Communications for Statistical Applications and Methods
2015, Vol. 22, No. 4, 321–331

DOI: http://dx.doi.org/10.5351/CSAM.2015.22.4.321
Print ISSN 2287-7843 / Online ISSN 2383-4757

Geodesic Clustering for Covariance Matrices

Haesung Lee[a], Hyun-Jung Ahn[b], Kwang-Rae Kim[c], Peter T. Kim[d], Ja-Yong Koo[1,e]

[a]Department of Statistics, Pennsylvania State University, USA; [b]Kantar Health, Seoul, Korea;
[c]School of Mathematical Sciences, University of Nottingham, UK;
[d]Department of Mathematics and Statistics, University of Guelph, Canada;
[e]Department of Statistics, Korea University, Korea

Abstract

The K-means clustering algorithm is a popular and widely used method for clustering. For covariance matrices, we consider a geodesic clustering algorithm based on the K-means clustering framework in consideration of symmetric positive definite matrices as a Riemannian (non-Euclidean) manifold. This paper considers a geodesic clustering algorithm for data consisting of symmetric positive definite (SPD) matrices, utilizing the Riemannian geometric structure for SPD matrices and the idea of a K-means clustering algorithm. A K-means clustering algorithm is divided into two main steps for which we need a dissimilarity measure between two matrix data points and a way of computing centroids for observations in clusters. In order to use the Riemannian structure, we adopt the geodesic distance and the intrinsic mean for symmetric positive definite matrices. We demonstrate our proposed method through simulations as well as application to real financial data.

Keywords: Euclidean distance, extrinsic mean, geodesic distance, intrinsic mean, K-means, KOSPI, Riemannian geometry, SPD, stock data

1. Introduction

The K-means clustering algorithm is a popular and widely used method for clustering, see Hartigan and Wong (1979). Typically the K-means algorithm is used with Euclidean distance in which case centroids become a component-wise means of data points in clusters. This leads to very low computational complexity and makes this procedure an attractive candidate for big data, see Xu and Wunsch (2005). However, suppose we observe covariances matrices and want to cluster the covariances. As a start one may use K-means clustering by regarding the covariances as vectors and apply the usual K-means algorithm using Euclidean distance; however, first attempt may have drawbacks since the covariances are symmetric positive definite (SPD) matrices and the space of SPD matrices has a Riemannian (non-Euclidean) geometric structure, see Schwartzman (2006).

Some works that attempt to use geometric structures include Kim *et al.* (2007) where they consider a soft geodesic kernel K-means algorithm that adapts to geodesic distance to cluster when the data have a geometric structure. Goh and Vidal (2008) proposed clustering and dimensionality reduction

The authors wish to thank Jae-Hwan Hong for help on the numerical study. Peter T. Kim's research was supported in part by NSERC RGPIN 46204-2011. Ja-Yong Koo's research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2013R1A1A2008619).

[1] Corresponding author: Department of Statistics, Korea University, 145, Anam-ro, Seongbuk-gu, Seoul 136-701, Korea. E-mail: jykoo@korea.ac.kr

Published 31 July 2015 / journal homepage: http://csam.or.kr

on Riemannian manifolds. Asgharbeygi and Maleki (2008) introduce a class of geodesic distances and extend the K-means clustering algorithm using this metric. Ai *et al.* (2010) use K-means clustering algorithm in order to cluster the summary data of different stocks by their Realized Trading Volatility (RTV) model. Jayasumana *et al.* (2013) introduce kernel methods on the Riemannian manifolds of SPD matrices. Bhattacharya and Patrangenaru (2003) develop nonparametric statistical inference procedures for measures of location of distributions on general manifolds such as spheres, real projective spaces and complex projective spaces. Fletcher and Joshi (2004) introduce methods for producing summary statistics such as averages and modes of variation, in the space of diffusion tensors, where they extend principal geodesic analysis to symmetric spaces and apply it to the computation of the variability of diffusion tensors. Diffusion tensor imaging (DTI) has become a popular *in vivo* diagnostic imaging technique in radiological sciences. Wang and Vemuri (2005) present a novel definition of tensor distance grounded in concepts from information theory and incorporate it in the segmentation of DTI.

This paper considers a geodesic clustering algorithm for data consisting of SPD matrices, utilizing the Riemannian geometric structure for SPD matrices and the idea of a K-means clustering algorithm. A K-means clustering algorithm is divided into two main steps for which we need a dissimilarity measure between two data points and a method to compute centroids for observations in clusters. We adopt the geodesic distance as the dissimilarity measure between two SPD matrices and the intrinsic mean as the centroid of observations in clusters. To the best of our knowledge we are not aware of any work on K-means clustering for SPD matrices using geodesic distance and the intrinsic mean.

The plan of the paper is as follows. Section 2 describes a brief overview about SPD matrices and describe the geodesic clustering algorithm. Section 3 illustrate the performance of the proposed method using both simulated and real examples. Concluding remarks are given in Section 4.

2. Geodesic Clustering

Denote by $\mathbb{R}^m$ and $\mathbb{R}^{m\times m}$, the set of m vectors and the set of $m \times m$ matrices, respectively. When $A^\top = A$ for $A \in \mathbb{R}^{m\times m}$, A is said to be symmetric where '$\top$' denotes the transpose. If a symmetric matrix $A \in \mathbb{R}^{m\times m}$ satisfies $x^\top Ax > 0$ for nonzero $x \in \mathbb{R}^m$, then A is an SPD matrix. The set of $m \times m$ symmetric matrices is denoted by $\mathcal{S}_m$ and the set of $m \times m$ SPD matrices by $\mathcal{P}_m$. It is known that $\mathcal{P}_m$ is a cone, see Schwartzman (2006).

Let $X^1, \ldots, X^N$ be a data set where $X^i \in \mathcal{P}_m$ for $i = 1, \ldots, N$. The problem is to partition this data of SPD matrices into K disjoint clusters $\hat{C}_1, \ldots, \hat{C}_K$ using ideas from the K-means clustering algorithm. In order to describe a K-means clustering algorithm, one need to define a similarity measure for any two data points and centroids of observations in clusters. Since $\mathcal{P}_m$ has a Riemannian structure, we can choose a similarity measure and centroids based on this structure.

Consider distances between matrices A and B where $A, B \in \mathcal{P}_m$. The Euclidean norm of A is given

$$||A|| = \sqrt{\text{tr}(A^2)},$$

where 'tr' denotes the trace operation. The Euclidean norm $||A||$ for a symmetric matrix $A \in \mathcal{S}_m$ is equal to the Frobenius norm of A. Using the Euclidean norm, we define the Euclidean distance between A and B by

$$d_E(A, B) = ||A - B||. \tag{2.1}$$

Let $A^{-1/2}$ be a matrix satisfying $A^{-1/2}A^{-1/2} = A^{-1}$. The geodesic distance between A and B is defined

by

$$d_G(A,B) = \sqrt{\sum_{i=1}^{m} \log^2 \lambda_i \left(A^{-\frac{1}{2}} B A^{-\frac{1}{2}}\right)}, \tag{2.2}$$

where $\lambda_i(X)$, $1 \le i \le m$ are eigenvalues of a $m \times m$ matrix X. Since $\lambda_i(A^{-1/2}BA^{-1/2}) > 0$ for $i = 1, \ldots, m$, the geodesic distance (2.2) is well defined for all $A, B \in \mathcal{P}_m$. Observe that the eigenvalues of $A^{-1/2}BA^{-1/2}$ are equal to those of $A^{-1}B$.

The arithematic mean $\bar{X}$ of $X^1, \ldots, X^N$ is given by

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} X^i. \tag{2.3}$$

The extrinsic sample mean $\hat{\mu}_E$ is the set of minimizers

$$\hat{\mu}_E = \underset{A \in \mathcal{P}_m}{\operatorname{argmin}}\, \hat{\sigma}_E^2(A), \tag{2.4}$$

where the extrinsic empirical variance function $\hat{\sigma}_E^2$ is defined by

$$\hat{\sigma}_E^2(A) = \frac{1}{N} \sum_{i=1}^{N} d_E^2(X^i, A).$$

The extrinsic sample mean is given by the arithematic mean $\bar{X}$, see Schwartzman (2006). The intrinsic sample mean $\hat{\mu}_I$ is the set of minimizers

$$\hat{\mu}_I = \underset{A \in \mathcal{P}_m}{\operatorname{argmin}}\, \hat{\sigma}_I^2(A), \tag{2.5}$$

where the intrinsic empirical variance function $\hat{\sigma}_I^2$ is defined by

$$\hat{\sigma}_I^2(A) = \frac{1}{N} \sum_{i=1}^{N} d_G^2(X^i, A).$$

Due to the uniqueness of the intrinsic mean, we can use the following gradient descent algorithm which is described in Cachier *et al.* (1999), Fletcher and Joshi (2004) and Schwartzman (2006). One may want to refer to, for example, Schwartzman (2006) for the definitions of Exp and Log, which respectively denote the Riemannian exponential map and the Riemannian logarithmic map in Algorithm 1.

Algorithm 1 Intrinsic mean

Input: $X^1, \ldots, X^N \in \mathcal{P}_m$

1. Set an initial value of μ_0.
2. For iteration k, compute

$$\mu_{k+1} = \mathrm{Exp}_{\mu_k}\left(\frac{1}{N} \sum_{i=1}^{N} \mathrm{Log}_{\mu_k} X^i\right).$$

3. For a specified $\epsilon > 0$, repeat step 2 until

$$\left\| \frac{1}{N} \sum_{i=1}^{N} \mathrm{Log}_{\mu_k} X^i \right\| < \epsilon. \tag{2.6}$$

Output: $\hat{\mu}_I = \mu_{\hat{k}}$ where $\mu_{\hat{k}}$ satisfies the criterion (2.6).

Algorithm 2 describes `Gclust`, the geodesic clustering algorithm, where we adopt the geodesic distance (2.2) for the similarity measure and the intrinsic mean (2.5) for the centroids in clusters. Given an initial set of K centroids, the `Gclust` proceeds by alternating between Assignment step and Update step until the assignments no longer change. We randomly choose K data matrices as an initial set of K centroids. `Gclust` algorithm stops when the sum of geodesic distances for centroids before and after update is small, say 10^{-5}.

Algorithm 2 `Gclust`

Given an initial set of K centroids, the algorithm proceeds by alternating between two steps:

Assignment step Assign each observation to the cluster whose intrinsic mean is closest in geodesic distance.

Update step Calculate the new centroids to be the intrinsic means of the observations in the new clusters.

The algorithm stops when the assignments cease to change.

3. Numerical Study

3.1. Two simulation examples

The first simulation example consists of three clusters C_i, $i = 1, 2, 3$ which corresponds to the Wishart distributions $\mathcal{W}(4, \sigma_i)$, $i = 1, 2, 3$ where

$$\sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \sigma_2 = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix} \quad \text{and} \quad \sigma_3 = \begin{bmatrix} 1 & 0.99 \\ 0.99 & 1 \end{bmatrix}.$$

Here $\mathcal{W}(k, \sigma)$ denotes the Wishart distribution with k degrees of freedom and the covariance matrix σ. We generate 50 observations from the Wishart distributions $\mathcal{W}(4, \sigma_i)$, $i = 1, 2, 3$ so that $N = 150$. In order to show that we need to consider the geometric structure of $\mathcal{P}_m$, we compare the performance of `Gclust` with that of `Eclust`, an Euclidean clustering algorithm which is a K-means clustering method using the Euclidean distance for similarity and the arithmetic mean for centroids.

Figure 1 displays the results of the first simulation, where the black ellipsoids display the cluster centers and the red ones the data points in clusters. The first row shows the true cluster centers and observations from three clusters, the second row shows the results of `Gclust` and the third row shows the results of `Eclust`. Tables 1 and 2 numerically summarize the clustering results for the first simulation. The $(i, j)^{th}$ cell of these tables denote the number of observations from C_i clustered into $\hat{C}_j$ for $i, j = 1, 2, 3$. For example, `Eclust` clusters 23 observations from C_1 into $\hat{C}_1$ so that the number

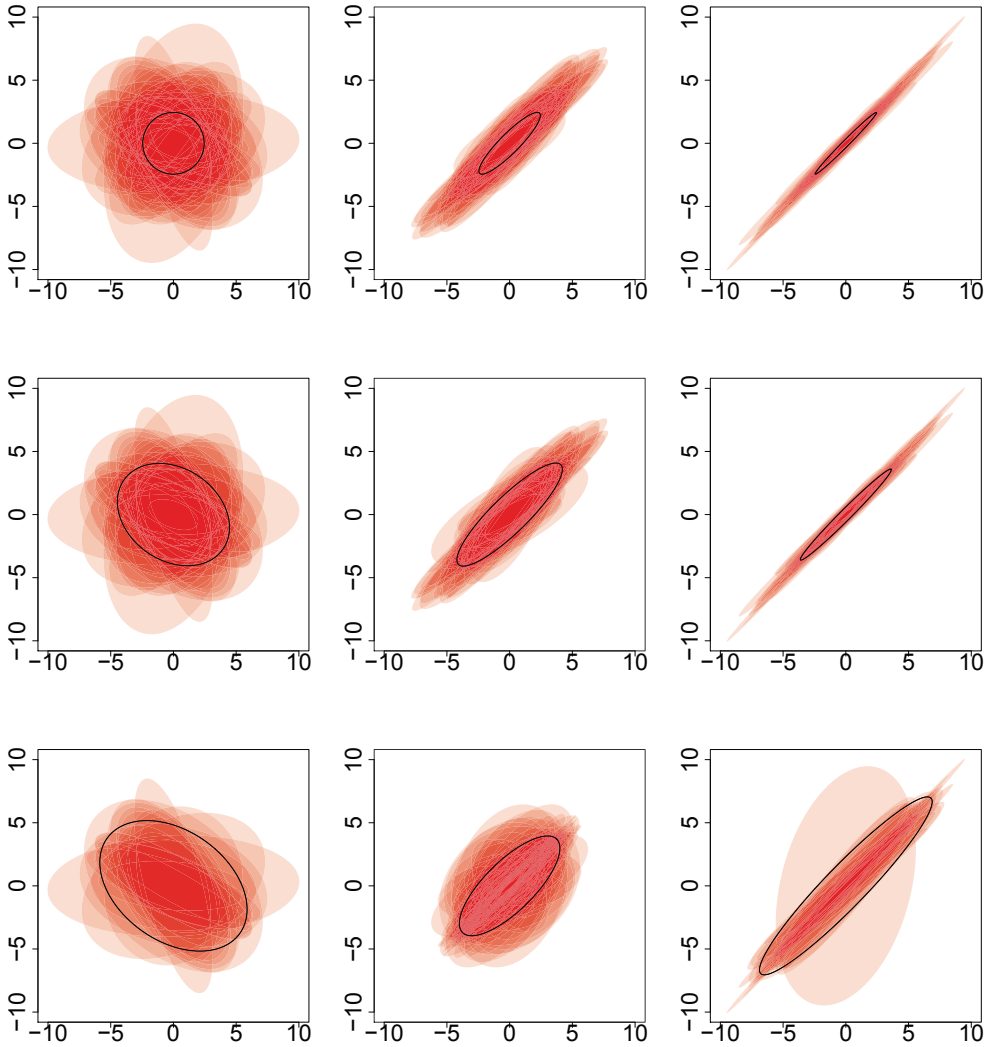

Figure 1: *Clustering results for the first simulation. Each ellipsoid corresponds to an observation or a centroid. The first row displays the observations with true cluster centroids, second row displays the clustering result by* `Gclust` *and third row displays the clustering result by* `Eclust`.

of the correctly classified observations is 23. It appears that `Gclust` performs better than `Eclust` especially in detecting the third cluster whose covariance matrix provides relatively bigger correlation coefficients. Presumably this is because the `Gclust` does consider the geometric structure of SPD matrices whereas `Eclust` regards SPD observations as vectors in Euclidean space.

We consider a more complicated example where the diagonal elements of the covariance matrices may be different. Let

$$\sigma_1 = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}, \quad \sigma_2 = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}, \quad \sigma_3 = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}, \quad \sigma_4 = \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix}.$$

Table 1: Numerical summary of the clustering results for `Eclust`. The $(i, j)^{th}$ cell denotes the number of observations from C_i clustered into $\hat{C}_j$ for $i, j = 1, 2, 3$

	$\hat{C}_1$	$\hat{C}_2$	$\hat{C}_3$
C_1	**23**	1	26
C_2	0	**17**	33
C_3	0	12	**38**

Table 2: Numerical summary of the clustering results for `Gclust`. The $(i, j)^{th}$ cell denotes the number of observations from C_i clustered into $\hat{C}_j$ for $i, j = 1, 2, 3$

	$\hat{C}_1$	$\hat{C}_2$	$\hat{C}_3$
C_1	**40**	10	0
C_2	0	**42**	8
C_3	0	0	**50**

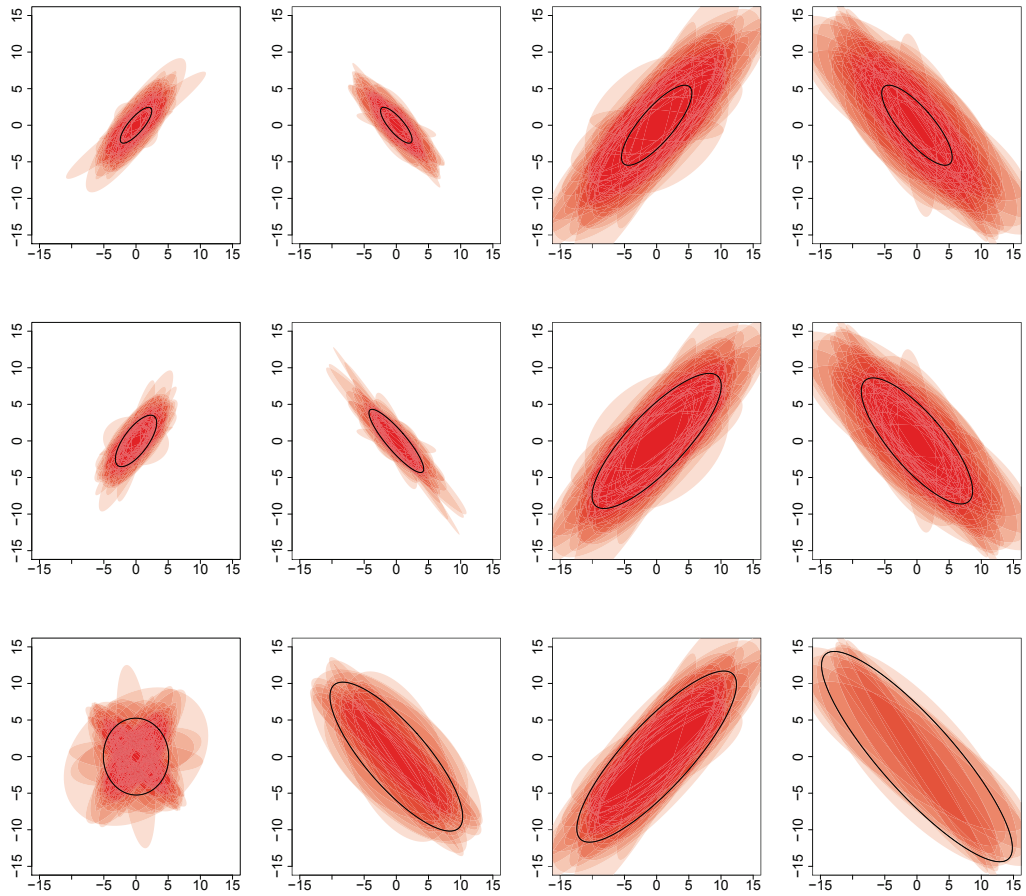

Figure 2: *Clustering results for the second simulation. Each ellipsoid corresponds to an observation or a centroid. The first row displays the observations with true cluster centroids, second row displays the clustering result by* `Gclust` *and third row displays the clustering result by* `Eclust`.

Table 3: Numerical summary of the clustering result for `Eclust`. The $(i, j)^{th}$ cell denotes the number of observations from C_i clustered into $\hat{C}_j$ for $i, j = 1, \ldots, 4$

	$\hat{C}_1$	$\hat{C}_2$	$\hat{C}_3$	$\hat{C}_4$
C_1	**49**	0	1	0
C_2	49	**1**	0	0
C_3	12	0	**38**	0
C_4	12	26	0	**12**

Table 4: Numerical summary of the clustering result for `Gclust`. The $(i, j)^{th}$ cell denotes the number of observations from C_i clustered into $\hat{C}_j$ for $i, j = 1, \ldots, 4$

	$\hat{C}_1$	$\hat{C}_2$	$\hat{C}_3$	$\hat{C}_4$
C_1	**46**	1	3	0
C_2	6	**40**	0	4
C_3	3	0	**46**	1
C_4	1	6	0	**43**

Table 5: Average running time in seconds for 100 repeated simulation of `Eclust` and `Gclust`

	Simulation 1	Simulation 2
Eclust	0.0243	0.0411
Gclust	5.5871	10.9572

The second simulation consists of four clusters C_i that correspond to the Wishart distributions $\mathcal{W}(4, \sigma_i)$, where $i = 1, \ldots, 4$. A random sample of size 50 is simulated for each cluster so that $N = 200$.

Figure 2 displays the results for this simulation, where the black ellipsoids display the cluster centers and the red ones the data points in clusters. The first row shows the true cluster centers and observations from three clusters, the second row the result of `Gclust` and the third row the result of `Eclust`. Tables 3 and 4 numerically summarize the clustering results for the simulation. The $(i, j)^{th}$ cell of these tables denote the number of observations from C_i clustered into $\hat{C}_j$ for $1 \leq i, j \leq 4$. Again, it seems that `Gclust` performs better than `Eclust` especially in detecting the clusters whose covariance matrix have relatively bigger correlation coefficients because the `Gclust` does consider the geometric structure of SPD matrices whereas `Eclust` regards SPD observations as vectors.

In order to compare the computational cost of `Gclust` and `Eclust`, we repeat 100 times `Gclust` and `Eclust` for the two simulation examples. Table 5 shows average running time in seconds for 100 repetitions of both methods; therefore, the `Gclust` demands relatively more computational time than `Eclust`.

3.2. Real data

A case study on applying `Gclust` to covariance matrices of stock data is provided. We consider the covariance matrix of stock price and trading volume in the Korean stock market. The variance of the logarithmic yield corresponds to the stock price volatility and the variance of the turnover rate of the stock volume volatility. We cluster the covariance matrices of logarithmic yields and turnover rates of stocks over specified periods. Covariances are SPD; therefore, `Gclust` may outperform `Eclust` as exhibited in the simulation results in Section 3.1, especially in the context of correlation structure.

The Korea Composite Stock Price Index (KOSPI) 200 index consists of 200 large companies the Korean stock market allows us to classify companies into several industrial sectors (Table 6). The yields and the turnover rates of stocks are observed for stocks included in the KOSPI 200 index during the period from March 18 to March 22 in 2013, where these numbers are obtained from the Korea

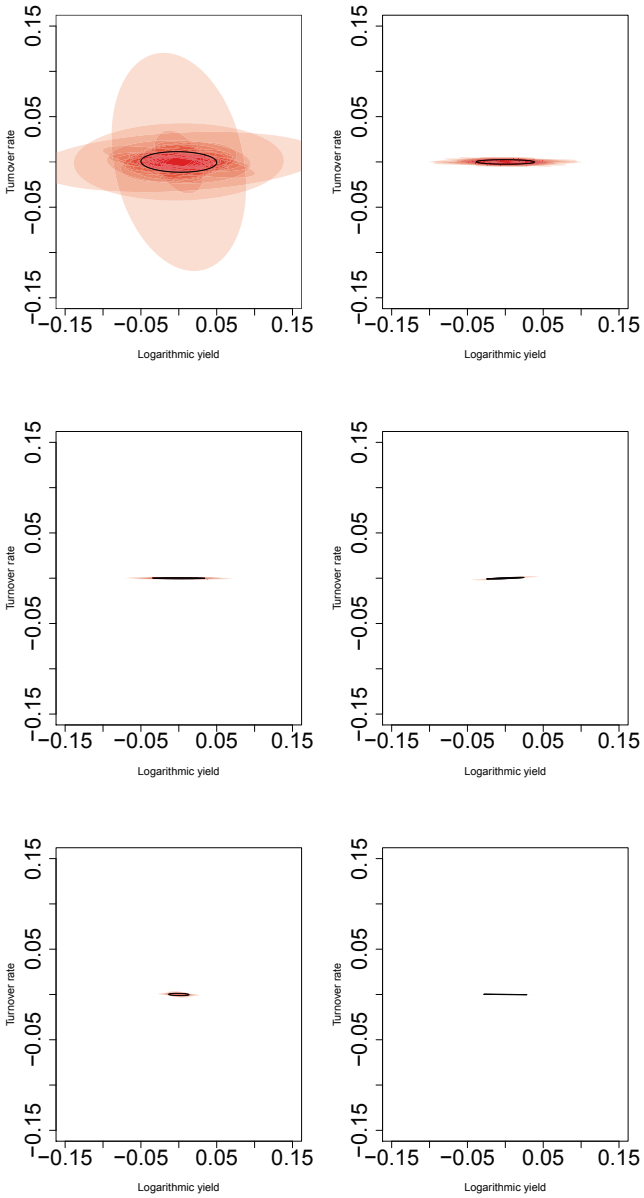

Figure 3: *The clustering results of Korea Composite Stock Price Index (KOSPI) stock data. The ellipsoids in other clusters appear linear since the stocks have high volatilities in cluster 1 and 2. We need to also adjust the scale of axis of turnover rate.*

Exchange website (`http://www.krx.co.kr`). We will examine $N = 199$ stocks (one stock out of the 200 stocks did not have transaction records during the period). Let $y_i(t)$ denote the logarithmic yield defined by

$$y_i(t) = \log p_i(t) - \log p_i(t-1),$$

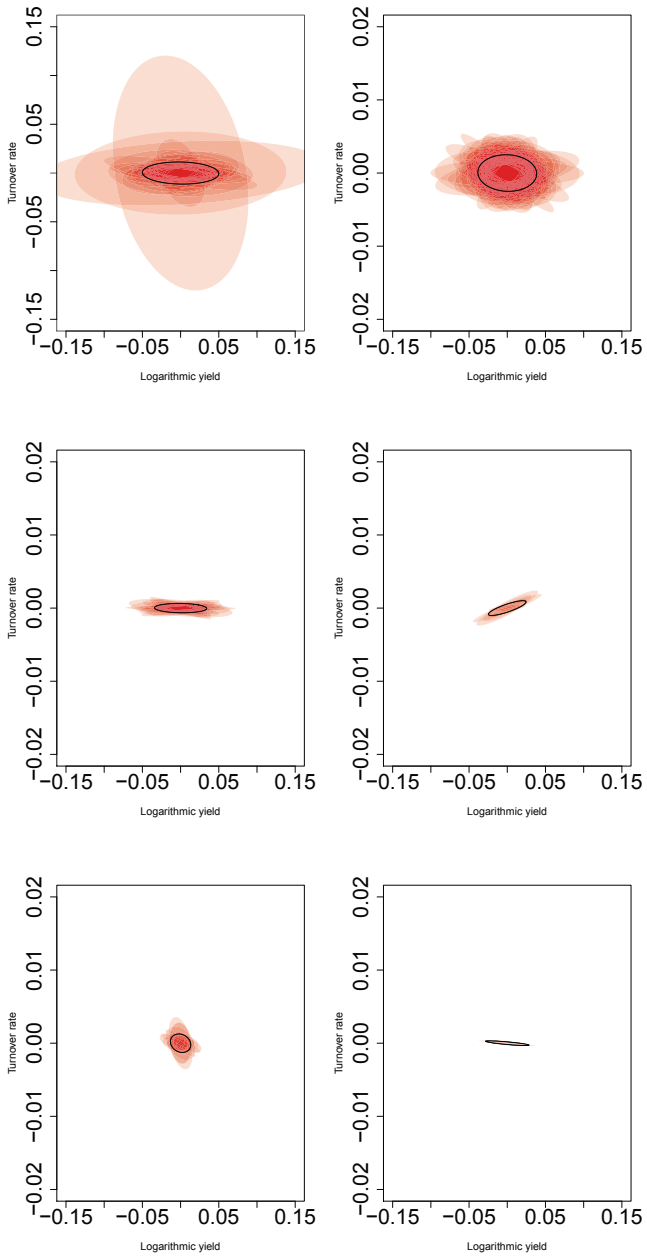

Figure 4: *The clustering result adjusted for the scale of axis for the turnover rate. In terms of correlation, each cluster has a specific volatility and correlation.*

where $p_i(t)$ denotes the settlement price for the i^{th} stock on the t^{th} day for $i = 1, \ldots, 199$, $t = 1, \ldots, 5$. Here $p_i(0)$ is given by the settlement price of the i^{th} stock on March 17 and the days for $t = 1, \ldots, 5$ respectively correspond to Monday, Tuesday, Wednesday, Thursday and Friday. The turnover rate

Table 6: Industries of 199 stocks in KOSPI. They can be grouped into 8 classes.

Industry	Frequency	Industry	Frequency
Manufacturing	108	Service	24
Transportation	24	Financial	17
Medicine	10	Retail	8
Construction	5	Telecommunications	3

Table 7: Summary of the clustering result of the geodesic method, where n is the number of stocks in each cluster, L indicates large-cap stocks and M is mid & small-cap stocks.

Cluster	n	Volatility	Correlation	Stocks
1	32	High	Weak	L-transportation, M-manufacturing
2	20	Mid	Weak	L-financial, M-manufacturing
3	32	Small	Negative	L-service, L-retail
4	21	Small	Positive	L-service, M-manufacturing
5	44	Very small	Negative	L-financial, M-manufacturing
6	50	Very small	Weak	L-transportation, L-service, manufacturing

$r_i(t)$ of the i^{th} stock on the t^{th} day is defined by

$$r_i(t) = \frac{q_i(t)}{Q_i},$$

where Q_i denotes the number of circulating shares of the i^{th} stock and $q_i(t)$ the turnover of the i^{th} stock on the t^{th} day. The covariance matrices X^i, $i = 1, \ldots, 199$ of the logarithmic yields $y_i(t)$ and the turnover rates $r_i(t)$ over the week corresponding to the period $t = 1, \ldots, 5$ are computed.

We present the clustering results for $K = 6$ which are a reasonable choice among several number of clusters. Figure 3 displays the clustering result of `Gclust` and Table 7 provides numerical summaries. The ellipsoids in other clusters appear linear since the stocks clustered in cluster 1 and 2 have high price and volume volatilities. We adjust the scale of axis of turnover rate in Figure 4 to examine the characteristics of each cluster.

We can check the characteristics of each cluster through Figure 4. In the geodesic method, high volatility stocks are only in cluster 1 and mid volatility stocks are in cluster 2. The stocks in cluster 3 tend to be negatively correlated and in cluster 4 tend to have positive correlation between price and volume generally. The last two clusters include very small volatility stock. The stocks in cluster 5 tend to be negatively correlated between price and volume, however, the correlation coefficients of stocks in cluster 6 are close to zero.

There is a classification of stocks which is determined according to company size based on total market value in KOSPI among the many classifications of stocks. Commonly, stocks which are included from the ranking of 1 to 100 are called large-capital stocks, and from 101 to the end are called mid & small-capital stocks. Table 7 summarizes the clustering result and general classification. Finally, we can check that the geodesic method could classify each cluster according to each characteristic better than classical K-means.

4. Concluding Remarks

We present an iterative relocation algorithm for clustering covariance matrices which considers the Riemannian geometric structure of SPD matrices. We consider a geodesic distance measure between two data points and intrinsic mean as the centroid of observations in clusters. Using both simulation

and real data sets, the geodesic clustering classifies each cluster according to each characteristic better than the Euclidian method; therefore, it would be useful to apply the `Gclust` algorithm to DTI data.

References

Ai, X. W., Hu, T., Li, X. and Xiong, H. (2010). Clustering high-frequency stock data for trading volatility analysis, In *Proceedings of 9th International Conference on Machine Learning and Applications (ICMLA)*, Washington, DC, 333–338.

Asgharbeygi, N. and Maleki, A. (2008). Geodesic k-means clustering, In *Proceedings of 19th International Conference on Pattern Recognition (ICPR 2008)*, Tampa, FL, 1–4.

Bhattacharya, R. and Patrangenaru, V. (2003). Large sample theory of intrinsic and extrinsic sample means on manifolds. I, *Annals of Statistics*, **31**, 1–29.

Cachier, P., Pennec, X. and Ayache, N. (1999). *Fast non rigid matching by gradient descent: Study and improvements of the "demons" algorithm*, RR-3706, Available from: https://hal.inria.fr/inria-00072962/

Fletcher, P. T. and Joshi, S. (2004). Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In M. Sonka, et al. (Eds.), *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, Springer, Heidelberg, 87–98.

Goh, A. and Vidal, R. (2008). Clustering and dimensionality reduction on Riemannian manifolds, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2008)*, Anchorage, AK, 1–7.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm, *Journal of the Royal Statistical Society Series C (Applied Statistics)*, **28**, 100–108.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., and Harandi, M. (2013). Kernel methods on the Riemannian manifold of symmetric positive definite matrices, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2013)*, Portland, OR, 73–80.

Kim, J., Shim, K. H. and Choi, S. (2007). Soft geodesic kernel k-means, In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2007)*, Honolulu, HI, 429–432.

Schwartzman, A. (2006). *Random ellipsoids and false discovery rates: Statistics for diffusion tensor imaging data (Doctoral dissertation)*, Stanford University, CA.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms, *IEEE Transactions on Neural Networks*, **16**, 645–678.

Wang, Z. and Vemuri, B. C. (2005). DTI segmentation using an information theoretic tensor dissimilarity measure, *IEEE Transactions on Medical Imaging*, **24**, 1267–1277.

Received April 2, 2015; Revised June 20, 2015; Accepted June 20, 2015