K-expectiles clustering

Bingling Wang *
Yingxing Li * *2
Wolfgang Karl Härdle * *3 *4 *5

* Humboldt-Universität zu Berlin, Germany
*2 Xiamen University, China
*3 Singapore Management University, Singapore
*4 Charles University, Czech Republic
*5 National Chiao Tung University, Taiwan

This research was supported by the Deutsche Forschungsgesellschaft through the International Research Training Group 1792 "High Dimensional Nonstationary Time Series".

http://irtg1792.hu-berlin.de
ISSN 2568-5619
Abstract

K-means clustering is one of the most widely-used partitioning algorithm in cluster analysis due to its simplicity and computational efficiency, but it may not provide ideal clustering results when applying to data with non-spherically shaped clusters. By considering the asymmetrically weighted distance, We propose the K-expectile clustering and search the clusters via a greedy algorithm that minimizes the within cluster \( \tau \)-variance. We provide algorithms based on two schemes: the fixed \( \tau \) clustering, and the adaptive \( \tau \) clustering. Validated by simulation results, our method has enhanced performance on data with asymmetric shaped clusters or clusters with a complicated structure. Applications of our method show that the fixed \( \tau \) clustering can bring some flexibility on segmentation with a decent accuracy, while the adaptive \( \tau \) clustering may yield better performance. All calculation can be redone via quantlet.com.

Keywords: clustering, expectiles, asymmetric quadratic loss, image segmentation
1 Introduction

Clustering is a useful technique to discover and identify homogenous groups of data points in a given sample. As an unsupervised learning algorithm, it aims to extract information on the underlying characteristics via dividing the data into groups that maximize common information. Obviously the information about homogeniety of groups is key in such a sample dividing mechanism. Among the simplest choice is the $K$-means clustering method described by Steinhaus (1956) and Hartigan (1975), which adopt the Euclidean distance as neighbourhood measure, thus leading to spheres as silhouettes and means as centers of clusters. Indeed, while keeping a balance between group size and information gain, $K$-means is the most widely used partitioning algorithm due to its simplicity, efficiency in computing and easiness of interpretation. Successful applications include signal processing, image identification, customer segmentation.

The principle of a partitioning clustering algorithm is to assign data points to the nearest cluster by optimising some objective function. The objective function of $K$-means is the sum of within-group variance, and thus the correspondence cluster centers are the mean of each cluster. Minimizing the objective function is equivalent to maximizing the log-likelihood function with independent Gaussian density. Although $K$-means clustering is often viewed as a "distribution free" algorithm, it is actually partitioning using equal sized spherical contour lines which can be considered as assuming independent identically distributed (i.i.d.) Gaussian clusters. Therefore, $K$-means approach works better for cluster in the symmetric distribution than the skewed ones.

On the other hand, when applied on skewed or asymmetric distributed data whose characteristics may not be fully captured by the first two moments, new methods are required for non-spherical cluster. To account for within-cluster skewness, Hennig et al. (2019) introduce the $K$-quantile clustering algorithm based on the asymmetric absolute discrepancy. Then they linked their approach to a fixed partition model of genralized asymmetric Laplace distributions. This quantile discrepancy based density relies on both the quantile level $\tau$ and some additional scale/penalty parameter $\lambda$. However, $\tau$ and $\lambda$ are assumed the same across different clusters to reduce the computation complexity. An analogous work on quantile based clustering is proposed in Zhang et al. (2019), where they developed a model-based iterative algorithm to identify subgroups with heterogeneous slopes. In particularly, they consider clustering across multiple quantiles to capture the full picture of heterogeneity. For that accordance, how to specify the appropriate quantile level vector $\tau$ could be a problem for large dimensional data.

This motivates us to consider a novel method, $K$-expectile clustering. This method is based on a similar idea as $K$-means but with an expectile cluster center and aims at minimizing the so-called $\tau$-variance, which is a weighted quadratic loss to take into account asymmetry. Besides being simple and fast, our algorithm can be applied on wider range of data compared with $K$-means. In particular, we consider two schemes, either with a pre-specify $\tau$ level or an adaptive $\tau$ that may vary across different dimensions or clusters, which accommodates either a fixed cluster shape or a data-driven cluster shape to capture heterogeniety.

To better understand the basic ideas of $K$-expectile clustering, we recall some basic knowledge about tail events. Quantile regression (Koenker and Bassett Jr, 1978) and expectile regression (Newey and Powell, 1987) have been suggested for displaying the whole picture of the conditional distribution of response variable on covariates, especially for data not sufficing the condition of homoskedasticity or conditional symmetry. For a
random variable $X \in \mathbb{R}$ drawn from distribution $F$, a location model of $\tau$-th tail event
measure with $\tau \in (0, 1)$ could be defined as:

$$x_i = \theta_\tau + \varepsilon_i, \; i = 1, \ldots, n.$$  

With an assumption on the $\tau$-th quantile or expectile of the cdf of $\varepsilon$ being zero, $\theta_\tau$ is by
definition the $\tau$-th quantile or expectile of $X$ accordingly. An estimator of the location
model of quantiles and expectiles can be naturally formed:

$$\hat{\theta}_\tau = \arg \min_{\mu \in \mathbb{R}} \mathbb{E}[\rho_\tau(X - \mu)],$$

where the loss function $\rho_\tau(\cdot)$ is defined as:

$$\rho_\tau(u) = |u|^\alpha \tau - \mathbf{I}_{\{u \leq 0\}},$$

with $\alpha = 1$ and $\alpha = 2$ respectively.

Although the concept of expectiles is natural analogues of quantiles, expectiles enjoy
the computation efficiency over quantiles (Schnabel, 2011). In finance, the expectile might
be preferred as a favorable risk measures due to its desirable properties such as coherence
and elicitability (Kuan et al., 2009, Ziegel, 2016). Recently, the use of expectiles attracts
more and more attention, such as the nonparametric expectile regression by Sobotka and
Kneib (2012) and Yang et al. (2018), the principle expectile analysis by Tran et al. (2019).
Our proposed $K$-expectile clustering allows us to take into account tail characteristics and
asymmetry when identifying homogenous groups of data, while simulation studies and
applications justify its excellent performance.

The rest of the paper is organized as follows. In section 2, we will briefly review the
classical $K$-means algorithm, and then propose our $K$-expectile clustering in two schemes.
In section 3, we present the simulation study that includes data from different distribution
and compare the performance of $K$-expectiles clustering with other methods. Section
4 applies our method to real crypto currency market analysis and image segmentation.
Codes of all the functions, applications and data are uploaded to quantlet.com.

## 2 Methodology

### 2.1 $K$-means clustering

Suppose the data set $X = \{X_i\}_{i=1}^n$ comes from a random sample in $\mathbb{R}^p$. A clustering algo-
rithm denoted by $Q(\cdot)$ generates $K$ subsets $\{G_1, G_2, \ldots, G_K\}$ each with distribution $p_k(X)$. Any clustering algorithm maps $X$ into a membership vector $C = (c(1), c(2), \ldots, c(n))$, i.e. $Q(x_i) = c(i)$, and $G_k = \{x_i : c(i) = k\}$, $c(i) \in \{1, 2, \ldots, K\}$.

A clustering criterion is defined via a cost function. In $K$-means clustering, the cost
is defined as the sum of squared Euclidean distance between cluster members to the
cluster centroids. Indeed the centroids can be considered as location parameters for
clusters. In $K$-means clustering, cluster centroids are actually cluster means and the cost
function is the sum of within-cluster variance. Let $G(\cdot)$ be the $K$-means objective function,
$\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$ be a set of cluster centroids with $\theta_k \in \mathbb{R}^p$,

$$G(\Theta, C, X) = \min_\Theta \sum_{k=1}^K \sum_{x_i \in G_k} \|x_i - \theta_k\|^2.$$
Clustering is now turned into an optimisation problem and is solved via iteration. For a fixed $\Theta$, partition $c(i)$ is achieved by assigning each point to the nearest cluster centroid.

$$c(i) = \arg\min_{k \in \{1, 2, \ldots, K\}} \| x_i - \theta_k \|^2.$$  

For a fixed membership vector $C$, the centroid $\theta_k$ can be estimated by taking the within-cluster mean.

$$\hat{\theta}_k = \bar{x}_k = \frac{1}{|G_k|} \sum_{x_i \in G_k} x_i.$$  

Suppose $X$ are drawn i.i.d. from an equally-weighted Gaussian Mixture Model, with conditional unknown expectation $\mu_k$ and variance $\Sigma_k$. The estimation of this model requires the EM algorithm, and the details are explained in Deisenroth et al. (2020). If let $\mu_k$ be the $k$-th cluster centroid, and covariance matrix simply equals to the identity matrix, then the $K$-means objective function is coincide with the expectation function in EM algorithm of a Gaussian Mixture Model with equal mixture weights. By a "hard" assignment of data points to nearest cluster centroid in $K$-means algorithm as described in MacKay and Mac Kay (2003), the computation of parameter $\Theta$ can be easily conducted by independently estimating cluster means. Simultaneously $K$-means clustering is a distribution-free but distance based clustering technique.

### 2.2 $K$-expectiles clustering

Now consider a set of data with skewed or asymmetrically distributed clusters, i.e., cluster centroids $\Theta$ are not located on means and cluster variances are heterogeneous on different sides around centroids. As said before, it is the information measure of homogeneity that yields the clusters. A distance with cluster centroids offset from means and a distance metric which takes asymmetry into account is certainly a more flexible way of dividing groups.

For that purpose, assume each cluster is a group of data drawn independently from some distribution. Define the cluster centroid $\theta_k$ as the $\tau$ expectile of cluster distribution, and assign points according to expectile distances. Take the univariate random variable $X \in \mathbb{R}$ as an example. For a fixed $\tau \in (0, 1)$, the $\tau$-th expectile of $X$ as proposed by Newey and Powell (1987) is identified as the minimizer of the asymmetric quadratic loss

$$e_\tau(X) = \arg\min_{\mu \in \mathbb{R}} \mathbb{E}[\rho_\tau(X - \mu)],$$  

where $(x)_+ = \max(x, 0)$. It is worth noting here that the expectile location estimator can be interpreted as a maximum likelyhood estimator of a normal distributed sample with an unequal weight placed on positive and negative disturbances, showed in Aigner et al. (1976).

For $X \in \mathbb{R}^p$, define $(X)_+ = ((X_1)_+, \ldots, (X_p)_+)^\top$, then the multivariate expectile is:

$$e_\tau(X) = \arg\min_{\mu \in \mathbb{R}^p} \mathbb{E}[\tau \|(X - \mu)_+\|^2 + (1 - \tau)\|\mu - X\|_1^2].$$  

Here the dependence is taken into account by using the norm. Note that the construction of the multivariate expectiles are related to the dependence structure of each components. The choice of dependence modelling may differ according to the practical goal. For
simplification, we first elaborate the case when the dependence structure is ignored. Then
the multivariate expectile \( e_\tau \in \mathbb{R}^p \) can be defined via the marginal univariate expectiles,
\( e_\tau(X) = (e_\tau(X_1), \ldots, e_\tau(X_p))^\top \), where
\[
e_\tau(X_j) = \arg \min_{\mu \in \mathbb{R}^p} \mathbb{E} \left[ \tau \{(X_j - \mu_j)^2 \} + (1 - \tau)\{(\mu_j - X_j)^2 \} \right]
\]
We could gain further flexibility and more power of expectiles in \( p \) dimension by setting \( \tau = (\tau_1, \tau_2, \ldots, \tau_p)^\top \in \mathbb{R}^p \), making the \( \tau \) level variable over dimensions. Then we obtain \( e_\tau(X) = (e_{\tau_1}(X_1), \ldots, e_{\tau_p}(X_p))^\top \), where
\[
e_{\tau_j}(X_j) = \arg \min_{\mu \in \mathbb{R}^p} \mathbb{E} \left[ \tau_j \{(X_j - \mu_j)^2 \} + (1 - \tau_j)\{(\mu_j - X_j)^2 \} \right]
\]
The empirical version reads as:
\[
\hat{e}_{\tau,n}(X) = \arg \min_{\mu \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^p \left\{ \tau_j (x_{ij} - \mu_j)^2 + (1 - \tau_j) (\mu_j - x_{ij})^2 \right\}, \quad (4)
\]
whose idea is to consider an asymmetrically weighted distance function with \( L_2 \) norm and fix
the cluster centroids at the empirical expectile of the \( k \)-th cluster, i.e. \( \hat{\theta}_k = \hat{e}_{\tau,n}(X \in G_k) \).
Instead of specifying an asymmetric form of distribution, we form a \( K \)-expectile
objective function by defining an asymmetric \( \tau \)-variance as described in Tran et al. (2019),
which yields to an axis-aligned ellipsoid unit ball. To include covariance or correlation,
usually a matrix form of multivariate expectile will be considered. By introducing a \( p \times p \)
symmetric matrix \( \Sigma \), one can form a score function as described in Maume-Deschamps
et al. (2016),
\[
e_{\tau}^\Sigma(X) \in \arg \min_{\mu \in \mathbb{R}^p} \mathbb{E} \left[ \tau (x - \mu)^\top \Sigma (x - \mu) + (1 - \tau) (x - \mu)^\top \Sigma (x - \mu) \right]. \quad (5)
\]
In Figure 2.1, the contour lines of unit circles of bivariate \( \tau \)-variance with various \( \tau \)
levels on each axis are shown along with unit circles of a symmetric variance in the back.
The covariance matrix is the inverse matrix of \( \Sigma \) in function (5). The last sub-plot shows
the unit circles with different scales on two axis. These are equivalent to the contour lines
of independent bivariate asymmetric normal distributions in comparison with the contour
lines of independent normal distributions. Figure 2.2 shows the 3D contour surface of
\( \tau \)-variance unit ball with different \( \tau \)-levels on each dimension, or the 3D cluster shapes.

### 2.3 Fixed \( \tau \) clustering

Recall that for a univariate observation \( x \in \mathbb{R} \), we could define \( \tau \)-distance as
\[
d(x, \tau, \theta) = \left\{ \tau + (1 - 2\tau) I_{\{x < \theta\}} \right\} (x - \theta)^2, \quad (6)
\]
which coincides with loss function (2). Therefore, for a pre-specified \( \tau \) vector, define the
objective function
\[
G^{Fixed}(\tau, \Theta, C, X) = \sum_{k=1}^K \sum_{x_i \in G_k} \sum_{j=1}^p d(x_{ij}, \tau_j, \theta_k) \quad (7)
\]
\[
= \sum_{k=1}^K \sum_{x_i \in G_k} \sum_{j=1}^p \left\{ \tau_j + (1 - 2\tau_j) I_{\{x_{ij} < \theta_{k,j}\}} \right\} (x_{ij} - \theta_{k,j})^2, \quad (8)
\]
Figure 2.1: Contour lines of unit balls of various $\tau$ variances in comparison to unit balls of symmetric variance

which aims to detect expectile-specified clusters by minimizing the sum of within-cluster $\tau$-variance.

For known $(\tau, C)$, cluster centroids $\Theta$ are found by:

$$\hat{\theta}_k = \arg \min_{\mu \in \mathbb{R}^p} \sum_{x_i \in G_k} \left\{ \tau_j + (1 - 2\tau_j) I_{x_{ij} < \mu_j} \right\} (x_{ij} - \mu_j)^2$$

$$= \arg \min_{\mu \in \mathbb{R}^p} \sum_{x_i \in G_k} \sum_{j=1}^p w_{ij}(\tau_j) (x_{ij} - \mu_j)^2$$

where $w(\tau)$ is a weight function which is related to $\mu(\tau)$, the location parameter at the given $\tau$ level.

$$w_{ij}(\tau_j) = \begin{cases} \tau_j & \text{if } x_{ij} \leq \mu_j(\tau_j) \\ 1 - \tau_j & \text{if } x_{ij} > \mu_j(\tau_j) \end{cases}$$

This implicit dependence of $w$ on $\mu(\tau)$ leads to the application of the Least Absolute Square Estimator (LAWS), a version of the Stochastic Gradient Algorithm. For a fixed $\mu_j(\tau_j)$, the weight $w_{ij}(\tau_j)$ in equation (11) is calculated, therefore a closed form solution of $\mu_j(\tau_j)$ can be expressed as

$$\mu_j(\tau_j) = \frac{\tau_j \sum_{i \in T_j} x_{ij} + (1 - \tau_j) \sum_{i \in T_j \setminus i} x_{ij}}{\tau_j n^+ + (1 - \tau_j) n^-}$$
Figure 2.2: 3D contour surface of unit balls of various $\tau$ variances. Left: $\tau = [0.1, 0.2, 0.1]$. Right: $\tau = [0.4, 0.5, 0.8]$.

**QKEC**_ cluster shapes

where

$$
\mathcal{I}_+ = \{i \in \{1, \ldots, n\} : w_{ij} = \tau_j, c(i) = k\}
$$

$$
\mathcal{I}_- = \{i \in \{1, \ldots, n\} : w_{ij} = 1 - \tau_j, c(i) = k\}
$$

$$
n^+ = |\mathcal{I}_+| \quad n^- = |\mathcal{I}_-|.
$$

Cluster centroids can be estimated by iteratively repeating the two steps until the location of $\mu_j(\tau_j)$ does not change, see:

---

**Algorithm 1 LAWS**

**Input:** The set of points in cluster $G_k$; The vector of parameter, $\tau$;

**Output:** Estimated cluster centroids, $\Theta$

1: Initialize $\mu_j^0(\tau_j)$ as mean of $j$-th column of $x_i \in G_k$

2: **repeat**

3: Assign weight $w_{ij}^{t+1}(\tau_j)$ to each point $x_{ij}$ based on $\mu^t(\tau)$

4: Update $\mu^{t+1}(\tau)$ according to equation (12) with input $w^{t+1}(\tau)$

5: **until** $\|\mu(\tau)^t, \mu(\tau)^{t-1}\| \leq \text{threshold}$

The $K$-expectiles clustering algorithm now read as follows:
Algorithm 2 Fixed $\tau$ clustering

Input: Data, $X$; Vector parameter, $\tau$; # of clusters, $K$;
Output: Cluster membership vector, $C$; Estimated cluster centroids, $\Theta$

1: Initialize centroids $\Theta^0 = \Theta_{k\text{-means}}$
2: repeat
3: Calculate cluster membership
   
   $$c(i)^{t+1} = \arg\min_{k \in \{1,2,...,K\}} \sum_{j=1}^{p} \sum_{c(i)=k} \left\{ \tau_j + (1 - 2\tau_j) I_{\{x_{ij} < \theta_{k,j}^t\}} \right\} (x_{ij} - \theta_{k,j}^t)^2$$
4: Update $\theta_{k,j}^{t+1}$ by applying Algorithm (1) with input $c(i)^{t+1}$
5: until $\|\theta^t - \theta^{t-1}\| \leq threshold$

2.4 Adaptive $\tau$ clustering

In the last section, clustering with fixed cluster shapes by pre-specifying the $\tau$ vector has been discussed, and this scenario is shown in Sub-plot 1 of Figure 2.3. In comparison, regarding the issue of clusters with different shapes as shown in Sub-plot 2 of Figure 2.3, we present the result of a fully adaptive algorithm, both for different dimensions and for different clusters. Without pre-defined $\tau$, we now assume $\tau$ is a $(K \times p)$ matrix. We optimize the following cluster objective function with respect to $\tau$ as well.

![Figure 2.3: Different scenarios regarding to cluster shapes](image)
\[ G^{Adaptive}(\tau, \Theta, C, X) = \sum_{k=1}^{K} \sum_{x_i \in G_k} \sum_{j=1}^{p} d(x_{ij}, \tau_{k,j}, \theta_{k,j}) \]
\[ = \sum_{k=1}^{K} \sum_{x_i \in G_k} \sum_{j=1}^{p} \left\{ \tau_{k,j} + (1 - 2\tau_{k,j}) I_{\{x_{ij} < \theta_{k,j}\}} \right\} (x_{ij} - \theta_{k,j})^2, \] (13)

For given \((\Theta, C)\), to optimize \(\tau\), require
\[ \hat{\tau}_k = \arg \min_{\tau \in (\mathbb{R}^p)^K} G^{Adaptive}(\tau, \Theta, C, X). \] (15)

By taking first order condition, we get the unique solution:
\[ \tau_{k,j} = \frac{\gamma_{k,j}}{1 + \gamma_{k,j}}, \] (16)

where
\[ \gamma_{k,j} = \frac{n^- \sum_{i \in I^-_k} \theta_{k,j} - x_{ij}}{n^+ \sum_{i \in I^+_k} x_{ij} - \theta_{k,j}}. \]

Then the clustering algorithm for adaptive \(\tau\) can be described as: The clustering algorithm can be implemented as follows:

**Algorithm 3** Adaptive \(\tau\) clustering

**Input:** Data, \(X\); \# of clusters, \(K\);  
**Output:** Cluster membership vector, \(C\); Estimated cluster centroids, \(\Theta\)

1: Initialize centroids \(\Theta^0 = \Theta_{k-means}\); \(\tau^0_{k,j} = 0.5\)
2: repeat  
3: Calculate the cluster membership by
\[ c(i)^{t+1} = \arg \min_{k \in \{1, 2, \ldots, K\}} \sum_{j=1}^{p} \sum_{i \in c(i)^{t+1} = k} \left\{ \tau_{k,j}^t + (1 - 2\tau_{k,j}^t) I_{\{x_{ij} < \theta_{k,j}^t\}} \right\} (x_{ij} - \theta_{k,j}^t)^2 \]
4: Update \(\theta_{k,j}^{t+1}\) by applying Algorithm (1) with input \(\tau_{k,j}^t\)
5: Update \(\tau_{k,j}^{t+1}\) according to equation (16) with input \(\theta_{k,j}^{t+1}\)
6: until \(\|\theta^t, \theta^{t-1}\| \leq \text{threshold}\)

Finally, we point out that our algorithm has the numerical convergence.

**Proposition 2.1.** The sequence produced by the adaptive \(\tau\) clustering algorithm could converge to its optimal solution.
3 Simulation

To evaluate the performance of \( K \)-expectiles clustering, we design four simulated samples with \( K \) clusters. Let \( k = 1, \ldots, K \), each cluster represented by \( G_k \) is an i.i.d. random sample drawn from a \( p \)-variate distribution in the size of \((n_1, n_2, \ldots, n_k)\). Each component of the multivariate distribution is assumed to be independent. Data set can be written as \( X = (G_1, G_2, \ldots, G_k) \). Scale, location and skewness of the distribution can cause the overlapping of multiple clusters which in turn influence the cluster shapes and within-cluster data density, thus hinder the accuracy of grouping results. The simulated samples are designed to reserve some extend of overlap while ensure certain discrimination between clusters, in order to achieve the purpose of evaluating the robustness of the algorithms.

**Sample 1:** In the first sample we generate \( K \) multivariate Gaussian clusters with unit variance and different location parameters. \( G_k \sim N(\mu_k, I_p) \), where \( \mu_1 \) is a \( p \)-dimensional integer vector whose elements are randomly generated in interval \((1, 10)\), and then shift the location of other clusters by \( \mu_k = \mu_1 + 2k \). Clusters are in the same size of \( n_k = n/k \).

**Sample 2:** To include some asymmetry on the basis of Sample 1, the second sample is designed as a mixture of \( K \) asymmetric normal distributions. Each cluster \( G_k \) is considered as a \( p \)-dimensional i.i.d. sub-sample, where \( G_k = (W_1, W_2, \ldots, W_p)^T \). The probability density function of \( W_j \) can be expressed as following, with \( \mu_j (j = 1, 2, \ldots, p) \) as location parameters, and \( \sigma_l, \sigma_r \) as standard deviation of two sides around \( \mu \),

\[
p(W_j | \theta) = \prod_{j=1}^{p} \sqrt{\frac{2}{\pi \sigma_l + \sigma_r}} \left\{ \begin{array}{ll}
\exp \left\{ -\frac{(W_j - \mu_j)^2}{2\sigma_l^2} \right\} & W_j < \mu_j \\
\exp \left\{ -\frac{(W_j - \mu_j)^2}{2\sigma_r^2} \right\} & W_j \geq \mu_j
\end{array} \right.,
\]

Now let \( \tau \) (\( \tau \)-expectile of the variable) be the location parameter, and \( \sigma_j \) be the overall standard deviation of the variable, the density function of asymmetric normal distribution can be rewritten as:

\[
p(Z_j | \tau, \sigma_j, \tau) = \prod_{j=1}^{p} \left( \frac{1}{2\pi(1-\tau)^{-\frac{1}{2}} \sigma_j} \right) \frac{2\sqrt{1-\tau}}{\sqrt{1-\tau} + \sqrt{\tau}} \exp \left\{ -\frac{(Z_j - \tau \sigma_j)^2}{2(1-\tau)^{-\frac{1}{2}} \sigma_j^2} \right\} Z_j < \tau
\]

\[
= \prod_{j=1}^{p} \left( \frac{1}{2\pi(1-\tau)^{-\frac{1}{2}} \sigma_j} \right) \frac{2\sqrt{\tau}}{\sqrt{1-\tau} + \sqrt{\tau}} \exp \left\{ -\frac{(Z_j - \tau \sigma_j)^2}{2(1-\tau)^{-\frac{1}{2}} \sigma_j^2} \right\} Z_j \geq \tau,
\]

which means the asymmetric normally distributed variable \( W_j^k \) can be converted from univariate Gaussian distributed variables \( Z_j^k \) according to the formula:

\[
W_j^k = \left\{ \begin{array}{ll}
\frac{2\sqrt{\tau_j^k}}{\sqrt{1-\tau_j^k} + \sqrt{\tau_j^k}} \cdot \frac{1}{\sqrt{1-\tau_j^k}} \cdot Z_j^k + \tau_j^k & Z_j^k < 0 \\
\frac{2\sqrt{1-\tau_j^k}}{\sqrt{1-\tau_j^k} + \sqrt{\tau_j^k}} \cdot \frac{1}{\sqrt{\tau_j^k}} \cdot Z_j^k + \tau_j^k & Z_j^k \geq 0
\end{array} \right.,
\]
in our sample, each $Z_j^k \sim N(0, 25)$. Parameter $\tau_j^k$ is given by using random generator with interval $[0.1, 0.9]$, and location parameter $e_{r_j}$ is randomly generated in $(0, 10)k$-th cluster, then the location of $k$-th cluster can be shifted by $e_{r_j} = e_{r_j} + 7(-1)^{(j-1)}$.

**Sample 3:** In the third sample we want to test the algorithm on skewed but not leptokurtic clusters, namely $Beta$-distributed clusters. For variables in cluster $\{k = 2c + 1, c \in \mathbb{Z}\}$, $W_j^1 \sim Beta(a_j, b_j), (j = 1, 3, \ldots, p-1)$, and in cluster $\{k = 2c, c \in \mathbb{Z}\}$, $W_j^2 \sim Beta(b_j, a_j), (j = 2, 4, \ldots, p)$. We generate parameter $a_j$ randomly from interval $(1, 10)$ and $b_j$ from interval $(10, 20)$, again $K = 3$.

**Sample 4:** For the last sample, skewed and leptokurtic clusters are being considered. We set 2 different scenarios:

- $K$ skewed generalized $t$-distributed samples. We first generated a random sample with dimension $p = 2$, parameters $df = [10, 10, 10], nc = [3,-1.5, 2.5]$, location randomly fluctuated with the difference of 0.5 around $[[0, 2], [1, 0], [0.5, 1]]$, scale $= 0.5$. And generate data repeatedly until $p$ reaches 10 and 50.
- $K$ multivariate $F$-distributed clusters. For variables in the first cluster, $W_j^1 \sim F(a_j, a_j) + 1$, and when $j = 1, 3, \ldots, p-1$; $W_j^2 \sim F(b_j, b_j) + 1$, when $j = 2, 4, \ldots, p$, where $a_j$ and $b_j$ are integers randomly selected from interval $(51, 60)$ and $(21, 30)$. In the second cluster, $W_j^2 \sim F(b_j, b_j), j = 1, 3, \ldots, p-1, W_j^2 \sim F(a_j, a_j), j = 2, 4, \ldots, p$, where $a_j$ and $b_j$ are integers randomly selected from interval $(5, 15)$ and $(25, 35)$. In the third cluster, $W_j^3 \sim F(a_j, b_j), j = 1, 3, \ldots, p-1$, and $W_j^3 \sim F(b_j, a_j), j = 2, 4, \ldots, p$, where $a_j$ and $b_j$ are integers randomly selected from interval $(15, 25)$ and $(60, 70)$.

For each of the first three samples, we evaluate combinations of $p = 50, 100, 500, n = 300, 1500$. For the last sample, $p = 2, 10, 50$. For each simulation setting, we re-generate the data 50 times and test the algorithms each round, and take the average of the Adjusted Rand Index (ARI) of the yielded classification compared with the true cluster membership. Rand Index measures the pair-wised agreement between data clustering. When it is adjusted for the chance grouping of elements, this is the Adjusted Rand Index. Given two partitions $X = X_1, X_2, \ldots, X_r$, $Y = Y_1, Y_2, \ldots, Y_s$, and the contingency table,

<table>
<thead>
<tr>
<th>$X$ \ $Y$</th>
<th>$Y_1$</th>
<th>$Y_2$</th>
<th>$\ldots$</th>
<th>$Y_s$</th>
<th>sums</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>$n_{11}$</td>
<td>$n_{12}$</td>
<td>$\ldots$</td>
<td>$n_{1s}$</td>
<td>$a_1$</td>
</tr>
<tr>
<td>$X_2$</td>
<td>$n_{21}$</td>
<td>$n_{22}$</td>
<td>$\ldots$</td>
<td>$n_{2s}$</td>
<td>$a_2$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\ddots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$X_r$</td>
<td>$n_{r1}$</td>
<td>$n_{r2}$</td>
<td>$\ldots$</td>
<td>$n_{rs}$</td>
<td>$a_r$</td>
</tr>
<tr>
<td>sums</td>
<td>$b_1$</td>
<td>$b_2$</td>
<td>$\ldots$</td>
<td>$b_s$</td>
<td>$\sum$</td>
</tr>
</tbody>
</table>

the Ajusted Rand Index is defined as:

$$ARI = \frac{\Sigma_{ij} \binom{a_{ij}}{2} - \left[ \Sigma_i \binom{a_i}{2} \right] \left[ \Sigma_j \binom{b_j}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[ \Sigma_i \binom{a_i}{2} + \Sigma_j \binom{b_j}{2} \right] - \left[ \Sigma_i \binom{a_i}{2} \right] \left[ \Sigma_j \binom{b_j}{2} \right] / \binom{n}{2}}.$$

Results are shown in Apendix, where the demonstrated values are the 100 times of ARI. We have considered other distance based clustering algorithms such as $K$-means denoted by $K$-means, spectral clustering (Shi and Malik, 2000) denoted by spectral, Ward hierarchical clustering (Ward Jr, 1963) denoted by $h$-ward, and Quantile based clustering (Hennig
et al., 2019) as well as $K$-expectile clustering with adaptive $\tau$. Note that Quantile based clustering algorithm allows quantile level (skewness parameter) $\tau$ to change variable-wisely and introduces a scale/penalty parameter. $CU, CS, VU, VS$ stands for the four modes of Quantile based clustering, corresponding to Common skewness parameter and Unscaled variables, Common skewness parameter and Scaled variable-wise, Variable-wise skewness parameter and Unscaled variables, Variable-wise skewness parameter and Scaled variable-wise. Note that spectral clustering sometimes does not work appropriate on data with outliers which lead to a not fully connected graph. This scenario can be easily occurred in a highly skewed sample or sample with large dimensionality.

From Table 7.1 we can conclude that $K$-expectile, as an algorithm that generalize $K$-means, works as good as but sometimes even better than $K$-means on spherical clusters. Meanwhile it is better than all the other clustering algorithms, incluing Ward hierarchical clustering (hierarchical clustering with the metric of Euclidean distance), spectral clustering, and quantile based clustering. For asymmetric normal distributed clusters, as Table 7.2 shows, $K$-expectile outperforms all the listed algorithms. Since the contour lines of the real distribution of the data correspond to the assumption of $K$-expectile cluster shapes, $K$-expectile yields a significantly better result than other algorithms. For more general skewed distributed clusters as demonstrated in Table 7.2, 7.3, 7.4 and 7.5, $K$-expectile still has a robust and outstanding performance.
4 Application

4.1 Application of adaptive $\tau$ clustering on CRIX and VCRIX data

An application based on the CRIX and VCRIX data is presented in this section. CRIX (CRyptocurrency IndeX) developed by Trimborn and Härdle (2018) provides a CC market price index weighted by market capitalization with a dynamic changing number of constituents of representative cryptos. The mechanism of selecting CRIX constituents with Akaike Information Criterion is introduced in the mentioned paper. VCRIX, developed by Kim et al. (2019) is a volatility index built on CRIX which offers a forecast for the mean annualized volatility of the next 30 days, re-estimated daily by using Heterogeneous Auto-Regressive (HAR) model.

The data are downloaded from thecrix.de, consists of two time series, CRIX and VCRIX, collected daily from 2017-01-02 to 2021-02-09, in total 1497 observations in two dimensions. Here we scaled the data by dividing each variable by their standard deviations to ensure the data has equal variance. The descriptive statistics and density plots of the two variables are listed as following.

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max</th>
<th>Skewness</th>
<th>Kurtosis</th>
<th>JB statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRIX</td>
<td>0.080</td>
<td>0.621</td>
<td>1.056</td>
<td>1.246</td>
<td>1.443</td>
<td>7.257</td>
<td>2.450</td>
<td>10.988</td>
<td>5481.283</td>
</tr>
<tr>
<td>VCRIX</td>
<td>0.801</td>
<td>1.814</td>
<td>2.225</td>
<td>2.458</td>
<td>2.884</td>
<td>6.565</td>
<td>1.370</td>
<td>5.360</td>
<td>816.166</td>
</tr>
</tbody>
</table>

Table 4.1: Descriptive statistics of location and dispersion for 1497 scaled data for the period from January 02, 2017 to February 09, 2021.

From Table 4.1, it is evident that neither of the two variables are normally distributed and both of them are skewed. This fact can be seen in Figure 4.1 as well, due to the longer right tail of the densities of both variables. From the plot of marginal distribution one might suspect several clusters exist.

Results of $K$-means clustering, $K$-expectile clustering with adaptive $\tau$ and Spectral clustering are shown in Figure 6.3. To select the appropriate $K$, we consider the clustering evaluation criteria including silhouette score and Davies-Bouldin score (Figure 6.1 and 6.2), both of them showed that $K = 3$ might be an appropriate choice which balances the cluster efficiency and number of clusters from a maximising similarity perspective. Hence we fix $K = 3$ and let the algorithm find the optimal location of the cluster center based on the skewness nature of the data. Our adaptive $\tau$ clustering approach yields a form of a $(K \times p)$ matrix $[[0.515, 0.448]^T, [0.222, 0.301]^T, [0.299, 0.300]^T]$, corresponding to the blue, green and grey clusters in Figure 6.3 respectively. For comparison, we consider $K$-means, $K$-expectile and spectral clustering.

Figure 2.3 shows the shapes and distribution of the three clusters on the two dimensional space consisting of CRIX and VCRIX. From the plot we can observe that the three clusters of $K$-expectile represents different types of correlation between price and volatility index. The three clusters can be described as 'low-price-low-volatility cluster', 'low-price-high-volatility cluster', and 'positively correlated price and volatility cluster', corresponding to color blue, green and grey.
It is worth noting that the grey cluster only appears shortly in the end of 2017 and from the end of 2020 till now. Positive correlation between price and volatility of crypto markets means that the volatility and price drives each other in the same direction. Higher price and higher volatility shows an 'exciting' signal other than a 'panic' expectation, this phenomenon mostly occurs in the securities market dominated by individual investors, where increased volatility is a signal of market activation. On the other hand, low-volatility cluster appears in most period of the CC market, which means CC market is highly dominated by institutional investors most of the time. High volatility means unstable market sentiment and high trading volume, and the green cluster often appears when the price start to change.

4.2 Application of fixed $\tau$ clustering on image segmentation

Image segmentation is a technique widely used in image processing, which partition an image into multiple parts sharing similar characteristics. Image segmentation includes separating foreground from background, or clustering regions of pixels based on color or shape. One of the commonly used methods in color-based segmentation is $K$-means clustering. In this case, pixel values are regarded as independent random samples in the 8 bits color space, and divided into $K$ discrete regions which has minimal variances. The output of $K$-means segmentation can be visualized by converting all the pixels in a group to the color of the cluster centers.
By applying $K$-expectile clustering, we expect a more flexible choice of centers and thus a more flexible segmentation output of the image. Besides the adaptive $\tau$ clustering approach, we also consider two ways to prespecified $\tau$, which put more weights on group-wised and dimension-wised tail behavior respectively. In particular, we set a group-specified $\tau$ as $[0.2, 0.7, 0.1, 0.9]$ to include groups emphasizes on both left tail and right tail for a choice of $K = 4$ groups. We also consider a dimension-specified $\tau$ as $[0.1, 0.8, 0.9]$ for these 3-dimensional RGB valued data to involve information on both tails. To evaluate the performance of $K$-expectile clustering, we take $K$-means clustering result as benchmark and bring Quantile based clustering results into comparison.

![Image segmentation results using different clustering methods.](image)

**Figure 4.2:** Image segmentation results using different clustering methods.
The original image is an aerial photo of Berlin, for pre-processing, we transform the image into pixel values in RGB color space, and flatten the data into a two-dimensional array. Figure 4.2 shows the original image and the segmented image, which can be considered as filtered image with 4 color clusters. Important information can be extracted from the image by displaying some clusters and mute others. The subplots showed in Figure 4.3 are image with only one cluster enabled.

Figure 4.3: Segmented image with only one cluster displayed.

To evaluate the performance of segmentation methods, we use two indices, Mean Square Error (MSE) and Peak to Signal Noise Ratio (PSNR). Given an $m \times n$ monochrome image $I$, Mean Square Error measures how much the approximation $K$ differs from it. $MSE$ is defined as:

$$MSE = \frac{1}{mn} \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \{I(i,j) - K(i,j)\}^2.$$
Peak to Signal Noise Ratio is usually used to measure the quality of the compressed image. PSNR is the proportion between maximum attainable powers and the corrupting noise that influence likeness of image. It is defined as following:

\[ PSNR = 10 \log_{10}(\frac{MAX^2}{MSE}), \]

where \( MAX_I \) is the maximum possible pixel value of the image \( I \), which equals to 225 when the sample is in 8 bits. The higher value of PSNR and the lower value of MSE, the better the fitting of the approximated image.

Table 4.2 shows the MSE and PSNR values of segmented image using multiple methods. Although MSE is usually calculated on monochrome data, here we take the average MSE on three RGB dimensions. Moreover, we convert both the original image and the segmented image from RGB data into GRAY and YCrCb color space. From the table, it can be concluded that the pre-specified parameter \( \tau \) scheme, in both senario, gives us some flexibility to customize segmentation, but the adaptive \( \tau \) scheme performs better.

<table>
<thead>
<tr>
<th>Method</th>
<th>GREY MSE</th>
<th>GREY PSNR</th>
<th>YCrCb MSE</th>
<th>YCrCb PSNR</th>
<th>RGB MSE</th>
<th>RGB PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>509.18</td>
<td>21.06</td>
<td>839.12</td>
<td>18.89</td>
<td>742.53</td>
<td>28.96</td>
</tr>
<tr>
<td>K-expectiles_vtau</td>
<td>429.47</td>
<td>21.80</td>
<td>835.66</td>
<td>18.91</td>
<td>741.17</td>
<td>28.97</td>
</tr>
<tr>
<td>CS</td>
<td>2001.30</td>
<td>15.12</td>
<td>2886.34</td>
<td>13.52</td>
<td>2841.61</td>
<td>23.14</td>
</tr>
<tr>
<td>CU</td>
<td>2217.68</td>
<td>14.67</td>
<td>3546.76</td>
<td>12.63</td>
<td>3639.46</td>
<td>22.06</td>
</tr>
<tr>
<td>VS</td>
<td>5338.85</td>
<td>10.86</td>
<td>2799.57</td>
<td>13.66</td>
<td>2398.98</td>
<td>23.87</td>
</tr>
<tr>
<td>VU</td>
<td>2030.67</td>
<td>15.05</td>
<td>3332.15</td>
<td>12.90</td>
<td>3507.04</td>
<td>22.22</td>
</tr>
<tr>
<td>K-expectiles_gp_spec_tau</td>
<td>519.81</td>
<td>20.97</td>
<td>1449.73</td>
<td>16.51</td>
<td>1203.65</td>
<td>26.87</td>
</tr>
<tr>
<td>K-expectiles_dim_spec_tau</td>
<td>876.31</td>
<td>18.70</td>
<td>1304.86</td>
<td>16.97</td>
<td>1214.55</td>
<td>26.83</td>
</tr>
</tbody>
</table>

**Table 4.2:** Performance of different clustering methods on image segmentation. Data is transformed into RGB, GREY and YCrCb space. Algorithms listed from top to bottom are K-means, K-expectile with adaptive \( \tau \), four modes of Quantile based clustering: Common skewness parameter and Scaled variable-wise, Common skewness parameter and Unscaled variables, Variable-wise skewness parameter and Scaled variable-wise, Variable-wise skewness parameter and Unscaled variables, K-expectile with group-specified \( \tau \) and K-expectile with dimension-specified \( \tau \).
References


5 APPENDIX

Proof of Proposition 1: In order to prove the convergence of the algorithm, we first recall the objective function:

\[ G^{Adaptive}(\tau, \Theta, C, X) = \sum_{k=1}^{K} \sum_{x_i \in G_k} \sum_{j=1}^{p} d(x_{ij}, \tau_{k,j}, \theta_{k,j}) \]

\[ = \sum_{k=1}^{K} \sum_{x_i \in G_k} \sum_{j=1}^{p} \{ \tau_{k,j} + (1 - 2\tau_{k,j}) I_{\{x_{ij} < \theta_{k,j}\}} \} (x_{ij} - \theta_{k,j})^2. \]

Then define

\[ \hat{\theta}_{C(i),j} = \arg \min_{\theta_j} \sum_{C(i)=k} \sum_{j=1}^{p} \{ \tau_{C(i),j} + (1 - 2\tau_{C(i),j}) I_{\{x_{ij} < \theta_j\}} \} (x_{ij} - \theta_j)^2, \]

\[ \hat{\tau}_{C(i),j} = \arg \min_{\tau_j} \sum_{C(i)=k} \sum_{j=1}^{p} \{ \tau_j + (1 - 2\tau_j) I_{\{x_{ij} < \theta_{C(i),j}\}} \} (x_{ij} - \theta_{C(i),j})^2. \]

Let \( C^{(t-1)} \) be the previous partition, \( \hat{\theta}_{k,j}^{(t-1)} \) and \( \hat{\tau}_{k,j}^{(t-1)} \) be previous estimated centroid and \( \tau \) parameters, \( C^{(t)} \) be the new partition,

\[ G(C^{(t)}) \leq \sum_{k=1}^{K} \sum_{C^{(t)}=k} \sum_{j=1}^{p} \{ \hat{\tau}_{k,j}^{(t-1)} + (1 - 2\hat{\tau}_{k,j}^{(t-1)}) I_{\{x_{ij} < \theta_j\}} \} (x_{ij} - \hat{\theta}_{k,j}^{(t-1)})^2. \]

New partition \( C^{(t)} \) minimises \( \sum_{k=1}^{K} \sum_{C^{(t)}=k} \sum_{j=1}^{p} \{ \hat{\tau}_{k,j}^{(t-1)} + (1 - 2\hat{\tau}_{k,j}^{(t-1)}) I_{\{x_{ij} < \theta_j\}} \} (x_{ij} - \hat{\theta}_{k,j}^{(t-1)})^2 \)

over all possible partitions:

\[ \leq \sum_{k=1}^{K} \sum_{C^{(t)}=k} \sum_{j=1}^{p} \{ \hat{\tau}_{k,j}^{(t-1)} + (1 - 2\hat{\tau}_{k,j}^{(t-1)}) I_{\{x_{ij} < \theta_j\}} \} (x_{ij} - \hat{\theta}_{k,j}^{(t-1)})^2. \]

Therefore, \( G(C^{(t)}) \leq G(C^{(t-1)}) \). As \( G(C^{(t)}) - G(C^{(t-1)}) \) is a monotonically nonincreasing sequence that converges to 0, we conclude that the limit point obtained from the adaptive \( \tau \) clustering algorithm is the optimal solution.
6 Figures

**Figure 6.1:** Silhouette score of $K$-expectiles clustering results with different number of clusters

[KEC_application]

**Figure 6.2:** Davies-Bouldin score of $K$-expectiles clustering results with different number of clusters

[KEC_application]
**Figure 6.3:** Clustering results of $K$-means, $K$-expectiles and Spectral clustering. The two variables are plot along x-axis and y-axis. Clusters are shown in different colours, whereas cluster centroids are shown by stars.

**Figure 6.4:** Clustering results of $K$-means, $K$-expectiles and Spectral clustering on CRIX. Clusters are shown in different colours.
Figure 6.5: Clustering results of $K$-means, $K$-expectiles and Spectral clustering on VCRIX. Clusters are shown in different colours.

KEC_application
7 Tables

For all tables below, algorithms listed from top to bottom are \( K \)-expectile with adaptive \( \tau \), \( K \)-means, Spectral clustering, Ward hierarchical clustering and four modes of Quantile based clustering: Common skewness parameter and Scaled variable-wise, Common skewness parameter and Unscaled variables, Variable-wise skewness parameter and Scaled variable-wise, Variable-wise skewness parameter and Unscaled variables, \( K \)-expectile with group-specified \( \tau \) and \( K \)-expectile with dimension-specified \( \tau \). We report the adjusted rand index (ARI).

Table 7.1: Sample 1: Simulation results of Gaussian clusters

<table>
<thead>
<tr>
<th></th>
<th>( n = 1500 )</th>
<th>( n = 300 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( p=10 )</td>
<td>( p=50 )</td>
</tr>
<tr>
<td>( \text{ARI} )</td>
<td>( 99.36 )</td>
<td>( 99.60 )</td>
</tr>
<tr>
<td>K-expectiles_vtau</td>
<td>( 99.36 )</td>
<td>( 99.60 )</td>
</tr>
<tr>
<td>K-means</td>
<td>( 99.20 )</td>
<td>( 99.60 )</td>
</tr>
<tr>
<td>Spectral</td>
<td>( 31.22 )</td>
<td>( 86.74 )</td>
</tr>
<tr>
<td>h-ward</td>
<td>( 99.24 )</td>
<td>( 99.60 )</td>
</tr>
<tr>
<td>CS</td>
<td>( 99.24 )</td>
<td>( 99.60 )</td>
</tr>
<tr>
<td>CU</td>
<td>( 99.28 )</td>
<td>( 99.60 )</td>
</tr>
<tr>
<td>VS</td>
<td>( 99.20 )</td>
<td>( 99.60 )</td>
</tr>
<tr>
<td>VU</td>
<td>( 99.24 )</td>
<td>( 99.60 )</td>
</tr>
</tbody>
</table>

Table 7.2: Sample 2: Simulation results of Asymmetric normal clusters

<table>
<thead>
<tr>
<th></th>
<th>( n = 1500 )</th>
<th>( n = 300 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( p=10 )</td>
<td>( p=50 )</td>
</tr>
<tr>
<td>( \text{ARI} )</td>
<td>( 93.22 )</td>
<td>( 99.60 )</td>
</tr>
<tr>
<td>K-expectiles_vtau</td>
<td>( 91.19 )</td>
<td>( 99.59 )</td>
</tr>
<tr>
<td>K-means</td>
<td>( 77.19 )</td>
<td>( 99.52 )</td>
</tr>
<tr>
<td>Spectral</td>
<td>( 86.61 )</td>
<td>( 99.58 )</td>
</tr>
<tr>
<td>h-ward</td>
<td>( 80.73 )</td>
<td>( 99.28 )</td>
</tr>
<tr>
<td>CS</td>
<td>( 88.86 )</td>
<td>( 99.59 )</td>
</tr>
<tr>
<td>CU</td>
<td>( 85.74 )</td>
<td>( 99.55 )</td>
</tr>
<tr>
<td>VS</td>
<td>( 85.74 )</td>
<td>( 99.55 )</td>
</tr>
</tbody>
</table>
Table 7.3: Sample 3: Simulation results of Beta-distributed clusters

<table>
<thead>
<tr>
<th>KEC_simulations</th>
<th>n = 1500</th>
<th>n = 300</th>
</tr>
</thead>
<tbody>
<tr>
<td>p = 10</td>
<td>p = 50</td>
<td>p = 100</td>
</tr>
<tr>
<td>ARI</td>
<td>ARI</td>
<td>ARI</td>
</tr>
<tr>
<td>K-expectiles_vtau</td>
<td>94.04</td>
<td>99.60</td>
</tr>
<tr>
<td>K-means</td>
<td>94.79</td>
<td>99.60</td>
</tr>
<tr>
<td>Spectral</td>
<td>93.63</td>
<td>99.60</td>
</tr>
<tr>
<td>h-ward</td>
<td>94.80</td>
<td>99.60</td>
</tr>
<tr>
<td>CS</td>
<td>68.92</td>
<td>96.89</td>
</tr>
<tr>
<td>CU</td>
<td>68.14</td>
<td>94.08</td>
</tr>
<tr>
<td>VS</td>
<td>93.28</td>
<td>97.47</td>
</tr>
<tr>
<td>VU</td>
<td>94.03</td>
<td>94.69</td>
</tr>
</tbody>
</table>

Table 7.4: Sample 4-1: Simulation results of generalized t-distributed clusters

<table>
<thead>
<tr>
<th>KEC_simulations</th>
<th>n = 1500</th>
<th>n = 300</th>
</tr>
</thead>
<tbody>
<tr>
<td>p = 2</td>
<td>p = 10</td>
<td>p = 50</td>
</tr>
<tr>
<td>ARI</td>
<td>ARI</td>
<td>ARI</td>
</tr>
<tr>
<td>K-expectiles_vtau</td>
<td>96.50</td>
<td>97.99</td>
</tr>
<tr>
<td>K-means</td>
<td>96.26</td>
<td>97.99</td>
</tr>
<tr>
<td>Spectral</td>
<td>96.21</td>
<td>26.31</td>
</tr>
<tr>
<td>h-ward</td>
<td>96.24</td>
<td>97.99</td>
</tr>
<tr>
<td>CS</td>
<td>96.48</td>
<td>97.99</td>
</tr>
<tr>
<td>CU</td>
<td>96.08</td>
<td>97.99</td>
</tr>
<tr>
<td>VS</td>
<td>96.48</td>
<td>97.99</td>
</tr>
<tr>
<td>VU</td>
<td>96.07</td>
<td>97.99</td>
</tr>
</tbody>
</table>

Table 7.5: Sample 4-2: Simulation results of F-distributed clusters

<table>
<thead>
<tr>
<th>KEC_simulations</th>
<th>n = 1500</th>
<th>n = 300</th>
</tr>
</thead>
<tbody>
<tr>
<td>p = 2</td>
<td>p = 10</td>
<td>p = 50</td>
</tr>
<tr>
<td>ARI</td>
<td>ARI</td>
<td>ARI</td>
</tr>
<tr>
<td>K-expectiles_vtau</td>
<td>95.80</td>
<td>99.60</td>
</tr>
<tr>
<td>K-means</td>
<td>95.19</td>
<td>99.60</td>
</tr>
<tr>
<td>Spectral</td>
<td>94.89</td>
<td>26.31</td>
</tr>
<tr>
<td>h-ward</td>
<td>96.82</td>
<td>99.60</td>
</tr>
<tr>
<td>CS</td>
<td>97.96</td>
<td>99.60</td>
</tr>
<tr>
<td>CU</td>
<td>95.42</td>
<td>99.60</td>
</tr>
<tr>
<td>VS</td>
<td>97.72</td>
<td>99.60</td>
</tr>
<tr>
<td>VU</td>
<td>95.44</td>
<td>99.60</td>
</tr>
</tbody>
</table>
For a complete list of Discussion Papers published, please visit http://irtg1792.hu-berlin.de.

001 "Surrogate Models for Optimization of Dynamical Systems" by Kainat Khowaja, Mykhaylo Shcherbatyy, Wolfgang Karl Härdle, January 2021.


003 "K-expected clustering" by Bingling Wang, Yingxing Li, Wolfgang Karl Härdle, March 2021.