough ideas about what the correct clustering result should look like, the function provides uses with some controls over the clustering process. To be specific, the users can specify the minimum number of observations assigned to each cluster and the maximum number of clusters. These constraints parameters allow users to regulate the number and size of clusters. The function is used as the follow:

SuperCluster(s,w=NULL,k=NULL,iter=1000,random=1,Path=F)

The function has six input arguments:

* **s** is a similarity matrix. For a dataset containing n observations, the size of s is n x n. The element in row i and column j of s indicates the similarity score between observation i and observation j. The higher the score is, the more similar the two observations are. This matrix can be generated by the simiMatrix() from a feature-based data matrix.
* **w** is the minimum number of members that are required to be assigned to a cluster. It is a constrain parameter that regulates the cluster assignment of Shrinkage Clustering algorithm. By default, this constraint is not applied to the algorithm, but users can specify its value as a way to avoid clusters with too few observations in the result when such small clusters are not desirable.
* **k** is another constraint parameter which represents the maximum number of clusters allowed. It is also not used by default.
* **iter** specifies the number of iterations Shrinkage Clustering algorithm will be run to update the cluster assignments. More iterations will take long time to run but will be more likely to converge to a local optimum.
* **random** is an integer value that specifies the number of random initializations, which is essentially how many times the function will re-run the algorithm with different initial cluster assignments. The Shrinkage Clustering algorithm is only guaranteed to find local optimum but not global optimum, similar to other clustering algorithms like k-means. Therefore, it is often desirable to run the algorithm with different random initializations (in this case the initial cluster assignment matrix) in the hope of approaching the global optimum. As mentioned in the paper of Hu et al. (2018), the performance of Shrinkage Clustering is relatively stable in terms of finding a good clustering solution. Despite this stability in performance, we consider it’s better to allow users to determine the number of random initializations.
* **Path** is a Boolean value. If it's true, then the function will track and record the number of remaining clusters in each iteration. It allows users to see how the number of clusters changes in each iteration. Normally, in the Shrinkage Clustering, the number of clusters decreases continuously until the cluster assignment converges to a local optimum.

The function returns three outputs:

* **c** is a list representing the predicted cluster membership of every observation. It has a size of n x 1, where n is the number of observations in the dataset. The nth element in the list indicates which cluster the nth observation belongs to. If the input argument random is set greater than 1, then c will contain multiple columns, where each column is a cluster membership list corresponding to one random initialization.
* **a** is a cluster assignment matrix of size n x k, where n is the number of observations and k is the number of predicted clusters. The elements in the matrix are all 1 or 0. aij is 1 if observation i is assigned to cluster j; otherwise aij would be 0. If the input argument random is set greater than 1, multiple assignment matrices will be returned, with each corresponding to a random initialization.
* **score** is an indicator of convergence.

**The simiMtrix function**

The simiMatrix function is a helper function that convert a feature-based data matrix to a similarity matrix so that the data can be used as input for the SuperCluster() function. If data in the form of similarity matrices are already available, users do not need to use this function. For a feature-based data (size n x k) which has n observations and k features, simiMatrix() calculate the corresponding similarity matrix (size n x n). The function first computes a matrix of Euclidean distance between each pair of observations. Next, it calculates a parameter beta and then the similarity matrix. The formula for computing beta and the similarity scores are described in detail in the original Shrinkage Clustering paper (Hu et al. 2018).

The function has one input argument, which is a feature-based data matrix with n examples and k features. Depending on the data sets, it might be favorable to preprocess the original data set before inputting it into the simiMatrix function. For example, the means and standard deviations of all features might need to be scaled so that their contributions to the overall similarity score are comparable.

The function returns a similarity matrix S (size n x n) of the input data set. sij represents the similarity score between observation i and j. The greater the value is, the more similar the two observations are.

**The scplot function**

The scplot function is a helper function that shows the predicated cluster assignment on a principal component analysis (PCA) plot. It takes the original data matrix and a clustering result as input and generates a pdf file of the plot. The function uses library ggbiplot to generate the clustering assignment PCA plot. To install ggbiplot, users can use the following method.

library(devtools)

install\_github('ggbiplot','vqv')

library(ggbiplot)

**The evaluation function**

The evaluation function was created to assess the accuracy of the Shrinkage Clustering algorithm on some test data sets. It quantitatively assesses the accuracy of a predicted clustering result by comparing it to the ground-truth cluster membership and calculating three metrics, which are normalized mutual information (NMI), Rand index, and F1 score (Manning et al., 2008). To calculate NMI, the R package NMI is required to be installed first.

The input of the function is a predicted cluster membership list and a ground-truth cluster membership list. The output contains the three metrics.

**Examples**

In this section, we use the Breast Cancer Wisconsin Diagnostic (BCWD) dataset as an example to demonstrate the use of the **shrinkageClust** package. The BCWD dataset is a preprocessed dataset originally from a study of breast cancer diagnostics (Street et al., 1993 and Mangasarian et al., 1995). It is used as a case study in the original paper of the Shrinkage Clustering algorithm (Hu et al., 2018).  The BCWD dataset contains 569 breast cancer samples (357 benign and 212 malignant) with 30 characteristic features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. The dataset is available on the UCI machine learning repository (Bache and Lichman, 2013) and is one of the most popularly tested dataset for clustering and classification.

We can run the functions in the shrinkageClust package with this dataset. The R code is shown below. After loading the data and label, we use function simiMatrix to compute the similarity matrix from the original feature-based data matrix. Then, we can check the distribution of all the similarity scores in the similarity matrix. In general, the Shrinkage Clustering algorithm tends to perform well if the similarity scores are distributed symmetrically with two peaks at the two ends and a mean around 0.5. If the shape of the distribution is very different from this shape, additional preprocessing of the data set may need to be performed. From Figure 1, we can see that the distribution of similarity scores of the BCWD dataset is close to the ideal shape. Thus, the Shrinkage Clustering method is appropriate. Then, we call the SuperCluster function to run the Shrinkage Clustering algorithm on this dataset, which results in a predicted clustering assignment. Using this predicted assignment and the ground-truth label, we can calculate three accuracy metrics of clustering (Table 1) and visualize the clustering assignment on a PCA plot (Figure 2). The relatively high values of NMI, Rand index, and F1 score show that the clustering membership of BCWD dataset predicted by the Shrinkage Clustering algorithm is relatively accurate.

# load BCWD dataset (569 samples, 30 dimensions, 2 clusters)

data("bcwd\_data")

label = data$label

data = data[, c(1:30)]

# compute similarity matrix

S = simiMatrix(data)

# view distribution of similarity scores in S with different beta values

hist(S)

# run shrinkage clustering

set.seed(10)

clust = SuperCluster(s=S,w=100,k=20,iter=500,random=1)

clust\_membership = clust$c[,1]

# quantitatively assess the accuracy of the clustering result

eval\_scores = evaluation(clust\_membership, label)

eval\_scores$NMI # Normalized mutual information

eval\_scores$RI # Rand index

eval\_scores$F1 #F1 score

# generate a PCA plot to visualize the clustering result

scplot(data, clust\_membership)



**Figure 1.** Distribution of similarity scores of the BCWD dataset.

|  |  |
| --- | --- |
|  | Shrinkage Clustering – BCWD Dataset (2 clusters) |
| NMI | 0.50 |
| Rand Index | 0.77 |
| F1 Score | 0.80 |
| Number of Clusters | 2 |

**Table 1.** Accuracy of the clustering result of BCWD dataset.



**Figure 2.** Visualization of the predicted cluster membership of observations on a PCA plot.

**Summary**

This paper introduces the R package shrinkageClust, which provides fast and accurate clustering of data using the Shrinkage Clustering algorithm. The algorithm clusters data and finds the optimal number of clusters at the same time. The package provides users with the customizability to set constraints on the minimum cluster size and maximum number of clusters so that the domain knowledge might be used to better perform the clustering task. Moreover, the package also provides easy methods to assess the clustering results and visualize the clustering assignments. Due the efficiency, accuracy, and flexibility of the shrinkage Clustering algorithm, we believe the shrinkageClust package will benefit R communities, especially in biomedical and clinical studies.

**Bibliography**

1. Hu, Chenyue W., Hanyang Li, and Amina A. Qutub. “Shrinkage Clustering: A Fast and Size-Constrained Clustering Algorithm for Biomedical Applications.” BMC Bioinformatics 19 (2018): 19. PMC. Web. 7 May 2018.
2. Manning CD, Raghavan P, Schütze H, et al. Introduction to information retrieval, vol. 1. Cambridge: Cambridge university press; 2008.
3. Street WN, Wolberg WH, Mangasarian OL. Nuclear feature extraction for breast tumor diagnosis. In: IS&T/SPIE’s Symposium on Electronic Imaging: Science and Technology. San Jose: International Society for Optics and Photonics; 1993. p. 861–70.
4. Mangasarian OL, Street WN, Wolberg WH. Breast cancer diagnosis and prognosis via linear programming. Oper Res. 1995;43(4):570–7.
5. Bache K, Lichman M. UCI Machine Learning Repository: University of California, Irvine, School of Information and Computer Sciences; 2013.

<http://archive.ics.uci.edu/ml>.