

SPECTRAL CLUSTERING AND LONG TIMESERIES CLASSIFICATION

NADEZDA SUKHORUKOVA , JAMES WILLARD-TURTON ,
GEORGINA GARWOLI, CLAIRE MORGAN  and ALINA ROKEY 

(Received 12 December, 2023; accepted 20 April, 2024; first published online 18 September, 2024)

Abstract

Clustering is a method of allocating data points in various groups, known as clusters, based on similarity. The notion of expressing similarity mathematically and then maximizing it (minimize dissimilarity) can be formulated as an optimization problem. Spectral clustering is an example of such an approach to clustering, and it has been successfully applied to visualization of clustering and mapping of points into clusters in two and three dimensions. Higher dimension problems remained untouched due to complexity and, most importantly, lack of understanding what “similarity” means in higher dimensions. In this paper, we apply spectral clustering to long timeseries EEG (electroencephalogram) data. We developed several models, based on different similarity functions and different approaches for spectral clustering itself. The results of the numerical experiment demonstrate that the created models are accurate and can be used for timeseries classification.

2020 Mathematics subject classification: primary 90C90; secondary 05C90, 68T10.

Keywords and phrases: spectral clustering, similarity functions, timeseries classification.

1. Introduction

Clustering is the process of assigning “similar” points into the same group (cluster), while the points that are not “similar” form a different cluster. This technique is very popular in many fields where some form of data analysis is required, such as computer science, biology, engineering, finance and many others.

Clustering can be formulated as an optimization problem, where the dissimilarity function is minimized. This is a very natural view, but there are two main obstacles.

Swinburne University of Technology, John Street, Hawthorn, Victoria 3128, Australia;
e-mail: nsukhorukova@swin.edu.au, 101738797@student.swin.edu.au,
102107026@student.swin.edu.au, 101445343@student.swin.edu.au, 100998875@student.swin.edu.au
© The Author(s), 2024. Published by Cambridge University Press on behalf of Australian Mathematical Publishing Association Inc. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<https://creativecommons.org/licenses/by/4.0>), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

First of all, it is not an easy task to identify the efficient similarity function. In some cases, it is assumed that the distance between the points can be taken as their measure of dissimilarity, but it is still a problem to choose a type of measure (for example, Euclidean distance, Manhattan distance and so forth). Second, even when the dissimilarity measure is known, it is not always easy to minimize the corresponding objective functions. Finally, in the case of clustering-based classification, there is an additional problem: how to define the cluster prototype, which can be used for creating classification rules.

Spectral clustering is just one of many clustering approaches. This method relies on spectral properties of matrices (eigenvalues). The mathematical background of spectral clustering has its origins in the so-called spectral graph theory [3]. There are several types of spectral clustering. We will discuss them in Section 3. Spectral clustering is a very popular tool for neural scientists due to its simplicity and efficiency when dimension is low [4, 5]. When the dimension is increasing, it is much harder to define the notion of similarity. This was the main obstacle in the application of spectral clustering to high-dimensional data, especially timeseries.

The main contribution of this paper is to demonstrate that spectral clustering can be applied to high-dimensional timeseries. The key is to choose the similarity measure and the cluster prototype (cluster centre) correctly. It was also demonstrated that, in some cases, solving simpler problems (dimension reduction) leads to more accurate models [9, 13, 18]. This is a very important phenomenon and it was observed in many practical applications: when the exact optimization problems are complex, it may be better to work with approximate models and optimize these models instead. Essentially, if the obtained local minimum for a complex optimization problem is not “deep enough”, it is better to find a “deep” local minimum for a suitable approximation of the original model. This will work if the approximations are accurate and simple at the same time. Therefore, the construction of these models is the most important step in dealing with such problems. Examples of approximation-based approaches for dimension reduction can be found in [13, 18, 19]. Essentially, approximations play the role of a filter that removes unnecessary noise in data.

In this paper, we demonstrate that spectral clustering can be applied to electroencephalogram (EEG) (brain wave) analysis. We use a publicly available dataset, collected by the epileptic centre at the University of Bonn, Germany [1]. This dataset consists of 500 signal segments, each signal segment contains 4097 recordings. Application of spectral clustering to such long timeseries is particularly challenging, since it is hard to identify a suitable similarity function. Indeed, a standard approach of treating timeseries as points in \mathbf{R}^n does not make any distance-based similarity measures always efficient. This is especially clear for the dataset we use ($n = 4097$). Data can be downloaded from their website¹. More details are provided in Section 4.

¹https://ebrary.net/59044/education/details_public_databases

The paper is organized as follows. In Section 2, we provide a more detailed background of clustering problems and their connection with optimization. In Section 3, we provide the mathematical background for spectral clustering. Section 4 is dedicated to the numerical experiments. Finally, Section 5 contains conclusions and further research directions.

2. Clustering background

Clustering is a process of assigning similar points into groups, called clusters. Cluster analysis is used to identify the structure of data.

The k -means algorithm is a fast and efficient clustering method that groups points in \mathbf{R}^n . This method (and its name) was first proposed in 1967 by MacQueen [10]. Mathematically, this method is based on the minimization of the sum of squares of the Euclidean distances between the points and the cluster centres to which these points are assigned. This function is called the dissimilarity function which is the objective function in the corresponding optimization problems. This group of methods also includes a number of closely related methods (for example, Manhattan distance based k -median). We refer to the work of Bagirov and Ugon [2] and Späth [15] for further references and details.

Another clustering method that is also popular is the k -medoids. The k -medoids problems are known to be NP-complete [12], but therefore, there are a number of modifications of this method, where the dissimilarity function is based on other types of distances [8] and, essentially, such methods solve certain modifications of the original k -medoid problem (that is, approximations of the original problems that are accurate and easier to solve). A comprehensive review of such methods can be found in [7].

The clustering approach can be applied to signals and timeseries. In this paper, the terms “timeseries” and “signals” are unchangeable, since all the timeseries we study are coming from EEG signals (brainwaves). There are two main approaches for representing timeseries:

- (1) each time moment is an independent dimension and, therefore, each timeseries of length n is just a point in \mathbf{R}^n ;
- (2) a prototype curve for each group of similar timeseries.

In this paper, we apply the former. The inspiration for this study comes from the work of Woods et al. [16], where the authors applied standard clustering techniques: k -means [10] and k -medoid [7, 8] to timeseries classification. They used the same dataset and, in this paper, we compare their results and ours. It may appear that standard clustering approaches of grouping points in \mathbf{R}^n completely ignore the fact that the timeseries values that are close in time are strongly related to each other. However, the corresponding optimization problems are simple and can be applied to high-dimensional problems. The results of numerical experiments in [16] demonstrate that these approaches are reasonably accurate. In the current paper, we

continue working in a similar direction, applying spectral clustering algorithms to preprocessed timeseries. Our preprocessing aims at extracting essential features for further classification. The results of our numerical experiments demonstrate spectral clustering classification models, combined with preprocessing, are more accurate than the models developed by Sukhorukova and Kelly [16].

REMARK 1. A prototype curve-based approach can be applied if some key characteristics are known. These characteristics can be provided by experts in the field (for example, medical doctors).

3. Optimization, spectral clustering and classification problems

3.1. Optimization behind spectral clustering Spectral clustering is a technique that uses graph Laplacian matrices. These Laplacians are constructed from similarity graphs. There exists a rich area of spectral graph theory [3]. This approach was successfully applied in many application areas, including human brain studies [4] and latent structure models [14]. An excellent modern review of the area of spectral clustering and applications (other than clustering of brain signals) can be found in the paper by Mondal et al. [11]. The idea is to connect similar points and therefore create a graph (also known as a similarity graph). Then, for this graph, we construct certain matrices (adjacency, Laplacian) and then the spectrum of these matrices is used to establish clusters.

A similarity graph is a graph where each node is a data point (x_1, \dots, x_n) , and the edge that joins two nodes x_i and x_j ($i \neq j, i, j = 1, \dots, n$) is weighted as S_{ij} , where S_{ij} is the similarity measure between these two points. It is usually assumed that the similarity measure S_{ij} is symmetric ($S_{ij} = S_{ji}$). A detailed tutorial on spectral clustering methods, mathematical theory behind them and practical implementations can be found in [17].

In this study, we use three types of spectral clustering algorithms:

- unnormalized Laplacian (L_{unnorm});
- symmetric normalized Laplacian (L_{sym});
- random walk normalized Laplacian (L_{rw}).

All three algorithms (approximately) solve specific optimization problems, whose objectives are to minimize the dissimilarity within clusters and maximize dissimilarities between the clusters, (see [17] for details and further references). Therefore, all these approaches are based on mathematical optimization, statistics and graph theory.

3.2. Similarity functions To create a similarity function (SF), we use unlabelled training data points (that is, the class labels are removed). Assume that the number of points is n . In this study, we use the following similarity functions.

SF1 is a simple similarity function; however, it treats every time-moment as an independent dimension. In particular, it does not reflect the fact that similar timeseries that have a small time-shift in the recording may appear as very different waves. Therefore,

Algorithm 1: SF1 (Squared Euclidean distance)

1. **Input** : Data points x_1, \dots, x_n .
 2. **Compute:**
 - the dissimilarity matrix \mathcal{B} , where each element β_{ij} is the squared Euclidean distance between the points x_i and x_j ;
 - the similarity threshold is $\alpha = \max_{i,j=1,\dots,n} \{\beta_{ij}\}$; the similarity matrix S with elements $S_{ij} = \alpha - \beta_{ij}$.
 3. **Output** : Similarity matrix, S .
-

we also used other similarity measures that take into account “segment”-based features, rather than just isolated time-moments. These “segment”-based features may include signal frequency, amplitude, variation and so forth. It appeared in our experiments that “variation-based” features that can be obtained from simple computations of max and min lead to accurate separations between classes.

For the remaining SFs, the data segments are trimmed: we use 4000 time-moments, cutting the last 97 time-moments. The main reason for this trimming is that each trimmed segment of 4000 time-moments is divided into 40 subsegments and each subsegment contains 100 time-moments.

REMARK 2. Other trimming methods are also possible, but we recommend that each trimmed segment contains consecutive time-moments.

Algorithm 2: SF2 (Average of sampled maximal differences)

1. **Input** : Data points x_1, \dots, x_n .
 2. **Trimming:** Each time segment x_i is trimmed to 4000 time-moments (cutting off the last 97 elements, for ease of calculations) and divided into 40 subsegments y_{ij} , $i = 1, \dots, n$, $j = 1, \dots, 40$ (100 time-moments each).
 3. **Compute** : for each subsegment y_{ij} ($i = 1, \dots, n$, $j = 1, \dots, 40$), find the maximum (M_{ij}) and the minimum (m_{ij}), then take the difference: $\Delta_{ij} = M_{ij} - m_{ij}$;
 - $\Delta_i = \frac{\sum_{j=1}^{40} \Delta_{ij}}{40}$, $i = 1, \dots, n$;
 - the dissimilarity matrix \mathcal{B} with elements $\beta_{ij} = |\Delta_i - \Delta_j|$, $i, j = 1, \dots, n$;
 - the similarity threshold is $\alpha = \max_{i,j=1,\dots,n} \beta_{ij}$;
 - the similarity matrix with elements $S_{ij} = \alpha - \beta_{ij}$.
 4. **Output** : Similarity matrix, S .
-

Algorithm 3: SF3 (Average sampled max and min plus Euclidean distance)

1. **Input** : Data points x_1, \dots, x_n .
2. **Trimming:** Each time segment x_i is trimmed to 4000 time-moments (cutting off the last 97 elements, for ease of calculations) and divided into 40 subsegments y_{ij} , $i = 1, \dots, n$, $j = 1, \dots, 40$ (100 time-moments each).

3. Compute :

- for each subsegment y_{ij} ($i = 1, \dots, n$, $j = 1, \dots, 40$), find the maximum (M_{ij}) and the minimum (m_{ij}), take the average over j for each i and construct the pair:

$$[\Delta_i, \delta_i] = \left[\frac{\sum_{j=1}^{40} M_{ij}}{40}, \frac{\sum_{j=1}^{40} m_{ij}}{40} \right];$$

- the dissimilarity matrix \mathcal{B} with elements β_{ij} is the matrix whose components are computed as the Euclidean distance between $[\Delta_i, \delta_i]$ and $[\Delta_j, \delta_j]$;
- the similarity threshold is $\alpha = \max_{i,j=1,\dots,n} \beta_{ij}$;
- the similarity matrix with elements $S_{ij} = \alpha - \beta_{ij}$.

4. **Output** : Similarity matrix, S .
-

Algorithm 4: SF4 (Average sampled max and min plus Manhattan distance)

1. **Input** : Data points x_1, \dots, x_n .
2. **Trimming:** Each time segment x_i is trimmed to 4000 time-moments (cutting off the last 97 elements, for ease of calculations) and divided into 40 subsegments y_{ij} , $i = 1, \dots, n$, $j = 1, \dots, 40$ (100 time-moments each).

3. Compute :

- for each subsegment y_{ij} ($i = 1, \dots, n$, $j = 1, \dots, 40$), find the maximum (M_{ij}) and the minimum (m_{ij}), take the average over j for each i and construct the pair:

$$[\Delta_i, \delta_i] = \left[\frac{\sum_{j=1}^{40} M_{ij}}{40}, \frac{\sum_{j=1}^{40} m_{ij}}{40} \right];$$

- the dissimilarity matrix \mathcal{B} with elements β_{ij} is the matrix whose components are computed as Manhattan (L_1) distance between $[\Delta_i, \delta_i]$ and $[\Delta_j, \delta_j]$:

$$\beta_{ij} = |M_i - M_j| + |m_i - m_j|;$$

- the similarity threshold is $\alpha = \max_{i,j=1,\dots,n} \beta_{ij}$;
- the similarity matrix with elements $S_{ij} = \alpha - \beta_{ij}$.

4. **Output** : Similarity matrix, S .
-

Essentially, for SF2, each timeseries x_i , $i = 1, \dots, n$ is substituted by a single number Δ_i . This is a considerable reduction in the dimension.

Next, for SF3, each timeseries (4000 scalars) is replaced by two numbers:

$$i\text{th entry} \rightarrow [\Delta_i, \delta_i].$$

Then the Euclidean distance is used to compute the dissimilarity matrix.

The construction of the pair $[\Delta_i, \delta_i]$ for the i th timeseries within the dataset is identical for SF3 and SF4. The difference comes in the construction of \mathcal{B} : SF3 uses the Euclidean distance, while SF4 is based on the Manhattan (L_1) distance.

3.3. Classification

Overall classification procedure. Data classification follows a number of rules (general framework).

- (1) The dataset is divided into two parts: training and test sets.
- (2) The classification rules are developed on the training set.
- (3) The classification rules developed at Step 2 are applied to the points from the test set. The classification accuracy is the proportion (percentage) of correctly classified instances from the test set.

For more information on data classification framework, we refer the reader to [6]. This is an excellent textbook on data classification and its mathematical background (mostly, algebra, analysis and optimization).

In this paper, the classification is based on clustering. We find clusters in each class of the training set. The classification rule is to assign a new point (test set) to the “nearest” cluster. One common approach is to use cluster centres. This is a very natural approach, but in the case of spectral clustering, the shapes of the clusters may be very different from “ball-like” clusters. In this study, we use three different approaches for cluster centre identification (cluster prototype methods). In combination with four different similarity functions and three options for the graph Laplacian, we have $3 \times 4 \times 3 = 36$ different approaches.

Cluster prototype method 1 (CPM1): Each cluster is divided into two sub-clusters: timeseries that start with a positive number (A^+) and that start with a non-positive number (A^-). For each sub-cluster, find the centre as the barycentre of the points assigned to the sub-cluster. Both sub-centres are treated as cluster centres.

The main reason for the division into two sub-clusters is that “similar” signals may appear to be different if the recording started at different phases. The division into two groups is a simple attempt to take this into account.

Cluster prototype method 2 (CPM2): Apply trimming as in SF2. For each timeseries i of the cluster, compute Δ_i as in SF2 and take the average Δ . Then Δ is treated as the cluster centre.

It is clear that CPM2 reduces the dimension. Therefore, if this method is used, the same preprocessing (trimming and computation of Δ) should be applied to the test data points.

Cluster prototype method 3 (CPM3): Apply trimming as in SF3. For each timeseries i of the cluster, compute Δ_i and δ_i as in SF3 and take the average $[\Delta, \delta]$. Then $[\Delta, \delta]$ is treated as the cluster centre.

Similar to CPM2, CPM3 reduces the dimension and requires additional preprocessing for test set points.

4. Numerical experiments

4.1. Dataset description In our study, we use EEG timeseries (also known as brain waves), from publicly available data. This dataset was prepared by the epileptic centre at the University of Bonn [1]. There are five classes in this dataset: 1, 2, 3, 4 and 5. Each class contains 100 timeseries of 23.6 seconds recordings with a sampling frequency of 173.61 Hz (that is, 4097 points per timeseries).

Class 1 and class 2 of this dataset were used by Peiris et al. [13], where the authors applied a more advanced and time-consuming optimization-based approach for data classification. The models in [13] are comparable with ours in terms of the classification results (class 1 and class 2), but our approach is simple and fast. Moreover, our approach is accurate for separating all five classes.

Classes 1 and 2 were recorded from healthy volunteers: awake with eyes open (class 1) and awake with eyes closed (class 2). Classes 3 and 4 correspond to the patients who are seizure free during the recording, but had a seizure in the past and it occurred in the opposite hemisphere of the brain (class 3) or in the same hemisphere of the brain (class 4). So, the meaning of separating of class 3 and class 4 volunteers essentially means the detection of which brain hemisphere was affected by seizure. Finally, class 5 timeseries correspond to the active stage of seizure.

In this paper, we study pair-wise class separation rather than the direct separation of all five classes. This is how it was done by Peiris et al. [13], and Sukhorukova and Kelly [16]. Each class (100 timeseries) was divided into the training set (top 75 timeseries of each class) and the test set (final 25 timeseries of each class).

4.2. Results In all our numerical experiments, we use MATLAB. The code and table with all the classification results (in addition to Table 1) are available on GitHub¹.

Table 1 represents the best test set classification accuracy, obtained by the 36 combinations (four similarity functions, three clustering prototype methods and three graph Laplacians). In some cases, more than one combination achieved the highest accuracy. Column two corresponds to the minimal number of clusters per class that are required. A larger number of clusters may lead to the improvement in the

¹<https://github.com/jt-404/Spectral-Clustering-for-the-classification-of-long-time-series.git>

TABLE 1. Best classification results for all combinations of similarity functions, clustering methods and graph Laplacians.

Classes	Clusters per class	Graph Laplacian	Similarity Function	Clustering method	Accuracy
1,2	4	L_{unnorm}	SF1, SF2, SF4	CPM3	94%
1,3	3	L_{rw}	SF4	CPM1	84%
1,4	7	L_{sym}	SF3, SF4	CPM1	84%
1,5	4	L_{sym}	SF1-4	CPM2	100%
2,3	10	L_{sym}	SF1	CPM1	82%
2,4	6	L_{sym}	SF1	CPM1	82%
2,5	8	L_{rw}	SF1, SF2, SF3	CPM3	100%
3,4	6	L_{sym}	SF1	CPM1	86%
3,5	6	L_{sym}	SF1, SF2, SF3	CPM3	100%
4,5	8	L_{rw}	SF1, SF2, SF3	CPM3	98%

classification accuracy, but we refer to the results from the simplest model. From Table 1, the classification accuracy was significantly improved, compared with the results of Wood et al. [16], where the classification accuracy was approximately 70–80% (direct application of the k -means method and k -medoid). From the results, it is not very clear which combination of methods is best, but there are a number of possible recommendations.

- (1) In most cases, the best classification result was reached at more than one similarity function.
- (2) The only case when L_{unnorm} produced best classification results was in the case when the recording was performed on healthy volunteers and the task was to detect if their eyes are open or not.
- (3) L_{sym} performed well in most cases. L_{rw} was not so efficient, but was performing well when one of the classes was class 5 (active seizure).
- (4) The choice of the clustering prototype method is the hardest. The only clear pattern is that if class 5 is involved, then CPM3 was performing well.
- (5) In most cases, it was not very hard to separate classes if one of them was class 5. Other combinations of classes are much harder to separate.
- (6) In the absence of the information as to which similarity measure and clustering prototype methods to use, we suggest the following procedure:
 - if class 5 (active seizure) is one of the classes to separate, use CPM2 or CPM3 (any similarity function and graph Laplacian);
 - if the task is to separate class 1 and class 2 (healthy individuals, eyes are open or closed), use CPM2 or CPM3 and L_{unnorm} ;
 - in all other combinations, it is recommended to use the Euclidean distance as the similarity measure and clustering prototype method 1 (CPM1), where

the cluster is split into two sub-clusters and the barycentre of each sub-cluster is used as the prototype and L_{sym} . These problems appeared to be difficult to separate and, in most cases, we had to find more than five clusters in each class.

Therefore, in most cases, simpler models are accurate, despite the size reduction. Also, it was not always observed that the increase in the number of clusters leads to more accurate classification models.

5. Conclusions and further research directions

In this paper, we apply spectral clustering to the classification of long timeseries EEG brain waves. The classification tasks were performed pair-wise (separation between pairs of classes), and the accuracy between 82% and 100% was achieved for all pairs. The classification accuracy achieved a substantial improvement compared with [16]. Results clearly demonstrate that spectral clustering can be a suitable method to apply to long timeseries EEG data, but in most cases, the choice of similarity measures and the choice of the cluster prototype are the key factor.

Squared Euclidean distance was shown to be an effective similarity function for these timeseries, though not unique in classifying with high accuracy. Furthermore, the symmetric Laplacian (L_{sym}) generally yielded better results in comparison to the unnormalized and the random walk Laplacians, though the difference was marginal.

Future work will include testing models using various dimension reductions in combination with k -means. Exploring the use of Fourier transforms and more complex similarity functions would also be useful.

Acknowledgments

Nadezda Sukhorukova was supported by the Australian Research Council (ARC): Solving hard Chebyshev approximation problems through nonsmooth analysis (Discovery Project DP180100602). This paper was inspired by discussions and presentations at MATRIX Program: Mathematics of the Interactions between Brain Structure and Brain Functions (November 2022).

References

- [1] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state", *Phys. Rev. E* **64** (2001) Article ID: 061907; doi:10.1103/PhysRevE.64.061907.
- [2] A. Bagirov and J. Ugon, "Nonsmooth DC programming approach to clusterwise linear regression: optimality conditions and algorithms", *Optim. Methods Softw.* **33** (2018) 194–219; doi:10.1080/10556788.2017.1371717.
- [3] F. R. K. Chung, *Spectral graph theory* (American Mathematical Society, Providence, RI, 1997) 1–29; <https://api.semanticscholar.org/CorpusID:60624922>.

- [4] C. Craddock, A. James, P. Holtzheimer, X. Hu and H. Mayberg, “A whole brain fMRI atlas generated via spatially constrained spectral clustering”, *Hum. Brain Mapp.* **33** (2012) 1914–1928; doi:[10.1002/hbm.21333](https://doi.org/10.1002/hbm.21333).
- [5] K. Dillon and Y.-P. Wang, “Resolution-based spectral clustering for brain parcellation using functional MRI”, *J. Neurosci. Methods* **335** (2020) Article ID: 108628; doi:[10.1016/j.jneumeth.2020.108628](https://doi.org/10.1016/j.jneumeth.2020.108628).
- [6] I. Goodfellow, Y. Bengio and A. Courville, *Deep learning* (MIT Press, Cambridge, MA, 2016) 1–773; <http://www.deeplearningbook.org>.
- [7] T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning data mining, inference, and prediction*, Springer Ser. Statist. (Springer, New York, 2008) 1–745; doi:[10.1007/978-0-387-84858-7](https://doi.org/10.1007/978-0-387-84858-7).
- [8] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis* (John Wiley & Sons Inc., Antwerp, Belgium, 1990); doi:[10.1002/9780470316801](https://doi.org/10.1002/9780470316801).
- [9] S. E. Lazic, “Why we should use simpler models if the data allow this: relevance for anova designs in experimental biology”, *BMC Physiol.* **8** (2008) Article ID: 125560; doi:[10.1186/1472-6793-8-16](https://doi.org/10.1186/1472-6793-8-16).
- [10] J. MacQueen, “Some methods for classification and analysis of multivariate observations”, in: *Proceedings of 5th Berkeley symposium on mathematical statistics and probability, Volume 5.1* (eds. L. M. Le Cam and J. Neyman) (University of California Press, California, 1967) 281–297; <https://api.semanticscholar.org/CorpusID:6278891>.
- [11] R. Mondal, E. Ignatova, D. Walke, D. Broneske, G. Saake and R. Heyer, “Clustering graph data: the roadmap to spectral techniques”, *Discov. Artif. Intell.* **4** (2024) Article ID: 7; doi:[10.1007/s44163-024-00102-x](https://doi.org/10.1007/s44163-024-00102-x).
- [12] C. Papadimitriou, “Worst–case and probabilistic analysis of a geometric location problem”, *SIAM J. Comput.* **10** (1981) 542–557; doi:[10.1137/0210040](https://doi.org/10.1137/0210040).
- [13] V. Peiris, N. Sharon, N. Sukhorukova and J. Ugon, “Generalised rational approximation and its application to improve deep learning classifiers”, *Appl. Math. Comput.* **389** (2021) Article ID: 125560; doi:[10.1016/j.amc.2020.125560](https://doi.org/10.1016/j.amc.2020.125560).
- [14] F. Sanna Passino and N. A. Heard, “Latent structure blockmodels for Bayesian spectral graph clustering”, *Stat. Comput.* **32** (2022) 22; doi:[10.1007/s11222-022-10082-6](https://doi.org/10.1007/s11222-022-10082-6).
- [15] H. Späth, *Cluster analysis algorithms for data reduction and classification of objects* (Ellis Horwood Limited, Chichester, 1980) 1–226; <https://books.google.com.au/books?id=4ofgAAAAMAAJ>.
- [16] N. Sukhorukova and L. Kelly, “*k*-means clustering in EEG (brain waves) timeseries”, in: *2021–2022 MATRIX annals* (eds. D. Wood, J. de Gier, C. Praeger and T. E. Tao) (Springer, Cham, 2024) 1–7; https://www.matrix-inst.org.au/wp_Matrix2016/wp-content/uploads/2023/07/Sukhorukova.pdf.
- [17] U. von Luxburg, “A tutorial on spectral clustering”, *Stat. Comput.* **17** (2007) 395–416; doi:[10.1007/s11222-007-9033-z](https://doi.org/10.1007/s11222-007-9033-z).
- [18] Z. R. Zamir and N. Sukhorukova, “Linear least squares problems involving fixed knots polynomial splines and their singularity study”, *Appl. Math. Comput.* **282** (2015) 204–215; doi:[10.1016/j.amc.2016.02.011](https://doi.org/10.1016/j.amc.2016.02.011).
- [19] Z. R. Zamir, N. Sukhorukova, H. Amiel, A. Ugon and C. Philippe, “Optimization-based features extraction for K-complex detection”, in: *Proceedings of the 11th Biennial Engineering Mathematics and Applications Conference, EMAC-2013* (eds. M. Nelson, T. Hamilton, M. Jennings and J. Bunder) (Queensland University of Technology, Brisbane, Australia, 2013) C384–C398; doi:[10.21914/anziamj.v55i0.7802](https://doi.org/10.21914/anziamj.v55i0.7802).