ERASMUS UNIVERSITY ROTTERDAM

ERASMUS SCHOOL OF ECONOMICS

Clustering Methods of Multivariate Extremes:

A model comparison analysis with empirical case studies

Student(ID)

Jingyi CHENG (499636)

Supervisor

Dr. Phyllis WAN

Second Assessor

Dr. Alex KONING

July 4, 2021

Abstract

Finding the lower-dimensional representations of data is valuable in high-dimensional multivariate extremes analysis. Clustering approaches with unsupervised learning features have drawn attentions recently. In this paper, I discuss and comparatively analyze the spherical k-means clustering by Janßen and Wan (2020) and spherical k-principal-components clustering by Fomichov and Ivanovs (2020). And evidence has been found that both approaches give similar results in both simulation experiments and applications to empirical cases. Nevertheless, this paper also suggests spherical k-means clustering to be computationally lighter solution for complex data with high dimensions and potentially large number of clusters.

Contents

1 Introduction

Extreme value analysis is a statistics study that is valuable in providing accurate estimates of the risks of rare events (Davison and Huser, 2015). It looks into heavy-tailed observations, where classic statistical tools often fail to perform. There are various applications in the field of natural disasters, such as flooding and earthquakes, as well as in the field of finance, where abnormal losses are caused during financial crises. Although the theoretical framework and statistical tools for univariate data are well-developed, real-life cases often contain high-dimensional features, especially for some complex networks like stock markets and spatial systems like river discharges (Asadi et al., 2018; Poon et al., 2003). Hence, multivariate extreme value theory (MEVT) is an essential topic of extreme value analysis. In MEVT, the extremal dependence structure among variables of a multivariate system is modeled to evaluate the overall risk level of a system and to hint at the underlying sparse extremal patterns of a high-dimensional network (Engelke and Ivanovs, 2020).

Therefore, the main challenge of MEVT is to reduce the complexity of multivariate structure by dimension reduction (Coles et al., 1999; Ledford and Tawn, 1997). The mainstream methods can be roughly summarized into three types.

- The first type of methods utilize conditional independence structure and extremal graphical models to decompose a high-dimensional system into several causal networks, leading to the detection of the probabilistic sparse patterns (Engelke and Ivanovs, 2020). And it enables causality research by introducing parsimonious parametric models (Engelke and Hitz, 2020; Engelke and Volgushev, 2020; Gnecco et al., 2019).

- The second direction focuses on concomitant extremes. The concept of concomitant extreme refers to the case where some subsets of variables presumably obtain large values simultaneously. The detection of such subsets can be challenging, as it typically leads to a sparse model when the size of concomitant subsets is small. Various approaches have been introduced to find the group of faces and their corresponding mass via angular probability measures (see Chiapino and Sabourin, 2016; Chiapino et al., 2019; Goix et al., 2017; Meyer and Wintenberger, 2019).

- With the successful application of unsupervised based dependency structure detection in high-dimensional data like language modeling (Gao and Suzuki, 2003) and feature selec-

tions (Ding et al., 2017), some unsupervised learning techniques have been introduced to the field of MEVT. Hence, the last class of approaches, which is also the focus of this paper, is clustering and principal component analysis (PCA) methods adapted from unsupervised learning for dimension deduction. In the clustering approach, similar observations are grouped into k different clusters by minimizing the dissimilarity criteria. Janßen and Wan (2020) adapt the spherical k-means algorithm from Dhillon and Modha (2001), and they propose the acquired cluster centers to be spectral prototypes for extremes. While centroid-based angular clustering desires to represent each group of extremal observations by its direction proxy, PCA linearly projects the multivariate observations into a series of eigenvectors. Obtained from an iterative procedure, those eigenvectors together span a lower-dimensional proxy of the original data (see Cooley and Thibaud, 2019; Drees and Sabourin, 2019). Given the visualization function of angular approximations, these two nonparametric approaches are convenient to apply in dependency exploratory empirical study (Engelke and Ivanovs, 2020).

Moreover, it is also possible to combine PCA and clustering into a hybrid model. Chautru et al. (2015) proposes a two-step approach by firstly extracting spherical principal component nested spheres, which are then clustered by a spherical k-means approach for further dimension reduction. And the approach introduced by Fomichov and Ivanovs (2020) combines the explanatory capability of the first principal eigenvectors and the optimal searching capability of k-classification.

In this study, I focus on the analysis of the spherical k-means (k-means) and spherical k-principal-component (k-PC) clustering methods in dependence structure learning. Although it has been shown by Fomichov and Ivanovs (2020) that the spherical k-PC approach outperforms k-means in the task of concomitant extremes identification, the performance and efficiency of applying both methods with data of different settings have not been fully discussed yet. Hence, it is interesting to see how results differentiate by applying these two iterative-clustering methods with entirely different updating criteria. With this study, numerical experiments with the max-linear model as well as empirical studies are performed for such exploration. Derived from the horse-racing numerical experiments, evidence has been found that both methods provide similar results. While k-PC might have slightly better performance, it also takes longer to run. Also, they share common difficulties in handling high dimensional and/or high number of cluster data. Nevertheless, the applications on empirical cases further indicate their similarity in results.

The rest of this paper is organized as follows: At Section 2, an overview of the relevant MEVT background knowledge is presented. Section 3 introduces the methodology for k-means and k-PC clustering. Then, Section 4 presents the simulation experiments of this study. And in Section 5, empirical cases of stock returns and dietary data are analyzed with the proposed clustering methods and procedure. Lastly, this paper is wrapped up in Section 6 with conclusions over the findings from simulation and empirical studies, followed by future research suggestions.

2 Preliminary: Multivariate Extremes

This section services as a recap for the basic knowledge of multivariate extreme value theory, which involves the fundamental assumptions and essential data handling procedures. Surveys by Davison and Huser (2015) and Engelke and Ivanovs (2020) provide detailed descriptions, and are suggested as further reading material.

First, this paper defines the input data as a random vector $X \in \mathbb{R}^d$ with a series of independent and identically distributed (i.i.d) copies of observations $(X_1^i, X_2^i, \ldots, X_d^i), i \in \mathbb{N}$, where each observation contains $d \geq 2$ dimensions. As justified in univariate extreme theory, there are mainly two kinds of extreme selection approaches: block maxima method and the peaks-over-threshold method. For the former method, we assume that there exists some constants α_j^n and β_j^n, $j \in \{1, 2, \ldots, d\}, n \in \{1, 2, \ldots, \mathbb{N}\}$, for the following convergence of each component X_j

$$\frac{max_{i=1,\ldots,n} X_j^i - \beta_j^n}{\alpha_j^n} \quad \Rightarrow \quad z, \qquad n \to \infty, \tag{1}$$

where the distribution of z is a type of max-stable generalized extreme value (GEV) distribution. Hence, we can assume the following *max-domain of attraction of the extreme value distribution* $G(x_1, \ldots, x_d)$ of convergence g as the convergent distribution of X:

$$(\frac{max_{i=1,\ldots,n} X_1^i - \beta_1^n}{\alpha_1^n} \leq x_1, \ldots, \frac{max_{i=1,\ldots,n} X_d^i - \beta_d^n}{\alpha_d^n} \leq x_d) \quad \Rightarrow \quad g, \qquad n \to \infty. \tag{2}$$

Given the convergence to GEV distribution, there are mainly two steps to analyze the heavy-tailed behavior of the random vector X. First of all, we can obtain the marginal distribution function G_j of X, which has been well-studied in the topic of univariate extremes. I refer readers to Davison and Huser (2015) for an elaborate summary of its mathematical properties. In this study, I focus on the other step, which is looking for the extremal dependence structure. To continue, we first

need to standardize the input vector X, such that all the components of X are scaled to the same magnitude and the definition for extreme remains across components. Assuming it follows standard Pareto distribution: $\mathbb{P}(X_j) = 1 - \frac{1}{x}$, for $x \geq 1$, the standardized vector Y is defined as,

$$Y = (\frac{1}{1 - F_1(X_1)}, \ldots, \frac{1}{1 - F_d(X_d)}), \tag{3}$$

where F_j is the continuous marginal distribution of the component $X_j, j \in 1, 2, \ldots, d$. With the assumption of standard Pareto distribution, the marginal distribution of transformed vector Y follows standard Fréchet distribution denoted by $G_0(x_1, \ldots, x_d)$. The convergence in (2) can be specified as:

$$(\frac{max_{i=1,\ldots,n} Y_1^i}{n} \leq x_1, \ldots, \frac{max_{i=1,\ldots,n} Y_d^i}{n} \leq x_d) \quad \Rightarrow \quad g_0, \qquad n \to \infty, \tag{4}$$

for a series of constants $(x_1, \ldots, x_d) \in [0, \infty)^d$. And $G_0(x_1, \ldots, x_d)$, which is the distribution of g_0, can be expressed as a function of the *exponent measure* λ

$$G_0(x_1, \ldots, x_d) = exp\{-\lambda(\mu_1, \ldots, \mu_d)\}, \quad \exists j : \mu_j > x_j, \tag{5}$$

where λ is defined on space $\varepsilon \in [0, \infty]^d \setminus \{0\}$, spanned by $(\mu_1, \ldots, \mu_d)$. See Chapter 2.3 in Engelke and Ivanovs (2020), the regularity property of exponent measure λ over Y, also referred as multivariate regular variation, enables us to describe the angular patterns of extremal values as a probability expression. Given the convergence in (2) and (4), we can define the spectral (angular) distribution $S(B)$ of the transformed vector Y by

$$\lim_{\mu \to \infty} \mathbb{P}(\frac{Y}{\|Y\|} \in B \mid \|Y\| > \mu) = S(B), \tag{6}$$

where B is a continuous Borel set on the probability measure S, μ is some arbitrarily large thresholds for extreme selection. And S is on the simplex $\mathbb{S}_+^{d-1} := \{\|x\| = 1, x \in \mathbb{R}_+^d\}$. Additionally, the angular measure S is informative in describing the direction of an extremal observation or a group of selected extremes. Prototypes can be estimated by those direction subsets of S with high-frequency of repentance as explained by Buchta et al. (2012), where such an idea is applied in text mining. Note that $\|\cdot\|$ can be any type of norm as proven by Janßen and Wan (2020), but it has to be consistent through out the calculation procedure. For the convenience of applying

k-means and k-PC clustering later on, we adopt Euclidean norm here. In this study, I focus on the identification of extremal dependence structure with unsupervised clustering methods. Hence, my component of interest is the spectral distribution data $S(B)$.

The last concept to cover in this section is asymptotic dependence. For any two random variables x and y, their pairwise asymptotic dependence is evaluated by the tail dependence coefficient χ defined as:

$$\chi = \lim_{\mu \to 1} P(\, x > F_x^{-1}(\mu) \,|\, y > F_y^{-1}(\mu) \,), \quad \chi \in [0, 1]. \tag{7}$$

When $\chi = 0$, we say the vectors x and y are asymptotically independent. Or x and y asymptotically dependent to each other when $\chi \neq 0$. In MEVT, the asymptotic dependency of components may reveal informative patterns.

3 Methodology: Spherical Clustering Methods

In this paper, I introduce two recent clustering techniques, spherical k-means clustering (k-means) and spherical k-principle components clustering (k-PC), in the topic of dependency structure learning of multivariate extreme theory.

Clustering is a classic type of unsupervised machine learning technique, which is able to detect the intrinsic structure with unlabelled data. The main task of clustering is to detect the centroids of observations and assign the nearby observations to clusters, such that the estimated centroids represent as prototypes of the subsets of observations they connect with. The algorithm attempts to search for the centroids by minimizing the cost when it is based on dissimilarity function (such as distance used by k-means) and maximizing the gains when it is based on similarity criteria on the desired properties (such as explanatory power used by k-PC). Clustering is promising in lower dimension detection of extremal dependency structure mainly in two scenarios, summarized by Engelke and Ivanovs (2020): The first scenario is when the spectral distribution $S(B)$ converges to some points in $\mathbb{S}_+^{d-1}$, thus the support of $S(B)$ has a lower dimension than d; The second scenario is when the mass of all centroids mostly charged on small groups of variables, such that the groups of variables concomitantly extreme.

3.1 *Spherical K-means Clustering*

Firstly formulated by MacQueen et al. (1967), the k-means clustering is a classic unsupervised learning method which minimizes distance-to-centers dissimilarity to group the observations. Inspired by this procedure, Janßen and Wan (2020) introduces an extremal analysis procedure utilizing the *spherical k-means* procedure by Dhillon and Modha (2001) to cluster the extremal directions $\Theta = \{\theta_1, \ldots, \theta_n\}$, which has a corresponding distribution $S(B)$ defined at (6).

Let $d(\cdot) : \mathbb{R}^d \times \mathbb{R}^d$ be the continuous angular dissimilarity measure of two spectral directions, which is the so-called "distance" function. Following the definition in Chautru et al. (2015) and Janßen and Wan (2020), $d(\cdot)$ is calculated as the angular difference between two vectors

$$d(x, y) = d_\theta(x, y) := 1 - cos(x, y) = 1 - \frac{x^T y}{\|x\|_2 \cdot \|y\|_2}, \tag{8}$$

where x and y are two directional vectors of dimension d. Let us assume that the current set of cluster centers as $A = \{a_1, a_2, \ldots, a_k\}$, $a_c \in \mathbb{R}^d$ for cluster $c \in \{1, 2, \ldots, k\}$, k is the number of clusters. Following the transformation functions (3) and (6), the d-dimensional angular representation $\theta_i, i \in \{1, \ldots, n\}$ of the corresponding observation X^i is restricted on the Euclidean unit sphere. Hence, the dissimilarity (8) can be simplified as $d(\theta_i, a_c) = 1 - \theta_i^T a_c$, which is the spherical distance of the transformed i^{th} observation to a centroid on A. Since both θ_i and a_c are unit vectors with identical orientation, the dissimilarity is equivalent to their point difference.

With the dissimilarity function defined, a general dissimilarity function $W(A, P)$ proposed by Janßen and Wan (2020) can be computed as

$$W(A, P) := \int_{\theta \in \Theta} \min_{a \in A} d(\theta, a) P(d\theta) \tag{9}$$

, where P is the probability measure for the vector θ on the corresponding continuous Borel set $B(\Theta)$. $W(A, P)$ is the probability-weighted distance from an arbitrary observation X^i to the nearest center in A, ranging from 0 to ∞. By arranging an equal mass $\frac{1}{n}$ to each observation in X, $W(A, P)$ can be simplified as

$$W(A_k^n) := \frac{\sum_{i=1}^n \min_{a_c \in A} d(\theta_i, a_c)}{n}, \tag{10}$$

which measures the averaged distance from observations X^i to its nearest cluster center on A.

At each iteration, the k-means algorithm firstly updates the clustering conditions for each ob-

servation and calculates the current dissimilarity by (9) or (10) based on the latest set of estimated centroids A. Then, the average spectral vectors are selected within each cluster to be its new centroid. The pseudocode for a single iteration of k-means is shown in Appendix II.

Just like many kinds of k-means clustering problems, the spherical k-means can also be NP-hard, and the computational cost to optima relies on the initial centroids (Kleinberg et al., 1998). For the numerical experiments in Section 4 and empirical analysis in Section 5, I apply the **R**-package **skmeans** by Buchta et al. (2012), where stable outcomes are produced. Another advantage for using this k-means procedure is that the run time complexity of an optimal single object move is reduced to $\mathcal{O}(n+k)$ ($\mathcal{O}(n_c+1)$ for finding the centers of each cluster, and $n = \sum_{c=1}^{k} n_c$), when object-prototype dissimilarities are provided and fixed observation amount n and cluster number k (Buchta et al., 2012).

3.2 *Spherical K-principal-components Clustering*

Taking advantages of clustering approach in handling high dimensional extremal data, Fomichov and Ivanovs (2020) propose the *Spherical K-principal-components Clustering* method. It differs from k-means in two aspects: firstly, k-PC estimates cluster centers by employing the first principal component operations instead of taking the averages; the second difference is that, the k-PC algorithm uses an alternative evaluation function. Theoretical inference and a numerical simulation test has been provided by Fomichov and Ivanovs (2020) to prove k-PC's superiority in handling concomitant extremes, while its performance in general extremal structure learning hasn't been fully examined.

In this section, I introduce the procedure of k-principal-components clustering by Fomichov and Ivanovs (2020). Let us assume again that we have angular inputs $\Theta = \{\theta_1, \ldots, \theta_n\}, \theta_i \in \mathbb{S}_+^{d-1}$ obtained from n i.i.d observations $X^1, \ldots, X^n$. Also, A contains the current set of cluster centers $a_1, \ldots, a_k \in \mathbb{S}_+^{d-1}$. Similar to the k-means procedure, the initial set of A is composed by k random vectors of dimension d. Take the example of a single iteration, the beginning step is to update the dot-product matrix M by,

$$\underset{n\times k}{M} = \Theta^T A = (\theta_1, \ldots, \theta_n)^T (a_1, \ldots, a_k). \tag{11}$$

Given the row $M_i, i \in \{1, \ldots, n\}$ of matrix M corresponds to its sample angular θ_i, each entry of $M_{i,c}$ of M_i represents the level of similarity of θ_i to any of the current cluster centers a_c. Thus,

k-PC utilizes this row-wise similarity and defines its evaluator v as the average row-maxima of M:

$$v = \frac{1}{n}\sum_{i=1}^{n} \underset{1\leq c\leq k}{max}(M_{i,c}), \tag{12}$$

where the mass is distributed evenly to each observation. Meanwhile, a matrix $\underset{k\times n}{\Gamma}$ is created to store the location of the (first) maxima of each observation. That is, for rows $\gamma_1, \ldots, \gamma_k$ of Γ, if the binary indicator $\gamma_{c,i} = 1$, then the observation i is assigned to the cluster c, otherwise $\gamma_{c,i} = 0$. Each observation should only be assigned to exactly one cluster:

$$\sum_{c=1}^{k} \gamma_{c,i} = 1 \qquad i = 1, \ldots, n. \tag{13}$$

Thus, the new clusters are generated and the indicator matrix Γ contains the clustering information of all samples at the current iteration. The following step is to find the new cluster centers. In the k-PC algorithm, the cluster center is taken as the first principle eigenvector of the matrix composed by all angular items of this cluster. Hence, for each new cluster c:

$$\Sigma_c = \frac{1}{n}\sum_{i=1}^{n}(\theta_i^T \theta_i \gamma_{c,i}), \qquad c = 1, \ldots, k. \tag{14}$$

Assume that the biggest eigenvalue of Σ_c is $\lambda_{1,c}$, then its first principal eigenvector a_c^* is calculated as

$$\Sigma_c a_c^* = \lambda_{1,c} a_c^*, \tag{15}$$

which explains most spectral variation of each cluster. Hence, the new centroid $\hat{a}_c$ of cluster c is the a_c^{*T}.

Note that, the k-PC algorithm differs from traditional principal-components analysis. In the k-PC method, only the information of the first principal-components are adopted as prototypes, which requires finding k individual first principal eigenvectors of clustered subsets at each iteration. While the other principal-components approaches on MEVT use a series of eigenvectors of the same matrix for dimension reduction (Cooley and Thibaud, 2019; Drees and Sabourin, 2019).

The pseudocode of the k-PC algorithm is shown at Appendix III, which I realize with the demo code provided by Fomichov and Ivanovs (2020) [1]. The running time complexity of the k-PC algorithm is $\mathcal{O}(k(nd + f(d))$, with $f(d)$ as the complexity for finding the first principal

[1] k-principal-component demo code: https://sites.google.com/site/jevgenijsivanovs/files

eigenvector of a d-by-d matrix. Fomichov and Ivanovs (2020) suggests that $f(d) \approx d^2 log(d)$, when the angular distance ϵ is strictly larger than 0.

4 Numerical Experiments

4.1 *Research Question Formulation*

With the methodology of k-means and k-PC algorithm introduced in Section 3, it is obvious that those two methods have very different iteration frameworks. The k-means algorithm tries to cluster observations by their Euclidean distance from the centroids, whearas the centroids are updated as the averaged spectral vectors of each cluster at each iteration. The k-PC algorithm uses a similarity score for clustering, and it updates the list of centroids as the first principal eigenvector of the sub-matrix composed by each cluster. In dependency structure learning, we are interested in the clustering of observations. Since the defintions of centroids are different between those two methods, it is surely possible to obtain different estimated centroids even provided with the identical groups of observations. However, clustering the observations is the main focus to explore dependency structure. Although finding the exact centroids is less important, it enables us to measure and directly compare the deviation from estimated results to actual data. So we come to the first and the most fundamental research question: *" Do k-means and k-PC methods produce similar clustering results, giving that they are quite different in the mathematical nature? If not, which method is better?"*

Aside of the final results, we are also interested in the computation efficacy of these methods. Mentioned in Section 3, the running time complexity of k-PC ($\mathcal{O}(k(nd + f(d))$) is much higher than that of k-means ($\mathcal{O}(n + k)$), especially when k and d are large. Hence, we would like to explore via a simulation test: *"Whether the actual computational costs for the same number of iterations are inline with the suggested theoretical complexity of the k-means and k-PC methods?"*

In empirical studies, the correct number of clusters k is often unknown beforehand. And we prefer models that are able to perform and provide useful information, even when k is not preset accurately. Therefore, it is also beneficial to construct tests to evaluate the robustness of approaches by providing over- and under-estimated k. A misspecification test for $d = 4$ is present in Janßen and Wan (2020), which suggests that the k-means approach being the most robust solution among the other approaches discussed. It is interesting to explore the robustness of k-PC in misspecified conditions and extend the experiments of Janßen and Wan (2020) to a higher

dimensional scenario. Thus, it guides us to the last topic of interest: *"How do the performance of k-PC and k-means approaches compare in the cases where the number of clusters k is misspecified?"*

4.2 *Experiment Design*

Monte Carlo Simulation: Max-linear Model

Following the Monte Carlo simulation designed by Janßen and Wan (2020), the simulation samples are produced by the popular max-linear model, which provides full information about the real cluster centers. Although the max-linear model itself is too simplistic to explain the real world data, it enables simple construction of simulation data, where the performance is highly measurable compared to other substitutions. We define a coefficient matrix $\underset{k\times d}{B}$ as our model specification, $B = [b_1, b_2, \ldots, b_k]^T$. Each row represents the real cluster of the simulation sample. Let $b_c \in \mathbb{R}_+^d, c = 1, \ldots, k$. The set of simulation samples X is determined by a max-linear model as:

$$\underset{n\times d}{X} = (X_1, \ldots, X_d); \quad where \quad X_j = \max_{c=1,\ldots,k} b_{c,j} \times \Lambda_c, \quad j = 1, \ldots, d, \tag{16}$$

where $\Lambda_1, \ldots, \Lambda_k \in \mathbb{R}^n$ are the i.i.d random vectors following standard Fréchet-distribution. Moreover, we assume that each column of coefficient matrix B sums up to 1, $\sum_{c=1}^k b_{c,j} = 1, \forall j = 1, \ldots d$. Thus, standard Fréchet distribution also apply to the margins of X.

Note that, the Λ_c vectors determine the magnitudes and the coefficient vectors b_c determine the directions of the observations. Janßen and Wan (2020) points out that, the true probability for points of support $a_c = b_c / \|b_c\|$ for the angular measure S, are proportional to the sum of norms of all coefficient vectors: $p_c = \|b_c\| / (\sum_{\psi=1}^k \|b_\psi\|)$.

Having specified the original simulation data, we can then transform X into its angular presentations and select the extremal samples following (3) and (6) from Section 2 (The pseudocode for transformation is shown in Appendix I). In this simulation study, the top 10% observations with largest norm are selected as extremes. For a reliable comparison, the max-linear simulation generates 100 different model specifications (i.e. coefficient matrices B) given the number of clusters k and dimensions d, where each randomly specified model contains $n = 1000$ observations. The random vectors b_c used by this study are created by uniform randomizer $U_i \in [0, 1], i \in \mathbb{N}$, according to the specifications (see Appendix IV) for each pair of (d, k) setting.

Measures

Following the design of Janßen and Wan (2020), the distance between angular measure S and the estimator $\hat{S}$ is checked by two types of evaluators, $d_s(S, \hat{S})$ and $W_s(S, \hat{S})$. $d_s(S, \hat{S})$ only considers the distances from estimated cluster centers to their corresponding real centers:

$$d_s(S, \hat{S}) := \min_{\Pi} \sqrt{\sum_{c=1}^{k} \left\| \hat{a}_{\Pi(c)} - a_c \right\|_2^2}, \tag{17}$$

where Π is the permutation of $\{1, \ldots, k\}$ such that cluster center estimations are matched to the nearest real centers. The other evaluator is a type of Wasserstein distance (Vallender, 1974) with probability measures taken into account,

$$W_s(S, \hat{S}) := \inf_{\mathbb{P} \in \varrho(S, \hat{S})} \int_{\mathbb{S}_+^{d-1} \times \mathbb{S}_+^{d-1}} \left\| x_i - x_j \right\|_2 \mathbb{P}(dx_i, dx_j), \tag{18}$$

where the set of possible probability measures on $\mathbb{S}_+^{d-1} \times \mathbb{S}_+^{d-1}$ spanned by $(S, \hat{S})$ is summarized as $\varrho(S, \hat{S})$. The calculations of $W_s(S, \hat{S})$ are done by the **R**-package **transport** (Schuhmacher et al., 2017). And the complete simulation codes are adapted from Janßen and Wan (2020).

A side note for result interpretations using distance measurements $d_s(S, \hat{S})$ and Wasserstein distance $W_s(S, \hat{S})$ is that, the evaluation might be of advantage to the k-means approach and slightly underestimate the k-PC performance in comparisons, as the k-means algorithm adopts distance-based criterion to estimate centroids. But generally the difference between correctly estimated k-means and k-PC centroids is mild.

Experiment Set-up

In this study, I propose two types of simulation experiments, which are inspired by the simulations by Janßen and Wan (2020) and improved upon. The first type of experiment aims to compare the performance of k-means and k-PC approaches when the number of cluster is correctly specified in the clustering. And the other type of experiment help to interpret the robustness by letting k be too large or too small compared to the actual values. Before going through the detailed set-ups, the numerical experiments from the original papers of the two clustering methods are summarized. In Janßen and Wan (2020), simulations are constructed for amount of cluster k up to 6 and dimension d up to 10. And in Fomichov and Ivanovs (2020), a series of analyses

with different estimators are performed for simulation data with dimension d up to 100, whereas the discussion is limited to $k = 2$. Additionally, the k-misspecified tests have only been done for k-means approach in Janßen and Wan (2020). Nevertheless, information about the running time hasn't been provided for any of the clustering approaches.

Thus, in order to answer the research questions from previous section 4.1 and extend previous studies, I include three improvements to my experiments. First of all, variety is gained at both correctly-specified and misspecified tests by extending simulation data with high/low dimensions and high/low number of actual clusters. Secondly, the running time of each clustering is recorded for computational cost indication. Lastly, the k-PC approach is applied to data with k higher than 2 and evaluated with distance measures, which completes its formal numerical analysis.

Moreover, since both k-means and k-PC generally do not improve much after 100 iterations with our simulation setting, so the maximum runs are set to be 100 for both algorithms.

4.3 *Simulation Experiment Results*

In the following, I will answer the previous research questions using the outcomes of correctly-specified and misspecified simulation experiments (see table 1 and table 2).

Table 1: Model estimation performance for different value of d and k

	spherical k-means			spherical k-PC		
	$d_s(S, \hat{S})$	$W_s(S, \hat{S})$	*Time(Sec.)*	$d_s(S, \hat{S})$	$W_s(S, \hat{S})$	*Time(Sec.)*
$d = 4, k = 2$	0.0486(0.0282)	0.0447(0.0293)	0.0757(0.0186)	0.0485(0.0281)	0.0447(0.0293)	0.4328(0.0493)
$d = 4, k = 6$	0.4074(0.2643)	0.1337(0.0316)	0.1990(0.0342)	0.3676(0.2474)	0.1315(0.0316)	1.7612(0.2050)
$d = 20, k = 2$	0.0611(0.0209)	0.0529(0.0255)	0.1365(0.0270)	0.0607(0.0209)	0.0528(0.0255)	1.2393(0.1418)
$d = 20, k = 10$	0.6650(0.3792)	0.2018(0.0312)	0.2035(0.0257)	0.6970(0.4067)	0.2009(0.0307)	6.6008(0.7933)

* The recorded distances $d_s(S, \hat{S})$ and $W_s(S, \hat{S})$, as well as the program running time, are the mean values over 100 simulations, with the corresponding standard deviations in the brackets.

- *" Do k-means and k-PC methods produce similar clustering results, giving that they are quite different in the mathematical nature? If not, which method is better?"*

Judged from the results of the correctly-specified max-linear simulations (Table 1), k-means and k-PC methods have very similarly good performance. Values are close for both mean distance measures and their standard deviations. Additionally, the performance of k-PC may be superior to that of k-means for most of the cases, only by a little bit according to the distance measurements $d_s(S, \hat{S})$ and $W_s(S, \hat{S})$. Even though the $d_s(S, \hat{S})$ of k-means method is averagely smaller than that

of k-PC by around 0.032 for the sample $d = 20, k = 10$, the $W_s(S, \hat{S})$ of k-PC is mildly smaller than k-means'. As mentioned previously, the distance measures can slightly deviate even when the clustering outcomes are the same, since the two approaches adopts different interpretations for cluster centers. The other observation is that the estimations from both methods deviate more when the k is large, and they are less affected when d is large.

- *"Whether the actual computational costs for the same number of iterations are inline with the suggested theoretical complexity of the k-means and k-PC methods?"*

Table 1 lists the mean computation time of both approaches. Apparently, the computational cost of k-means is pretty much inline with its theoretical complexity $\mathcal{O}(n + k)$. The average running time for 100 iterations of k-means is the highest at $k = 10$, which also decreases at the number of clusters decreases. Note that the running time of $d = 20, k = 2$ is almost the twice as that of $d = 4, k = 2$ for k-means, meaning higher dimensions potentially adds substantial complexity to the calculation. If we compare k-means and k-PC, the latter approach obviously is more costly to run as expected beforehand, especially when d and k are large simultaneously (e.g, $d = 20, k = 10$ in the correctly-specified experiment and $d = 20, k^{\uparrow} = 8$ in the misspecified experiment). It indicates that k-PC approach can be expensive in the applications of complex extremes data with high dimensions and potentially large amount of clusters, in which case k-means seems like a more economic solution to start with.

Table 2: Comparison of $W_s(S, \hat{S})$ when model cluster parameter k is misspecified

True: d = 4, k = 6	spherical k-means	spherical k-PC
Misspecified $k^{\downarrow}$ = 4	0.2037(0.0442)	0.1997(0.0414)
Time(Sec.)	0.1394(0.0221)	1.1359(0.1279)
Misspecified $k^{\uparrow}$ = 8	0.1320(0.0255)	0.1316(0.0256)
Time(Sec.)	0.2468(0.0327)	2.2210(0.2640)
True: d = 20, k = 3	spherical k-means	spherical k-PC
Misspecified $k^{\downarrow}$ = 2	0.2487(0.0343)	0.2459(0.0340)
Time(Sec.)	0.1313(0.0158)	1.1352(0.1499)
Misspecified $k^{\uparrow}$ = 4	0.0988(0.0291)	0.0989(0.0287)
Time(Sec.)	0.2174(0.0212)	2.6846(0.4862)
True: d = 20, k = 5	spherical k-means	spherical k-PC
Misspecified $k^{\downarrow}$ = 2	0.4691(0.0449)	0.4486(0.0505)
Time(Sec.)	0.1779(0.0550)	2.0701(1.2155)
Misspecified $k^{\uparrow}$ = 8	0.1471(0.0292)	0.1478(0.0286)
Time(Sec.)	0.4183(0.1628)	9.6673(5.1878)

* The recorded distances $W_s(S, \hat{S})$ and the running time are the mean values over 100 simulations, with the corresponding standard deviations in the brackets.

- *"How do the performance of k-PC and k-means approaches compare in the cases where the number of clusters k is misspecified?"*

There are a couple of phenomena we can observe from the series of misspecification experiments. Firstly, we find that the performance of k-means and k-PC are again highly close to each other in all three simulations. Thus, these two methods are comparable also from the perspective of robustness in case of misspecified k. Secondly, the estimation deviations from clustering with under-specified k are commonly larger than applying over-specified k. Because having extra clusters means splitting some actual clusters into smaller subsets, where the probability for the samples to be clustered to the right face may not be impacted significantly. However having too few clusters is worse, because the genuine information is thrown away to pursue the wrong clustering. It suggests that researchers should choose an over-estimated k rather than an under-estimated k, if accuracy is prioritized.

5 Empirical Case Studies

In this section, we replicate the empirical analysis of *Financial Portfolio Losses* and *Dietary Intakes* done by Janßen and Wan (2020) using the discussed k-means and k-PC approaches. In those data, their estimated centroids by k-means are typically found to be vectors with only few big positive components, while the other entries are 0 or close to 0. It thus hints asymptotic independence and dependence among certain subset of variables within each cluster. I wonder if the estimated cluster centers by k-PC also suggest similar dependence outcomes, given that they are shown to have close accuracy from Section 4. Hence, to extend the comparison to complex empirical data, we reproduce the analysis with both of k-means and k-PC for the exact datasets.

5.1 *Financial Portfolio Losses*

Financial risk and return is a classic field for extreme theory application. In this case, I use the 30 industry portfolios containing value-averaged daily returns from each NYSE, AMEX, and NASDAQ stock. [2] There are in total 16694 observations recorded from year 1950-2015. Since I am looking into the loss extremes, I convert the loss extremes to the right tail by multiplying -1 for the raw data. Next, the transformation is performed on the data set following the procedure in

[2]The 30 industry portfolios dataset is produced by Kenneth French Data Library: http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/det_30_ind_port.html

pseudocode 1 at Appendix I, after which the extreme sample is selected as the top 5% transformed observations with the largest Euclidean norms.

Before the clustering, the number of cluster k is decided. Although there is no convincing theory for selecting k, it is beneficial to explore how general the clustering is compared to the others with a range of number of clusters. Janßen and Wan (2020) suggests plotting $W(A_k^n)$ (see equation(10)) generated by using a range of different k in k-means clustering, where the value of $W(A_k^n)$ decreases as k increases. Hence, we can utilize this "elbow plot" to look for a proper value k, such that clustering with more clusters cannot bring in significant decrease in $W(A_k^n)$. Even though there is no comparable evaluation tool suggested by Fomichov and Ivanovs (2020) for k-PC, this paper proposes to plot the explanatory index v (see equation(12)) against k for this purpose. Note that, k and v increase together. As shown in figure 1, the plots don't suggest a clear value for k. And both plots exhibit similar pattern. So following the same decision by Janßen and Wan (2020), I compare $k = 5$ and $k = 10$ for analysis.

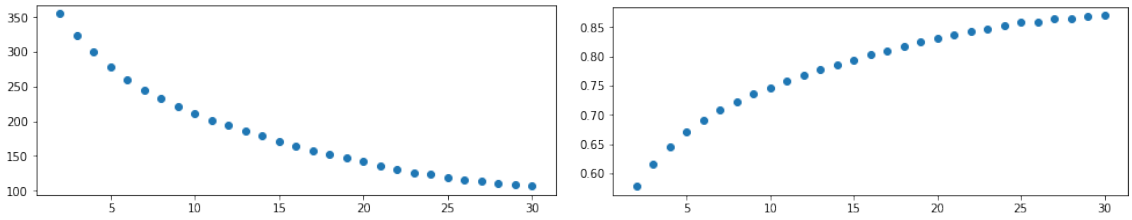

Figure 1: The information plots constructed by different values of k in the financial portfolio loss data. Left: y-axis represents the corresponding value of the minimized mean distance $W(A_k^n)$(equation(10)) obtained from a k-means clustering. Right: y-axis represents the corresponding explanatory index v (equation(12)) obtained from a k-pc clustering.

In Figure 2 and Figure 3, the estimated cluster centers produced by k-means approach are present at the top panels, and the corresponding cluster classification results of observations are plotted against time at the bottom panels. And Figure 5 (see AppendixV) contains the plots of estimated prototypes by k-PC, which are almost identical to the k-means' except for some small value difference potentially caused by their different ways to compute the cluster centers. The prototypes plots of k-means are slightly different from those in Janßen and Wan (2020), that is because they are not normalized and scaled to 1 for a clear comparisons to the estimations produced by k-PC.

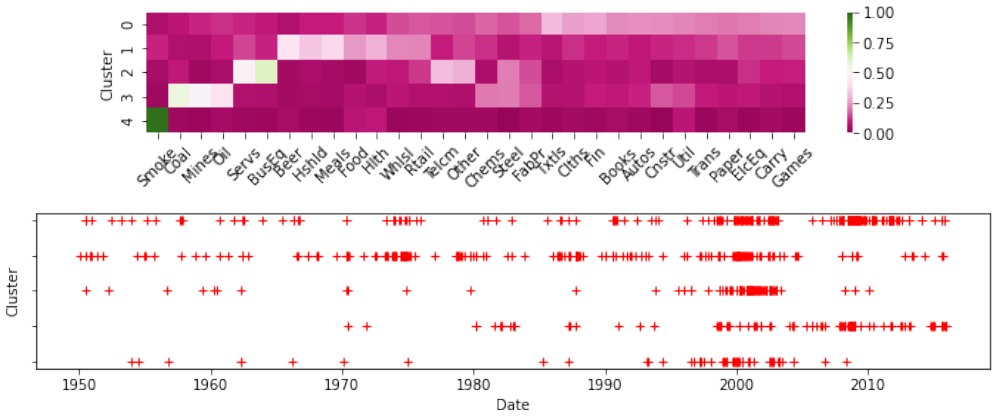

Figure 2: The clustering outcomes for transformed financial portfolio losses extremes by k-means approach when $k = 5$ (the cluster order is 0 to 4 from top to bottom on the y-axis of the time plot at the bottom panel)

Based on Figure 2, the 30 industries are grouped into 5 sectors. Consumer goods and personal necessity industries are concentrated in cluster 1. Cluster 2 involves the business and service sector. Cluster 3 focuses on the resources and commodities industries. Cluster 4 indicates the asymptotic independence of tobacco industry to the others. And cluster 0 doesn't suggest a clear sector but it contains the remaining industries. We can verify the clustering by cross-checking the time plot of the classified observations. The time plot for cluster 1 shows that the losses extremes in consumer oriented industries match the time points of US recessions. Since the income and employment rates of people are crucial to those industries' performance, it explains the relation. All cluster have a dense period around 2000, which is the year when dot-com bubble crashed and the whole stock market was affected. Extreme losses occurred frequently around that period in the business and IT sector, which is the origin of dot-com bubble. For the energy and resources industries focused by cluster 3, the extreme events are likely caused by the long-lasting energy crisis and the mining regulations enforced from the 2000s. Additionally, the conflicts in middle-east since 2000 also caused oil stocks to be highly volatile. The extremal observations labelled by cluster 4 are related to the serious of lawsuits filed by the U.S government against tobacco industry from 1999 to 2000. In 2000 and 2002, more regulations were released and the 'smokefree' law was passed in most states of America, leading another crash to tobacco industry.

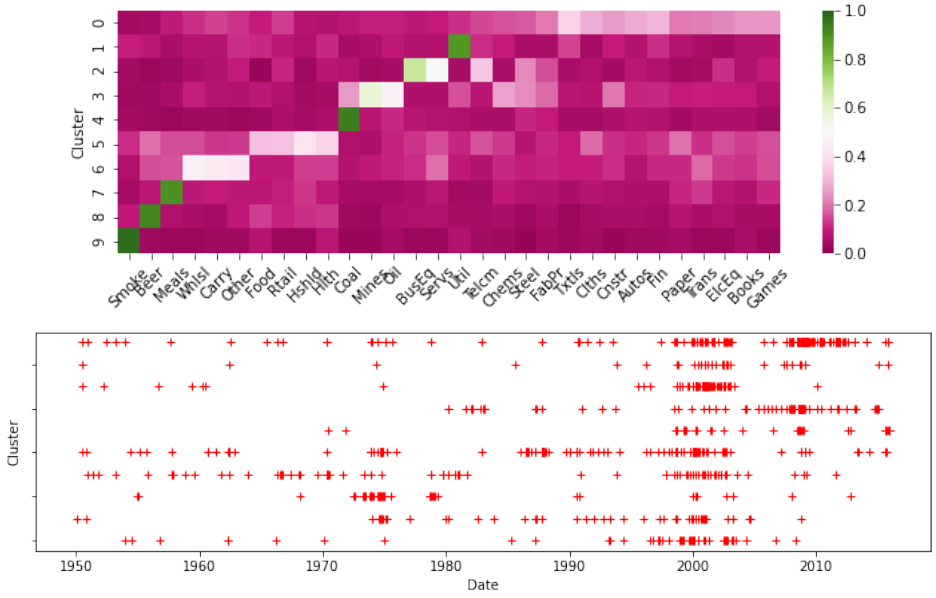

Figure 3: The clustering outcomes for transformed financial portfolio losses extremes by k-means approach when $k = 10$ (the cluster order is 0 to 9 from top to bottom on the y-axis of the time plot at the bottom panel)

Figure 3 presents the clustering results of the financial losses extreme data using $k = 10$. The connections found from $k = 5$ still apply, especially for the links associate to the business sector (cluster 2), the resource and energy sector (cluster 3) and the consumer goods sector (cluster 5), hinting asymptotic dependency. Furthermore, some previous bigger groups of components are now split into several clusters where only one or few components exhibit large values. Asymptotic independence is hinted for utility, coal, meals, beer and tobacco industries (cluster 1,4,7,8,9). Similar conclusions can be drawn from the corresponding time plots.

Nevertheless, this financial data set covers a total time length of 65 years, during which dependence structure might alter. Also the observations are not i.i.d to each other for continuous time series data, which violates the assumptions made in Section 2.

5.2 *Dietary Intakes*

2015–2016 NHANES report conducted a dietary interview, where they documented the food and beverage intakes of participants within a day and calculate the amount nutrients absorbed[3]. It would be informative to discover the inner dependent relationships of nutrients' dosages, as it helps to reveal the hidden risks from daily diets. In this section, I use the exact data set of 38

[3]This dataset is available at: https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/DR1TOT$_I$.XPT

selected nutrients analyzed by Janßen and Wan (2020) and reproduce the extremal analysis with both k-means and k-PC approaches.

After the data set is transformed with similar procedure as introduced in the financial losses sample in Section 5.1, the top 5% transformed observations with the largest Euclidean norm are selected as extremes. Like before, the information plots are firstly provided for the selection of k (see Figure 6 from Appendix). But it does not seem to have a clear value of k yet, according to both k-means' and k-PC's evaluation. So the number of cluster $k = 15$ and $k = 20$ utilized Janßen and Wan (2020) are reused for the following clustering.

Figure 4 present the estimated cluster centers result generated by the k-means clustering approach. Apparently, many nutrients are the only components that takes big value on estimated cluster centers, which hint asymptotic independence for those nutrients. For nutrients that show potential asymptotic dependence are: lutein and vitamin k, which richly exist in green leafy vegetables; fat and fat-acid, which often appear together; sugar and carbs, which are the key component in staple food; vitamin B6, vitamin B2 and Niacin, which are rich in chicken; retinol, vitamin A and vitamin E, which are rich in liver; magnesium and copper, which are rich in nuts.

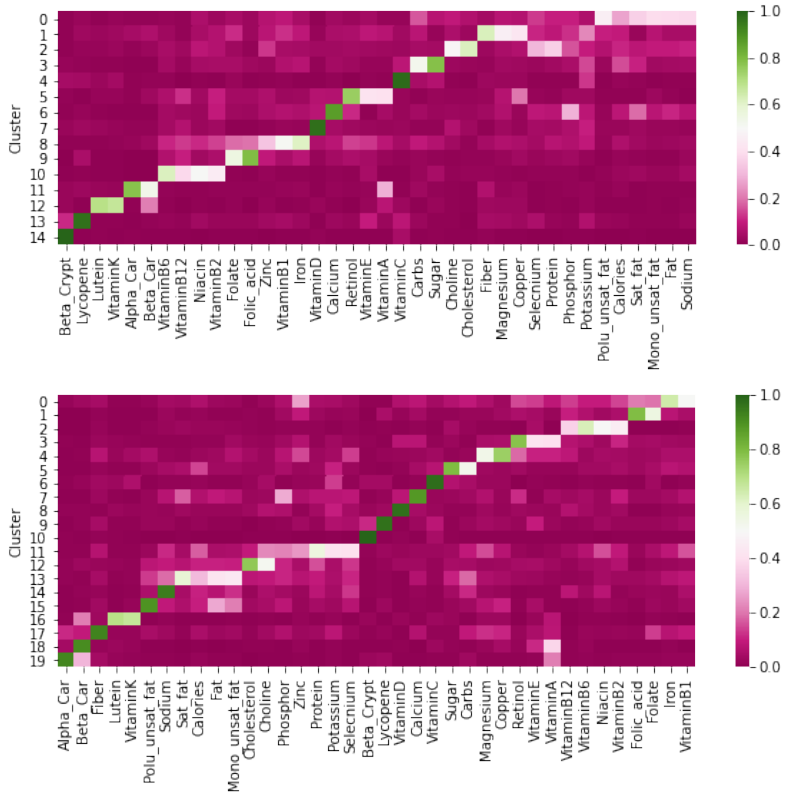

Figure 4: The clustering outcomes for transformed dietary intakes extremes by k-means approach when $k = 15$ and $k = 20$

As concluded from the numerical experiments in Section 4, when the dimensions d and number of cluster k is large, the performance of both clustering approaches worsen. It may explain that the clustering outcomes by k-PC (see Figure 7 in Appendix VII) deviates from the k-means'. Although the results are largely the same, the few difference indicates that the underlying dependency relations may not be fully explored.

6 Conclusions and Future Research

In this paper, recent clustering techniques spherical k-means and spherical k-principal-components are analyzed and compared by means of simulation experiments and empirical case studies. The results from both approaches are very similar. Although k-PC approach may perform slightly better in term of accuracy from simulation data, k-means is much lighter in computation, while the running time complexity of k-PC dramatically increases as number of clusters and dimensions become larger. The above conclusions also apply to our empirical data analysis, where the outcomes are really close at the portfolio data with smaller k and d, and the difference becomes noticeable for dietary intakes data with rather large k and d. Lastly, the misspecification experiments suggest that both methods perform better when k is too large rather than too small.

Future Research

Further Simulation Experiments

Although both approaches discussed in this paper provide similar performance according to my numerical experiments, it might give a misleading idea that k-PC makes the model unnecessarily complex compared to the k-means solution. However, this may not be correct, since my simulation tests haven't compared the gains of those methods at each iteration. It is completely possible that k-PC can approach the optimal quicker. k-PC considers centroids as the first principal eigenvectors, which are the scaled towards the variant-heavy side and are equivalent to the mean vector if and only if all covariances between angular samples hold the exact same value. I wonder whether looking for the principal eigenvectors at early iterations provides closer-to-actual prototypes, which enables k-PC to obtain a good level of estimations faster and be more robust to faraway initial centroids. So, I can formulate the further simulation research question as: *"Which method moves faster towards the optima at each iteration?"*

To measure the corresponding gains of each iteration, we can utilize the estimated centroids

provided by each iteration calculate and record their distances to the real centroids respectively. In this way, the deceased distance from last iteration $\Delta_{T,T-1}$ can be recognized as the gain achieved from the T^{th} iteration, take k-means for example.

Hierarchical Clustering

Both Janßen and Wan (2020) and Fomichov and Ivanovs (2020) have laid solid mathematical foundations in applying partitional clustering for dependency structure exploration. While parallel grouping is certainly a good starting point to describe the unknown structure of data, we can also attempt other clustering logic to receive richer information, such as adopting a divisive (top-down) hierarchical clustering approach (Johnson, 1967). Note that the main difference between partitional clustering and hierarchical clustering is that the former requires a good knowledge of the number of groups before searching, while the latter doesn't. Partitional clustering, such as k-means and k-pc, requires a constant number of clusters to define the initial centroids as part of the input, which then is updated iteratively toward optimum based on the criterion. In contrast, a top-down hierarchical clustering can start with separating the population into two clusters, after which the new subset of samples in each cluster are further decomposed into two new clusters. Instead of presetting a fixed amount of clusters, this algorithm requires stopping criterion such as the number of 'layers' of this process, a minimal amount of samples on each node, and a within-group dissimilarity threshold to enable the next decomposition. Roughly speaking, this hierarchic-based clustering might be superior to k-means or k-pc approaches in empirical analysis when the assessment of the 'correct' amount of cluster is dubious. Because it presents the full process of how samples are decomposed to smaller subsets, thus users can observe the way clustering shifts and decide with external knowledge which layer of clustering makes the most sense among those 'branches' provided by the algorithm. Therefore, I recommend adopting the k-means and k-pc approaches into hierarchical-means and hierarchical-pc clustering as a topic of further research for this paper.

Acknowledgements

The author gratefully acknowledges Phyllis Wan for the valuable insights, helpful discussions and kind supervision regarding this bachelor thesis.

References

Asadi, P., Engelke, S., & Davison, A. C. (2018). Optimal regionalization of extreme value distributions for flood estimation. *Journal of Hydrology*, *556*, 182–193.

Buchta, C., Kober, M., Feinerer, I., & Hornik, K. (2012). Spherical k-means clustering. *Journal of statistical software*, *50*(10), 1–22.

Chautru, E. et al. (2015). Dimension reduction in multivariate extreme value analysis. *Electronic journal of statistics*, *9*(1), 383–418.

Chiapino, M., & Sabourin, A. (2016). Feature clustering for extreme events analysis, with application to extreme stream-flow data. *International Workshop on New Frontiers in Mining Complex Patterns*, 132–147.

Chiapino, M., Sabourin, A., & Segers, J. (2019). Identifying groups of variables with the potential of being large simultaneously. *Extremes*, *22*(2), 193–222.

Coles, S., Heffernan, J., & Tawn, J. (1999). Dependence measures for extreme value analyses. *Extremes*, *2*(4), 339–365.

Cooley, D., & Thibaud, E. (2019). Decompositions of dependence for high-dimensional extremes. *Biometrika*, *106*(3), 587–604.

Davison, A. C., & Huser, R. (2015). Statistics of extremes. *Annual Review of Statistics and its Application*, *2*, 203–235.

Dhillon, I. S., & Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine learning*, *42*(1), 143–175.

Ding, S., Zhang, N., Zhang, J., Xu, X., & Shi, Z. (2017). Unsupervised extreme learning machine with representational features. *International Journal of Machine Learning and Cybernetics*, *8*(2), 587–595.

Drees, H., & Sabourin, A. (2019). Principal component analysis for multivariate extremes. *arXiv preprint arXiv:1906.11043*.

Engelke, S., & Hitz, A. S. (2020). Graphical models for extremes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *82*(4), 871–932.

Engelke, S., & Ivanovs, J. (2020). Sparse structures for multivariate extremes. *Annual Review of Statistics and its Application*, *8*.

Engelke, S., & Volgushev, S. (2020). Structure learning for extremal tree models. *arXiv preprint arXiv:2012.06179*.

Fomichov, V., & Ivanovs, J. (2020). Detection of groups of concomitant extremes using clustering. *arXiv preprint arXiv:2010.12372*.

Gao, J., & Suzuki, H. (2003). Unsupervised learning of dependency structure for language modeling.

Gnecco, N., Meinshausen, N., Peters, J., & Engelke, S. (2019). Causal discovery in heavy-tailed models. *arXiv preprint arXiv:1908.05097*.

Goix, N., Sabourin, A., & Clémençon, S. (2017). Sparse representation of multivariate extremes with applications to anomaly detection. *Journal of Multivariate Analysis*, *161*, 12–31.

Janßen, A., & Wan, P. (2020). *k*-means clustering of extremes. *Electronic Journal of Statistics*, *14*(1), 1211–1233.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, *32*(3), 241–254.

Kleinberg, J., Papadimitriou, C., & Raghavan, P. (1998). A microeconomic view of data mining. *Data mining and knowledge discovery*, *2*(4), 311–324.

Ledford, A. W., & Tawn, J. A. (1997). Modelling dependence within joint tail regions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *59*(2), 475–499.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, *1*(14), 281–297.

Meyer, N., & Wintenberger, O. (2019). Detection of extremal directions via euclidean projections. *arXiv preprint arXiv:1907.00686*.

Poon, S.-H., Rockinger, M., & Tawn, J. (2003). Modelling extreme-value dependence in international stock markets. *Statistica Sinica*, 929–953.

Schuhmacher, D., Bähre, B., Bonneel, N., Gottschlich, C., Hartmann, V., Heinemann, F., Schmitzer, B., Schrieber, J., & Wilm, T. (2017). Package 'transport'. *R package version*, *1*(4).

Vallender, S. (1974). Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, *18*(4), 784–786.

Appendix I: Pseudocode for Transformation Procedure

Algorithm 1: Transformation

input : Raw data $\underset{n\times d}{X} = \{X_1, \ldots, X_d\}$

output: Extremal direction data $\underset{n\times d}{\Theta}$

begin

Standardization: $Y \leftarrow (\frac{1}{1-F_1(X_1)}, \ldots, \frac{1}{1-F_d(X_d)})$;

Extreme selection: $Y^* \leftarrow \|Y\| > \mu$, μ: threshold;

Direction transformation on unit sphere: $\Theta \leftarrow \frac{Y^*}{\|Y^*\|}$;

end

Appendix II: Pseudocode for k-means Approach

Algorithm 2: Spherical k-means clustering (a single iteration)

input : Extremal direction data $\theta_1, \ldots, \theta_n$ and current centroids $a_1, \ldots, a_k$

output: new centroids $\hat{a}_1, \ldots, \hat{a}_k$ and updated minimized distance $W(A_k^n)$

begin

Centroid update:

Cluster $\theta_1, \ldots, \theta_n$ their closest centroids $a_1, \ldots, a_k \Leftrightarrow \min_{a_c \in A} d(\theta_i, a_c)$;

Minimized distance update: $W(A_k^n) \leftarrow \frac{1}{n} \sum_{i=1}^{n} \min_{a_c \in A} d(\theta_i, a_c)$;

end

Appendix III: Pseudocode for k-PC Approach

Algorithm 3: Spherical k-principal-components clustering (a single iteration)

input : Extremal direction data $\theta_1, \ldots, \theta_n$ and current centroids $a_1, \ldots, a_k$

output: new centroids $\hat{a}_1, \ldots, \hat{a}_k$ and updated explanatory index v

begin

Dot-product matrix: $\underset{n\times k}{M} \leftarrow (\theta_1, \ldots, \theta_n)^T (a_1, \ldots, a_n)$;

Explanatory Index: $v \leftarrow$ the average row-maxima M ;

Cluster location matrix:

Γ with entry $\gamma_{c,i}$ storing whether location c is the (first) maxima of each observation i;

Centroid Update:

for $c = 1, \ldots, k$ **do**

Matrix consists of all observations of cluster c**:** $\Sigma_c \leftarrow \frac{1}{n} \sum_{i=1}^{n} (\theta_i^T \theta_i \gamma_{c,i})$;

New centroid of cluster c**:** $\hat{a}_c \leftarrow$ the first principal eigenvector of Σ_c

end

end

Appendix IV: Specifications of The Random Coefficient Factor b_c

For each type of simulation data , the following specification only determines the first $k-1$ factors of it, since the last factor can be automatically calculated by the standardization assumption of B. The detailed specifications are listed as follows, where U_i are the random vectors generated from an uniform distribution:

- $d = 4, k = 2$: First factor is $(U_1, U_2, U_3, U_4)/2$
- $d = 4, k = 6$: First five factors are $(U_1, U_2, U_3, U_4)/3$,$(U_5, 0, U_6, 0)/3$,$(0, U_7, 0, U_8)/3$,$(U_9, U_10, 0, 0)/3$,$(0, 0, U_1$
- $d = 20, k = 2$: First factor is $(U_1, U_2, \ldots, U_{20})/2$
- $d = 20, k = 3$: First two factors are $(U_1, U_2, \ldots, U_{20})/2$, $(U_1, U_2, \ldots, U_{10}, 0, \ldots, 0)/2$
- $d = 20, k = 5$: First four factors are $(U_1, U_2, \ldots, U_5, 0, \ldots, 0) \times 0.9$, $(0, \ldots, 0, U_6, U_7 \ldots, U_{10}, 0, \ldots, 0) \times 0.9$, $(0, \ldots, 0, U_{11}, U_{12}, \ldots, U_{15}, 0, \ldots, 0) \times 0.9$, $(0, \ldots, 0, U_{16}, U_{17}, \ldots, U_{20}) \times 0.9$
- $d = 20, k = 10$: First nine factors are $(U_1, U_2, \ldots, U_{20})/2$, $(U_1, U_2, 0, \ldots, 0)/2$, $(0, 0, U_3, U_4, 0, \ldots, 0)/2$, $(0, \ldots, 0, U_5, U_6, 0, \ldots, 0)/2$, $(0, \ldots, 0, U_7, U_8, 0, \ldots, 0)/2$, $(0, \ldots, 0, U_9, U_{10}, 0, \ldots, 0)/2$,

$$(0,\ldots,0,U_{11},U_{12},0,\ldots,0)/2,\ (0,\ldots,0,U_{13},U_{14},0,\ldots,0)/2,\ (0,\ldots,0,U_{15},U_{16},0,\ldots,0)/2,$$

Appendix V: The Estimated Prototypes of Financial Portfolio Losses using k-Principal-Components Approach

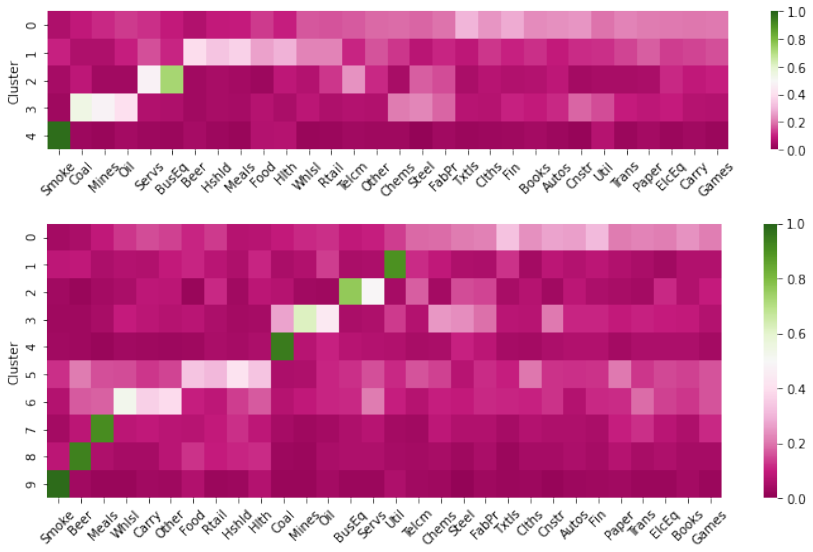

Figure 5: The clustering outcomes for transformed financial portfolio losses extremes by k-PC approach when $k = 5$(the top panel) and $k = 10$(the bottom panel)

Appendix VII: The Information Plots for Dietary Intakes Data

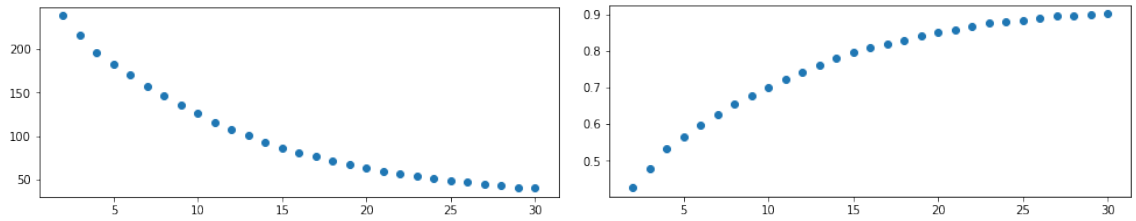

Figure 6: The information plots constructed by different values of k in the dietary intakes data. Left: y-axis represents the corresponding value of the minimized mean distance $W(A_k^n)$ (equation (10)) obtained from a k-means clustering. Right: y-axis represents the corresponding explanatory index v (equation (12)) obtained from a k-PC clustering.

Appendix VII: The Estimated Prototypes of Dietary Intakes using k-Principal-Components Approach

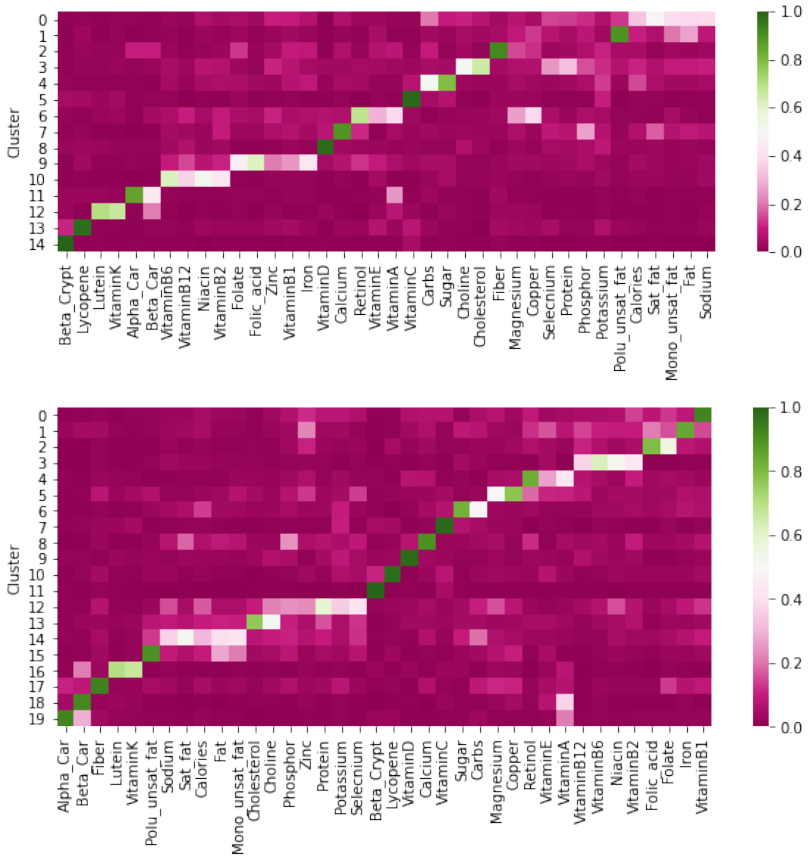

Figure 7: The clustering outcomes for transformed dietary intakes extremes by k-means approach when $k = 15$(the top panel) and $k = 20$(the bottom panel)

Appendix VIII: Description for codes

In this research, the main language for data process and visualization is *Python*, and the main language for k-means and k-PC clustering algorithm for the numerical experiments in Section 4 and empirical case analysis in Section 5 is *R*. Thus, there are three sections of codes, namely, data process and visualization, simulation and empirical case studies. Note that, the *R* codes are edited on *Kaggle*, which provides notebook format instead of R-Studio format. Codes for this study is attached in the zip-folder with this article, which are also accessible via Github page https://github.com/RinaPiggy/k-clustering for an easy notebook vision.