Austrian Journal of Statistics
2025, Volume 54, 27–42.
http://www.ajs.or.at/
doi:10.17713/ajs.v54i3.2061

circlus: An R Package for Circular and Spherical Clustering Using Poisson Kernel-Based and Spherical Cauchy Distributions

Lukas Sablica
WU Wien

Kurt Hornik
WU Wien

Bettina Grün
WU Wien

Abstract

This paper introduces **circlus**, an R package designed for clustering circular and spherical data using Poisson kernel-based (PKB) distributions and spherical Cauchy distributions. The package leverages the general framework for Expectation-Maximization (EM) estimation implemented by package **flexmix** and provides model drivers for estimating PKB and spherical Cauchy distributions in the components. The drivers implement two approaches for the M-step. The first is a direct maximization approach implemented in C++ via **Rcpp**, while the second incorporates covariates by solving the M-step using neural networks with the **torch** package. The package is particularly suited for high-dimensional clustering tasks, such as text embeddings on a spherical space, and supports models both with and without covariates. As a case study, we apply **circlus** to cluster the abstracts of papers co-authored by Fritz Leisch and demonstrate the use with and without the inclusion of co-author count as a covariate.

Keywords: spherical data, model-based clustering, embeddings, **flexmix**, R.

1. Introduction

Clustering is a fundamental technique in data analysis and machine learning, commonly used to uncover underlying patterns in data by grouping similar items. Traditional clustering methods, such as k-means (Macqueen 1967) and Gaussian mixture models (Dempster, Laird, and Rubin 1977), assume that data lie in Euclidean space, which works well for many applications. However, certain types of data, such as directional data, biological data, and text data, are often more appropriately modeled on spherical or circular spaces. In such cases, applying Euclidean-based methods can lead to suboptimal or misleading results.

The extension of k-means clustering to spherical data uses the cosine similarity as distance (Maitra and Ramler 2010) and an implementation for the R environment for statistical computing and graphics (R Core Team 2024) is available in package **skmeans** (Hornik, Feinerer, Kober, and Buchta 2012). Similar to how Gaussian model-based clustering is the generalization of k-means to a model-based approach (see, for example Grün 2019), mixtures of von Mises-Fisher distributions (Banerjee, Dhillon, Ghosh, and Sra 2005) have been proposed as generalization of spherical k-means and an R implementation is available in package **movMF**

(Hornik and Grün 2014).

Model-based clustering of spherical data based on finite mixtures is provided for specific component distributions by separate R packages. E.g., package **movMF** provides fitting of finite mixtures of von Mises-Fisher distributions and package **QuadratiK** (Saraceno, Markatou, Mukhopadhyay, and Golzy 2024) covers finite mixtures of Poisson-kernel-based distributions. However, neither of these packages allows the inclusion of covariates to control for differences in the component-specific parameters in dependence of covariates. A more general implementation within the **flexmix** framework (Leisch 2004), allowing for different component distributions and the inclusion of covariates, is thus warranted. This gap is filled by **circlus** (Sablica, Hornik, Gruen, and Leydold 2024), the R package we introduce in this paper and that is freely available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=circlus`.

Building on the solid foundation provided by **flexmix**, **circlus** extends its capabilities to circular and spherical clustering by allowing to specify as component distributions the Poisson kernel-based distribution (Golzy and Markatou 2020) and the spherical Cauchy distribution (Kato and McCullagh 2020). Package **circlus** contributes new models that enable the clustering of data on the surface of a sphere. Two estimation methods for the M-step are offered for each of the two distributions: one implemented in C++ for direct and efficient calculation, and another using neural networks via the **torch** package (Falbel and Luraschi 2024), which allows for the incorporation of covariates into the clustering process. This neural network approach maps the covariate space to clustering parameters, facilitating the inclusion of additional data, such as metadata or context, in the clustering model.

The rest of this paper is organized as follows: in the next section, we define the Poisson kernel-based and spherical Cauchy distributions that underlie the models in package **circlus**. In Section 3, we discuss the strengths and advantages of clustering on the sphere compared to Euclidean methods. Section 4 introduces the **circlus** software package, detailing its architecture and implementation. This is followed by an application section, where we demonstrate package **circlus** in action by clustering abstracts written by Fritz Leisch, with and without the inclusion of co-author count as a covariate. Finally, we conclude by summarizing the key contributions of package **circlus**.

2. Spherical distributions for clustering

Various rotationally symmetric distributions have been developed for modeling data on the unit sphere, including the von Mises-Fisher (vMF) distribution (Khatri and Mardia 1977), Poisson kernel-based distribution (Golzy and Markatou 2020), and spherical Cauchy distribution (Kato and McCullagh 2020). In more detail, these are as follows.

von Mises-Fisher (vMF) distribution. Let $S^{d-1} = \{x \in \mathbb{R}^d : \|x\| = 1\}$ represent the unit sphere in $\mathbb{R}^d$. A random vector $x \in S^{d-1}$ has a von Mises-Fisher (vMF) distribution with parameters $\kappa \geq 0$ and $\mu \in S^{d-1}$ if its probability density function is given by

$$f_{\mathrm{vMF}}(x|\kappa, \mu) = \frac{e^{\kappa \mu' x}}{H_{d/2-1}(\kappa)},$$

where $H_\nu(\kappa) = {}_0F_1(; \nu + 1; \kappa^2/4) = \frac{\Gamma(\nu+1)}{(\kappa/2)^\nu} I_\nu(\kappa)$ with ${}_0F_1$ and I_ν being the confluent hypergeometric limit function (e.g., Mardia and Jupp 2009, page 352) and modified Bessel function of the first kind (DLMF 2024, Eq. 10.25.2), respectively. The vMF distribution is widely used for spherical data due to its simplicity, with a concentration parameter κ determining the level of clustering and μ as the location.

However, due to the exponential decay of its density function, the vMF distribution can struggle in scenarios with outliers or broader variability, as it tends to sharply cluster data around the mean direction. This makes it less suitable for datasets that require more flexibility in capturing heavy-tailed structures. Additionally, while a closed-form expression for the normalizing constant exists, its computation can be numerically demanding and can easily overflow for large parameter values (Hornik and Grün 2014).

Poisson kernel-based (PKB) distribution. The Poisson kernel-based (PKB) distribution provides an alternative with better stability and computational efficiency. The PKB distribution with parameters $0 \leq \rho < 1$ and $\mu \in S^{d-1}$ has the following density function with respect to the uniform distribution on the unit sphere:

$$f_{\text{PKB}}(x|\rho,\mu) = \frac{1-\rho^2}{\|x-\rho\mu\|^d}, \quad x \in S^{d-1}.$$

For $\rho = 0$, this distribution reduces to the uniform distribution on the sphere, and as $\rho \to 1^-$, it tends toward the Dirac distribution centered at μ. The PKB distribution belongs to a family of densities of the form:

$$f(x) \propto \|x-\rho\mu\|^{-\xi}, \quad x \in S^{d-1}, \quad \xi > 0.$$

The PKB distribution arises for $\xi = d$, making it particularly useful for modeling spherical data. One key advantage is that the PKB distribution allows for straightforward density evaluation without the need for complex special functions, unlike the vMF or Watson distributions (Sablica and Hornik 2023).

Spherical Cauchy distribution. The spherical Cauchy distribution is another member of the same family of distributions. Its density is closely related to that of the PKB distribution, and the two distributions coincide when $d = 2$. The density of the spherical Cauchy distribution with parameters $0 \leq \rho < 1$ and $\mu \in S^{d-1}$ with respect to the uniform distribution on the unit sphere is given by:

$$f_{\text{Cauchy}}(x|\rho,\mu) = \left(\frac{1-\rho^2}{\|x-\rho\mu\|^2}\right)^{d-1}, \quad x \in S^{d-1}.$$

Similar to the PKB distribution, when $\rho = 0$, the distribution reduces to the uniform distribution on the sphere, and as $\rho \to 1^-$, it tends toward the Dirac distribution centered at μ.

The spherical Cauchy and PKB distributions have the following advantages compared to the vMF distribution: (1) They have heavier tails, making them ideal for capturing large deviations and outliers, much like the role of the Cauchy and Student-t distributions in Euclidean space. (2) They are much simpler and computationally more efficient to evaluate on modern accelerators such as GPUs. Both distributions avoid the need for computing complex normalizing constants that must be sequentially evaluated, which would otherwise hinder parallel processing on GPUs. The density evaluation for both PKB and spherical Cauchy distributions essentially reduces to matrix operations, such as computing norms and matrix multiplications, which are highly optimized for GPU architectures. This allows for efficient and scalable implementation of spherical clustering, making these distributions particularly well-suited for modern, large-scale data processing tasks.

The PKB has slightly heavier tails than the spherical Cauchy, offering a good option when dealing with data containing more extreme outliers. The spherical Cauchy distribution provides a balance between the traditional von Mises-Fisher and the extremely heavy-tailed PKB, making it suitable for data with moderate outliers or when a balance between robustness and computational efficiency is desired.

3. Clustering on the sphere

3.1. Spherical clustering for high-dimensional data

Clustering data on the sphere has become increasingly important with the rise of high-dimensional data representations, particularly in natural language processing and machine learning. Embeddings, such as those derived from models like BERT (Devlin, Chang, Lee, and Toutanova 2019) or other transformer-based architectures, are often normalized to lie on the surface of a unit sphere. This normalization occurs because the magnitude of the embeddings, which represent the strength or scale of the data points, is irrelevant in most contexts, what matters is their direction. Clustering on the sphere allows for a better understanding of relationships between data points, as it operates in the correct geometric space, making the results more accurate and meaningful.

3.2. Regression-based clustering with covariates

While spherical clustering is powerful on its own, adding the ability to incorporate covariates into the clustering model further enhances its usefulness. Covariates provide a way to control for known factors in the data that could influence clustering, allowing the model to focus on discovering more subtle or latent patterns.

In many practical applications, the data being clustered comes with associated metadata or known characteristics that can be incorporated into the model. For instance, imagine clustering the text embeddings of financial reports from various companies. Without any additional information, the clustering algorithm might primarily group companies based on their industry or market segment, which is an obvious correlation often reflected in the text of such reports.

By incorporating covariates, we can "control" for this known information and allow the model to focus on other latent patterns within the data. For example, if we input the market segment as a covariate into the clustering model, the algorithm can focus on distinguishing companies within the same market segment based on their risk exposure, financial strategy, or other factors that are not immediately obvious from surface-level industry groupings.

Another example is the clustering of patient health records, incorporating age, gender, or pre-existing conditions as covariates. This approach could allow the model to focus on discovering patterns related to treatment effectiveness, lifestyle impacts, or specific health risks that are not simply reducible to demographic categories.

More formally, this approach can be viewed within the framework of model-based clustering. In this context, we aim to maximize the likelihood of a multivariate mixture model, given by:

$$f(y_1, \dots, y_n | x_1, \dots, x_n, \pi_1, \dots, \pi_K, A_1, \dots, A_K) = \prod_{i=1}^{n} \sum_{j=1}^{K} \pi_j f_j(y_i | A_j, x_i),$$

where $y_i \in S^{d-1}$ is the observed spherical response and $x_i \in \mathbb{R}^m$ the covariate vector for observation i, π_j is the prior probability of belonging to cluster j (with $\sum_{j=1}^{K} \pi_j = 1$), $f_j(y_i|A_j, x_i)$ is the spherical density function for cluster j, and K is the number of clusters. In our case, f_j could be the PKB distribution or the spherical Cauchy distribution.

To incorporate covariates into this framework, we link the parameters of the spherical distribution, μ (location) and ρ (concentration), to the covariates x_i through a cluster-specific linear map represented by the matrix $A_j \in \mathbb{R}^{m \times d}$. This mapping is given by:

$$\theta_{j,i} = A_j' x_i,$$

where $\theta_{j,i} \in \mathbb{R}^d$ is an unrestricted parameter vector characterizing the spherical distribution for cluster j. The matrices A_j contain the learnable parameters and can be estimated using optimization techniques such as those employed in neural networks.

Since μ must lie on the unit sphere and ρ must remain positive and bounded, a direct linear mapping is insufficient. For that reason we then link $\theta_{j,i}$ to the parameters μ and ρ through suitable transformations

$$\mu(\theta_{j,i}) = \frac{\theta_{j,i}}{\|\theta_{j,i}\|}, \quad \text{and} \quad \rho(\theta_{j,i}) = \frac{\|\theta_{j,i}\|}{1 + \|\theta_{j,i}\|}.$$

This mapping provides a 1-to-1 correspondence between $\mathbb{R}^d$ and the set of parameters (μ, ρ), ensuring that μ is always a unit vector and that ρ remains within the valid range $(0, 1)$, thereby guaranteeing well-defined model parameters. While this parametrization does not allow for completely independent modeling of location and concentration, in our experiments, it has proven general enough to provide robust estimates while offering the advantages of computational speed and simplicity. More flexible parametrizations, allowing for more nuanced relationships between covariates and concentration, could be explored in future research.

This model-based clustering framework, with the inclusion of covariates, allows us to group together observations that exhibit similar relationships between covariates and the parameters of the spherical response distribution. Essentially, we are clustering observations based on the similarity of their covariate-response mappings.

In the case of only one mixture component ($K = 1$), this simplifies to a simple regression task, as we are effectively estimating a single mapping from covariates to the response distribution. Conversely, with only an intercept as a covariate, we recover standard clustering of the responses, as the model focuses solely on grouping similar responses without considering any additional covariate information. The mixture likelihood can be estimated using various methods, including the Expectation-Maximization (EM) algorithm (Dempster *et al.* 1977).

4. Software

The **circlus** package extends the flexibility of the EM framework implemented in package **flexmix** to handle spherical clustering using Poisson kernel-based and spherical Cauchy distributions. The package implements four M-step drivers, which provide the two distributions (PKB and spherical Cauchy distribution), with and without covariates. Each model is designed to integrate seamlessly into the **flexmix** implementation, providing methods for maximum likelihood estimation through the EM algorithm (Dempster *et al.* 1977). Below, we introduce the key functions for clustering and explain their functionality and parameters.

4.1. Clustering with PKB distributions

The **circlus** package provides two main functions for performing the M-step of clustering using the Poisson kernel-based distribution:

```
FLXMCpkb(formula = .~.)
```

and

```
FLXMRpkb(formula = .~., EPOCHS = 100, LR = 0.1,
          max_iter = 200, adam_iter = 5, free_iter = adam_iter,
          line_search_fn = "strong_wolfe")
```

The first function, FLXMCpkb, uses C++ code to perform the M-step, making it highly efficient and well-suited for tasks where speed is crucial. The "C" in FLXMCpkb stands for "clustering", indicating that this function focuses purely on clustering without the inclusion of covariates. The second function, FLXMRpkb, leverages neural networks through the **torch** package to incorporate covariates into the estimation process, with the "R" standing for "regression", signifying the function's ability to handle covariates in the clustering model.

The FLXMCpkb function performs the M-step using a direct C++ implementation via **Rcpp** (Eddelbuettel and François 2011) based on Golzy and Markatou (2020), which is designed to handle the estimation process efficiently without the need for any additional optimization frameworks. This approach excels in both speed and simplicity, making it ideal for scenarios where covariates are not needed. We note that this algorithm has also been implemented in pure R within the **QuadratiK** package (Saraceno *et al.* 2024), which offers a robust suite of methods for working with spherical data, including tests for multivariate normality, tests for uniformity on the sphere, and clustering algorithms, among other valuable tools. To extend the possible applications, particularly for models that incorporate covariates, enhance performance through C++, and leverage the wide range of functions offered by the **flexmix** framework, we developed **circlus**. By building on the strengths of existing tools, **circlus** offers users additional flexibility and scalability, particularly for larger datasets and more complex clustering tasks involving covariates.

The FLXMRpkb function, on the other hand, uses a neural network to perform the M-step. Function FLXMRpkb can be used with or without inclusion of covariates into the clustering model. More specifically, in case covariates are included, the algorithm maps the covariate space to the space of response variables using a simple linear transformation network without a bias term and links the mapped vector to parameters μ and ρ exactly as discussed in Section 3.2.

The optimization process in FLXMRpkb starts with the robust Adam optimizer (Kingma and Ba 2014) and resets the weights of the neural network at every iteration to prevent local minima and ensure robustness in the early stages of training. After an initial phase controlled by the adam_iter parameter, the algorithm switches to the quasi-Newton L-BFGS method, which is better suited for fast convergence in the later stages of optimization. The number of epochs for the Adam optimizer and the maximum iterations for L-BFGS are controlled by the EPOCHS and max_iter parameters, respectively. The learning rate for both optimizers is set by the LR parameter. Additionally, the line_search_fn parameter specifies the line search function used in the L-BFGS optimizer, with the "strong_wolfe" method being the default. For more details on this parameter, see the documentation of package **torch**.

4.2. Clustering with spherical Cauchy distributions

The **circlus** package also provides two key functions for clustering based on the spherical Cauchy distribution:

```
FLXMCspcauchy(formula = . ~ .)
```

and

```
FLXMRspcauchy(formula = . ~ ., EPOCHS = 100, LR = 0.1,
              max_iter = 200, adam_iter = 5, free_iter = adam_iter,
              line_search_fn = "strong_wolfe")
```

These functions follow a similar design to the PKB distribution functions, leveraging the **flexmix** framework to perform model-based clustering with and without covariates.

The FLXMCspcauchy function provides a direct, fast, and efficient solution for spherical Cauchy clustering without covariates, utilizing C++ for the M-step. It uses the method of Algorithm 4.1 in Kato and McCullagh (2020). This algorithm is extended to cover the mixture model case by weighting the sum by the posterior probabilities of the data points belonging to each cluster, with the weights normalized to sum to 1 for each cluster. The initial ψ_0 for the algorithm is estimated using the method of moments, as outlined in Subsection 4.1 of the reference, where $\bar{Y}$ in Equation 4.1 is also weighted according to the posterior probabilities. In addition, `FLXMRspcauchy` incorporates covariates into the clustering process using a neural network based on the **torch** package. The neural network maps covariates to the parameters of the spherical Cauchy distribution in the same manner as the PKB distribution model.

To our knowledge, there are currently no other implementations available that provide spherical Cauchy-based clustering with or without covariates, making **circlus** the first package to offer this capability within the **flexmix** framework.

4.3. Simulation methods

In addition to the clustering functions, the **circlus** package also provides random sampling methods for both the Poisson kernel-based and spherical Cauchy distributions. These random sampling methods are valuable because they allow for further analysis of clusters through techniques like the parametric bootstrap, where one can assess the variability or stability of the identified clusters by resampling from the fitted model. For instance, in applications such as text analysis or financial modeling, simulated data can be used to validate model performance or to test the sensitivity of clustering results. See for example McLachlan (1987) and O'Hagan, Murphy, Scrucca, and Gormley (2019).

Function `rpkb(n, rho, mu, method = "ACG")` generates random samples from the PKB distribution. The user can specify the number of random draws `n` and the desired parameters `rho` (the concentration) and `mu` (the location). The `method` argument allows the user to choose between two sampling approaches: the first uses the Angular Central Gaussian (ACG) distribution as the envelope in a rejection sampling scheme, while the second method is based on the projected Saw distribution. Both methods are efficient and follow the approach described in Sablica, Hornik, and Leydold (2023). The ability to switch between these methods offers flexibility depending on the specific use case or computational requirements. We note that PKB random sample generation is also available in the packages **QuadratiK** and **Directional** (Tsagris, Athineou, Adam, and Yu 2024).

For the spherical Cauchy distribution, function `rspcauchy(n, rho, mu)` provides a direct method for generating random samples. The number of samples `n`, concentration `rho`, and location `mu` can be specified. This method is based on the Möbius transformation of uniform samples on the sphere, as detailed in Kato and McCullagh (2020).

5. Case study: Clustering Fritz Leisch's work

Friedrich "Fritz" Leisch was a highly respected figure in the field of statistical computing, known for his broad contributions that spanned multiple domains. His work on flexible clustering models has had a lasting impact on the world of data analysis. Leisch's research was not only theoretically innovative but also highly practical, enabling users across disciplines to apply sophisticated clustering techniques to real-world problems.

Throughout his prolific career, Leisch was involved in a wide range of research topics, contributing to areas such as benchmarking, computational statistics, and reproducible research. His work was characterized by a collaborative spirit, with nearly 300 unique co-authors, reflecting his strong belief in interdisciplinary research and the importance of working with others to advance science. Leisch's collaborations and innovations have helped shape modern statistical methodology, making his work essential reading for statisticians and data scientists alike.

For this analysis, we compiled a dataset of 129 abstracts from the works of Fritz Leisch, which were verified through the Crossref API (CrossRef 2024). Alongside the abstracts, we collected important metadata, including the number of pages, digital object identifier (DOI), journal name, names of the co-authors, and the year of publication. The dataset is made available as `Abstracts` within the **circlus** package. Co-author information has been encoded using 272 dummy variables, where each co-author is represented as a binary variable. To numerically represent the textual data, we transformed the abstracts into embeddings using four different models. The first method employed the `gte-large-en-v1.5` embedding model from Alibaba (Zhang, Zhang, Long, Xie, Dai, Tang, Lin, Yang, Xie, Huang *et al.* 2024), which produced embeddings with 1024 dimensions. The remaining three methods used OpenAI's `text-embedding-3-large` model (OpenAI 2024), with output dimensions of 3072, 512, and 256, respectively. These embeddings outputs are available as the last four columns of the `Abstracts` dataset. Overall this results in a dataset with 129 rows and 283 columns, offering a comprehensive view of Leisch's collaborations across different publications.

Given the relatively small number of abstracts and their thematic similarity, we selected the 256-dimensional embeddings for our analysis. We found that this dimensionality was sufficient to capture the essential semantic relationships between the abstracts without overcomplicating the clustering process. In general, the choice of dimensionality should be guided by a combination of factors such as sample size, data complexity, and computational constraints. For larger or more complex datasets, higher-dimensional embeddings might be necessary, but it is important to balance the need for detail with the risk of overfitting and increased computational burden. It is recommended to explore different dimensionalities and embedding models to find the best fit for the specific data and task.

5.1. Mixtures of distributions

In the first stage of our analysis, we clustered the dataset without incorporating any covariates. Assuming that the word usage distribution differs by research area, this approach aims to cluster the abstracts such that the clusters correspond to research areas. The estimation of the mixture model was carried out using the default parameters of **flexmix** for the EM algorithm. This implies that the EM algorithm is randomly initialized by assigning a-posteriori probabilities to observations and then continuing with an M-step. When specifying only the number of clusters, observations are randomly assigned to clusters with equal probability and weights of 0.9 assigned to these clusters and weights of 0.1 for the remaining ones. In addition the parameter `minprior` is set to `0.05`. This parameter ensures that any cluster representing less than 5% of the data is eliminated from the estimation process during the EM iterations. This provides a more stable estimation avoiding estimation issues in the M-step in case of very small components which could result in degenerate solutions where only observations with identical values have positive weights for this component. In addition, a minimum cluster size is usually of interest for an interpretable solution. In our experiments where we fitted the spherical cluster model with different, higher numbers of components, the model consistently reduced to eight clusters during the EM algorithm. We adopted this as the final number of clusters for our analysis. This number of clusters provided a meaningful balance between interpretability and capturing the diversity of research topics within the dataset.

We chose the spherical Cauchy distribution for this case study, as the even heavier tails of the PKB distribution were not necessary for data that is relatively similar. All abstracts represent scientific contributions in neighboring disciplines. The clustering was performed using both available models in the **circlus** package: `FLXMCspcauchy`, which leverages the direct estimation, and `FLXMRspcauchy`, which incorporates a neural network to estimate the M-step. The code used for this analysis is shown below, with the clustering results following.

First we loaded the necessary packages and the dataset as well as extracted the embedding obtained for OpenAI with dimension 256:

```
R> library("flexmix")
R> library("circlus")
```

```
R> data("Abstracts", package = "circlus")
R> OAI256 <- do.call(rbind, Abstracts[, "OpenAI_embeddings256"])
```

We applied the FLXMCspcauchy() model:

```
R> set.seed(1)
R> (SC_abstract_8 <- flexmix(OAI256 ~ 1, k = 8,
+    model = FLXMCspcauchy()))

Call:
flexmix(formula = OAI256 ~ 1, k = 8, model = FLXMCspcauchy())

Cluster sizes:
 1  2  3  4  5  6  7  8
15  9 16 19  9 16 19 26

convergence after 16 iterations
```

The default print() method for objects fitted with flexmix() indicates how the object was created by showing the call as well as the cluster sizes of the partition obtained by assigning observations to the cluster where their a-posteriori probability is maximum. The output shows that the clusters vary in size and have between 9 and 26 abstracts assigned.

Next, we used the neural network-based model for comparison:

```
R> set.seed(1)
R> torch::torch_manual_seed(1)
R> (SCNN_abstract_8 <- flexmix(OAI256 ~ 1, k = 8,
+    model = FLXMRspcauchy(LR = 0.02, adam_iter = 0, free_iter = 5)))

Call:
flexmix(formula = OAI256 ~ 1, k = 8, model = FLXMRspcauchy(LR = 0.02,
adam_iter = 0, free_iter = 5))

Cluster sizes:
 1  2  3  4  5  6  7  8
15  9 16 19  9 16 19 26

convergence after 18 iterations
```

Both models resulted in identical cluster allocations, with each cluster containing the same number of abstracts. Setting the random seed to the same value ensures that the EM algorithm is initialized using the same a-posteriori probabilities for the first M-step. Hence, only the optimization step in the M-step differs for these model fits. This congruence in results shows that the M-step implementations are robust and produce consistent clustering outcomes, even though they rely on different optimization techniques. While it is common for estimates obtained with different optimization algorithms to yield slight different results, in this case, the resulting cluster allocation sizes were fully aligned, highlighting the reliability of the estimation for this particular dataset.

In terms of log-likelihood, FLXMCspcauchy achieved a final log-likelihood of 12674.6, while FLXMRspcauchy reached 12674.61, confirming that both methods converged to virtually identical solutions.

We analyzed the content of the abstracts within each cluster to verify that indeed spherical clustering identified eight distinct clusters that represent different areas of Fritz Leisch's research. Based on this inspection of the content, we classified the clusters and assigned titles as shown in Table 1 which capture the primary scientific focus of each group.

Table 1: Cluster titles based on abstract content

Cluster No.	Cluster Title
1	Genetic Influences on Psychiatric and Behavioral Disorders
2	Advancements in Model Validation and Benchmarking Techniques
3	Travel Behavior and Environmental Impact in Transportation and Tourism
4	Environmental and Biological Effects of Agrochemicals
5	Market Segmentation Techniques and Applications
6	Biopharmaceutical Production Through Data-Driven Approaches
7	Finite Mixture Models and Their Applications
8	Clustering Techniques in Data Analysis

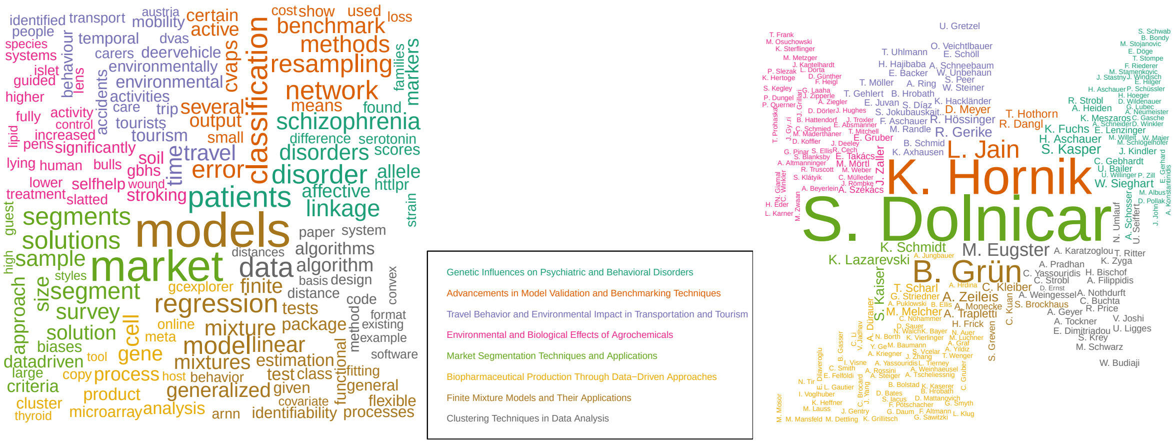

Figure 1: Word cloud visualization of the most frequently occurring terms (left) and co-author networks (right) across the clusters

Inspecting the concentration parameters of the clusters indicates how compact or spread out the identified clusters are. Clusters 4 and 8 have the smallest concentration parameter values, with $\rho = 0.346$ for Cluster 4 and $\rho = 0.397$ for Cluster 8, indicating that they represent the least compact clusters. These clusters include observations that are not easily assigned to the more specialized clusters, serving as broader, less-defined groups that capture data points with weaker associations to the other, more focused clusters. Cluster 8 (titled "Clustering Techniques in Data Analysis"), in particular, acts as a background cluster for the more data-driven clusters such as Clusters 2, 6, and 7, as can be seen by comparing the cosine distances between the location parameters μ of the individual clusters.

To further visualize the content of these clusters, we calculated a term frequency matrix for the dataset as a whole with respect to the individual clusters. This enabled us to identify the most frequently occurring terms within each cluster. Using the **wordcloud** package (Fellows 2018), we created visual representations of the key terms for each cluster (see the left sub-image of Figure 1).

In addition to the thematic analysis, we aimed to highlight the extensive network of collaborations that Fritz Leisch fostered throughout his career. For each of the eight clusters, we generated a co-author frequency matrix, which quantified the presence of each co-author within the publications assigned to that cluster. Using the `comparison.cloud()` function from the **wordcloud** package, we visualized the co-author networks for each cluster in the right sub-image of Figure 1, showcasing the diverse and widespread collaborations across different fields of research.

One key insight from examining the dataset is the variation in the number of co-authors across

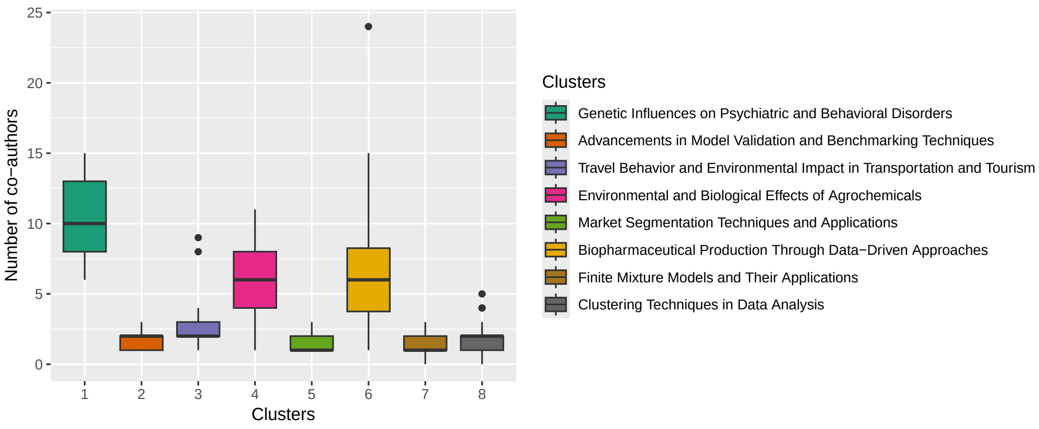

Figure 2: Boxplot of the number of co-authors across clusters

different scientific disciplines. Certain fields tend to have more co-authors per publication, reflecting the collaborative nature of these research areas. This variation is also evident in the clusters identified which correspond to different research areas where Fritz Leisch has contributed to. Figure 2 visualizes the number of co-authors for each of the eight clusters in a parallel boxplot.

5.2. Mixtures with covariates

As illustrated in Figure 2, clusters corresponding to genetics, biology and biopharmaceutics generally exhibit a higher number of co-authors compared to other clusters. This is valuable information that can be explicitly included in our clustering model as a covariate, allowing the model to account for the number of co-authors while focusing more on other abstract features such as the content and semantic relationships within the embeddings.

To incorporate this into our analysis, we re-estimated the model using the number of co-authors as a covariate and the spherical Cauchy distribution as the component distribution. The following code shows the estimation process:

```
R> (SCNN_abstract_8b <- flexmix(OAI256 ~ 1 + num_of_coauthors, k = 8,
+    model = FLXMRspcauchy(EPOCHS = 200, LR = 0.02, adam_iter = 10)))

Call:
flexmix(formula = OAI256 ~ 1 + num_of_coauthors, k = 8, model =
FLXMRspcauchy(EPOCHS = 200, LR = 0.02, adam_iter = 10))

Cluster sizes:
 1  2  3  4  5  6
15 30 27 15 19 23

convergence after 28 iterations
```

As seen from the output, while we initially started with eight clusters, the automatic removal of clusters with a small component size during the iterative procedure of the EM algorithm reduced the number of clusters to six clusters. Our experiments revealed that incorporating the number of co-authors as a covariate often leads to fewer clusters, as the model uses the covariate information to account for variation between abstracts within a cluster depending on the number of co-authors. In this case, the achieved log-likelihood was 14192.16, which indicates that the model found a better fit with six clusters, compared to the previous clustering solution that used eight clusters without covariates.

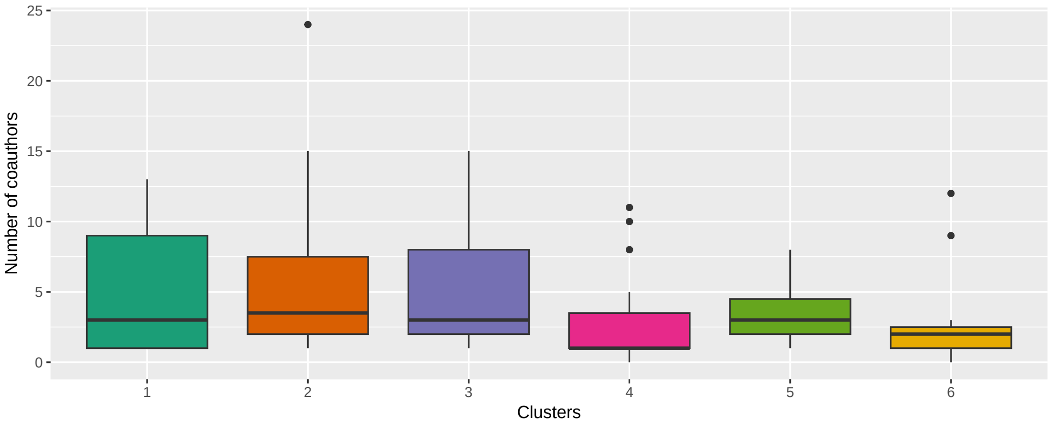

Figure 3: Boxplot of the number of co-authors in the cluster solution with covariates

By including the number of co-authors, the model successfully captures the variation in the number of collaborators across different research areas, without needing to rediscover this pattern. This allows the model to focus on other important relationships in the data, such as thematic content or research methodologies. To illustrate this further, we can once again plot the number of co-authors across the newly formed six clusters using a parallel boxplot. As shown in Figure 3, the number of co-authors across the six clusters has become more uniform, which demonstrates that the covariate was successfully controlled for in the model. This approach shows the power of incorporating known metadata into clustering models to reduce the number of clusters and account for within-cluster heterogeneity due to this covariate in an efficient way.

When we compare the clustering results before and after including the number of co-authors as a covariate, we can observe notable shifts in how the abstracts are assigned to clusters. The addition of this covariate introduces a significant factor in determining cluster membership, leading to observable movements of certain abstracts between clusters. The table below shows the comparison, where the rows represent the assignments with covariate and the columns represent the cluster assignments without covariate:

```
R> table(with_num_of_coauthors = clusters(SCNN_abstract_8b),
+     without_num_of_coauthors = clusters(SC_abstract_8)
```

```
                    without_num_of_coauthors
with_num_of_coauthors  1  2  3  4  5  6  7  8
                    1  6  0  0  2  7  0  0  0
                    2  0  0 16 11  0  3  0  0
                    3  9  9  0  2  0  1  0  6
                    4  0  0  0  4  2  0  0  9
                    5  0  0  0  0  0 10  0  9
                    6  0  0  0  0  0  2 19  2
```

We observe that Clusters 2, 3, 5, and 7 of the clustering solution without covariates exhibit little to no movement (with respective changes of 0, 0, 2, and 0 abstracts). In contrast, the other clusters get moved considerably. In particular, Cluster 8, which had previously been identified as the background cluster for the more statistically oriented clusters, shows the strongest decomposition when the number of co-authors is included. This is likely due to the catch-all nature of this cluster in the solution without covariates, which captured abstracts that did not fit neatly into more specific research categories. Cluster 4, which has an even smaller concentration parameter than Cluster 8, also shows significant decomposition, further underscoring its role as one of the least compact clusters in the clustering without covariates.

By incorporating co-author information, we decompose these background clusters, leading to a clearer separation of abstracts and improving the overall clustering model.

One of the strengths of the **flexmix** package is the suite of high-level functions it offers for investigating clustering results, such as the `summary` and `plot` methods for objects returned by `flexmix`. These methods focus on providing valuable insights into the quality of the clustering, i.e., indicate how close the a-posteriori probabilities are to a 0-1 distribution. The `summary` function shows key metrics like the prior probabilities (the overall proportion of data belonging to each cluster), the number of assigned observations per cluster, and the ratio of assigned observations to those with a positive posterior probability (by default using a threshold of `eps = 1e-04`). This ratio is particularly useful for understanding how well-separated each cluster is from the others. A high ratio (close to 1) indicates that most assigned observations have a strong affinity for their assigned cluster and are unlikely to belong to other clusters. A lower ratio suggests some observations might have comparable probabilities for multiple clusters, indicating potential overlap.

```
R> summary(SCNN_abstract_8b)

Call:
flexmix(formula = OAI256 ~ 1 + num_of_coauthors, k = 8,
model = FLXMRspcauchy(EPOCHS = 200, LR = 0.02, adam_iter = 10))

       prior size post>0 ratio
Comp.1 0.116   15     15 1.000
Comp.2 0.233   30     34 0.882
Comp.3 0.209   27     29 0.931
Comp.4 0.116   15     15 1.000
Comp.5 0.147   19     19 1.000
Comp.6 0.178   23     23 1.000

'log Lik.' 14192.16 (df=3077)
AIC: -22230.32   BIC: -13430.67
```

In our results, four clusters (1, 4, 5, and 6) are perfectly separated with a ratio of 1.000, while Clusters 2 and 3 show some overlap, with ratios of 0.882 and 0.931 respectively. This indicates that there are observations which have some posterior probability to be from Clusters 2 and 3 but are eventually assigned to a different cluster. But overall, the clusters are well separated, reflecting a strong clustering fit with a clear assignment of observations to clusters.

6. Conclusion

In this paper, we introduced the **circlus** package, which extends the EM framework implemented by package **flexmix** to perform spherical clustering using the Poisson kernel-based and spherical Cauchy distributions. By providing efficient C++ implementations and flexible neural network-based models, package **circlus** allows for clustering on the sphere, with and without covariates. The inclusion of covariate-based models adds a significant advantage, enabling users to account for known metadata and focus the clustering process on uncovering deeper relationships within the data.

Our case study of Fritz Leisch's published works demonstrates the practical application of spherical clustering, highlighting how different research areas can be clustered based on their abstract embeddings. By incorporating metadata, such as the number of co-authors, we further showed how covariates can enhance the clustering model's interpretability and accuracy. This approach allows the model to focus on more nuanced aspects of the data, leading to alternative clustering solutions.

The results of our analysis reveal that spherical clustering, combined with covariate information, offers a powerful tool for handling high-dimensional and complex datasets, such as text embeddings. The flexibility of the **circlus** package makes it suitable for a wide range of applications, from natural language processing to biological and social sciences, where data naturally lie on a sphere.

Overall, the **circlus** package builds upon the legacy of Fritz Leisch's contributions to statistical computing, offering modern and scalable tools for model-based clustering on spherical spaces. We hope that this work will continue to support research in these areas and inspire further advancements in clustering methodology and statistical modeling.

References

Banerjee A, Dhillon IS, Ghosh J, Sra S (2005). "Clustering on the Unit Hypersphere Using von Mises-Fisher Distributions." *Journal of Machine Learning Research*, **6**(September), 1345–1382. URL https://jmlr.csail.mit.edu/papers/v6/banerjee05a.html.

CrossRef (2024). "CrossRef REST API." Available at https://api.crossref.org.

Dempster AP, Laird NM, Rubin DB (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1), 1–22. doi:10.1111/j.2517-6161.1977.tb01600.x.

Devlin J, Chang MW, Lee K, Toutanova K (2019). "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." In *Proceedings of NAACL-HLT 2019*, pp. 4171–4186. URL https://aclanthology.org/N19-1423.pdf.

DLMF (2024). "*NIST Digital Library of Mathematical Functions*." Release 1.0.19 of 2018-06-22. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds., URL https://dlmf.nist.gov/.

Eddelbuettel D, François R (2011). "**Rcpp**: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.

Falbel D, Luraschi J (2024). ***torch**: Tensors and Neural Networks with GPU Acceleration*. R package version 0.13.0, URL https://torch.mlverse.org/docs.

Fellows I (2018). ***wordcloud**: Word Clouds*. R package version 2.6, URL https://CRAN.R-project.org/package=wordcloud.

Golzy M, Markatou M (2020). "Poisson Kernel-Based Clustering on the Sphere: Convergence Properties, Identifiability, and a Method of Sampling." *Journal of Computational and Graphical Statistics*, **29**(4), 758–770. doi:10.1080/10618600.2020.1740713.

Grün B (2019). "Model-based Clustering." In S Frühwirth-Schnatter, G Celeux, CP Robert (eds.), *Handbook of Mixture Analysis*, Chapman & Hall/CRC Handbooks of Modern Statistical Methods, pp. 157–192. Chapman and Hall/CRC.

Hornik K, Feinerer I, Kober M, Buchta C (2012). "Spherical k-Means Clustering." *Journal of Statistical Software*, **50**(10), 1–22. doi:10.18637/jss.v050.i10.

Hornik K, Grün B (2014). "**movMF**: An R Package for Fitting Mixtures of Von Mises-Fisher Distributions." *Journal of Statistical Software*, **58**(10), 1–31. doi:10.18637/jss.v058.i10.

Kato S, McCullagh P (2020). "Some Properties of a Cauchy Family on the Sphere Derived from the Möbius Transformations." *Bernoulli*, **26**(4). doi:10.3150/20-bej1222.

Khatri CG, Mardia KV (1977). "The von Mises-Fisher Matrix Distribution in Orientation Statistics." *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1), 95–106. doi:10.1111/j.2517-6161.1977.tb01610.x.

Kingma DP, Ba J (2014). "Adam: A Method for Stochastic Optimization." doi:10.48550/arXiv.1412.6980. ArXiv Preprint arXiv:1412.6980.

Leisch F (2004). "FlexMix: A General Framework for Finite Mixture Models and Latent Class Regression in R." *Journal of Statistical Software*, **11**(8), 1–18. doi:10.18637/jss.v011.i08.

Macqueen J (1967). "Some Methods for Classification and Analysis of Multivariate Observations." In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press.

Maitra R, Ramler IP (2010). "A k-Mean-Directions Algorithm for Fast Clustering of Data on the Sphere." *Journal of Computational and Graphical Statistics*, **19**(2), 377–396. doi:10.1198/jcgs.2009.08054.

Mardia KV, Jupp PE (2009). *Directional Statistics*, volume 494. John Wiley & Sons.

McLachlan GJ (1987). "On Bootstrapping the Likelihood Ratio Test Stastistic for the Number of Components in a Normal Mixture." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **36**(3), 318–324. doi:10.2307/2347790.

OpenAI (2024). "Text-Embedding-3-Large Model." URL https://openai.com/index/new-embedding-models-and-api-updates/.

O'Hagan A, Murphy TB, Scrucca L, Gormley IC (2019). "Investigation of Parameter Uncertainty in Clustering Using a Gaussian Mixture Model via Jackknife, Bootstrap and Weighted Likelihood Bootstrap." *Computational Statistics*, **34**(4), 1779–1813. doi:10.1007/s00180-019-00897-9.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Sablica L, Hornik K (2023). "Family of Integrable Bounds for the Logarithmic Derivative of Kummer's Functions." *Journal of Mathematical Analysis and Applications*, **537**(1), 128262. doi:10.1016/j.jmaa.2024.128262.

Sablica L, Hornik K, Gruen B, Leydold J (2024). ***circlus**: Clustering and Simulation of Spherical Cauchy and PKBD Models*. R package version 0.0.1, URL https://CRAN.R-project.org/package=circlus.

Sablica L, Hornik K, Leydold J (2023). "Efficient Sampling from the PKBD Distribution." *Electronic Journal of Statistics*, **17**(2), 2180–2209. doi:10.1214/23-ejs2149.

Saraceno G, Markatou M, Mukhopadhyay R, Golzy M (2024). "Goodness-of-Fit and Clustering of Spherical Data: The **QuadratiK** Package in R and Python." doi:10.48550/arXiv.2402.02290. ArXiv Preprint arXiv:2402.02290.

Tsagris M, Athineou G, Adam C, Yu Z (2024). ***Directional**: A Collection of Functions for Directional Data Analysis*. R package version 7.0, URL https://CRAN.R-project.org/package=Directional.

Zhang X, Zhang Y, Long D, Xie W, Dai Z, Tang J, Lin H, Yang B, Xie P, Huang F, *et al.* (2024). "mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval." doi:10.48550/arXiv.2407.19669. ArXiv Preprint arXiv:2407.19669.

Affiliation:

Lukas Sablica
Institute for Statistics and Mathematics
Vienna University of Economics and Business
Welthandelsplatz 1, 1020 Wien
Telephone: +43 1 31336-5058
E-mail: lukas.sablica@wu.ac.at

Kurt Hornik
Institute for Statistics and Mathematics
Vienna University of Economics and Business
Welthandelsplatz 1, 1020 Wien
Telephone: +43 1 31336-4756
E-mail: kurt.hornik@wu.ac.at

Bettina Grün
Institute for Statistics and Mathematics
Vienna University of Economics and Business
Welthandelsplatz 1, 1020 Wien
Telephone: +43 1 31336-5286
E-mail: bettina.gruen@wu.ac.at

Austrian Journal of Statistics http://www.ajs.or.at/
published by the Austrian Society of Statistics http://www.osg.or.at/
Volume 54 *Submitted:* 2024-10-12
2025 *Accepted:* 2025-02-26