이학석사학위논문

Elliptical k-means Algorithm and Hyperparameter Selection Strategy for Prediction and Clustering on the Torus by Conformal Prediction

토러스 공간 상에서의 적합예측 기반 예측 및 클러스터링을 위한 타원형 k-평균 알고리즘과 초모수 선택 전략

홍승기

2022 년 8 월

Elliptical k-means Algorithm and Hyperparameter Selection Strategy for Prediction and Clustering on the Torus by Conformal Prediction

토러스 공간 상에서의 적합예측 기반 예측 및 클러스터링을 위한 타원형 k-평균 알고리즘과 초모수 선택 전략

지도교수 정 성 규

이 논문을 이학석사 학위논문으로 제출함

2022 년 4 월

서울대학교 대학원

통계학과

홍 승 기

홍승기의 이학석사 학위논문을 인준함

2022 년 7 월

위 원 장	오희석	(인)
부위원장	정성규	(인)
위 원	이권상	(인)

초록

단백질 구조 데이터는 다차원 토러스 상의 각도들로 구성되어 있다. 이러한 특성을 가진 데이터에 대한 연구는 단백질의 기능적 특성을 파악하는 데에 중요한 열쇠가 되어왔다. 그러나 대부분의 통계적 방법론들은 유클리드 공간을 가정하기 때문에 다차원 각도 데이터에 부적합하다. 본 논문에서는 타원형 k-평균 알고리즘을 활용하여 다차원 각도 데이터를 분석하는 법을 소개한다. 특히 본 논문에서는 적합예측집합을 구성하고 혼합 모형 추정을 통한 예측 클러스터링 방법론을 소개한다. 또한 안정성과 계산 효율성을 확보한 새로운 초모수 선택 전략을 제시한다. 마지막으로, 본 논문의 방법론을 구현한 R 패키지 **ClusTorus**를 활용하여 실제 데이터셋에 적용한 예시를 소개한다.

주요어: 토러스 공간, 적합예측, 귀납적 적합예측, 클러스터링, 타원형 k-평균 알고리즘, 초모수 선택.

학번: 2020-25859

목차

제 1 장 Introduction

Multivariate angular or circular data have found applications in some research domains including geology (e.g., paleomagnetic directions) and bioinformatics (e.g., protein dihedral angles). Due to the cyclic nature of angles, usual vector-based statistical methods are not directly applicable to such data. A p-variate angle $\theta = (\theta_1, \cdots, \theta_p)^T$ lies on the p-dimensional torus $\mathbb{T}^p = [0, 2\pi)^p$ in which the angles 0 and 2π are identified as the same point. Likewise, angles θ and $\theta \pm 2\pi$ are the same data point on the torus. Thus, statistical models and predictions on the torus should reflect this geometric constraint.

A prominent example in which multivariate angular data appear is the analysis of protein structures. As described in Branden and Tooze (1999), the functional properties of proteins are determined by the ordered sequences of amino acids and their spatial structures. These structures are determined by several dihedral angles, and thus, protein structures are commonly described on multidimensional tori. The p-dimensional torus $\mathbb{T}^p$ is the sample space we consider in this paper. Especially, for the 2-dimensional case, the backbone chain angles ϕ, ψ of a protein are commonly visualized by the Ramachandran plot, a scatter plot of dihedral angles in a 2-dimensional flattened torus $\mathbb{T}^2$ (Lovell et al., 2003; Oberholser, 2010). In Figure 1.1, several clustering results are visualized on the Ramachandran plot for the protein angles of SARS-CoV-2 virus, which caused the 2020-2021 pandemic (Coronaviridae Study Group of the International Committee on Taxonomy of Viruses. et al., 2020). Since the structures in protein

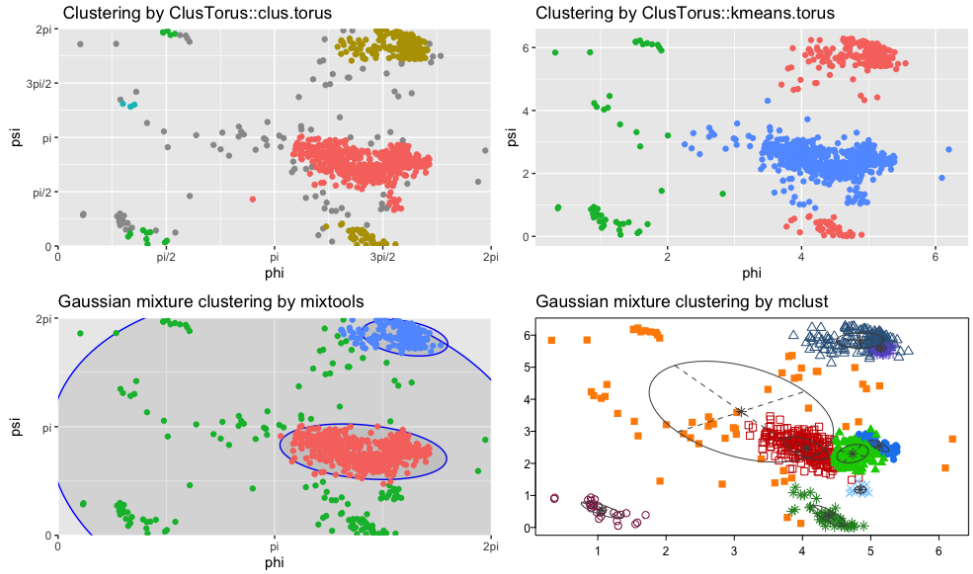

그림 1.1 Several clustering results on Ramachandran plot for SARS-CoV-2 by using `clus.torus` (top left) and `kmeans.torus` (top right), both implemented in **ClusTorus**, `mixtools::mvnormalmixEM` (bottom left), in which the number of components 3 is pre-specified, and `mclust::Mclust` (bottom right), in which the number of components is chosen by BIC. Gray points in the top-left panel are "outliers", automatically assigned by `clus.torus`.

angles are related to functions of the protein, it is of interest to analyze the scatter of the angles through, for example, density estimation and clustering. Note that the protein structure data are routinely collected and publicly available at Protein Data Bank (Berman et al., 2003) and importing such data into R is made easy by the package **bio3d** (Grant et al., 2006, 2021).

For data on the torus, there are a few previous works for mixture modeling and clustering. Mardia et al. (2007) proposed a mixture of bivariate von Mises distributions for data on $\mathbb{T}^2$, with an application to modeling protein

backbone chain angles. Mardia et al. (2012) proposed a density estimation on the torus, based on a mixture of approximated von Mises sine distributions, for higher dimensional cases, but the proposed EM algorithm tends to be unstable when sample sizes are limited. The R package **BAMBI** (Chakraborty and Wong, 2019, 2020) provides routines to fit such von Mises mixture models using MCMC, but is only applicable to bivariate (and univariate) angles in $\mathbb{T}^2$.

Algorithmic clustering for data on the torus has also been proposed. For example, Gao et al. (2018) used an extrinsic k-means algorithm for clustering protein angles. The top right panel of Figure 1.1 depicts the result of applying this algorithm with $k = 3$. Note that the popular R packages **mixtools** (Benaglia et al., 2009) and **mclust** (Scrucca et al., 2016) provide misleading clustering results, when applied to data on the torus. As we illustrate in Figure 1.1, these tools do not take into account the cyclic nature of the angular data.

In this paper, we introduce a novel approach for prediction and clustering multivariate angular data on the torus. The main contribution is extension-and-combination of the predictive clustering approaches of Jung et al. (2021) and Shin et al. (2019). For this, the conformal prediction framework of Vovk et al. (2005) is extended for multivariate angular data. The conformal prediction is a distribution-free method of constructing prediction sets, and we uses mixture models based on the multivariate von Mises distribution (Mardia et al., 2012). Furthermore, by using Gaussian-like approximations of the von Mises distributions and a graph-theoretic approach, flexible clusters, composed of unions of ellipsoids on $\mathbb{T}^p$, can be identified. We will introduce an elliptical k-means algorithm for fitting mixture models and a novel hyperparameter selection strategy which shows dramatically faster and relatively satisfactory clustering results

ompared to existing methods.

The result of the predictive clustering using our method is visualized in the top left panel of Figure 1.1. The dataset `SARS_CoV_2`, included in **ClusTorus**, an R package which is an implementation of our approaches, collects the dihedral angles ϕ, ψ in the backbone chain B of SARS-CoV-2 spike glycoprotein. The raw coronavirus protein data are available at Protein Data Back with id 6VXX (Walls et al., 2020), and can be retrieved by using R package **bio3d**.

The rest of this article focuses on introducing the four core procedures: (i) the conformal prediction framework, including our choices of the conformity scores, (ii) parameter estimation for mixture models using elliptical k-means algorithm, (iii) cluster assignment and (iv) hyperparameter selection.

제 2 장 Conformal prediction

2.1 Conformal prediction framework

The conformal prediction framework (Vovk et al., 2005) is one of the main ingredients of our development. Based on the work of Vovk et al. (2005) and Lei et al. (2013, 2015), we briefly introduce the basic concepts and properties of conformal prediction. Suppose that we observe a sample of size n, $X_i \sim F$ where $X_i \in \mathbb{T}^p$ for each i and that the sequence $\mathbb{X}_n = \{X_1, \cdots, X_n\}$ is *exchangeable*. Then, for a new $X_{n+1} \sim F$, the prediction set $C_n = C_n(\mathbb{X}_n)$ is said to be valid at level $1-\alpha$ if:

$$P(X_{n+1} \in C_n) \geq 1-\alpha, \quad \alpha \in (0,1), \tag{2.1}$$

where P is the corresponding probability measure for $\mathbb{X}_{n+1} = \mathbb{X}_n \cup \{X_{n+1}\}$.

For a given $x \in \mathbb{T}^p$, write $\mathbb{X}_n(x) = \mathbb{X}_n \cup \{x\}$. Consider the null hypothesis $H_0 : X_{n+1} = x$, where $X_{n+1} \sim F$. To test the hypothesis, the conformal prediction framework uses *conformity scores* σ_i defined as follows:

$$\begin{aligned} \sigma_i(x) &:= g(X_i, \mathbb{X}_n(x)), \ \forall i = 1, \cdots, n+1, \\ \sigma(x) &:= g(x, \mathbb{X}_n(x)) = \sigma_{n+1}(x), \end{aligned}$$

for some real valued function g, which measures the conformity or similarity of a point to the given set. If $X_{(1)}, \cdots, X_{(n+1)}$ are ordered to satisfy $\sigma_{(1)} \leq \cdots \leq \sigma_{(n+1)}$ for $\sigma_{(i)} = g\left(X_{(i)}, \mathbb{X}_{n+1}\right)$, then we may say that $X_{(n+1)}$ is the most similar point to $\mathbb{X}_{n+1}$.

Consider the following quantity:

$$\pi(x) = \frac{1}{n+1} \sum_{i=1}^{n+1} I\left(\sigma_i(x) \leq \sigma_{n+1}(x)\right), \quad I(A) = \begin{cases} 1, & A \text{ is true,} \\ 0, & \text{otherwise,} \end{cases}$$

which can be understood as a p-value for the null hypothesis H_0. The *conformal prediction set* of level $1-\alpha$ is constructed as

$$C_n^\alpha = \{x : \pi(x) > \alpha\}. \tag{2.2}$$

Because the sequence $\mathbb{X}_n(x)$ is exchangeable under H_0, $\pi(x)$ is uniformly distributed on $\left\{\frac{1}{n+1}, \cdots, 1\right\}$. With this property, it can be shown that the conformal prediction set is valid for finite samples, i.e., (2.1) holds with C_n replaced by C_n^α for any F, that is, the prediction set is distribution-free (Lei et al., 2013). The performance of the conformal prediction highly depends on the choice of conformity score σ. In some previous works on conformal prediction (Lei et al., 2013, 2015; Shin et al., 2019; Jung et al., 2021), the quality of prediction sets using density based conformity scores has been satisfactory.

2.2 Inductive conformal prediction

If the sample size n and the number N of grid points over $\mathbb{T}^p$ are large, evaluating $n+N$ conformity scores may take a long time. That is, constructing the conformal prediction set suffers from high computational costs. A workaround for this inefficiency is *inductive conformal prediction*, which enjoys significantly lower computational cost. The inductive conformal prediction framework is based on spliting the data into two sets. The algorithm for inductive conformal prediction is given in Algorithm 1.

Algorithm 1 Inductive Conformal Prediction

procedure INDUCTIVE CONFORMAL PREDICTION($\{X_1, \cdots, X_n\}, \alpha, n_1 < n$)

Split the data randomly into $\mathbb{X}_1 = \{X_1, \cdots, X_{n_1}\}$, $\mathbb{X}_2 = \{X_{n_1+1}, \cdots, X_n\}$.

Construct σ with $\sigma(x) = g(x, \mathbb{X}_1)$ for some function g.

Put $\sigma_i = g(X_{n_1+i}, \mathbb{X}_1)$ and order as $\sigma_{(1)} \leq \cdots \leq \sigma_{(n_2)}$, where $n_2 = n - n_1$.

Construct $\hat{C}_n^{\alpha} = \left\{x : \sigma(x) \geq \sigma_{\left(i_{n_2,\alpha}\right)}\right\}$ where $i_{n,\alpha} = \lfloor (n+1)\alpha \rfloor$.

end procedure

While the sizes n_1 and n_2 of two splitted data sets can be of any size, they are typically set as equal sizes. It is well-known that the output $\hat{C}_n^{\alpha}$ of the algorithm also satisfies the distribution-free finite-sample validity (Vovk et al., 2005; Lei et al., 2015). For fast computation, the inductive conformal prediction is primarily used in constructing prediction sets and clustering, in our implementation of **ClusTorus**. As already mentioned, we need to choose the conformity score σ carefully for better clustering performances.

2.3 Conformity scores from mixtures of multivariate von Mises

Our suggestions of conformity scores are based on mixture models. Since the multivariate normal distributions are not defined on $\mathbb{T}^p$, we instead use the multivariate von Mises distribution (Mardia et al., 2008), whose density on $\mathbb{T}^p$ is

$$f(y; \mu, \kappa, \Lambda) = C(\kappa, \Lambda) \exp\left\{-\frac{1}{2}\left[\kappa^T (2 - 2c(y, \mu)) + s(y, \mu)^T \Lambda s(y, \mu)\right]\right\} \tag{2.3}$$

where $y = (y_1, \cdots, y_p)^T \in \mathbb{T}^p$, $\mu = (\mu_1, \cdots, \mu_p)^T \in \mathbb{T}^p$, $\kappa = (\kappa_1, \ldots, \kappa_p)^T \in (0, \infty)^p$, $\Lambda = (\lambda_{j,l})$ for $1 \leq j, l \leq p$, $-\infty < \lambda_{jl} < \infty$,

$$c(y, \mu) = (\cos(y_1 - \mu_1), \cdots, \cos(y_p - \mu_p))^T,$$
$$s(y, \mu) = (\sin(y_1 - \mu_1), \cdots, \sin(y_p - \mu_p))^T,$$
$$(\Lambda)_{jl} = \lambda_{jl} = \lambda_{lj}, \ j \neq l, \quad (\Lambda)_{jj} = \lambda_{jj} = 0,$$

and for some normalizing constant $C(\kappa, \Lambda) > 0$. We write $f(y; \theta) = f(y; \mu, \kappa, \Lambda)$ for $\theta = (\mu, \kappa, \Lambda)$.

For any positive integer J and a mixing probability $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_J)$, consider a J-mixture model:

$$p(u; \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{j=1}^{J} \pi_j f(u; \theta_j) \tag{2.4}$$

where $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_J)$, $\theta_j = (\mu_j, \kappa_j, \Lambda_j)$ for $j = 1, \cdots, J$. Let $\left(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}\right)$ be appropriate estimators of $(\boldsymbol{\pi}, \boldsymbol{\theta})$ based on $\mathbb{X}_1$. The plug-in density estimate based on (2.4) is then

$$p\left(\cdot; \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}\right) = \sum_{j=1}^{J} \hat{\pi}_j f\left(\cdot; \hat{\theta}_j\right), \tag{2.5}$$

which can be used as a conformity score by setting $g(\cdot, \mathbb{X}_1) = \hat{p}(\cdot)$. Assuming high concentrations, an alternative conformity score can be set as $g(\cdot, \mathbb{X}_1) = p^{max}\left(\cdot, \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}\right)$ where

$$p^{max}\left(u; \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}\right) := \max_{j=1,\cdots,J}\left(\hat{\pi}_j f\left(u; \hat{\theta}_j\right)\right) \approx p\left(u; \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}\right). \tag{2.6}$$

On the other hand, Mardia et al. (2012) introduced an approximated density function f^* for the p-variate von Mises sine distribution (2.3) for sufficiently high concentrations and when $\Sigma \succ 0$:

$$f^*(y; , \mu, \Sigma) = (2\pi)^{-p/2} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}\left[\kappa^T (2 - 2c(y, \mu)) + s(y, \mu)^T \Lambda s(y, \mu)\right]\right\}$$

where $\left(\Sigma^{-1}\right)_{jl} = \lambda_{jl}, \left(\Sigma^{-1}\right)_{jj} = \kappa_j,\ j \neq l$. By further approximating via $\theta \approx \sin\theta$, $1 - \frac{\theta^2}{2} \approx \cos\theta$, we write

$$f^*\left(y;,\mu,\Sigma\right) \approx (2\pi)^{-p/2} \left|\Sigma\right|^{-1/2} \exp\left\{-\frac{1}{2}\left[\left(y \ominus \mu\right)^T \Sigma^{-1} \left(y \ominus \mu\right)\right]\right\}, \tag{2.7}$$

where the angular subtraction $\ominus$ stands for

$$X \ominus Y := \left(\arg\left(e^{i(\phi_{x1}-\phi_{y1})}\right), \cdots, \arg\left(e^{i(\phi_{xp}-\phi_{yp})}\right)\right)^T,$$

for $X = \left(\phi_{x1}, \cdots, \phi_{xp}\right)^T \in \mathbb{T}^p$ and $Y = \left(\phi_{y1}, \cdots, \phi_{yp}\right)^T \in \mathbb{T}^p$ as defined in Jung et al. (2021) for $p = 2$. By replacing the von Mises density f in (2.6) with the approximate normal density (2.7), $\log\left(p^{max}\left(\cdot;\boldsymbol{\pi},\boldsymbol{\theta}\right)\right)$ is approximated by

$$\begin{aligned}\log\left(p^{max}\left(u;\boldsymbol{\pi},\boldsymbol{\theta}\right)\right) &\approx \frac{1}{2}\max_j e\left(u;\pi_j,\theta_j\right) + c,\\ e\left(u;\pi_j,\theta_j\right) &= -\left(u \ominus \mu_j\right)^T \Sigma_j^{-1}\left(u \ominus \mu_j\right) + 2\log\pi_j - \log\left|\Sigma_j\right|\end{aligned} \tag{2.8}$$

where $\theta_j = \left(\mu_j, \Sigma_j\right)$, $\mu_j = \left(\mu_{1j}, \cdots, \mu_{pj}\right)^T \in \mathbb{T}^p$, $\Sigma_j \in \mathbb{R}^{p\times p}$ and a constant $c \in \mathbb{R}$. Our last choice of the conformity score is

$$g\left(\cdot, \mathbb{X}_1\right) = \max_j e\left(\cdot, \hat{\pi}_j, \hat{\theta}_j\right). \tag{2.9}$$

Note that with this choice of conformity score, the conformal prediction set can be expressed as the union of ellipsoids on the torus. That is, the following equalities are satisfied (Shin et al., 2019; Jung et al., 2021): Let C_n^e be the level $1-\alpha$ prediction set using (2.9). Then

$$\begin{aligned}C_n^e &:= \left\{x \in \mathbb{T}^p : g\left(x, \mathbb{X}_1\right) \geq g\left(X_{\left(i_{n_2,\alpha}\right)}, \mathbb{X}_1\right)\right\}\\ &= \bigcup_{j=1}^{J} \hat{E}_j\left(\sigma_{\left(i_{n_2,\alpha}\right)}\right)\end{aligned} \tag{2.10}$$

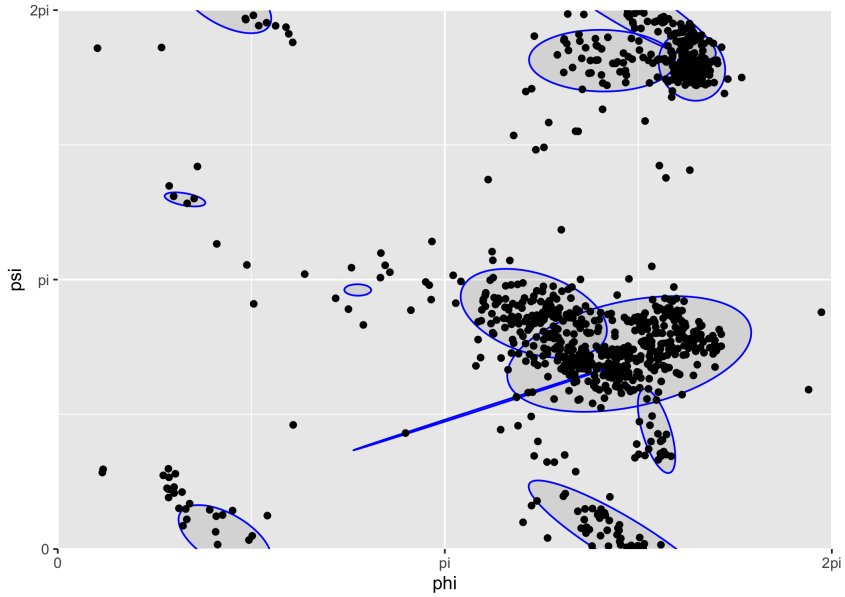

그림 2.1 The Ramachandran plot for SARS-CoV-2, with conformal prediction set whose conformity score is (2.9) with $J = 12$ for level $\alpha = 0.1111$. The plot demonstrates the union of ellipses as (2.10).

where $\hat{E}_j(t) = \left\{ x \in \mathbb{T}^p : (x \ominus \hat{\mu}_j)^T \hat{\Sigma}_j^{-1} (x \ominus \hat{\mu}_j) \leq 2 \log \hat{\pi}_j - \log \left|\hat{\Sigma}_j\right| - t \right\}$ for $t \in \mathbb{R}$. Note that $\hat{E}_j(t)$ is automatically vanished if $t \geq 2 \log \hat{\pi}_j - \log \left|\hat{\Sigma}_j\right|$. Figure 2.1 demonstrates that the shape of conformal prediction set is actually a union of ellipsoids as (2.10), when using (2.9) as the conformity score.

Conformity scores based on mixture model and its variants need appropriate estimators of the parameters, $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$. If the parameters are poorly estimated, the conformal prediction sets will be constructed trivially and thus become useless. There can be two methods of estimation: EM algorithms and the elliptical k-means algorithm, also known as the generalized Lloyd's algorithm (Sung and Poggio, 1998; Bishop, 2006; Shin et al., 2019). EM algorithms for the mixture model (2.5) are described in Jung et al. (2021), for the 2-dimensional case. Since

the EM estimates require long computation time and large sample sizes, extensions to higher-dimensional tori do not seem to apt. The EM estimates of the mixture model parameters can be naturally used for the case of max-mixture (2.6) and ellipsoids (2.9) as well. On the other hand, the elliptical k-means algorithm converges much faster even for moderately high-dimensional tori. The elliptical k-means algorithm is used for estimating parameters in the approximated normal density (2.7), and for computation of the conformity score of ellipsoids (2.9). The elliptical k-means algorithms for data on the torus are further discussed in the next section.

제 3 장 Parameter estimation for multivariate von Mises

3.1 Elliptical k-means algorithm

In this section, we outline the elliptical k-means algorithm for the data on the torus. The algorithm is used to estimate the parameters of the mixture model (2.4), approximated as in (2.7). Note that the EM algorithm can be used for parameter estimation for mixture models in low dimensions. For $p > 3$, EM algorithms suffer from high computational costs (Mardia et al., 2012). To circumvent this problem, we estimate the parameters by modifying the generalized Lloyd's algorithm (Shin et al., 2019), also known as the elliptical k-means algorithm (Sung and Poggio, 1998; Bishop, 2006). For vector-valued data, Shin et al. (2019) showed that the elliptical k-means algorithm estimates the parameters sufficiently well for the max-mixture density case as (2.6).

Suppose $y_1, \cdots, y_n \in \mathbb{T}^p$ are an independent and identically distributed sample. Using the approximated density (2.7), the approximated likelihood, L', is

$$L'(\mu, \Sigma) = (2\pi)^{-np/2} |\Sigma|^{-n/2} \exp\left[-\frac{n}{2} tr\left(S\Sigma^{-1}\right)\right] \tag{3.1}$$

where $S = \frac{1}{n}\sum_{i=1}^{n} (y_i \ominus \mu)(y_i \ominus \mu)^T$. Thus, if μ is known, $\hat{\Sigma} = S$ maximizes L'. Following Mardia et al. (2012), the mean μ is estimated as follows. Let $\bar{U}_j = \sum_{i=1}^{n} \cos(y_{ij})/n$ and $\bar{V}_j = \sum_{i=1}^{n} \sin(y_{ij})/n$ for $j = 1, \cdots, p$. Then, $\hat{\mu} =$

$(\hat{\mu}_1, \cdots, \hat{\mu}_p)^T$,

$$\hat{\mu}_j = \arctan \frac{\bar{V}_j}{\bar{U}_j}, \quad j = 1, \cdots, p \tag{3.2}$$

which is the maximum likelihood estimator of mean direction of von Mises-Fisher distribution (Mardia and Jupp, 1999).

With these approximated maximum likelihood estimators, the elliptical k-means algorithm, described in Algorithm 2, maximizes the likelihood corresponding to the max-mixture model (2.6).

Algorithm 2 Elliptical k-means algorithm for the torus

1: **procedure** ELLIPTICAL K-MEANS($\{X_1, \cdots, X_n\}, J$)

2: Initialize $\pi_j, \theta_j = (\mu_j, \Sigma_j)$, $j = 1, \cdots, J$

3: set

$$w_{i,j} = \begin{cases} 1, & \text{if } j = \arg\max_l \left[-\left(X_i \ominus \mu_l\right)^T \Sigma_l^{-1} \left(X_i \ominus \mu_l\right) - \log|\Sigma_l| + 2\log \pi_l \right] \\ 0, & \text{otherwise} \end{cases}$$

$$I_j = \{i \in \{1, \cdots, n\} \,|\, w_{i,j} = 1\}$$

4: Update μ_j as (3.2) with $\{X_i\}_{i \in I_j}$ for $j = 1, \cdots, J$

5: Update $\Sigma_j = \frac{1}{\sum_{i=1}^n w_{i,j}} \sum_{i=1}^n w_{i,j} \left(X_i \ominus \mu_j\right) \left(X_i \ominus \mu_j\right)^T$ for $j = 1, \cdots, J$

6: Update $\pi_j = \frac{1}{n} \sum_{i=1}^n w_{i,j}$ for $j = 1, \cdots, J$

7: Repeat step 3-6 until converge

8: **end procedure**

Note that the initial values require an initial clustering. For this, one may use other clustering algorithms such as the extrinsic k-means or the hierarchical clustering algorithms.

3.2 Constraints for mixture models

The protein structure data we aim to analyze typically consist of hundreds of angles (observations). Fitting the mixture with a large number of components may give inefficient estimators. Thus, one can consider following three options for reducing the number of model parameters, by constraining the shape of the ellipsoids, or the covariance matrices. Applying the constraints lead much faster convergence for estimating parameters (Grim, 2017). We list three types of constraints for covariance matrices Σ_j.

- $\Sigma_j = \sigma_j^2 I_p$ for some $\sigma_j^2 > 0$ for all j, and the prediction set will be the union of spheres. Furthermore, if $\sigma_1^2 = \cdots = \sigma_J^2$ and $\pi_j = 1/J$ for all j, then all the spheres have the same radii.
- $\Sigma_j = diag\left(\sigma_{jk}^2\right)_{k=1,\cdots,p}$ for $\sigma_{jk}^2 > 0$, and the fitted ellipsoids $\hat{E}_j$ $(j = 1, \cdots, J)$ are the axis-aligned ellipsoids.
- No constraint for Σ_j, and $\hat{E}_j$ $(j = 1, \cdots, J)$ are any ellipsoids.

제 4 장 Clustering by conformal prediction

We now describe our clustering strategies using the conformal prediction sets. Suppose for now that the level α and the hyperparameter J of the prediction set are given. The basic idea of clustering is to take each connected component of the prediction set as a cluster. For this, we need an algorithm identifying connected components from any prediction set. Since the prediction sets are in general of irregular shapes, such an identification is a quite difficult task. However, as shown in Jung et al. (2021), if the conformal prediction set is of the form (2.10), clusters are identified by testing the intersection of ellipsoids. Suppose $C_n^e = \cup_{j=1}^J \hat{E}_j$ where each $\hat{E}_j$ is an ellipsoid. Let the (i,j)th entry of a square matrix A be 0 if $\hat{E}_i \cap \hat{E}_j = \varnothing$, 1 otherwise. Then, A is the adjacent matrix of a graph whose nodes and edges represent the ellipsoids and intersections, respectively. The adjacent matrix A gives a partition $I_1, \cdots, I_K \subseteq \{1, \cdots, J\}$ satisfying

$$\hat{E}_{i_k} \cap \hat{E}_{i_{k'}} = \varnothing, \quad k \neq k'$$

where $1 \leq k, k' \leq K, i_k \in I_k, i_{k'} \in I_{k'}$. This implies that the union of ellipsoids, $U_k = \cup_{i \in I_k} \hat{E}_i$, whose indices are in a connected component I_k for some k, can be regarded as a cluster. That is, $U_1, \cdots, U_K$ are the disjoint clusters. With this, the conformal prediction set naturally generates K clusters. Note that testing the intersection of ellipsoids can be done efficiently (which is a univariate root finding problem (Gilitschenski and Hanebeck, 2012)), while testing the intersection of arbitrarily shaped sets is not feasible in general. This is the

reason why we only use the conformity score of the form (2.9), the prediction set from which is exactly the union of ellipsoids.

We now describe how the cluster labels are assigned to data points. Each data point included in the prediction set is automatically assigned to the cluster which contains the point. For the data points which are not included in the conformal prediction set, we have implemented two different types for cluster assignment, as defined in Jung et al. (2021). The first is to assign the *closest* cluster label. The notion of closest cluster can be defined either by the Mahalanobis distance $(x \ominus \hat{\mu}_j)^T \hat{\Sigma}_j^{-1}(x \ominus \hat{\mu}_j)$, the approximate log-density (2.8), or the largest posterior probability $\hat{P}(Y = k|X = x)$. For example, for $x \notin C_n^e$, let E_i be the set with the largest approximate log-density $\hat{e}_i(x)$. If $i \in I_k$, then x is assigned to the cluster k. These provide three choices of cluster assignment, depending on the definition of "closeness." The last choice is to regard the excluded points as outliers. That is, if $x \notin C_n^e$, then the point x is labeled as "outlier." This outlier-disposing clustering may be more appropriate for the cases where some of data points are truely outliers. Figure 4.1 compares the two different types of clustering assignment.

그림 4.1 The Ramachandran plot for SARS-CoV-2, with clustering generated by conformal prediction set whose conformity score is (2.9) with $J = 12$ for $\alpha = 0.1111$. Left panel shows the cluster assignment based on approximate log-density, and the right panel shows the outlier disposing clustering assignment.

제 5 장 Hyperparameter selection

Poor choices of conformity score result in too wide prediction sets. Thus, we need to choose the hyperparameters elaborately for a better conformal prediction set and for a better clustering performance. The hyperparameters are the number of mixture components J and the level α. There have been some efforts to select the optimal hyperparameters by introducing adequate criteria. Lei et al. (2013) and Jung et al. (2021) each proposed criteria based on the minimum volume of the conformal prediction set. However, as we shall see, these approaches become computationally infeasible for higher dimensions.

We briefly review the criterion used in Jung et al. (2021). Assume for now that mixture models are used; that is, (J, α) are the hyperparameters of interest. For a set $C \subseteq \mathbb{T}^p$, let $\mu(C)$ be the volume of C. Without loss of generality, we can assume that $\mu\left(\mathbb{T}^p\right) = 1$. For a given level α, the optimal choice of hyperparameter J minimizes $\mu\left(C_n(\alpha, J)\right)$ of conformal prediction set $C_n\left(\alpha, J\right)$. To choose α and J altogether, Jung et al. (2021) proposed to use the following criterion:

$$\left(\hat{\alpha}, \hat{J}\right) = \arg\min_{\alpha, J} \alpha + \mu\left(C_n\left(\alpha, J\right)\right). \tag{5.1}$$

In computing the criterion (5.1), the volume $\mu\left(C_n\left(\alpha, J\right)\right)$ is numerically approximated. This is feasible for data on $\mathbb{T}^2 = [0, 2\pi)^2$ by inspecting the inclusion of each point of a fine grid. However, for high dimensional cases, for example $\mathbb{T}^4$, evaluating the volume becomes computationally infeasible. In fact, as the dimension increases, the number of required inspections grows exponentially.

Furthermore, the function $(\alpha, J) \to \alpha + \mu\left(C_n(\alpha, J)\right)$ is typically not a convex function and has multiple local minima. Thus, the choice of $\left(\hat{\alpha}, \hat{J}\right)$ by (5.1) tends to be unstable, resulting in high variability of the clustering results. Therefore, evaluating (5.1) is not practical for high-dimensional data.

To this end, we have developed a computationally more efficient procedure for hyperparameter selection, which also provides more stable clustering results. This procedure is a two-step procedure, first choosing the model parameter J, then choosing the level α. Our approach is in contrast to the approaches in Lei et al. (2013) and Shin et al. (2019) in which they only choose the model parameter for a prespecified level α.

The first step of the procedure is to choose J, without making any reference to the level α. Choosing J can be regarded as selecting an appropriate mixture model. The model selection is based on either the (prediction) risk, Akaike information criterion (Akaike, 1974), or Bayesian information criterion (Schwarz, 1978). Since the mixture model-based conformity scores (2.5), (2.6) and (2.9) are actually the density or the approximated log-density of the mixture model, we use the conformity scores in place of the likelihood. For example, the sum of the conformity scores (2.9) over the given data is exactly the fitted log-likelihood. Specifically, let $\mathbb{X}_1, \mathbb{X}_2$ be the splitted datasets given by Algorithm 1 and $\mathbb{X} = \mathbb{X}_1 \cup \mathbb{X}_2$. Let $\sigma(\cdot) = \log g\left(\cdot; \mathbb{X}_1\right)$ if g is given by (2.5) and (2.6) or $\sigma(\cdot) = g\left(\cdot; \mathbb{X}_1\right)$ if g is given by (2.9). Recall that g is the conformity score, and it depends on the estimated model $\hat{p}$. Then, the function σ we defined above

also depends on the model $\hat{p}$, and the criterion R can be defined as follows:

$$R(\mathbb{X}, \hat{p}) = \begin{cases} -2\sum_{x \in \mathbb{X}_2} \sigma(x) & \text{if the criterion is the risk,} \\ -2\sum_{x \in \mathbb{X}_1} \sigma(x) + 2k & \text{if the criterion is AIC,} \\ -2\sum_{x \in \mathbb{X}_1} \sigma(x) + k \log n_1 & \text{if the criterion is BIC,} \end{cases}$$

where k is the number of model parameters and n_1 is the cardinality of $\mathbb{X}_1$. This procedure is summarized in Algorithm 3.

Algorithm 3 hyperparam.J

procedure HYPERPARAM.J($\mathbb{X} \subset \mathbb{T}^p$, fitted models $\hat{p}_{j_1}, \cdots, \hat{p}_{j_n}$, criterion R)
  Evaluate $R_j = R(\mathbb{X}, \hat{p}_j)$ for $j = j_1, \cdots, j_n$.
  Evaluate $\hat{J} = \arg\min_{j \in \{j_1, \cdots, j_n\}} R_j$.
  Output $\hat{J}, \hat{p}_{\hat{J}}$.
end procedure

The second step is to choose the level $\alpha \in (0, 1)$ for the chosen $\hat{J}$ and $\hat{p}_{\hat{J}}$, so that the clustering result is stable over perturbations of α. If the number of clusters does not change by varying the level $\alpha \in I$ for some interval I, we regard that the clustering result is stable on I. If I is sufficiently wide, it is reasonable to choose an $\alpha \in I$. Thus, our strategy is to find the most wide interval $I = [a, b] \subseteq (0, 1)$ whose elements construct the same number of clusters, and to set $\hat{\alpha}$ as the midpoint of the interval, i.e. $\hat{\alpha} = (a + b)/2$. However, choosing α large, e.g. $\alpha > 0.5$, results in a too small coverage $1 - \alpha$ of the prediction set. Thus, we restrict the searching area as $[0, M]$ for $M \in (0, 1)$ which is close to 0, and find the desirable I in the restricted area $[0, M]$ rather than the whole interval $[0, 1]$. This strategy is described in Algorithm 4.

Note that we could alternatively input an array of levels, if there is a pre-specified searching area. In our experience, setting $M = 0.15$ gives generally

Algorithm 4 hyperparam.alpha

procedure HYPERPARAM.ALPHA(fitted model $\hat{p}$, $n_2 := |\mathbb{X}_2|$, $M \in [0, 1]$)

Evaluate the number of clusters c_{α_j} for $\alpha_j = j/n_2$, $j = 1, \cdots, \lfloor n_2 M \rfloor$.

Set $A = \{j : c_{\alpha_{j-1}} \neq c_{\alpha_j}, \quad j = 2, \cdots, \lfloor n_2 M \rfloor\}$.

For $A = \{\alpha_{j_1}, \cdots, \alpha_{j_N}\}$ find $i = \arg\max_{k \in \{1, \cdots, N-1\}} \alpha_{j_{k+1}} - \alpha_{j_k}$.

Output $\hat{\alpha} = \left(\alpha_{j_{i+1}} + \alpha_{j_i}\right)/2$

end procedure

satisfying results. By setting $M = 0.15$, at most 15% of the data points are not included in the prediction set, and at most 15% of the data can be regarded as the outliers. We may interpret this level selecting procedure as finding the representative modes for the given mixture model; the chosen level is the cutoff value for which the most stable modes are not vanished.

In summary, we first choose the number of model components J in view of model selection, and then find the most stable level $\hat{\alpha}$ in the sense of invariability of the number of clusters. In the next section, the two-step procedures for hyperparameter selection are used in a cluster analysis of data on $\mathbb{T}^4$.

제 6 장 Clustering data on $\mathbb{T}^4$

In this section, we give an example of clustering `ILE` data in $\mathbb{T}^4$. `ILE` is a dataset included in **ClusTorus**, which represents the structure of the isoleucine. This dataset is obtained by collecting several different `'.pdb'` files in the Protein Data Bank (Berman et al., 2003). We used PISCES (Wang and Dunbrack, 2003) to select high-quality protein data, by using several benchmarks—resolution is 1.6Å or better, R-factor is 0.22 or better, sequence percentage identity is equal to or less than 25—as described in Harder et al. (2010) and Mardia et al. (2012). The `ILE` data consist of $n = 8080$ instances of four angles $(\phi, \psi, \chi_1, \chi_2) \in \mathbb{T}^4$, and is displayed in Figure 6.1.

For predictive clustering of `ILE` data, the conformal prediction sets and scores are built from mixture models, fitted with the elliptical k-means algorithm. The number J of components in the mixture model needs to be tuned, and we set the candidates for J as $\{10, \ldots, 40\}$.

Next step is to select the hyperparameter J, and the level α of the prediction set. We use the two-step procedure, discussed in the previous section, but apply all three available criteria ("risk", "AIC", and "BIC") in choosing $\hat{J}$.

The details of hyperparameter selection can be visualized, and are shown in Figure 6.2. The first row of the figure shows that the evaluated prediction risk is the smallest at $\hat{J} = 29$. On the right panel, it can be seen that the longest streak of the number of clusters over varying level α occurs at 16, which is given by a range of levels around $\hat{\alpha} = 0.1093$. The second and third rows are similarly

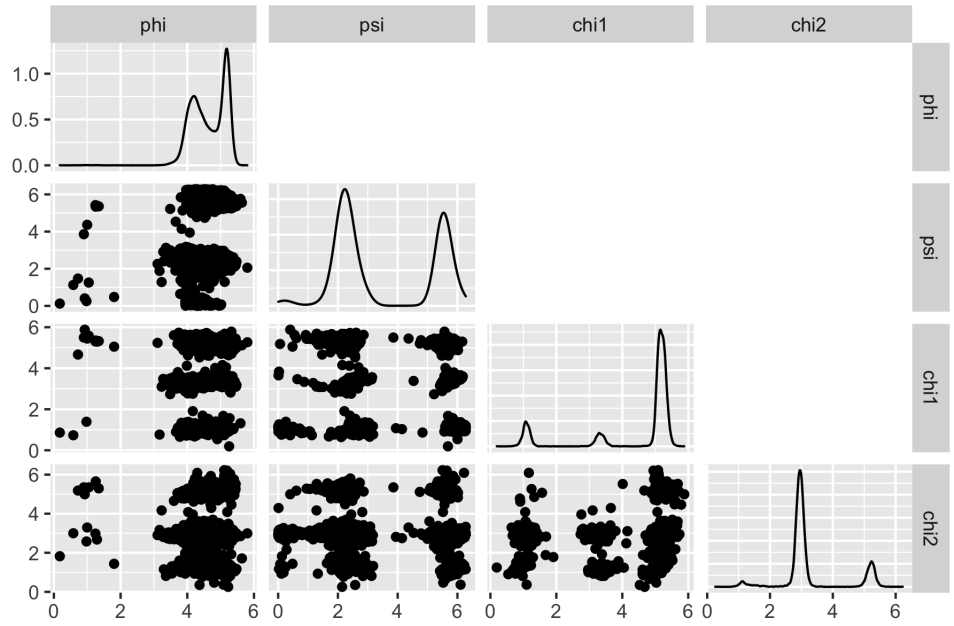

그림 6.1 The pairwise scatter plots of `ILE` data.

generated, and they show the results of AIC- and BIC-based hyperparameter selection. While the results of hyperparameter selection from the three criteria do not always agree with each other, we observe that using BIC tends to choose parsimonious models than others, for this and many other data sets we tested.

The number of clusters, given by the conformal prediction set $C_n(\hat{\alpha}, \hat{J})$, can be seen in the right panels of Figure 6.2. For example, in the top right panel, with $\hat{J} = 29$ and $\hat{\alpha} = 0.1093$, the number of clusters is 16 (the vertical position of the blue-colored longest streak). For the subsequent analysis, we use the risk criterion, thus choosing $(\hat{J}, \hat{\alpha}) = (29, 0.1093)$.

Finally, using the cluster assignment method described in previous section, the assigned cluster memberships can be displayed on the pairwise scatter plots of the four angles. We demonstrate the outlier-disposing membership assignment, as well as the membership assignment based on the maximum of log-densities. Figure 6.3 displays the clustering result with scatter plots.

Since the conformal prediction set is a union of 4-dimensional toroidal ellipsoids, projections of such ellipsoids onto coordinate planes are shown in Figure 6.4.

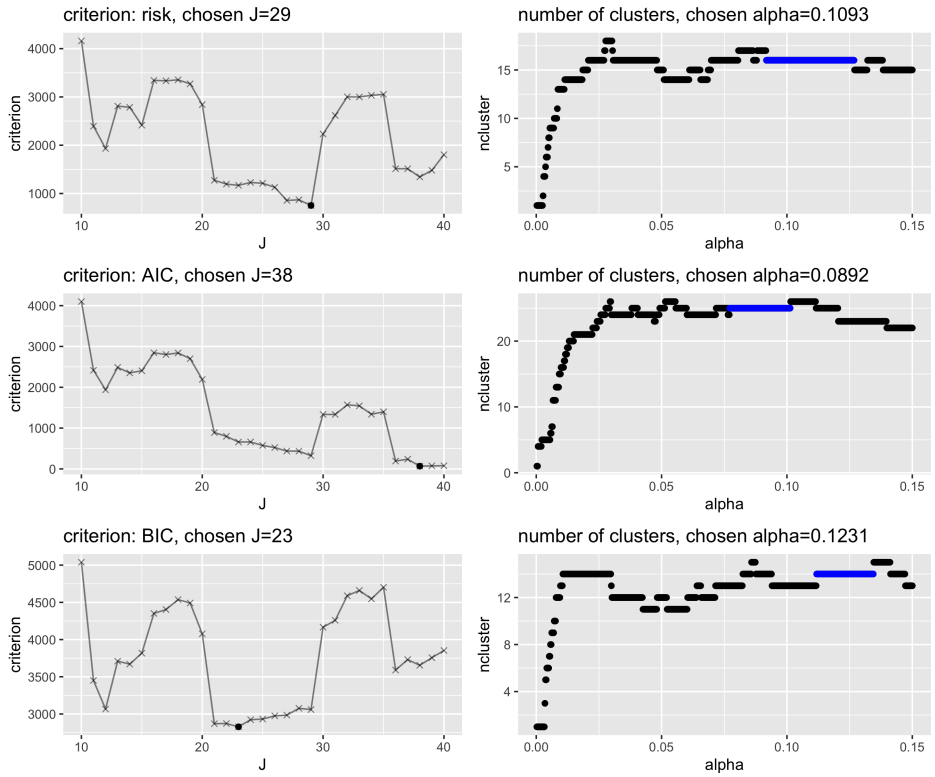

그림 6.2 Hyperparameter selection for `ILE` data. Rows correspond to different choices of criteria "risk", "AIC" and "BIC". In each row, the left panel shows the values of criterion over J, with the optimal $\hat{J}$ indicated by a thicker dot; the right panel shows the number of clusters over varying α, in which the longest streak is highlighted. The optimal $\hat{\alpha}$ is the midpoint of the longest streak.

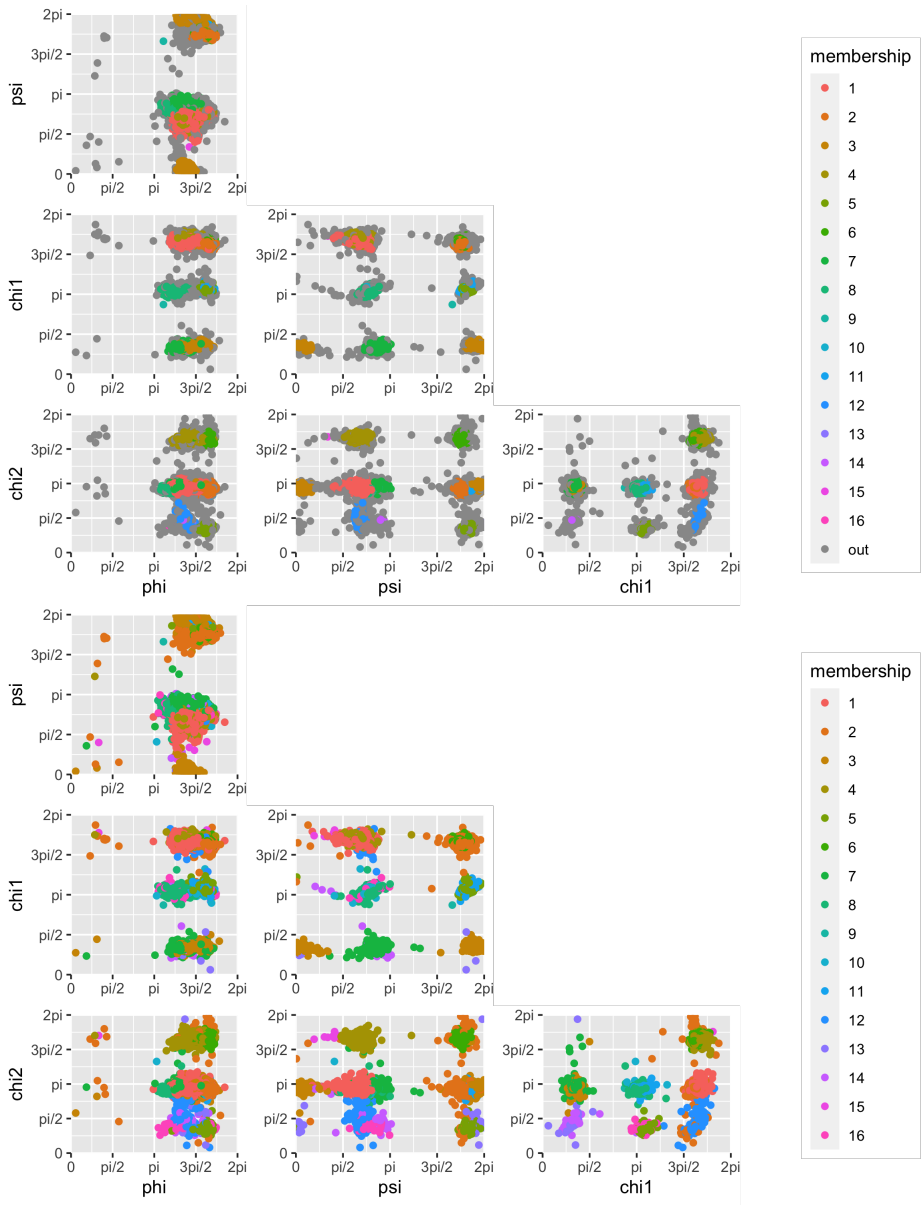

그림 6.3 The pairwise scatter plots of `ILE` data with cluster assignments. (Top) "outlier". (Bottom) "log.density".

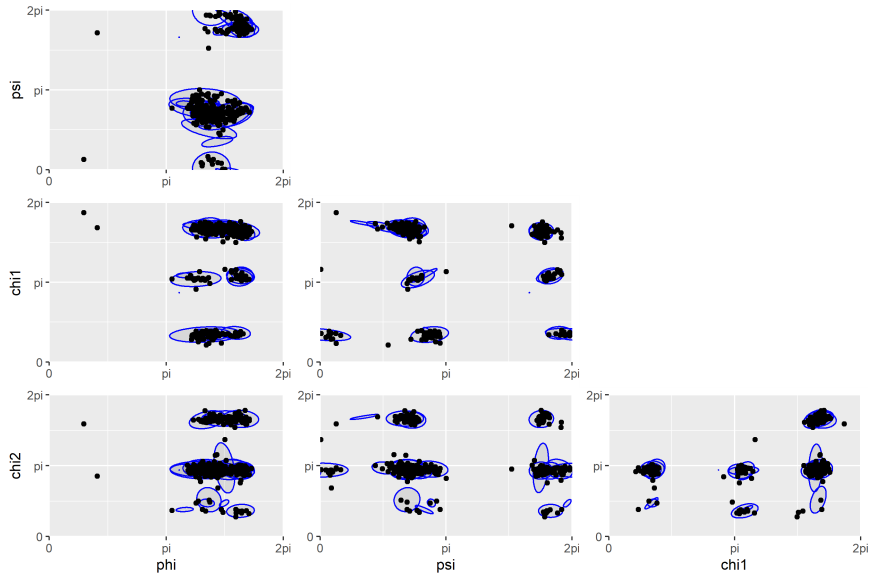

그림 6.4 The pairwise scatter plots of `ILE` data, overlaid with the (projected) ellipsoids that constitute the conformal prediction set $C_n(\hat{\alpha}, \hat{J})$.

제 7 장 Summary and discussion

In this paper, we introduced an approach for prediction and clustering on the torus by conformal prediction framework. We used multivariate von Mises mixture models as a choice of conformity scores, and suggested elliptical k-means algorithm for the mixture models which is feasible for high dimensional cases. We also introduced the two-step hyperparameter selection strategy, which is computationally efficient compared to extisting methods, and demonstrated our implementation with data on $\mathbb{T}^4$. The clustering method based on graph-theoretical approach can result in cluster assignment either with or without an outlier class. The package **MoEClust** (Murphy and Murphy, 2020, 2021) can also dispose some points as outliers. However, **MoEClust** only works on Euclidean space, not on $\mathbb{T}^p$.

There are some possible future developments. First, EM algorithms for von Mises mixture models on high dimensional tori (e.g., $\mathbb{T}^4$) can be implemented assuming independence of angles in each component. Using closed-form approximations of maximum likelihood estimators for univariate von Mises-Fisher distributions (Banerjee et al., 2005; Hornik and Bettina, 2014), fitting mixtures of product components can be done efficiently (Grim, 2017). Another direction is obtained by viewing clustering based on (2.10) by varying α as surveying birth and death of connected components. This can be dealt with a persistence diagram, a concept of topological data analysis. Hence, instead of using Algorithm 4, one may choose desirable α using persistence diagram.

참고문헌

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**(6):716–723, 1974.

A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von mises- fisher distributions. *Journal of Machine Learning Research*, **6**(46):1345–1382, 2005.

T. Benaglia, D. Chauveau, D. R. Hunter, and D. Young. mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, **32**(6):1–29, 2009.

H. Berman, K. Henrick, and H. Nakamura. Announcing the worldwide protein data bank. *Nature Structural and Molecular Biology*, **10**:980, 2003.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, Inc., 33 Spring Street, New York, NY 10013, USA, 2006.

C. Branden and J. Tooze. *Introduction to Protein Structure*. Garland Publishing, Inc, 19 Union Square West, New York, 1999.

S. Chakraborty and S. W. Wong. *BAMBI: Bivariate Angular Mixture Models*, 2020. R package version 2.3.0.

S. Chakraborty and S. W. K. Wong. *BAMBI: An R package for Fitting Bivariate Angular Mixture Models*, 2019.

Coronaviridae Study Group of the International Committee on Taxonomy of Viruses., A. e. Gorbalenya, S. C. baker, R. S. baric, R. J. de Groot, C. Drosten, A. A. Gulyaeva, bart l. Haagmans, C. lauber, A. M. leontovich, benjamin W. Neuman, D. Penzar, S. Perlman, leo l. M. Poon11, D. V. Samborskiy, I. A. Sidorov, I. Sola, and J. Ziebuhr. The species severe acute respiratory syndrome- related coronavirus: classifying 2019-ncov and naming it sars-cov-2. *Nature Microbiology*, **5**:536—-544, 2020.

Y. Gao, S. Wang, and M. Deng. Raptorx-angle: real-value prediction of protein backbone dihedral angles through a hybrid method of clustering and deep learning. *BMC Biometrics*, **19**(100), 2018.

I. Gilitschenski and U. D. Hanebeck. A robust computational test for overlap of two arbitrary- dimensional ellipsoids in fault-detection of kalman filters. *In 2012 15th International Conference on Information Fusion*, pages 396–401, 2012.

B. Grant, X.-Q. Yao, L. Skjaerven, and J. Ide. *bio3d: Biological Structure Analysis*, 2021. R package version 2.4-2.

B. J. Grant, A. P. C. Rodrigues, K. M. ElSawy, J. A. McCammon, and L. S. D. Caves. Bio3d: an R package for the comparative analysis of protein structures. *Bioinformatics*, **22**(21):2695–2696, 08 2006. ISSN 1367-4803.

J. Grim. Approximation of unknown multivariate probability distributions by using mixtures of product components: A tutorial. *International Journal of Pattern Recognition and Artificial Intelligence*, **31**(09):1750028, 2017.

T. Harder, W. Boomsma, M. Paluszewski, J. Frellsen, K. E. Johansson, and T.

Hamelryck. Beyond rotamers: a generative, probabilistic model of side chains in proteins. *BMC Bioinformatics*, **11**:306, 2010.

K. Hornik and G. Bettina. On maximum likelihood estimation of the concentration parameter of von mises–fisher distributions. *Computational Statistics*, **29**:945—-957, 2014.

S. Jung and S. Hong. *ClusTorus: Prediction and Clustering on the Torus by Conformal Prediction*, 2021. R package version 0.2.1.

S. Jung, K. Park, and B. Kim. Clustering on the torus by conformal prediction. *Annals of Applied Statistics*, **15**(4):1583–1603, 2021.

J. Lei, J. Robins, and L. Wasserman. Distribution-free prediction sets. *Journal of the American Statistical Association*, **108**(501):278–287, 2013.

J. Lei, A. Rinaldo, and L. Wasserman. A conformal prediction approach to explore functional data. *Ann Math Artif Intell*, **74**:29–43, 2015.

S. C. Lovell, I. W. Davis, W. B. Arendall III, P. I. De Bakker, J. M. Word, M. G. Prisant, J. S. Richardson, and D. C. Richardson. Structure validation by c$\boldsymbol{\alpha}$ geometry: ϕ, ψ and $c\beta$ deviation. *Proteins: Structure, Function, and Bioinformatics*, **50**(3):437–450, 2003.

K. V. Mardia and P. E. Jupp. *Directional Statistics*. Wiley, New York, 1999.

K. V. Mardia, C. C. Taylor, and G. K. Subramaniam. Protein bioinformatics and mixtures of bivariate von mises distributions for angular data. *Biometrics*, **63**(2):505—-512, 2007.

K. V. Mardia, G. Hughes, C. C. Taylor, and H. Singh. A multivariate von mises distribution with applications to bioinformatics. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, **36**(1):99–109, 2008.

K. V. Mardia, J. T. Kent, Z. Zhang, and C. C. T. . T. Hamelryck. Mixtures of concentrated multivariate sine distributions with applications to bioinformatics. *Journal of Applied Statistics*, **39**(11):2475–2492, 2012.

M. D. Marzio, A. Panzera, and C. C. Taylor. Kernel density estimation on the torus. *Journal of Statistical Planning and Inference*, **141**(6):2156–2173, 2011. ISSN 0378-3758. doi: https://doi.org/10.1016/j.jspi. 2011.01.002.

K. Murphy and T. B. Murphy. Gaussian parsimonious clustering models with covariates and a noise component. *Advances in Data Analysis and Classification*, **14**(2):293–325, 2020.

K. Murphy and T. B. Murphy. *MoEClust: Gaussian Parsimonious Clustering Models with Covariates and a Noise Component*, 2021. R package version 1.4.2.

K. Oberholser. Proteopedia entry: Ramachandran plots. *Biochemistry and Molecular Biology Education*, **38**(6):430–430, 2010.

G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, **6**(2):461 – 464, 1978.

L. Scrucca, M. Fop, T. B. Murphy, and A. E. Raftery. mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, **8**(1):289–317, 2016.

J. Shin, A. Rinaldo, and L. Wasserman. Predictive clustering, 2019.

K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(1):39–51, 1998.

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer Science+Business Media, Inc., 33 Spring Street, New York, NY 10013, USA, 2005.

A. C. Walls, Y.-J. Park, M. A. Tortorici, A. Wall, A. T. McGuire, and D. Veesler. Structure, function, and antigenicity of the sars-cov-2 spike glycoprotein. *Cell*, **181**, 2020.

G. Wang and J. Dunbrack, Roland L. PISCES: a protein sequence culling server. *Bioinformatics*, **19** (12):1589–1591, 08 2003. ISSN 1367-4803.

Abstract

Protein structure data consist of several dihedral angles, lying on a multi-dimensional torus. Analyzing such data has been and continues to be key in understanding functional properties of proteins. However, most of the existing statistical methods assume that data are on Euclidean spaces, and thus they are improper to deal with angular data. In this paper, we introduce a novel approach specialized to analyzing multivariate angular data, based on elliptical k-means algorithm. Our approach enables the construction of conformal prediction sets and predictive clustering based on mixture model estimates. Moreover, we also introduce a novel hyperparameter selection strategy for predictive clustering, with improved stability and computational efficiency. We demonstrate our achievements with the package **ClusTorus**, one of our implementations, in clustering protein dihedral angles from two real data sets.

Keywords: Toroidal space, conformal prediction, inductive conformal prediction, clustering, elliptical k-means algorithm, hyperparameter selection.

Student Number: 2020-25859